



Διεθνές Πανεπιστήμιο της Ελλάδος
Πολυτεχνική Σχολή
Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Ανάπτυξη και Αξιολόγηση Web Client για Σχεσιακές Βάσεις Δεδομένων με Υποστήριξη Ερωτημάτων Φυσικής Γλώσσας μέσω LLM



Φοιτητής:

Μαργαρίτα Δεληγιαννίδη
Αριθμός Μητρώου: 2021034

Επιβλέπων:

Στέφανος Ουγιάρογλου

4 Σεπτεμβρίου 2026

Ανάπτυξη και Αξιολόγηση Web Client για Σχεσιακές Βάσεις Δεδομένων με Υποστήριξη
Ερωτημάτων Φυσικής Γλώσσας μέσω LLM

Κωδικός Δ.Ε.: 26173

Ονοματεπώνυμο Φοιτητή: Μαργαρίτα Δεληγιαννίδη

Ονοματεπώνυμο εισηγητή: Στέφανος Ουγιάρογλου

Ημερομηνία ανάληψης: 18-03-2026

Ημερομηνία περάτωσης: 04-09-2026

Βεβαιώνω ότι είμαι ο συγγραφέας αυτής της εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία. Επίσης, έχω καταγράψει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών, εικόνων και κειμένου, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επιπλέον, βεβαιώνω ότι αυτή η εργασία προετοιμάστηκε από εμένα προσωπικά, ειδικά ως διπλωματική εργασία, στο Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του ΔΙ.ΠΑ.Ε.

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή Μαργαρίτα Δεληγιαννίδη που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης, ο συγγραφέας/δημιουργός εκχωρεί στο Διεθνές Πανεπιστήμιο της Ελλάδος άδεια χρήσης του δικαιώματος αναπαραγωγής, δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσης της εργασίας διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος. Η ανοικτή πρόσβαση στο πλήρες κείμενο της εργασίας, δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού, ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, πώληση, εμπορική χρήση, διανομή, έκδοση, μεταφόρτωση (downloading), ανάρτηση (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού.

Η έγκριση της διπλωματικής εργασίας από το Τμήμα Μηχανικών Πληροφορικής και Ηλεκτρονικών Συστημάτων του Διεθνούς Πανεπιστημίου της Ελλάδος, δεν υποδηλώνει απαραίτητα και αποδοχή των απόψεων του συγγραφέα, εκ μέρους του Τμήματος.

Αφιέρωση

Θα ήθελα αρχικά να ευχαριστήσω την οικογένειά μου για τη συνεχή στήριξη, την αγάπη και την ενθάρρυνση που μου προσέφερε όλα αυτά τα χρόνια των σπουδών μου. Η παρουσία και η υποστήριξή της υπήρξαν πολύ σημαντικές σε όλη αυτή την πορεία.

Ιδιαίτερες ευχαριστίες θα ήθελα να εκφράσω στον επιβλέποντα της διπλωματικής μου εργασίας, Επίκουρο Καθηγητή κ. Στέφανο Ουγιάρογλου, για την καθοδήγηση, τη βοήθεια και τις πολύτιμες παρατηρήσεις του κατά την εκπόνηση της παρούσας εργασίας.

Abstract

The increasing volume of data and the widespread use of relational databases create a need for efficient and accessible ways of retrieving information. However, interacting with relational databases through SQL requires knowledge of both SQL syntax and database structure, making this process difficult for users with limited technical expertise. Recent advances in Large Language Models (LLMs) have enabled the automatic generation of SQL queries from natural language, providing new possibilities for more accessible database interaction. This diploma thesis presents the development and evaluation of AskSQL, a web-based application that enables users to interact with relational databases using either SQL or natural language queries. The application utilizes LLMs to analyze natural language requests and automatically generate the corresponding SQL queries, which can then be executed against the connected database.

AskSQL provides a user-friendly environment for database connection management, SQL query execution, and result visualization, supporting relational database management systems such as MySQL and PostgreSQL. The application is designed to accommodate both experienced database users and users with limited SQL knowledge. Its effectiveness is evaluated experimentally through natural language query sets, examining different prompting approaches and LLMs for SQL generation. The usability of the application is also evaluated using the System Usability Scale (SUS). The work demonstrates the potential of combining LLM-based NL2SQL techniques with web-based database management systems to simplify interaction with relational data.

Περίληψη

Η συνεχής αύξηση του όγκου των δεδομένων και η ευρεία χρήση σχεσιακών βάσεων δεδομένων δημιουργούν την ανάγκη για αποδοτικούς και εύκολους τρόπους πρόσβασης στην πληροφορία. Ωστόσο, η αλληλεπίδραση με σχεσιακές βάσεις δεδομένων μέσω της γλώσσας SQL απαιτεί εξειδικευμένες γνώσεις τόσο στη σύνταξη της γλώσσας όσο και στη δομή της βάσης δεδομένων, γεγονός που μπορεί να δυσκολέψει χρήστες με περιορισμένη τεχνική κατάρτιση. Η πρόσφατη εξέλιξη των Μεγάλων Γλωσσικών Μοντέλων (Large Language Models - LLMs) έχει δημιουργήσει νέες δυνατότητες για την κατανόηση ερωτημάτων φυσικής γλώσσας και την αυτόματη παραγωγή SQL ερωτημάτων. Στην παρούσα διπλωματική εργασία παρουσιάζεται η ανάπτυξη και αξιολόγηση της AskSQL, μιας διαδικτυακής εφαρμογής που επιτρέπει στους χρήστες να αλληλεπιδρούν με σχεσιακές βάσεις δεδομένων χρησιμοποιώντας είτε SQL είτε φυσική γλώσσα.

Η εφαρμογή αξιοποιεί LLMs για την ανάλυση των ερωτημάτων φυσικής γλώσσας και την αυτόματη παραγωγή των αντίστοιχων SQL ερωτημάτων, τα οποία μπορούν στη συνέχεια να εκτελεστούν στη συνδεδεμένη βάση δεδομένων. Παρέχονται επίσης λειτουργίες διαχείρισης συνδέσεων βάσεων δεδομένων, εκτέλεσης SQL ερωτημάτων και προβολής αποτελεσμάτων, με υποστήριξη συστημάτων MySQL και PostgreSQL. Η αποτελεσματικότητα της εφαρμογής αξιολογείται πειραματικά μέσω συνόλων ερωτημάτων φυσικής γλώσσας, εξετάζοντας διαφορετικές τεχνικές prompting και LLMs ως προς την παραγωγή SQL. Παράλληλα, αξιολογείται η ευχρηστία της εφαρμογής μέσω της κλίμακας System Usability Scale (SUS).

Περιεχόμενα

Αφιέρωση	ii
Περίληψη	iv
Περιεχόμενα	v
Κατάλογος Σχημάτων	vi
Κατάλογος Πινάκων	vii
1 Εισαγωγή	1
1.1 Σχεσιακές Βάσεις Δεδομένων και SQL	1
1.2 Μετατροπή φυσικής γλώσσας σε SQL (Natural Language to SQL)	3
1.3 Μεγάλα Γλωσσικά Μοντέλα για μετατροπή φυσικής γλώσσας σε SQL	5
1.4 Κίνητρο	6
1.5 Συνεισφορά	7
1.6 Οργάνωση της Διπλωματικής	8
2 Ανασκόπηση της βιβλιογραφίας	10
2.1 Η εξέλιξη του Text-to-SQL	10
2.2 Datasets και Benchmarks	11
2.3 Σύγχρονες προσεγγίσεις και τεχνικές Prompting	12
2.4 Προκλήσεις των Text-to-SQL συστημάτων	12
2.5 Υφιστάμενα εργαλεία και συσχέτιση με την παρούσα εργασία	13
3 Τεχνολογίες	15
3.1 HTML5	15
3.2 CSS3	15
3.3 Bootstrap	16
3.4 JavaScript	16
3.5 PHP	17
3.6 MySQL	17
3.7 PostgreSQL	17
3.8 Μεγάλα Γλωσσικά Μοντέλα	18
3.9 Web Speech API	18
3.10 Google Authentication	19
3.11 GitHub	19
4 Πειραματικές μετρήσεις	20
4.1 Βάσεις δεδομένων	20

4.1.1	Congestion Charging	20
4.1.2	Chinook	21
4.1.3	Hotel Room Reservation	22
4.2	Σύνολο ερωτημάτων αξιολόγησης	23
4.3	Τεχνικές prompting	29
4.4	Μεγέθη μέτρησης	29
4.5	Σύγκριση τεχνικών prompting	30
4.6	Σύγκριση μοντέλων LLM	31
5	Σχεδίαση και Υλοποίηση του AskSQL	35
5.1	Λειτουργικές Απαιτήσεις	35
5.2	Αρχιτεκτονική του AskSQL	37
5.3	Χρήστες της Εφαρμογής	39
5.4	Υλοποίηση του Backend	39
5.4.1	Σχεδίαση της βάσης δεδομένων	39
5.4.2	Web API	43
5.4.3	Επικοινωνία με τα Μεγάλα Γλωσσικά Μοντέλα (LLMs)	44
5.4.4	Διαχείριση βάσεων δεδομένων	47
5.5	Υλοποίηση Frontend	51
6	Παρουσίαση του AskSQL	60
6.1	Σύνδεση	60
6.2	Εγγραφή	60
6.3	Ανάκτηση κωδικού	60
6.4	Διαχείριση συνδέσεων	61
6.5	Δημιουργία νέας σύνδεσης	61
6.6	Κεντρική σελίδα	62
6.7	Διαχείριση λογαριασμού	63
6.8	Σελίδα About	63
6.9	Διαχείριση χρηστών	63
7	Αξιολόγηση του AskSQL	72
7.1	System Usability Scale (SUS)	72
7.2	Αποτελέσματα Αξιολόγησης	73
8	Συμπεράσματα και μελλοντικές επεκτάσεις	74

Κατάλογος σχημάτων

1.1	Βασική διαδικασία μετατροπής ερωτήματος φυσικής γλώσσας σε SQL και ανά-κτησης των αποτελεσμάτων.	4
4.1	Διάγραμμα ER της βάσης Congestion Charging	21
4.2	Διάγραμμα ER της βάσης Chinook	33
4.3	Διάγραμμα ER της βάσης Hotel Room Reservation	34
5.1	Διάγραμμα ροής του AskSQL.	54
5.2	Αρχιτεκτονική του AskSQL.	55
5.3	Entity-Relationship (ER) διάγραμμα της βάσης δεδομένων του AskSQL.	56
5.4	Αρχεία HTML Κώδικα για τις Σελίδες της Εφαρμογής	58
5.5	Αρχεία CSS Κώδικα για τις Σελίδες της Εφαρμογής	58
5.6	Αρχεία JavaScript Κώδικα για τις Σελίδες της Εφαρμογής	59
6.1	Σελίδα σύνδεσης	61
6.2	Σύνδεση μέσω λογαριασμού Google	62
6.3	Σελίδα εγγραφής	64
6.4	Σελίδα forgot	65
6.5	Σελίδα για επαναφορά κωδικού	65
6.6	Σελίδα με τις αποθηκευμένες συνδέσεις	66
6.7	Νέα σύνδεση σε βάση δεδομένων	67
6.8	Εκτέλεση ερωτημάτων σε SQL mode	68
6.9	Εκτέλεση ερωτημάτων σε NL mode	69
6.10	Μήνυμα επιβεβαίωσης για την εκτέλεση επικίνδυνου ερωτήματος	69
6.11	Φόρμα αλλαγής κωδικού πρόσβασης	70
6.12	Σελίδα about	70
6.13	Εμφάνιση όλων των εγγεγραμμένων χρηστών	71
6.14	Αλλαγή δικαιωμάτων χρήστη	71

Κατάλογος πινάκων

1.1	Βασικές εντολές SQL	2
4.1	Ερωτήματα αξιολόγησης για τη βάση δεδομένων Chinook.	23
4.2	Ερωτήματα αξιολόγησης για τη βάση δεδομένων Congestion Charging.	26
4.3	Ερωτήματα αξιολόγησης για τη βάση δεδομένων Hotel Room Reservation.	27
4.4	Αποτελέσματα σύγκρισης των τεχνικών prompting	30
4.5	Σύγκριση των γλωσσικών μοντέλων	31
5.1	Πίνακας users της βάσης δεδομένων	40
5.2	Πίνακας password_resets της βάσης δεδομένων	41
5.3	Πίνακας connections της βάσης δεδομένων	43
5.4	Πίνακας query_history της βάσης δεδομένων	44
5.5	Public API Endpoints της εφαρμογής	45
5.6	Protected API Endpoints της εφαρμογής	57
7.1	Ερωτήσεις του ερωτηματολογίου System Usability Scale (SUS).	72
7.2	Τελική βαθμολογία SUS ανά συμμετέχοντα.	73

Κεφάλαιο 1

Εισαγωγή

1.1 Σχεσιακές Βάσεις Δεδομένων και SQL

Οι βάσεις δεδομένων αποτελούν βασικό στοιχείο των σύγχρονων πληροφοριακών συστημάτων, καθώς χρησιμοποιούνται καθημερινά από οργανισμούς υγείας, επιχειρήσεις, πανεπιστήμια και διαδικτυακές υπηρεσίες για την αποθήκευση και διαχείριση μεγάλου όγκου δεδομένων. Η συνεχής αύξηση των παραγόμενων δεδομένων δημιούργησε την ανάγκη ανάπτυξης αποδοτικών μηχανισμών οργάνωσης, αποθήκευσης και ανάκτησης πληροφορίας [1, 2].

Ως δεδομένα ορίζονται τα ακατέργαστα στοιχεία τα οποία δεν έχουν υποστεί επεξεργασία ώστε να προκύψει χρήσιμη πληροφορία. Η πληροφορία αποτελεί το αποτέλεσμα της επεξεργασίας των δεδομένων με τρόπο που να αποκαλύπτεται το νόημα και η χρησιμότητά τους για τον τελικό χρήστη. Μία βάση δεδομένων αποτελεί μία οργανωμένη συλλογή δεδομένων που σχετίζονται μεταξύ τους με λογικό τρόπο και μπορεί να περιλαμβάνει τόσο δεδομένα τελικού χρήστη όσο και μεταδεδομένα, δηλαδή δεδομένα που χρησιμοποιούνται για την περιγραφή και διαχείριση των αποθηκευμένων πληροφοριών [1, 2].

Οι σχεσιακές βάσεις δεδομένων (Relational Databases) αποτελούν το επικρατέστερο μοντέλο αποθήκευσης δομημένων δεδομένων. Το σχεσιακό μοντέλο διατυπώθηκε για πρώτη φορά το 1970 από τον Edgar F. Codd και βασίζεται στη θεωρία συνόλων και στην κατηγορηματική λογική. Βασική αρχή του μοντέλου είναι η λογική αναπαράσταση των δεδομένων μέσω σχέσεων (relations), οι οποίες υλοποιούνται με τη μορφή πινάκων [1, 2].

Ένας πίνακας αποτελεί μία δισδιάστατη δομή που περιλαμβάνει γραμμές και στήλες. Κάθε γραμμή αντιστοιχεί σε μία εγγραφή, ενώ κάθε στήλη αντιπροσωπεύει ένα συγκεκριμένο χαρακτηριστικό της οντότητας. Ως οντότητα μπορεί να θεωρηθεί ένα πρόσωπο, ένα αντικείμενο, ένα γεγονός ή μία έννοια για την οποία απαιτείται η αποθήκευση πληροφοριών. Οι σχέσεις μεταξύ διαφορετικών πινάκων πραγματοποιούνται μέσω κλειδιών, όπως τα πρωτεύοντα και τα ξένα κλειδιά, τα οποία συμβάλλουν στη μοναδική αναγνώριση των εγγραφών και στη διατήρηση της ακεραιότητας των δεδομένων [1].

Για τη διαχείριση και επεξεργασία των δεδομένων χρησιμοποιούνται Συστήματα Διαχείρισης Βάσεων Δεδομένων (Database Management Systems - DBMS). Ένα DBMS αποτελεί μία συλλογή προγραμμάτων που επιτρέπει τη δημιουργία, οργάνωση και διαχείριση βάσεων δεδομένων, λειτουργώντας ως ενδιάμεσο επίπεδο μεταξύ του χρήστη και της βάσης δεδομένων. Επιπλέον, επιτρέπει την ταυτόχρονη πρόσβαση πολλών χρηστών στα δεδομένα και διευκολύνει την παραγωγή δυναμικών ερωτημάτων και την ανάκτηση πληροφορίας [1, 2].

Η επικοινωνία με τα σχεσιακά συστήματα διαχείρισης βάσεων δεδομένων πραγματοποιείται κυρίως μέσω της γλώσσας SQL (Structured Query Language). Η SQL αποτελεί την πρότυπη γλώσσα διαχείρισης σχεσιακών βάσεων δεδομένων και παρέχει δυνατότητες δημιουργίας, τροποποίησης και επεξεργασίας δεδομένων. Μέσω της SQL μπορούν να πραγματοποιηθούν λειτουργίες όπως η δημιουργία πινάκων, η εισαγωγή και ενημέρωση δεδομένων, η διαγραφή εγγραφών καθώς και η ανάκτηση πληροφορίας μέσω ερωτημάτων [2].

Οι εντολές της SQL διακρίνονται σε διαφορετικές κατηγορίες, όπως η γλώσσα ορισμού δεδομένων (Data Definition Language - DDL), η γλώσσα χειρισμού δεδομένων (Data Manipulation Language - DML) και η γλώσσα ελέγχου δεδομένων (Data Control Language - DCL). Οι εντολές DDL χρησιμοποιούνται για τη δημιουργία και τροποποίηση αντικειμένων της βάσης δεδομένων, όπως πίνακες και όψεις, ενώ οι εντολές DML χρησιμοποιούνται για την εισαγωγή, τροποποίηση, διαγραφή και ανάκτηση δεδομένων. Αντίστοιχα, οι εντολές DCL σχετίζονται με τη διαχείριση δικαιωμάτων πρόσβασης και τον έλεγχο ασφαλείας της βάσης δεδομένων [2].

Ενδεικτικά, η εντολή SELECT χρησιμοποιείται για την ανάκτηση δεδομένων από έναν ή περισσότερους πίνακες, η INSERT για την εισαγωγή νέων εγγραφών, η UPDATE για την τροποποίηση δεδομένων και η DELETE για τη διαγραφή εγγραφών. Η ευρεία υιοθέτηση της SQL οφείλεται στην τυποποίηση, τη φορητότητα και την ευκολία ενσωμάτωσής της σε διαφορετικά συστήματα διαχείρισης βάσεων δεδομένων [1].

Πίνακας 1.1: Βασικές εντολές SQL

Εντολή	Περιγραφή
SELECT	Ανάκτηση δεδομένων από έναν ή περισσότερους πίνακες.
INSERT	Εισαγωγή νέων εγγραφών σε έναν πίνακα.
UPDATE	Τροποποίηση υπαρχόντων δεδομένων.
DELETE	Διαγραφή εγγραφών από έναν πίνακα.
CREATE	Δημιουργία αντικειμένων βάσης δεδομένων, όπως πίνακες και βάσεις δεδομένων.
ALTER	Τροποποίηση της δομής ενός υπάρχοντος πίνακα.
DROP	Διαγραφή αντικειμένων της βάσης δεδομένων.

Ανάμεσα στα πιο διαδεδομένα συστήματα διαχείρισης σχεσιακών βάσεων δεδομένων συγκαταλέγονται τα MySQL και PostgreSQL, τα οποία χρησιμοποιούνται ευρέως τόσο σε ακαδημαϊκά όσο και σε επαγγελματικά περιβάλλοντα. Το MySQL DBMS είναι ιδιαίτερα δημοφιλές σε διαδικτυακές εφαρμογές λόγω της απλότητας και της υψηλής απόδοσής του, ενώ το PostgreSQL προσφέρει προηγμένες δυνατότητες διαχείρισης δεδομένων και υποστήριξη σύνθετων ερωτημάτων [3, 4]. Στην παρούσα εργασία χρησιμοποιούνται τεχνολογίες σχεσιακών βάσεων δεδομένων για την υλοποίηση και εκτέλεση SQL ερωτημάτων μέσω της εφαρμογής AskSQL.

Παρά τη μεγάλη διάδοση και τις δυνατότητες που προσφέρει η SQL, η αποτελεσματική συγγραφή ερωτημάτων απαιτεί εξειδικευμένες γνώσεις σχετικά με τη δομή της βάσης δεδομένων και τη σύνταξη της γλώσσας. Ως αποτέλεσμα, πολλοί χρήστες δυσκολεύονται να αλληλεπιδρά-

σουν άμεσα με τις βάσεις δεδομένων χωρίς τεχνικές γνώσεις. Η ανάγκη απλοποίησης της διαδικασίας αυτής οδήγησε στην ανάπτυξη συστημάτων μετατροπής φυσικής γλώσσας σε SQL (Natural Language to SQL), τα οποία αναλύονται στην επόμενη ενότητα.

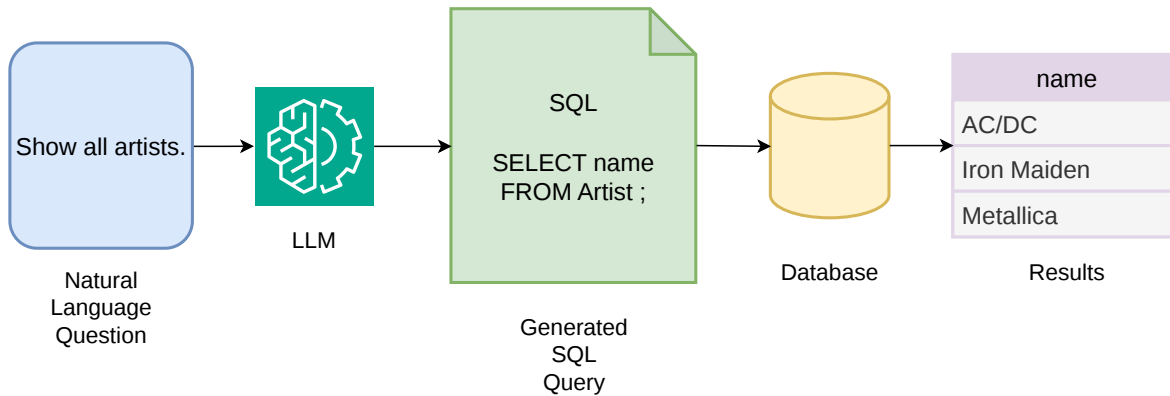
1.2 Μετατροπή φυσικής γλώσσας σε SQL (Natural Language to SQL)

Η αλληλεπίδραση με σχεσιακές βάσεις δεδομένων πραγματοποιείται παραδοσιακά μέσω της γλώσσας SQL. Παρότι η SQL αποτελεί ιδιαίτερα ισχυρό εργαλείο για την ανάκτηση και διαχείριση δεδομένων, η χρήση της απαιτεί τεχνικές γνώσεις σχετικά με τη σύνταξη της γλώσσας και τη δομή της βάσης δεδομένων. Ως αποτέλεσμα, μη εξειδικευμένοι χρήστες συχνά δυσκολεύονται να διατυπώσουν πολύπλοκα ερωτήματα και να αποκτήσουν πρόσβαση στην επιθυμητή πληροφορία χωρίς τη βοήθεια ειδικών ή προγραμματιστών [5, 6].

Η ανάγκη απλοποίησης της αλληλεπίδρασης με τις βάσεις δεδομένων οδήγησε στην ανάπτυξη συστημάτων μετατροπής φυσικής γλώσσας σε SQL (Natural Language to SQL - NL2SQL), γνωστών και ως Text-to-SQL συστημάτων. Στόχος των συστημάτων αυτών είναι η μετατροπή ενός ερωτήματος διατυπωμένου σε φυσική γλώσσα σε ένα ισοδύναμο ερώτημα SQL, το οποίο μπορεί να εκτελεστεί σε μία σχεσιακή βάση δεδομένων ώστε να επιστραφούν τα κατάλληλα αποτελέσματα στον χρήστη. Η διαδικασία αυτή θεωρείται πρόβλημα σημασιολογικής ανάλυσης (semantic parsing), καθώς το σύστημα καλείται να μετατρέψει φυσική γλώσσα σε μία τυπική γλώσσα ερωτημάτων διατηρώντας τη σημασιολογική σημασία του αρχικού ερωτήματος [5, 7]. Η μετατροπή φυσικής γλώσσας σε SQL αποτελεί έναν συνδυασμό τεχνικών επεξεργασίας φυσικής γλώσσας (Natural Language Processing - NLP), τεχνητής νοημοσύνης και συστημάτων βάσεων δεδομένων. Η επεξεργασία φυσικής γλώσσας αποτελεί υποτομέα της τεχνητής νοημοσύνης και επικεντρώνεται στην κατανόηση, ερμηνεία και παραγωγή ανθρώπινης γλώσσας από υπολογιστικά συστήματα. Στο πλαίσιο των συστημάτων NL2SQL πραγματοποιείται συντακτική και σημασιολογική ανάλυση του ερωτήματος του χρήστη, με στόχο την κατανόηση της πρόθεσης και της σημασίας του [6, 7].

Η λειτουργία ενός συστήματος NL2SQL περιλαμβάνει συνήθως πολλαπλά στάδια. Αρχικά, το ερώτημα φυσικής γλώσσας αναλύεται ώστε να εντοπιστούν οι βασικές οντότητες, οι λέξεις-κλειδιά και οι συνθήκες του ερωτήματος. Στη συνέχεια πραγματοποιείται αντιστοίχιση των στοιχείων αυτών με το σχήμα της βάσης δεδομένων, δηλαδή με πίνακες, στήλες και σχέσεις μεταξύ των δεδομένων. Η διαδικασία αυτή είναι γνωστή ως schema linking [8, 9] και αποτελεί κρίσιμο στάδιο για την παραγωγή ορθών SQL ερωτημάτων. Τέλος, παράγεται το αντίστοιχο ερώτημα SQL, το οποίο εκτελείται στη βάση δεδομένων και επιστρέφει τα αποτελέσματα στον χρήστη [7].

Παρά τη σημαντική πρόοδο του πεδίου, η ανάπτυξη συστημάτων NL2SQL παραμένει ιδιαίτερα απαιτητική διαδικασία. Η φυσική γλώσσα χαρακτηρίζεται από ασάφεια και πολλαπλούς τρόπους έκφρασης της ίδιας πρόθεσης, γεγονός που δυσκολεύει την ακριβή κατανόηση των ερωτημάτων. Κοινές λέξεις μπορεί να έχουν διαφορετική σημασία ανάλογα με τα συμφραζόμενα, ενώ σε αρκετές περιπτώσεις τα ερωτήματα των χρηστών δεν περιλαμβάνουν όλες τις απαραίτητες πληροφορίες για τη δημιουργία ενός ορθού SQL ερωτήματος. Επιπλέον, ένα ερώτημα φυσικής γλώσσας μπορεί να αντιστοιχεί σε περισσότερα από ένα πιθανά SQL ερωτήματα, γεγονός που αυξάνει την πολυπλοκότητα της διαδικασίας [5, 7, 8, 10].



Σχήμα 1.1: Βασική διαδικασία μετατροπής ερωτήματος φυσικής γλώσσας σε SQL και ανάκτησης των αποτελεσμάτων.

Ένα επιπλέον πρόβλημα αφορά την ανάγκη κατανόησης της δομής της βάσης δεδομένων. Για την παραγωγή ορθών SQL ερωτημάτων, το σύστημα πρέπει να γνωρίζει τις σχέσεις μεταξύ πινάκων, τα ονόματα των πεδίων και τη σημασιολογία των δεδομένων. Σε πραγματικά περιβάλλοντα, η πολυπλοκότητα αυξάνεται ακόμη περισσότερο, καθώς οι βάσεις δεδομένων μπορεί να διαθέτουν μεγάλα και σύνθετα σχήματα με πολλαπλούς πίνακες και σχέσεις [7, 11, 12]. Οι πρώτες προσεγγίσεις NL2SQL βασίζονταν κυρίως σε κανόνες και χειροκίνητες αντιστοιχίσεις μεταξύ λέξεων φυσικής γλώσσας και στοιχείων SQL. Οι μέθοδοι αυτές παρουσίαζαν περιορισμένη ευελιξία και δυσκολεύονταν να διαχειριστούν σύνθετα ή ασαφή ερωτήματα. Στη συνέχεια αναπτύχθηκαν μέθοδοι βασισμένες σε νευρωνικά δίκτυα και μοντέλα ακολουθίας προς ακολουθία (sequence-to-sequence), τα οποία βελτίωσαν σημαντικά την ικανότητα αυτόματης παραγωγής SQL ερωτημάτων [6, 13]. Παρά τη βελτίωση αυτή, αρκετές φορές τα παραγόμενα ερωτήματα περιείχαν συντακτικά ή σημασιολογικά σφάλματα [8].

Τα τελευταία χρόνια, η εξέλιξη των προεκπαιδευμένων γλωσσικών μοντέλων (Pre-trained Language Models - PLMs) και κυρίως των Μεγάλων Γλωσσικών Μοντέλων (Large Language Models - LLMs) οδήγησε σε σημαντική βελτίωση της απόδοσης των συστημάτων NL2SQL [6, 12]. Τα μοντέλα αυτά μπορούν να αξιοποιούν γνώσεις που έχουν αποκτηθεί από μεγάλα σύνολα δεδομένων και να κατανοούν καλύτερα τόσο τη φυσική γλώσσα όσο και τη δομή της βάσης δεδομένων. Παράλληλα, σύγχρονες προσεγγίσεις αξιοποιούν τεχνικές prompting και in-context learning, παρέχοντας στο μοντέλο παραδείγματα ερωτημάτων και πληροφορίες σχετικά με το σχήμα της βάσης δεδομένων ώστε να βελτιωθεί η ακρίβεια παραγωγής SQL ερωτημάτων [8, 14].

Παράλληλα με την εξέλιξη των μεθόδων, αναπτύχθηκαν και εξειδικευμένα σύνολα δεδομένων για την εκπαίδευση και αξιολόγηση των συστημάτων NL2SQL. Ενδεικτικά παραδείγματα αποτελούν τα WikiSQL, Spider και BIRD, τα οποία χρησιμοποιούνται ευρέως στην ερευνητική κοινότητα για την αξιολόγηση της ικανότητας των μοντέλων να μετατρέπουν ερωτήματα φυσικής γλώσσας σε έγκυρα και σημασιολογικά σωστά SQL ερωτήματα. Τα σύνολα δεδομένων αυτά περιλαμβάνουν ερωτήματα διαφορετικής πολυπλοκότητας και χρησιμοποιούνται ως βασικά benchmarks για τη σύγκριση της απόδοσης διαφορετικών προσεγγίσεων NL2SQL [6, 12]. Η συνεχής πρόοδος στον τομέα του NL2SQL συμβάλλει στη δημιουργία πιο φιλικών και προσβάσιμων συστημάτων διαχείρισης δεδομένων, επιτρέποντας σε χρήστες χωρίς εξειδικευμένες

τεχνικές γνώσεις να αλληλεπιδρούν αποτελεσματικά με σχεσιακές βάσεις δεδομένων. Ιδιαίτερα η αξιοποίηση των Μεγάλων Γλωσσικών Μοντέλων έχει μεταβάλει σημαντικά το πεδίο, επιτρέποντας ακριβέστερη κατανόηση της φυσικής γλώσσας και παραγωγή πιο σύνθετων SQL ερωτημάτων. Οι τεχνολογίες αυτές παρουσιάζονται αναλυτικότερα στην επόμενη ενότητα.

1.3 Μεγάλα Γλωσσικά Μοντέλα για μετατροπή φυσικής γλώσσας σε SQL

Τα τελευταία χρόνια, η ανάπτυξη των Μεγάλων Γλωσσικών Μοντέλων (Large Language Models - LLMs) έχει επηρεάσει σημαντικά τον τομέα της επεξεργασίας φυσικής γλώσσας και της τεχνητής νοημοσύνης. Τα μοντέλα αυτά εκπαιδεύονται σε πολύ μεγάλο όγκο δεδομένων κειμένου και είναι ικανά να κατανοούν, να επεξεργάζονται και να παράγουν φυσική γλώσσα με υψηλό βαθμό ακρίβειας. Η συνεχής αύξηση της κλίμακας και των δυνατοτήτων των LLMs οδήγησε στην αξιοποίησή τους σε ένα μεγάλο εύρος εφαρμογών, όπως η αυτόματη μετάφραση, η παραγωγή κειμένου, τα συστήματα διαλόγου και η μετατροπή φυσικής γλώσσας σε SQL [15].

Η ανάπτυξη των σύγχρονων Μεγάλων Γλωσσικών Μοντέλων βασίζεται σε αρχιτεκτονικές που επιτρέπουν την αποτελεσματική επεξεργασία μεγάλων ακολουθιών κειμένου και τη διαχείριση των συμφραζομένων. Η εξέλιξη αυτή συνέβαλε σημαντικά στη βελτίωση της κατανόησης και παραγωγής φυσικής γλώσσας από υπολογιστικά συστήματα [12, 15]. Ενδεικτικά παραδείγματα Μεγάλων Γλωσσικών Μοντέλων αποτελούν τα GPT, Llama, Gemini και Claude, τα οποία χρησιμοποιούνται σε ένα ευρύ φάσμα εφαρμογών τεχνητής νοημοσύνης.

Η αξιοποίηση των LLMs στον τομέα της μετατροπής φυσικής γλώσσας σε SQL δημιούργησε νέες δυνατότητες στην αλληλεπίδραση χρηστών με σχεσιακές βάσεις δεδομένων. Τα συστήματα που βασίζονται σε LLM μπορούν να κατανοούν καλύτερα τη σημασιολογία και την πρόθεση του χρήστη, επιτρέποντας τη δημιουργία πιο σύνθετων και ακριβών SQL ερωτημάτων χωρίς να απαιτούνται εξειδικευμένες τεχνικές γνώσεις. Παράλληλα, η χρήση τεχνικών prompting επιτρέπει την καθοδήγηση του μοντέλου μέσω παραδειγμάτων και πληροφοριών σχετικά με το σχήμα της βάσης δεδομένων, βελτιώνοντας την ποιότητα των παραγόμενων SQL ερωτημάτων [9, 15].

Η διαδικασία μετατροπής φυσικής γλώσσας σε SQL με χρήση LLM περιλαμβάνει συνήθως τρία βασικά στάδια. Το πρώτο στάδιο αφορά την κατανόηση της ερώτησης του χρήστη σε φυσική γλώσσα. Το σύστημα αναλύει το ερώτημα ώστε να εντοπίσει τις βασικές οντότητες, τις συνθήκες και τη σημασιολογική πρόθεση του χρήστη. Το δεύτερο στάδιο αφορά την κατανόηση του σχήματος της βάσης δεδομένων. Το μοντέλο πρέπει να αναγνωρίσει τους σχετικούς πίνακες, τις στήλες και τις σχέσεις μεταξύ των δεδομένων ώστε να μπορέσει να δημιουργήσει ένα έγκυρο SQL ερώτημα. Στο τρίτο στάδιο πραγματοποιείται η παραγωγή του τελικού SQL ερωτήματος με σωστή σύνταξη και λογική δομή, ώστε να μπορεί να εκτελεστεί επιτυχώς στη βάση δεδομένων [7, 12, 14].

Η εξέλιξη των συστημάτων NL2SQL ακολούθησε μια πορεία από προσεγγίσεις βασισμένες σε κανόνες και προκαθορισμένα πρότυπα προς νευρωνικές μεθόδους και μοντέλα sequence-to-sequence. Η εμφάνιση των προεκπαιδευμένων γλωσσικών μοντέλων και, στη συνέχεια, των Μεγάλων Γλωσσικών Μοντέλων, οδήγησε σε σημαντική βελτίωση της απόδοσης των συστημάτων Text-to-SQL, ιδιαίτερα στην κατανόηση της φυσικής γλώσσας και στη διαχείριση πιο σύνθετων ερωτημάτων [6, 8, 12].

Παρά τις σημαντικές δυνατότητες των LLMs, εξακολουθούν να υπάρχουν αρκετές προκλήσεις. Η φυσική γλώσσα χαρακτηρίζεται από ασάφεια, ενώ διαφορετικοί χρήστες μπορεί να διατυπώνουν την ίδια ερώτηση με εντελώς διαφορετικό τρόπο. Επιπλέον, τα ονόματα πινάκων και στηλών μιας βάσης δεδομένων δεν αντιπροσωπεύουν πάντα με σαφήνεια το πραγματικό περιεχόμενο των δεδομένων, γεγονός που μπορεί να δημιουργήσει σημασιολογικό κενό μεταξύ του ερωτήματος του χρήστη και του σχήματος της βάσης δεδομένων. Σε ορισμένες περιπτώσεις, τα μοντέλα ενδέχεται να παράγουν λανθασμένα ή μη εκτελέσιμα SQL ερωτήματα, ιδιαίτερα όταν το ερώτημα είναι σύνθετο ή όταν το σχήμα της βάσης δεδομένων είναι μεγάλο και πολύπλοκο. Επιπλέον, ένα σημαντικό πρόβλημα αποτελεί το φαινόμενο των hallucinations, κατά το οποίο το μοντέλο μπορεί να παράγει πληροφορίες που δεν αντιστοιχούν στο πραγματικό σχήμα της βάσης δεδομένων, όπως ανύπαρκτα ονόματα πινάκων ή στηλών και λανθασμένες SQL εντολές [5, 7, 8].

Επιπλέον, τα συστήματα που βασίζονται σε LLM συνοδεύονται από αυξημένες απαιτήσεις υπολογιστικών πόρων και κόστος εκτέλεσης, ιδιαίτερα όταν χρησιμοποιούνται μεγάλα μοντέλα σε πραγματικό χρόνο. Παρά τους περιορισμούς αυτούς, τα LLMs θεωρούνται ιδιαίτερα σημαντικά για το μέλλον των ευφύων συστημάτων βάσεων δεδομένων, καθώς προσφέρουν αυξημένη ευελιξία, καλύτερη κατανόηση φυσικής γλώσσας και δυνατότητα αλληλεπίδρασης με μη εξειδικευμένους χρήστες [7, 12].

Η αξιοποίηση των Μεγάλων Γλωσσικών Μοντέλων δημιουργεί νέες δυνατότητες ανάπτυξης ευφύων διεπαφών βάσεων δεδομένων, επιτρέποντας την αποτελεσματική μετατροπή φυσικής γλώσσας σε SQL. Στην παρούσα εργασία, οι τεχνολογίες αυτές αξιοποιούνται για την ανάπτυξη της εφαρμογής AskSQL, με στόχο τη διευκόλυνση της αλληλεπίδρασης χρηστών με σχεσιακές βάσεις δεδομένων μέσω φυσικής γλώσσας και τη βελτίωση της προσβασιμότητας σε λειτουργίες SQL.

1.4 Κίνητρο

Η συνεχής αύξηση του όγκου των δεδομένων και η ευρεία χρήση σχεσιακών βάσεων δεδομένων σε σύγχρονα πληροφοριακά συστήματα δημιουργούν την ανάγκη για αποδοτική και εύκολη πρόσβαση στην πληροφορία. Παρότι η γλώσσα SQL αποτελεί το βασικό μέσο αλληλεπίδρασης με τις σχεσιακές βάσεις δεδομένων, η αποτελεσματική χρήση της απαιτεί τεχνικές γνώσεις σχετικά με τη σύνταξη της γλώσσας και τη δομή της βάσης δεδομένων. Ως αποτέλεσμα, χρήστες χωρίς εμπειρία στις βάσεις δεδομένων ενδέχεται να δυσκολεύονται να διατυπώσουν SQL ερωτήματα και να ανακτήσουν την επιθυμητή πληροφορία.

Τα τελευταία χρόνια, η πρόοδος των Μεγάλων Γλωσσικών Μοντέλων (Large Language Models-LLMs) δημιούργησε νέες δυνατότητες στην κατανόηση φυσικής γλώσσας και στην αυτόματη παραγωγή SQL ερωτημάτων. Η δυνατότητα αλληλεπίδρασης με μία βάση δεδομένων μέσω φυσικής γλώσσας μπορεί να απλοποιήσει σημαντικά τη διαδικασία ανάκτησης πληροφορίας και να καταστήσει τις λειτουργίες των βάσεων δεδομένων περισσότερο προσβάσιμες σε χρήστες με διαφορετικό επίπεδο τεχνικών γνώσεων.

Ωστόσο, η πρακτική αξιοποίηση των δυνατοτήτων των LLMs σε ένα ολοκληρωμένο περιβάλλον βάσεων δεδομένων απαιτεί τη συνδυασμένη υποστήριξη διαφορετικών λειτουργιών. Εκτός από την παραγωγή SQL μέσω φυσικής γλώσσας, ένα τέτοιο σύστημα χρειάζεται να υποστηρίξει τη σύνδεση με βάσεις δεδομένων, την προβολή του σχήματος και των διαθέσιμων πινάκων,

την εκτέλεση ερωτημάτων και την παρουσίαση των αποτελεσμάτων σε ένα ενιαίο και εύχρηστο περιβάλλον.

Η δυνατότητα διατύπωσης ερωτημάτων σε φυσική γλώσσα μπορεί να μειώσει το τεχνικό εμπόδιο χρήσης των βάσεων δεδομένων και να βελτιώσει την εμπειρία αλληλεπίδρασης με τα δεδομένα. Παράλληλα, η διατήρηση της δυνατότητας άμεσης συγγραφής και εκτέλεσης SQL επιτρέπει την υποστήριξη και χρηστών με μεγαλύτερη τεχνική εξοικείωση. Με τον τρόπο αυτό, μία εφαρμογή μπορεί να προσφέρει διαφορετικούς τρόπους αλληλεπίδρασης ανάλογα με τις ανάγκες και το επίπεδο γνώσεων του χρήστη.

Το κίνητρο της παρούσας διπλωματικής εργασίας είναι η σχεδίαση και ανάπτυξη μίας διαδικτυακής εφαρμογής που συνδυάζει λειτουργίες αλληλεπίδρασης με σχεσιακές βάσεις δεδομένων και δυνατότητα μετατροπής ερωτημάτων φυσικής γλώσσας σε SQL με αξιοποίηση υπαρχόντων Μεγάλων Γλωσσικών Μοντέλων. Η εφαρμογή AskSQL στοχεύει στην παροχή ενός ενιαίου και εύχρηστου περιβάλλοντος για τη σύνδεση με βάσεις δεδομένων, την εκτέλεση SQL ερωτημάτων και την αλληλεπίδραση με δεδομένα μέσω φυσικής γλώσσας.

1.5 Συνεισφορά

Η παρούσα διπλωματική εργασία επικεντρώνεται στη σχεδίαση και ανάπτυξη της διαδικτυακής εφαρμογής AskSQL, η οποία παρέχει ένα ενιαίο περιβάλλον αλληλεπίδρασης με σχεσιακές βάσεις δεδομένων και υποστηρίζει τόσο τη συγγραφή ερωτημάτων SQL όσο και τη διατύπωση ερωτημάτων σε φυσική γλώσσα. Η εργασία διερευνά την πρακτική αξιοποίηση των Μεγάλων Γλωσσικών Μοντέλων στο πλαίσιο μιας ολοκληρωμένης διαδικτυακής εφαρμογής βάσεων δεδομένων.

Η βασική συνεισφορά της εργασίας είναι η σχεδίαση και υλοποίηση της εφαρμογής AskSQL, η οποία συνδυάζει λειτουργίες σύνδεσης και αλληλεπίδρασης με σχεσιακές βάσεις δεδομένων με δυνατότητες παραγωγής SQL μέσω φυσικής γλώσσας. Η εφαρμογή υποστηρίζει τη σύνδεση με συστήματα διαχείρισης βάσεων δεδομένων MySQL και PostgreSQL, την προβολή στοιχείων του σχήματος της βάσης δεδομένων, τη συγγραφή και εκτέλεση SQL ερωτημάτων, καθώς και την παρουσίαση των αποτελεσμάτων σε ένα ενιαίο διαδικτυακό περιβάλλον.

Επιπλέον, η εφαρμογή αξιοποιεί τα υπάρχοντα Μεγάλα Γλωσσικά Μοντέλα μέσω των αντίστοιχων APIs για τη μετατροπή ερωτημάτων φυσικής γλώσσας σε SQL. Με τον τρόπο αυτό, παρέχεται ένας εναλλακτικός τρόπος αλληλεπίδρασης με τις βάσεις δεδομένων, ο οποίος μπορεί να διευκολύνει χρήστες που δεν είναι εξοικειωμένοι με τη σύνταξη της SQL, ενώ παράλληλα διατηρείται η δυνατότητα άμεσης συγγραφής και εκτέλεσης SQL για πιο έμπειρους χρήστες. Στο πλαίσιο της ανάπτυξης της εφαρμογής πραγματοποιήθηκαν πειράματα με διαφορετικές προσεγγίσεις prompting και με διαφορετικά Μεγάλα Γλωσσικά Μοντέλα. Σκοπός των πειραμάτων δεν ήταν η γενική αξιολόγηση ή κατάταξη των LLMs ως συστημάτων Text-to-SQL, αλλά η διερεύνηση της συμπεριφοράς των επιλεγμένων προσεγγίσεων και η υποστήριξη τεχνικών αποφάσεων για την επιλογή της καταλληλότερης λύσης για την εφαρμογή AskSQL.

Τέλος, η χρησιμότητα της εφαρμογής αξιολογήθηκε από χρήστες με τη χρήση του ερωτηματολογίου System Usability Scale (SUS). Η αξιολόγηση αυτή παρέχει πληροφορίες σχετικά με την ευχρηστία και τη συνολική εμπειρία χρήσης της εφαρμογής.

1.6 Οργάνωση της Διπλωματικής

Η παρούσα διπλωματική εργασία οργανώνεται σε οκτώ κεφάλαια. Στο πρώτο κεφάλαιο παρουσιάζονται οι βασικές έννοιες και το θεωρητικό υπόβαθρο που σχετίζονται με το αντικείμενο της εργασίας. Αρχικά, γίνεται αναφορά στις σχεσιακές βάσεις δεδομένων και στη γλώσσα SQL, ενώ στη συνέχεια παρουσιάζεται η έννοια του Text-to-SQL, η εξέλιξή της και η συμβολή των Μεγάλων Γλωσσικών Μοντέλων (Large Language Models - LLMs) στην αυτόματη παραγωγή SQL ερωτημάτων από ερωτήματα διατυπωμένα σε φυσική γλώσσα. Στη συνέχεια παρουσιάζονται συνοπτικά τα κεφάλαια που ακολουθούν.

Στο Κεφάλαιο 2 παρουσιάζεται η σχετική βιβλιογραφία και έρευνα στον τομέα του Text-to-SQL. Αρχικά, εξετάζονται οι πρώτες προσεγγίσεις για τη μετατροπή ερωτημάτων φυσικής γλώσσας σε SQL και στη συνέχεια παρουσιάζεται η εξέλιξη του πεδίου μετά την εμφάνιση και την ευρεία αξιοποίηση των Μεγάλων Γλωσσικών Μοντέλων. Παράλληλα, αναλύονται οι βασικότερες μέθοδοι που έχουν προταθεί, καθώς και οι προκλήσεις και οι περιορισμοί που παρουσιάζονται κατά τη διαδικασία παραγωγής SQL μέσω LLMs.

Στο Κεφάλαιο 3 παρουσιάζονται οι τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής. Για την υλοποίηση του frontend χρησιμοποιήθηκαν τεχνολογίες όπως HTML, CSS, Bootstrap και JavaScript, με στόχο τη δημιουργία μιας εύχρηστης και λειτουργικής γραφικής διεπαφής. Για την υλοποίηση του backend χρησιμοποιήθηκε η γλώσσα προγραμματισμού PHP, μέσω της οποίας αναπτύχθηκε το Web API της εφαρμογής. Το API επικοινωνεί τόσο με την εσωτερική βάση δεδομένων MySQL, στην οποία αποθηκεύονται πληροφορίες απαραίτητες για τη λειτουργία της εφαρμογής, όσο και με τις εξωτερικές βάσεις δεδομένων στις οποίες συνδέονται οι χρήστες. Επιπλέον, το backend επικοινωνεί με τα APIs των επιλεγμένων Μεγάλων Γλωσσικών Μοντέλων για την παραγωγή SQL ερωτημάτων. Στο ίδιο κεφάλαιο παρουσιάζονται επίσης οι τεχνολογίες που χρησιμοποιήθηκαν για την αποστολή ηλεκτρονικών μηνυμάτων μέσω του PHPMailer, για τη μετατροπή ομιλίας σε κείμενο (Speech-to-Text), καθώς και οι υπηρεσίες της Google που χρησιμοποιούνται για τη σύνδεση των χρηστών μέσω λογαριασμού.

Στο Κεφάλαιο 4 παρουσιάζεται η διερεύνηση διαφορετικών προσεγγίσεων για την παραγωγή SQL ερωτημάτων μέσω Μεγάλων Γλωσσικών Μοντέλων στο πλαίσιο της ανάπτυξης της εφαρμογής. Πραγματοποιήθηκαν πειράματα με στόχο τη διερεύνηση της καταλληλότερης τεχνικής prompting για την εφαρμογή AskSQL. Αρχικά, συγκρίθηκαν τρεις διαφορετικές μέθοδοι, οι οποίες διαφοροποιούνται ως προς την πληροφορία που παρέχεται στο LLM κατά τη διαδικασία παραγωγής του ερωτήματος. Στη συνέχεια, εξετάστηκε η συμπεριφορά δύο διαφορετικών LLMs με στόχο την υποστήριξη της επιλογής της καταλληλότερης προσέγγισης για την εφαρμογή.

Στο Κεφάλαιο 5 παρουσιάζεται αναλυτικά η σχεδίαση και η υλοποίηση της εφαρμογής AskSQL. Ειδικότερα, αναλύονται οι λειτουργικές απαιτήσεις της εφαρμογής και παρουσιάζεται η αρχιτεκτονική της μέσω κατάλληλων διαγραμμάτων και σχημάτων. Παράλληλα, εξετάζονται οι διαφορετικοί τύποι χρηστών που υποστηρίζονται από την εφαρμογή. Στη συνέχεια, παρουσιάζεται αναλυτικά η υλοποίηση του backend και του frontend. Ιδιαίτερη έμφαση δίνεται στο backend, όπου περιγράφονται η βάση δεδομένων που αναπτύχθηκε για την υποστήριξη της εφαρμογής, καθώς και το Web API και τα βασικά endpoints που υλοποιήθηκαν για την επικοινωνία μεταξύ των επιμέρους συστατικών του συστήματος.

Στο Κεφάλαιο 6 παρουσιάζεται το εγχειρίδιο χρήσης της εφαρμογής και της γραφικής διεπαφής της. Περιγράφονται η διαδικασία εγγραφής (Sign Up) και σύνδεσης (Log In), η ανάκτηση

κωδικού πρόσβασης και η διαχείριση του λογαριασμού του χρήστη. Παρουσιάζεται επίσης η διαδικασία σύνδεσης με τη βάση δεδομένων του χρήστη και η διαχείρισή της. Επιπλέον, παρουσιάζεται η διαδικασία διατύπωσης ερωτημάτων σε φυσική γλώσσα, η αυτόματη παραγωγή των αντίστοιχων SQL ερωτημάτων με τη βοήθεια των LLMs και η εκτέλεσή τους στη βάση δεδομένων. Τέλος, παρουσιάζονται οι πρόσθετες λειτουργίες και δυνατότητες που παρέχονται στους διαχειριστές της εφαρμογής.

Στο Κεφάλαιο 7 παρουσιάζονται τα αποτελέσματα της αξιολόγησης της χρησιμότητας της εφαρμογής μέσω του ερωτηματολογίου System Usability Scale (SUS). Τα αποτελέσματα αναλύονται με στόχο την αποτίμηση της αντιλαμβανόμενης ευχρηστίας και της συνολικής εμπειρίας χρήσης του AskSQL από την πλευρά των συμμετεχόντων.

Τέλος, στο Κεφάλαιο 8 παρουσιάζονται τα συμπεράσματα που προκύπτουν από την παρούσα εργασία. Συνοψίζονται τα βασικά στοιχεία της σχεδίασης και υλοποίησης της εφαρμογής, τα αποτελέσματα της διερεύνησης των τεχνικών προσεγγίσεων και της αξιολόγησης της χρησιμότητας, ενώ παράλληλα παρουσιάζονται οι περιορισμοί της εργασίας και πιθανές κατευθύνσεις για μελλοντική επέκταση του συστήματος.

Κεφάλαιο 2

Ανασκόπηση της βιβλιογραφίας

2.1 Η εξέλιξη του Text-to-SQL

Η μετατροπή ερωτημάτων φυσικής γλώσσας σε ερωτήματα SQL, γνωστή ως Text-to-SQL ή Natural Language to SQL (NL2SQL), αποτελεί ένα σημαντικό ερευνητικό πεδίο στην περιοχή της επεξεργασίας φυσικής γλώσσας και της αλληλεπίδρασης ανθρώπου-μηχανής με βάσεις δεδομένων. Στόχος των συστημάτων Text-to-SQL είναι η αυτόματη μετατροπή ενός ερωτήματος διατυπωμένου σε φυσική γλώσσα σε ένα αντίστοιχο ερώτημα SQL, το οποίο μπορεί να εκτελεστεί σε ένα σύστημα διαχείρισης βάσεων δεδομένων. Με αυτόν τον τρόπο, οι χρήστες μπορούν να ανακτούν πληροφορίες από βάσεις δεδομένων χωρίς να απαιτείται να γνωρίζουν τη σύνταξη και τη σημασιολογία της SQL [5, 6].

Οι πρώτες προσεγγίσεις Text-to-SQL βασίστηκαν κυρίως σε κανόνες, χειροποίητα πρότυπα και προκαθορισμένες αντιστοιχίσεις μεταξύ εκφράσεων φυσικής γλώσσας και SQL. Παρότι οι προσεγγίσεις αυτές μπορούσαν να επιτύχουν ικανοποιητικά αποτελέσματα σε περιορισμένα και καλά καθορισμένα περιβάλλοντα, παρουσίαζαν σημαντικούς περιορισμούς ως προς την προσαρμοστικότητα, την κλιμάκωση και τη δυνατότητα γενίκευσης σε διαφορετικές βάσεις δεδομένων και διαφορετικές διατυπώσεις ερωτημάτων [5, 11].

Η ανάπτυξη των νευρωνικών δικτύων οδήγησε σε σημαντική αλλαγή στην προσέγγιση του προβλήματος. Οι νευρωνικές μέθοδοι επέτρεψαν την εκμάθηση της αντιστοίχισης μεταξύ φυσικής γλώσσας και SQL από δεδομένα, περιορίζοντας την ανάγκη για χειροποίητους κανόνες. Στο πλαίσιο αυτό αναπτύχθηκαν αρχιτεκτονικές βασισμένες σε μοντέλα sequence-to-sequence, τα οποία αντιμετώπιζαν τη δημιουργία του SQL ερωτήματος ως πρόβλημα παραγωγής ακολουθίας [6, 10].

Χαρακτηριστικό παράδειγμα νευρωνικής προσέγγισης αποτελεί το SQLNet, το οποίο προτάθηκε ως εναλλακτική στις κλασικές sequence-to-sequence αρχιτεκτονικές. Αντί να αντιμετωπίζει την παραγωγή του SQL ως απλή σειριακή πρόβλεψη, το SQLNet χρησιμοποιεί μια sketch-based προσέγγιση, όπου η δομή του SQL ερωτήματος περιγράφεται μέσω ενός προκαθορισμένου SQL sketch. Η προσέγγιση αυτή αξιοποιεί επίσης column attention και sequence-to-set prediction, ώστε να λαμβάνεται υπόψη ότι η σειρά με την οποία εμφανίζονται ορισμένα στοιχεία ενός SQL ερωτήματος δεν είναι απαραίτητα σημαντική. Με τον τρόπο αυτό επιτυγχάνεται καλύτερη αντιστοίχιση μεταξύ των στοιχείων της φυσικής γλώσσας και των στηλών της βάσης δεδομένων [16].

Μια επόμενη σημαντική κατεύθυνση ήταν η αξιοποίηση προεκπαιδευμένων μοντέλων γλώσσ-

σας (pre-trained language models), τα οποία μπορούν να αξιοποιούν γλωσσική γνώση που έχει αποκτηθεί από μεγάλης κλίμακας δεδομένα κατά την εκπαίδευσή τους. Η χρήση Transformer-based μοντέλων και τεχνικών αναπαράστασης του σχήματος της βάσης δεδομένων οδήγησε σε βελτιωμένη κατανόηση τόσο της φυσικής γλώσσας όσο και των σχέσεων μεταξύ πινάκων και στηλών [6, 8].

Σε αυτή την κατεύθυνση εντάσσεται και το RAT-SQL, το οποίο αξιοποιεί relation-aware schema encoding για την αναπαράσταση των σχέσεων μεταξύ των στοιχείων του σχήματος της βάσης δεδομένων. Η προσέγγιση αυτή δίνει ιδιαίτερη έμφαση στην κατανόηση των σχέσεων μεταξύ πινάκων, στηλών και του ερωτήματος, γεγονός που είναι ιδιαίτερα σημαντικό για τη δημιουργία σύνθετων SQL ερωτημάτων που περιλαμβάνουν πολλαπλούς πίνακες και joins [17].

Η πιο πρόσφατη εξέλιξη στον χώρο σχετίζεται με τη χρήση Μεγάλων Γλωσσικών Μοντέλων (Large Language Models - LLMs). Μοντέλα όπως τα GPT, Gemini και Claude διαθέτουν σημαντικές δυνατότητες κατανόησης και παραγωγής φυσικής γλώσσας και μπορούν να χρησιμοποιηθούν για Text-to-SQL χωρίς να απαιτείται απαραίτητα η εκπαίδευση ενός εξειδικευμένου μοντέλου για κάθε βάση δεδομένων [12, 14, 15].

Σε αντίθεση με τις προηγούμενες προσεγγίσεις, τα LLMs μπορούν να αξιοποιηθούν μέσω τεχνικών prompting, συμπεριλαμβανομένων των zero-shot και few-shot προσεγγίσεων. Στην περίπτωση του zero-shot prompting, το μοντέλο καλείται να δημιουργήσει το SQL ερώτημα χωρίς να του παρέχονται παραδείγματα αντίστοιχων ερωτημάτων. Αντίθετα, στο few-shot prompting παρέχονται στο μοντέλο ένα ή περισσότερα παραδείγματα ερωτημάτων και των αντίστοιχων SQL εντολών, ώστε να καθοδηγηθεί η διαδικασία παραγωγής [8, 15].

Ιδιαίτερα σημαντική για την απόδοση των LLM-based Text-to-SQL συστημάτων είναι η παροχή πληροφοριών σχετικά με το σχήμα της βάσης δεδομένων. Πληροφορίες όπως τα ονόματα των πινάκων, οι στήλες και οι μεταξύ τους σχέσεις μπορούν να ενσωματωθούν στο prompt και να χρησιμοποιηθούν από το μοντέλο για την παραγωγή ενός SQL ερωτήματος που ανταποκρίνεται στη συγκεκριμένη βάση δεδομένων. Ως αποτέλεσμα, η έρευνα των τελευταίων ετών έχει στραφεί, μεταξύ άλλων, στη βελτιστοποίηση της δομής των prompts και στην αποτελεσματικότερη αναπαράσταση του database schema [8, 12, 15].

2.2 Datasets και Benchmarks

Η ανάπτυξη και η αξιολόγηση συστημάτων Text-to-SQL βασίζεται σε σύνολα δεδομένων (datasets) τα οποία περιλαμβάνουν ερωτήματα διατυπωμένα σε φυσική γλώσσα και τις αντίστοιχες SQL εντολές. Τα datasets αυτά χρησιμοποιούνται για την εκπαίδευση, την επικύρωση και την αξιολόγηση των μοντέλων και επιτρέπουν τη σύγκριση διαφορετικών προσεγγίσεων κάτω από κοινές συνθήκες [5, 6].

Ένα από τα σημαντικά datasets στον χώρο του Text-to-SQL είναι το WikiSQL. Το WikiSQL περιλαμβάνει ζεύγη ερωτημάτων φυσικής γλώσσας και SQL ερωτημάτων τα οποία βασίζονται σε πίνακες που προέρχονται από τη Wikipedia. Το πρόβλημα που ορίζεται στο WikiSQL αφορά κυρίως ερωτήματα πάνω σε έναν πίνακα και, ως εκ τούτου, είναι πιο περιορισμένο σε σχέση με μεταγενέστερα benchmarks που υποστηρίζουν σύνθετες βάσεις δεδομένων [18].

Ένα σημαντικότερο benchmark για την αξιολόγηση της δυνατότητας γενίκευσης των Text-to-SQL συστημάτων είναι το Spider. Το Spider σχεδιάστηκε ως cross-domain benchmark και περιλαμβάνει ερωτήματα που βασίζονται σε πολλαπλές βάσεις δεδομένων διαφορετικών θε-

ματικών περιοχών. Σε αντίθεση με απλούστερα datasets, τα ερωτήματα του Spider μπορούν να περιλαμβάνουν πολλαπλούς πίνακες, joins, aggregations, nested queries και άλλες σύνθετες SQL λειτουργίες. Επιπλέον, οι βάσεις δεδομένων που χρησιμοποιούνται για την αξιολόγηση δεν είναι απαραίτητα ίδιες με αυτές που χρησιμοποιούνται κατά την εκπαίδευση, γεγονός που καθιστά το benchmark ιδιαίτερα χρήσιμο για τη μελέτη της γενίκευσης σε άγνωστα schemas [19].

Τα τελευταία χρόνια έχει αποκτήσει ιδιαίτερη σημασία και το BIRD. Το BIRD σχεδιάστηκε με στόχο να αξιολογήσει τις δυνατότητες των σύγχρονων Text-to-SQL συστημάτων σε περισσότερο ρεαλιστικά περιβάλλοντα. Περιλαμβάνει μεγάλες βάσεις δεδομένων και σύνθετα ερωτήματα, τα οποία απαιτούν όχι μόνο παραγωγή συντακτικά έγκυρου SQL, αλλά και ουσιαστική κατανόηση των δεδομένων και του σχήματος της βάσης. Ως αποτέλεσμα, το BIRD αποτελεί ένα ιδιαίτερα απαιτητικό benchmark για σύγχρονες προσεγγίσεις που βασίζονται σε LLMs [20].

Τα παραπάνω datasets και benchmarks έχουν συμβάλει σημαντικά στην εξέλιξη του πεδίου, καθώς παρέχουν κοινά περιβάλλοντα αξιολόγησης και επιτρέπουν τη σύγκριση διαφορετικών αρχιτεκτονικών και τεχνικών Text-to-SQL. Παράλληλα, η αύξηση της πολυπλοκότητας των benchmarks, από απλούς πίνακες προς σύνθετες και μεγαλύτερες βάσεις δεδομένων, συμβάλει στην προσπάθεια της ερευνητικής κοινότητας να προσεγγίσει πραγματικά περιβάλλοντα χρήσης [8, 12].

2.3 Σύγχρονες προσεγγίσεις και τεχνικές Prompting

Η αξιοποίηση LLMs στο Text-to-SQL έχει οδηγήσει σε διαφορετικές στρατηγικές για την καθοδήγηση των μοντέλων. Η απλούστερη προσέγγιση είναι το zero-shot prompting, όπου το μοντέλο λαμβάνει το ερώτημα του χρήστη και τις απαραίτητες πληροφορίες σχετικά με το database schema χωρίς προηγούμενα παραδείγματα [8, 15].

Μια πιο καθοδηγούμενη προσέγγιση είναι το few-shot prompting, κατά την οποία στο prompt προστίθενται παραδείγματα ερωτημάτων φυσικής γλώσσας μαζί με τα αντίστοιχα SQL ερωτήματα. Τα παραδείγματα μπορούν να βοηθήσουν το μοντέλο να κατανοήσει καλύτερα τη ζητούμενη μορφή και τη λογική των SQL ερωτημάτων [12, 15].

Ιδιαίτερη σημασία έχει επίσης το prompt engineering, δηλαδή ο σχεδιασμός των οδηγιών και των πληροφοριών που παρέχονται στο μοντέλο. Στα Text-to-SQL συστήματα, το prompt μπορεί να περιλαμβάνει πληροφορίες για τους πίνακες, τις στήλες, τους τύπους δεδομένων και τις σχέσεις μεταξύ των πινάκων. Η κατάλληλη επιλογή και οργάνωση αυτών των πληροφοριών μπορεί να επηρεάσει σημαντικά την ποιότητα του παραγόμενου SQL [8, 14].

Η χρήση των LLMs επιτρέπει, επομένως, την ανάπτυξη Text-to-SQL συστημάτων χωρίς την ανάγκη εκπαίδευσης ενός νέου εξειδικευμένου μοντέλου για κάθε διαφορετικό σχήμα της βάσης. Αυτό αποτελεί σημαντικό πλεονέκτημα σε περιβάλλοντα όπου οι βάσεις δεδομένων μεταβάλλονται ή όπου απαιτείται υποστήριξη πολλών διαφορετικών βάσεων [6, 12].

2.4 Προκλήσεις των Text-to-SQL συστημάτων

Παρά τη σημαντική πρόοδο που έχει επιτευχθεί, η μετατροπή φυσικής γλώσσας σε SQL εξακολουθεί να παρουσιάζει αρκετές προκλήσεις. Μία από τις σημαντικότερες αφορά την κατα-

νόηση του σχήματος της βάσης δεδομένων. Το μοντέλο πρέπει να αναγνωρίσει ποιους πίνακες και ποιες στήλες σχετίζονται με το ερώτημα του χρήστη και, όταν απαιτείται, να προσδιορίσει τις κατάλληλες σχέσεις μεταξύ τους [8, 12].

Μια επιπλέον πρόκληση είναι η ασάφεια της φυσικής γλώσσας. Το ίδιο πληροφοριακό αίτημα μπορεί να διατυπωθεί με διαφορετικούς τρόπους, ενώ μία φράση μπορεί να έχει διαφορετικές ερμηνείες ανάλογα με το πλαίσιο. Επομένως, το σύστημα πρέπει να είναι σε θέση να αναγνωρίζει την πρόθεση του χρήστη και να τη μεταφράζει με ακρίβεια στη λογική της SQL [5, 10].

Η παραγωγή σύνθετων SQL ερωτημάτων αποτελεί επίσης σημαντική πρόκληση. Ερωτήματα που περιλαμβάνουν πολλαπλά joins, nested queries, aggregations, grouping και σύνθετες συνθήκες απαιτούν συνδυασμό γλωσσικής κατανόησης και λογικού συλλογισμού. Η δυσκολία αυξάνεται ακόμα περισσότερο όταν η βάση περιλαμβάνει μεγάλο αριθμό πινάκων και στηλών [6, 7, 12].

Ένα ακόμη πρόβλημα των LLM-based προσεγγίσεων είναι η πιθανότητα παραγωγής λανθασμένων ή ανύπαρκτων πληροφοριών, φαινόμενο που είναι γνωστό ως hallucination. Στην περίπτωση του Text-to-SQL, αυτό μπορεί να οδηγήσει στην παραγωγή SQL ερωτημάτων που περιέχουν ανύπαρκτους πίνακες, στήλες ή λανθασμένες σχέσεις μεταξύ τους [8, 15].

Επιπλέον, η χρήση εξωτερικών LLM APIs μπορεί να δημιουργεί ζητήματα που σχετίζονται με το κόστος και τον χρόνο απόκρισης. Στα Text-to-SQL συστήματα, η διαχείριση πληροφοριών σχετικά με το σχήμα της βάσης δεδομένων απαιτεί επίσης ιδιαίτερη προσοχή, ιδιαίτερα όταν χρησιμοποιούνται εξωτερικές υπηρεσίες. [7, 12].

Τέλος, ένα SQL ερώτημα που έχει παραχθεί από ένα LLM δεν πρέπει να θεωρείται αυτομάτως κατάλληλο προς εκτέλεση. Ιδιαίτερα σε εφαρμογές που επιτρέπουν άμεση επικοινωνία με πραγματικές βάσεις δεδομένων, απαιτείται κατάλληλος έλεγχος των παραγόμενων ερωτημάτων και μηχανισμός προστασίας από ανεπιθύμητες ή επικίνδυνες εντολές.

2.5 Υφιστάμενα εργαλεία και συσχέτιση με την παρούσα εργασία

Η ανάπτυξη των Μεγάλων Γλωσσικών Μοντέλων έχει οδηγήσει στη δημιουργία εργαλείων που επιτρέπουν την αλληλεπίδραση με βάσεις δεδομένων μέσω φυσικής γλώσσας. Τα εργαλεία αυτά επιχειρούν να συνδυάσουν τις παραδοσιακές λειτουργίες ενός συστήματος διαχείρισης βάσεων δεδομένων με δυνατότητες παραγωγής SQL μέσω LLMs.

Ένα χαρακτηριστικό παράδειγμα αποτελεί το Chat2DB, ένα database client με δυνατότητες υποβοηθούμενες από τεχνητή νοημοσύνη, το οποίο υποστηρίζει την αλληλεπίδραση με βάσεις δεδομένων μέσω φυσικής γλώσσας και την αυτόματη παραγωγή SQL ερωτημάτων με τη χρήση μιας διαδραστικής agent-based προσέγγισης [21]. Η προσέγγιση αυτή παρουσιάζει ομοιότητες με τη φιλοσοφία του AskSQL, καθώς συνδυάζει παραδοσιακές λειτουργίες διαχείρισης βάσεων δεδομένων με δυνατότητες Text-to-SQL βασισμένες σε Μεγάλα Γλωσσικά Μοντέλα.

Ένα ακόμη σχετικό εργαλείο είναι το SQL Chat, ένα open-source, web-based SQL client που επικεντρώνεται στη συνομιλιακή αλληλεπίδραση με βάσεις δεδομένων μέσω φυσικής γλώσσας [22]. Το SQL Chat επιτρέπει στους χρήστες να διατυπώνουν ερωτήματα σε φυσική γλώσσα και να αξιοποιούν LLMs για την παραγωγή των αντίστοιχων SQL ερωτημάτων.

Η εφαρμογή AskSQL τοποθετείται στην κατεύθυνση αυτών των σύγχρονων εργαλείων, αξιοποιώντας έτοιμα Μεγάλα Γλωσσικά Μοντέλα μέσω των αντίστοιχων APIs για τη μετατροπή

ερωτημάτων φυσικής γλώσσας σε SQL. Σε αντίθεση με προσεγγίσεις που απαιτούν την εκπαίδευση ενός εξειδικευμένου μοντέλου για συγκεκριμένα datasets ή schemas, η εφαρμογή χρησιμοποιεί πληροφορίες σχετικά με το σχήμα της επιλεγμένης βάσης δεδομένων, επιτρέποντας την προσαρμογή της σε διαφορετικές βάσεις και δομές δεδομένων χωρίς την ανάγκη εκ νέου εκπαίδευσης ενός μοντέλου.

Παράλληλα, το AskSQL συνδυάζει τη δυνατότητα αλληλεπίδρασης μέσω φυσικής γλώσσας με λειτουργίες ενός ολοκληρωμένου database client. Συγκεκριμένα, υποστηρίζει σύνδεση με βάσεις δεδομένων MySQL και PostgreSQL, τόσο μέσω άμεσης σύνδεσης όσο και μέσω SSH tunneling, εξερεύνηση του σχήματος και των διαθέσιμων πινάκων, άμεση συγγραφή και εκτέλεση SQL ερωτημάτων, καθώς και αποθήκευση ιστορικού ερωτημάτων. Επιπλέον, υποστηρίζεται έλεγχος πρόσβασης βάσει ρόλων (role-based access control) για τη διαχείριση διαφορετικών τύπων χρηστών.

Σε σύγκριση με εργαλεία όπως το Chat2DB και το SQL Chat, το AskSQL δίνει ιδιαίτερη έμφαση στη συνδυασμένη υποστήριξη διαφορετικών τρόπων αλληλεπίδρασης με τη βάση δεδομένων. Ο χρήστης μπορεί να επιλέξει μεταξύ της άμεσης συγγραφής SQL και της διατύπωσης ερωτημάτων σε φυσική γλώσσα, ανάλογα με τις ανάγκες και το επίπεδο τεχνικής γνώσης του. Επιπλέον, η εφαρμογή υποστηρίζει φωνητική εισαγωγή ερωτημάτων σε ελληνική και αγγλική γλώσσα.

Κεφάλαιο 3

Τεχνολογίες

Στην παρούσα ενότητα παρουσιάζονται οι βασικές τεχνολογίες που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής AskSQL. Συγκεκριμένα, περιγράφονται οι γλώσσες προγραμματισμού, οι βιβλιοθήκες και οι υπηρεσίες που επιλέχθηκαν.

3.1 HTML5

Η HTML (HyperText Markup Language) αποτελεί την κύρια γλώσσα σήμανσης που χρησιμοποιείται για τη δημιουργία ιστοσελίδων και διαδικτυακών εφαρμογών. Χρησιμοποιείται για τον καθορισμό της δομής και του περιεχομένου μιας ιστοσελίδας, επιτρέποντας τον ορισμό στοιχείων όπως επικεφαλίδες, παράγραφοι, σύνδεσμοι, εικόνες, φόρμες και άλλα στοιχεία της διεπαφής χρήστη. Η HTML βασίζεται στη χρήση ετικετών (tags) και στοιχείων (elements), τα οποία καθορίζουν τη δομή και τη σημασιολογία του περιεχομένου. Για παράδειγμα, η ετικέτα `<p>` χρησιμοποιείται για τη δημιουργία μιας παραγράφου [23].

Η HTML5 εισήγαγε σημαντικές βελτιώσεις στη γλώσσα, μεταξύ των οποίων νέα σημασιολογικά στοιχεία για την καλύτερη οργάνωση της δομής μιας ιστοσελίδας, όπως τα `<header>`, `<nav>`, `<main>`, `<section>`, `<article>`, `<aside>` και `<footer>`. Παράλληλα, προστέθηκε βελτιωμένη υποστήριξη για πολυμέσα και σύγχρονες λειτουργίες που απαιτούνται από διαδικτυακές εφαρμογές [23].

Σήμερα, η ανάπτυξη με HTML πραγματοποιείται στο πλαίσιο του HTML Living Standard, το οποίο συνεχώς εξελίσσεται. Η HTML χρησιμοποιείται σε συνδυασμό με την CSS και τη JavaScript, όπου η HTML καθορίζει τη δομή της σελίδας, η CSS την εμφάνιση των στοιχείων, ενώ η JavaScript προσθέτει δυναμική και διαδραστική λειτουργικότητα [23].

Στην εφαρμογή AskSQL, η HTML χρησιμοποιήθηκε για τη δημιουργία της δομής των επιμέρους σελίδων της εφαρμογής, όπως οι σελίδες σύνδεσης, εγγραφής, διαχείρισης συνδέσεων και εκτέλεσης ερωτημάτων.

3.2 CSS3

Η CSS (Cascading Style Sheets) αποτελεί την τεχνολογία που χρησιμοποιείται για τον καθορισμό της εμφάνισης και της μορφοποίησης των στοιχείων ενός εγγράφου HTML. Μέσω της CSS

μπορούν να καθοριστούν χαρακτηριστικά όπως τα χρώματα, οι γραμματοσειρές, τα περιθώρια, οι διαστάσεις, η θέση και η διάταξη των στοιχείων μιας ιστοσελίδας [24].

Οι κανόνες CSS μπορούν να εφαρμοστούν με διαφορετικούς τρόπους, όπως μέσω inline CSS, internal CSS ή external CSS. Η χρήση εξωτερικών αρχείων CSS επιτρέπει τον διαχωρισμό της δομής της ιστοσελίδας από την εμφάνισή της και διευκολύνει τη συντήρηση και την επαναχρησιμοποίηση του κώδικα. Η CSS χρησιμοποιεί selectors, όπως οι κλάσεις (classes), τα αναγνωριστικά (IDs) και οι selectors στοιχείων, για την εφαρμογή συγκεκριμένων κανόνων μορφοποίησης [24].

Στην εφαρμογή AskSQL, η CSS χρησιμοποιήθηκε σε συνδυασμό με την HTML και το Bootstrap για τη διαμόρφωση της γραφικής διεπαφής.

3.3 Bootstrap

Το Bootstrap είναι ένα framework ανάπτυξης διεπαφών ιστού, το οποίο παρέχει έτοιμα στοιχεία και κλάσεις CSS και JavaScript για τη δημιουργία σύγχρονων και responsive διαδικτυακών εφαρμογών. Περιλαμβάνει επαναχρησιμοποιήσιμα στοιχεία διεπαφής, όπως κουμπιά, φόρμες, πίνακες, μενού πλοήγησης, modals και άλλα στοιχεία που χρησιμοποιούνται ευρέως σε διαδικτυακές εφαρμογές [25].

Η χρήση του Bootstrap διευκολύνει τη διαδικασία ανάπτυξης, καθώς παρέχει έτοιμες δομές και μηχανισμούς μορφοποίησης, μειώνοντας την ανάγκη υλοποίησης κοινών στοιχείων από την αρχή. Παράλληλα, παρέχει μηχανισμούς για τη δημιουργία responsive διατάξεων, ώστε η εφαρμογή να προσαρμόζεται σε διαφορετικές διαστάσεις οθόνης [25].

Στην εφαρμογή AskSQL χρησιμοποιήθηκε το Bootstrap 5.3.0 για την υλοποίηση και τη μορφοποίηση της γραφικής διεπαφής. Μέσω των έτοιμων κλάσεων υλοποιήθηκαν στοιχεία όπως κουμπιά, φόρμες, πίνακες, μενού και παράθυρα διαλόγου. Επιπλέον, χρησιμοποιήθηκε η βιβλιοθήκη Bootstrap Icons για την ενσωμάτωση εικονιδίων στη διεπαφή της εφαρμογής.

3.4 JavaScript

Η JavaScript (JS) είναι μια υψηλού επιπέδου, δυναμικά τυποποιημένη γλώσσα προγραμματισμού, η οποία χρησιμοποιείται ευρέως για την ανάπτυξη διαδραστικών διαδικτυακών εφαρμογών. Εκτελείται κυρίως στην πλευρά του πελάτη (client-side), μέσω του προγράμματος περιήγησης, και επιτρέπει τη δυναμική τροποποίηση του περιεχομένου και της συμπεριφοράς μιας ιστοσελίδας χωρίς να απαιτείται η επαναφόρτωση ολόκληρης της σελίδας [26].

Η JavaScript παρέχει τη δυνατότητα πρόσβασης και τροποποίησης των στοιχείων μιας HTML σελίδας μέσω του Document Object Model (DOM), καθώς και τη δυνατότητα ασύγχρονης επικοινωνίας με τον διακομιστή μέσω HTTP αιτημάτων. Για τον σκοπό αυτό μπορούν να χρησιμοποιηθούν σύγχρονα Web APIs, όπως το Fetch API, επιτρέποντας την ανταλλαγή δεδομένων μεταξύ frontend και backend [26].

Στην εφαρμογή AskSQL, η JavaScript χρησιμοποιήθηκε για την υλοποίηση της δυναμικής λειτουργικότητας του frontend και για την επικοινωνία με το backend μέσω HTTP αιτημάτων (χρησιμοποιήθηκε το Fetch API). Συγκεκριμένα, χρησιμοποιείται για τη φόρτωση των διαθέσιμων βάσεων δεδομένων και πινάκων, την ανάκτηση του σχήματος της βάσης δεδομένων, την απο-

στολή ερωτημάτων στο backend, την εμφάνιση των αποτελεσμάτων και τη διαχείριση του ιστορικού ερωτημάτων. Επιπλέον, υλοποιήθηκαν λειτουργίες όπως η δυναμική εναλλαγή μεταξύ SQL και Natural Language Mode, η εμφάνιση του παραγόμενου SQL ερωτήματος, η διαχείριση επικίνδυνων ερωτημάτων και η σελιδοποίηση των αποτελεσμάτων.

3.5 PHP

Η PHP (Hypertext Preprocessor) είναι μια γλώσσα προγραμματισμού ανοικτού κώδικα, γενικού σκοπού, η οποία χρησιμοποιείται ευρέως για την ανάπτυξη διαδικτυακών εφαρμογών. Σε αντίθεση με τη JavaScript που χρησιμοποιείται κυρίως στην πλευρά του πελάτη, η PHP χρησιμοποιείται κυρίως στην πλευρά του διακομιστή (server-side), όπου μπορεί να επεξεργάζεται δεδομένα, να εκτελεί λειτουργίες και να δημιουργεί δυναμικό περιεχόμενο [27].

Η PHP μπορεί να συνεργάζεται με βάσεις δεδομένων και χρησιμοποιείται για την υλοποίηση Web APIs, μέσω των οποίων το frontend μιας εφαρμογής επικοινωνεί με τον server. Η εκτέλεση του κώδικα πραγματοποιείται στον διακομιστή και στον browser επιστρέφονται τα αντίστοιχα δεδομένα ή αποτελέσματα [27].

Στην εφαρμογή AskSQL, η PHP χρησιμοποιήθηκε ως βασική γλώσσα ανάπτυξης του backend και του Web API. Μέσω των PHP scripts πραγματοποιούνται λειτουργίες όπως η αυθεντικοποίηση και η διαχείριση χρηστών, η διαχείριση των συνδέσεων με βάσεις δεδομένων, η ανάκτηση πληροφοριών σχετικά με τις βάσεις, η εκτέλεση SQL ερωτημάτων και η αποθήκευση του ιστορικού ερωτημάτων. Επιπλέον, μέσω του backend πραγματοποιείται η επικοινωνία με τα APIs των Μεγάλων Γλωσσικών Μοντέλων για τη μετατροπή ερωτημάτων φυσικής γλώσσας σε SQL.

3.6 MySQL

Η MySQL είναι ένα σύστημα διαχείρισης σχεσιακών βάσεων δεδομένων (Relational Database Management System - RDBMS), το οποίο χρησιμοποιεί τη γλώσσα SQL για τη δημιουργία, διαχείριση και ανάκτηση δεδομένων. Αποτελεί ένα ευρέως χρησιμοποιούμενο σύστημα διαχείρισης βάσεων δεδομένων και υποστηρίζει βασικές λειτουργίες όπως η δημιουργία και διαχείριση βάσεων, πινάκων και ερωτημάτων SQL [3].

Στην εφαρμογή AskSQL, η MySQL χρησιμοποιείται ως σύστημα αποθήκευσης των δεδομένων που απαιτούνται για τη λειτουργία της ίδιας της εφαρμογής. Σε αυτή αποθηκεύονται, μεταξύ άλλων, οι λογαριασμοί των χρηστών, οι αποθηκευμένες συνδέσεις με εξωτερικές βάσεις δεδομένων και το ιστορικό των ερωτημάτων. Η επικοινωνία της PHP με τη MySQL πραγματοποιείται μέσω των κατάλληλων μηχανισμών σύνδεσης και εκτέλεσης SQL ερωτημάτων.

3.7 PostgreSQL

Η PostgreSQL είναι ένα σύστημα διαχείρισης αντικειμενο-σχεσιακών βάσεων δεδομένων (Object-Relational Database Management System - ORDBMS) ανοικτού κώδικα. Υποστηρίζει τη γλώσσα SQL και παρέχει δυνατότητες για τη διαχείριση σχεσιακών δεδομένων, ξένων κλειδιών, σύνθετων ερωτημάτων και άλλων λειτουργιών που απαιτούνται για την ανάπτυξη εφαρμογών βάσεων δεδομένων [4].

Στο AskSQL, η PostgreSQL μαζί με την MySQL υποστηρίζεται ως ένα από τα συστήματα διαχείρισης βάσεων δεδομένων στα οποία μπορεί να συνδεθεί ο χρήστης. Η εφαρμογή μπορεί να ανακτήσει πληροφορίες σχετικά με τις διαθέσιμες βάσεις και τους πίνακες, να αντλήσει πληροφορίες για το σχήμα της βάσης και να εκτελέσει SQL ερωτήματα. Με αυτόν τον τρόπο, η εφαρμογή υποστηρίζει τόσο βάσεις MySQL όσο και PostgreSQL, παρέχοντας αντίστοιχη λειτουργικότητα ανεξάρτητα από το σύστημα διαχείρισης βάσεων δεδομένων που χρησιμοποιείται.

3.8 Μεγάλα Γλωσσικά Μοντέλα

Τα Μεγάλα Γλωσσικά Μοντέλα (LLMs) αποτελούν μοντέλα τεχνητής νοημοσύνης τα οποία έχουν εκπαιδευτεί σε μεγάλο όγκο δεδομένων και είναι σε θέση να επεξεργάζονται και να παράγουν κείμενο σε φυσική γλώσσα. Τα μοντέλα αυτά μπορούν να χρησιμοποιηθούν για την κατανόηση ερωτημάτων χρηστών και την παραγωγή απαντήσεων σύμφωνα με τις πληροφορίες που τους παρέχονται [28].

Στο πλαίσιο της παρούσας εργασίας, τα LLMs χρησιμοποιούνται για τη μετατροπή ερωτημάτων φυσικής γλώσσας σε SQL ερωτήματα. Συγκεκριμένα, στην εφαρμογή AskSQL ενσωματώθηκαν τα μοντέλα GPT-4o mini της OpenAI και DeepSeek Chat της DeepSeek. Ο χρήστης μπορεί να επιλέξει το επιθυμητό μοντέλο κατά τη χρήση της λειτουργίας Natural Language Mode.

Το GPT-4o mini [28] χρησιμοποιήθηκε μέσω του OpenAI API για την επεξεργασία του ερωτήματος του χρήστη και τη δημιουργία του αντίστοιχου SQL ερωτήματος, λαμβάνοντας υπόψη τις πληροφορίες σχετικά με τη δομή της επιλεγμένης βάσης δεδομένων.

Αντίστοιχα, το DeepSeek Chat [29] χρησιμοποιήθηκε μέσω του αντίστοιχου API για τη μετατροπή ερωτημάτων φυσικής γλώσσας σε SQL. Η ενσωμάτωση περισσότερων του ενός LLMs επιτρέπει την επιλογή διαφορετικών μοντέλων από τον χρήστη και παράλληλα καθιστά δυνατή τη συγκριτική αξιολόγηση της απόδοσής τους στο πλαίσιο της παρούσας εργασίας.

3.9 Web Speech API

Για την υποστήριξη φωνητικών ερωτημάτων στην εφαρμογή AskSQL χρησιμοποιήθηκε το Web Speech API. Πρόκειται για ένα Web API που παρέχει λειτουργίες αναγνώρισης ομιλίας (Speech Recognition) μέσω του προγράμματος περιήγησης. Η λειτουργία αυτή επιτρέπει τη μετατροπή της ομιλίας του χρήστη σε κείμενο, το οποίο μπορεί στη συνέχεια να χρησιμοποιηθεί ως ερώτημα στην εφαρμογή [30].

Στο AskSQL, η αναγνώριση ομιλίας ενσωματώθηκε στη λειτουργία Natural Language Mode. Ο χρήστης μπορεί να επιλέξει τη γλώσσα αναγνώρισης μεταξύ ελληνικών και αγγλικών και να υποβάλει το ερώτημά του μέσω φωνητικής εντολής. Το αναγνωρισμένο κείμενο εισάγεται αυτόματα στο κατάλληλο πεδίο και μπορεί στη συνέχεια να υποβληθεί στο επιλεγμένο LLM για τη δημιουργία του αντίστοιχου SQL ερωτήματος.

3.10 Google Authentication

Για τη διευκόλυνση της διαδικασίας αυθεντικοποίησης των χρηστών, η εφαρμογή AskSQL υποστηρίζει σύνδεση μέσω λογαριασμού Google. Η συγκεκριμένη λειτουργία βασίζεται σε μηχανισμούς OAuth 2.0, οι οποίοι επιτρέπουν σε μία εφαρμογή να πραγματοποιεί αυθεντικοποίηση χρηστών μέσω ενός εξωτερικού παρόχου ταυτότητας, χωρίς να απαιτείται από την εφαρμογή η αποθήκευση ή η διαχείριση του κωδικού πρόσβασης του λογαριασμού Google [31].

Στο AskSQL, ο χρήστης μπορεί να επιλέξει τη σύνδεση μέσω Google και να ολοκληρώσει τη διαδικασία αυθεντικοποίησης στο αντίστοιχο περιβάλλον της Google. Μετά την επιτυχή ολοκλήρωση της διαδικασίας, ο χρήστης επιστρέφει στην εφαρμογή και αποκτά πρόσβαση στον αντίστοιχο λογαριασμό του. Με αυτόν τον τρόπο παρέχεται ένας εναλλακτικός και απλούστερος τρόπος αυθεντικοποίησης σε σχέση με την κλασική σύνδεση μέσω username/password.

3.11 GitHub

Το GitHub είναι μια διαδικτυακή πλατφόρμα φιλοξενίας και διαχείρισης κώδικα, η οποία βασίζεται στο σύστημα ελέγχου εκδόσεων Git. Παρέχει δυνατότητες παρακολούθησης των αλλαγών σε ένα έργο λογισμικού και διαχείρισης διαφορετικών εκδόσεων του πηγαίου κώδικα [32].

Κατά την ανάπτυξη της εφαρμογής AskSQL, το GitHub χρησιμοποιήθηκε για την αποθήκευση και διαχείριση του πηγαίου κώδικα. Η χρήση του Git επέτρεψε την καταγραφή των αλλαγών που πραγματοποιήθηκαν κατά τη διάρκεια της ανάπτυξης μέσω commits και τη διατήρηση του ιστορικού των τροποποιήσεων του έργου.

Ο πηγαίος κώδικας της εφαρμογής AskSQL είναι διαθέσιμος ως έργο ανοικτού κώδικα (open source) στο GitHub, επιτρέποντας την πρόσβαση και την περαιτέρω αξιοποίησή του από ενδιαφερόμενους χρήστες και προγραμματιστές. Το αποθετήριο της εφαρμογής είναι διαθέσιμο στον ακόλουθο σύνδεσμο:

<https://github.com/margaritadeligiannidi/AskSQL>

Κεφάλαιο 4

Πειραματικές μετρήσεις

Το παρόν κεφάλαιο παρουσιάζει τα πειράματα που πραγματοποιήθηκαν στο πλαίσιο της ανάπτυξης της εφαρμογής AskSQL, με σκοπό τη διερεύνηση διαφορετικών τεχνικών και επιλογών που σχετίζονται με την παραγωγή SQL ερωτημάτων από ερωτήματα φυσικής γλώσσας.

Αρχικά εξετάζονται τρεις διαφορετικές τεχνικές prompting, οι οποίες διαφοροποιούνται ως προς την πληροφορία σχετικά με τη βάση δεδομένων που παρέχεται στο γλωσσικό μοντέλο. Στόχος είναι η αξιολόγηση της επίδρασης κάθε τεχνικής και η επιλογή της καταλληλότερης προσέγγισης για την εφαρμογή AskSQL.

Στη συνέχεια, με χρήση της επιλεγμένης τεχνικής prompting, πραγματοποιείται αξιολόγηση δύο γλωσσικών μοντέλων (LLMs), προκειμένου να εξεταστεί η επίδραση της επιλογής του μοντέλου στην παραγωγή SQL ερωτημάτων υπό τις συγκεκριμένες συνθήκες των πειραμάτων. Η αξιολόγηση αυτή δεν αποτελεί γενική σύγκριση ή κατάταξη των γλωσσικών μοντέλων, αλλά επικεντρώνεται στη διερεύνηση των τεχνικών επιλογών που χρησιμοποιήθηκαν για την ανάπτυξη της εφαρμογής.

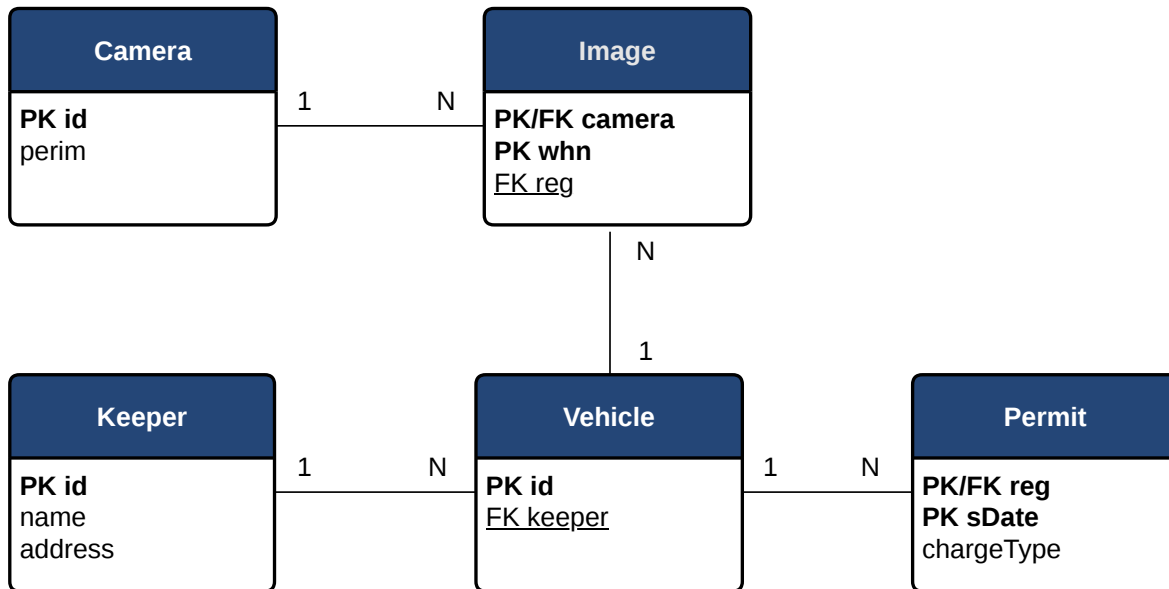
4.1 Βάσεις δεδομένων

Για την πραγματοποίηση των πειραμάτων χρησιμοποιήθηκαν τρεις διαφορετικές βάσεις δεδομένων, οι οποίες επιλέχθηκαν ώστε να εξεταστεί η εφαρμογή AskSQL σε διαφορετικά σχήματα και τύπους ερωτημάτων. Η πρώτη είναι η *Congestion Charging*, η οποία αφορά τη διαχείριση ενός συστήματος χρέωσης οχημάτων που εισέρχονται σε ζώνη κυκλοφοριακής συμφόρησης. Η δεύτερη είναι η *Chinook*, η οποία περιλαμβάνει δεδομένα ενός ηλεκτρονικού καταστήματος πώλησης μουσικής. Τέλος, χρησιμοποιήθηκε η *Hotel Room Reservation*, η οποία αποθηκεύει πληροφορίες σχετικά με κρατήσεις δωματίων ξενοδοχείων. Η δομή των βάσεων δεδομένων παρουσιάζεται αναλυτικά στις επόμενες υποενότητες.

4.1.1 Congestion Charging

Η βάση δεδομένων *Congestion Charging* προσομοιώνει το σύστημα χρέωσης των οχημάτων που εισέρχονται στον εσωτερικό δακτύλιο του Λονδίνου. Στόχος του συστήματος είναι η καταγραφή των διελεύσεων των οχημάτων και ο έλεγχος της ύπαρξης έγκυρης άδειας κυκλοφορίας (permit). Η απλουστευμένη έκδοση της βάσης, η οποία χρησιμοποιήθηκε στα πειράματα, απο-

τελείται από πέντε πίνακες που περιγράφουν τις κάμερες του συστήματος, τους ιδιοκτήτες των οχημάτων, τα οχήματα, τις καταγραφές που πραγματοποιούνται από τις κάμερες και τις άδειες εισόδου στον δακτύλιο. Η δομή της βάσης παρουσιάζεται στο Σχήμα 4.1.



Σχήμα 4.1: Διάγραμμα ER της βάσης Congestion Charging

Οι πίνακες της βάσης συνοψίζονται ως εξής:

- *camera(id, perim)*: Αποθηκεύει τις κάμερες του συστήματος και τον τύπο τους (είσοδος ή έξοδος από τη ζώνη χρέωσης).
- *keeper(id, name, address)*: Περιέχει τα στοιχεία των ιδιοκτητών των οχημάτων.
- *vehicle(id, keeper)*: Συνδέει κάθε όχημα με τον αντίστοιχο ιδιοκτήτη του.
- *image(camera, whn, reg)*: Καταγράφει κάθε φωτογραφία που λαμβάνεται από τις κάμερες, μαζί με την ώρα λήψης και το όχημα που αναγνωρίστηκε.
- *permit(reg, sDate, chargeType)*: Περιέχει τις άδειες κυκλοφορίας των οχημάτων στη ζώνη χρέωσης, καθώς και τον τύπο και την ημερομηνία έκδοσής τους.

4.1.2 Chinook

Η βάση δεδομένων *Chinook* μοντελοποιεί ένα ηλεκτρονικό κατάστημα πώλησης μουσικής. Περιλαμβάνει πληροφορίες για καλλιτέχνες, άλμπουμ, μουσικά κομμάτια, πελάτες, εργαζομένους, λίστες αναπαραγωγής και συναλλαγές αγοράς. Το ER διάγραμμα της βάσης παρουσιάζεται στο Σχήμα 4.2.

Οι πίνακες της βάσης συνοψίζονται ως εξής:

- *Track(TrackId, Name, AlbumId, MediaTypeId, GenreId, Composer, Milliseconds, Bytes, UnitPrice)*: Αποθηκεύει πληροφορίες για τα μουσικά κομμάτια, όπως το όνομα, τον συνθέτη, τη διάρκεια, το μέγεθος του αρχείου και την τιμή τους. Επιπλέον, συνδέει κάθε κομμάτι με το αντίστοιχο άλμπουμ, είδος μουσικής και τύπο μέσου.
- *Album(AlbumId, Title, ArtistId)*: Περιέχει τα άλμπουμ μουσικής και τα συσχετίζει με τον αντίστοιχο καλλιτέχνη.
- *Artist(ArtistId, Name)*: Αποθηκεύει τα στοιχεία των καλλιτεχνών.
- *Genre(GenreId, Name)*: Περιλαμβάνει τις κατηγορίες (είδη) μουσικής στις οποίες ανήκουν τα κομμάτια.
- *MediaType(MediaTypeId, Name)*: Αποθηκεύει τους διαθέσιμους τύπους μέσων αποθήκευσης ή διανομής των μουσικών αρχείων.
- *Playlist(PlaylistId, Name)*: Περιέχει τις λίστες αναπαραγωγής μουσικών κομματιών.
- *PlaylistTrack(PlaylistId, TrackId)*: Αποτελεί πίνακα συσχέτισης μεταξύ λιστών αναπαραγωγής και μουσικών κομματιών, επιτρέποντας σε κάθε λίστα να περιλαμβάνει πολλά κομμάτια.
- *Customer(CustomerId, FirstName, LastName, Company, Address, City, State, Country, PostalCode, Fax, Email, SupportRepId)*: Αποθηκεύει τα στοιχεία των πελατών του ηλεκτρονικού καταστήματος, καθώς και τον υπεύθυνο εργαζόμενο που τους εξυπηρετεί.
- *Employee*: Περιέχει πληροφορίες για τους εργαζομένους του καταστήματος, όπως τα προσωπικά στοιχεία τους, τη θέση εργασίας και την ιεραρχική τους σχέση με άλλους εργαζομένους.
- *Invoice(InvoiceId, CustomerId, InvoiceDate, BillingAddress, BillingCity, BillingState, BillingCountry, BillingPostalCode, Total)*: Αποθηκεύει τα τιμολόγια των αγορών που πραγματοποιούν οι πελάτες, καθώς και στοιχεία χρέωσης και το συνολικό ποσό κάθε συναλλαγής.
- *InvoiceLine(InvoiceLineId, InvoiceId, TrackId, UnitPrice, Quantity)*: Περιέχει τις επιμέρους γραμμές κάθε τιμολογίου, δηλαδή τα μουσικά κομμάτια που αγοράστηκαν, την ποσότητα και την τιμή τους.

4.1.3 Hotel Room Reservation

Η βάση δεδομένων *Hotel Room Reservation* μοντελοποιεί ένα σύστημα διαχείρισης κρατήσεων ξενοδοχείων. Περιλαμβάνει πληροφορίες για τα ξενοδοχεία, τους διαθέσιμους τύπους δωματίων, τους πελάτες, τη διαθεσιμότητα των δωματίων και τις κρατήσεις που πραγματοποιούνται. Το ER διάγραμμα της βάσης παρουσιάζεται στο Σχήμα 4.3.

Οι πίνακες της βάσης συνοψίζονται ως εξής:

- *Hotel(hid, Name, Address, City)*: Αποθηκεύει τα στοιχεία των ξενοδοχείων, όπως το όνομα, τη διεύθυνση και την πόλη στην οποία βρίσκονται.

- *ROOMTYPE(tid, Description)*: Περιλαμβάνει τους διαθέσιμους τύπους δωματίων και την περιγραφή τους.
- *HotelCustomer(cid, Name)*: Αποθηκεύει τα στοιχεία των πελατών που πραγματοποιούν κρατήσεις στα ξενοδοχεία.
- *HRooms(hid, tid, NofRooms, Price)*: Περιγράφει τη διαθεσιμότητα των δωματίων κάθε ξενοδοχείου, αποθηκεύοντας για κάθε τύπο δωματίου το πλήθος των διαθέσιμων δωματίων και την αντίστοιχη τιμή διαμονής.
- *RoomReservation(rid, hid, tid, cid, ArrivalDate, DepartDate, ReservDate)*: Αποθηκεύει τις κρατήσεις των πελατών, περιλαμβάνοντας το ξενοδοχείο, τον τύπο δωματίου, τον πελάτη, καθώς και τις ημερομηνίες κράτησης, άφιξης και αναχώρησης.

4.2 Σύνολο ερωτημάτων αξιολόγησης

Για τις ανάγκες των πειραμάτων της παρούσας εργασίας δημιουργήθηκε ένα σύνολο ερωτημάτων αξιολόγησης (*test dataset*), το οποίο αποτελείται από 60 ερωτήματα διατυπωμένα σε φυσική γλώσσα και αφορούν τις τρεις βάσεις δεδομένων που παρουσιάστηκαν στην προηγούμενη ενότητα. Το σύνολο αυτό σχεδιάστηκε με στόχο τη σύγκριση των διαφορετικών τεχνικών prompting και των γλωσσικών μοντέλων στο πλαίσιο της αξιολόγησης της εφαρμογής AskSQL. Τα ερωτήματα έχουν διαφορετικό επίπεδο δυσκολίας και καλύπτουν ένα ευρύ φάσμα λειτουργιών της SQL, από απλές αναζητήσεις έως σύνθετα ερωτήματα που περιλαμβάνουν πολλαπλές συνενώσεις πινάκων (*JOIN*), συνθήκες φιλτραρίσματος (*WHERE*), ομαδοποίηση αποτελεσμάτων (*GROUP BY*), συναρτήσεις συγκέντρωσης (*COUNT, SUM, AVG*), καθώς και πιο σύνθετες λειτουργίες, όπως ο τελεστής διαίρεσης (*division*).

Για την εξέταση της συμπεριφοράς των εξεταζόμενων προσεγγίσεων σε διαφορετικά επίπεδα πολυπλοκότητας, τα ερωτήματα κατηγοριοποιήθηκαν σε τρεις κατηγορίες δυσκολίας: εύκολα, μεσαίας δυσκολίας και δύσκολα, με 20 ερωτήματα σε κάθε κατηγορία.

Για λόγους οργάνωσης και αναγνωσιμότητας, το σύνολο των ερωτημάτων παρουσιάζεται σε τρεις ξεχωριστούς πίνακες, έναν για κάθε βάση δεδομένων. Συγκεκριμένα, τα ερωτήματα που αφορούν τη βάση δεδομένων *Chinook* παρουσιάζονται στον Πίνακα 4.1, τα ερωτήματα της βάσης δεδομένων *Congestion Charging* στον Πίνακα 4.2, ενώ τα ερωτήματα της βάσης δεδομένων *Hotel Room Reservation* παρουσιάζονται στον Πίνακα 4.3.

Πίνακας 4.1: Ερωτήματα αξιολόγησης για τη βάση δεδομένων *Chinook*.

A/A	Δυσκολία	Ερώτημα
1	Εύκολο	Να εμφανιστούν οι τίτλοι των άλμπουμ του συγκροτήματος 'Queen', στους οποίους περιλαμβάνεται η λέξη 'Hits'.
2	Εύκολο	Να βρεθεί το πλήθος των κομματιών τα οποία είναι είδους 'Rock' ή 'Blues'.

A/A	Δυσκολία	Ερώτημα
3	Εύκολο	Να υπολογιστεί το συνολικό ποσό χρέωσης (total) που αντιστοιχεί σε πελάτες από τις Η.Π.Α. (USA) ή από τον Καναδά (Canada).
4	Εύκολο	Να εμφανιστεί το πλήθος των κομματιών που έχει συνθέσει ο Dickinson (μόνος του ή σε συνεργασία με άλλους μουσικούς - LIKE %Dickinson%) και βρίσκονται στην λίστα με κωδικό 5 (PlaylistID)
5	Εύκολο	Να εμφανιστούν τα ονόματα των κομματιών είδους 'Rock' και ο συνθέτης τους (composer), τα οποία περιλαμβάνουν τη λέξη 'Baby' στον τίτλο τους.
6	Εύκολο	Να εμφανιστούν τα άλμπουμ του συγκροτήματος Led Zeppelin που περιέχουν το όνομα του συγκροτήματος στον τίτλο τους. Στο αποτέλεσμα: [κωδικός άλμπουμ/albumid, τίτλος άλμπουμ/title]
7	Εύκολο	Να εμφανιστούν τα είδη μουσικής (όνομα) με περισσότερα από 50 τραγούδια στη βάση.
8	Εύκολο	Να εμφανιστούν όλα τα ονόματα καλλιτεχνών ταξινομημένα αλφαβητικά.
9	Εύκολο	Να εμφανιστούν οι πελάτες που έχουν πραγματοποιήσει περισσότερες από 10 αγορές.
10	Μεσαίο	Το πλήθος των κομματιών (track) για κάθε είδος μουσικής (genre) σε φθίνουσα κατάταξη ως προς το πλήθος. Στο αποτέλεσμα: [όνομα_είδους/Genre.name, πλήθος]
11	Μεσαίο	Για κάθε υπάλληλο, το πλήθος των πελατών που αυτός εξυπηρετεί. Να εμφανίζονται και οι υπάλληλοι που δεν εξυπηρετούν κανέναν πελάτη. Στο αποτέλεσμα: [όνομα_υπαλλήλου, επώνυμο_υπαλλήλου, πλήθος]
12	Μεσαίο	Να βρεθεί το πλήθος των κομματιών και το πλήθος των αντίστοιχων διαφορετικών συνθετών (composer), τα οποία είναι είδους 'Rock' ή 'Blues'.
13	Μεσαίο	Υπάρχουν τραγούδια που έχει συνθέσει ο 'Eddie Vedder' τα οποία είναι ίδιου είδους μουσικής (genre) με τραγούδια που έχει συνθέσει ο 'Dave Grohl'; Να εμφανιστούν τα ονόματα αυτών των ειδών μουσικής.
14	Μεσαίο	Να εμφανιστούν οι κωδικοί των λιστών (PlaylistId) εκείνων που περιλαμβάνουν τραγούδια είδους 'Metal' ή οι κωδικοί των λιστών (PlaylistId) εκείνων που περιλαμβάνουν τραγούδια αποθηκευμένα σε μορφή 'MPEG audio file' και είδους 'Pop'.
15	Μεσαίο	Να βρεθούν τα ονόματα των μουσικών λιστών (playlist name), στις οποίες υπάρχουν τραγούδια που έχει συνθέσει ο 'Eddie Vedder' και τραγούδια που έχει συνθέσει ο 'Chris Cornell'

A/A	Δυσκολία	Ερώτημα
16	Μεσαίο	Να βρεθούν τα ονόματα των μουσικών λιστών (playlist name), στις οποίες υπάρχουν τα τραγούδια με τίτλους 'The Unforgiven' και 'Crazy'.
17	Μεσαίο	Να βρεθούν τα ονόματα των μουσικών λιστών (playlist name), στις οποίες δεν υπάρχει το τραγούδι με τίτλο 'Angel'.
18	Μεσαίο	Να βρεθούν τα ονόματα των συνθετών, οι οποίοι δεν έχουν συνθέσει τραγούδια είδους 'Rock' ή 'Pop'.
19	Μεσαίο	Να εμφανιστούν οι καλύτεροι πελάτες (κωδικός, όνομα, επώνυμο) και το συνολικό ποσό που έχουν δαπανήσει σε αγορές. Οι καλύτεροι πελάτες είναι εκείνοι που έχουν δαπανήσει συνολικά μεγαλύτερο ποσό των 40 μονάδων.
20	Μεσαίο	Να εμφανιστούν οι καλλιτέχνες (κωδικός, όνομα) που διαθέτουν περισσότερα από 100 τραγούδια στη βάση δεδομένων, μαζί με το πλήθος των τραγουδιών τους.
21	Μεσαίο	Να εμφανιστούν τα άλμπουμ (κωδικός, τίτλος) και το πλήθος των τραγουδιών που περιέχουν. Να εμφανιστούν μόνο τα άλμπουμ που έχουν περισσότερα από 20 τραγούδια.
22	Δύσκολο	Οι πελάτες που έχουν αγοράσει κομμάτι (track) από όλα ανεξαιρέτως τα είδη μουσικής (genre) που το όνομά τους αρχίζει με "S". Στο αποτέλεσμα: [όλα τα στοιχεία των πελατών]
23	Δύσκολο	Να βρεθούν τα ονόματα των μουσικών λιστών (playlist name), στις οποίες υπάρχουν ΚΑΙ τραγούδια που έχει συνθέσει ο 'Eddie Vedder' ΚΑΙ τραγούδια που έχει συνθέσει ο 'Chris Cornell'.
24	Δύσκολο	Να βρεθούν τα ονόματα των μουσικών λιστών (playlist name), στις οποίες υπάρχουν τα τραγούδια με τίτλους 'The Unforgiven' ΚΑΙ 'Crazy'.
25	Δύσκολο	Να εμφανιστούν οι υπάλληλοι (κωδικός, όνομα, επώνυμο) που εξυπηρέτησαν όλους τους πελάτες από από τις ΗΠΑ (USA).
26	Δύσκολο	Να εμφανιστούν οι πελάτες (κωδικός, όνομα, επώνυμο) που έχουν αγοράσει από όλα τα Media Types.
27	Δύσκολο	Να εμφανιστούν οι Playlists (κωδικός, όνομα) που περιέχουν όλα τα τραγούδια του καλλιτέχνη με όνομα Nirvana.
28	Δύσκολο	Να εμφανιστούν οι καλλιτέχνες (κωδικός, όνομα) που διαθέτουν περισσότερα από 100 τραγούδια στη βάση δεδομένων, μαζί με το πλήθος των τραγουδιών τους.
29	Δύσκολο	Να εμφανιστούν οι πελάτες (κωδικός, όνομα, επώνυμο) που έχουν αγοράσει τραγούδια από τουλάχιστον 7 διαφορετικά είδη μουσικής.

A/A	Δυσκολία	Ερώτημα
30	Δύσκολο	Να εμφανιστούν οι Playlists (κωδικός, όνομα) που περιέχουν όλα τα τραγούδια του άλμπουμ με τίτλο 'Nevermind'.

Πίνακας 4.2: Ερωτήματα αξιολόγησης για τη βάση δεδομένων Congestion Charging.

A/A	Δυσκολία	Ερώτημα
1	Εύκολο	Να βρεθεί το πλήθος αδειών που έχουν εκδοθεί για κάθε ιδιοκτήτη. Να εμφανίσετε τον κωδικό, το όνομα και το αντίστοιχο πλήθος για τους ιδιοκτήτες που έχουν βγάλει τουλάχιστον 2 άδειες.
2	Εύκολο	Να εμφανίσετε το πλήθος αδειών που εκδόθηκαν από το 2007 και μετά ανά τύπο αδείας
3	Εύκολο	Να εμφανιστούν ο αριθμός κυκλοφορίας κάθε οχήματος και το πλήθος των φωτογραφιών που έχουν ληφθεί για αυτό από τις κάμερες του συστήματος. Να εμφανιστούν μόνο τα οχήματα που έχουν καταγραφεί περισσότερες από τρεις φορές, με ταξινόμηση κατά φθίνουσα σειρά πλήθους καταγραφών.
4	Μεσαίο	Να βρεθεί το πλήθος των οχημάτων που κατέχει ο κάθε ένας ιδιοκτήτης. Να εμφανίσετε τον κωδικό, το όνομα και το αντίστοιχο πλήθος για τους ιδιοκτήτες που κατέχουν πάνω από 4 οχήματα.
5	Μεσαίο	Να εμφανίσετε τον κωδικό οχήματος και την πιο πρόσφατη άδεια του (ημερομηνία) για τα οχήματα εκείνα για τα οποία έχουν εκδοθεί τουλάχιστον δύο άδειες τύπου Daily
6	Μεσαίο	Να εμφανίσετε τον αριθμό πινακίδας αυτοκινήτων για τα οποία δεν έχει εκδοθεί ποτέ άδεια
7	Δύσκολο	Ποιοί είναι οι αριθμοί κυκλοφορίας των οχημάτων για τα οποία έχει εκδοθεί και εβδομαδιαία (Weekly) και μηνιαία (Monthly) άδεια εισόδου στο δακτύλιο
8	Δύσκολο	Να εμφανιστούν τα στοιχεία των πολιτών οχήματα των οποίων έχουν καταγραφεί και από την κάμερα με κωδικό 2, και από την κάμερα με κωδικό 5.
9	Δύσκολο	Αριθμός κυκλοφορίας αυτοκινήτων για τα οποία έχουν εκδοθεί ΟΛΟΙ οι τύποι της άδειας οι οποίοι έχουν εκδοθεί για το αυτοκίνητο με αριθμό κυκλοφορίας 'SO 02 GTP'
10	Δύσκολο	Τα πλήρη στοιχεία των πολιτών οι οποίοι έχουν εκδώσει ΟΛΟΥΣ τους τύπους αδειών που καταχωρούνται στη βάση, χρησιμοποιώντας το ίδιο ή διαφορετικά αυτοκίνητα.

Πίνακας 4.3: Ερωτήματα αξιολόγησης για τη βάση δεδομένων Hotel Room Reservation.

A/A	Δυσκολία	Ερώτημα
1	Εύκολο	Ονόματα ξενοδοχείων και τύποι δωματίων που διατίθενται σε πιο χαμηλή τιμή από την τιμή που χρεώνει το ξενοδοχείο 'Astir' το δωμάτιο τύπου 'Monoklino'. Τα αποτελέσματα να εμφανιστούν ταξινομημένα με βάση το όνομα του ξενοδοχείου
2	Εύκολο	Εμφανίστε για κάθε ξενοδοχείο (κωδικό, όνομα) το συνολικό πλήθος δωματίων του (δώστε το όνομα Rooms στο πεδίο), με αύξουσα ταξινόμηση κατά το όνομα ξενοδοχείου.
3	Εύκολο	Ονόματα ξενοδοχείων στα οποία έχουν γίνει οι μεγαλύτερες (σε χρονική διάρκεια) κρατήσεις δωματίων. Βοήθεια: χρονική διάρκεια κράτησης δωματίου σε ημέρες = departdate-arrivaldate. Να εμφανιστούν μια φορά το κάθε ένα.
4	Εύκολο	Ονόματα ξενοδοχείων που διαθέτουν το ακριβότερο σε κόστος ημερήσιας διαμονής δωμάτιο
5	Εύκολο	Για κάθε ξενοδοχείο, να βρεθεί ο μέγιστος αριθμός διανυκτερεύσεων (δώστε όνομα nights) ανά τύπο δωματίου. Να εμφανιστούν το όνομα ξενοδοχείου, ο τύπος δωματίου και ο μέγιστος αριθμός διανυκτερεύσεων με φθίνουσα ταξινόμηση κατά αριθμό διανυκτερεύσεων και μετά σε αύξουσα κατά όνομα ξενοδοχείου. (Αριθμός διανυκτερεύσεων = departdate-arrivaldate)
6	Εύκολο	Να εμφανιστούν το όνομα κάθε ξενοδοχείου και η πόλη στην οποία βρίσκεται, για τα ξενοδοχεία που διαθέτουν τουλάχιστον τρεις διαφορετικούς τύπους δωματίων. Τα αποτελέσματα να ταξινομηθούν αλφαβητικά ως προς το όνομα του ξενοδοχείου.
7	Εύκολο	Να εμφανιστούν ο κωδικός και το όνομα κάθε πελάτη, καθώς και το συνολικό πλήθος των κρατήσεων που έχει πραγματοποιήσει. Να εμφανιστούν μόνο οι πελάτες που έχουν πραγματοποιήσει περισσότερες από μία κράτηση, με ταξινόμηση κατά φθίνουσα σειρά πλήθους κρατήσεων.
8	Εύκολο	Να εμφανιστούν το όνομα κάθε ξενοδοχείου, η περιγραφή του τύπου δωματίου και η ημερήσια τιμή του, για όλα τα δωμάτια που έχουν τιμή μεγαλύτερη από 150. Τα αποτελέσματα να ταξινομηθούν κατά αύξουσα σειρά τιμής.
9	Μεσαίο	Τα στοιχεία των ξενοδοχείων που έχουν τον μεγαλύτερο αριθμό δωματίων τύπου 'Souita super lux'

A/A	Δυσκολία	Ερώτημα
10	Μεσαίο	Για κάθε ξενοδοχείο (κωδικό και όνομα), βρείτε το πλήθος κρατήσεων μιας ημέρας (ονομάστε το πεδίο OneDayReservation). Εμφανίστε μόνο τα ξενοδοχεία τα οποία έχουν συνολικό πλήθος κρατήσεων μιας ημέρας μεγαλύτερο από 5. Ταξινομήστε το αποτέλεσμα ανά πλήθος κρατήσεων σε αύξουσα σειρά. Σημείωση: αν αφαιρέσετε από την ημέρα αναχώρησης 'departdate' την ημέρα άφιξης 'arrivaldate' θα πάρετε το πλήθος των ημερών.
11	Μεσαίο	Για τον κάθε ένα πελάτη (κωδικός και όνομα) εμφανίστε το πλήθος των κρατήσεων δωματίων 'Souita lux' (δώστε όνομα στο πεδίο αυτό LuxReservations). Το αποτέλεσμα να ταξινομηθεί κατά όνομα πελάτη σε φθίνουσα ταξινόμηση και μετά κατά κωδικό πελάτη σε αύξουσα ταξινόμηση δεδομένου ότι υπάρχουν συνωνυμίες.
12	Μεσαίο	Δεδομένου ότι οι πελάτες πληρώνουν κατά την ημέρα αναχώρησης, εμφανίστε για κάθε ξενοδοχείο (κωδικός) τα συνολικά έσοδα του για το 2015 με το όνομα (tziros), ταξινομημένα σε φθίνουσα σειρά ανά τζίρο. Σημείωση: Ο τζίρος υπολογίζεται από το αριθμό των ημερών παραμονής (Ημ. Αναχώρησης μείον Ημ. άφιξης) επί την τιμή (price).
13	Μεσαίο	Εάν όλα τα δωμάτια από όλα τα ξενοδοχεία είναι νοικιασμένα, ποιός θα είναι ο συνολικός ημερήσιος τζίρος (nofrooms επί την τιμή price) του κάθε ξενοδοχείου; Εμφανίστε Όνομα Ξενοδοχείου, κωδικό και τζίρο (δώστε σαν όνομα στο πεδίο TzirosPerDay) σε φθίνουσα ταξινόμηση κατά ημερήσιο τζίρο, που έχουν ημερήσιο τζίρο μεγαλύτερο ή ίσο με 5000
14	Δύσκολο	Ξενοδοχεία με έδρα την Αθήνα (Athens) που διαθέτουν ΟΛΟΥΣ τους τύπους δωματίου. Στο αποτέλεσμα: Κωδικός_Ξενοδοχείου, Όνομα_Ξενοδοχείου.
15	Δύσκολο	Τύποι δωματίων που υπάρχουν σε ΟΛΑ τα ξενοδοχεία. Στο αποτέλεσμα: Κωδικός_Τύπου, Περιγραφή_Τύπου
16	Δύσκολο	Ονόματα πελατών (ένας πελάτης ανά γραμμή) που έχουν κάνει κράτηση στο πιο ακριβό δωμάτιο ξενοδοχείου της Θεσσαλονίκης (Thessaloniki)
17	Δύσκολο	Πελάτες (κωδικός και όνομα, μια εγγραφή ανά πελάτη) που επιλέγουν πάντα δωμάτια τύπου 'Monoklino me thea', με ταξινόμηση ανά όνομα πελάτη και μετά ανά κωδικό πελάτη. Σημείωση: Δεν υπάρχουν κρατήσεις σε άλλους τύπους δωματίων. Βοήθεια: Μπορεί να λυθεί με EXCEPT ή NOT EXISTS.
18	Δύσκολο	Ξενοδοχεία που διαθέτουν ΟΛΟΥΣ τους τύπους δωματίου. Στο αποτέλεσμα: Κωδικός_Ξενοδοχείου, Όνομα_Ξενοδοχείου.

A/A	Δυσκολία	Ερώτημα
19	Δύσκολο	Πελάτες (όνομα και κωδικός) ξενοδοχείων των Αθηνών (Athens) που δεν έχουν κάνει καμία διαμονή σε ξενοδοχείο της Θεσσαλονίκης
20	Δύσκολο	Να εμφανιστούν τα στοιχεία των ξενοδοχείων με έδρα την Θεσσαλονίκη (Thessaloniki) που διαθέτουν ΟΛΟΥΣ τους τύπους δωματίου. Στο αποτέλεσμα: Κωδικός_Ξενοδοχείου, Όνομα_Ξενοδοχείου.

4.3 Τεχνικές prompting

Στην πρώτη φάση των πειραμάτων συγκρίνονται τρεις διαφορετικές τεχνικές prompting, οι οποίες διαφοροποιούνται ως προς την πληροφορία σχετικά με τη βάση δεδομένων που παρέχεται στο LLM. Στόχος της σύγκρισης είναι να διερευνηθεί η επίδραση της ποσότητας της παρεχόμενης πληροφορίας στην ορθότητα των παραγόμενων SQL ερωτημάτων, ώστε να επιλεγεί η καταλληλότερη μέθοδος για την εφαρμογή AskSQL.

Μέθοδος 1 (M1): Περιληπτικό σχήμα

Στην πρώτη μέθοδο το LLM λαμβάνει μία συνοπτική περιγραφή του σχήματος της βάσης δεδομένων, με στόχο τον περιορισμό της κατανάλωσης tokens. Για κάθε πίνακα αποστέλλονται μόνο το όνομα του πίνακα και τα ονόματα των στηλών του. Για παράδειγμα: `HotelCustomer(cid, Name)`.

Μέθοδος 2 (M2): Πλήρες σχήμα της βάσης δεδομένων

Στη δεύτερη μέθοδο αποστέλλεται ολόκληρο το σχήμα της βάσης δεδομένων, δηλαδή οι εντολές `CREATE TABLE`. Με τον τρόπο αυτό το μοντέλο έχει πρόσβαση σε περισσότερες πληροφορίες σχετικά με τη δομή της βάσης, όπως οι τύποι δεδομένων, τα πρωτεύοντα και τα ξένα κλειδιά.

Μέθοδος 3 (M3): Πλήρες σχήμα και ενδεικτικά δεδομένα

Στην τρίτη μέθοδο αποστέλλεται το πλήρες σχήμα της βάσης δεδομένων, όπως και στη Μέθοδο 2, ενώ επιπλέον παρέχεται μία ενδεικτική εγγραφή για κάθε πίνακα. Στόχος είναι να διευκολυνθεί η κατανόηση του περιεχομένου της βάσης δεδομένων και του τρόπου αποθήκευσης των δεδομένων, ώστε να διερευνηθεί κατά πόσο η πρόσθετη αυτή πληροφορία συμβάλλει στην παραγωγή ορθότερων SQL ερωτημάτων.

4.4 Μεγέθη μέτρησης

Για τη σύγκριση των προσεγγίσεων που εξετάστηκαν στο πλαίσιο των πειραμάτων χρησιμοποιήθηκαν δύο βασικά μεγέθη: η `Execution Accuracy` (EA) και ο μέσος αριθμός των συνολικών tokens ανά ερώτημα.

Execution Accuracy (EA): Εκφράζει το ποσοστό των ερωτημάτων για τα οποία το παραγόμενο SQL ερώτημα επιστρέφει το ίδιο αποτέλεσμα με το αντίστοιχο ερώτημα αναφοράς. Ένα ερώτημα θεωρείται επιτυχές όταν το σύνολο των αποτελεσμάτων που επιστρέφει είναι ταυτόσημο με το αναμενόμενο αποτέλεσμα [12]. Το μέγεθος υπολογίζεται ως εξής:

$$EA = \frac{1}{N} \sum_{i=1}^N \mathbf{1}(V_i, G_i) \quad (4.1)$$

όπου:

- N είναι ο συνολικός αριθμός των ερωτημάτων του συνόλου αξιολόγησης,
- V_i είναι το αποτέλεσμα που επιστρέφει το παραγόμενο SQL ερώτημα για το i -οστό ερώτημα,
- G_i είναι το αντίστοιχο αποτέλεσμα του ερωτήματος αναφοράς (*Ground Truth*),
- $\mathbf{1}(\cdot)$ είναι η ενδεικτική συνάρτηση, η οποία επιστρέφει 1 όταν τα δύο αποτελέσματα είναι πανομοιότυπα και 0 σε κάθε άλλη περίπτωση.

Μέσος αριθμός tokens ανά ερώτημα: Η δεύτερη μετρική αφορά τον μέσο αριθμό των συνολικών *tokens* που καταναλώνονται για την παραγωγή ενός SQL ερωτήματος από το γλωσσικό μοντέλο. Η συγκεκριμένη μετρική χρησιμοποιείται για την εκτίμηση του υπολογιστικού κόστους κάθε προσέγγισης, καθώς μεγαλύτερος αριθμός *tokens* συνεπάγεται αυξημένο κόστος και μεγαλύτερο χρόνο επεξεργασίας.

4.5 Σύγκριση τεχνικών prompting

Στο πρώτο πείραμα συγκρίθηκαν οι τρεις τεχνικές prompting που παρουσιάστηκαν στην προηγούμενη ενότητα, με στόχο την επιλογή της καταλληλότερης προσέγγισης για την τελική υλοποίηση της εφαρμογής AskSQL. Για κάθε τεχνική μετρήθηκαν η *Execution Accuracy* (EA) και ο μέσος αριθμός των *tokens* ανά ερώτημα, χρησιμοποιώντας το σύνολο δεδομένων αξιολόγησης που περιγράφηκε προηγουμένως. Σε όλες τις μετρήσεις χρησιμοποιήθηκε το μοντέλο *GPT-4o mini*, ενώ οι υπόλοιπες συνθήκες εκτέλεσης διατηρήθηκαν ίδιες, ώστε οι διαφορές στα αποτελέσματα να αποδίδονται στον τρόπο με τον οποίο παρέχεται η πληροφορία στο μοντέλο. Τα αποτελέσματα παρουσιάζονται στον Πίνακα 4.4.

Πίνακας 4.4: Αποτελέσματα σύγκρισης των τεχνικών prompting

Μέθοδος	Εύκολες (EA%)	Μεσαίες (EA%)	Δύσκολες (EA%)	Σύνολο (EA%)	Avg. Tokens
M1	90	75	65	76.6	272
M2	90	95	65	83.3	1777
M3	100	95	75	90	1930

Στο συγκεκριμένο πειραματικό σύνολο, η μέθοδος M3 παρουσίασε την υψηλότερη *Execution Accuracy*, η οποία ανήλθε σε 90%. Η μέθοδος αυτή παρέχει στο γλωσσικό μοντέλο τόσο το πλήρες σχήμα της βάσης δεδομένων όσο και ενδεικτικά δεδομένα από κάθε πίνακα. Τα αποτελέσματα δείχνουν ότι, στις συγκεκριμένες συνθήκες των πειραμάτων, η πρόσθετη πληροφορία σχετικά με τη δομή και το περιεχόμενο της βάσης μπορεί να συμβάλει στη βελτίωση της ποιότητας των παραγόμενων SQL ερωτημάτων.

Η μέθοδος M3 παρουσίασε επίσης υψηλότερη απόδοση στις ερωτήσεις αυξημένης δυσκολίας σε σχέση με τις υπόλοιπες προσεγγίσεις. Επιπλέον, κατά την εξέταση των αποτελεσμάτων παρατηρήθηκε ότι η παροχή ενδεικτικών δεδομένων συνέβαλε στην αποφυγή ορισμένων σφαλμάτων που σχετίζονταν με τη σύνταξη των ερωτημάτων και με τη χρήση τιμών των δεδομένων, όπως η διάκριση μεταξύ κεφαλαίων και πεζών χαρακτήρων (*case sensitivity*).

Παρότι η μέθοδος M3 παρουσίασε τον μεγαλύτερο μέσο αριθμό `tokens` ανά ερώτημα, η αύξηση αυτή κρίθηκε αποδεκτή για τις ανάγκες της εφαρμογής, καθώς βασικό κριτήριο για την επιλογή της τεχνικής ήταν η μεγαλύτερη δυνατή ακρίβεια στην παραγωγή SQL ερωτημάτων. Με βάση τα αποτελέσματα του συγκεκριμένου πειράματος, η μέθοδος M3 επιλέχθηκε για χρήση στην τελική υλοποίηση της εφαρμογής.

4.6 Σύγκριση μοντέλων LLM

Στη δεύτερη φάση των πειραμάτων συγκρίθηκαν δύο γλωσσικά μοντέλα μεγάλου μεγέθους (LLMs), με στόχο τη διερεύνηση της επίδρασης της επιλογής του μοντέλου στην παραγωγή SQL ερωτημάτων στο πλαίσιο της εφαρμογής AskSQL. Για τη σύγκριση χρησιμοποιήθηκε η τεχνική *prompting* M3, η οποία επιλέχθηκε με βάση τα αποτελέσματα του προηγούμενου πειράματος, καθώς και το ίδιο σύνολο δεδομένων αξιολόγησης. Τα μοντέλα που εξετάστηκαν ήταν τα *GPT-4o mini* και *DeepSeek Chat*. Τα αποτελέσματα παρουσιάζονται στον Πίνακα 4.5.

Πίνακας 4.5: Σύγκριση των γλωσσικών μοντέλων

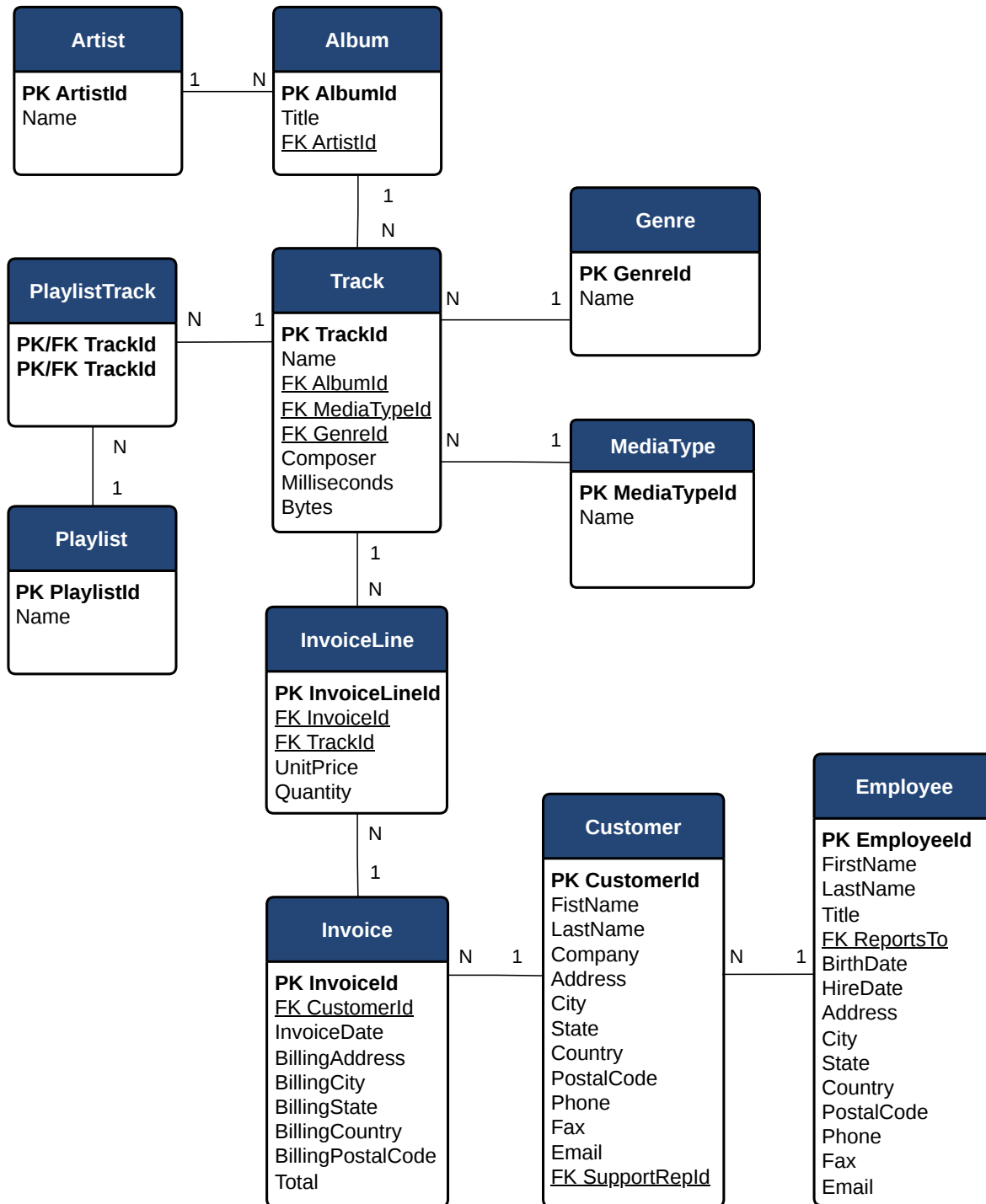
Μοντέλο	Εύκολες (EA %)	Μεσαίες (EA %)	Δύσκολες (EA %)	Σύνολο (E %)	Avg. Tokens
DeepSeek Chat	100	95	90	95	1990
GPT-4o mini	100	95	75	90	1930

Στο συγκεκριμένο πειραματικό σύνολο, το *DeepSeek Chat* παρουσίασε υψηλότερη τιμή *Execution Accuracy*, επιτυγχάνοντας ποσοστό 95%, σε σύγκριση με το *GPT-4o mini*, το οποίο πέτυχε ποσοστό 90%. Η διαφορά αυτή εμφανίζεται κυρίως στα ερωτήματα αυξημένης δυσκολίας, όπου το *DeepSeek Chat* πέτυχε ποσοστό 90%, έναντι 75% του *GPT-4o mini*.

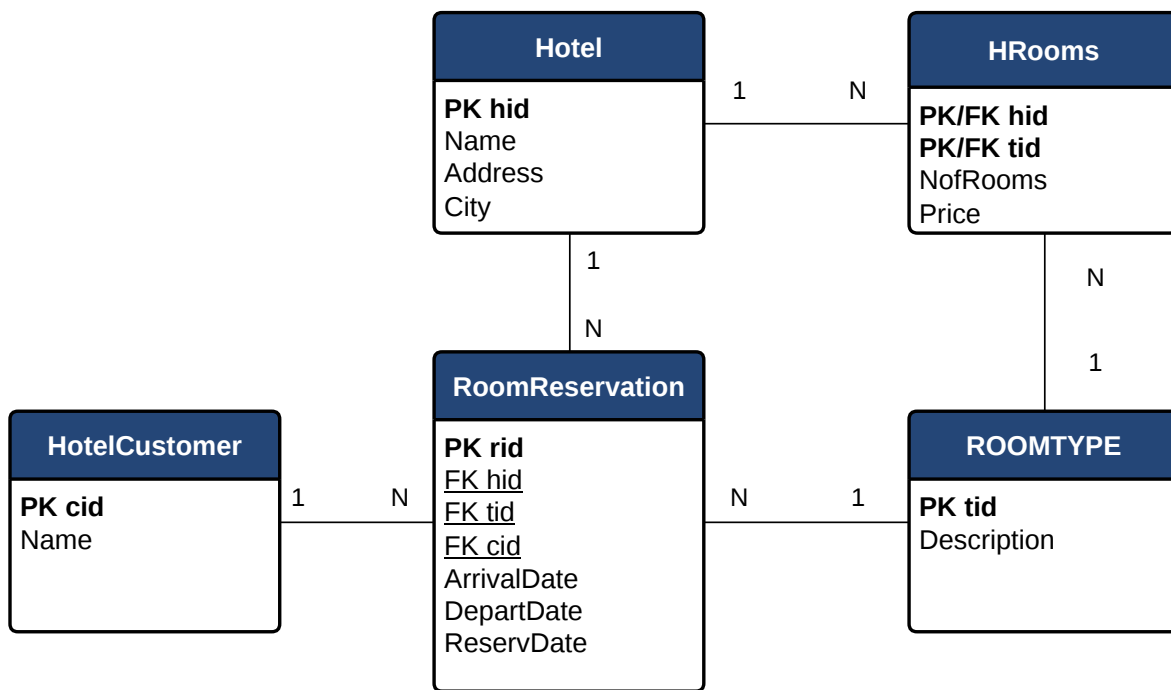
Τα αποτελέσματα αφορούν το συγκεκριμένο σύνολο δεδομένων αξιολόγησης, τις επιλεγμένες βάσεις δεδομένων και τις συνθήκες *prompting* που χρησιμοποιήθηκαν στα πειράματα. Επομένως, δεν αποτελούν γενική αξιολόγηση ή κατάταξη των δύο μοντέλων, αλλά συγκριτική διερεύνηση της συμπεριφοράς τους στις συγκεκριμένες συνθήκες της εφαρμογής AskSQL. Και τα δύο μοντέλα παρουσίασαν υψηλή *Execution Accuracy*, γεγονός που δείχνει ότι μπορούν να χρησιμοποιηθούν αποτελεσματικά για την παραγωγή SQL ερωτημάτων στο συγκεκριμένο πλαίσιο. Για τον λόγο αυτό, στην τελική υλοποίηση της εφαρμογής AskSQL ενσωματώθηκαν και τα δύο μοντέλα, παρέχοντας στον χρήστη τη δυνατότητα επιλογής του γλωσσικού μοντέλου που θα χρησιμοποιηθεί για τη μετατροπή ερωτημάτων φυσικής γλώσσας σε SQL.

Παράλληλα, μετρήθηκε ο μέσος αριθμός των συνολικών `tokens` που καταναλώθηκαν για κάθε ερώτημα. Ο αριθμός των `tokens` ανακτήθηκε από τα μεταδεδομένα (*metadata*) που επιστρέφει το API κάθε μοντέλου. Ωστόσο, επειδή οι διαφορετικοί πάροχοι εφαρμόζουν διαφορετικές μεθόδους *tokenization* και υπολογισμού των `tokens`, οι αντίστοιχες τιμές δεν μπορούν

να συγκριθούν απόλυτα μεταξύ τους. Επομένως, τα αποτελέσματα σχετικά με τα `tokens` θα πρέπει να θεωρηθούν ενδεικτικά και να χρησιμοποιηθούν μόνο ως μια προσεγγιστική εκτίμηση του υπολογιστικού κόστους κάθε μοντέλου.



Σχήμα 4.2: Διάγραμμα ER της βάσης Chinook



Σχήμα 4.3: Διάγραμμα ER της βάσης Hotel Room Reservation

Κεφάλαιο 5

Σχεδίαση και Υλοποίηση του AskSQL

5.1 Λειτουργικές Απαιτήσεις

Η παρούσα ενότητα παρουσιάζει τις βασικές λειτουργικές απαιτήσεις της εφαρμογής AskSQL, που περιγράφουν τις δυνατότητες του συστήματος από την πλευρά του χρήστη. Οι απαιτήσεις αυτές προέκυψαν κατά τη διαδικασία ανάλυσης και σχεδίασης της εφαρμογής και αποτυπώνουν τις κύριες λειτουργίες που προσφέρει το σύστημα. Η εφαρμογή AskSQL αναπτύχθηκε με σκοπό να επιτρέπει στους χρήστες να συνδέονται με σχεσιακές βάσεις δεδομένων και να εκτελούν SQL ερωτήματα είτε με την χρήση της γλώσσας SQL, είτε μέσω φυσικής γλώσσας και τεχνητής νοημοσύνης. Παράλληλα, παρέχονται δυνατότητες διαχείρισης λογαριασμών, διαχείρισης συνδέσεων βάσεων δεδομένων, προβολής αποτελεσμάτων και διαχείρισης χρηστών από διαχειριστές.

Εγγραφή Χρήστη: Ο χρήστης έχει τη δυνατότητα να δημιουργήσει νέο λογαριασμό μέσω της ειδικής φόρμας εγγραφής της εφαρμογής. Κατά τη διαδικασία εγγραφής συμπληρώνει απαραίτητα στοιχεία, όπως όνομα χρήστη (username), κωδικό πρόσβασης και διεύθυνση ηλεκτρονικού ταχυδρομείου (email). Μετά την επιτυχή καταχώριση των στοιχείων, αποστέλλεται αίτημα επιβεβαίωσης στην ηλεκτρονική διεύθυνση του χρήστη, ώστε να ολοκληρωθεί η ενεργοποίηση του λογαριασμού.

Σύνδεση Χρήστη: Ο χρήστης μπορεί να συνδεθεί στην εφαρμογή είτε μέσω της φόρμας εισόδου, εισάγοντας το όνομα χρήστη και τον κωδικό πρόσβασής του, είτε μέσω του λογαριασμού του Google αξιοποιώντας τη διαδικασία αυθεντικοποίησης OAuth 2.0. Εφόσον η αυθεντικοποίηση ολοκληρωθεί επιτυχώς, ο χρήστης αποκτά πρόσβαση στις λειτουργίες της εφαρμογής και μεταφέρεται στο κεντρικό περιβάλλον διαχείρισης.

Ανάκτηση Κωδικού Πρόσβασης: Σε περίπτωση που ο χρήστης ξεχάσει τον κωδικό πρόσβασής του, παρέχεται δυνατότητα επαναφοράς μέσω της λειτουργίας «Forgot Password». Ο χρήστης εισάγει τη διεύθυνση email που έχει συνδεθεί με τον λογαριασμό του και το σύστημα αποστέλλει μήνυμα ηλεκτρονικού ταχυδρομείου με σύνδεσμο επαναφοράς. Μέσω της ειδικής σελίδας επαναφοράς, ο χρήστης μπορεί να ορίσει νέο κωδικό πρόσβασης.

Επεξεργασία Στοιχείων Προφίλ: Ο χρήστης έχει τη δυνατότητα να προβάλλει και να επεξεργάζεται πληροφορίες που σχετίζονται με τον λογαριασμό του μέσω της σελίδας προφίλ της εφαρμογής. Στη σελίδα αυτή εμφανίζονται στοιχεία όπως το username, το email, ο ρόλος του χρήστη (Demo, Full ή Admin), καθώς και η ημερομηνία δημιουργίας του λογαριασμού. Επιπλέον, παρέχεται δυνατότητα αλλαγής κωδικού πρόσβασης.

Σύνδεση σε Βάση Δεδομένων: Ο χρήστης μπορεί να δημιουργήσει νέα σύνδεση με βάση δεδομένων, συμπληρώνοντας τα απαραίτητα στοιχεία σύνδεσης, όπως host, username, password και όνομα βάσης δεδομένων. Η εφαρμογή υποστηρίζει συστήματα διαχείρισης βάσεων δεδομένων MySQL και PostgreSQL, τόσο μέσω απευθείας σύνδεσης TCP/IP όσο και μέσω ασφαλούς σύνδεσης SSH tunneling.

Επανασύνδεση σε Αποθηκευμένη Βάση: Η εφαρμογή παρέχει τη δυνατότητα αποθήκευσης προηγούμενων συνδέσεων, ώστε ο χρήστης να μπορεί να επανασυνδεθεί γρήγορα σε μία βάση δεδομένων χωρίς να απαιτείται εκ νέου εισαγωγή όλων των στοιχείων σύνδεσης.

Μετονομασία Αποθηκευμένης Σύνδεσης: Ο χρήστης έχει τη δυνατότητα να τροποποιήσει το όνομα μιας αποθηκευμένης σύνδεσης, διευκολύνοντας έτσι την οργάνωση και αναγνώριση πολλαπλών βάσεων δεδομένων μέσα στην εφαρμογή.

Διαγραφή Αποθηκευμένης Σύνδεσης: Ο χρήστης μπορεί να διαγράψει μία ή περισσότερες αποθηκευμένες συνδέσεις βάσεων δεδομένων από το προφίλ του.

Αποσύνδεση Χρήστη: Η εφαρμογή παρέχει δυνατότητα ασφαλούς αποσύνδεσης του χρήστη μέσω σχετικής επιλογής στο μενού πλοήγησης. Κατά την αποσύνδεση διαγράφονται τα στοιχεία αυθεντικοποίησης που χρησιμοποιούνται για την πρόσβαση στην εφαρμογή, αποτρέποντας τη μη εξουσιοδοτημένη πρόσβαση από τον συγκεκριμένο φυλλομετρητή.

Προβολή Δομής Βάσεων Δεδομένων: Ο χρήστης μπορεί να προβάλλει τις διαθέσιμες βάσεις δεδομένων στις οποίες έχει πρόσβαση. Επιλέγοντας μία βάση δεδομένων, εμφανίζονται οι πίνακες που περιέχει, ενώ παρέχονται και πληροφορίες σχετικά με τις στήλες κάθε πίνακα, όπως το όνομα, ο τύπος δεδομένων, τα πρωτεύοντα κλειδιά και άλλες ιδιότητες.

Εκτέλεση SQL Ερωτημάτων: Η εφαρμογή επιτρέπει στον χρήστη να εισάγει και να εκτελεί SQL ερωτήματα μέσω ειδικού πεδίου επεξεργασίας. Τα ερωτήματα αποστέλλονται στη συνδεδεμένη βάση δεδομένων και τα αποτελέσματα εμφανίζονται άμεσα στο περιβάλλον της εφαρμογής.

Προβολή Ιστορικού Ερωτημάτων: Ο χρήστης έχει πρόσβαση στο ιστορικό των πρόσφατων SQL ερωτημάτων που έχει εκτελέσει. Το ιστορικό αυτό περιλαμβάνει τα τελευταία επιτυχημένα ερωτήματα και διευκολύνει την επανεκτέλεσή τους.

Διαγραφή Ιστορικού Ερωτημάτων: Η εφαρμογή επιτρέπει τη διαγραφή του ιστορικού, δίνοντας στον χρήστη τη δυνατότητα καλύτερης διαχείρισης των αποθηκευμένων ερωτημάτων του.

Υποβολή Ερωτήματος σε Φυσική Γλώσσα: Ο χρήστης μπορεί να υποβάλει ερωτήματα χρησιμοποιώντας φυσική γλώσσα αντί για SQL σύνταξη. Η ερώτηση αποστέλλεται σε μοντέλο τεχνητής νοημοσύνης (LLM), το οποίο αναλαμβάνει τη μετατροπή του σε κατάλληλο SQL ερώτημα.

Επεξεργασία Παραγόμενου SQL Ερωτήματος: Πριν από την εκτέλεση ενός SQL ερωτήματος που έχει παραχθεί από το μοντέλο τεχνητής νοημοσύνης, ο χρήστης μπορεί να το επεξεργαστεί χειροκίνητα. Η λειτουργία αυτή επιτρέπει τη διόρθωση ή βελτιστοποίηση του παραγόμενου ερωτήματος.

Εισαγωγή Φωνητικού Ερωτήματος: Η εφαρμογή υποστηρίζει την υποβολή ερωτημάτων μέσω φωνητικής εισόδου. Ο χρήστης μπορεί να εκφωνήσει το αίτημά του, το οποίο μετατρέπεται σε κείμενο και στη συνέχεια επεξεργάζεται από το σύστημα για τη δημιουργία του κατάλληλου SQL ερωτήματος.

Προβολή Αποτελεσμάτων Ερωτημάτων: Τα αποτελέσματα των SQL ερωτημάτων εμφανίζονται σε μορφή πίνακα στο κεντρικό περιβάλλον της εφαρμογής. Για τη βελτίωση της απόδοσης

σε μεγάλα σύνολα δεδομένων, χρησιμοποιείται μηχανισμός pagination, επιτρέποντας τη σταδιακή φόρτωση των εγγραφών.

Εξαγωγή Αποτελεσμάτων σε CSV: Ο χρήστης έχει τη δυνατότητα να εξάγει τα αποτελέσματα ενός ερωτήματος σε αρχείο μορφής CSV. Η λειτουργία αυτή διευκολύνει την αποθήκευση, μεταφορά και περαιτέρω επεξεργασία των δεδομένων.

Διαχείριση Χρηστών από Διαχειριστή: Ο χρήστης με ρόλο διαχειριστή (Admin) έχει πρόσβαση σε ειδικό πίνακα διαχείρισης χρηστών. Μέσω αυτού μπορεί να προβάλλει όλους τους εγγεγραμμένους χρήστες της εφαρμογής, καθώς και βασικές πληροφορίες που σχετίζονται με αυτούς.

Τροποποίηση Ρόλων Χρηστών: Ο διαχειριστής μπορεί να τροποποιεί τον ρόλο ενός χρήστη μέσα από το περιβάλλον διαχείρισης. Η εφαρμογή υποστηρίζει διαφορετικά επίπεδα δικαιωμάτων, όπως Demo, Full και Admin, τα οποία καθορίζουν το βαθμό χρήσης της εφαρμογής.

Έλεγχος Πρόσβασης Βάσει Ρόλων: Η εφαρμογή χρησιμοποιεί μηχανισμό ελέγχου πρόσβασης βάσει ρόλων χρηστών. Οι χρήστες με ρόλο Demo διαθέτουν περιορισμένες δυνατότητες χρήσης (χρήση μόνο Select ερωτημάτων), οι χρήστες Full έχουν πρόσβαση στο πλήρες σύνολο λειτουργιών της εφαρμογής, ενώ οι χρήστες Admin διαθέτουν επιπλέον δικαιώματα διαχείρισης χρηστών και συστήματος.

5.2 Αρχιτεκτονική του AskSQL

Όπως αναφέρθηκε και στις προηγούμενες ενότητες, στόχος της εφαρμογής AskSQL είναι η διευκόλυνση της πρόσβασης και της διαχείρισης σχεσιακών βάσεων δεδομένων μέσω μιας σύγχρονης διαδικτυακής εφαρμογής. Η εφαρμογή απευθύνεται τόσο σε χρήστες με εμπειρία στις βάσεις δεδομένων όσο και σε αρχάριους, καθώς ενσωματώνει μηχανισμό αυτόματης μετατροπής ερωτημάτων φυσικής γλώσσας σε ισοδύναμα ερωτήματα SQL με τη χρήση Μεγάλων Γλωσσικών Μοντέλων (Large Language Models - LLMs).

Μετά την επιτυχή αυθεντικοποίηση, οι εγγεγραμμένοι χρήστες αποκτούν πρόσβαση στις αποθηκευμένες συνδέσεις τους με βάσεις δεδομένων, τις οποίες μπορούν να διαχειρίζονται μέσω της εφαρμογής. Συγκεκριμένα, παρέχεται η δυνατότητα δημιουργίας νέας σύνδεσης, επανασύνδεσης σε ήδη αποθηκευμένες βάσεις δεδομένων χωρίς να απαιτείται η εκ νέου εισαγωγή των στοιχείων σύνδεσης, μετονομασίας και διαγραφής των συνδέσεων. Η εφαρμογή υποστηρίζει σύνδεση τόσο μέσω του πρωτοκόλλου TCP/IP όσο και μέσω ασφαλούς σύνδεσης SSH tunneling, προσφέροντας μεγαλύτερη ευελιξία και ασφάλεια.

Επιπλέον, παρέχεται πλήρης εξερεύνηση της συνδεδεμένης βάσης δεδομένων. Ο χρήστης μπορεί να επιλέξει οποιαδήποτε διαθέσιμη βάση δεδομένων, να προβάλει τους πίνακες που αυτή περιέχει, καθώς και τη δομή κάθε πίνακα, συμπεριλαμβανομένων των στηλών και των αντίστοιχων τύπων δεδομένων.

Η εφαρμογή υποστηρίζει δύο διαφορετικούς τρόπους υποβολής ερωτημάτων προς τη βάση δεδομένων. Ο πρώτος αφορά τη χειροκίνητη συγγραφή και εκτέλεση SQL ερωτημάτων από τον ίδιο τον χρήστη. Ο δεύτερος βασίζεται στη χρήση Μεγάλων Γλωσσικών Μοντέλων (LLMs), μέσω των οποίων ο χρήστης μπορεί να διατυπώσει το ερώτημά του σε φυσική γλώσσα. Η εισαγωγή του ερωτήματος μπορεί να πραγματοποιηθεί είτε μέσω κειμένου είτε με χρήση φωνητικών εντολών, αξιοποιώντας τη λειτουργία μετατροπής ομιλίας σε κείμενο (Speech-to-Text). Στη συνέχεια, ο χρήστης επιλέγει το επιθυμητό γλωσσικό μοντέλο, εξετάζει το παραγόμενο SQL

ερώτημα, έχει τη δυνατότητα να το τροποποιήσει εφόσον απαιτείται και τελικά να το εκτελέσει στη βάση δεδομένων.

Τα αποτελέσματα των ερωτημάτων παρουσιάζονται σε μορφή πίνακα, διευκολύνοντας την ανάγνωση και την ανάλυση των δεδομένων. Παράλληλα, παρέχεται η δυνατότητα εξαγωγής των αποτελεσμάτων σε μορφή CSV, καθώς και διαχείρισης του ιστορικού των ερωτημάτων που έχουν εκτελεστεί.

Όλες οι παραπάνω λειτουργίες είναι διαθέσιμες τόσο μέσω της γραφικής διεπαφής της εφαρμογής όσο και μέσω του Web API. Η πρόσβαση στη γραφική διεπαφή πραγματοποιείται είτε με χρήση των στοιχείων σύνδεσης (όνομα χρήστη και κωδικός πρόσβασης) είτε μέσω λογαριασμού Google, αξιοποιώντας τη διαδικασία αυθεντικοποίησης OAuth 2.0. Αντίστοιχα, η πρόσβαση στο Web API πραγματοποιείται μέσω των στοιχείων αυθεντικοποίησης του χρήστη, ενώ τα αποτελέσματα των αιτημάτων επιστρέφονται σε μορφή JSON, ώστε να είναι δυνατή η εύκολη αξιοποίησή τους από εξωτερικές εφαρμογές και υπηρεσίες.

Το Σχήμα 5.1 παρουσιάζει συνοπτικά τη ροή λειτουργίας της εφαρμογής, αποτυπώνοντας τα βασικά βήματα που ακολουθούνται από την είσοδο του χρήστη μέχρι την εκτέλεση ενός ερωτήματος και την επιστροφή των αποτελεσμάτων.

Η αρχιτεκτονική του AskSQL αποτελείται από πέντε βασικά υποσυστήματα: το Frontend, το Web API (Backend), την εσωτερική βάση δεδομένων της εφαρμογής, τις εξωτερικές βάσεις δεδομένων του χρήστη και τις εξωτερικές υπηρεσίες. Ο πυρήνας της εφαρμογής είναι το Web API, το οποίο έχει αναπτυχθεί στη γλώσσα προγραμματισμού PHP και λειτουργεί ως ενδιάμεσο επίπεδο μεταξύ της γραφικής διεπαφής, των βάσεων δεδομένων και των εξωτερικών υπηρεσιών. Μέσω αυτού υλοποιούνται οι λειτουργίες αυθεντικοποίησης, η διαχείριση των συνδέσεων των χρηστών, η επικοινωνία με τις συνδεδεμένες βάσεις δεδομένων, η διαχείριση του ιστορικού ερωτημάτων, καθώς και η εκτέλεση των SQL ερωτημάτων.

Στο Backend χρησιμοποιείται η MySQL ως βάση δεδομένων της εφαρμογής, στην οποία αποθηκεύονται οι πληροφορίες που αφορούν τους χρήστες, τις αποθηκευμένες συνδέσεις, το ιστορικό ερωτημάτων και τα υπόλοιπα δεδομένα που απαιτούνται για τη λειτουργία του συστήματος. Παράλληλα, το Web API αλληλεπιδρά με μια σειρά εξωτερικών υπηρεσιών. Συγκεκριμένα, χρησιμοποιείται η βιβλιοθήκη PHPMailer για την αποστολή μηνυμάτων ηλεκτρονικού ταχυδρομείου που αφορούν την επιβεβαίωση λογαριασμού και την επαναφορά κωδικού πρόσβασης. Επιπλέον, αξιοποιούνται τα APIs των Μεγάλων Γλωσσικών Μοντέλων για τη μετατροπή ερωτημάτων φυσικής γλώσσας σε SQL, ενώ η υπηρεσία Google OAuth χρησιμοποιείται για την αυθεντικοποίηση χρηστών μέσω λογαριασμού Google. Τέλος, η λειτουργία εισαγωγής ερωτημάτων μέσω φωνητικής εντολής υλοποιείται με χρήση του Web Speech API, το οποίο μετατρέπει την ομιλία του χρήστη σε κείμενο πριν από την αποστολή της στο επιλεγμένο γλωσσικό μοντέλο.

Η γραφική διεπαφή της εφαρμογής έχει αναπτυχθεί με χρήση των τεχνολογιών HTML, CSS, Bootstrap και JavaScript, όπως παρουσιάστηκε στο προηγούμενο κεφάλαιο. Μέσω αυτής παρέχεται ένα σύγχρονο και εύχρηστο περιβάλλον αλληλεπίδρασης, το οποίο επιτρέπει την εύκολη πρόσβαση σε όλες τις λειτουργίες της εφαρμογής. Η συνολική αρχιτεκτονική του AskSQL παρουσιάζεται στο Σχήμα 5.2.

5.3 Χρήστες της Εφαρμογής

Η εφαρμογή υλοποιεί μηχανισμό ελέγχου πρόσβασης βάσει ρόλων (Role-Based Access Control - RBAC), μέσω του οποίου καθορίζονται τα δικαιώματα κάθε χρήστη. Συνολικά υποστηρίζονται τρεις κατηγορίες χρηστών: Admin, Demo και Full.

Admin: Οι χρήστες με ρόλο Admin διαθέτουν πλήρη δικαιώματα διαχείρισης της εφαρμογής. Συγκεκριμένα, μπορούν να προβάλλουν όλους τους εγγεγραμμένους χρήστες, καθώς και να τροποποιούν τα δικαιώματα πρόσβασης τους, μεταβάλλοντας τον ρόλο τους από Demo σε Full και αντίστροφα. Η απόδοση του ρόλου Admin πραγματοποιείται απευθείας μέσω της βάσης δεδομένων ή από ήδη εξουσιοδοτημένο διαχειριστή.

Demo: Οι χρήστες με ρόλο Demo διαθέτουν περιορισμένα δικαιώματα εκτέλεσης ερωτημάτων. Συγκεκριμένα, επιτρέπεται αποκλειστικά η εκτέλεση ερωτημάτων ανάκτησης δεδομένων (SELECT), χωρίς δυνατότητα τροποποίησης της βάσης δεδομένων.

Full: Οι χρήστες με ρόλο Full διαθέτουν πλήρη δικαιώματα εκτέλεσης SQL εντολών. Εκτός από ερωτήματα ανάκτησης δεδομένων (SELECT), μπορούν να εκτελούν εντολές ορισμού και τροποποίησης της βάσης δεδομένων (DDL και DML), όπως CREATE, ALTER, INSERT, UPDATE, DELETE και DROP.

5.4 Υλοποίηση του Backend

5.4.1 Σχεδίαση της βάσης δεδομένων

Όπως αναφέρθηκε και στην προηγούμενη ενότητα, η εφαρμογή χρησιμοποιεί το Σύστημα Διαχείρισης Βάσεων Δεδομένων MySQL για την αποθήκευση και διαχείριση των δεδομένων της. Η βάση δεδομένων υποστηρίζει όλες τις βασικές λειτουργίες της εφαρμογής, όπως η εγγραφή και αυθεντικοποίηση των χρηστών, η αποθήκευση των συνδέσεων με εξωτερικές βάσεις δεδομένων, καθώς και η διατήρηση του ιστορικού των ερωτημάτων που εκτελούνται μέσω της εφαρμογής.

Για την ανάπτυξη και διαχείριση της βάσης χρησιμοποιήθηκε το εργαλείο phpMyAdmin, μέσω του οποίου δημιουργήθηκε η βάση δεδομένων με όνομα `naturalsqladmin`. Το phpMyAdmin αποτελεί μία διαδικτυακή εφαρμογή ανοικτού κώδικα, η οποία παρέχει ένα φιλικό γραφικό περιβάλλον για τη διαχείριση βάσεων δεδομένων MySQL και MariaDB, διευκολύνοντας την εκτέλεση λειτουργιών όπως η δημιουργία πινάκων, η επεξεργασία δεδομένων και η εκτέλεση SQL ερωτημάτων.

Για την κάλυψη των λειτουργικών απαιτήσεων της εφαρμογής σχεδιάστηκαν και υλοποιήθηκαν τέσσερις βασικοί πίνακες: `users`, `password_resets`, `connections` και `query_history`. Ο πίνακας `users` αποθηκεύει τα στοιχεία των εγγεγραμμένων χρηστών της εφαρμογής, καθώς και πληροφορίες που σχετίζονται με την αυθεντικοποίηση και τα δικαιώματα πρόσβασης τους. Ο πίνακας `password_resets` χρησιμοποιείται για την προσωρινή αποθήκευση των απαραίτητων πληροφοριών που απαιτούνται κατά τη διαδικασία επαναφοράς κωδικού πρόσβασης. Ο πίνακας `connections` αποθηκεύει τις συνδέσεις των χρηστών με εξωτερικές βάσεις δεδομένων, καθώς και όλα τα απαραίτητα στοιχεία που απαιτούνται για την πραγματοποίηση της σύνδεσης. Τέλος, ο πίνακας `query_history` διατηρεί το ιστορικό των SQL ερωτημάτων που έχουν εκτελεστεί από κάθε χρήστη, επιτρέποντας την προβολή και διαχείριση των προηγούμενων ερωτημάτων.

μέσω της εφαρμογής.

Ο Πίνακας *users* 5.1 περιλαμβάνει εννέα βασικά πεδία, τα οποία περιγράφονται παρακάτω:

- *id*: Αποτελεί το πρωτεύον κλειδί (Primary Key) του πίνακα και το μοναδικό αναγνωριστικό κάθε χρήστη. Το πεδίο χρησιμοποιεί την ιδιότητα `AUTO_INCREMENT`, ώστε η τιμή του να δημιουργείται αυτόματα κατά την εισαγωγή νέας εγγραφής.
- *username*: Αποθηκεύει το όνομα χρήστη (username) που χρησιμοποιείται για τη σύνδεση στην εφαρμογή.
- *email*: Αποθηκεύει τη διεύθυνση ηλεκτρονικού ταχυδρομείου του χρήστη.
- *password*: Αποθηκεύει τον κωδικό πρόσβασης σε μορφή hash.
- *role*: Καθορίζει τον ρόλο και τα δικαιώματα του χρήστη στην εφαρμογή. Οι διαθέσιμες τιμές είναι `admin`, `demo` και `full`.
- *is_verified*: Δηλώνει αν ο λογαριασμός έχει επιβεβαιωθεί μέσω ηλεκτρονικού ταχυδρομείου. Λαμβάνει τις τιμές 0 (μη επιβεβαιωμένος) ή 1 (επιβεβαιωμένος).
- *verification_token*: Αποθηκεύει το μοναδικό διακριτικό (token) που χρησιμοποιείται για την επιβεβαίωση του λογαριασμού.
- *verify_expires*: Αποθηκεύει την ημερομηνία και ώρα λήξης του token επιβεβαίωσης.
- *created_at*: Αποθηκεύει την ημερομηνία και ώρα δημιουργίας του λογαριασμού.

Πίνακας 5.1: Πίνακας *users* της βάσης δεδομένων

Name	Datatype	Length/Set
id	INT	10
username	VARCHAR	100
email	VARCHAR	255
password	VARCHAR	255
role	ENUM	admin,demo,full
is_verified	TINYINT	-
verification_token	VARCHAR	255
verify_expires	TIMESTAMP	-
created_at	TIMESTAMP	-

Ο Πίνακας *password_resets* 5.2 περιλαμβάνει πέντε βασικά πεδία, τα οποία χρησιμοποιούνται για την ασφαλή διαδικασία επαναφοράς κωδικού πρόσβασης των χρηστών.

- *id*: Αποτελεί το πρωτεύον κλειδί (Primary Key) του πίνακα και το μοναδικό αναγνωριστικό κάθε εγγραφής. Το πεδίο χρησιμοποιεί την ιδιότητα *AUTO_INCREMENT*, ώστε η τιμή του να δημιουργείται αυτόματα κατά την εισαγωγή νέας εγγραφής.
- *user_id*: Αποτελεί ξένο κλειδί (Foreign Key), το οποίο αναφέρεται στο πεδίο *id* του πίνακα *users*. Χρησιμοποιείται για τη συσχέτιση κάθε αιτήματος επαναφοράς κωδικού με τον αντίστοιχο χρήστη.
- *token*: Αποθηκεύει το μοναδικό διακριτικό (token) που δημιουργείται κατά τη διαδικασία επαναφοράς κωδικού πρόσβασης και χρησιμοποιείται για την ταυτοποίηση του αιτήματος.
- *expires_at*: Περιέχει την ημερομηνία και ώρα λήξης του token, μετά την οποία αυτό θεωρείται μη έγκυρο και δεν μπορεί να χρησιμοποιηθεί.
- *created_at*: Περιλαμβάνει την ημερομηνία και ώρα δημιουργίας του αιτήματος επαναφοράς κωδικού.

Πίνακας 5.2: Πίνακας password_resets της βάσης δεδομένων

Name	Datatype	Length/Set
id	INT	10
user_id	INT	10
token	VARCHAR	255
expires_at	TIMESTAMP	-
created_at	TIMESTAMP	-

Ο Πίνακας *connections* 5.3 περιλαμβάνει δεκαπέντε βασικά πεδία, τα οποία χρησιμοποιούνται για την αποθήκευση των συνδέσεων των χρηστών με εξωτερικές βάσεις δεδομένων.

- *id*: Αποτελεί το πρωτεύον κλειδί (Primary Key) του πίνακα και το μοναδικό αναγνωριστικό κάθε σύνδεσης. Το πεδίο χρησιμοποιεί την ιδιότητα *AUTO_INCREMENT*, ώστε η τιμή του να δημιουργείται αυτόματα κατά την εισαγωγή νέας εγγραφής.
- *user_id*: Αποτελεί ξένο κλειδί (Foreign Key), το οποίο αναφέρεται στο πεδίο *id* του πίνακα *users*. Χρησιμοποιείται για τη συσχέτιση κάθε σύνδεσης με τον χρήστη στον οποίο ανήκει.
- *db_type*: Αποθηκεύει τον τύπο του συστήματος διαχείρισης βάσης δεδομένων (π.χ. *mysql* ή *pgsql*).
- *name*: Αποθηκεύει το όνομα που έχει ορίσει ο χρήστης για τη συγκεκριμένη σύνδεση.
- *host*: Αποθηκεύει τη διεύθυνση (hostname ή IP) του διακομιστή στον οποίο βρίσκεται η βάση δεδομένων.

- *port*: Αποθηκεύει τον αριθμό της θύρας (port) που χρησιμοποιείται για τη σύνδεση με τη βάση δεδομένων.
- *db_name*: Αποθηκεύει το όνομα της βάσης δεδομένων στην οποία πραγματοποιείται η σύνδεση. Το πεδίο μπορεί να παραμένει κενό όταν δεν απαιτείται επιλογή συγκεκριμένης βάσης.
- *db_username*: Αποθηκεύει το όνομα χρήστη που χρησιμοποιείται για την αυθεντικοποίηση στη βάση δεδομένων.
- *db_password*: Αποθηκεύει, σε κρυπτογραφημένη μορφή, τον κωδικό πρόσβασης που χρησιμοποιείται για τη σύνδεση στην εξωτερική βάση δεδομένων.
- *ssh*: Δηλώνει αν η σύνδεση πραγματοποιείται απευθείας μέσω TCP/IP ή μέσω SSH tunneling. Το πεδίο λαμβάνει τις τιμές 0 και 1, όπου η τιμή 0 αντιστοιχεί σε απευθείας σύνδεση και η τιμή 1 σε σύνδεση μέσω SSH.
- *ssh_host*: Αποθηκεύει τη διεύθυνση του SSH διακομιστή που χρησιμοποιείται για τη δημιουργία της ασφαλούς σύνδεσης.
- *ssh_port*: Αποθηκεύει τον αριθμό της θύρας (port) του SSH διακομιστή.
- *ssh_user*: Αποθηκεύει το όνομα χρήστη που χρησιμοποιείται για την αυθεντικοποίηση στον SSH διακομιστή.
- *ssh_password*: Αποθηκεύει σε κρυπτογραφημένη μορφή, τον κωδικό πρόσβασης του λογαριασμού SSH.
- *created_at*: Αποθηκεύει την ημερομηνία και ώρα δημιουργίας της σύνδεσης.

Ο Πίνακας *query_history* 5.4 περιλαμβάνει επτά βασικά πεδία, τα οποία χρησιμοποιούνται για την αποθήκευση του ιστορικού των ερωτημάτων που εκτελούν οι χρήστες.

- *id*: Αποτελεί το πρωτεύον κλειδί (Primary Key) του πίνακα και το μοναδικό αναγνωριστικό κάθε εγγραφής. Το πεδίο χρησιμοποιεί την ιδιότητα `AUTO_INCREMENT`, ώστε η τιμή του να δημιουργείται αυτόματα κατά την εισαγωγή νέας εγγραφής.
- *connection_id*: Αποτελεί ξένο κλειδί (Foreign Key), το οποίο αναφέρεται στο πεδίο *id* του πίνακα *connections*. Χρησιμοποιείται για τη συσχέτιση κάθε ερωτήματος με τη σύνδεση της βάσης δεδομένων στην οποία εκτελέστηκε.
- *database_name*: Αποθηκεύει το όνομα της βάσης δεδομένων στην οποία εκτελέστηκε το αντίστοιχο ερώτημα.
- *mode*: Αποθηκεύει τον τρόπο δημιουργίας του ερωτήματος. Το πεδίο λαμβάνει τις τιμές `sql` ή `nl`, ανάλογα με το αν το ερώτημα δημιουργήθηκε με χειροκίνητη σύνταξη SQL ή μέσω φυσικής γλώσσας.

Πίνακας 5.3: Πίνακας connections της βάσης δεδομένων

Name	Datatype	Length/Set
id	INT	10
user_id	INT	10
db_type	ENUM	mysql, pgsql
name	VARCHAR	255
host	VARCHAR	255
port	INT	11
db_name	VARCHAR	255
db_username	VARCHAR	255
db_password	text	-
ssh	TINYINT	1
ssh_host	VARCHAR	255
ssh_port	INT	11
ssh_user	VARCHAR	255
ssh_password	text	-
created_at	TIMESTAMP	-

- *question*: Αποθηκεύει την ερώτηση που εισήγαγε ο χρήστης σε φυσική γλώσσα. Το πεδίο συμπληρώνεται μόνο όταν το ερώτημα δημιουργείται μέσω της λειτουργίας Natural Language.
- *sql_query*: Αποθηκεύει το τελικό SQL ερώτημα που εκτελέστηκε στη βάση δεδομένων.
- *created_at*: Αποθηκεύει την ημερομηνία και ώρα δημιουργίας της εγγραφής στο ιστορικό ερωτημάτων.

Στο Σχήμα 5.3 παρουσιάζεται το διάγραμμα οντοτήτων-συσχετίσεων (ER diagram) της Βάσης Δεδομένων της εφαρμογής, το οποίο δημιουργήθηκε με τη χρήση του λογισμικού drawio.

5.4.2 Web API

Το Web API αποτελεί τον πυρήνα λειτουργίας της εφαρμογής *AskSQL*, καθώς μέσω αυτού υλοποιούνται όλες οι βασικές λειτουργίες του συστήματος, από την αυθεντικοποίηση των χρηστών έως την εκτέλεση SQL ερωτημάτων και την αλληλεπίδραση με τα μοντέλα μεγάλων γλωσσών (LLMs). Αποτελεί το ενδιάμεσο επίπεδο επικοινωνίας μεταξύ του Frontend και του Backend, αναλαμβάνοντας την επεξεργασία των αιτημάτων των χρηστών και την επιστροφή των αντίστοιχων αποτελεσμάτων.

Πίνακας 5.4: Πίνακας query_history της βάσης δεδομένων

Name	Datatype	Length/Set
id	INT	10
connection_id	INT	10
database_name	VARCHAR	255
mode	ENUM	sql, nl
question	text	-
sql_query	text	-
created_at	TIMESTAMP	-

Η επικοινωνία μεταξύ του Frontend και του Web API υλοποιείται μέσω ασύγχρονων HTTP αιτημάτων, αξιοποιώντας το Fetch API της JavaScript. Η προσέγγιση αυτή επιτρέπει την αποστολή αιτημάτων και την επεξεργασία των αποκρίσεων χωρίς ανανέωση της ιστοσελίδας, συμβάλλοντας στην άμεση αλληλεπίδραση του χρήστη με την εφαρμογή.

Κάθε λειτουργία της εφαρμογής υλοποιείται μέσω ενός ξεχωριστού API endpoint. Για την εκτέλεση μιας λειτουργίας απαιτείται η κλήση του αντίστοιχου endpoint, το οποίο εκτελεί το κατάλληλο αρχείο PHP και επιστρέφει τα αποτελέσματα σε μορφή JSON. Ανάλογα με τη λειτουργία που εκτελείται, χρησιμοποιούνται οι μέθοδοι HTTP GET ή POST, ενώ κάθε αίτημα συνοδεύεται από τις απαραίτητες παραμέτρους εισόδου.

Τα endpoints του Web API διακρίνονται σε δύο κατηγορίες: τα δημόσια και τα προστατευμένα.

Δημόσια (Public) endpoints, τα οποία είναι διαθέσιμα χωρίς να απαιτείται προηγούμενη αυθεντικοποίηση του χρήστη και αφορούν λειτουργίες όπως η εγγραφή, η σύνδεση, η επιβεβαίωση λογαριασμού και η ανάκτηση κωδικού πρόσβασης.

Προστατευμένα (Protected) endpoints, στα οποία η πρόσβαση επιτρέπεται μόνο σε αυθεντικοποιημένους χρήστες μέσω έγκυρου JSON Web Token (JWT). Τα endpoints αυτά υλοποιούν όλες τις κύριες λειτουργίες της εφαρμογής, όπως η διαχείριση συνδέσεων βάσεων δεδομένων, η εξερεύνηση του σχήματος, η δημιουργία SQL ερωτημάτων μέσω LLMs, η εκτέλεση ερωτημάτων, η διαχείριση του ιστορικού καθώς και οι λειτουργίες διαχείρισης χρηστών από τον διαχειριστή (Administrator).

Το σύνολο των δημόσιων API endpoints παρουσιάζεται στον Πίνακα 5.5, ενώ τα προστατευμένα endpoints παρουσιάζονται στον Πίνακα 5.6. Όλα τα αρχεία του Web API είναι οργανωμένα σε ξεχωριστό κατάλογο της εφαρμογής, διευκολύνοντας τη συντήρηση και την επέκταση του συστήματος.

5.4.3 Επικοινωνία με τα Μεγάλα Γλωσσικά Μοντέλα (LLMs)

Η βασική λειτουργία της εφαρμογής βασίζεται στη χρήση Μεγάλων Γλωσσικών Μοντέλων (LLMs), τα οποία αναλαμβάνουν τη μετατροπή ερωτημάτων φυσικής γλώσσας σε ισοδύναμα ερωτήματα SQL. Με τον τρόπο αυτό, ο χρήστης μπορεί να αλληλεπιδρά με τη σχεσιακή βάση δεδομένων χρησιμοποιώντας καθημερινή γλώσσα, χωρίς να απαιτείται γνώση της γλώσσας

Πίνακας 5.5: Public API Endpoints της εφαρμογής

A/A	ΜΕΘΟΔΟΣ HTTP	ENDPOINT	ΛΕΙΤΟΥΡΓΙΑ
1	POST	register.php	Εγγραφή νέου χρήστη
2	POST	login.php	Σύνδεση χρήστη
3	GET	login_google.php	Έναρξη σύνδεσης μέσω Google OAuth
4	GET	google_callback.php	Callback αυθεντικοποίησης Google
5	POST	forgot_password.php	Αποστολή email επαναφοράς κωδικού
6	POST	reset_password.php	Ορισμός νέου κωδικού πρόσβασης
7	GET	verify.php	Επιβεβαίωση λογαριασμού μέσω email

SQL.

Η εφαρμογή υποστηρίζει την επιλογή διαφορετικών πάροχων LLM, προσφέροντας τη δυνατότητα χρήσης των μοντέλων GPT-4o mini της OpenAI και DeepSeek Chat. Για την επικοινωνία με τα αντίστοιχα APIs απαιτείται η δημιουργία και ασφαλής αποθήκευση προσωπικών API keys, τα οποία χρησιμοποιούνται για την αυθεντικοποίηση κάθε αιτήματος προς τον αντίστοιχο πάροχο.

Η διαδικασία μετατροπής ενός ερωτήματος φυσικής γλώσσας σε SQL ακολουθεί τα παρακάτω βήματα:

1. Ο χρήστης εισάγει το ερώτημά του σε φυσική γλώσσα μέσω της διεπαφής της εφαρμογής.
2. Το backend ανακτά το σχήμα της επιλεγμένης βάσης δεδομένων (Create statements και ενδεικτικές εγγραφές) και δημιουργεί το κατάλληλο prompt.
3. Το prompt, το οποίο περιλαμβάνει το ερώτημα του χρήστη, το σχήμα της βάσης και οδηγίες για την παραγωγή έγκυρης SQL, αποστέλλεται μέσω του αντίστοιχου API στο επιλεγμένο LLM.
4. Το LLM επιστρέφει το παραγόμενο SQL ερώτημα, το οποίο εμφανίζεται στον χρήστη.
5. Ο χρήστης έχει τη δυνατότητα να ελέγξει ή να τροποποιήσει το παραγόμενο SQL πριν από την εκτέλεσή του.
6. Μετά την επιβεβαίωση, το SQL ερώτημα αποστέλλεται στη βάση δεδομένων και τα αποτελέσματα επιστρέφονται στη διεπαφή της εφαρμογής.

Η συγκεκριμένη προσέγγιση παρέχει ευελιξία, καθώς ο χρήστης μπορεί να επιλέξει διαφορετικό γλωσσικό μοντέλο ανάλογα με τις ανάγκες του, χωρίς να επηρεάζεται η συνολική λειτουργία της εφαρμογής. Παράλληλα, η εκτέλεση του παραγόμενου SQL πραγματοποιείται μόνο μετά από επιβεβαίωση του χρήστη, γεγονός που προσφέρει μεγαλύτερο έλεγχο και μειώνει την πιθανότητα εκτέλεσης λανθασμένων ή ανεπιθύμητων ερωτημάτων. Παρακάτω παρουσιάζεται το prompt που στέλνεται στο LLM .

Το prompt που αποστέλλεται στο LLM

```

$prompt = "

You are a SQL generator.

STRICT RULES:
- Return ONLY ONE SQL query
- No explanations
- No comments
- No multiple statements
- No markdown

Database schema: $ddl

User request: $question
";

```

Η επικοινωνία της εφαρμογής με τα APIs των γλωσσικών μοντέλων πραγματοποιείται μέσω του endpoint `/v1/chat/completions`. Η αυθεντικοποίηση βασίζεται σε προσωπικά API Keys, τα οποία αποθηκεύονται στο αρχείο `.env`, ώστε να μην είναι ενσωματωμένα στον πηγαίο κώδικα της εφαρμογής. Για τη δημιουργία των SQL ερωτημάτων χρησιμοποιείται το μοντέλο `gpt-4o-mini` της OpenAI, ενώ η εφαρμογή υποστηρίζει και το μοντέλο `deepseek-chat` του παρόχου DeepSeek. Η επιλογή του παρόχου, του αντίστοιχου μοντέλου και του endpoint επικοινωνίας πραγματοποιείται δυναμικά πριν από την αποστολή του αιτήματος, επιτρέποντας την εναλλαγή μεταξύ διαφορετικών LLM χωρίς να απαιτούνται αλλαγές στον υπόλοιπο κώδικα της εφαρμογής.

Στη συνέχεια δημιουργείται το payload που αποστέλλεται στο API του επιλεγμένου παρόχου. Το αίτημα περιλαμβάνει ένα μήνυμα τύπου `system`, το οποίο καθορίζει ότι το μοντέλο πρέπει να επιστρέφει αποκλειστικά SQL ερωτήματα, καθώς και ένα μήνυμα τύπου `user`, το οποίο περιέχει το prompt που δημιουργείται δυναμικά από την εφαρμογή με βάση το σχήμα της βάσης δεδομένων και το ερώτημα του χρήστη. Επιπλέον, η παράμετρος `temperature` ορίζεται ίση με μηδέν, ώστε να επιτυγχάνεται μεγαλύτερη συνέπεια και επαναληψιμότητα στα παραγόμενα αποτελέσματα.

Παρακάτω φαίνεται η δυναμική επιλογή του παρόχου, του μοντέλου και του endpoint επικοινωνίας. Επιπλέον, παρουσιάζεται το payload που αποστέλλεται στο API του επιλεγμένου γλωσσικού μοντέλου.

Δυναμική επιλογή παρόχου και μοντέλου LLM

```

// SELECT PROVIDER
switch ($provider) {

    case "deepseek":

        $apiKey = $_ENV['DEEPSEEK_API_KEY'] ?? '';
        $url = "https://api.deepseek.com/chat/completions";
        $model = "deepseek-chat";

```

```

        break;

    case "openai":
    default:

        $apiKey = $_ENV['OPENAI_API_KEY'] ?? '';
        $url = "https://api.openai.com/v1/chat/completions";
        $model = "gpt-4o-mini";
        break;
}

```

Payload που αποστέλλεται στο API του LLM

```

// PAYLOAD
$payload = [

    "model" => $model,
    "messages" => [

        [
            "role" => "system",
            "content" => "You generate SQL only."
        ],

        [
            "role" => "user",
            "content" => $prompt
        ]
    ],

    "temperature" => 0
];

```

5.4.4 Διαχείριση βάσεων δεδομένων

Σύνδεση με την βάση δεδομένων

Η σύνδεση με το επιλεγμένο σύστημα διαχείρισης βάσεων δεδομένων πραγματοποιείται μέσω της βιβλιοθήκης PDO (PHP Data Objects), η οποία παρέχει ένα ενιαίο προγραμματιστικό περιβάλλον για διαφορετικά ΣΔΒΔ. Ανάλογα με τον τύπο της βάσης (MySQL ή PostgreSQL), δημιουργείται δυναμικά το κατάλληλο DSN (Data Source Name) και στη συνέχεια δημιουργείται το αντικείμενο PDO. Επιπλέον, ενεργοποιείται η λειτουργία PDO::ERRMODE_EXCEPTION, ώστε τυχόν σφάλματα σύνδεσης ή εκτέλεσης ερωτημάτων να αντιμετωπίζονται μέσω εξαιρέσεων (exceptions), βελτιώνοντας την αξιοπιστία και τον χειρισμό σφαλμάτων της εφαρμογής. Παρακάτω φαίνεται ο κώδικας για να γίνει πιο κατανοητή η διαδικασία.

Κώδικας για την σύνδεση με την βάση δεδομένων

```
// CONNECT PDO
try {
    if ($type === 'mysql') {
        // If database is empty, connect without specifying a database
        if (!empty($database)) {

            $dsn = "mysql:host=$host;port=$port;dbname=$database;charset=utf8mb4";

        } else {

            $dsn = "mysql:host=$host;port=$port;charset=utf8mb4";
        }
    } else {

        if (empty($database)) {

            echo json_encode([
                "error" => "Database name is required for PostgreSQL"
            ]);

            exit;
        }

        $dsn = "pgsql:host=$host;port=$port;dbname=$database;options='-
-client_encoding=UTF8'";
    }

    $pdo = new PDO(
        $dsn,
        $username,
        $password,
        [
            PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION
        ]
    );
} catch (Exception $e) {

    echo json_encode([
        "error" => "Connection failed"
    ]);

    exit;
}
```

Δημιουργία ssh tunnel

Όταν ο χρήστης επιλέγει σύνδεση μέσω SSH, η εφαρμογή δημιουργεί ένα τοπικό SSH tunnel προς τον απομακρυσμένο διακομιστή. Η προώθηση θυρών (port forwarding) πραγματοποιείται με την παράμετρο -L, η οποία αντιστοιχίζει μια τυχαία τοπική θύρα στη θύρα της απομακρυσμένης βάσης δεδομένων. Με τον τρόπο αυτό, η εφαρμογή επικοινωνεί με τη βάση μέσω του τοπικού υπολογιστή (127.0.0.1), ενώ η μεταφορά των δεδομένων πραγματοποιείται κρυπτογραφημένα μέσω SSH. Παρακάτω φαίνεται ο κώδικας για την δημιουργία ssh tunnel.

Κώδικας για τη δημιουργία ssh tunnel

```
// SSH TUNNEL
if ((int)$useSSH === 1) {

    $localPort = rand(20000, 65000);

    $remoteDbPort =
        ($type === 'mysql')
        ? 3306
        : 5432;

    $ssh_host = escapeshellarg($ssh_host_input);
    $ssh_user = escapeshellarg($ssh_user_input);
    $escaped_host = escapeshellarg($host);

    if (!empty($ssh_pass_input)) {

        $ssh_pass = escapeshellarg($ssh_pass_input);
        $cmd = "sshpass -p $ssh_pass ssh "
            . "-o UserKnownHostsFile=/dev/null "
            . "-o StrictHostKeyChecking=no "
            . "-o ExitOnForwardFailure=yes "
            . "-o LogLevel=ERROR "
            . "-f -N "
            . "-L {$localPort}:{$escaped_host}:{$remoteDbPort} "
            . "-p {$ssh_port_input} {$ssh_user}@{$ssh_host}";

    } else {

        $cmd = "ssh "
            . "-o UserKnownHostsFile=/dev/null "
            . "-o StrictHostKeyChecking=no "
            . "-o ExitOnForwardFailure=yes "
            . "-o LogLevel=ERROR "
            . "-f -N "
            . "-L {$localPort}:{$escaped_host}:{$remoteDbPort} "
            . "-p {$ssh_port_input} {$ssh_user}@{$ssh_host}";

    }

}
```

```

}

$homeDir = sys_get_temp_dir();

exec(
    "HOME=" . escapeshellarg($homeDir) . " " . $cmd . " 2>&1",
    $output,
    $result
);

```

Εξαγωγή του σχήματος της βάσης δεδομένων

Για την εξαγωγή του σχήματος της βάσης κρατάμε όλες τις create statment της βάσης ώστε να έχουμε το πλήρες σχήμα και επιπλέον 1 εγγραφή με δεδομένα για κάθε πίνακα. Τα δεδομένα αυτά χρησιμοποιούνται αποκλειστικά ως δείγμα της μορφής και του τύπου των αποθηκευμένων πληροφοριών και ενσωματώνονται στο prompt που αποστέλλεται στο γλωσσικό μοντέλο. Με τον τρόπο αυτό το μοντέλο αποκτά καλύτερη κατανόηση της δομής και του περιεχομένου της βάσης, γεγονός που συμβάλλει στη δημιουργία ακριβέστερων SQL ερωτημάτων. Μετά την επιτυχή σύνδεση με τη βάση δεδομένων, η εφαρμογή προχωρά στην αυτόματη εξαγωγή του σχήματος (schema). Για τη MySQL αξιοποιείται το εργαλείο mysqldump με την παράμετρο --no-data, ενώ για την PostgreSQL χρησιμοποιείται το pg_dump με την παράμετρο --schema-only. Με τον τρόπο αυτό εξάγονται αποκλειστικά οι ορισμοί των πινάκων, των στηλών και των περιορισμών, χωρίς τα πραγματικά δεδομένα της βάσης. Παρακάτω φαίνεται ο κώδικας για την εξαγωγή του σχήματος σε mysql και PostgreSQL.

Κώδικας για την εξαγωγή του σχήματος σε mySQL

```

$dumpCmd =
    "mysqldump " .
    "-h $escapedHost " .
    "-P $escapedPort " .
    "-u $escapedUser " .
    "$escapedDb " .
    "--no-data " .
    "--single-transaction " .
    "--skip-lock-tables " .
    "--force " .
    "--no-tablespaces " .
    "--skip-comments " .
    "2>&1";

$dumpOutput = shell_exec($dumpCmd);

```

Κώδικας για την εξαγωγή του σχήματος σε PostgreSQL

```

$dumpCmd =
    "pg_dump " .

```

```

"-h $escapedHost " .
"-p $escapedPort " .
"-U $escapedUser " .
"-d $escapedDb " .
"--schema-only " .
"--no-owner " .
"--no-privileges " .
"--no-comments " .
"2>&1";

```

```
$dumpOutput = shell_exec($dumpCmd);
```

Έλεγχος και ασφαλής εκτέλεση SQL ερωτημάτων

Για την προστασία των δεδομένων, η εφαρμογή εφαρμόζει μηχανισμό ελέγχου των SQL ερωτημάτων πριν από την εκτέλεσή τους. Ο μηχανισμός αυτός λειτουργεί τόσο σε επίπεδο διεπαφής χρήστη όσο και στο backend, ώστε ο έλεγχος ασφαλείας να μην βασίζεται αποκλειστικά στο frontend.

Αρχικά, πραγματοποιείται έλεγχος των δικαιωμάτων του χρήστη βάσει του ρόλου του. Οι χρήστες με ρόλο *demo* έχουν δικαίωμα εκτέλεσης αποκλειστικά ερωτημάτων τύπου SELECT, ενώ εντολές που μπορούν να τροποποιήσουν τη βάση δεδομένων απορρίπτονται. Αντίθετα, οι χρήστες με ρόλο *full* ή *admin* μπορούν να εκτελούν και ερωτήματα τροποποίησης δεδομένων.

Επιπλέον, οι εντολές που ενδέχεται να επιφέρουν αλλαγές στη βάση δεδομένων, όπως DROP, DELETE, TRUNCATE, ALTER, UPDATE και INSERT, χαρακτηρίζονται ως επικίνδυνες. Για την εκτέλεσή τους απαιτείται προηγούμενη επιβεβαίωση από τον χρήστη. Σε περίπτωση που δεν έχει δοθεί επιβεβαίωση, εμφανίζεται σχετικό μήνυμα επιβεβαίωσης στη διεπαφή της εφαρμογής.

Με αυτόν τον τρόπο μειώνεται ο κίνδυνος εκτέλεσης ενεργειών που μπορεί να τροποποιήσουν ή να διαγράψουν δεδομένα, ενώ παράλληλα εφαρμόζεται έλεγχος πρόσβασης ανάλογα με τον ρόλο του χρήστη.

5.5 Υλοποίηση Frontend

Όλες οι λειτουργίες του AskSQL είναι προσβάσιμες μέσω μίας εύχρηστης γραφικής διεπαφής χρήστη. Το frontend επικοινωνεί με το backend μέσω τεχνολογιών ιστού και είναι υπεύθυνο για την αλληλεπίδραση του χρήστη με την εφαρμογή. Η διεπαφή χρήστη αναπτύχθηκε με τις τεχνολογίες HTML5, CSS3, JavaScript και το πλαίσιο Bootstrap 5.

Συγκεκριμένα, η γλώσσα HTML χρησιμοποιήθηκε για τη δομή και οργάνωση όλων των σελίδων της εφαρμογής. Στο Σχήμα 5.4, απεικονίζονται τα HTML αρχεία που αντιπροσωπεύουν τις σελίδες της εφαρμογής. Τα αρχεία αυτά βρίσκονται στον φάκελο 'html', ενώ το αρχείο 'index.php' βρίσκεται στον κεντρικό κατάλογο, καθώς αντιπροσωπεύει την αρχική σελίδα της εφαρμογής, η οποία φορτώνεται αυτόματα όταν ο χρήστης επισκέπτεται τον σχετικό σύνδεσμο. Αναλυτικότερα, τα αρχεία *login.html*, *register.html* υλοποιούν τις σελίδες σύνδεσης και εγγραφής των χρηστών αντίστοιχα και περιέχουν την κατάλληλη φόρμα για την συμπλήρωση των στοιχείων του χρήστη. Οι σελίδες *verify.html* και *google_success.html* χρησιμοποιούνται για την εμφάνιση της επιβεβαίωσης των λογαριασμών των χρηστών μετά την εγγραφή είτε με κανο-

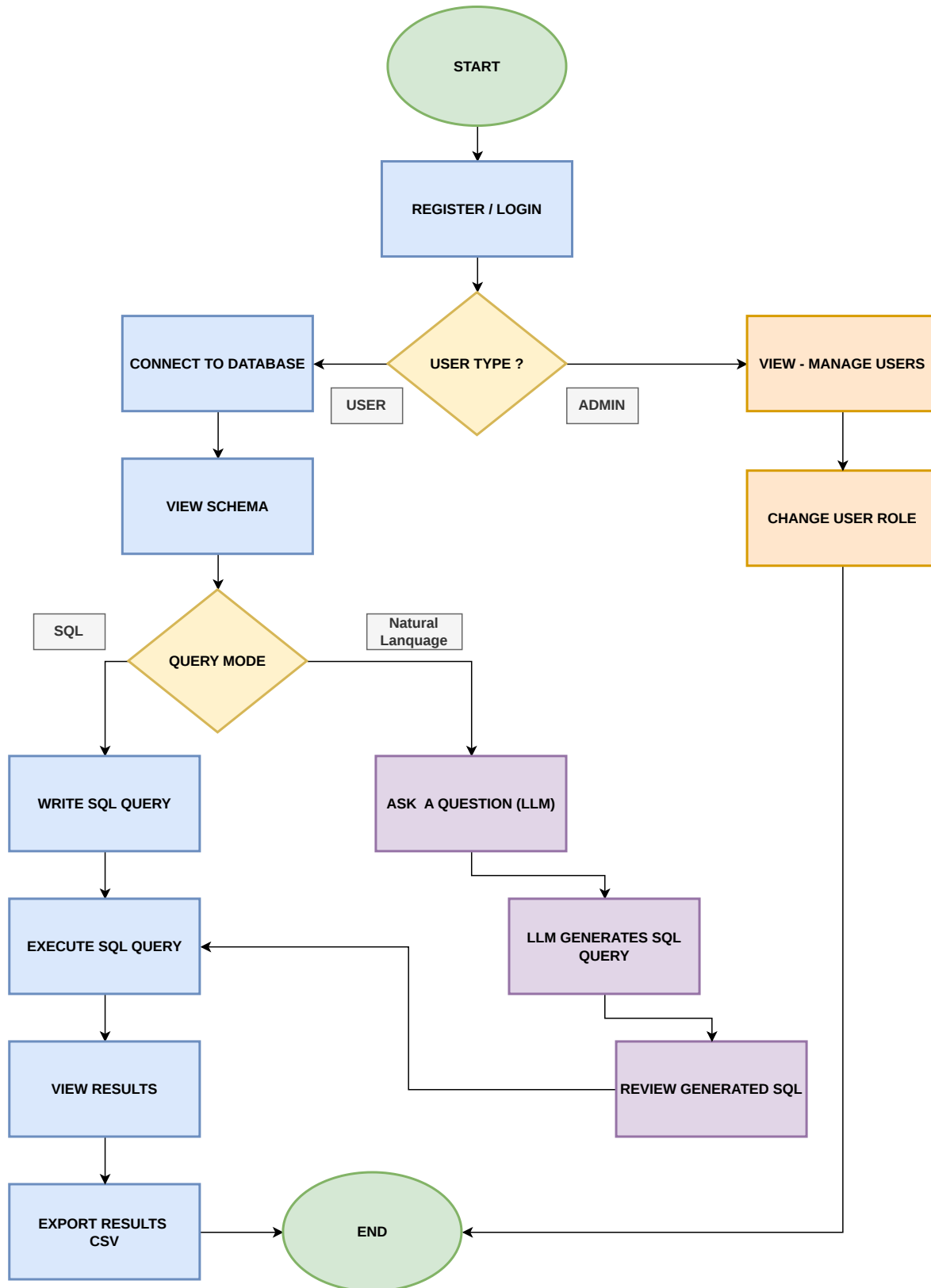
νική εγγραφή είτε μέσω Google account . Οι σελίδες *forgot.html* και *password_reset.html* εξυπηρετούν περιπτώσεις όπου ένας χρήστης έχει ξεχάσει τον κωδικό πρόσβασης του. Η σελίδα *forgot.html* παρέχει τη φόρμα για την εισαγωγή του email του χρήστη, στο οποίο θα του αποσταλεί ο σύνδεσμος επαναφοράς κωδικού, ενώ η σελίδα με το αρχείο *password_reset.html* παρέχει τη φόρμα για την εισαγωγή του νέου κωδικού. Το αρχείο *connections.html* σχεδιάστηκε για να παρουσιάζει την δομή της σελίδας στην οποία εμφανίζονται οι αποθηκευμένες συνδέσεις του χρήστη, μέσω της οποίας ο χρήστης μπορεί να πραγματοποιήσει επανασύνδεση, μετονομασία και διαγραφή των αποθηκευμένων συνδέσεων του. Μέσω της σελίδας *new_connection.html* ο χρήστης μπορεί να δημιουργήσει μια νέα σύνδεση σε βάση δεδομένων επιλέγοντας τον τρόπο σύνδεσης (TCP/IP, ssh tunnel) και συμπληρώνοντας τα κατάλληλα πεδία στην φόρμα. Το αρχείο *dashboard.html* περιέχει την δομή για την κεντρική σελίδα της εφαρμογής. Η σελίδα αυτή είναι προσβάσιμη μόνο σε χρήστες που έχουν συνδεθεί επιτυχώς με μία βάση δεδομένων. Η σελίδα αυτή επιτρέπει στους χρήστες την εξερεύνηση του σχήματος της βάσης, την εκτέλεση ερωτημάτων (με SQL ή φυσική γλώσσα), την οπτικοποίηση των αποτελεσμάτων και την αποθήκευσή τους. Οι σελίδες *about.html* *profile.html* είναι βοηθητικές και είναι προσβάσιμες από το κεντρικό μενού της εφαρμογής. Η σελίδα *about.html* περιέχει πληροφορίες σχετικά με την εφαρμογή. Η σελίδα *profile.html* σχεδιάστηκε για να περιέχει την δομή της σελίδας για την διαχείριση του λογαριασμού του χρήστη. Περιέχει τις πληροφορίες του χρήστη και μέσω αυτής υπάρχει επιλογή αλλαγής κωδικού πρόσβασης. Τέλος, η σελίδα *admin.html* είναι προσβάσιμη μόνο από τους χρήστες Admin της εφαρμογής. Στην σελίδα αυτή εμφανίζονται πληροφορίες για όλους τους χρήστες, ενώ παρέχεται η δυνατότητα τροποποίησης των δικαιωμάτων πρόσβασης των χρηστών.

Με τη γλώσσα CSS καθορίστηκε η εμφάνιση των σελίδων της εφαρμογής. Επιπλέον, χρησιμοποιήθηκε το πλαίσιο Bootstrap 5 για την ανάπτυξη μιας σύγχρονης και αισθητικά βελτιωμένης διεπαφής χρήστη, καθώς και για την υλοποίηση προσαρμοστικής σχεδίασης (responsive design), ώστε η εφαρμογή να προσαρμόζεται σε διαφορετικούς τύπους και μεγέθη συσκευών. Όπως φαίνεται στο Σχήμα 5.5, τα αρχεία CSS είναι αποθηκευμένα στον κατάλογο css της εφαρμογής. Τα αρχεία αυτά χρησιμοποιήθηκαν για τη μορφοποίηση των στοιχείων HTML, όπως οι φόρμες εισαγωγής, τα κουμπιά, οι σύνδεσμοι και τα μενού πλοήγησης. Επιπλέον, αξιοποιήθηκε η βιβλιοθήκη Bootstrap Icons για την ενσωμάτωση εικονιδίων, με στόχο τη βελτίωση της αισθητικής και της ευχρηστίας της εφαρμογής.

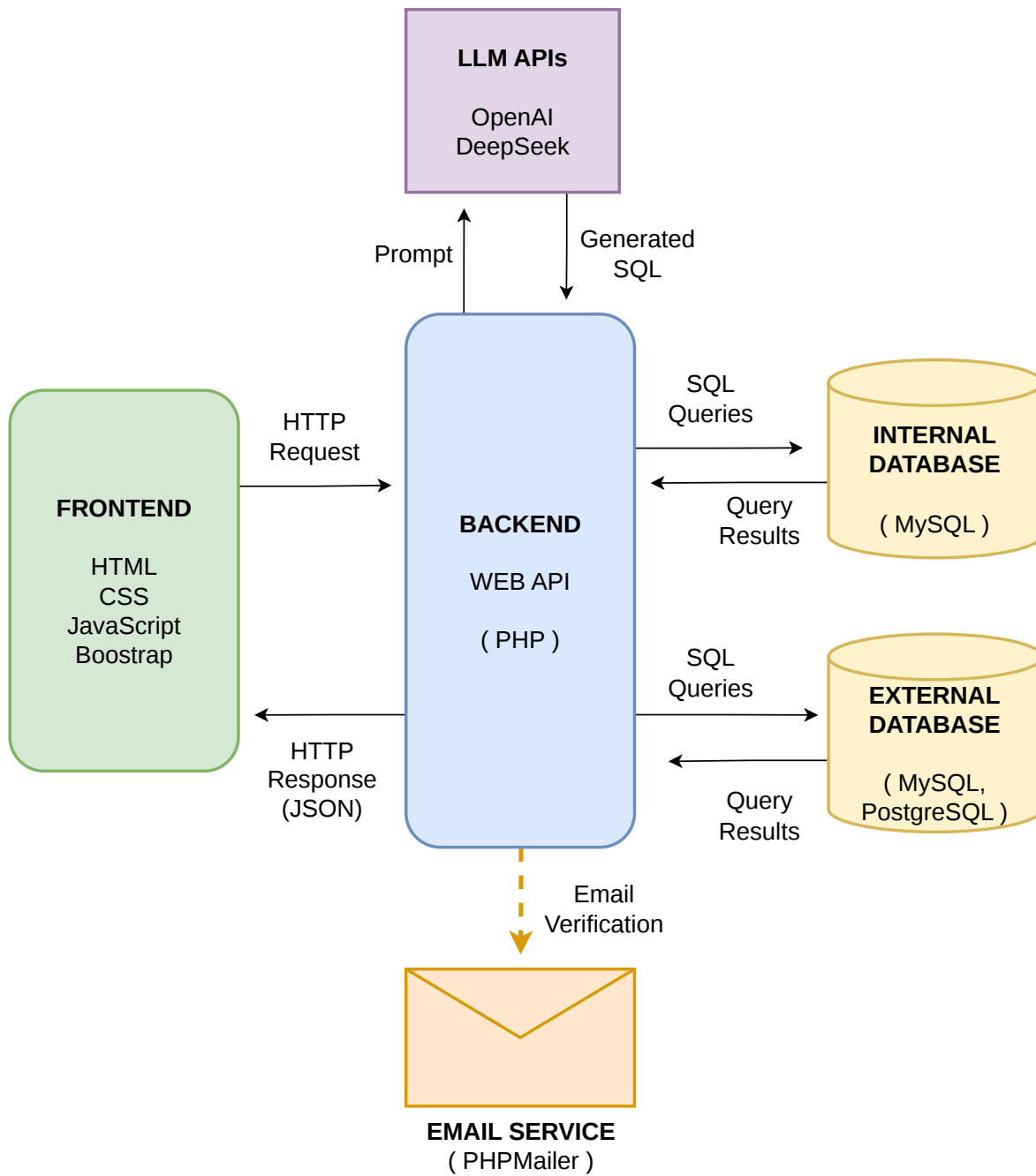
Με τη γλώσσα προγραμματισμού JavaScript υλοποιήθηκε η λειτουργικότητα της εφαρμογής από την πλευρά του πελάτη (client-side). Η επικοινωνία με το Web API πραγματοποιείται μέσω ασύγχρονων HTTP αιτημάτων, με τη χρήση της συνάρτησης *fetch()*, η οποία παρέχει έναν σύγχρονο και απλό τρόπο αποστολής και διαχείρισης αιτημάτων προς τα διαθέσιμα endpoints του API. Με αυτόν τον τρόπο, τα δεδομένα μπορούν να ανταλλάσσονται δυναμικά μεταξύ frontend και backend χωρίς να απαιτείται η επαναφόρτωση ολόκληρης της σελίδας, προσφέροντας πιο άμεση και ομαλή εμπειρία χρήσης. Στο Σχήμα 5.6 παρουσιάζονται τα αρχεία JavaScript που αναπτύχθηκαν για την υλοποίηση της λειτουργικότητας της εφαρμογής. Ακολουθεί επίσης ένα παράδειγμα χρήσης της συνάρτησης *fetch()*, με σκοπό την καλύτερη κατανόηση της επικοινωνίας μεταξύ frontend και backend (Κώδικας ??).

Κάθε σελίδα της εφαρμογής συνοδεύεται από το αντίστοιχο αρχείο JavaScript, το οποίο υλοποιεί τη λειτουργικότητα που απαιτείται για τη συγκεκριμένη σελίδα. Για τον σκοπό αυτό αναπτύχθηκαν αρχεία όπως τα *login.js*, *register.js*, *dashboard.js*, *connections.js*, *profile.js* κ.ά. Ιδιαίτερη σημασία έχει το αρχείο *auth.js*, το οποίο χρησιμοποιείται από όλες τις σελίδες της εφαρμογής

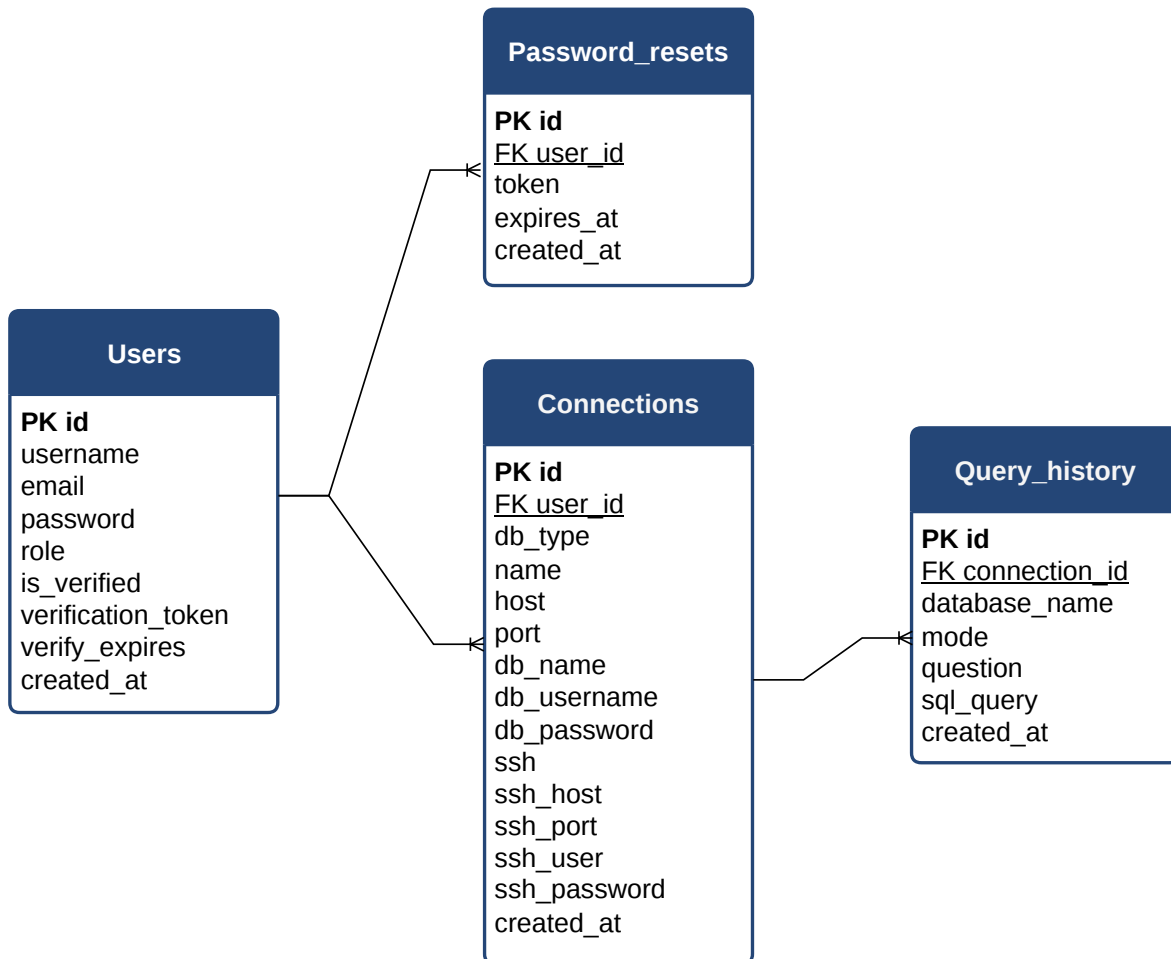
και είναι υπεύθυνο για τη διαχείριση της αυθεντικοποίησης του χρήστη. Συγκεκριμένα, ελέγχει την ύπαρξη του JWT που αποθηκεύεται στον φυλλομετρητή μετά την επιτυχή σύνδεση και επιτρέπει την πρόσβαση μόνο σε εξουσιοδοτημένους χρήστες. Παράλληλα, διαχειρίζεται το αναγνωριστικό της ενεργής σύνδεσης με τη βάση δεδομένων (`connection_id`), το οποίο χρησιμοποιείται για τον έλεγχο πρόσβασης στις λειτουργίες που απαιτούν ενεργή σύνδεση με μία βάση δεδομένων. Επιπλέον, παρέχει τη συνάρτηση *apiFetch()*, η οποία χρησιμοποιείται για την επικοινωνία με το REST API, προσθέτοντας αυτόματα το JWT στην επικεφαλίδα Authorization κάθε αιτήματος και διαχειριζόμενη περιπτώσεις μη εξουσιοδοτημένης πρόσβασης (HTTP 401).



Σχήμα 5.1: Διάγραμμα ροής του AskSQL.



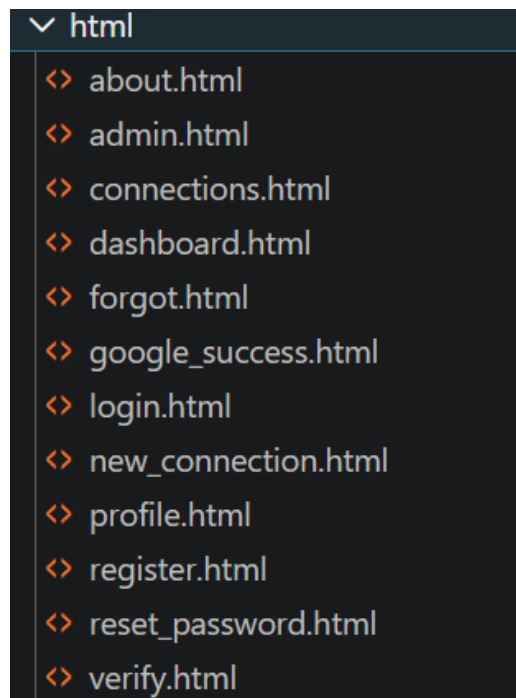
Σχήμα 5.2: Αρχιτεκτονική του AskSQL.



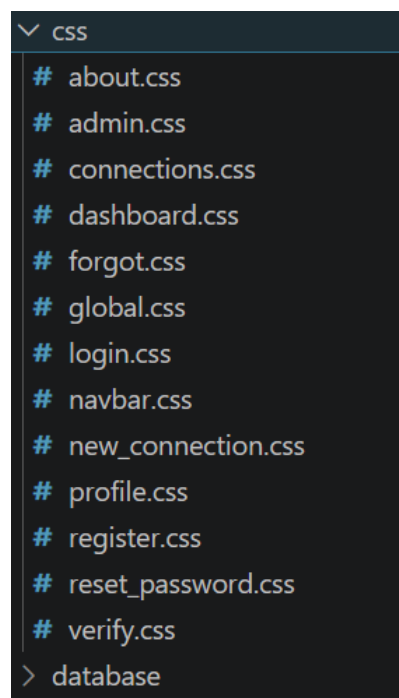
Σχήμα 5.3: Entity-Relationship (ER) διάγραμμα της βάσης δεδομένων του AskSQL.

Πίνακας 5.6: Protected API Endpoints της εφαρμογής

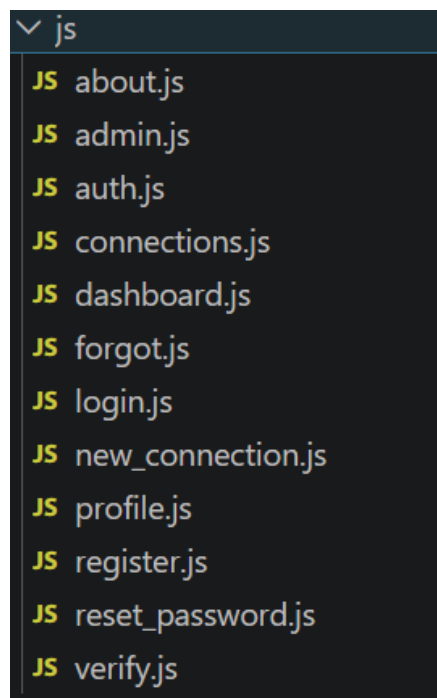
A/A	ΜΕΘΟΔΟΣ HTTP	ENDPOINT	ΛΕΙΤΟΥΡΓΙΑ
1	GET	me.php	Επιστροφή στοιχείων τρέχοντος χρήστη
2	POST	profile.php	Ενημέρωση στοιχείων προφίλ
3	POST	change_password.php	Αλλαγή κωδικού πρόσβασης
4	GET	get_connections.php	Ανάκτηση αποθηκευμένων συνδέσεων
5	POST	connect.php	Δημιουργία νέας σύνδεσης βάσης δεδομένων
6	POST	rename_connection.php	Μετονομασία σύνδεσης
7	POST	delete_connection.php	Διαγραφή σύνδεσης
8	GET	get_databases.php	Ανάκτηση διαθέσιμων βάσεων δεδομένων
9	GET	get_tables.php	Ανάκτηση πινάκων βάσης δεδομένων
10	GET	get_columns.php	Ανάκτηση στηλών πίνακα
11	GET	get_schema.php	Ανάκτηση σχήματος βάσης δεδομένων
12	POST	nl2sql.php	Μετατροπή ερωτήματος φυσικής γλώσσας σε SQL μέσω LLM
13	POST	execute_query.php	Εκτέλεση SQL ερωτήματος στη βάση δεδομένων
14	POST	save_query.php	Αποθήκευση ιστορικού ερωτημάτων
15	GET	get_history.php	Ανάκτηση ιστορικού ερωτημάτων
16	POST	delete_history.php	Διαγραφή ιστορικού ερωτημάτων
17	GET	export.php	Εξαγωγή αποτελεσμάτων ερωτήματος
18	GET	check_cache.php	Έλεγχος προσωρινής αποθήκευσης σχήματος
19	GET	get_users.php	Ανάκτηση λίστας χρηστών από τον Admin
20	POST	update_role.php	Τροποποίηση ρόλου χρήστη από τον Admin



Σχήμα 5.4: Αρχεία HTML Κώδικα για τις Σελίδες της Εφαρμογής



Σχήμα 5.5: Αρχεία CSS Κώδικα για τις Σελίδες της Εφαρμογής



Σχήμα 5.6: Αρχεία JavaScript Κώδικα για τις Σελίδες της Εφαρμογής

Κεφάλαιο 6

Παρουσίαση του AskSQL

Η λειτουργική έκδοση της εφαρμογής AskSQL έχει αναρτηθεί σε διακομιστή του Διεθνούς Πανεπιστημίου της Ελλάδος και είναι διαθέσιμη μέσω διαδικτύου. Ο χρήστης μπορεί να αποκτήσει πρόσβαση στην εφαρμογή μέσω του ακόλουθου συνδέσμου: <https://nireas.iee.ihu.gr/asksql>

Με την πρόσβαση στην εφαρμογή, ο χρήστης μεταφέρεται στη σελίδα σύνδεσης, η οποία αποτελεί το αρχικό σημείο αλληλεπίδρασης με το σύστημα.

6.1 Σύνδεση

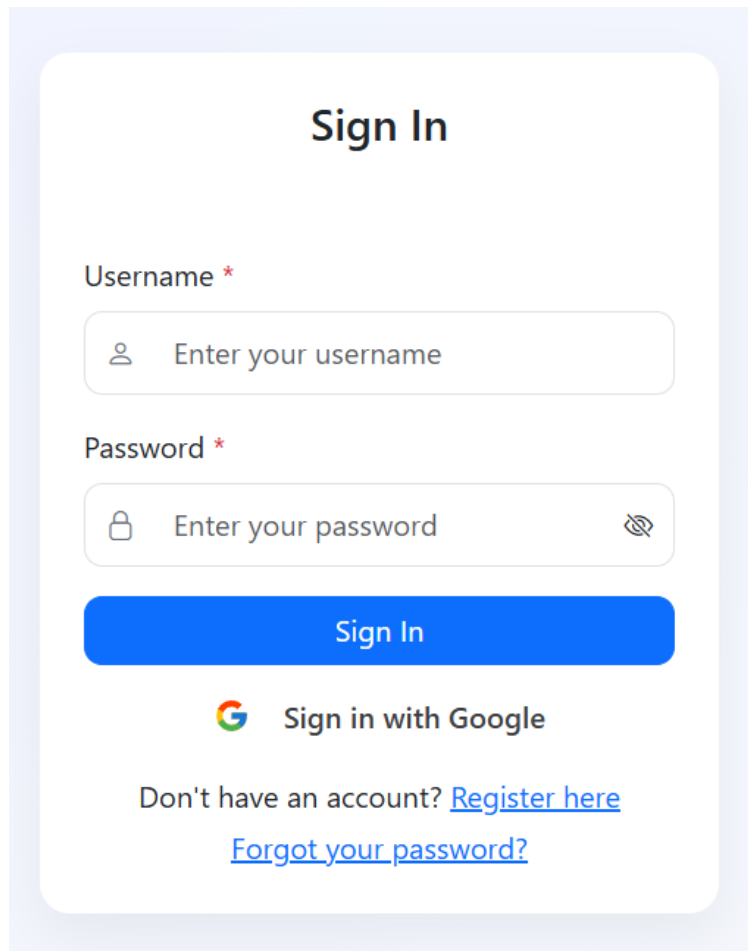
Με την εκκίνηση της εφαρμογής, ο χρήστης μεταφέρεται στη σελίδα σύνδεσης. Όπως φαίνεται στο Σχήμα 6.1, η σελίδα περιλαμβάνει μία φόρμα στην οποία ο χρήστης καλείται να εισαγάγει το όνομα χρήστη και τον κωδικό πρόσβασής του, προκειμένου να συνδεθεί στην εφαρμογή. Επιπλέον, παρέχεται η δυνατότητα σύνδεσης μέσω λογαριασμού Google για ταχύτερη και ευκολότερη πρόσβαση (Σχήμα 6.2). Τέλος, υπάρχουν επιλογές ανακατεύθυνσης προς τη σελίδα εγγραφής για νέους χρήστες, καθώς και προς τη διαδικασία επαναφοράς κωδικού πρόσβασης σε περίπτωση που ο χρήστης θέλει να τον επαναφέρει.

6.2 Εγγραφή

Στη σελίδα εγγραφής (Σχήμα 6.3) ο χρήστης μπορεί να δημιουργήσει νέο λογαριασμό στην εφαρμογή. Για τον σκοπό αυτό συμπληρώνει τα απαιτούμενα στοιχεία στη σχετική φόρμα, όπως το όνομα χρήστη, το email και τον κωδικό πρόσβασης. Με την ολοκλήρωση της εγγραφής αποστέλλεται μήνυμα ηλεκτρονικού ταχυδρομείου που περιέχει σύνδεσμο επιβεβαίωσης του λογαριασμού. Μετά την επιτυχή επιβεβαίωση, ο χρήστης μπορεί να συνδεθεί στην εφαρμογή.

6.3 Ανάκτηση κωδικού

Η εφαρμογή παρέχει τη δυνατότητα επαναφοράς του κωδικού πρόσβασης σε περίπτωση που ο χρήστης αδυνατεί να συνδεθεί. Ο χρήστης μεταβαίνει στη σελίδα ανάκτησης κωδικού (Σχήμα 6.4), όπου εισάγει το email που αντιστοιχεί στον λογαριασμό του. Εφόσον το email

A screenshot of a 'Sign In' form. The form is white with rounded corners and is centered on a light blue background. It features a title 'Sign In' at the top. Below the title are two input fields: 'Username *' and 'Password *'. The 'Username' field has a person icon on the left and the placeholder text 'Enter your username'. The 'Password' field has a lock icon on the left, the placeholder text 'Enter your password', and an eye icon on the right to toggle visibility. Below these fields is a blue 'Sign In' button. Under the button is a 'Sign in with Google' link with the Google logo. At the bottom, there are two links: 'Don't have an account? Register here' and 'Forgot your password?'.

Sign In

Username *

Enter your username

Password *

Enter your password

Sign In

Sign in with Google

Don't have an account? [Register here](#)

[Forgot your password?](#)

Σχήμα 6.1: Σελίδα σύνδεσης

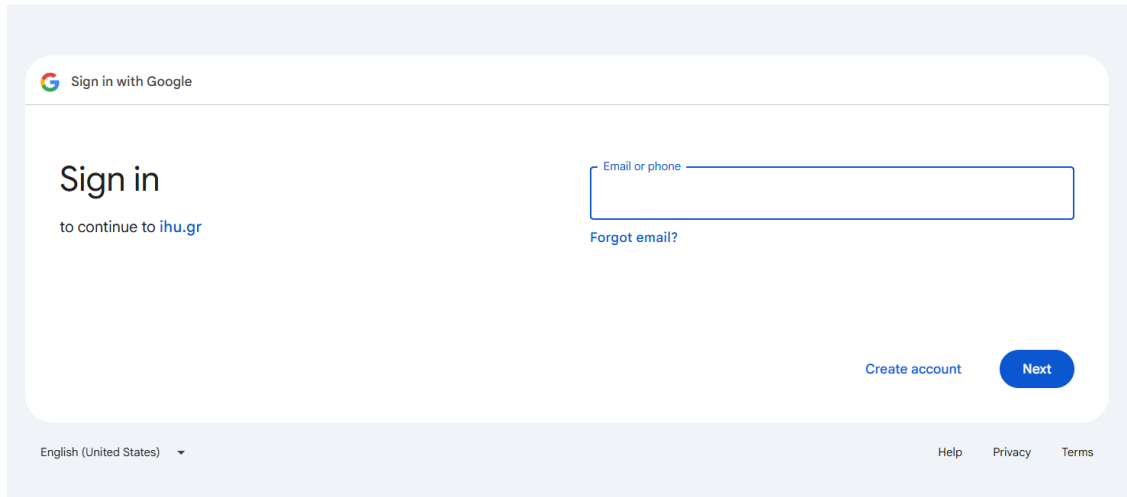
είναι έγκυρο, αποστέλλεται μήνυμα ηλεκτρονικού ταχυδρομείου με σύνδεσμο που οδηγεί στη σελίδα επαναφοράς κωδικού (Σχήμα 6.5). Στη συνέχεια, ο χρήστης εισάγει τον νέο κωδικό πρόσβασης και ολοκληρώνει τη διαδικασία αλλαγής του.

6.4 Διαχείριση συνδέσεων

Στη σελίδα διαχείρισης συνδέσεων (Σχήμα 6.6) ο χρήστης μπορεί να δει και να διαχειριστεί όλες τις αποθηκευμένες συνδέσεις με τις βάσεις δεδομένων του. Παρέχεται η δυνατότητα μετονομασίας και διαγραφής των αποθηκευμένων συνδέσεων, καθώς και επανασύνδεσης σε κάποια από αυτές χωρίς να απαιτείται εκ νέου εισαγωγή των στοιχείων σύνδεσης. Επιπλέον, στο μενού υπάρχει δυνατότητα δημιουργίας νέας σύνδεσης.

6.5 Δημιουργία νέας σύνδεσης

Η σελίδα δημιουργίας νέας σύνδεσης (Σχήμα 6.7) επιτρέπει στον χρήστη να δημιουργήσει μία νέα σύνδεση με βάση δεδομένων. Ο χρήστης συμπληρώνει τα απαραίτητα στοιχεία, όπως το



Σχήμα 6.2: Σύνδεση μέσω λογαριασμού Google

όνομα του διακομιστή (host), τη θύρα (port), το όνομα χρήστη, τον κωδικό πρόσβασης, το όνομα της βάσης δεδομένων και δίνει ένα όνομα για τη συγκεκριμένη σύνδεση. Η εφαρμογή υποστηρίζει συνδέσεις με συστήματα διαχείρισης βάσεων δεδομένων MySQL και PostgreSQL. Επιπλέον, παρέχεται η δυνατότητα σύνδεσης είτε απευθείας μέσω TCP/IP είτε μέσω SSH Tunnel. Στη δεύτερη περίπτωση απαιτείται η συμπλήρωση των επιπλέον στοιχείων σύνδεσης SSH, όπως το SSH host, η θύρα, το όνομα χρήστη και ο κωδικός πρόσβασης SSH.

6.6 Κεντρική σελίδα

Μετά την επιτυχή σύνδεση σε μία βάση δεδομένων, ο χρήστης μεταφέρεται στην κεντρική σελίδα της εφαρμογής, όπου πραγματοποιούνται όλες οι βασικές λειτουργίες του AskSQL. Στο αριστερό τμήμα της σελίδας εμφανίζεται η δομή της επιλεγμένης βάσης δεδομένων, δίνοντας τη δυνατότητα προβολής των πινάκων και των αντίστοιχων στηλών τους.

Στο κεντρικό τμήμα της σελίδας πραγματοποιείται η δημιουργία και εκτέλεση των ερωτημάτων. Ο χρήστης μπορεί να επιλέξει μεταξύ δύο τρόπων λειτουργίας. Στη λειτουργία SQL Mode συντάσσει και εκτελεί χειροκίνητα SQL ερωτήματα (Σχήμα 6.8), ενώ στη λειτουργία Natural Language Mode (NL Mode) μπορεί να διατυπώσει το ερώτημά του σε φυσική γλώσσα (Σχήμα 6.9). Στην περίπτωση αυτή επιλέγει το επιθυμητό γλωσσικό μοντέλο (chatGPT ή DeepSeek), ενώ παρέχεται και δυνατότητα εισαγωγής της ερώτησης μέσω φωνητικής εντολής (σε ελληνικά και αγγλικά). Το SQL ερώτημα που παράγεται από το επιλεγμένο LLM εμφανίζεται στην οθόνη, δίνοντας στον χρήστη τη δυνατότητα να το επεξεργαστεί πριν από την εκτέλεσή του.

Πριν από την εκτέλεση ενός SQL ερωτήματος πραγματοποιείται έλεγχος για τον εντοπισμό εντολών που ενδέχεται να τροποποιήσουν ή να διαγράψουν δεδομένα, όπως οι DELETE, UPDATE, INSERT, DROP, ALTER και TRUNCATE. Για χρήστες με ρόλο demo, η εκτέλεση ερωτημάτων που δεν είναι τύπου SELECT δεν επιτρέπεται. Για χρήστες με αυξημένα δικαιώματα, όταν εντοπίζεται ένα πιθανό επικίνδυνο ερώτημα, εμφανίζεται μήνυμα επιβεβαίωσης πριν από την εκτέλεσή του, όπως παρουσιάζεται στο Σχήμα 6.10.

Στο δεξί τμήμα της σελίδας εμφανίζονται τα αποτελέσματα της εκτέλεσης του ερωτήματος, με χρήση μηχανισμού σελιδοποίησης (pagination), διευκολύνοντας την προβολή μεγάλου όγκου δεδομένων. Στο κάτω μέρος της σελίδας εμφανίζεται το ιστορικό των τελευταίων ερωτημάτων που έχει εκτελέσει ο χρήστης, διαχωρισμένο ανάλογα με τον τρόπο λειτουργίας (SQL Mode ή NL Mode). Μέσω του ιστορικού αυτού ο χρήστης μπορεί να επανεκτελέσει εύκολα προηγούμενα ερωτήματα, χωρίς να απαιτείται η εκ νέου πληκτρολόγησή τους. Επιπλέον, παρέχεται η δυνατότητα διαγραφής του ιστορικού, επιτρέποντας στον χρήστη να διαχειρίζεται τα αποθηκευμένα ερωτήματα σύμφωνα με τις ανάγκες του. Τέλος, ο χρήστης μπορεί να επιλέξει από το μενού την εξαγωγή των αποτελεσμάτων του ερωτήματος σε αρχείο μορφής CSV, ώστε να είναι δυνατή η αποθήκευση και η περαιτέρω επεξεργασία τους με εξωτερικές εφαρμογές.

6.7 Διαχείριση λογαριασμού

Η σελίδα διαχείρισης λογαριασμού είναι προσβάσιμη μέσω του κεντρικού μενού της εφαρμογής. Σε αυτή εμφανίζονται τα βασικά στοιχεία του λογαριασμού, όπως το όνομα χρήστη, η διεύθυνση ηλεκτρονικού ταχυδρομείου, ο ρόλος του χρήστη (ADMIN, DEMO, FULL) και η ημερομηνία δημιουργίας του λογαριασμού. Επιπλέον, παρέχεται η δυνατότητα αλλαγής του κωδικού πρόσβασης (Σχήμα 6.11).

6.8 Σελίδα About

Η σελίδα About (Σχήμα 6.12) είναι προσβάσιμη μέσω του κεντρικού μενού της εφαρμογής και παρέχει συνοπτικές πληροφορίες σχετικά με τον σκοπό ανάπτυξης του AskSQL, καθώς και τις βασικές δυνατότητες και λειτουργίες που προσφέρει στους χρήστες.

6.9 Διαχείριση χρηστών

Η σελίδα αυτή είναι διαθέσιμη μόνο για τους χρήστες της εφαρμογής με δικαιώματα Admin. Σε αυτή οι διαχειριστές μπορούν να δουν όλους τους εγγεγραμμένους χρήστες της εφαρμογής και τις πληροφορίες για το λογαριασμό τους (Σχήμα 6.13). Επίσης, μπορούν να αλλάξουν τα δικαιώματα των χρηστών (demo, full, admin) (Σχήμα 6.14).

Create Account

Username *

Password *

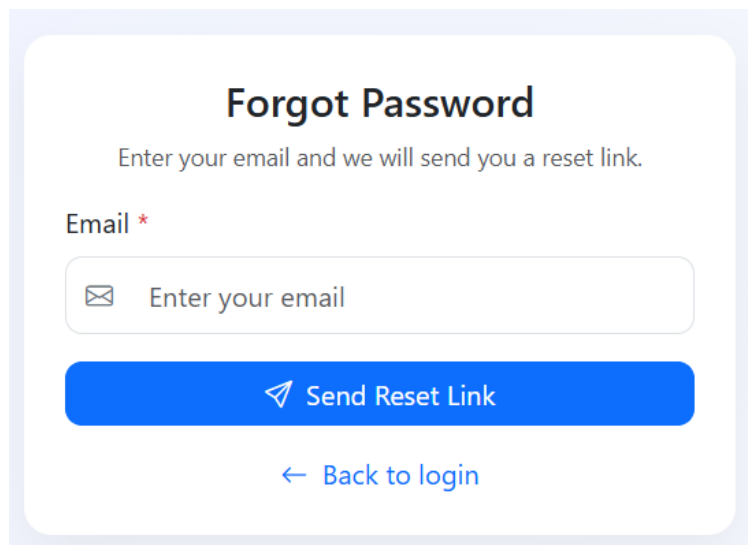
Confirm Password *

Email *

Create Account

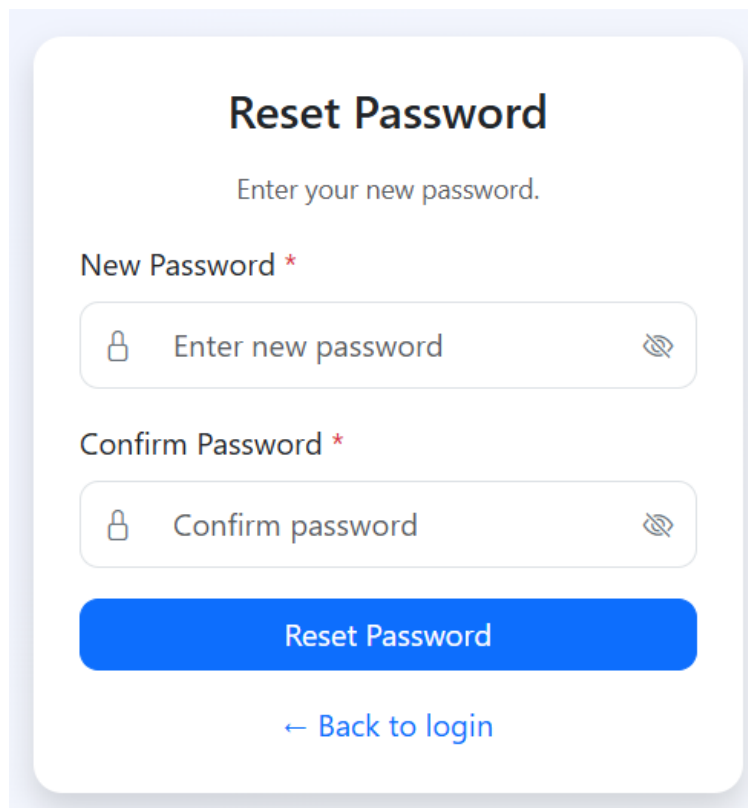
Already have an account? [Sign in](#)

Σχήμα 6.3: Σελίδα εγγραφής



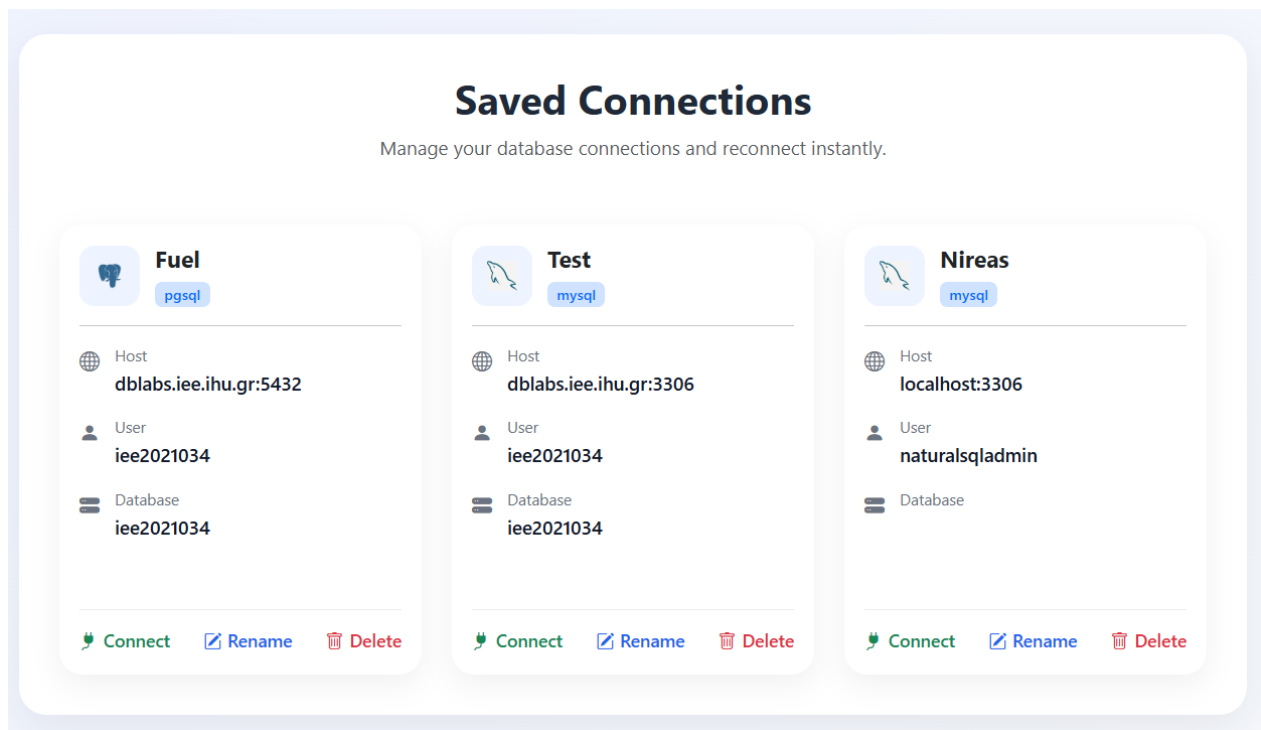
The image shows a 'Forgot Password' form. At the top, the title 'Forgot Password' is centered in a bold, dark font. Below it, a subtitle reads 'Enter your email and we will send you a reset link.' The form contains a single input field labeled 'Email *' with a red asterisk indicating a required field. The input field has a light gray border and a placeholder text 'Enter your email' preceded by an envelope icon. Below the input field is a prominent blue button with white text that says 'Send Reset Link' and a paper plane icon. At the bottom of the form, there is a blue link with a left-pointing arrow and the text 'Back to login'.

Σχήμα 6.4: Σελίδα forgot



The image shows a 'Reset Password' form. At the top, the title 'Reset Password' is centered in a bold, dark font. Below it, a subtitle reads 'Enter your new password.' The form contains two input fields. The first is labeled 'New Password *' with a red asterisk. It has a light gray border, a placeholder text 'Enter new password' preceded by a lock icon, and a toggle icon (an eye with a slash) on the right. The second input field is labeled 'Confirm Password *' with a red asterisk. It also has a light gray border, a placeholder text 'Confirm password' preceded by a lock icon, and a toggle icon on the right. Below these two fields is a prominent blue button with white text that says 'Reset Password'. At the bottom of the form, there is a blue link with a left-pointing arrow and the text 'Back to login'.

Σχήμα 6.5: Σελίδα για επαναφορά κωδικού



Σχήμα 6.6: Σελίδα με τις αποθηκευμένες συνδέσεις

New Database Connection

Create a new connection to your database

Name *

Connection name

Connection Type

MySQL (TCP/IP) ▼

Host *

e.g. localhost

Port *

3306

User *

e.g. root

Password

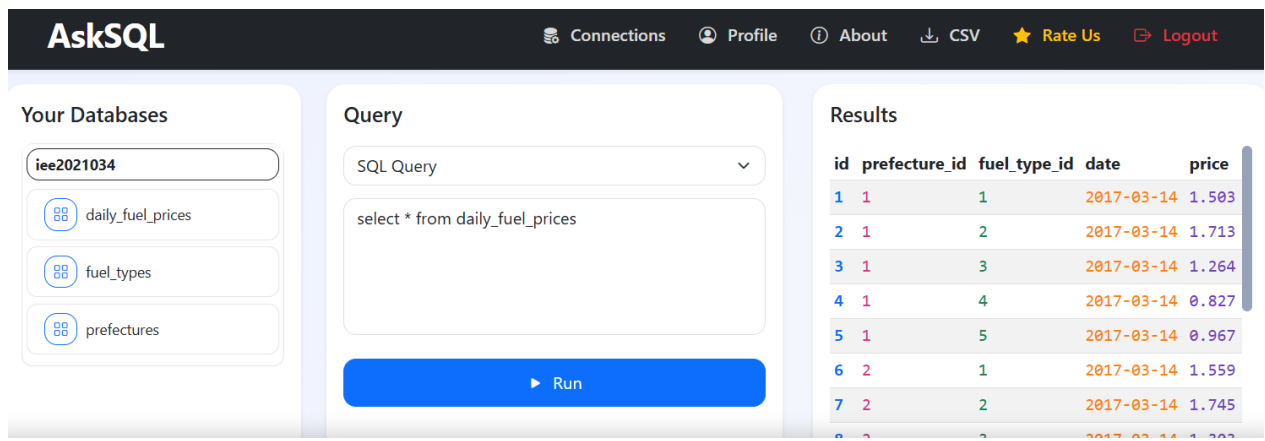
Database password 

Database

Database name

Connect to Database

Σχήμα 6.7: Νέα σύνδεση σε βάση δεδομένων



The screenshot shows the AskSQL web interface. On the left, under 'Your Databases', there is a list of databases: 'iee2021034', 'daily_fuel_prices', 'fuel_types', and 'prefectures'. The 'Query' section shows a dropdown menu set to 'SQL Query' and a text area containing the query 'select * from daily_fuel_prices'. Below the text area is a blue 'Run' button. The 'Results' section displays a table with 5 columns: 'id', 'prefecture_id', 'fuel_type_id', 'date', and 'price'. The table contains 7 rows of data. Below the results, there is a section titled 'Table: daily_fuel_prices' which shows the table's schema with columns: Name, Type, Nullable, Key, Default, and Extra.

Name	Type	Nullable	Key	Default	Extra
id	bigint	NO			
prefecture_id	int	NO			
fuel_type_id	int	NO			
date	date	NO			
price	decimal	NO			

Σχήμα 6.8: Εκτέλεση ερωτημάτων σε SQL mode

The screenshot displays the AskSQL interface. On the left, the 'Query' section shows a natural language input: 'Find the number of tracks for each music genre, ordered in descending order by the number of tracks. Return the genre name and the number of tracks.' Below this, the 'Generated SQL' section shows the corresponding SQL query:

```
SELECT g.name, COUNT(t.trackid) AS track_count
FROM iee2021034.genre g
LEFT JOIN iee2021034.track t ON g.genreid = t.genreid
GROUP BY g.name
ORDER BY track_count DESC
```

On the right, the 'Results' section shows a table with two columns: 'name' and 'track_count'.

name	track_count
Rock	1297
Latin	579
Metal	374
Alternative & Punk	332
Jazz	130
TV Shows	93
Blues	81
Classical	74
Drama	64
R&B/Soul	61
Reggae	58
Pop	48
Soundtrack	43
Alternative	40
... (truncated)	...

Σχήμα 6.9: Εκτέλεση ερωτημάτων σε NL mode

The screenshot shows a 'Confirm Execution' dialog box with a yellow warning icon. The text inside the dialog reads: 'This query may modify or delete data.' At the bottom, there are two buttons: 'Cancel' (red) and 'Execute' (blue).

Σχήμα 6.10: Μήνυμα επιβεβαίωσης για την εκτέλεση επικίνδυνου ερωτήματος

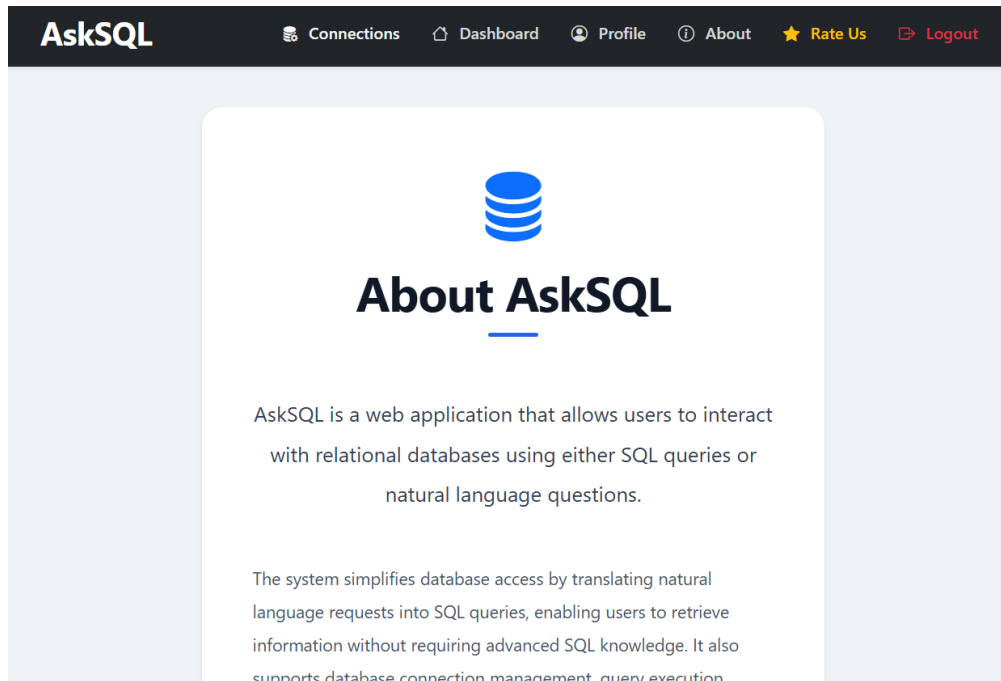
Change Password

New password

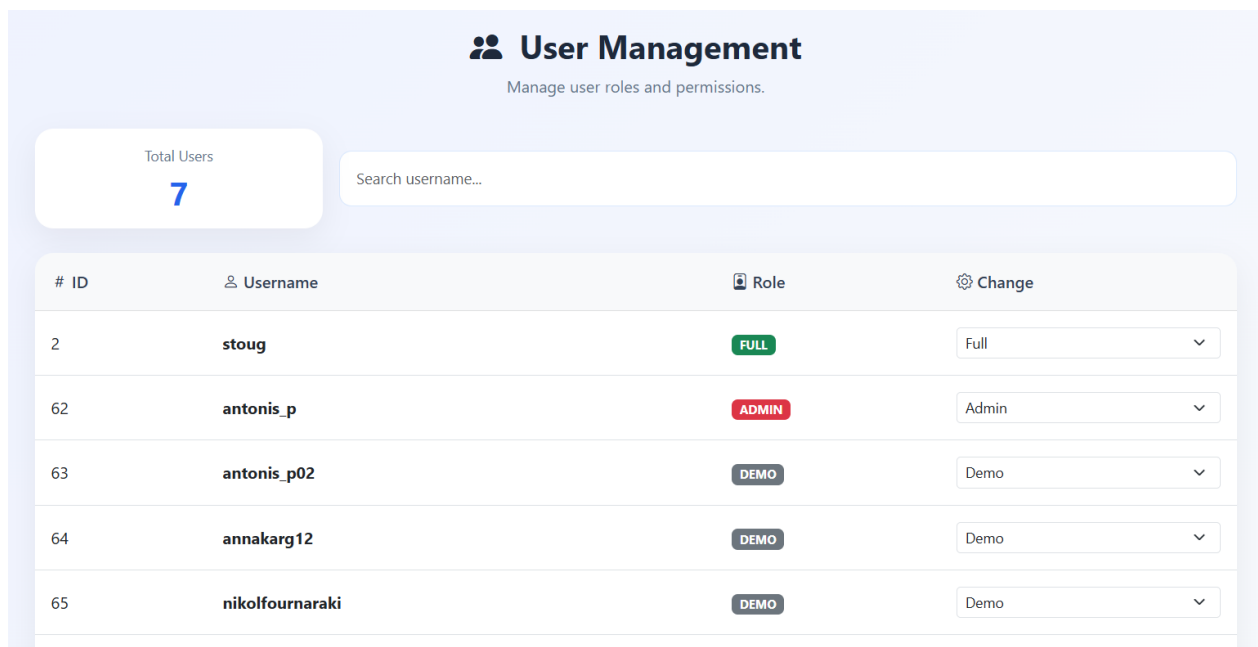
Confirm password

Change Password

Σχήμα 6.11: Φόρμα αλλαγής κωδικού πρόσβασης



Σχήμα 6.12: Σελίδα about



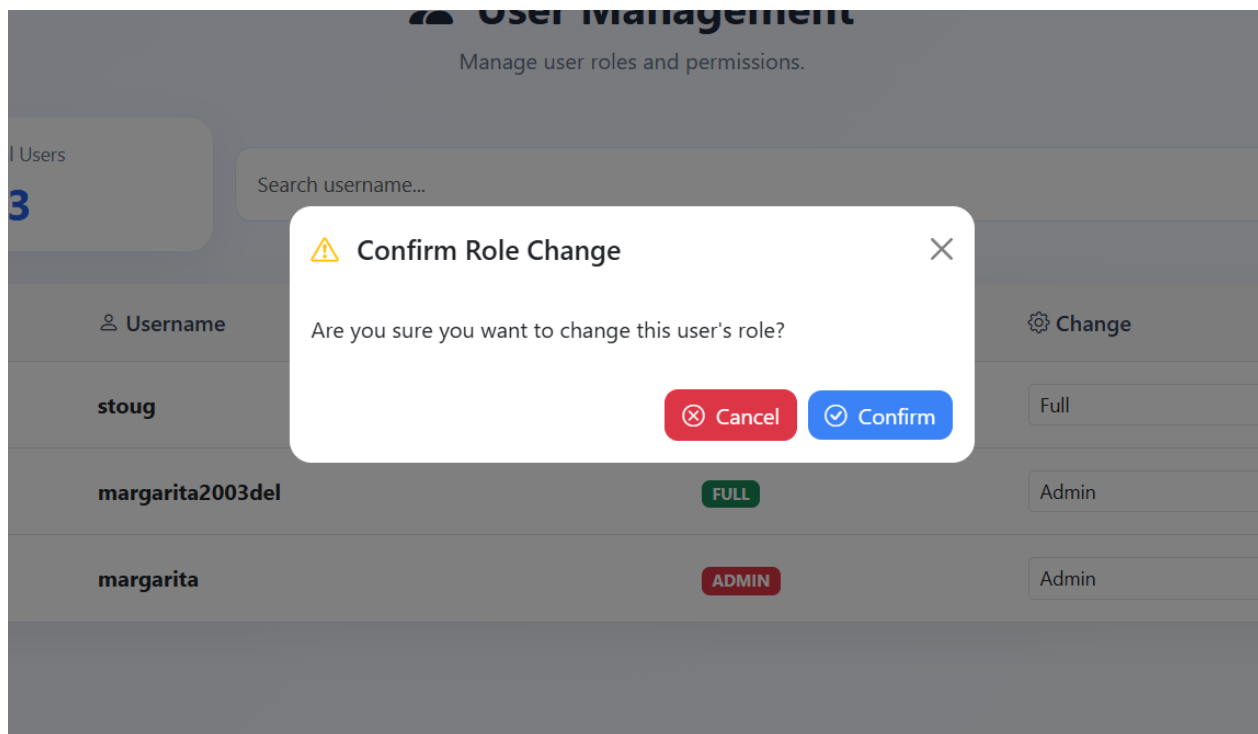
User Management
Manage user roles and permissions.

Total Users: **7**

Search username...

# ID	Username	Role	Change
2	stoug	FULL	Full
62	antonis_p	ADMIN	Admin
63	antonis_p02	DEMO	Demo
64	annakarg12	DEMO	Demo
65	nikolfournaraki	DEMO	Demo

Σχήμα 6.13: Εμφάνιση όλων των εγγεγραμμένων χρηστών



Σχήμα 6.14: Αλλαγή δικαιωμάτων χρήστη

Κεφάλαιο 7

Αξιολόγηση του AskSQL

7.1 System Usability Scale (SUS)

Η χρηστικότητα της εφαρμογής αξιολογήθηκε με τη χρήση του System Usability Scale (SUS) [33], ενός ευρέως χρησιμοποιούμενου ερωτηματολογίου για την αξιολόγηση της αντιλαμβανόμενης ευχρηστίας ενός συστήματος.

Το ερωτηματολόγιο αποτελείται από δέκα προτάσεις, στις οποίες οι συμμετέχοντες καλούνται να δηλώσουν τον βαθμό συμφωνίας τους χρησιμοποιώντας πενταβάθμια κλίμακα Likert, όπου η τιμή 1 αντιστοιχεί στο «Διαφωνώ απόλυτα» και η τιμή 5 στο «Συμφωνώ απόλυτα». Οι προτάσεις του ερωτηματολογίου παρατίθενται στον Πίνακα 7.1.

Πίνακας 7.1: Ερωτήσεις του ερωτηματολογίου System Usability Scale (SUS).

A/A	Ερώτηση
1	Νομίζω ότι θα ήθελα να χρησιμοποιώ αυτή την εφαρμογή συχνά.
2	Βρήκα την εφαρμογή περιττά πολύπλοκη.
3	Βρήκα την εφαρμογή εύκολη στη χρήση.
4	Νομίζω ότι θα χρειαστώ βοήθεια από κάποιον ειδικό για να χρησιμοποιήσω αυτή την εφαρμογή.
5	Βρήκα τις διάφορες λειτουργίες της εφαρμογής καλά ενσωματωμένες.
6	Νομίζω ότι υπήρχαν πάρα πολλές ασυνέπειες στην εφαρμογή.
7	Πιστεύω ότι οι περισσότεροι άνθρωποι θα μάθαιναν να χρησιμοποιούν αυτή την εφαρμογή πολύ γρήγορα.
8	Βρήκα την εφαρμογή πολύ δύσχρηστη.
9	Ένιωσα άνετα χρησιμοποιώντας την εφαρμογή.
10	Έπρεπε να μάθω πολλά πράγματα πριν μπορέσω να χρησιμοποιήσω την εφαρμογή.

Η βαθμολογία SUS υπολογίστηκε σύμφωνα με την καθιερωμένη διαδικασία. Για τις θετικά

διατυπωμένες προτάσεις (1, 3, 5, 7 και 9) αφαιρείται η μονάδα από την τιμή που επέλεξε ο συμμετέχων, ενώ για τις αρνητικά διατυπωμένες προτάσεις (2, 4, 6, 8 και 10) αφαιρείται η επιλεγμένη τιμή από το 5. Το άθροισμα των επιμέρους βαθμολογιών πολλαπλασιάζεται με συντελεστή 2,5, ώστε να προκύψει η τελική βαθμολογία SUS, η οποία κυμαίνεται από 0 έως 100 [33].

Για την ερμηνεία των αποτελεσμάτων χρησιμοποιήθηκε η κλίμακα αξιολόγησης που προτείνεται στη βιβλιογραφία. Σύμφωνα με αυτή, βαθμολογίες κάτω από 50 υποδηλώνουν χαμηλή χρηστικότητα, τιμές μεταξύ 68 και 80 θεωρούνται ικανοποιητικές, ενώ βαθμολογίες μεγαλύτερες από 85 αντιστοιχούν σε εξαιρετική χρηστικότητα (Excellent) [34].

7.2 Αποτελέσματα Αξιολόγησης

Η εφαρμογή αξιολογήθηκε από 20 συμμετέχοντες, οι περισσότεροι από τους οποίους ήταν φοιτητές πληροφορικής. Για κάθε έναν υπολογίστηκε η τελική βαθμολογία, σύμφωνα με τη διαδικασία που περιγράφηκε στην προηγούμενη ενότητα.

Η μέση βαθμολογία SUS που προέκυψε από τις αξιολογήσεις των συμμετεχόντων ήταν **87,9/100**. Σύμφωνα με την κλίμακα ερμηνείας που χρησιμοποιήθηκε, βαθμολογία μεγαλύτερη από 85 αντιστοιχεί στην κατηγορία *Excellent* [34]. Το αποτέλεσμα αυτό δείχνει ότι οι συμμετέχοντες αξιολόγησαν το AskSQL ως μια ιδιαίτερα εύχρηστη εφαρμογή.

Στον Πίνακα 7.2 παρουσιάζεται η τελική βαθμολογία SUS που υπολογίστηκε για κάθε συμμετέχοντα.

Πίνακας 7.2: Τελική βαθμολογία SUS ανά συμμετέχοντα.

Χρήστης	Βαθμολογία	Χρήστης	Βαθμολογία
1	87,5	11	95,0
2	97,5	12	87,5
3	97,5	13	100,0
4	95,0	14	82,5
5	90,0	15	95,0
6	100,0	16	52,5
7	50,0	17	75,0
8	80,0	18	87,5
9	95,0	19	95,0
10	100,0	20	95,0
Μέση βαθμολογία SUS		87,9	

Όπως προκύπτει από τα αποτελέσματα, η πλειονότητα των συμμετεχόντων αξιολόγησε την εφαρμογή με υψηλές βαθμολογίες χρηστικότητας. Μόνο δύο συμμετέχοντες παρουσίασαν σχετικά χαμηλότερες βαθμολογίες σε σύγκριση με τους υπόλοιπους. Η μέση βαθμολογία SUS των 87,9 μονάδων υποδεικνύει ότι η εφαρμογή προσφέρει πολύ καλή ευχρηστία και θετική εμπειρία χρήσης.

Κεφάλαιο 8

Συμπεράσματα και μελλοντικές επεκτάσεις

Στην παρούσα διπλωματική εργασία αναπτύχθηκε η διαδικτυακή εφαρμογή AskSQL, με στόχο τη διευκόλυνση της αλληλεπίδρασης των χρηστών με σχεσιακές βάσεις δεδομένων. Η εφαρμογή επιτρέπει την υποβολή και εκτέλεση ερωτημάτων τόσο σε SQL όσο και σε φυσική γλώσσα. Στην περίπτωση των ερωτημάτων φυσικής γλώσσας, αξιοποιούνται Μεγάλα Γλωσσικά Μοντέλα (Large Language Models - LLMs) για την αυτόματη παραγωγή των αντίστοιχων SQL ερωτημάτων, τα οποία μπορούν στη συνέχεια να ελεγχθούν και να εκτελεστούν από τον χρήστη.

Παράλληλα, υλοποιήθηκαν λειτουργίες για τη διαχείριση συνδέσεων με βάσεις δεδομένων MySQL και PostgreSQL, τόσο μέσω TCP/IP όσο και μέσω SSH tunnel. Η εφαρμογή παρέχει επίσης δυνατότητες διαχείρισης λογαριασμού, αποθήκευσης και διαχείρισης ιστορικού ερωτημάτων, φωνητικής εισαγωγής ερωτημάτων και εξαγωγής των αποτελεσμάτων σε αρχείο CSV. Με τον τρόπο αυτό δημιουργήθηκε ένα ολοκληρωμένο περιβάλλον για την υποβολή και εκτέλεση ερωτημάτων σε βάσεις δεδομένων, το οποίο μπορεί να χρησιμοποιηθεί τόσο από χρήστες με γνώσεις SQL όσο και από χρήστες με περιορισμένη εμπειρία στη χρήση σχεσιακών βάσεων δεδομένων.

Σημαντική συνεισφορά της εργασίας αποτέλεσαν τα πειράματα που πραγματοποιήθηκαν για την υποστήριξη αποφάσεων σχετικά με την ανάπτυξη της εφαρμογής AskSQL. Στόχος των πειραμάτων ήταν η διερεύνηση του τρόπου με τον οποίο διαφορετικές τεχνικές prompting και διαφορετικά LLMs επηρεάζουν την παραγωγή SQL στο συγκεκριμένο περιβάλλον της εφαρμογής. Για τον σκοπό αυτό χρησιμοποιήθηκε ένα σύνολο ερωτημάτων που σχεδιάστηκε για τις ανάγκες της παρούσας εργασίας και περιλαμβάνει διαφορετικά επίπεδα δυσκολίας και κατηγορίες λειτουργιών της SQL.

Αρχικά συγκρίθηκαν τρεις διαφορετικές τεχνικές prompting, οι οποίες διέφεραν ως προς την ποσότητα και το είδος της πληροφορίας που παρεχόταν στο γλωσσικό μοντέλο σχετικά με τη δομή και το περιεχόμενο της βάσης δεδομένων. Στο συγκεκριμένο πειραματικό σύνολο, τα αποτελέσματα έδειξαν ότι η παροχή του πλήρους σχήματος της βάσης δεδομένων σε συνδυασμό με ενδεικτικές εγγραφές οδήγησε στην καλύτερη απόδοση, επιτυγχάνοντας Execution Accuracy ίση με 90%. Επιπλέον, η συγκεκριμένη προσέγγιση παρουσίασε καλύτερη απόδοση και σε ερωτήματα αυξημένης πολυπλοκότητας.

Με βάση τα αποτελέσματα, η συγκεκριμένη τεχνική επιλέχθηκε ως η καταλληλότερη από τις

τρεις προσεγγίσεις για χρήση στην εφαρμογή AskSQL. Τα αποτελέσματα αυτά δείχνουν ότι η παροχή περισσότερης πληροφορίας σχετικά με τη βάση δεδομένων μπορεί να βελτιώσει την ποιότητα των παραγόμενων SQL ερωτημάτων.

Στη συνέχεια, χρησιμοποιώντας την επιλεγμένη τεχνική prompting, πραγματοποιήθηκε σύγκριση μεταξύ των δύο γλωσσικών μοντέλων που εξετάστηκαν στο πλαίσιο της εφαρμογής, του DeepSeek Chat και του GPT-4o mini. Χρησιμοποιώντας το ίδιο σύνολο ερωτημάτων και τις ίδιες συνθήκες εκτέλεσης, το DeepSeek Chat παρουσίασε Execution Accuracy 95%, ενώ το GPT-4o mini πέτυχε ποσοστό 90%.

Η εφαρμογή αξιολογήθηκε επιπλέον ως προς την ευχρηστία της μέσω του ερωτηματολογίου System Usability Scale (SUS). Το συνολικό σκορ της αξιολόγησης ήταν 87.9, παρέχοντας μία συνολική ένδειξη σχετικά με την ευχρηστία της εφαρμογής από τους συμμετέχοντες.

Όσον αφορά τις μελλοντικές επεκτάσεις, μία βασική κατεύθυνση είναι η υποστήριξη περισσότερων συστημάτων διαχείρισης βάσεων δεδομένων, όπως η SQLite. Με τον τρόπο αυτό η εφαρμογή θα μπορούσε να χρησιμοποιηθεί σε μεγαλύτερο εύρος διαφορετικών περιβαλλόντων και εφαρμογών. Επιπλέον, θα μπορούσαν να ενσωματωθούν περισσότερα γλωσσικά μοντέλα, παρέχοντας στους χρήστες μεγαλύτερη ευελιξία ως προς την επιλογή του μοντέλου που χρησιμοποιείται για την παραγωγή SQL ερωτημάτων.

Τέλος, μία επιπλέον μελλοντική επέκταση θα μπορούσε να είναι η ανάπτυξη έκδοσης της εφαρμογής για κινητές συσκευές, ώστε οι βασικές λειτουργίες του συστήματος να είναι διαθέσιμες και σε διαφορετικά περιβάλλοντα χρήσης.

Συνολικά, η ανάπτυξη του AskSQL έδειξε ότι η αξιοποίηση των Μεγάλων Γλωσσικών Μοντέλων μπορεί να υποστηρίξει αποτελεσματικά τη μετατροπή ερωτημάτων φυσικής γλώσσας σε SQL στο πλαίσιο μιας ολοκληρωμένης εφαρμογής αλληλεπίδρασης με σχεσιακές βάσεις δεδομένων.

Βιβλιογραφία

- [1] C. Coronel and S. Morris, *Συστήματα Βάσεων Δεδομένων*. Κριτική, 2015.
- [2] J. A. Hoffer, V. Ramesh, and H. Topi, *Βάσεις Δεδομένων: Σύγχρονη Διαχείριση*. Εκδόσεις Τζιόλα, 13 ed., 2016.
- [3] Oracle Corporation, "Mysql 8.4 reference manual." <https://dev.mysql.com/doc/refman/8.4/en/>. Accessed: 31-08-2026.
- [4] PostgreSQL Global Development Group, "Postgresql documentation." <https://www.postgresql.org/docs/>. Accessed: 31-08-2026.
- [5] A. Kumar, P. Nagarkar, P. Nalhe, and S. Vijayakumar, "Deep learning driven natural languages text to sql query conversion: A survey," 2022.
- [6] B. Li, Y. Luo, C. Chai, G. Li, and N. Tang, "The dawn of natural language to sql: Are we fully ready?," *Proceedings of the VLDB Endowment*, vol. 17, no. 11, pp. 3318--3331, 2024.
- [7] A. Singh, A. Shetty, A. Ehtesham, S. Kumar, and T. T. Khoei, "A survey of large language model-based generative ai for text-to-sql: Benchmarks, applications, use cases, and challenges," in *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 15--21, 2025.
- [8] Y. Huang, J. Guo, W. Mao, C. Gao, P. Han, C. Liu, and Q. Ling, "Exploring the landscape of text-to-sql with large language models: Progresses, challenges and opportunities," 2025.
- [9] Y. Ding and F. Wan, "Text-to-sql generation for medical databases via large language models and sql decomposition," in *2025 International Conference on Signal Processing and Communication Technology (SPCT)*, pp. 114--118, 2025.
- [10] A. B. Kanburoğlu and F. B. Tek, "Text-to-sql: A methodical review of challenges and models," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 32, no. 3, pp. 403--419, 2024.
- [11] P. Zhao, M. Shao, J. Cui, and H. Zhou, "A survey of nl2sql research: From foundations to frontiers," in *2025 2nd International Conference on Intelligent Perception and Pattern Recognition (IPPR)*, pp. 396--402, 2025.
- [12] X. Liu, S. Shen, B. Li, P. Ma, R. Jiang, Y. Zhang, J. Fan, G. Li, N. Tang, and Y. Luo, "A survey of text-to-sql in the era of llms: Where are we, and where are we going?," *IEEE Transactions on Knowledge and Data Engineering*, vol. 37, no. 10, pp. 5735--5754, 2025.

- [13] Z. Shaikhiyeva, M. Mansurova, and G. Amirkhanova, "Text to sql transformation using llm: A comparative research of t5, seq2seq, and sqlnet models," in *2024 9th International Conference on Computer Science and Engineering (UBMK)*, pp. 938--943, 2024.
- [14] L. Yan, H. Gong, P. Han, and Y. Xu, "A survey of large language model-based text-to-sql methods," in *2025 Asian Conference on Artificial Intelligence Technology (ACAIT)*, pp. 167--176, 2025.
- [15] L. Shi, Z. Tang, N. Zhang, X. Zhang, and Z. Yang, "A survey on employing large language models for text-to-sql tasks," *ACM Computing Surveys*, vol. 58, no. 2, 2025.
- [16] X. Xu, C. Liu, and D. Song, "Sqlnet: Generating structured queries from natural language without reinforcement learning," *arXiv preprint arXiv:1711.04436*, 2017.
- [17] B. Wang, R. Shin, X. Liu, O. Polozov, and M. Richardson, "Rat-sql: Relation-aware schema encoding and linking for text-to-sql parsers," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7567--7578, Association for Computational Linguistics, 2020.
- [18] V. Zhong, C. Xiong, and R. Socher, "Seq2sql: Generating structured queries from natural language using reinforcement learning," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 1331--1340, Association for Computational Linguistics, 2017.
- [19] T. Yu, R. Zhang, K. Yang, M. Yasunaga, D. Wang, Z. Li, J. Ma, I. Li, Q. Yao, Z. Roman, Z. Zhang, and D. Radev, "Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3911--3921, Association for Computational Linguistics, 2018.
- [20] J. Li, B. Hui, G. Qu, J. Yang, B. Li, B. Li, R. Wang, B. Qin, R. Cao, R. Geng, N. Huo, X. Zhou, C. Ma, K. Chang, F. Chen, G. Cheng, Y. Li, *et al.*, "Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls," in *Advances in Neural Information Processing Systems*, 2023.
- [21] B. Xu, Y. Cai, S. Shi, Z. Hao, and R. Cai, "Chat2db: Chatting to the database with interactive agent assisted language models," in *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, pp. 4620--4623, IEEE, 2025.
- [22] SQL Chat, "Chat-based sql client and editor for the next decade." <https://github.com/sqlchat/sqlchat>, 2026. GitHub repository. Accessed: 04-09-2026.
- [23] MDN Web Docs, "Html: Hypertext markup language." <https://developer.mozilla.org/en-US/docs/Web/HTML>. Accessed: 31-08-2026.
- [24] MDN Web Docs, "Css: Cascading style sheets." <https://developer.mozilla.org/en-US/docs/Web/CSS>. Accessed: 31-08-2026.

- [25] The Bootstrap Authors, "Bootstrap v5.3 documentation." <https://getbootstrap.com/docs/5.3/getting-started/introduction/>. Accessed: 31-08-2026.
- [26] MDN Web Docs, "Javascript." <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. Accessed: 31-08-2026.
- [27] PHP Documentation Group, "Php: Hypertext preprocessor." <https://www.php.net/intro>. Accessed: 31-08-2026.
- [28] OpenAI, "Gpt-4o mini model." <https://developers.openai.com/api/docs/models/gpt-4o-mini>. Accessed: 04-09-2026.
- [29] DeepSeek, "Deepseek api documentation." <https://api-docs.deepseek.com/>. Accessed: 04-09-2026.
- [30] MDN Web Docs, "Web speech api." https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API. Accessed: 31-08-2026.
- [31] Google, "Openid connect." <https://developers.google.com/identity/openid-connect/openid-connect>. Accessed: 31-08-2026.
- [32] GitHub, "Github documentation." <https://docs.github.com/en>. Accessed: 31-08-2026.
- [33] J. Brooke, "Sus: A quick and dirty usability scale," in *Usability Evaluation in Industry* (P. W. Jordan, B. Thomas, B. A. Weerdmeester, and I. L. McClelland, eds.), pp. 189--194, London, U.K.: Taylor & Francis, 1996.
- [34] A. Bangor, P. T. Kortum, and J. T. Miller, "An empirical evaluation of the system usability scale," *International Journal of Human-Computer Interaction*, vol. 24, no. 6, pp. 574--594, 2008.