Deep Science Publishing, 2025
https://doi.org/10.70593/978-93-7185-191-6_6

DeepScience
Open Access Books

Chapter 6: SQL competency development for technical interviews: A framework for query practice, mock testing, and portfolio building

Mohanraju Muppala

1. Introduction to SQL Interviews

SQL supports a wide range of both technical and business-oriented roles within various organizations, reflecting its versatility and importance in the modern data-driven landscape [1,2]. For business users, it is crucial to grasp the nuances of data to effectively extract pertinent and actionable information that can drive decisions and strategies. On the other hand, for technical users, the creation, management, and optimization of databases represent essential tasks that require a deep understanding of SQL's capabilities. In November 2022, there was a notable selection of standout SQL interview questions that focused on various important topics such as JOINs, Subqueries, Aggregates, GROUP BY, and Multiple-Table queries [3-5]. These questions were carefully collected from recent SQL interviews conducted at some of the world's leading and most innovative companies, highlighting the growing demand for SQL proficiency in the job market.

Preparing for SQL queries can be both a fun experience and a challenging endeavor at the same time. A great starting point is to take a detailed job description and create an SQL portfolio that aligns with the skills and requirements outlined in that description. Following this, one can set up a mock interview by utilizing the query list as a basis. The selected queries provided below span a diverse range of relevant topics, ensuring a comprehensive

understanding and application of SQL principles. This process not only enhances one's knowledge but also boosts confidence in handling real-world SQL challenges.

2. Common SQL Interview Questions

Interviewers frequently ask SQL coding questions during the interview process to thoroughly verify a candidate's knowledge and comfort level with SQL. These questions typically fall into two main categories: writing SQL queries and crafting SQL functions. SQL queries are designed to retrieve data from one or more tables within a database, which is fundamental for understanding data interaction. Meanwhile, SQL functions are essential for manipulating data in various ways, such as rounding decimal numbers, aggregating data, or concatenating two strings into a single value [2,6]. These skills are critical to effective database management and demonstrating proficiency in SQL.

Because SQL is specifically designed for data retrieval and manipulation, it excels remarkably well even in the face of complex interview questions. SQL tasks that may appear quite intricate and complicated in other programming languages often simplify down to just a few lines of code when written in SQL. At the same time, possessing a deeper and more thorough understanding of JOIN operators as well as foreign keys—concepts that are unique and specific to SQL—significantly increases the difficulty of coding and logic-related questions. Many individuals who are not well-versed in SQL struggle to write even a simple JOIN query that effectively retrieves matching columns from two distinct tables, even after having gone through multiple tutorials and instructional resources. This gap in understanding underscores the inherent challenges for those new to SQL.

2.1. Basic SQL Queries

These are some of the most common queries that a beginner SQL developer is expected to understand and answer. SQL usually means a single query is run once [7-9]. A client may want multiple pieces of information, so multiple queries are issued when the important bits are wanted.) In almost all cases, the company wants to see you think about the QA part of the queries because that is your primary job—know what the client wants, know what the DB can provide. Understand requests at a basic level. Some companies do this better than others [10,11].

That is, a client might ask to see details for all orders in the last 30 days because they might need to contact customers who placed the orders for purposes of making a complaint. You hear the complaint part and go, “Oh, the client needs customer contact info, so I must add that info to the output.” This keeps the client happy, as the request is fulfilled on the first request rather than waiting for the follow-up request.) Imagine an e-commerce website that keeps track of customer data, product data, and historical order data.

2.2. Joins and Subqueries

A join is an operation that combines columns from one or more tables in a relational database.

In a typical join, the resulting rows contain columns from rows that satisfy the join predicate. For example, a join on an equality between two columns of two nonempty tables will produce the empty table only if the columns contain no common values [12-14]. Classes of joins include natural join, inner join, outer join, left join, right join, semi-join, and anti-join.

Four types of joins are commonly supported by the SQL standard: INNER JOIN: the join predicate matches rows between two tables; LEFT JOIN: rows in the right table are matched with rows in the left table, or unconditionally included if there is no match; RIGHT JOIN: rows in the left table are matched with rows in the right table, or unconditionally included if there is no match; FULL JOIN: performs the effect of both the LEFT JOIN and RIGHT JOIN.

A join is a means for combining fields from two tables by using values common to each. Joining tables is a database operation that relates the tables based on data common to each. For example, a bank database could consist of a table with one row for each customer and a table with one row for each account. Each account is related to a customer by storing the identifier of the customer who owns it; the same identifier is stored in the row for that customer.

A subquery is a SELECT statement nested within another statement such as SELECT, INSERT, UPDATE, DELETE, or another subquery.

2.3. Aggregate Functions

Aggregate functions are one of the most common used types of SQL functions. They perform a calculation on a set of values and return a single value. A common example is the AVG( ) function, which calculates the average of the provided values.

Not all clauses and keywords work with aggregate functions. For example, the WHERE clause cannot be used with aggregate functions. The following is a list of aggregate functions in SQL: AVG(), CHECKSUM_AGG(), COUNT(), COUNT_BIG(), GROUPING(), GROUPING_ID(), MAX(), MIN(), SUM() and STDEV().

2.4. Data Manipulation Language (DML)

Although Structured Query Language (SQL) statements are divided into Data Definition Language (DDL) statements and Data Manipulation Language (DML) statements, the focus is mainly on the latter. Data manipulation language statements involve data definition and alteration. Operations include inserting, deleting, updating, selecting, merging, and calling procedures and functions.

SQL is a database computer language designed for querying, modifying, and managing data. No SQL statement can retrieve data from a database without first selecting the data to be retrieved. The SELECT statement has an array of clauses and conditions to assist queries. The following are some important SQL commands used for data manipulation: INSERT into TABLE (col1, col2) VALUES (val1, val2) (to insert new rows); UPDATE TABLE SET col1= val1 WHERE condition (to update existing rows); DELETE FROM TABLE WHERE condition (to delete existing rows).

2.5. Data Definition Language (DDL)

Understanding the different types of SQL languages is important. SQL statements are grouped into the following categories: Data Definition Language (DDL), Data Manipulation Language (DML), Data Control Language (DCL), and Transaction Control Language (TCL). The Table_V0001 below lists different SQL statements groups.

DDL statements define database objects like schema, tables, indexes, etc. These operations affect logical structure and can be done only by users having a certain database privilege. These do not affect logical structure of data. Since the content of a table is not affected by these statements, these cannot be rolled back. The auto commitment is implicit.

Data Definition Language (DDL) statements define database objects like schema, tables, indexes, and more [3,15-17]. Such operations affect their logical structure and require users to possess specific database privileges. Since the content of tables remains unaffected, these operations cannot be rolled back and are implicitly auto-committed.

2.6. Advanced SQL Concepts

Advanced SQL concepts enable developers to create efficient queries necessary for complex applications. As data sizes grow, it becomes essential to understand specific capabilities of SQL constructs—and how to wield that power—to write programs that will run efficiently on increasingly vast volumes of data. For example, while the HAVING clause was created to allow filtering on aggregated columns, because HAVING is applied after GROUP BY, filtering any conditions that do not require aggregated computations can be pushed into the WHERE clause.

Window functions offer several capabilities that make processing much easier than using correlated subqueries or other techniques. As a result, window functions often outperform other techniques since they reduce the need for correlated subqueries. The SQL CASE expression, performing a logical IF-THEN-ELSE, offers great flexibility for writing queries by allowing multiple tests to be developed and executed in sequence. Neutralizing NULLs is essential when writing queries—otherwise results may be incomplete or incorrect. The new FILTER clause offers a much easier way to write conditional aggregates, yet also a means to improve performance by filtering out unnecessary rows before aggregation begins.

3. Case Studies from Leading Companies

This section delves deeper into a range of diverse SQL case studies meticulously drawn from the extensive experiences of several leading organizations across various industries. It offers a comprehensive and detailed overview of the myriad challenges that have been effectively addressed and the innovative strategies that have been thoughtfully implemented. This information has been compiled from the rich insights and unique approaches employed by industry giants such as Amazon, Google, Facebook, Apple, Microsoft, and Netflix, among many others. These compelling accounts not only illustrate a variety of important SQL concepts but also demonstrate how SQL can be skillfully integrated with other powerful query languages. This integration allows organizations to fully harness the sheer power of Big Data and the advancements in artificial intelligence, leading to enhanced data processing, analysis, and the ability to derive valuable insights from vast amounts of information.

Amazon has developed a range of innovative SQL-based applications and employs PostgreSQL in combination with MadLib, a library for scalability, cloud computing, and machine learning. Google and Facebook implement BigTable, and Google utilizes F1, a query language based on SQL but extended for the requirements of its Spanner database. Apple manages a fleet of thousands of machines running PostgreSQL, supported by a workflow engine that orchestrates complex operations. Microsoft employs a hybrid approach that integrates SQL and Cosmos DB to leverage Big Data, artificial intelligence, and SQL Machine Learning. Netflix addresses enormous volumes of data by employing a bidirectional SQL join with a MapReduce-based query engine.

3.1. Company A: SQL Challenges

Companies occasionally use a database with potentially questionable SQL code to present challenging questions to prospective employees [18-20]. An example from the mentioned company provides the following scenario: employees are represented by the emp_table, jobs by the job_table, and the relevant qualifications of an employee by the emp_skills_table for the jobs in the job_table. The stated question is whether the SQL code given in the example answers the intended question correctly. The query is designed to list jobs for which there are currently no qualified employees and those with less than five. It uses a Common Table Expression (CTE) to aggregate the count of qualified employees per job and then selects the job name and count of qualified employees with a condition to include only those having fewer than five qualified employees.

The actual query in the database differs slightly from the example. The job_name column is not output but used instead as a filter criterion in the WHERE clause to include only specific job names. For instance, when searching for the head of Research and Development, the query would look for a count of qualified employees for that particular job name. A question from this scenario asks how to write a query that returns a list showing the number of currently qualified employees for all jobs, including those with five or more employees, and what changes are necessary to the original query to achieve this. The answers provided draw on the tables and data given, using particular examples such as the job_name 'Head of R&D, Coca-Cola' with a count of two.

3.2. Company B: Data Analysis Case

With a strong interest in SQL, one candidate applied for a Data Analysis internship position at Company B. Although the candidate had limited SQL

knowledge, the final interview stage entailed presenting a written case. Hence, the candidate studied the topic extensively and completed all eight exercises, providing solutions that encompassed advanced analytical functionalities and operations. The case required analyzing a database of iFood deliveries and customer orders in Brazil, using two tables: "orders" (information on all delivered orders) and "order_items" (detailed information on the dishes within each order).

The first part of the case consisted of the following questions and their respective solutions.

Which business questions would you answer by extracting information from these tables? Raise as many questions as you find interesting, commenting on the reasoning for each one. What other information would you like to have in each table that would help your analysis? What are the constraints of these questions? What kind of question would you not be able to answer with these tables?

Question Reasoning Additional Information Constraints

1. What percentage of orders have more than one dish? By differently identifying orders with a different number of dishes, it is possible to understand whether most customers consume more than one dish on each visit.

– Detailed price information on each dish would allow an analysis of how much customers generally spend on each visit.

– This analysis does not consider the number of people that each visit intends to serve.

2. What is the expected revenue per dish order? This information would assist in setting standards for microwave times for each dish—categorizing them as dish 1, dish 2, and so on.

– The current structure does not specify the preparation time required for each ordered dish.

3. What percentage of orders were made by customers who have only evaluated the store in this order? It is important to assess the reliability of the store evaluation.

– Customer evaluation data in the orders table and order owner details in the customer table would enhance this analysis.

– Evaluation reliability may also vary depending on the dishes ordered.

4. What is the distribution of sales per month/quarter/semester/organization? Analysis of broad sales distributions using appropriate filters can be informative.

– The current tables, while including order dates, do not delineate specific items that allow for identification of individual sales.

5. Is it possible to identify items that are produced in cold or hot seasoning? Grouping dishes by category can facilitate an analysis of which types of dishes yield higher profits or popularity.

– While seasonality can be categorized separately, the present table lacks a categorical classification of dishes.

3.3. Company C: Performance Tuning

The following notes summarise existing SQL interview questions and answers relating to Company C.

Company C – Case Studies: Performance Tuning

Question 1: How do you improve the Performance of the SQL query? In Brief

Query Tuning refers to making changes to the query to reduce response time and resource consumption. A poorly written query can utilize more system resources and may also get timed out on execution. An efficient query will improve the performance.

Following are some of the steps to be followed for query tuning: Check the execution plan and check if there is a full table scan on the table or index scan and check the time for reading first few rows and total time and if possible try to rewrite the SQL query answering the same business question and observe the execution plan and response time. Identify the step or table accessing operation which is taking more time or cost and check for the alternative method of doing the same."

3.4. Company D: Practical Uses and Real-World Implementations

When people think about SQL, the very first real interaction that often comes to mind for most individuals usually originates from attempts to access various things on the ever-expanding internet. Behind every successful e-commerce business, there is consistently a well-structured database that logically organizes and stores all the critical product information, including comprehensive details about customers and employees alike. Learning to effectively run operations on

these extensive databases can sometimes feel like a daunting and overwhelming task; thus, these carefully constructed examples provide one of the best possible approaches to mastering various SQL commands. The following queries specifically cover some of the most common needs for interaction with a database, so let's begin by taking a look at a hypothetical company that focuses on online book delivery services. In this scenario, a STUDENT is in the process of preparing meticulously for an important interview. QUESTION! 1. Write a query that will display all books that belong to a particular author, considering relevant details. 2. Construct a query to display all necessary details about a book when given its title, ensuring clarity and accuracy in the output..

4. Preparing for the SQL Interview

One vital phase in SQL interview preparation consists of finding quality questions and taking care in finding questions with detailed answers that delve beneath the surface by covering syntax, exploring multiple approaches, and clarifying abstractions. Quality preparatory material tends to be exhaustive and permits use in several ways for automation and tailoring. It usually presents questions at different levels to suit a broad spectrum of applicants, categorizing and sorting questions accordingly. It typically drills down into the usage, syntax, abstractions, limitations, and alternatives of the varied concepts and methods. Moreover, such preparatory material strives to assemble a complete set of questions for a given job and create variations on these questions in order to yield a project-specific subset [21-23]. The overall intent is to assist readers in searching precisely and efficiently through sets of questions and answers for different accounts and project groups.

Besides individual practice, another tactic to prepare for interviews is to get acquainted with the interview environment in advance. There are dedicated platforms on the internet for this purpose, where practice enables individuals to experience the actual interview room setup prior to the official interview. Since SQL is the language of choice for practically every database interview, a company sending SQL questions can present them in myriad different forms. Data can be presented to candidates either graphically or in tabular form, and many online practice sources adopt this approach.

4.1. Understanding the Job Description

A DBA must be able to multi-task and collaborate with multiple teams. An Oracle DBA helps designers, developers, testers, and implementation managers

in resolving Oracle database related issues and implements database triggers, functions, and procedures that help in querying and extracting the database information. Mainly the roles of DBA are to administer, monitor, implement, and upgrade the database and maintain the database objects according to user requirements. Also the role of SQL Queries in Software testing includes creating tables, inserting data, creating different queries, verifying joins, file-operations, subqueries, bulk loading, extract files, and various Tortoise commands.

Generally, a Database Administrator is responsible for the performance, integrity and security of a database. Other important tasks may include planning, development, troubleshooting, fine-tuning, migration, backup and recovery, and monitoring of both the physical and logical aspects of the database to ensure consistency and integrity of the data holders. Its role matrix also includes data warehouse support, extraction, loading & transformation, backups, and space management.

4.2. Building a SQL Portfolio

In this chapter, you learn how to prepare for difficult SQL interview questions by building a SQL projects portfolio—an online collection of SQL projects that you can share with hiring managers. Moreover, any hiring manager can view an online SQL project. Correspondingly, this is a distinct advantage in your interview preparation because such a hiring manager can check your skills on actual work instead of solely relying on you. The portfolio develops independently while you focus on other tasks. You add SQL projects to your portfolio as you add skills, improving the projects and your interview preparation over time.

The portfolio development enforces a ISforSQL preparation methodology: Interview preparation planning, questions selection and optimization, project execution, and summarization. Prior steps produce questions, The interview preparation questions chosen concentrate on challenging interview questions that are suitable for project execution [9,24,25]. Collection of questions on a required topic is possible through search and class conditions. The applications execute questions on the database extracted from the target project, saving results in local files and databases. Video recordings during execution show that the computer worked actively on the problem. Post-execution video analysis provides insights into the development life cycle. Additionally, project summaries provide SQL examples for a selected topic.

4.3. Mock Interviews and Practice

Mock interviews are an essential and integral part of the preparation process for the real interview that candidates will inevitably face. They serve as a valuable tool in significantly reducing nerves and equipping candidates with effective strategies to tackle a variety of interview questions. Engaging in early practice sessions helps to identify weaknesses in knowledge or performance—topics that are frequently missed during these mock tests often emerge as critical areas that require reinforcement and focused attention. By thoroughly covering all topics and ensuring that there are no more failed mock interviews, candidates can develop a deep and comprehensive understanding of the material they need to master. As the real interview date approaches, conducting mock interviews becomes increasingly realistic and beneficial, allowing candidates to simulate the actual interview experience more closely. This practice not only enhances their confidence but also significantly improves their readiness for any challenges that lie ahead in the interview process itself. With consistent preparation through mock interviews, candidates can feel better poised and prepared to showcase their skills effectively when it truly matters.

The accompanying SQL mock interview test offers a thorough and extensive collection of eighty highly realistic and thoughtfully curated questions, meticulously generated algorithmically to guarantee both reliability and richness for the candidates. Following each of these answers, you will find an informative mini-lecture that is presented with clarity, thoroughly exploring the relevant topic in great detail. The breadth and depth of these extensive preparations typically represent a significant commitment, often amounting to about six months of dedicated and focused study, allowing candidates to rigorously test their knowledge and effectively improve their skills in various areas of SQL. This preparation process is crucial for those aiming for SQL certification.Pathway and Roadmap for Success

A comprehensive roadmap to SQL certification is carefully presented, starting with the critical process of identifying personal and professional goals. One potential goal might involve advancing in SQL programming skills significantly to ultimately achieve the prestigious title of Software Architect, which can open doors to numerous opportunities. Another possible aim could be to generate a steady stream of passive income by becoming an Oracle SQL Certified Associate, enabling the individual to offer valuable training or specialized consulting services to others in need. In either scenario, a well-thought-out and strategically planned route to success is not only beneficial but absolutely essential in navigating the complexities of this field.

Key steps include envisioning VISION, MISSION, PURPOSE, and LONG-TERM GOALS. For instance: Acquiring certification for career advancement through disciplined study, including a couple of hours per day during weekdays. The forthcoming Oracle SQL certification will validate the skills, having acquired a solid foundation through several SQL and PL/SQL projects. The completed projects, particularly in PL/SQL, highlight expertise in the chosen database. Subsequent steps involve outlining the entire journey with a list of resources, setting targets ET1 to ETn, and scheduling them on a calendar in sequence. For example, ET1 to ET3 might focus on preparation, followed by registration (ET4) and the examination (ET5).

5.1. Overview of SQL Certifications

Many SQL-related certifications are provided by professional organizations and private vendors. For example, the Oracle Database SQL Certified Associate Certification enables users to demonstrate proficiency in SQL concepts and fundamentals associated with Oracle Database products, such as extracting, manipulating, and transforming data from tables, executing data definition language statements, controlling user access to the data, and managing objects. The Microsoft Technology Associate (MTA) Database Fundamentals Certification covers core database concepts, including creating database objects, manipulating data, querying Microsoft SQL Server, and understanding how SQL Server controls data access [26-28].

Database administrators and developers require expertise in these domains. Other business intelligence and data professional certifications related to data query and manipulation include the Microsoft Certified Solutions Expert in Business Intelligence (MCSE) and the IBM Certified Database Administrator – DB2 specialization. Numerous courses prepare candidates for these examinations, and many hosting companies provide hosted database environments for exploration and training.

5.2. Choosing the Right Certification

Certification plays a significant role in clearly distinguishing candidates during the hiring process, and it is especially valuable for individuals seeking validation of their skills and knowledge. In cases where certification is not an explicit requirement set by employers, it can still serve as a powerful tool to demonstrate proficiency and a genuine commitment to continuous learning and professional development. This level of dedication is highly appealing to prospective employers and can positively influence hiring decisions.

Numerous institutions provide specialized training and certification programs specifically focused on SQL, fully aligned with the final assessment component necessary for completion. A well-structured certification path adds considerable value and depth to the entire preparation journey for aspiring database professionals. Several industry-grade certifications have placed significant emphasis on this area, successfully delivering comprehensive programs tailored to mastering SQL. The written assessment may very well require a profound understanding of the various concepts detailed within the SQL Assessment: Key Concepts section, which is crucial for achieving certification. This structured approach ensures that candidates are not only knowledgeable but also adequately prepared for the challenges of real-world applications in SQL environments.

5.3. Study Materials and Resources

A thorough SQL interview preparation typically involves engaging with a plethora of pertinent study materials, such as the well-regarded "SQL Interview Questions" self-study guide. Numerous free online resources further complement and enrich these foundational materials, providing insight and a broader perspective. An inquiry into the effectiveness of various paid preparation courses reveals that these can significantly enhance a candidate’s ability to foresee, analyze, and answer interview questions effectively, contributing to a more focused and comprehensive preparation strategy. By adopting a practiced approach and methodically working through these questions, candidates often discover how adeptly they can solve problems during the actual interview. Navigating through the complexity of potential SQL scenarios and questions posed by interviewers requires both practice and a deep understanding of the subject matter to excel and stand out among the competition.

The question-answer format provides candidates with a systematic approach to carefully work through various questions, allowing them to identify and spot any errors, thoroughly regard the job description, and critically evaluate their answers from the perspective of the interview environment. This thorough preparation is crucial as it helps candidates feel more confident and ready for real-world scenarios. Even after Rhonda, who is a dedicated software engineer working at a progressive data warehousing company, successfully obtains the job, she remains committed to preparing for future SQL interview questions focusing on important concepts, various functions, essential keywords, and different types of commands, as well as joins, which are all vital for her continued professional development. Other commonly sought after positions in

this evolving field include roles such as Data Engineer, Database Administrator, and Analytic Engineer, each requiring a robust understanding of SQL and data management principles. For optimal learning and effective practice, Visual Studio Code is often recommended as the ideal editor for writing queries, providing a helpful platform for those looking to enhance their SQL skills.

5.4. Exam Preparation Strategies

Exams test the ability to recall and apply knowledge in set time frames. Many students find that standard study techniques, such as "cramming," can lead to the specialist studying not only the main topics but also the fringes, producing confusion and leading to less-than-ideal results [6,29-31]. The goal of preparation is to maximize memory retention of the fundamental concepts and facts related to the discipline. Applying appropriate preparation strategies during the semester will leave the student confident and better prepared for the exams. The memory effect erodes rapidly during sleep, dialling back what has been studied, and specificity of preparation improves the long-term effect, i.e., the remembered answer is the prepared one, and frequently correct answers are given to frequently practiced questions.

6. Soft Skills for SQL Interviews

Since a candidate's answers in an interview are in written rather than verbal form, the communication skills a candidate demonstrates while answering must also be considered. This includes spelling, grammar, and sentence construction on written answers, as well as the logic and thoroughness of the answers.

Non-technical questions arise in all SQL interviews and should form part of interview preparation also. For example, consider that in an interview the candidate states, "We used SQL Server Management Server for the project". This may not be relevant to the question and the candidate may have meant to refer to SQL Server Management Studio, but inappropriate terms may confuse a recruiter, especially if English is not the candidate's first language [32,33].

6.1. Communication Skills

Although SQL is not primarily a communication language, it does have well-defined protocols for communication between clients and servers. Certain aspects of this communication protocol sometimes leak into standard SQL. SQL statement terminators, used by parsers to identify the end of one statement and the beginning of the next, are a good example. A statement can be

terminated in almost any way, but the accepted end of statement terminator is the semicolon (;). This convention originated from the SQL communication protocol.

Another example is the inclusion of comments in the code. Although SQL comments are ignored by the SQL parser, the client-to-server communication protocol supports comments within an SQL statement. These comments are sent to the server during SQL-server communication, making comments a part of SQL. Comments are placed at the beginning of the statement before the keywords or identifiers, as supported by the communication protocol.

6.2. Problem-Solving Approach

A large set of SQL problems collected and categorized systematically allows for structured practice. The problems are group-indexed according to subject categories and sorted in ascending order of difficulty. Practice proceeds by selecting a category and then attempting questions from easier to more difficult.

Structured practice encompasses more than the mere resolution of questions and submission of answers. Guidelines for start, resolution, and submission of a question are provided in the QnA section. Detailed guidelines encompassing preparation and execution phases optimize the solution process. The preparation phase involves setting up the necessary environment, deciding the order of solving, and scope determination. The execution phase addresses coding, testing, submission with time management, and seeks troubleshooting guidance when needed.

6.3. Teamwork and Collaboration

SQL can perform complex operations on structured data and share the final result with a team of developers, filtered to suit each person's needs. This enables two developers to collaborate on the same table, independently adding new data and then combining their work using UNION, even if the data schemas differ [34-36]. In complex projects, completely different teams may work on different tables; these can be combined using JOIN operations. However, the most common task using JOIN is displaying information stored in different tables during user searches. For example, a table of job positions could be joined with a table of candidates for each of those positions so the hiring team can see the names of the candidates. Managers who are responsible for the same department’s projects but a different table, can also use JOIN to combine their work, then filter and sort the data to suit each person’s needs.

7. Post-Interview Strategies

The post-interview phase warrants immediate attention, where it is highly advised to send a thank you note upon completion of the interview. This message should not only express gratitude but also provide another opportunity to mention any relevant information inadvertently omitted during the interview. It must be succinct, complimentary yet non-sycophantic, thereby ensuring the interviewer recalls the candidate positively.

After the interview, thoroughly reviewing and analyzing both the questions posed and personal responses is crucial. Sharing the interview experience—including questions, answers, and impertinent details, omitting the site's name—with communities such as r/SQL allows others to benefit. Such exchanges also serve as excellent preparation for upcoming interviews and contribute regular updates to a community-maintained Questions and Answers collection.

7.1. Follow-Up Communication

Follow-up communication helps you stay on the interviewer's radar and shows your continued interest in the position. The best form of follow-up communication is email. It's easy to send, doesn't require scheduling, and lets you arrange your thoughts carefully. The follow-up email should be concise and polite, and reiterate your enthusiasm for the position. If the interviewer gave a specific date by which he or she will make a decision, wait until that time has passed before sending the follow-up email [16,37-38]. If when asked the interviewer did not supply a decision date, send the email two business days after the last interview.

7.2. Learning from Feedback

Natural Language Processing has advanced to such an extent that end users of a system can specify what is to be done using natural language sentences. Additionally, researchers have developed in terms of Supervised and Unsupervised Machine Learning methods to identify and acquire new features for a system. The challenge for the system then becomes to select what is an appropriate feature set for an input statement, which in turn depends on the underlying knowledge database.

In many real-world situations, the domain of the solution space is large or a sequence of commands is expected during a conversation. In this context, users generally provide feedback for each response of the system; however, it is difficult for the system to learn from that feedback and self-improve. Learning-

by-Doing implemented using Answer Tree-based Supervised Machine Learning helps a system learn appropriate features based on the feedback provided during a conversation with the user.

7.3. Continuous Improvement

Automation becomes increasingly important with continuous deployment. At the very least, test execution is automated. In the best development organizations, every aspect of testing is automated. Changes are committed, and a build and test cycle kicks off automatically. If the build finishes without errors, the tests start running. When the tests finish, an email arrives in the developers' inboxes. (Usually, these emails are color coded, to make the results as obvious as possible.) In addition, an indicator light on the wall changes color. That way, if a bug is detected, no matter where the developers are, they immediately know what went wrong. It couldn't be easier to follow the advice of Boehm and Papaccio: "Fix the errors soon after they are injected." Test results are available so quickly that bugs are detected a day after they appear in code, and then incorporated into a build that goes live within a few days. This process beats the old recommended approach of "Find the errors as close to the time of introduction as possible."

Continuous monitoring can be applied to test coverage as well. In addition to receiving information about coverage, developers should be able to uncover these so-called untested parts of the application, and actually add automated tests that cover these parts. Thus, automation leads to continuous coverage improvement, where coverage not only is reported, but also becomes a matter of inspection. The best way to ensure better test coverage is to openly share coverage information and make it visible in a culture of continuous improvement.

8. Conclusion

This document enumerates key SQL interview questions and answers, covering design, architecture, and development. Topics include SQL operations, querying tables, joins, set operations, SQL tools, normalization, stored procedures, triggers, indexes, views, functions, transactions, and differences between DELETE, TRUNCATE, and DROP statements.

SQL, a structured query language, enables database access and control. It supports Oracle, MySQL, and MS SQL Server. SQL statements manage data

and database objects. DDL commands create schema objects; DML commands manipulate data; and DCL commands grant and revoke privileges. A schema defines the database's logical structure, while an instance is its content at a given time. The Exam–Prepare–Practice approach builds expertise.

References:

[1] Chiligireddy S. The Importance of Technical Competencies in the Engineering Manager Role and How Companies Evaluate Them in Interviews. IEEE Engineering Management Review. 2025 Mar 12.

[2] Tomova M, Hofmann M, Hütterer C, Mäder P. Assessing the utility of text-to-SQL approaches for satisfying software developer information needs. Empirical Software Engineering. 2024 Jan;29(1):15.

[3] Tomova M, Hofmann M, Hütterer C, Mäder P. Assessing the utility of text-to-SQL approaches for satisfying software developer information needs. Empirical Software Engineering. 2024 Jan;29(1):15.

[4] Cui J, Zhang R, Zhou F, Li R, Song Y, Gehringer E. How much effort do you need to expend on a technical interview? a study of leetcode problem solving statistics. In2024 36th International Conference on Software Engineering Education and Training (CSEE&T) 2024 Jul 29 (pp. 1-10). IEEE.

[5] Machhale G, Kadam S, Patki V. Integration of AI Based Chatbot in Tech Interview Prep App. In2024 2nd World Conference on Communication & Computing (WCONF) 2024 Jul 12 (pp. 1-8). IEEE.

[6] Khan W, Kumar T, Zhang C, Raj K, Roy AM, Luo B. SQL and NoSQL database software architecture performance analysis and assessments—a systematic literature review. Big Data and Cognitive Computing. 2023 May 12;7(2):97.

[7] Hong Z, Yuan Z, Zhang Q, Chen H, Dong J, Huang F, Huang X. Next-generation database interfaces: A survey of llm-based text-to-sql. arXiv preprint arXiv:2406.08426. 2024 Jun 12.

[8] Islam S. Future trends in SQL databases and big data analytics: Impact of machine learning and artificial intelligence. Available at SSRN 5064781. 2024 Aug 6.

[9] de Oliveira VF, Pessoa MA, Junqueira F, Miyagi PE. SQL and NoSQL Databases in the Context of Industry 4.0. Machines. 2021 Dec 27;10(1):20.

[10] Rockoff L. The language of SQL. Addison-Wesley Professional; 2021 Nov 4.

[11] Shivadekar S, Halem M, Yeah Y, Vibhute S. Edge AI cosmos blockchain distributed network for precise ablh detection. Multimedia tools and applications. 2024 Aug;83(27):69083-109.

[12] Fotache M, Munteanu A, Strîmbei C, Hrubaru I. Framework for the assessment of data masking performance penalties in SQL database servers. Case Study: Oracle. IEEE Access. 2023 Feb 22;11:18520-41.

[13] Panda SP. Augmented and Virtual Reality in Intelligent Systems. Available at SSRN. 2021 Apr 16.

[14] Karwin B. SQL Antipatterns, Volume 1: Avoiding the Pitfalls of Database Programming. The Pragmatic Programmers LLC; 2022 Oct 24.

[15] Nasereddin M, ALKhamaiseh A, Qasaimeh M, Al-Qassas R. A systematic review of detection and prevention techniques of SQL injection attacks. Information Security Journal: A Global Perspective. 2023 Jul 4;32(4):252-65.
[16] Chakraborty S, Aithal PS. CRUD Operation on WordPress Database Using C# SQL Client. International Journal of Case Studies in Business, IT, and Education (IJCSBE). 2023 Nov 28;7(4):138-49.
[17] Panda SP. The Evolution and Defense Against Social Engineering and Phishing Attacks. International Journal of Science and Research (IJSR). 2025 Jan 1.
[18] Choi H, Lee S, Jeong D. Forensic recovery of SQL server database: Practical approach. IEEE Access. 2021 Jan 18;9:14564-75.
[19] Thalji N, Raza A, Islam MS, Samee NA, Jamjoom MM. Ae-net: Novel autoencoder-based deep features for sql injection attack detection. IEEE access. 2023 Nov 28;11:135507-16.
[20] Chakraborty S, Paul S, Hasan KA. Performance comparison for data retrieval from nosql and sql databases: a case study for covid-19 genome sequence dataset. In2021 2nd International Conference on Robotics, electrical and signal processing techniques (ICREST) 2021 Jan 5 (pp. 324-328). IEEE.
[21] Crespo-Martínez IS, Campazas-Vega A, Guerrero-Higueras ÁM, Riego-DelCastillo V, Álvarez-Aparicio C, Fernández-Llamas C. SQL injection attack detection in network flow data. Computers & Security. 2023 Apr 1;127:103093.
[22] Antas J, Rocha Silva R, Bernardino J. Assessment of SQL and NoSQL systems to store and mine COVID-19 data. Computers. 2022 Feb 21;11(2):29.
[23] Shivadekar S, Kataria DB, Hundekar S, Wanjale K, Balpande VP, Suryawanshi R. Deep learning based image classification of lungs radiography for detecting covid-19 using a deep cnn and resnet 50. International Journal of Intelligent Systems and Applications in Engineering. 2023;11:241-50.
[24] Ashlam AA, Badii A, Stahl F. Multi-phase algorithmic framework to prevent SQL injection attacks using improved machine learning and deep learning to enhance database security in real-time. In2022 15th International Conference on Security of Information and Networks (SIN) 2022 Nov 11 (pp. 01-04). IEEE.
[25] Tanimura C. SQL for Data Analysis: Advanced Techniques for Transforming Data Into Insights. " O'Reilly Media, Inc."; 2021 Sep 9.
[26] Brunner U, Stockinger K. Valuenet: A natural language-to-sql system that learns from database information. In2021 IEEE 37th International Conference on Data Engineering (ICDE) 2021 Apr 19 (pp. 2177-2182). IEEE.
[27] Panda SP. Relational, NoSQL, and Artificial Intelligence-Integrated Database Architectures: Foundations, Cloud Platforms, and Regulatory-Compliant Systems. Deep Science Publishing; 2025 Jun 22.
[28] Lawson JG, Street DA. Detecting dirty data using SQL: Rigorous house insurance case. Journal of Accounting Education. 2021 Jun 1;55:100714.
[29] Zhang B, Ren R, Liu J, Jiang M, Ren J, Li J. SQLPsdem: A proxy-based mechanism towards detecting, locating and preventing second-order SQL injections. IEEE Transactions on Software Engineering. 2024 May 14;50(7):1807-26.
[30] Panda SP. Artificial Intelligence Across Borders: Transforming Industries Through Intelligent Innovation. Deep Science Publishing; 2025 Jun 6.

[31] Gandhi N, Patel J, Sisodiya R, Doshi N, Mishra S. A CNN-BiLSTM based approach for detection of SQL injection attacks. In2021 International conference on computational intelligence and knowledge economy (ICCIKE) 2021 Mar 17 (pp. 378-383). IEEE.
[32] Dhanaraj RK, Ramakrishnan V, Poongodi M, Krishnasamy L, Hamdi M, Kotecha K, Vijayakumar V. Random forest bagging and x-means clustered antipattern detection from SQL query log for accessing secure mobile data. Wireless communications and mobile computing. 2021;2021(1):2730246.
[33] Panda SP, Muppala M, Koneti SB. The Contribution of AI in Climate Modeling and Sustainable Decision-Making. Available at SSRN 5283619. 2025 Jun 1.
[34] Shivadekar S. Artificial Intelligence for Cognitive Systems: Deep Learning, Neuro-symbolic Integration, and Human-Centric Intelligence. Deep Science Publishing; 2025 Jun 30.
[35] Davidson L. Pro SQL Server Relational Database Design and Implementation: Best Practices for Scalability and Performance. Apress; 2021.
[36] Zhang W, Li Y, Li X, Shao M, Mi Y, Zhang H, Zhi G. Deep Neural Network-Based SQL Injection Detection Method. Security and Communication Networks. 2022;2022(1):4836289.
[37] Roy P, Kumar R, Rani P. SQL injection attack detection by machine learning classifier. In2022 International conference on applied artificial intelligence and computing (ICAAIC) 2022 May 9 (pp. 394-400). IEEE.
[38] Katsogiannis-Meimarakis G, Koutrika G. A survey on deep learning approaches for text-to-SQL. The VLDB Journal. 2023 Jul;32(4):905-36.