https://www.javatpoint.com/mysql-queries

========================================

http://www.mysqltutorial.org/basic-mysql-tutorial.aspx

https://www.geeksforgeeks.org/sql-interview-questions/

ELECT

column\_1, column\_2, ...

FROM

table\_1

[INNER | LEFT |RIGHT] JOIN table\_2 ON conditions

WHERE

conditions

GROUP BY column\_1

HAVING group\_conditions

ORDER BY column\_1

LIMIT offset, length;

SELECT followed by a list of comma-separated columns or an asterisk (\*) to indicate that you want to return all columns.

FROM specifies the table or view where you want to query the data.

JOIN gets related data from other tables based on specific join conditions.

WHERE clause filters row in the result set.

GROUP BY clause groups a set of rows into groups and applies aggregate functions on each group.

HAVING clause filters group based on groups defined by GROUP BY clause.

ORDER BY clause specifies a list of columns for sorting.

LIMIT constrains the number of returned rows.

SELECT DISTINCT

columns

FROM

table\_name

WHERE

where\_conditions;

ELECT column\_list

FROM t1

INNER JOIN t2 ON join\_condition1

INNER JOIN t3 ON join\_condition2

...

WHERE where\_conditions;

ELECT column\_list

FROM t1

INNER JOIN t2 ON join\_condition;

SELECT

productCode,

productName,

textDescription

FROM

products t1

INNER JOIN

productlines t2 ON t1.productline = t2.productline;

Because the joined columns of both tables have the same name productline, you can use the following syntax

ELECT

productCode,

productName,

textDescription

FROM

products

INNER JOIN

productlines USING (productline);

MySQL INNER JOIN using operator other than equal

SELECT

orderNumber,

productName,

msrp,

priceEach

FROM

products p

INNER JOIN

orderdetails o ON p.productcode = o.productcode

AND p.msrp > o.priceEach

WHERE

p.productcode = 'S10\_1678';

**30 Advanced SQL Interview Questions for Professionals (With Code)**

**Table of Contents**

1. Concept & Theoretical Questions
2. Questions Based on SQL Query & Codes
3. Tricky Questions

Unlock your SQL mastery and conquer those 2024 interview questions. Prepare with essential queries, from basic joints to advanced optimization to server and query questions. SQL up your skills, answer questions with ease, practice the codes and secure your dream data role. Let's crack the code on advanced queries, indexing, data security and advanced features with these 30 must-know questions that are sure to come up in your next interview.

**Start with 30 advanced and technical SQL interview questions to master the upcoming technical interview rounds.**

**1. Explain the difference between a correlated subquery and a nested subquery.**

* **Answer:** A correlated subquery references data from the outer query in its WHERE clause. A nested subquery can be placed anywhere in the outer query and doesn't directly reference the outer table.

**2. Write a query to find the top 5 customers with the highest total order amounts.**

* **Answer:**

**SQL Code**

SELECT CustomerID, SUM(OrderAmount) AS TotalOrderAmount  
FROM Orders  
GROUP BY CustomerID  
ORDER BY TotalOrderAmount DESC  
LIMIT 5;

**3. How can you optimize a slow-running query?**

* **Answer:** There are several techniques, including:
  + Using appropriate indexes
  + Avoiding unnecessary joins and subqueries
  + Using efficient functions and operators
  + Analyzing execution plans to identify bottlenecks

**4. Explain the concept of normalization in database design.**

* **Answer:** Normalization is the process of organizing tables to minimize redundancy and improve data integrity. It involves breaking down tables into smaller, more focused tables with defined relationships.

**5. Describe the purpose of window functions in SQL.**

* **Answer:** Window functions perform calculations on a set of rows within a query partition. Examples include ranking, calculating moving averages, and finding percentiles.

**6. Write a query to find the average salary for each department, excluding employees with salaries above a certain threshold.**

* **Answer:**

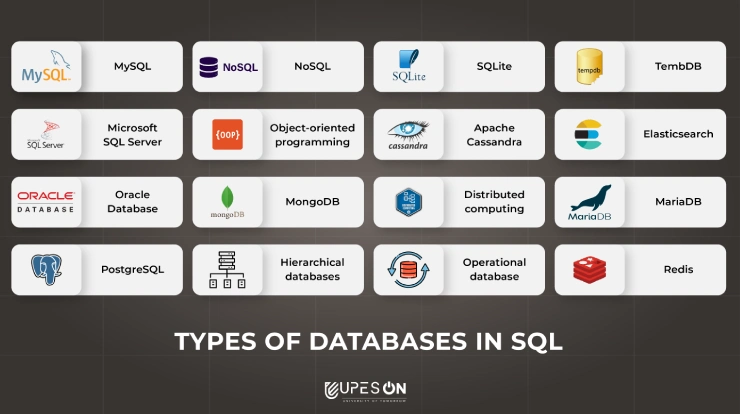
**SQL Code**

SELECT Department, AVG(Salary) AS AverageSalary  
FROM (  
  SELECT Department, Salary  
  FROM Employees  
  WHERE Salary <= (SELECT MAX(Salary) FROM Employees) / 2  
) AS Subquery  
GROUP BY Department;

**7. How can you handle missing values (NULL) in your queries?**

* **Answer:** You can use functions like ISNULL, COALESCE, or CASE statements to handle missing values. These functions allow you to specify alternative values or perform different operations based on NULL checks.

**8. Explain the concept of materialized views and their benefits.**

* **Answer:** Materialized views are pre-computed copies of database queries stored as tables. They improve query performance by providing pre-aggregated or filtered data, reducing processing needs for repeated queries.

**9. Describe the concept of transactions in SQL and their ACID properties.**

* **Answer:** A transaction is a unit of work that maintains data consistency. ACID stands for Atomicity (all or nothing), Consistency (data integrity), Isolation (concurrent transactions don't interfere), and Durability (changes persist).

**10. How can you secure your SQL queries against SQL injection attacks?**

* **Answer:** Use prepared statements with parameterized queries. These statements separate code from data, preventing malicious code execution.

**11. Explain the difference between UNION and UNION ALL in SQL.**

* **Answer:** UNION removes duplicate rows from the combined result set, while UNION ALL includes all rows, even duplicates, from the selected queries.

**12. Describe the concept of common table expressions (CTEs) and their benefits.**

* **Answer:** CTEs are temporary named result sets defined within a query. They improve readability and modularity by allowing complex logic to be pre-defined and reused within the main query.

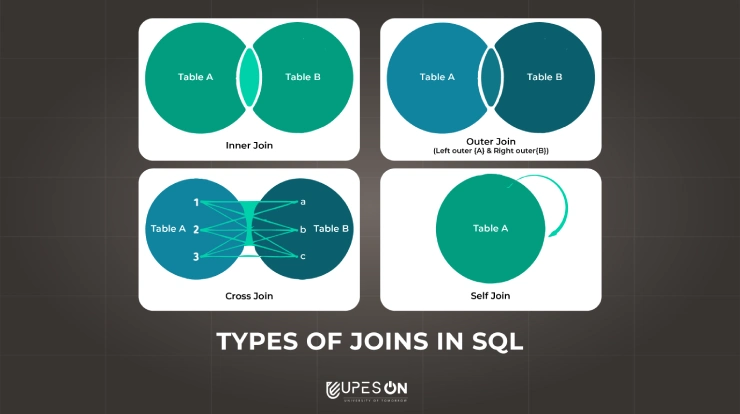
**13. Explain the concept of temporal tables and how they can be used for historical data tracking.**

* **Answer:** Temporal tables store data with historical versions, allowing you to track changes over time. You can use them to analyze historical trends, audit data modifications, or revert to previous states.

**14. Describe the functionality of triggers in SQL and their different types.**

* **Answer:** Triggers are stored procedures that automatically execute in response to specific events on a table (like INSERT, UPDATE, DELETE). They can be used for data validation, enforcing business logic, or maintaining data consistency.

**15. Explain the difference between full outer joins and full joins in SQL.**

* **Answer:** Both join types return all rows from both tables, but a full outer join preserves NULL values in unmatched columns, while a full join may replace them with default values depending on the database engine.

**16. Describe the purpose of regular expressions in SQL and how they can be used in queries.**

* **Answer:** Regular expressions are patterns used to search and manipulate text data within SQL queries. They allow for complex string matching and manipulation, enabling tasks like data validation or extracting specific information.

**17. Explain the concept of database partitioning and its benefits.**

* **Answer:** Database partitioning divides a large table into smaller, manageable segments based on a chosen key. This improves performance by allowing queries to target specific partitions and reducing I/O operations.

**18. Describe the functionality of MERGE statements in SQL.**

* **Answer:** MERGE statements combine INSERT, UPDATE, and DELETE operations into a single statement. They offer efficient data manipulation, allowing you to perform conditional actions based on existence or matching criteria.

**SQL Interview Questions on Query**

**19. Write a query to find the difference in days between the order date and the ship date for each order.**

* **Answer:**

**SQL Code:**

SELECT OrderID, DATEDIFF(day, OrderDate, ShipDate) AS DaysDiff  
FROM Orders;

**20. Write a query to find the manager for each employee in a company, even if the employee doesn't have a manager assigned.**

* **Answer:**

**SQL Code:**

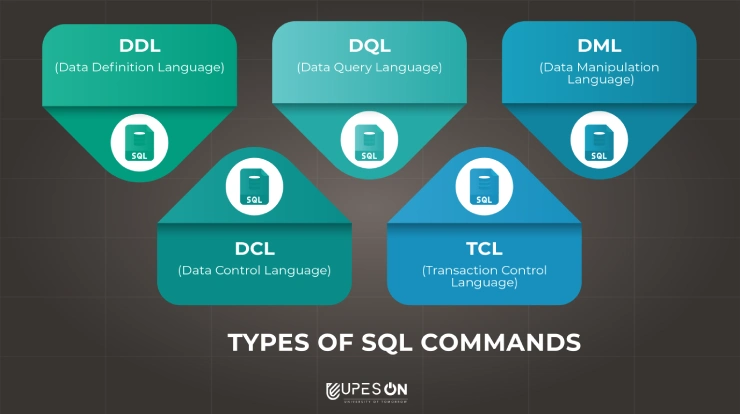
SELECT e.EmployeeID, m.ManagerName  
FROM Employees e  
LEFT JOIN Employees m ON e.ManagerID = m.EmployeeID;

**21. Write a query to pivot data from rows to columns, showing the total sales for each product category by month.**

* **Answer:**

**SQL Code:**

SELECT Month,  
       SUM(CASE WHEN ProductCategory = 'Electronics' THEN Sales ELSE 0 END) AS Electronics,  
       SUM(CASE WHEN ProductCategory = 'Clothing' THEN Sales ELSE 0 END) AS Clothing,  
       ... (add more categories)  
FROM SalesData  
GROUP BY Month;



**22. Write a query to find employees who have never placed an order.**

* **Answer:**

**SQL Code:**

SELECT e.EmployeeID, e.EmployeeName  
FROM Employees e  
LEFT JOIN Orders o ON e.EmployeeID = o.CustomerID  
WHERE o.CustomerID IS

**23. Write a query to find the department with the highest average salary for employees who have been with the company for more than 2 years.**

* **Answer:**

**SQL Code:**

SELECT d.DepartmentName, AVG(e.Salary) AS AverageSalary  
FROM Employees e  
INNER JOIN Departments d ON e.DepartmentID = d.DepartmentID  
WHERE e.HireDate < DATEADD(year, -2, GETDATE())  
GROUP BY d.DepartmentName  
ORDER BY AverageSalary DESC  
LIMIT 1;

**24. Write a query to find the nth highest salary in an employee table.**

* **Answer (using a subquery):**

**SQL Code:**

SELECT Salary  
FROM Employees  
WHERE Salary IN (  
  SELECT TOP 1 Salary  
  FROM (  
    SELECT Salary, ROW\_NUMBER() OVER (ORDER BY Salary DESC) AS RowNum  
    FROM Employees  
  ) AS Subquery  
  WHERE RowNum = n  
);

**25. Write a query to find the total number of customers who placed orders in each quarter of the last year.**

* **Answer:**

**SQL Code:**

SELECT DATEPART(quarter, OrderDate) AS Quarter, COUNT(DISTINCT CustomerID) AS Customers  
FROM Orders  
WHERE OrderDate >= DATEADD(year, -1, GETDATE())  
GROUP BY DATEPART(quarter, OrderDate)  
ORDER BY Quarter;

**26. Write a query to find the manager hierarchy for a specific employee, showing all levels up to the CEO.**

* **Answer (using a recursive CTE)**

**SQL Code:**

WITH ManagerHierarchy (EmployeeID, ManagerID, Level) AS (  
  SELECT EmployeeID, ManagerID, 1 AS Level  
  FROM Employees  
  WHERE EmployeeID = <employee\_id>  
  UNION ALL  
  SELECT e.EmployeeID, m.ManagerID, h.Level + 1  
  FROM Employees e  
  INNER JOIN ManagerHierarchy h ON e.EmployeeID = h.ManagerID  
  INNER JOIN Employees m ON e.ManagerID = m.EmployeeID  
  WHERE m.ManagerID IS NOT NULL  
)  
SELECT EmployeeID, ManagerID, Level  
FROM ManagerHierarchy  
ORDER BY Level DESC;

**27. Write a query to find the product categories with the highest and lowest total sales for the previous year.**

* **Answer:**

**SQL Code:**

SELECT ProductCategory, SUM(SalesAmount) AS TotalSales  
FROM SalesData  
WHERE SaleDate >= DATEADD(year, -1, GETDATE())  
GROUP BY ProductCategory  
ORDER BY TotalSales DESC, TotalSales ASC  
LIMIT 2;

**28. Write a query to find employees who earn more than the average salary in their department.**

* **Answer:**

**SQL Code:**

SELECT e.EmployeeID, e.EmployeeName, d.DepartmentName, e.Salary  
FROM Employees e  
INNER JOIN Departments d ON e.DepartmentID =

**Tricky Interview Questions to confuse Candidates**

Designed to challenge the solver in you, these questions will assess your problem-solving skills. Use the table below for refrence and answer the questions accordingly.

* **Data Setup:** Use the sample tables for these questions.
* **Customers Table:**

|  |  |  |
| --- | --- | --- |
| **Customer ID** | **Name** | **City** |
| 1 | John Levi | New York |
| 2 | Jane Tye | Los Angeles |
| 3 | Mike Foley | Chicago |
| 4 | Alice White | New York |

* **Orders Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Order ID** | **Customer ID** | **Order Date** | **Order Total** |
| 100 | 1 | 2023-07-01 | 100.00 |
| 101 | 2 | 2023-06-15 | 50.00 |
| 102 | 3 | 2023-07-05 | 150.00 |
| 103 | 1 | 2023-07-07 | 75.00 |
| 104 | 4 | 2023-07-02 | 200.00 |

**Questions:**

**1. Find the total number of orders placed by each customer, excluding orders placed in June.**

* **Tricky Aspect:**Excluding a specific month required filtering based on the date.
* **Answer**:

**SQL Code:**

SELECT c.name, COUNT(\*) AS num\_orders  
FROM Customers c  
INNER JOIN Orders o ON c.customer\_id = o.customer\_id  
WHERE MONTH(order\_date) <> 6  
GROUP BY c.name

**2. Find the customer who has placed the highest total order value.**

* **Tricky aspect:** Required aggregation and sorting based on the total order value.
* **Answer:**

**SQL Code:**

SELECT c.name, SUM(order\_total) AS total\_order\_value  
FROM Customers c  
INNER JOIN Orders o ON c.customer\_id = o.customer\_id  
GROUP BY c.name  
ORDER BY total\_order\_value DESC  
LIMIT 1;

**3. List all orders placed on specific dates (eg., 2023-07-04 and 2023-07-06) and their corresponding customer names.**

* **Tricky Aspect:**Requires filtering based on multiple specific dates.
* **Answer:**

**SQL Code:**

SELECT c.name, o.order\_date, o.order\_total  
FROM Customers c  
INNER JOIN Orders o ON c.customer\_id = o.customer\_id  
WHERE order\_date IN ('2023-07-04', '2023-07-06');

**4. Find the average order value for each city.**

* **Tricky Aspect:**Requires joining tables and grouping by city.
* **Answer:**

**SQL Code:**

SELECT c.city, AVG(o.order\_total) AS avg\_order\_value  
FROM Customers c  
INNER JOIN Orders o ON c.customer\_id = o.customer\_id  
GROUP BY c.city

**5. Identify customers who haven't placed any orders.**

* **Tricky Aspect:**Requires using LEFT JOIN and filtering for null values.
* **Answer:**

**SQL Code:**

SELECT c.name  
FROM Customers c  
LEFT JOIN Orders o ON c.customer\_id = o.customer\_id  
WHERE o.order\_id IS NULL;

**6. Find the month with the highest total order value.**

* **Tricky Aspect:** Requires extracting the month from the date and grouping by month
* **Answer:**

**SQL Code:**

SELECT MONTH(order\_date) AS order\_month, SUM(order\_total) AS total\_order\_value  
FROM Orders  
GROUP BY MONTH(order\_date)  
ORDER BY total\_order\_value DESC  
LIMIT 1;

**7. Write a query to display the top 2 customers with the most orders in the last 30 days.**

* **Tricky Aspect:**Requires filtering by date range and using window functions for ranking.
* **Answer:**

**SQL Code:**

SELECT c.name, COUNT(\*) AS num\_orders  
FROM Customers c  
INNER JOIN Orders o ON c.customer\_id = o.customer\_id  
WHERE order\_date >= DATE\_SUB (CURDATE)

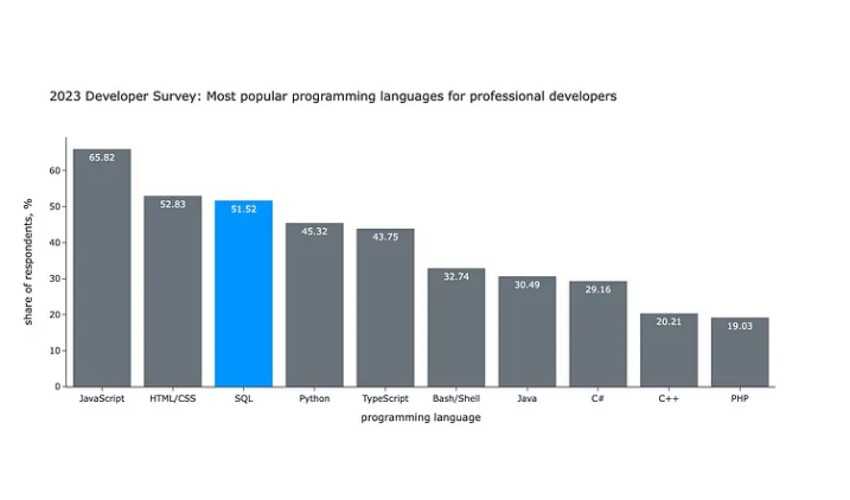
**Complex SQL Interview Questions for Interview Preparation**

Last updated by Swaminathan Iyer on Dec 22, 2024 at 11:06 PM

| Reading Time: 10 minutes

**Contents**

Complex SQL Interview Questions: According to the leading statistics, SQL is one of the most popular languages used by professional developers. It has been found that after JavaScript and Python, SQL is used by 51.52% of developers.



For an SQL developer hoping to bag their dream job, mastering complex SQL queries is a must for cracking that SQL interview. In this competitive space, the importance of knowing the kind of complex [**SQL interview questions for experienced professionals**](https://interviewkickstart.com/blogs/interview-questions/sql-interview-questions-for-experienced-professionals) you can expect can’t be overstated. This article offers sample complex SQL interview questions to help you assess and direct your SQL interview preparation.

In this article, we have presented some of the complex SQL interview questions for interview preparation for experienced developers and some that you can follow to practice your answers.

**Complex SQL Interview Questions and Answers for Experienced Developers**

Here are some sample complex SQL interview questions and answers to help you with your complex SQL preparation:

**Q1. Write a Query to Get the Last Record from a Table**

To get the last record from a table, you can use the `ORDER BY` clause in combination with the `LIMIT` clause (or its equivalent in different SQL dialects). Here’s how you can do it:

**For SQL databases like MySQL, PostgreSQL, and SQLite:**

|  |
| --- |
| SELECT \* FROM Student ORDER BY RowID DESC LIMIT 1; |

**For Oracle:**

|  |
| --- |
| SELECT \* FROM ( SELECT \* FROM Student ORDER BY RowID DESC ) WHERE ROWNUM = 1; |

**For SQL Server:**

|  |
| --- |
| SELECT TOP 1 \* FROM Student ORDER BY RowID DESC; |

These queries will fetch the last record from the `Student` table based on the `RowID` column.

**Q2. Write a Query to Get the First Record from a Table?**

The first record can be fetched in two ways, one similar to the last record case, but with use of min:

First way: select \* from Student where RowID = select min(RowID) from Student;

The other method is by printing just one (first) row of the table:

select \* from Student where Rownum = 1;

**Q3. Write a Query to Display the First Ten Records from a Table.**

Consider the table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Student ID | First Name | Last Name | Age | Grade |
| 1 | John | Doe | 20 | A |
| 2 | Jane | Smith | 21 | B |
| 3 | Michael | Johnson | 22 | A |
| 4 | Emily | Davis | 19 | B |
| 5 | David | Wilson | 23 | C |
| 6 | Sarah | Brown | 20 | A |
| 7 | Chris | Taylor | 21 | B |
| 8 | Linda | More | 22 | A |
| 9 | James | Clark | 20 | B |
| 10 | Olivia | Lewis | 19 | A |
| 11 | Robert | Walker | 22 | B |
| 12 | Jessica | Hall | 21 | A |

To list out only the top 10 rows use this code using ROWNUM:

SELECT \* FROM Student WHERE ROWNUM <= 10;

Using LIMIT:

SELECT \* FROM Customers LIMIT 10;

**Q4. Create a Table with the Same Structure with Data as in the Student Table.**

To create a table with the same structure and data as the `**Student**` table, you can use the `**CREATE TABLE AS SELECT**` statement in SQL. Here’s how you can do it:

|  |
| --- |
| CREATE TABLE Student\_Copy AS SELECT \* FROM Student; |

This query will create a new table called `**Student\_Copy**` with the same structure and data as the `Student` table. If you only want to copy the structure without the data, you can add a `**WHERE**` clause that evaluates to false:

|  |
| --- |
| CREATE TABLE Student\_Copy AS SELECT \* FROM Student WHERE 1=0; |

This will create the `**Student\_Copy**` table with the same structure but without copying any data from the original `**Student**` table.

**Q5. Show Only Common Records Between Two Tables.**

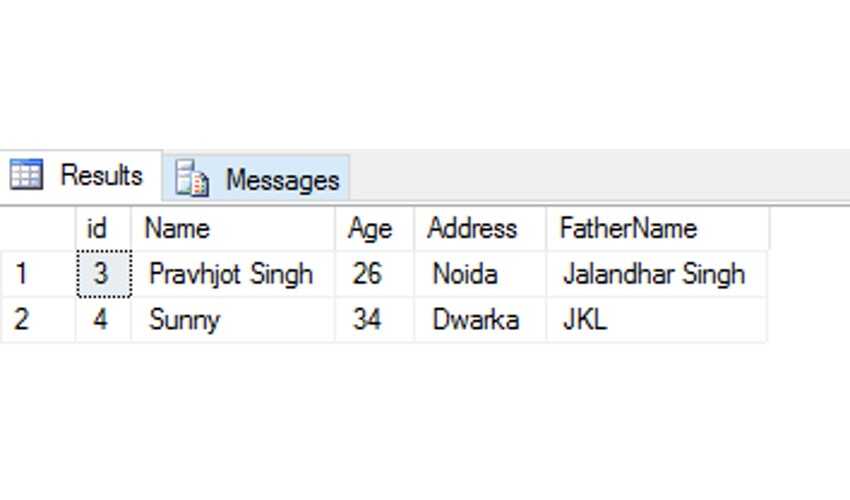
Here let’s consider 2 tables: student and student1.

Table 1: student

Select \* from student



Table 2: student1

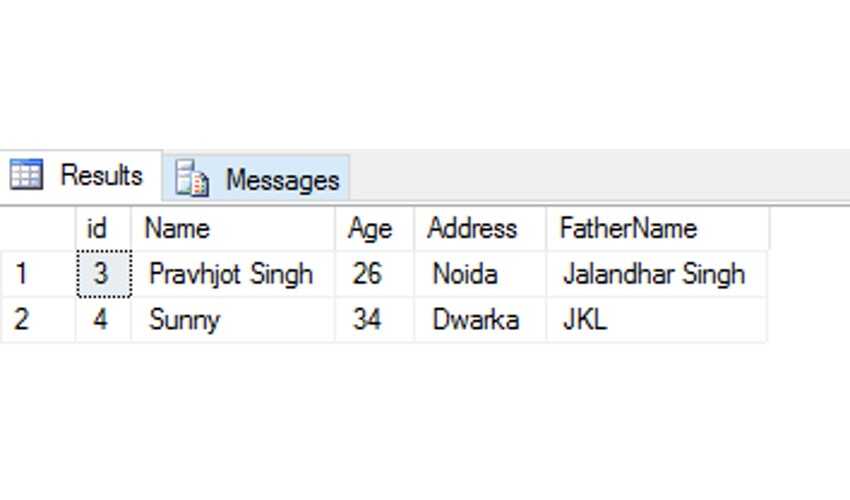


Select \* from student1

Final Query:

(Select \* from student) Intersect (Select \* from student1)

Final Output: Here in the output table we can see only the common rows from both tables being displayed.



**Complex SQL Interview Questions for Practice**

Now, check out these frequently asked complex SQL interview questions to gauge your interview preparation:

**Complex PL SQL Server Interview Questions**

These complex PL SQL server questions regularly feature at tech interviews for SQL developers:

6. Define and describe the usage of a linked server.

7. Name and explain the different types of Joins.

8. Explain the different types of authentication modes.

9. Which stored procedure would you run when adding a linked server?

10. Where do you think the user names and passwords will be stored in the SQL server?

11. How would you add email validation using only one query?

**Complex SQL Queries Interview Questions**

Interview questions related to complex SQL queries are quite common at interviews for developers. These involve questions like “what SQL query would you use to”:

12. Fetch the number of Weekends in the current month

13. Get the last 5 Records from the Student table

14. Get the common records present in two different tables that have no joining conditions.

15. Display records 5 to 9 from the Employee table.

16. Display the last record of a table

17. Display the third-last record of a table

18. Convert seconds into time format

19. Remove duplicate rows from a table

20. Find the number of duplicate rows

21. Find the fourth-highest score in the Students table using self-join

22. Show the max and min salary together from the Employees table

23. Display date in a DD-MM-YYYY table

24. Create Employee\_C table, which is the exact replica of the Employee table

25. Drop all user tables from Oracle

26. Calculate the number of rows in a table without using count

27. Find repeated characters from your name

28. Display department and month-wise maximum salary

29. Find the second-highest salary in the Employee table.

30. Select all the records from the Student table, where the names are either Anu or Dan.

31. Select all the records from the Student table where the name is not Anu and Dan.

32. Get Nth Record from the Student table.

33. Get the 3 Highest salaries records from the Student table

34. Show Odd rows in the Student table

35. Show Even rows in the Student table

36. Get the DDL of a table

37. Get all the records from Employees who have joined in the year 2020.

38. Find the maximum salary of each department.

39. Find all Employees with their managers.

40. Display the name of employees who joined in 2020 and have a salary is greater than 50000.

41. Get the first 5 Records from the Student table.

42. Get information of Employees where Employee is not assigned to any department.

43. Show 1 to 100 Numbers

44. Find duplicate rows in a table

45. Get the previous month’s last day.

46. Display a string vertically.

47. The marks column in the Student table contains comma-separated values. How would you calculate the number of those comma-separated values?

48. Get the 3rd highest salary using Rank Function.

49. Create a table with its structure the same as the structure of the Student table.

50. Display first 25% records from the Student table

51. Display last 25% records from the Student table

52. Create a table with the same structure and data as the Student table

53. Get only the common records between two tables

54. Get unique records from the table without using distinct keywords.

55. Find the admission date of the Student in YYYY-DAY-Date format.

56. Convert the System time into seconds.

57. Display monthly Salary of Employee given annual salary.

58. Get the first record from the Student table

59. Get the last record from the Student table

**Ready to Nail Your Next Coding Interview?**

Whether you’re a coding engineer gunning for a software developer or software engineer role, a tech lead, or you’re targeting management positions at top companies, IK offers [**Backend Engineering course**](https://interviewkickstart.com/courses/back-end-engineering-interview-masterclass) specifically designed for your needs to help you with your technical interview preparation!

As pioneers in the field of technical interview preparation, we have trained [**thousands of software engineers to crack the toughest coding interviews**](https://interviewkickstart.com/reviews) and land jobs at their dream companies, such as Google, Facebook, Apple, Netflix, Amazon, and more!

**FAQs on Complex SQL Interview Questions**

**Q1. What is a Complex SQL Query?**

Complex SQL queries are parameter queries that go beyond the standard SQL usage of SELECT and WHERE and use two or more parameters. They also often heavily use AND and OR clauses. Complex queries are helpful because we can make more precise and accurate database searches with them.

**Q2. How Many Types of SQL Are There?**

There are five types of SQL commands on the basis of the functionalities performed by them: DDL(Data Definition Language), DQL(Data Query Language), DCL(Data Control Language), DML(Data Manipulation Language) and TCL(Transaction Control Language).

**Q3. Give a Comparison Between PostgreSQL and MongoDB.**

PostgreSQL is a SQL database that uses tables with organized rows and columns for storing data. It is compatible with notions like JOINS and referential integrity entity-relationship. PostgreSQL uses SQL as the query language.

On the other hand, MongoDB is a NoSQL database. It is capable of storing raw data because a schema is not necessary. Data is kept in BSON documents, and the user can modify the document’s structure. The query language used by MongoDB is JavaScript.

**Q4. What is the Difference Between Simple and Complex Views in SQL?**

Simple views in SQL can contain only one base table. On the other hand, complex views in SQL have more than one base table. Complex views can also have a group by clause, join conditions, and order by clause.

**Q5. What is a Unique Key In SQL?**

The collection of data or columns in a table that allows us to recognize records distinctively is known as a unique key in SQL. The unique key ensures that the columns in the database are all unique. It is equivalent to the primary key; however, it may accept a null value compared to it.

====================================================================================

60 MySQL Interview Questions and Answers Every Developer Should Know

[By Sachin Satish](https://www.simplilearn.com/authors/sachin-satish)

Share This Article:

Last updated on Nov 12, 202420087

[MySQL](https://www.simplilearn.com/tutorials/mysql-tutorial) is the most popular open-source relational database management system (RDBMS), typically used with [PHP](https://www.simplilearn.com/tutorials/php-tutorial/what-is-php). It is fast, reliable, and easy to run on the web and the server. MySQL is the world's most popular open-source [database software](https://www.simplilearn.com/tutorials/sql-tutorial/create-mysql-database) and a preferred choice for critical business applications by giants like Yahoo, Suzuki, and NASA. Naturally, there are a lot of career opportunities for MySQL experts. This article is a Q/A guide on how to answer MySQL interview questions.

MySQL uses standard [SQL programming](https://www.simplilearn.com/tutorials/sql-tutorial/what-is-sql) for the creation, modification, and extraction of data from a relational database. The data is stored in tables consisting of rows and columns. Users can interact directly with MySQL or use it to implement applications that need relational database capability. MySQL jobs range from MySQL Developer, [MySQL Database Administrator](https://www.simplilearn.com/how-to-launch-career-in-database-administration-article), MySQL Database Engineer, and more.

Here are some of the most frequently asked MySQL interview questions and how to answer them.

Basic MySQL Interview Questions

1. What is MySQL?

MySQL is a relational database management system based on SQL (Structured Query Language). It is an open source software owned by Oracle and can run on various platforms. Most websites or web applications are developed using MySQL.

2. In which language has MySQL been written?

MySQL is written in [C and C++](https://www.simplilearn.com/tutorials/cpp-tutorial/difference-between-c-and-cpp). Its SQL parser is written in yacc.

3. What are the advantages of using MySQL?

MySQL is a fast, stable, and reliable solution that provides advantages like:

* Data Security – most secure and reliable [database management](https://www.simplilearn.com/what-is-database-management-article) system
* Flexibility – runs on all operating systems; features 24X7 support and enterprise indemnification
* High Performance – powerful, designed to meet highly demanding applications while maintaining optimum speed and high performance
* On-demand Scalability – offers on-demand scalability and complete customization
* Enterprise-level SQL Features – the enterprise edition includes advanced features and management tools, and technical support for enterprise
* Full-text Indexing and Searching – has support for full-text indexing and searching
* Query Caching – unique memory caches help enhance the speed of MySQL greatly
* Replication – one MySQL server can be duplicated on another, resulting in numerous benefits

4. What is a database?

A database is a structured repository of data stored electronically in a computer system and organized in a way that data can be quickly searched and information rapidly retrieved. A database is generally controlled by a database management system.

5. What does 'MySQL' stand for?

'My' in MySQL represents the first name of its co-founder, Michael Widenius' daughter, My Widenius. SQL is an abbreviation for the term "Structured Query Language". SQL is also used in databases like Oracle and Microsoft SQL Server.

6. How to check MySQL version?

The command 'MySQL-v' can be used to check MySQL version on Linux

7. What does a MySQL database contain?

A MySQL database contains one or many tables, with each table containing several records or rows. Within these rows, data is contained in various columns or fields.

8. List the ways to interact with MySQL.

There are 3 main ways users can interact with MySQL:

* Using a command line
* Through a web interface
* Using a [programming language](https://www.simplilearn.com/tutorials/programming-tutorial/first-programming-language)

9. What are the different tables in MySQL?

They are:

* MyISAM
* HeapMerge
* INNO DB
* ISAM

10. What are MySQL Database Queries?

A query is a request for data or information from a database. Users can query a database for specific information, and the resultant record/records are returned by MySQL.

Become a Software Development Professional

**Full Stack Java Developer Masters Program**

* + Kickstart Full Stack Java Developer career with industry-aligned curriculum by experts
  + Hands-on practice through 20+ projects, assessments, and tests

7 months

[**View Program**](javascript:void(0))

**Full Stack Developer - MERN Stack Masters Program**

* + 40+ micro skilling exercises & 6+ work-like professional projects
  + Develop expertise in 10+ full stack development tools and frameworks

6 months

[**View Program**](javascript:void(0))

Here's what learners are saying regarding our programs:

* Jonathan Mabiala

Jonathan Mabiala

Java Software Developer, Desjardins

I chose to upskill after moving from the United States to Canada in 2019. My journey in programming began during engineering, but I lost touch after college. Realizing I needed certification to advance, I enrolled in Simplilearn. The live classes boosted my confidence, allowing me to transition to a Java Software Developer role.

* Mayur Kharad

Mayur Kharad

Product Engineer, IKS Health

During the lockdown, I realized I needed to upskill myself, and my journey with Simplilearn has been fantastic. I learned many things during the full stack java developer course, thanks to trainer Virendra Sharma. I've always wanted to work in this sector, and after completing my certification in Fullstack Java Development, I got placed at IKS Health through Simplilearn.

Not sure what you’re looking for?[View all Related Programs](https://www.simplilearn.com/mobile-and-software-development?source=InpageBannerCategory)

11. What are some common MySQL commands?

Some common MySQL commands are:

* CREATE – To create Table
* INSERT – To insert data
* JOIN – To join tables
* DELETE – To delete a row from a table
* ALTER – To alter database or table
* BACKUP – to back up a table
* DROP – To delete a database or table
* CREATE INDEX – to add indexing to a column in a table
* GRANT – To change user privileges
* TRUNCATE – Empty a table
* EXIT – to exit

12. How to create a database in MySQL?

The CREATE DATABASE command can be used to create a new database.

13. How to create table using MySQL?

The following query can be used to create a table:

CREATE TABLE 'history' (

'author' VARCHAR(128),

'title' VARCHAR(128),

'type' VARCHAR(16),

'year' CHAR(4))

ENGINE = InnoDB;

A table "history" gets created in the selected database.

14. How to insert data in MySQL?

The INSERT INTO statement is used to insert new records in a table in MySQL.

The two main syntaxes are:

INSERT INTO table\_name (column 1, column 2, column 3,…columnN)

VALUES (value 1, value 2, value 3,...valueN)

15. How do you remove a column form a database?

The DROP command is used to remove a column from a database.

Alter table 'history' drop column title;

16. How to create an index?

There are different types of indexes in MySQL, like a regular INDEX, a [PRIMARY KEY,](https://www.simplilearn.com/tutorials/sql-tutorial/primary-key-in-sql) or a FULLTEXT index. Indexes are created on a column basis. Indexing helps to quickly search for results, either by ordering the data on disk or by telling the SQL engine which location to find your data in.

Syntax:

ALTER TABLE history ADD INDEX(author(10));

17. How do you delete data from MySQL table?

We use the DELETE statement to remove records from a table.

The syntax is as follows:

DELETE FROM table\_name WHERE column\_name

18. How can you view a database in MySQL?

The SHOW DATABASES command allows the user to view all databases on the MySQL server host.

mysql> SHOW DATABASES;

19. How to import database in MySQL?

There are two ways to import database or move data from one place to another:

* Command Line Tool
* MySQL Workbench

20. What are numeric data types in MySQL?

There are numeric data types for integer, fixed-point, floating-point, and bit values in MySQL. Except for BIT, the other numeric data types can be signed or unsigned.

Examples:

INT - Standard Integer

TINYINT - Very Small Integer

SMALLINT - Small Integer

MEDIUMINT - Medium-sized Integer

BIGINT - Large Integer

DECIMAL - Fixed-point number

FLOAT - Single-precision floating-point number

DOUBLE - Double-precision floating-point number

BIT - Bit-field

21. What are string data types in MySQL?

The string [data types in MySQL](https://www.simplilearn.com/tutorials/sql-tutorial/sql-data-types) are:

* CHAR
* VARCHAR
* BINARY
* VARBINARY
* TINYBLOB
* BLOB
* MEDIUMBLOB
* LONGBLOB
* TINYTEXT
* TEXT
* MEDIUMTEXT
* LONGTEXT
* ENUM
* SET
* NULL

22. What are temporal data types in MySQL?

MySQL provides temporal data types for date and time, as well as a combination of date and time. These are:

DATE - A date value in CCYY-MM-DD Format

TIME - A Time value in hh : mm :ss format

DATETIME - Date and time value in CCYY-MM-DD hh : mm :ss format

TIMESTAMP - A timestamp value in CCYY-MM-DD hh : mm :ss format

YEAR - A year value in CCYY or YY format

23. What is BLOB?

BLOB is an acronym for a binary large object. It is a string data type used to hold a variable amount of data.

24. How do you add users in MySQL?

The CREATE command, along with necessary credentials, can be used to add users.

CREATE USER 'testuser' IDENTIFIED BY 'sample password';

Intermediate MySQL Interview Questions

25. What are views in MySQL?

A view is a set of rows returned when a particular query is executed in MySQL. It is also known as a virtual table, which does not store any data of its own but displays data stored in other tables.

26. How to create and execute views?

The CREATE VIEW command is used to create a view in MySQL.

The syntax is:

CREATE VIEW [databasename.] view\_name [(column\_list)] AS select-statement;

27. What are MySQL triggers?

A task that is executed automatically in response to a predefined database event is known as a trigger. Each trigger is associated with a table and is activated by commands like INSERT, DELETE, or UPDATE.

28. How many triggers are possible in MySQL?

There are 6 different types of triggers in MySQL:

* Before Insert
* After Insert
* Before Update
* After Update
* Before Delete
* After Delete

29. What is MySQL server?

The server 'mysqld' is the MySQL server, which performs all manipulation of databases and tables.

30. What are the clients and utilities in MySQL?

There are several MSQL programs available to help users communicate with the server. Some important ones for administrative tasks are:

.mysql – this interactive program helps to send [SQL statements](https://www.simplilearn.com/tutorials/sql-tutorial/sql-case-statement) to the server and view the results. One can even use MySQL to use batch scripts.

.mysqladmin – this administrative program helps perform tasks like shutting down the server, checking configuration, monitoring status if it is not functioning properly.

.mysqldump – for backing up databases or copying them to another server

.mysqlcheck and myisamchk – these programs help perform table checking, analysis, and optimization, plus repairs for damaged tables.

31. What types of relationships are used in MySQL?

Three types of relationships are used in MySQL:

* One-to-one – items with one-to-one relation can be included as columns in the same table
* One-to-many – or many-to-one relationships are seen when one row in a table is related to multiple rows in another table
* Many-to-many – many rows in a table are linked to many rows in another table

Become a Software Development Professional

**Full Stack Java Developer Masters Program**

* + Kickstart Full Stack Java Developer career with industry-aligned curriculum by experts
  + Hands-on practice through 20+ projects, assessments, and tests

7 months

[**View Program**](javascript:void(0))

**Full Stack Developer - MERN Stack Masters Program**

* + 40+ micro skilling exercises & 6+ work-like professional projects
  + Develop expertise in 10+ full stack development tools and frameworks

6 months

[**View Program**](javascript:void(0))

Here's what learners are saying regarding our programs:

* Jonathan Mabiala

Jonathan Mabiala

Java Software Developer, Desjardins

I chose to upskill after moving from the United States to Canada in 2019. My journey in programming began during engineering, but I lost touch after college. Realizing I needed certification to advance, I enrolled in Simplilearn. The live classes boosted my confidence, allowing me to transition to a Java Software Developer role.

* Mayur Kharad

Mayur Kharad

Product Engineer, IKS Health

During the lockdown, I realized I needed to upskill myself, and my journey with Simplilearn has been fantastic. I learned many things during the full stack java developer course, thanks to trainer Virendra Sharma. I've always wanted to work in this sector, and after completing my certification in Fullstack Java Development, I got placed at IKS Health through Simplilearn.

Not sure what you’re looking for?[View all Related Programs](https://www.simplilearn.com/mobile-and-software-development?source=InpageBannerCategory)

Advanced MySQL Interview Questions

32. Explain the logical architecture of MySQL

The top layer comprises the services required by most network-based client/server tools like connection handling, security, authentication, etc.

The 2nd layer comprises code for query parsing, optimization, analysis, caching, and all built-in functions.

The 3rd layer comprises storage engines where storage and retrieval of data stored in MySQL is performed.

33. What is Scaling?

Scaling capacity in MySQL is the ability to handle the load in terms of:

* Data quantity
* Number of users
* User activity
* Size of related datasets

34. What is Sharding?

The process of breaking up large tables into smaller chunks or shards spread across many servers is called sharding. It makes querying, maintenance, and other tasks faster.

35. What are Transaction Storage Engines?

The InnoDB storage engine enables users to use the transaction facility of MySQL.

We hope this list of MySQL interview questions will be helpful for you. Register with Simplilearn today to get access to top-rated courses on [database training](https://www.simplilearn.com/sql) and [full stack web development.](https://www.simplilearn.com/java-full-stack-developer-certification)

36. How does MySQL differ from PostgreSQL?

MySQL and PostgreSQL are both popular relational database management systems (RDBMS) but have differences in features, performance, and syntax. MySQL is known for its speed and ease of use, while PostgreSQL is praised for its advanced features, including support for complex data types, transactions, and advanced indexing.

37. Can you explain the difference between MyISAM and InnoDB storage engines?

MyISAM is a storage engine in MySQL known for its simplicity and speed, but lacks support for transactions and foreign keys. InnoDB, on the other hand, is a more robust storage engine that supports transactions, foreign keys, and row-level locking, making it suitable for mission-critical applications.

38. What is a primary key in MySQL?

A primary key in MySQL is a unique identifier for each row in a table. It ensures that each record can be uniquely identified and provides a way to enforce entity integrity. A primary key can consist of one or more columns, and its values cannot be null.

39. Explain the concept of a foreign key.

A foreign key in MySQL establishes a relationship between two tables by linking a column or group of columns in one table to the primary key column(s) in another table. It enforces referential integrity, ensuring that values in the foreign key column(s) match values in the referenced primary key column(s) of the related table.

40. Describe the difference between DELETE and TRUNCATE commands.

The DELETE command is used to remove rows from a table based on specified criteria, allowing for selective deletion. TRUNCATE, on the other hand, removes all rows from a table, resetting auto-increment values, and is faster than DELETE as it does not generate transaction logs.

41. What does the JOIN statement do in MySQL? Explain the different types of joins.

The JOIN statement in MySQL is used to retrieve data from multiple tables based on a related column between them. Different types of joins include INNER JOIN (returns rows when there is a match in both tables), LEFT JOIN (returns all rows from the left table and matching rows from the right table), RIGHT JOIN (returns all rows from the right table and matching rows from the left table), and FULL JOIN (returns all rows when there is a match in either table).

42. How can you optimize a MySQL query?

MySQL query optimization involves various techniques such as indexing, using appropriate data types, minimizing the number of queries, optimizing table structure, avoiding unnecessary calculations, and utilizing query caching.

43. Explain the concept of normalization in database design.

Normalization is the process of organizing data in a database to reduce redundancy and dependency. It involves breaking down tables into smaller, related tables and defining relationships between them to ensure data integrity and minimize anomalies.

44. Describe denormalization and when you might use it.

Denormalization is the process of intentionally introducing redundancy into a database design to improve performance by reducing the number of joins required to retrieve data. It is often used in read-heavy applications where query performance is critical, at the expense of some data redundancy and update complexity.

45. What are transactions in MySQL and how do you manage them?

Transactions in MySQL are sequences of SQL operations that are executed as a single unit of work, either all succeed or all fail. They are managed using the BEGIN, COMMIT, and ROLLBACK statements to start, commit, and roll back transactions, respectively.

46. How would you implement ACID properties in MySQL?

ACID (Atomicity, Consistency, Isolation, Durability) properties can be implemented in MySQL by using transactions to ensure that database operations are atomic, consistent, isolated, and durable.

47. What is the significance of HAVING clause in MySQL?

The HAVING clause in MySQL is used to filter rows returned by a GROUP BY clause based on specified conditions. It is similar to the WHERE clause but is applied after grouping and aggregation functions.

48. Explain the difference between CHAR and VARCHAR data types.

CHAR and VARCHAR are both string data types in MySQL. CHAR stores fixed-length strings, while VARCHAR stores variable-length strings. CHAR is padded with spaces to its defined length, while VARCHAR only stores the actual length of the string.

49. How do you perform a full-text search in MySQL?

Full-text search in MySQL is performed using the MATCH() AGAINST() syntax, where MATCH() specifies the columns to search and AGAINST() specifies the search query. It is applicable only on columns indexed as FULLTEXT.

50. Explain the LIKE clause in MySQL.

The LIKE clause in MySQL is used to search for patterns in strings. It allows the use of wildcard characters such as '%' (matches zero or more characters) and '\_' (matches any single character) to perform flexible pattern matching.

51. Describe the use of GROUP BY and ORDER BY in MySQL.

GROUP BY in MySQL is used to group rows that have the same values into summary rows, typically in conjunction with aggregate functions like SUM or COUNT. ORDER BY is used to sort

52. How do you update a value in a MySQL table?

To update a value in a MySQL table, you can use the UPDATE statement followed by the SET clause to specify the column(s) to be updated and their new values, along with optional WHERE clause to filter which rows to update.

53. Explain the use of LIMIT in MySQL.

The LIMIT clause in MySQL is used to constrain the number of rows returned by a query. It is often used in conjunction with the SELECT statement to retrieve a limited number of rows, such as the first 10 rows, or to implement pagination.

54. Explain the difference between INNER JOIN and OUTER JOIN.

INNER JOIN returns only the rows that have matching values in both tables based on the join condition specified, while OUTER JOIN returns all rows from one or both tables, with unmatched rows filled with NULL values where the join condition is not met.

55. Explain the BETWEEN operator in MySQL.

The BETWEEN operator in MySQL is used to select values within a specified range. It includes both the start and end values in the range. For example, column BETWEEN value1 AND value2 selects rows where the column value is between value1 and value2.

56. What is the significance of the AUTO\_INCREMENT attribute?

The AUTO\_INCREMENT attribute in MySQL is used with numeric primary key columns to automatically generate a unique value for each new row inserted into the table. It simplifies the process of creating primary key values, ensuring uniqueness and sequentiality.

57. Describe how MySQL uses locking to manage concurrency.

MySQL uses locking mechanisms to manage concurrency and ensure data consistency in multi-user environments. It employs various types of locks, including table locks, row locks, and explicit locks, to control access to data and prevent conflicts between concurrent transactions.

58. How would you change a column's data type in an existing MySQL table?

To change a column's data type in an existing MySQL table, you can use the ALTER TABLE statement followed by the MODIFY COLUMN clause, specifying the column name and the new data type.

**SQL Interview Questions at a Glance**

Throughout this article, we'll explore a range of SQL interview questions and answers for practitioners at the beginner and intermediate levels. If you're looking for an overview, we've compiled some top tips below:

**For beginners**

1. **General questions**. Expect questions about your experience, the SQL flavors you're familiar with, and your level of proficiency.
2. **Technical questions**. These will cover the basics of SQL, such as what it is, its applications, SQL statements, SQL commands, and types of SQL queries, among others.

**For intermediate practitioners**

1. **Functions in SQL**. You should know about aggregate and scalar functions, as well as built-in and user-defined functions.
2. **Advanced commands**. Questions may cover topics like joins, primary and foreign keys, indexes, and SQL relationships.
3. **Database design**. Expect questions on normalization, denormalization, and the differences between various SQL statements like DELETE, TRUNCATE, and DROP.
4. **Advanced queries**. You may be asked about subqueries, both nested and correlated, as well as how to perform specific tasks like finding the nth highest value in a column.

**General SQL Interview Questions for Beginners**

Before asking you technical questions, your interviewer may ask you some general questions about your overall experience with SQL. You can expect the following questions:

* **What SQL flavors are you familiar with?**
* **How can you estimate your level of proficiency in SQL?**
* **For how long have you been working in SQL?**

While this information can be mentioned in your resume, be ready to talk about it. Naturally, there are no "right" answers to such questions, and there is no need to make up things when answering them.

Don't worry if your experience in SQL is limited: this is something your interviewer, most probably, already knows from your resume. Since they are interested in talking to you anyway, your profile was considered a good fit for their company.

Also, it's perfectly fine if you have only worked with one SQL flavor. Remember that all SQL dialects are fairly similar. Therefore, being familiar with only one of them is a solid basis for you to learn any others.

**Technical SQL Interview Questions for Beginners**

Now, let's move on to the technical SQL interview questions and some potential answers to them.

When answering technical questions, the best strategy is to give as precise answers as possible. It may look like an attempt to deviate from the main topic. In addition, it may provoke additional questions about which you can feel less confident.

**1. What is SQL?**

It stands for [**Structured Query Language,**](https://www.datacamp.com/blog/all-about-sql-the-essential-language-for-database-management) and it's a programming language used for interaction with relational database management systems (RDBMS). This includes fetching, updating, inserting, and removing data from tables.

**2. What are SQL dialects? Give some examples.**

The various versions of SQL, both free and paid, are also called SQL dialects. All the flavors of SQL have a very similar syntax and vary insignificantly only in additional functionality. Some examples are Microsoft SQL Server, PostgreSQL, MySQL, SQLite, T-SQL, Oracle, and MongoDB.

**3. What are the main applications of SQL?**

Using SQL, we can:

* create, delete, and update tables in a database
* access, manipulate, and modify data in a table
* retrieve and summarize the necessary information from a table or several tables
* add or remove certain rows or columns from a table

All in all, SQL allows querying a database in multiple ways. In addition, SQL easily integrates with other programming languages, such as Python or R, so we can use their combined power.

**4. What is an SQL statement? Give some examples.**

Also known as an SQL command. It's a string of characters interpreted by the SQL engine as a legal command and executed accordingly. Some examples of SQL statements are SELECT, CREATE, DELETE, DROP, REVOKE, and so on.

**5. What types of SQL commands (or SQL subsets) do you know?**

* **Data Definition Language (DDL)** – to define and modify the structure of a database.
* **Data Manipulation Language (DML)** – to access, manipulate, and modify data in a database.
* **Data Control Language (DCL)** – to control user access to the data in the database and give or revoke privileges to a specific user or a group of users.
* **Transaction Control Language (TCL)** – to control transactions in a database.
* **Data Query Language (DQL)** – to perform queries on the data in a database to retrieve the necessary information from it.

**6. Give some examples of common SQL commands of each type.**

* **DDL:** CREATE, ALTER TABLE, DROP, TRUNCATE, and ADD COLUMN
* **DML:** UPDATE, DELETE, and INSERT
* **DCL:** GRANT and REVOKE
* **TCL:** COMMIT, SET TRANSACTION, ROLLBACK, and SAVEPOINT
* **DQL:** – SELECT

**7. What is a database?**

A structured storage space where the data is kept in many tables and organized so that the necessary information can be easily fetched, manipulated, and summarized.

**8. What is DBMS, and what types of DBMS do you know?**

It stands for Database Management System, a software package used to perform various operations on the data stored in a database, such as accessing, updating, wrangling, inserting, and removing data. There are various types of DBMS, such as relational, hierarchical, network, graph, or object-oriented. These types are based on the way the data is organized, structured, and stored in the system.

**9. What is RDBMS? Give some examples of RDBMS.**

It stands for Relational Database Management System. It's the most common type of DBMS used for working with data stored in multiple tables related to each other by means of shared keys. The SQL programming language is designed to interact with RDBMS. Some examples of RDBMS are MySQL, PostgreSQL, Oracle, MariaDB, etc.

**10. What are tables and fields in SQL?**

A table is an organized set of related data stored in a tabular form, i.e., in rows and columns. A field is another term for a column of a table.

**11. What is an SQL query, and what types of queries do you know?**

A query is a piece of code written in SQL to access or modify data from a database.

There are two types of SQL queries: **select** and **action** queries. The first ones are used to retrieve the necessary data (this also includes limiting, grouping, ordering the data, extracting the data from multiple tables, etc.), while the second ones are used to create, add, delete, update, rename the data, etc.

**12. What is a subquery?**

Also called an inner query, a query placed inside another query, or an outer query. A subquery may occur in the clauses such as SELECT, FROM, WHERE, UPDATE, etc. It's also possible to have a subquery inside another subquery. The innermost subquery is run first, and its result is passed to the containing query (or subquery).

**13. What types of SQL subqueries do you know?**

* **Single-row** – returns at most one row.
* **Multi-row** – returns at least two rows.
* **Multi-column** – returns at least two columns.
* **Correlated** – a subquery related to the information from the outer query.
* **Nested** – a subquery inside another subquery.

**14. What is a constraint, and why use constraints?**

A set of conditions defining the type of data that can be input into each column of a table. Constraints ensure data integrity in a table and block undesired actions.

**15. What SQL constraints do you know?**

* DEFAULT – provides a default value for a column.
* UNIQUE – allows only unique values.
* NOT NULL – allows only non-null values.
* PRIMARY KEY – allows only unique and strictly non-null values (NOT NULL and UNIQUE).
* FOREIGN KEY – provides shared keys between two or more tables.

**16. What is a join?**

A clause used to combine and retrieve records from two or multiple tables. SQL tables can be joined based on the relationship between the columns of those tables. Check out our [SQL joins**tutorial**](https://www.datacamp.com/tutorial/introduction-to-sql-joins) for more context, plus our dedicated guide to [**SQL joins interview questions**](https://www.datacamp.com/blog/top-sql-joins-interview-questions).

**17. What types of joins do you know?**

* (INNER) JOIN – returns only those records that satisfy a defined join condition in both (or all) tables. It's a default SQL join.
* LEFT (OUTER) JOIN – returns all records from the left table and those records from the right table that satisfy a defined join condition.
* RIGHT (OUTER) JOIN – returns all records from the right table and those records from the left table that satisfy a defined join condition.
* FULL (OUTER) JOIN – returns all records from both (or all) tables. It can be considered as a combination of left and right joins.

**18. What is a primary key?**

A column (or multiple columns) of a table to which the PRIMARY KEY constraint was imposed to ensure unique and non-null values in that column. In other words, a primary key is a combination of the NOT NULL and UNIQUE constraints. The primary key uniquely identifies each record of the table. Each table should contain a primary key and can't contain more than one primary key.

**19. What is a unique key?**

A column (or multiple columns) of a table to which the UNIQUE constraint was imposed to ensure unique values in that column, including a possible NULL value (the only one).

**20. What is a foreign key?**

A column (or multiple columns) of a table to which the FOREIGN KEY constraint was imposed to link this column to the primary key in another table (or several tables). The purpose of foreign keys is to keep connected various tables of a database.

**21. What is an index?**

A special data structure related to a database table and used for storing its important parts and enabling faster data search and retrieval. Indexes are especially efficient for large databases, where they significantly enhance query performance.

**22. What types of indexes do you know?**

* **Unique index** – doesn't allow duplicates in a table column and hence helps maintain data integrity.
* **Clustered index** – defines the physical order of records of a database table and performs data searching based on the key values. A table can have only one clustered index.
* **Non-clustered index** – keeps the order of the table records that don't match the physical order of the actual data on the disk. It means that the data is stored in one place and a non-clustered index – in another one. A table can have multiple non-clustered indexes.

**23. What is a schema?**

A collection of database structural elements such as tables, stored procedures, indexes, functions, and triggers. It shows the overall database architecture, specifies the relationships between various objects of a database, and defines different access permissions for them. Read our [**database schema guide**](https://www.datacamp.com/tutorial/database-schema) for a deeper understanding.

**24. What is a SQL comment?**

A human-readable clarification of what a particular piece of code does. SQL code comments can be single-line (preceded by a double dash --) or span over multiple lines (as follows: /\*comment\_text\*/). When the SQL engine runs, it ignores code comments. The purpose of adding SQL code comments is to make the code more comprehensive for those people who will read it in the future.

**25. What is a SQL operator?**

A reserved character, a combination of characters, or a keyword used in SQL queries to perform a specific operation. SQL operators are commonly used with the WHERE clause to set a condition (or conditions) for filtering the data.

**26. What types of SQL operators do you know?**

* **Arithmetic** (+, -, \*, /, etc.)
* **Comparison** (>, <, =, >=, etc.)
* **Compound** (+=, -=, \*=, /=, etc.)
* **Logical** (AND, OR, NOT, BETWEEN, etc.)
* **String** (%, \_, +, ^, etc.)
* **Set** (UNION, UNION ALL, INTERSECT, and MINUS (or EXCEPT))

**27. What is an alias?**

A temporary name given to a table (or a column in a table) while executing a certain SQL query. Aliases are used to improve the code readability and make the code more compact. An alias is introduced with the AS keyword:

SELECT col\_1 AS column

FROM table\_name;

[**Powered By**](https://www.datacamp.com/datalab)

**28. What is a clause?**

A condition imposed on a SQL query to filter the data to obtain the desired result. Some examples are WHERE, LIMIT, HAVING, LIKE, AND, OR, ORDER BY, etc.

**29. What are some common statements used with the SELECT query?**

The most common ones are FROM, GROUP BY, JOIN, WHERE, ORDER BY, LIMIT, and HAVING.

**30. How to create a table?**

Using the CREATE TABLE statement. For example, to create a table with three columns of predefined datatypes, we apply the following syntax:

CREATE TABLE table\_name (col\_1 datatype,

col\_2 datatype,

col\_3 datatype);

[**Powered By**](https://www.datacamp.com/datalab)

**31. How to update a table?**

Using the UPDATE statement. The syntax is:

UPDATE table\_name

SET col\_1 = value\_1, column\_2 = value\_2

WHERE condition;

[**Powered By**](https://www.datacamp.com/datalab)

**32. How to delete a table from a database?**

Using the DROP TABLE statement. The syntax is: DROP TABLE table\_name;.

**33. How to get the count of records in a table?**

Using the COUNT() aggregate function with the asterisk passed as its argument: SELECT COUNT(\*) FROM table\_name;.

**34. How to sort records in a table?**

Using the ORDER BY statement:

SELECT \* FROM table\_name

ORDER BY col\_1;

[**Powered By**](https://www.datacamp.com/datalab)

We can specify that we need a descending order using the DESC keyword; otherwise, the order will be ascending by default. Also, we can sort by more than one column and specify for each one, ascending or descending order separately. For example:

SELECT \* FROM table\_name

ORDER BY col\_1 DESC, col\_3, col\_6 DESC;

[**Powered By**](https://www.datacamp.com/datalab)

**35. How to select all columns from a table?**

Using the asterisk \* with the SELECT statement. The syntax is: SELECT \* FROM table\_name;.

**36. How to select common records from two tables?**

Using the INTERSECT statement:

SELECT \* FROM table\_1

INTERSECT

SELECT \* FROM table\_1;

[**Powered By**](https://www.datacamp.com/datalab)

**37. What is the DISTINCT statement and how do you use it?**

This statement is used with the SELECT statement to filter out duplicates and return only unique values from a column of a table. The syntax is:

SELECT DISTINCT col\_1

FROM table\_name;

[**Powered By**](https://www.datacamp.com/datalab)

**38. What are relationships? Give some examples.**

Relationships are the connections and correlations between entities, basically meaning how two or more tables of a database are related to one another. For example, we can find an ID of the same client in a table on sales data and in a customer table.

**39. What is a NULL value? How is it different from zero or a blank space?**

A NULL value indicates the absence of data for a certain cell of a table. Instead, zero is a valid numeric value, and an empty string is a legal string of zero length.

**40. What is the difference between SQL and NoSQL?**

SQL databases are relational, structured, and use tables with predefined schemas, while NoSQL databases are non-relational, schema-less, and designed to handle unstructured or semi-structured data.

**41. What are some common challenges when working with SQL databases?**

Challenges include performance tuning for large datasets, managing indexing strategies, ensuring data integrity with constraints, handling concurrent transactions, and optimizing query execution.

**Intermediate SQL Interview Questions**

In this section, we take a look at the most popular intermediate SQL questions and answers so that you'll know what to expect from your interviewer.

**42. What is a function in SQL, and why use functions?**

A database object representing a set of SQL statements frequently used for a certain task. A function takes in some input parameters, performs calculations or other manipulations on them, and returns the result. Functions help improve code readability and avoid repetition of the same code snippets.

**43. What types of SQL functions do you know?**

* **Aggregate functions** – work on multiple, usually grouped records for the provided columns of a table, and return a single value (usually by group).
* **Scalar functions** – work on each individual value and return a single value.

On the other hand, SQL functions can be built-in (defined by the system) or user-defined (created by the user for their specific needs).

**44. What aggregate functions do you know?**

* AVG() – returns the average value
* SUM() – returns the sum of values
* MIN() – returns the minimum value
* MAX() – returns the maximum value
* COUNT() – returns the number of rows, including those with null values
* FIRST() – returns the first value from a column
* LAST()– returns the last value from a column

**45. What scalar functions do you know?**

* LEN() (in other SQL flavors – LENGTH()) – returns the length of a string, including the blank spaces
* UCASE() (in other SQL flavors – UPPER()) – returns a string converted to the upper case
* LCASE() (in other SQL flavors – LOWER()) – returns a string converted to the lower case
* INITCAP() – returns a string converted to the title case (i.e., each word of the string starts from a capital letter)
* MID() (in other SQL flavors – SUBSTR()) – extracts a substring from a string
* ROUND() – returns the numerical value rounded to a specified number of decimals
* NOW() – returns the current date and time

**46. What are case manipulation functions? Give some examples.**

Case manipulation functions represent a subset of character functions, and they're used to change the case of the text data. With these functions, we can convert the data into the upper, lower, or title case.

* UCASE() (in other SQL flavors – UPPER()) – returns a string converted to the upper case
* LCASE() (in other SQL flavors – LOWER()) – returns a string converted to the lower case
* INITCAP() – returns a string converted to the title case (i.e., each word of the string starts from a capital letter)

**47. What are character manipulation functions? Give some examples.**

Character manipulation functions represent a subset of character functions, and they're used to modify the text data.

* CONCAT() – joins two or more string values appending the second string to the end of the first one
* SUBSTR() – returns a part of a string satisfying the provided start and end points
* LENGTH() (in other SQL flavors – LEN()) – returns the length of a string, including the blank spaces
* REPLACE() – replaces all occurrences of a defined substring in a provided string with another substring
* INSTR() – returns the numeric position of a defined substring in a provided string
* LPAD() and RPAD() – return the padding of the left-side/right-side character for right-justified/left-justified value
* TRIM() – removes all the defined characters, as well as white spaces, from the left, right, or both ends of a provided string

**48. What is the difference between local and global variables?**

Local variables can be accessed only inside the function in which they were declared. Instead, global variables, being declared outside any function, are stored in fixed memory structures and can be used throughout the entire program.

**49. What is the difference between SQL and PL/SQL?**

SQL is a standard language for querying and managing relational databases, primarily used for data manipulation and retrieval. PL/SQL (Procedural Language/SQL) is an extension of SQL used in Oracle databases that includes procedural programming constructs like loops, conditions, and exception handling, allowing for complex business logic to be implemented within the database. We have another article on the [**Top 20 PL/SQL Interview Questions and Answers**](https://www.datacamp.com/blog/pl-sql-interview-questions) which is a good review if you know you will be asked about your knowledge of Oracle.

**50. What is the difference between LEFT JOIN and LEFT OUTER JOIN?**

There is no [**difference between LEFT JOIN and LEFT OUTER JOIN**](https://www.datacamp.com/tutorial/left-join-vs-left-outer-join). They are interchangeable. SQL allows the OUTER keyword to be optional, so LEFT JOIN is simply a shorthand for LEFT OUTER JOIN. Both return all records from the left table and the matching records from the right table.

**51. What is indexing in SQL, and how does it improve performance?**

Indexing creates a special data structure that speeds up data retrieval by allowing the database to find rows more efficiently. It works like an optimized lookup table, reducing the need for full table scans. However, excessive indexing can slow down insert, update, and delete operations due to the need for index maintenance.

**52. What is a stored procedure, and how is it different from a function?**

A [**stored procedure**](https://www.datacamp.com/tutorial/sql-stored-procedure) is a precompiled set of SQL statements that can be executed as a single unit to perform a specific task. Unlike functions, stored procedures can return multiple result sets and modify database objects, while functions typically return a single value and cannot modify data.

**53. What is the default data ordering with the ORDER BY statement, and how do you change it?**

By default, the order is ascending. To change it to descending, we need to add the DESC keyword as follows:

SELECT \* FROM table\_name

ORDER BY col\_1 DESC;

[**Powered By**](https://www.datacamp.com/datalab)

**54. What set operators do you know?**

* UNION – returns the records obtained by at least one of two queries (excluding duplicates)
* UNION ALL – returns the records obtained by at least one of two queries (including duplicates)
* INTERSECT – returns the records obtained by both queries
* EXCEPT (called MINUS in MySQL and Oracle) – returns only the records obtained by the first query but not the second one

**55. What operator is used in the query for pattern matching?**

The LIKE operator in combination with the % and \_ wildcards. The % wildcard represents any number of characters including zero, while \_ – strictly one character.

**56. What is the difference between a primary key and a unique key?**

While both types of keys ensure unique values in a column of a table, the first one uniquely identifies each record of the table, and the second one prevents duplicates in that column.

**57. What is a composite primary key?**

The primary key of a table, based on multiple columns.

**58. What is the order of appearance of the common statements in the SELECT query?**

SELECT – FROM – JOIN – ON – WHERE – GROUP BY – HAVING – ORDER BY – LIMIT

**59. In which order does the interpreter execute the common statements in the SELECT query?**

Here is the [**SQL order of execution**](https://www.datacamp.com/tutorial/sql-order-of-execution):

FROM – JOIN – ON – WHERE – GROUP BY – HAVING – SELECT – ORDER BY – LIMIT

**60. What is a view, and why use it?**

A virtual table containing a subset of data retrieved from one or more database tables (or other views). Views take very little space, simplify complex queries, limit access to the data for security reasons, enable data independence, and summarize data from multiple tables.

**61. Can we create a view based on another view?**

Yes. This is also known as nested views. However, we should avoid nesting multiple views since the code becomes difficult to read and debug.

**62. Can we still use a view if the original table is deleted?**

No. Any views based on that table will become invalid after deleting the base table. If we try to use such a view anyway, we'll receive an error message.

**63. What types of SQL relationships do you know?**

* **One-to-one** – each record in one table corresponds to only one record in another table
* **One-to-many** – each record in one table corresponds to several records in another table
* **Many-to-many** – each record in both tables corresponds to several records in another table

**64. What are the possible values of a BOOLEAN data field?**

In some SQL flavors, such as PostgreSQL, the BOOLEAN data type exists explicitly and takes values TRUE, FALSE, or NULL. In other flavors, such as Microsoft SQL Server, the BIT datatype is used to store Boolean values as integers 1 (true) or 0 (false).

**65. What is normalization in SQL, and why use it?**

Normalization is a process of database design that includes organizing and restructuring data in a way to reduce data redundancy, dependency, duplication, and inconsistency. This leads to enhanced data integrity, more tables within the database, more efficient data access and security control, and greater query flexibility.

**66. What is denormalization in SQL, and why use it?**

Denormalization is the process opposite of normalization: it introduces data redundancy and combines data from multiple tables. Denormalization optimizes the performance of the database infrastructure in situations when read operations are more important than write operations since it helps avoid complex joins and reduces the time of query running.

**67. What is the difference between renaming a column and giving an alias to it?**

Renaming a column means permanently changing its actual name in the original table. Giving an alias to a column means giving it a temporary name while executing an SQL query, with the purpose to make the code more readable and compact.

**68. What is the difference between nested and correlated subqueries?**

A correlated subquery is an inner query nested in a bigger (outer) query that refers to the values from the outer query for its execution, meaning that a correlated subquery depends on its outer query. Instead, a non-correlated subquery doesn't rely on the data from the outer query and can be run independently of it.

**69. What is the difference between clustered and non-clustered indexes?**

While a clustered index **defines the physical order of records** of a table and performs data searching based on the key values, a non-clustered index **keeps the order of records that do not match the physical order of the actual data** on the disk. A table can have only one clustered index but many non-clustered ones.

**70. What is the CASE() function?**

The way to implement the *if-then-else* logic in SQL. This function sequentially checks the provided conditions in the WHEN clauses and returns the value from the corresponding THEN clause when the first condition is satisfied. If none of the conditions is satisfied, the function returns the value from the ELSE clause in case it's provided, otherwise, it returns NULL. The syntax is:

CASE

WHEN condition\_1 THEN value\_1

WHEN condition\_2 THEN value\_2

WHEN condition\_3 THEN value\_3

...

ELSE value

END;

[**Powered By**](https://www.datacamp.com/datalab)

**71. What is the difference between the DELETE and TRUNCATE statements?**

DELETE is a reversible DML (Data Manipulation Language) command used to delete one or more rows from a table based on the conditions specified in the WHERE clause. Instead, TRUNCATE is an irreversible DDL (Data Definition Language) command used to delete all rows from a table. DELETE works slower than TRUNCATE. Also, we can't use the TRUNCATE statement for a table containing a foreign key.

**72. What is the difference between the DROP and TRUNCATE statements?**

DROP deletes a table from the database completely, including the table structure and all the associated constraints, relationships with other tables, and access privileges. TRUNCATE deletes all rows from a table without affecting the table structure and constraints. DROP works slower than TRUNCATE. Both are irreversible DDL (Data Definition Language) commands.

**73. What is the difference between the HAVING and WHERE statements?**

The first one works on aggregated data after they are grouped, while the second one checks each row individually. If both statements are present in a query, they appear in the following order: WHERE – GROUP BY – HAVING. The SQL engine interprets them also in the same order.

**74. How do you add a record to a table?**

Using the INSERT INTO statement in combination with VALUES. The syntax is:

INSERT INTO table\_name

VALUES (value\_1, value\_2, ...);

[**Powered By**](https://www.datacamp.com/datalab)

**75. How do you delete a record from a table?**

Using the DELETE statement. The syntax is:

DELETE FROM table\_name

WHERE condition;

[**Powered By**](https://www.datacamp.com/datalab)

In this way, we can also delete multiple records if they satisfy the provided condition.

**76. How do you add a column to a table?**

Using the ALTER TABLE statement in combination with ADD. The syntax is:

ALTER TABLE table\_name

ADD column\_name datatype;

[**Powered By**](https://www.datacamp.com/datalab)

**77. How do you rename a column of a table?**

Using the ALTER TABLE statement in combination with RENAME COLUMN ... TO ... The syntax is:

ALTER TABLE table\_name

RENAME COLUMN old\_column\_name TO new\_column\_name;

[**Powered By**](https://www.datacamp.com/datalab)

**78. How do you delete a column from a table?**

Using the ALTER TABLE statement in combination with DROP COLUMN. The syntax is:

ALTER TABLE table\_name

DROP COLUMN column\_name;

[**Powered By**](https://www.datacamp.com/datalab)

**79. How do you select all even or all odd records in a table?**

By checking the remainder of the division by 2. In some SQL versions (e.g., PostgreSQL and My SQL), we use the MOD function, in the others (Microsoft SQL Server and SQLite) – the modulo operator (%). To select all even records using MOD:

SELECT \* FROM table\_name

WHERE MOD(ID\_column, 2) = 0;

[**Powered By**](https://www.datacamp.com/datalab)

To select all even records using %:

SELECT \* FROM table\_name

WHERE ID\_column % 2 = 0;

[**Powered By**](https://www.datacamp.com/datalab)

To select all odd records, the syntax is identical in both cases, only that we would use the inequality operator <> instead of =.

**80. How to prevent duplicate records when making a query?**

Using the DISTINCT statement in combination with SELECT or creating a unique key for that table.

**81. How do you insert many rows in a table?**

Using the INSERT INTO statement in combination with VALUES. The syntax is:

INSERT INTO table\_name

VALUES (value\_1, value\_2, ...),

(value\_3, value\_4, ...),

(value\_5, value\_6, ...),

...;

[**Powered By**](https://www.datacamp.com/datalab)

**82. How do you find the nth highest value in a column of a table?**

Using the OFFSET clause. For example, to find the 6th highest value from a column, we would use the following syntax:

SELECT \* FROM table\_name

ORDER BY column\_name DESC

LIMIT 1

OFFSET 5;

[**Powered By**](https://www.datacamp.com/datalab)

**83. How do you find the values in a text column of a table that start with a certain letter?**

Using the LIKE operator in combination with the % and \_ wildcards. For example, we need to find all surnames in a table that start with "A". The query is:

SELECT \* FROM table\_name

WHERE surname LIKE 'A\_';

[**Powered By**](https://www.datacamp.com/datalab)

Here, we assume that a surname must contain at least two letters. Without this assumption (meaning that a surname can be just A), the query is as follows:

SELECT \* FROM table\_name

WHERE surname LIKE 'A%';

[**Powered By**](https://www.datacamp.com/datalab)

**84. How do you find the last id in a table?**

Using the MAX() function. Otherwise, in many SQL versions, we can use the following syntax:

SELECT id

FROM table\_name

ORDER BY id DESC

LIMIT 1;

[**Powered By**](https://www.datacamp.com/datalab)

or in Microsoft SQL Server:

SELECT TOP 1 id

FROM table\_name

ORDER BY id DESC

[**Powered By**](https://www.datacamp.com/datalab)

**85. How to select random rows from a table?**

Using the RAND() function in combination with ORDER BY and LIMIT. In some SQL flavors, such as PostgreSQL, it's called RANDOM(). For example, the following code will return five random rows from a table in MySQL:

SELECT \* FROM table\_name

ORDER BY RAND()

LIMIT 5;

[**Powered By**](https://www.datacamp.com/datalab)

**Upskilling Your Team with SQL**

While SQL interview preparation is crucial for job seekers and hiring managers, it’s equally important for businesses to invest in continuous SQL training for their teams. Being able to work with data is more important than ever, so ensuring that your employees have strong SQL skills can be a game-changer for your company's success.

If you’re a team leader or business owner looking to ensure your whole team is proficient in SQL, [**DataCamp for Business**](https://www.datacamp.com/business) offers tailored training programs that can help your employees master SQL skills, from the basics to advanced concepts. We can provide:

* **Targeted learning paths**: Customizable to your team’s current skill level and business needs.
* **Hands-on practice**: Real-world scenarios and exercises that reinforce learning and improve retention.
* **Progress tracking**: Tools to monitor and assess your team’s progress, ensuring they achieve their learning goals.

Investing in SQL upskilling through platforms like DataCamp not only enhances your team’s capabilities but also provides your business with a strategic advantage, enabling you to stay competitive and deliver results. Talk to one of our team and [**request a demo today**](https://www.datacamp.com/business/demo).

**Conclusion**

To sum up, we discussed the 85 essential beginner and intermediate SQL interview questions and the right answers to them. Hopefully, this information will help you to get ready for the interview and feel more confident, whether you're looking for a job in SQL or hiring candidates for an intermediate SQL position.

If you feel that you need more training to better prepare for an interview, consider the following [**SQL courses**](https://www.datacamp.com/category/sql) and tracks of DataCamp: