

Top 84 SQL Interview Questions and Answers for 2026

Get interview-ready with this comprehensive overview of essential SQL interview questions and answers for job hunters, hiring managers, and recruiters.

Updated Oct 17, 2025 · 15 min read



Elena Kosourova

IBM Certified Data Scientist proficient in Python, R, SQL, data analysis, and machine learning

TOPICS

SQL

Data Analysis

Career Services

Whether you're a job hunter looking for a new opportunity to apply your SQL skills or a hiring manager who is going to interrogate a candidate for a job opening in their company, knowing common SQL interview questions and answers is a must.

This article lists **84 SQL interview questions and answers for 2026**, covering key SQL topics to help you prepare, learn, and hire effectively, including:

- **Foundational Knowledge:** Clear definitions of RDBMS, keys, and SQL dialects.
- **Technical Mastery:** Deep dives into DDL vs. DML, constraints, and indexing.
- **Query Logic:** Explanations of Joins, Subqueries, and Aggregate Functions.
- **Practical Application:** Real-world coding solutions, such as finding the nth highest value or removing duplicates.

For those looking to enhance their knowledge and skills further, consider exploring our comprehensive courses in [data engineering](#), [cloud technologies](#), and [AWS](#).

Associate Data Engineer in SQL

Gain practical knowledge in ETL, SQL, and data warehousing for data engineering.

Explore Track

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 dialects 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

Before asking you technical questions, your interviewer may ask you some general questions about your overall experience with SQL.

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 dialect. 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.

1. What is SQL?

It stands for **Structured Query Language**, 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 dialects 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, and Oracle.

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

SQL Interview Questions for Beginners

Your interviewer may get the ball rolling by starting with the easier questions:

4. What is an SQL statement?

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 is an SQL query?

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: queries for data retrieval and data modification. 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, and rename the data.

6. What is an SQL 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).

7. What is an SQL 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](#) for more context, plus our dedicated guide to [SQL joins interview questions](#).

8. What is an 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.

9. What is an SQL 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  datalab

Technical SQL Interview Questions

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.

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.

10. What types of SQL commands 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.

11. Give some examples of common SQL commands.

- **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

12. 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.

13. 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.

14. 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.

15. 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.

16. 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.

17. 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.

18. 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.

Note: **FULL OUTER JOIN** is supported by PostgreSQL, SQL Server, Oracle, and MySQL 8.0 and above, but MySQL only allows it through **UNION** patterns; meanwhile, SQLite does not support **RIGHT JOIN**, which can be emulated using **LEFT JOIN** in combination with **UNION**.

19. What is a primary key in SQL?

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 can define at most one **PRIMARY KEY** (which may be composite). A **PRIMARY KEY** is strongly recommended but not strictly required by all engines.

20. What is a unique key in SQL?

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).

Note:

- SQL Server: Only one **NULL** allowed unless you use a filtered index.
- PostgreSQL / Oracle / MySQL: Multiple **NULL**s allowed because **NULL <> NULL**.

21. What is a foreign key in SQL?

A column (or multiple columns) of a table to which the **FOREIGN KEY** (or **UNIQUE 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.

22. What is an SQL 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.

23. 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.

24. 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](#) for a deeper understanding.

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 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.

28. 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`.

29. How do you create a table in SQL?

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  datalab

30. How to update a table?

Using the `UPDATE` statement. The syntax is:

```
UPDATE table_name  
SET col_1 = value_1, col_2 = value_2  
WHERE condition;
```



POWERED BY  datalab

31. How to delete a table from a database?

Using the `DROP TABLE` statement. The syntax is: `DROP TABLE table_name;` .

32. 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;` .

33. How to sort records in a table?

Using the `ORDER BY` statement:

```
SELECT * FROM table_name
ORDER BY col_1;
```



POWERED BY 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 datalab

34. How to select all columns from a table?

Using the asterisk `*` with the `SELECT` statement. The syntax is: `SELECT * FROM table_name;` .

35. How to select common records from two tables?

Using the `INTERSECT` statement:

```
SELECT col1, col2 FROM table_1
INTERSECT
SELECT col1, col2 FROM table_2;
```



POWERED BY datalab

Note: `INTERSECT` requires the same number of columns and compatible types.

36. 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 datalab

37. 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.

38. 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.

39. 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.

40. 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.

41. What is a function in SQL?

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.

42. 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).

43. What SQL 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

44. What SQL scalar functions do you know?

- `LEN()` (in other SQL dialects – `LENGTH()`) – returns the length of a string, including the blank spaces
- `UCASE()` (in other SQL dialects – `UPPER()`) – returns a string converted to the upper case
- `LCASE()` (in other SQL dialects – `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 dialects – `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

45. What are case manipulation functions in SQL?

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 dialects – `UPPER()`) – returns a string converted to the upper case
- `LCASE()` (in other SQL dialects – `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)

46. What are character manipulation functions in SQL?

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
- `SUBSTRING()` / `SUBSTR()` – returns a part of a string satisfying the provided start and end points
- `LENGTH()` (in other SQL dialects – `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

47. 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.

48. 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](#), which is a good review if you know you will be asked about your knowledge of Oracle.

49. What is the difference between LEFT JOIN and LEFT OUTER JOIN?

There is no [difference between LEFT JOIN and 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.

50. 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.

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

A **stored procedure** is a precompiled set of SQL statements executed as a unit to perform a task. Procedures can modify data or schema objects, manage transactions, and return zero or more result sets. Functions, on the other hand, are typically used in SQL expressions, must return a value (scalar or table-valued), and in many databases are restricted from side effects. Exact behavior differs by DB (e.g., T-SQL has scalar and table-valued functions; PostgreSQL distinguishes between functions and procedures).

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

Default is ascending (NULLS FIRST/LAST varies by DB). Use ASC/DESC per column keyword as follows:

```
SELECT * FROM table_name
ORDER BY col_1 DESC;
```



POWERED BY  datalab

53. What are SQL set operators?

- **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

54. 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.

55. What is the difference between a primary key and a unique key in SQL?

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.

56. What is a SQL composite primary key?

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

57. What is the typical order of SQL clauses in a SELECT statement?

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

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

Here is the **SQL order of execution**:

FROM → ON → JOIN → WHERE → GROUP BY → HAVING → SELECT → ORDER BY → LIMIT/OFFSET (FETCH)

59. What is a view in SQL?

A virtual table containing a subset of data retrieved from one or more database tables (or other views). Standard views store no data; materialized views do. Views can simplify queries, encapsulate logic, and, combined with privileges, restrict columns/rows. They can join/aggregate data from multiple tables.

60. Can we create a view based on another view in SQL?

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

61. 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.

62. 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

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

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

64. What is normalization in SQL?

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.

65. What is denormalization in SQL?

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.

66. 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.

67. 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.

68. 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.

69. 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  datalab

70. What is the difference between the DELETE and TRUNCATE statements?

`DELETE` is a DML (Data Manipulation Language) command used to remove one or more rows from a table based on a condition in the `WHERE` clause. It is reversible only if wrapped in a transaction (via `ROLLBACK`).

`TRUNCATE` is a DDL (Data Definition Language) command that removes all rows from a table by deallocating pages. It is faster but generally irreversible, and cannot be used on tables that are referenced by a foreign key.

71. 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. Both are DDL. `DROP` removes the table and its metadata; `TRUNCATE` deletes all rows but keeps the table definition. Performance and transactional behavior depend on the DB engine.

72. 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.

73. 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  datalab

74. How do you delete a record from a table?

Using the `DELETE` statement. The syntax is:

```
DELETE FROM table_name
WHERE condition;
```



POWERED BY datalab

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

75. 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 datalab

76. 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 datalab

77. 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 datalab

78. 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 MySQL), 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 datalab

To select all even records using `%`:

```
SELECT * FROM table_name
WHERE ID_column % 2 = 0;
```



POWERED BY datalab

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

79. 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.

80. 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  datalab

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

Using window functions to handle ties correctly:

```
SELECT column_name
FROM (
  SELECT column_name, DENSE_RANK() OVER (ORDER BY column_name DESC) AS rnk
  FROM table_name
) t
WHERE rnk = :n;
```



POWERED BY  datalab

For the nth row by order (ignoring ties): `ORDER BY column_name DESC OFFSET n-1 ROWS FETCH NEXT 1 ROW ONLY`.

82. 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  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  datalab

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

The simplest method is to use the [aggregate function](#) `MAX()`.

```
SELECT MAX(id) AS highest_id
FROM table_name;
```



POWERED BY  datalab

```
SELECT id
FROM table_name
ORDER BY id DESC
LIMIT 1;
```

POWERED BY  datalab

84. How to select random rows from a table?

Using the `RAND()` function in combination with `ORDER BY` and `LIMIT`. In some SQL dialects, 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  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](#) 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](#).

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](#) and tracks of DataCamp:

- [Introduction to SQL](#)
- [SQL Foundations](#)
- [Data Analyst in SQL](#)
- [Exploratory Data Analysis in SQL](#)

- [SQL for Business Analysts](#)
- [Analyzing Business Data in SQL](#)
- [Intermediate SQL](#)
- [Intermediate SQL Server](#)
- [Applying SQL to Real-World Problems](#)
- [SQL Server for Database Administrators](#)

Boost Your Team's SQL Proficiency

Train your team in SQL with DataCamp for Business. Comprehensive training, hands-on projects, and detailed performance metrics for your organization.

[Request a Demo Today!](#)



FAQs

How can I start learning SQL? ^

To start learning SQL, begin with the basics of database concepts and relational database management systems. DataCamp has many resources to help you get started, such as the [Introduction to SQL course](#), [Data Analyst in SQL Career Track](#) and the [SQL cheat sheet](#). Alternatively visit the [SQL courses](#) page to browse all resources.

Where can I find real-world SQL problems to practice? v

What are some common SQL interview questions for beginners? v

What are some common SQL interview questions for intermediate practitioners? v

How should I prepare for an SQL interview? v

What should I expect in a technical SQL interview and how can I best demonstrate my skills? v

How important is it to learn specific SQL dialects, and which ones should I focus on? v

Does DataCamp have extra FREE resources to prepare for SQL interviews? v



AUTHOR

Elena Kosourova

IBM Certified Data Scientist (2020), previously Petroleum Geologist/Geomodeler of oil and gas fields worldwide with 12+ years of international work experience. Proficient in Python, R, and SQL. Areas of expertise: data cleaning, data manipulation, data visualization, data analysis, data modeling, statistics, storytelling, machine learning. Extensive experience in managing data science communities and writing/reviewing articles and tutorials on data science and career topics.

TOPICS

SQL

Data Analysis

Career Services

**Training more people?**

Get your team access to the full DataCamp for business platform.

For Business

For a bespoke solution [book a demo](#).

Learn more about SQL with these courses!

COURSE

Data Manipulation in SQL

🕒 4 hr 👤 303.7K

Master the complex SQL queries necessary to answer a wide variety of data science questions and prepare robust data sets for analysis in PostgreSQL.

[See Details →](#)[Start Course](#)[See More →](#)

Related

BLOG

Top 30 SQL Server Interview Questions (2026)

BLOG

Top 30 Database Administrator Interview Questions for 2026

[See More →](#)

Grow your data skills with DataCamp for Mobile

Make progress on the go with our mobile courses and daily 5-minute coding challenges.



LEARN

[Learn Python](#)[Learn AI](#)[Learn Power BI](#)[Learn Data Engineering](#)[Assessments](#)[Career Tracks](#)[Skill Tracks](#)[Courses](#)[Data Science Roadmap](#)

DATA COURSES

[Python Courses](#)[R Courses](#)[SQL Courses](#)[Power BI Courses](#)[Tableau Courses](#)[Alteryx Courses](#)[Azure Courses](#)[AWS Courses](#)[Google Sheets Courses](#)[Excel Courses](#)

- AI Courses
- Data Analysis Courses
- Data Visualization Courses
- Machine Learning Courses
- Data Engineering Courses
- Probability & Statistics Courses

DATALAB

- Get Started
- Pricing
- Security
- Documentation

CERTIFICATION

- Certifications
- Data Scientist
- Data Analyst
- Data Engineer
- SQL Associate
- Power BI Data Analyst
- Tableau Certified Data Analyst
- Azure Fundamentals
- AI Fundamentals

RESOURCES

- Resource Center



**NEW YEAR
NEW SKILLS** 50% OFF

- Tutorials
- Docs
- Open Source
- RDocumentation
- Book a Demo with DataCamp for Business
- Data Portfolio

PLANS

- Pricing
- For Students
- For Business
- For Universities
- Discounts, Promos & Sales
- Expense DataCamp
- DataCamp Donates

FOR BUSINESS

- Business Pricing
- Teams Plan
- Data & AI Unlimited Plan
- Customer Stories
- Partner Program

ABOUT

- About Us
- Learner Stories
- Careers
- Become an Instructor
- Press
- Leadership
- Contact Us
- DataCamp Español
- DataCamp Português
- DataCamp Deutsch
- DataCamp Français

SUPPORT

- Help Center
- Become an Affiliate



[Privacy Policy](#) [Cookie Notice](#) [Do Not Sell My Personal Information](#) [Accessibility](#) [Security](#) [Terms of Use](#)

© 2026 DataCamp, Inc. All Rights Reserved.