

100

SQL Interview

Questions & Answers

Master These To Land Your Next Data Job



written and created by

INFORMESSOR

This guide gives you the 100 most common and practical SQL questions you are likely to face in real interviews, along with clear and accurate answers. It walks you from fundamentals like joins, keys, and filtering all the way to indexes, transactions, window functions, and performance tuning. Each answer is written in simple language so you can truly understand the concept and then explain it confidently to a hiring manager.

You will learn how to:

- Explain core SQL theory in your own words
- Read and reason about real world queries
- Understand when and why to use specific features
- Avoid common mistakes that cost candidates offers

Dedicated to students, job seekers, and career changers who are ready to build their future in data.

1) What is SQL and what problem does it solve?

SQL stands for Structured Query Language and is the standard language for working with data in relational databases. You use it to create tables, insert and change data, and read data back with precise conditions.

2) What is a relational database?

A relational database stores data in related tables made of rows and columns. Relationships between tables are defined through keys, which lets you join data safely and efficiently.

3) What is a table in SQL?

A table is a named structure that holds data in rows and columns, similar to a spreadsheet. Each column has a defined data type and each row represents a single record.

4) What are rows and columns?

A row represents one record, such as one customer or one order. A column models one attribute, such as name, email, or order date, and holds the same kind of value for every row.

5) What are the main categories of SQL statements?

SQL is usually grouped into four families. Data Definition Language defines and changes objects, Data Manipulation Language works with data inside those objects, Data Control Language manages permissions, and Transaction Control commands manage commits and rollbacks.

6) What is Data Definition Language DDL?

DDL statements define and change the structure of database objects like tables and indexes. Common DDL commands are CREATE, ALTER, and DROP.

7) What is Data Manipulation Language DML?

DML statements work with the data stored in tables. Typical DML commands are SELECT to read data, and INSERT, UPDATE, and DELETE to modify data.

8) What is Data Control Language DCL?

DCL statements manage which users can do what in the database. GRANT gives permissions and REVOKE removes them.

9) What is Transaction Control Language TCL?

TCL statements handle the life cycle of a transaction. COMMIT makes changes permanent, ROLLBACK undoes them, and SAVEPOINT marks a point inside a transaction you can roll back to.

10) What is a database schema?

A schema is a logical container inside a database that groups related objects such as tables, views, and procedures. It helps organize and secure objects, often per application or team.

11) How is a database different from a schema?

A database is the overall container that holds all data and objects on a server. A schema is a named subset inside a database that groups related objects like tables and views under a single namespace.

12) What is a column alias?

A column alias is a temporary name you assign to a column in a query result, such as `SELECT salary AS base_pay`, to make the output headings clearer or shorter without changing the actual column name in the table.

13) What is a table alias?

A table alias is a short nickname for a table used only inside a query, for example `FROM employees e`. It makes queries easier to read and is especially helpful when joining multiple tables or the same table more than once.

14) What is a primary key?

A primary key is one column or a set of columns that uniquely identifies each row in a table and cannot contain NULL values. It is the main way other tables reference that row.

15) What is a foreign key?

A foreign key is a column or set of columns in one table that points to the primary key of another table. It enforces that referenced values exist, preserving the relationship between the tables.

16) What is a composite key?

A composite key is a primary key made up of more than one column. The combination of all those columns must be unique for each row.

17) What is a candidate key?

A candidate key is any column or group of columns that could uniquely identify a row. One candidate key is chosen to be the primary key, and any remaining candidates are alternate unique identifiers.

18) How is a primary key different from a unique constraint?

Both enforce uniqueness, but a table has only one primary key and its columns cannot be NULL. Unique constraints can exist in multiple places on a table and may allow NULL values depending on the database.

19) What is a surrogate key?

A surrogate key is an artificially created identifier such as an auto increment integer, used as the primary key instead of a naturally meaningful business value.

20) What is a constraint and why is it important?

A constraint is a rule on a table or column, like NOT NULL, UNIQUE, CHECK, or FOREIGN KEY, that the data must follow. Constraints ensure that incorrect or inconsistent data cannot be stored.

21) What does the SELECT statement do?

SELECT retrieves data from one or more tables and returns it as a result set. You can control which columns appear, which rows are included, and how the results are sorted or grouped.

22) What is the purpose of the WHERE clause?

The WHERE clause limits which rows are processed by a query. Only rows that satisfy the condition in WHERE are considered for output, grouping, or updates.

23) What is the difference between WHERE and HAVING?

WHERE filters individual rows before any grouping happens. HAVING filters groups after GROUP BY and usually works with aggregate values like COUNT or SUM.

24) What does ORDER BY do in a query?

ORDER BY sorts the rows in the result set based on one or more columns or expressions, either in ascending or descending order.

25) What does GROUP BY do in a query?

GROUP BY collects rows that share the same values in specified columns into groups, so aggregate functions like COUNT or AVG can be calculated for each group.

26) What is the DISTINCT keyword used for?

DISTINCT removes duplicate rows from the result of a SELECT, returning only unique combinations of the selected columns.

27) What is the LIKE operator?

LIKE is used in WHERE conditions to match text patterns using wildcards. The percent sign matches any sequence of characters, and the underscore matches exactly one character.

28) What is the IN operator used for?

IN checks whether a value equals any value in a list or in the result of a subquery. It is a compact alternative to writing many OR comparisons on the same column.

29) What does the BETWEEN operator do?

BETWEEN tests whether a value lies within a specified inclusive range, acting like a combination of \geq and \leq on the same column.

30) What does the EXISTS operator check?

EXISTS returns true if a subquery produces at least one row. It is often used with correlated subqueries to check for the presence of related data rather than to retrieve it.

31) How do NULL values behave in comparisons?

NULL represents an unknown or missing value and does not compare equal to anything, not even another NULL. You must use IS NULL or IS NOT NULL to test for it instead of equality operators.

32) What is a join in SQL?

A join links rows from two tables based on a related column, often a primary key and foreign key pair, and returns a result set that includes columns from both tables.

33) What is an inner join?

An inner join returns only rows where the join condition finds matches in both tables. Rows without a matching partner are excluded from the results.

34) What is a left join?

A left join returns all rows from the left table and the matching rows from the right table. If there is no match on the right side, the right table's columns are filled with NULL.

35) What is a right join?

A right join is the reverse of a left join. It returns all rows from the right table and matching rows from the left table, using NULLs for missing left side values.

36) What is a full outer join?

A full outer join returns all rows from both tables. Matching rows are combined, and rows without a match on one side appear with NULLs for the columns from the other table.

37) What is a self join?

A self join is a join where a table is joined to itself using different aliases. It is useful for comparing rows within the same table, such as employees and their managers.

38) What is a cross join?

A cross join returns the Cartesian product of two tables, meaning every row from the first table combined with every row from the second table.

39) How is a join different from a subquery?

A join expands rows horizontally by adding columns from related tables. A subquery produces a separate result that is used as an input value, list, or derived table in the outer query.

40) How is a join different from a UNION?

A join combines tables side by side by adding columns. UNION stacks compatible result sets on top of each other by adding rows.

41) What is a subquery?

A subquery is a query placed inside another SQL statement, such as in SELECT, FROM, or WHERE, whose result is used by the outer statement.

42) What is a correlated subquery?

A correlated subquery is a subquery that refers to columns from the outer query. It is executed once per row of the outer result, making it dependent on that row's values.

43) When would you choose EXISTS instead of IN?

EXISTS is preferred when you only need to know whether related rows exist, not their actual values, and can be faster because the database can stop on the first match.

44) What is the difference between UNION and UNION ALL?

UNION merges two result sets and automatically removes duplicate rows. UNION ALL simply appends the results and leaves duplicates in place, usually with less processing cost.

45) What are INTERSECT and EXCEPT used for?

INTERSECT returns only the rows that appear in both result sets. EXCEPT also called MINUS in some systems returns rows that appear in the first result set but not in the second.

46) What is a temporary table?

A temporary table is a table that exists only for the duration of a session or connection and is used to store intermediate or scratch data during complex operations.

47) What is a view?

A view is a stored SELECT query that behaves like a virtual table. When you query a view, the database runs its underlying query and returns the results as if they were table rows.

48) What are the benefits and limitations of views?

Views simplify access to complex queries, hide implementation details, and restrict which columns users see, but they usually do not store their own data and some views are not easily updatable.

49) What is a materialized view?

A materialized view is a view whose result set is stored physically on disk. It speeds up repeated expensive queries but must be refreshed to stay in sync with the underlying data.

50) What is a stored procedure?

A stored procedure is a named program of SQL statements and optional logic stored in the database that can be executed on demand, often with input parameters.

51) What is a user defined function in SQL?

A user defined function is a routine created by the user that returns a single value or a table and can be used inside SQL expressions or queries.

52) How does a stored procedure differ from a function?

Procedures can perform data changes and do not have to return anything, while functions must return a value and are typically used within SELECT, WHERE, or other expressions.

53) What is a trigger?

A trigger is a special type of stored program that runs automatically when certain events occur on a table, such as insertions, updates, or deletions.

54) What are common reasons to use triggers?

Triggers are often used to maintain audit histories, enforce complex business rules, automatically maintain summary data, or keep related tables synchronized.

55) What is normalization?

Normalization is the process of structuring tables and relationships to minimize redundancy and avoid update anomalies by following a set of formal rules called normal forms.

56) What is first normal form?

First normal form requires that each column holds only a single value per row and that there are no repeating groups or arrays stored in a single column.

57) What is second normal form?

Second normal form states that, in tables with a composite primary key, every non key column must depend on the entire key, not just part of it.

58) What is third normal form?

Third normal form requires that non key columns depend only on the primary key and not on other non key columns, eliminating transitive dependencies.

59) What is denormalization and why would you use it?

Denormalization intentionally combines or duplicates data that would otherwise be split into multiple tables to improve read performance or simplify queries, especially in analytic systems.

60) How do OLTP and OLAP databases differ?

OLTP systems handle high volume, short transactions for day to day operations, while OLAP systems support complex analytical queries over large historical datasets, often for reporting and decision making.

61) What is an index in SQL?

An index is a data structure that allows the database to locate rows quickly using key column values rather than scanning the entire table.

62) What is a clustered index?

A clustered index defines the physical ordering of rows on disk. Because the data is stored in this order, a table normally has only one clustered index.

63) What is a non clustered index?

A non clustered index is a separate structure that stores key values and pointers to the table rows without changing their physical order; multiple non clustered indexes can exist on one table.

64) What is a composite index?

A composite index is an index that includes more than one column. It is useful when queries often filter or sort on the same combination of columns.

65) When can adding an index hurt performance?

Indexes slow down writes because insert, update, and delete operations must also update index entries, and too many or poorly chosen indexes can increase storage and maintenance costs.

66) What is a covering index?

A covering index is an index that contains all the columns a query needs, so the database can answer the query from the index alone without reading the table data.

67) What is a query execution plan?

A query execution plan is the strategy chosen by the database optimizer to run a query, showing steps such as scans, joins, and index usage, often inspectable with EXPLAIN.

68) How can you find a slow query in a database?

You can review slow query logs, use built in monitoring tools, or track queries with long runtimes, then inspect their execution plans and statistics to understand why they are slow.

69) What is the difference between DELETE, TRUNCATE, and DROP?

DELETE removes selected rows and can be filtered; TRUNCATE quickly removes all rows from a table but keeps its structure; DROP removes the table definition and all its data entirely.

70) What is a transaction in SQL?

A transaction is a set of operations that are treated as a single unit of work, so either all of them succeed together or all of them are rolled back.

71) What are the ACID properties?

ACID stands for Atomicity, Consistency, Isolation, and Durability, meaning transactions are all or nothing, preserve rules, behave as if run alone, and stay committed even after failures.

72) What does COMMIT do in a database session?

COMMIT finalizes the current transaction, permanently saving all changes made since the transaction began or since the last commit.

73) What does ROLLBACK do in a database session?

ROLLBACK cancels the current transaction, undoing all changes made since the transaction started or since the last savepoint.

74) What is a SAVEPOINT in a transaction?

A savepoint is a named marker inside a transaction that you can roll back to, undoing changes made after that point while keeping earlier work.

75) What is an isolation level?

An isolation level defines how visible one transaction's uncommitted changes are to other concurrent transactions, controlling issues like dirty reads and non repeatable reads.

76) What are the common isolation levels?

The usual isolation levels are read uncommitted, read committed, repeatable read, and serializable, each offering stronger consistency guarantees with increasing impact on concurrency.

77) What is a deadlock in SQL?

A deadlock occurs when two or more transactions hold locks that the others need and each waits for the other to release them, so none can continue without intervention.

78) What is locking in a database?

Locking is the mechanism by which the database controls simultaneous access to data, preventing conflicting changes but sometimes causing blocking or deadlocks.

79) What is the difference between pessimistic and optimistic concurrency?

Pessimistic concurrency assumes conflicts are likely and uses locks to prevent them before they happen; optimistic concurrency assumes conflicts are rare and checks for conflicts only when changes are committed.

80) What is a data warehouse?

A data warehouse is a large, centralized store of integrated data designed primarily for analytics and reporting rather than for day to day transaction processing.

81) What is a common table expression CTE?

A CTE is a named temporary result defined with WITH at the start of a query that you can reference as if it were a table in the rest of that statement.

82) What is a recursive CTE?

A recursive CTE is a CTE that repeatedly references itself to process hierarchical data, starting from a base set of rows and adding related rows step by step.

83) What is a window function?

A window function calculates a value across a set of related rows, defined by an OVER clause, without collapsing them into a single row as GROUP BY would.

84) What does the ROW_NUMBER function do?

ROW_NUMBER assigns a unique sequential number to each row within a partition of the result set, usually according to a specified ORDER BY.

85) How do RANK, DENSE_RANK, and ROW_NUMBER differ?

RANK assigns the same rank to tied values and leaves gaps in the sequence, DENSE_RANK assigns the same rank to ties but does not leave gaps, and ROW_NUMBER always gives unique numbers with no ties.

86) What are LAG and LEAD used for?

LAG retrieves a value from an earlier row in the same partition, and LEAD retrieves a value from a later row, allowing comparisons between a row and its neighbors.

87) What is the CASE expression in SQL?

CASE is a conditional expression that checks one or more conditions in order and returns different values depending on which condition is true.

88) What does COALESCE do in SQL?

COALESCE returns the first non NULL value from its list of arguments, letting you replace NULL values with a fallback or default value.

89) What are aggregate functions?

Aggregate functions such as COUNT, SUM, AVG, MIN, and MAX compute a single summary value from a set or group of rows.

90) How can you return the top N rows per group?

You can use ROW_NUMBER with PARTITION BY to assign numbers within each group and then filter to rows where the row number is less than or equal to N.

91) How do you implement pagination in SQL?

Pagination is usually done by sorting the rows with ORDER BY and using LIMIT with OFFSET or an equivalent clause, or by using ROW_NUMBER and filtering on a range of row numbers.

92) What is the difference between CHAR and VARCHAR?

CHAR stores fixed length strings and pads shorter ones with spaces, while VARCHAR stores variable length strings and only uses as much space as needed for each value.

93) How do INT, DECIMAL, and FLOAT differ?

INT stores whole numbers, DECIMAL stores exact numeric values with fixed precision and scale which is ideal for money, and FLOAT stores approximate values with possible rounding differences.

94) What are DATE, TIME, and TIMESTAMP data types?

DATE stores just a calendar date, TIME stores only the time of day, and TIMESTAMP stores both date and time and may include fractional seconds and time zone details depending on the database.

95) What is an auto increment or identity column?

An auto increment or identity column automatically generates a new numeric value for each inserted row, commonly used to create simple primary keys.

96) What is SQL injection and how is it prevented?

SQL injection is an attack where user input is inserted into SQL text to change what the query does. It is prevented by using parameterized queries, avoiding string concatenation of input, and validating inputs.

97) What is referential integrity?

Referential integrity ensures that foreign key values in a child table always match existing primary key values in the parent table, preventing broken or orphaned references.

98) What is cascading delete or update?

Cascading actions on a foreign key automatically delete or update child rows when the related parent row is deleted or its key changes, keeping relationships consistent.

99) What is a star schema?

A star schema is a common data warehouse design where a central fact table containing measurements connects to multiple dimension tables that store descriptive attributes such as date or product.

100) What is the difference between logical and physical database design?

Logical design defines tables, columns, keys, and relationships at the model level, while physical design translates that model into concrete implementation details like indexes, partitions, and storage layout on a specific database platform.