**Why Do Business Analysts Need SQL?**

With SQL, a business analyst can work more efficiently. SQL allows you to better understand and interpret documents that contain results from databases, and it enables you to conduct a deeper analysis of the data stored in relational databases.

Knowledge of SQL and relational databases is also very useful for an improved understanding of data flow in information systems, as it makes it easier to create reports and apply them to your needs. In addition, it allows you to better communicate with developers and database administrators.

**Sample SQL Interview Questions and Answers**

When you apply for a business analyst position, it is very likely you will also have to answer technical questions to demonstrate your knowledge. Increasingly, these are SQL-related questions. What are the differences between the SQL questions asked of business analyst candidates and those that are presented to IT candidates? For the former, the questions mainly deal with the specificity of the work to be performed—that is, they primarily concern basic knowledge of SQL and relational databases. These questions often test your understanding of the SELECT statement. However, questions will likely vary from company to company; it all depends on the responsibilities the particular position entails and what the company expects from its employees.

We chose a set of 16 questions that are most commonly asked by employers during interviews for business analyst positions.

**1. What is SQL?**

SQL is short for Structured Query Language and is used to communicate with relational databases. It is the standard language used to retrieve, update, insert, and delete data when working with relational databases.

**2. What do you use to get non-repeated values?**

The DISTINCT keyword is used in the SELECT statement to eliminate repetition of identical data. It is also used in aggregate functions. When DISTINCT is used with only one column or expression, the query will strictly return the unique values for that particular column or expression. Similarly, when DISTINCT is used with multiple columns or expressions, the query will return only the unique combinations of those columns or expressions. Note that the DISTINCT keyword doesn’t ignore the NULL value when sifting through data.

**3. What is the IN operator?**

IN is a conditional operator used in a WHERE clause and is shorthand for multiple OR conditional statements. It tests the expression that precedes it against a list of values that are passed in to the operator, which can either be comma-separated values or a subquery that returns a list of values. If the expression that precedes IN matches any of the elements in the list, the resulting value is TRUE, or 1; otherwise, the value is FALSE, or 0.

**4. What are aggregate functions?**

Aggregate functions perform calculations on a set of values and return a single value. The common aggregate functions are:

* COUNT (counts the number of rows in the table)
* SUM (returns the sum of all values of a numeric column)
* AVG (returns the average of all values of a numeric column)
* MIN (returns the lowest value of a numeric column)
* MAX (returns the highest value of a numeric column).

Aggregate functions are frequently used in combination with the GROUP BY statement.

**5. What is the GROUP BY statement used for?**

GROUP BY is a statement that divides the result for a particular query into groups of rows. It is often used with aggregate functions such as SUM, AVG, MAX, MIN, and COUNT, which calculate information about each group. The SELECT statement returns one row for each group.

**6. In what situations should you use WHERE and HAVING in a statement?**

Though both WHERE and HAVING are used to filter records, there is a subtle difference between the two. The WHERE clause is used to filter records from a result, whereas HAVING is used to filter *groups*. If the two are used together, the WHERE clause is applied first to filter rows; only after grouping finishes is the HAVING clause applied.

**7. What is self JOIN?**

Self JOIN is a query that joins a table with itself. This is used to compare the values of a particular column with other values in the same column of the same table. Self JOIN uses aliases to name the original and duplicate tables.

**8. What is CROSS JOIN?**

CROSS JOIN defines a Cartesian product on the sets of records from two or more joined tables, where the number of rows in the first table is multiplied by the number of rows in the second table.

**9. What Is INNER JOIN?**

INNER JOIN is the most common type of join. It returns all rows that are shared by two tables. Visually, it’s analogous to identifying the overlap, or intersection, between two sets of data.

**10. What is the difference between LEFT JOIN and RIGHT JOIN?**

A LEFT JOIN returns all records from the left table, even when they do not match in the right table. Missing values become NULL. In a similar manner, a RIGHT JOIN returns all records from the right table, even when they do not match those in the left table. Missing values become NULL.

**11. What is the difference between UNION and UNION ALL?**

UNION will omit duplicate records, whereas UNION ALL will include duplicate records. UNION requires the server to do the additional work of removing any duplicates.

**12. What is a subquery?**

A subquery is, as the name suggests, a query within another query. The outer query is called the main query, while the inner query is the subquery. You can think of this in terms of composition of functions. A subquery is always executed first, and its result is then passed on to the main query.

**13. What are the different types of subqueries?**

There are two types of subqueries: correlated and uncorrelated.

An uncorrelated subquery is a independent query whose output is substituted into the main query. A correlated subquery, on the other hand, uses values from the outer query and therefore depends on the outer query. Such a subquery executes repeatedly, once for each row that is selected by the outer query.

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

A primary key is a unique identifier for a particular record in a table. The primary key can’t be NULL. A primary key can be a single column or a combination of columns in a table. Each table can contain only one primary key.

**15. What is a view?**

A view is a virtual table that consists of a subset of data from a table. The content of a view is defined by the query. A view takes up little space because it doesn’t copy all data from the given table but only a subset of data as defined by the view. Note that a view can also display a combination of data from one or more tables. Views allow you to hide the complexity of large data and instead narrow in on areas of interest.

**16. What is a database transaction?**

A transaction is a single logical (atomic) unit of work, in which a sequence of operations (or none) must be executed. A transaction has a defined beginning and end. You can commit or roll back a transaction.