

Sample Viva Questions SQL

1. Are NULL values in a database the same as that of blank space or zero?
 - a. No, a NULL value is very different from that of zero and blank space as it represents a value that is assigned, unknown, unavailable, or not applicable as compared to blank space which represents a character and zero represents a number.
 - b. Example: NULL value in “number_of_courses” taken by a student represents that its value is unknown whereas 0 in it means that the student hasn’t taken any courses

2. Explain the difference between the DELETE and TRUNCATE command in a DBMS.

DELETE command: this command is needed to delete rows from a table based on the condition provided by the WHERE clause.

It deletes only the rows which are specified by the WHERE clause.

It can be rolled back if required.

It maintains a log to lock the row of the table before deleting it and hence it’s slow.

TRUNCATE command: this command is needed to remove complete data from a table in a database. It is like a DELETE command which has no WHERE clause.

It removes complete data from a table in a database.

It can be rolled back even if required. (truncate can be rolled back in some databases depending on their version but it can be tricky and can lead to data loss). Check this link for more details

It doesn’t maintain a log and deletes the whole table at once and hence it’s fast

3. What is meant by normalization and denormalization?
 - a. Normalization is a process of reducing redundancy by organizing the data into multiple tables. Normalization leads to better usage of disk spaces and makes it easier to maintain the integrity of the database.
 - b. Denormalization is the reverse process of normalization as it combines the tables which have been normalized into a single table so that data retrieval becomes faster. JOIN operation allows us to create a denormalized form of the data by reversing the normalization.

4. Explain different types of keys in a database.

a. There are mainly 7 types of keys in a database:

- **Candidate Key:** The candidate key represents a set of properties that can uniquely identify a table. Each table may have multiple candidate keys. One key amongst all candidate keys can be chosen as a primary key. In the below example since studentId and firstName can be considered as a Candidate Key since they can uniquely identify every tuple.
- **Super Key:** The super key defines a set of attributes that can uniquely identify a tuple. Candidate key and primary key are subsets of the super key, in other words, the super key is their superset.
- **Primary Key:** The primary key defines a set of attributes that are used to uniquely identify every tuple. In the below example studentId and firstName are candidate keys and any one of them can be chosen as a Primary Key. In the given example studentId is chosen as the primary key for the student table.
- **Unique Key:** The unique key is very similar to the primary key except that primary key don't allow NULL values in the column but unique keys allow them. So essentially unique keys are primary keys with NULL values.
- **Alternate Key:** All the candidate keys which are not chosen as primary keys are considered as alternate Keys. In the below example, firstname and lastname are alternate keys in the database.
- **Foreign Key:** The foreign key defines an attribute that can only take the values present in one table common to the attribute present in another table. In the below example courseId from the Student table is a foreign key to the Course table, as both, the tables contain courseId as one of their attributes.
- **Composite Key:** A composite key refers to a combination of two or more columns that can uniquely identify each tuple in a table. In the below example the studentId and firstName can be grouped to uniquely identify every tuple in the table.

5. What are Constraints in SQL?

- a. Constraints are used to specify the rules concerning data in the table. It can be applied for single or multiple fields in an SQL table during the creation of the table or after creating using the ALTER TABLE command. The constraints are:
- **NOT NULL** - Restricts NULL value from being inserted into a column.

- CHECK - Verifies that all values in a field satisfy a condition.
- DEFAULT - Automatically assigns a default value if no value has been specified for the field.
- UNIQUE - Ensures unique values to be inserted into the field.
- INDEX - Indexes a field providing faster retrieval of records.
- PRIMARY KEY - Uniquely identifies each record in a table.
- FOREIGN KEY - Ensures referential integrity for a record in another table

6. What are some common clauses used with SELECT query in SQL?

a. Some common SQL clauses used in conjunction with a SELECT query are as follows:

- WHERE clause in SQL is used to filter records that are necessary, based on specific conditions.
- ORDER BY clause in SQL is used to sort the records based on some field(s) in ascending (ASC) or descending order (DESC).

```
SELECT *  
FROM myDB.students  
WHERE graduation_year = 2019  
ORDER BY studentID DESC;
```

- GROUP BY clause in SQL is used to group records with identical data and can be used in conjunction with some aggregation functions to produce summarized results from the database.
- HAVING clause in SQL is used to filter records in combination with the GROUP BY clause. It is different from WHERE, since the WHERE clause cannot filter aggregated records.

```
SELECT COUNT(studentId), country  
FROM myDB.students  
WHERE country != "INDIA"  
GROUP BY country  
HAVING COUNT(studentID) > 5;
```

7. List the different types of relationships in SQL.

- One-to-One - This can be defined as the relationship between two tables where each record in one table is associated with the maximum of one record in the other table.

- One-to-Many & Many-to-One - This is the most commonly used relationship where a record in a table is associated with multiple records in the other table.
- Many-to-Many - This is used in cases when multiple instances on both sides are needed for defining a relationship.
- Self-Referencing Relationships - This is used when a table needs to define a relationship with itself.

8. What is an Alias in SQL?

- An alias is a feature of SQL that is supported by most, if not all, RDBMSs. It is a temporary name assigned to the table or table column for the purpose of a particular SQL query. In addition, aliasing can be employed as an obfuscation technique to secure the real names of database fields. A table alias is also called a correlation name.
- An alias is represented explicitly by the AS keyword but in some cases, the same can be performed without it as well. Nevertheless, using the AS keyword is always a good practice.

```
SELECT A.emp_name AS "Employee" /* Alias using AS keyword */  
       B.emp_name AS "Supervisor"  
FROM employee A, employee B /* Alias without AS keyword */  
WHERE A.emp_sup = B.emp_id;
```

9. What are the TRUNCATE, DELETE and DROP statements?

- DELETE statement is used to delete rows from a table.

```
DELETE FROM Candidates  
WHERE CandidateId > 1000;
```

- TRUNCATE command is used to delete all the rows from the table and free the space containing the table.

```
TRUNCATE TABLE Candidates;
```

- DROP command is used to remove an object from the database. If you drop a table, all the rows in the table are deleted and the table structure is removed from the database.

DROP TABLE Candidates;

10. What is the difference between DROP and TRUNCATE statements?

If a table is dropped, all things associated with the tables are dropped as well. This includes - the relationships defined on the table with other tables, the integrity checks and constraints, access privileges and other grants that the table has. To create and use the table again in its original form, all these relations, checks, constraints, privileges and relationships need to be redefined. However, if a table is truncated, none of the above problems exist and the table retains its original structure.

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

- The TRUNCATE command is used to delete all the rows from the table and free the space containing the table.
- The DELETE command deletes only the rows from the table based on the condition given in the where clause or deletes all the rows from the table if no condition is specified. But it does not free the space containing the table

12. What are the differences between Having and Where clause?

- The Where clause filters rows before grouping. Having clause filters rows after grouping.
- You cannot use aggregate functions in Where clause. In Having clause, you can use aggregate functions.
- WHERE Clause is used to filter the records from the table based on the specified condition.
- HAVING Clause is used to filter record from the groups based on the specified condition.

13. What is primary key, foreign key and unique key?

- Primary key: The primary key is used to uniquely identify each row in a table and does not allow null values.
- Foreign key: A foreign key is one or more columns whose values are based on the primary key values from another table.
- Unique Key: Unique key identifies each row in the table uniquely. Unique key allows null values.

14. What is Correlated query and Nested query?

- Subquery

- When a query is included inside another query, the Outer query is known as Main Query, and Inner query is known as Subquery.
- Nested Query –
 - In Nested Query, Inner query runs first, and only once. Outer query is executed with result from Inner query. Hence, Inner query is used in execution of Outer query.
- Correlated Query
 - In Correlated Query, Outer query executes first and for every Outer query row Inner query is executed. Hence, Inner query uses values from Outer query.

15. What is the advantage of joins over sub-queries?

- The advantage of a join includes that it executes faster. The retrieval time of the query using joins almost always will be faster than that of a subquery. By using joins, you can maximize the calculation burden on the database i.e., instead of multiple queries using one join query.