SQL IMPORTANT QUESTIONS AND ANSWERS

Q1. Why do we need SQL?

Ans: SQL is used to communicate with RDBMS. It is used to store, retrieve, manage and manipulate the data within a database management system.

Q2. Difference between DBMS & RDBMS.

Ans:

DBMS	RDBMS
1. DBMS Stores data as file	1. RDBMS Stores data in tabular form
2. Data is generally stored in hierarchical form or navigational form	2. Data is stored in the form of tables
3. Normalization is not present	3. Normalization is present
4. Uses file system to store data, so there will be no relation between data	4. Uses tables to store the data so there will be some relation between the data
5. It is used for small organization and it deals with small data	5. It is used to handle large amount of data
6. It supports single user	6. It supports multiple users
7. Examples: XML, Windows Registry etc.	7. Example: MySQL, PostgreSQL, Oracle etc.

Q3. Difference between RDBMS & Excel Sheet.

Ans:

RDBMS	EXCEL SHEET
1. RDBMS follows EF Codd rule or relation model	1. Doesn't follow EF Codd rule or relation model
2. It can store huge amount of data	2. Data storage is limited in excel sheet
3. RDBMS have features of data security	3. Doesn't have features of data security
4. Data retrieval is easy	4. Data retrieval isn't easy
5. We have to execute query using commands	5. No need to execute query.
6. Can store meta data	6. Cannot store meta data

Q4. Explain rules of E.F. CODD.

Ans: Rules of Edgar Frank CODD are:

1. The data stored in the cell must be a single value data.
2. In RDBMS we can store everything in the form of tables, including meta data.
3. According to E.F. CODD, we can store data in multiple tables, if needed we can establish connection between two tables using key attributes.
4. We can validate the data entered into the table in two steps:
   a. By assigning the datatypes.
   b. By assigning the constraints.

Q5, Q6. Explain Data-Types.

Ans: Datatypes are used to determine what type or kind of data will be stored in a particular memory location.

Datatypes in SQL:

1. **CHAR**
   a. Can store → A-Z, a-z, 0-9, Special Character (#, $, * etc.)
   b. Size → 2000
   c. Wastage of memory

2. **VARCHAR / VARCHAR2**
   a. Can store → A-Z, a-z, 0-9, Special Character (#, $, * etc.)
   b. Size → 2000, VARCHAR2 → 4000
   c. No wastage of memory

3. **NUMBER**
   a. Can accept two arguments, 1st Precision and 2nd Scale
   b. Precision: Number of digits in numerical place
   c. Scale: Number of digits in decimal place
   d. Size → Precision: 38, Scale: 127

4. **DATE:**
   a. Used to store date
   b. ‘DD-MON-YY’ or ‘DD-MON-YYYY’

5. **LARGE OBJECT**
   a. CLOB (Character Large Object):
      i. Can store characters up to 4GB
   b. BLOB (Binary Large Object):
      i. Can store binary numbers of images, videos, file etc. up to 4GB

Q7, Q8, Q9. Explain types of Constraints.

Ans: Constraints are the conditions that are assigned to a particular column to validate the data.

Types of Constraints:

1. **UNIQUE**
    a. Assigned to a column that cannot accept repeated or duplicated values

2. **NOT NULL**
    a. Assigned to a column that cannot be null or which are mandatory

3. **CHECK**
    a. Assigned to a column for extra validations
    b. Assigned with a condition, if it is true the value gets accepted else rejected

4. **PRIMARY KEY**
    a. Assigned to a column to identify a record uniquely from the table
    b. Characteristics:
        i. Only one primary key
        ii. Cannot accept repeated or duplicated values
        iii. Cannot accept null values
        iv. It is a combination of UNIQUE and NULL
        v. Not mandatory but recommend to have one in table

5. **FOREIGN KEY**
    a. It is used to establish the connection between two tables
    b. Characteristics:
        i. Can have ‘n’ number of foreign keys in a table
        ii. Can accept repeated and duplicated values
        iii. Can accept null
        iv. Not a combination of UNIQUE and NULL
        v. Present in the child table but actually belongs to parent table
        vi. Also referred as “Referential Integrity Constraint”

Q10. Why we use Asterisk (*)?

Ans: To get the all the columns or fields from the table, we use asterisk in select clause.

Q11. Why we use Alias?

Ans: Alias is used to create a temporary or alternative name for a column or expression in result table.

Q12. Difference between UNIQUE and DISTINCT.

Ans:

Unique:

1. It creates a constraint that all values to be inserted must be different from the others.
2. An error can be witnessed if one tries to enter a duplicate value / prevents two records from having identical/same values in a column

Distinct:

1. It is used with SELECT that helps to return distinct or different values in the result table/set
2. Helps to remove duplicate values when retrieving data

Q13. Difference between DISTINCT clause and WHERE clause.

Ans:

DISTINCT	WHERE
1. It is used to remove repeated or duplicated values in result table	1. It is used to filter the records
2. Distinct should be the first argument in select clause	2. Where clause is used after the from clause
3. We can pass column name or expression as an argument	3. We can pass filter condition as an argument
4. We can pass multiple columns	4. We can pass multiple condition in where clause using logical operators

Q14. Explain types of Operators.

Ans:

Types of Operators:

1. **Arithmetic (+, -, *, √)**
2. **Comparison (=, !=)**
3. **Relational (<, >, <=, >=)**
4. **Logical (AND, OR, NOT)**
5. **Concatenation (||)**
6. **Special (IN, NOT IN, BETWEEN, NOT BETWEEN, LIKE, NOT LIKE, IS, IS NOT)**
7. **Subquery (ALL, ANY, EXISTS, NOT EXISTS)**

Q15. Why we use LIKE operator?

Ans: Like operator is used when we need to match the pattern.

Q16. Explain ORDER BY and GROUP BY.

Ans:

ORDER BY:

1. It is used to sort the records in ascending or descending order.
2. Order by clause must be written as last clause in the statement.
3. Order by clause executes after the execution of select clause.
4. By default, order by clause sort the records in ascending order.
5. We can pass column name or expression as an argument in order by clause.

GROUP BY:

1. We use group by to group the records.
2. It executes row-by-row.
3. For group by clause, we can pass column name or expression as an argument
4. We can write group_by_expression along with multirow function in select clause.
5. Group_by_expression: Any column name or expression which is written in group by clause is known as group_by_expression.
6. After the execution of group by clause it creates group and if any clause executes after the group by clause, it executes group_by_group.

Q17. Difference between single-row SUBQUERY & multi-row SUBQUERY.

Ans:

Single Row Subquery:

1. A subquery which returns exactly one output is known as single row subquery.
2. We can use operators such as IN, NOT IN, ALL, ANY in single row subquery.

Multi Row Subquery:

1. A subquery which returns more than one output is known as multi row subquery.
2. We must use operators such as IN, NOT IN, ALL, ANY.

Q18. Difference between Single Row Functions (SRF) & Multi Row Functions (MRF).

Ans:

SRF	MRF
1. It executes row_by_row.	1. Multi row function is also known as group function or aggregate function. -It executes group by group.
2. It takes one input, executes and generates one output and then goes to next input.	2. It takes all the inputs at once aggregates it (combines) and generates one output.
3. If we pass 'n' number of inputs to single row function it returns 'n' number of outputs.	3. If we pass 'n' number of inputs to multi row functions, it returns a single output.

Q19. Explain types of Key-Attributes.

Ans:

Key Attributes: An attribute which is used to identify a record uniquely from the table is called key attribute.

Non-Key Attribute: All the attributes except key attribute are referred as non-key attributes.

Prime-Key Attribute: Among the key attributes an attribute is chosen to be the main attribute to identify the record uniquely from the table.

Non-Prime Key Attribute: All the key attributes except prime key attribute is referred as non-prime key attributes.

Composite Key Attribute: It is a combination of two or more non-key attributes which is used to identify the record uniquely from the table.

Super-Key Attribute: It is the set of all the key attributes.

Foreign-Key Attribute: It behaves as an attribute of another entity to represent the relation.

Q20. What is SUBSTR (), give example.

Ans: It is a single row function. This function is used to extract the part of the string from the given original string.

Syntax: SUBSTR (‘Original_String’, POSITION, [LENGTH])

Example:
Given String = ‘BANGALORE’
SUBSTR (‘BANGALORE’, 1, 2) → BA

Q21. What is Normalization?

Ans: It is a process of decomposing a large table into smaller table to remove the redundancy and anomaly.

Q22. Explain Functional Dependency.

Ans: Let us consider relation ‘R’ with two attributes ‘X’ and ‘Y’ respectively in which attribute ‘X’ determines attribute ‘Y’.
In other word ‘Y’ is dependent on ‘X’, there exists functional dependency.
R→ {X, Y}
X→Y
Y is dependent on X.

Types of Functional Dependency:

1. Total Functional Dependency:
    a. If all the attributes in a relation are determined by a single attribute which is a key attribute, then there exists total functional dependency.
    b. In total functional dependency there are no redundancy and anomaly.
    c. R→{A, B, C, D}, A is Key Attribute.
    d. A→{B, C, D} ➔A→B, A→C, A→D

2. Partial Functional Dependency:
    a. For a partial functional dependency to exist there must be a composite key attribute.
    b. One of the attributes in composite key relation determines another attribute separately and this is known as partial functional dependency.
    c. In partial functional dependency we have redundancy and anomaly.
    d. R→{A, B, C, D}, A&B is Composite Key Attribute
    e. (A, B) → (C, D)
    f. B→{C}

3. Transitive Functional Dependency:
    a. If an attribute is determined by a non-key attribute which internally determined by a key attribute, then there exists transitive functional dependency.
    b. In transitive functional dependency we have redundancy and anomaly.
    c. R→ {A, B, C, D}
    d. A→B
    e. A---->C
    f. B→C

Q23. Explain types of JOINS.

Ans: Joins statement is used to retrieve the data from multiple tables simultaneously.

Types of Joins:

1. **Cartesian Join or Cross Join:** In cartesian join a record from table one will be merged with all the records of table two.

2. **Inner Join or Equi Join:** We use inner join to obtain only the matched records or records which has pair. We use join condition to obtain the matched records.

3. **Outer Join:** In outer join we get the unmatched records along with the matched records.
    a. **Left Outer Join:** In left outer join we get unmatched records of left table along with matched records.
    b. **Right Outer Join:** In right outer join we get unmatched records of right table along with matched records.
    c. **Full Outer Join:** To obtain unmatched records of both tables along with matched records we use full outer join.

4. **Self Join:** Self join is used to join the same two tables or the table itself.

5. **Natural Join:** Whenever there is no table structure, we use natural join. In natural join we won't be writing any join condition. If the table contains similar columns, we get output of inner join and if the table is not having similar columns, we will get the output of cartesian join.

Q24. Explain types of SQL Statements.

Ans: Statements are used to perform CRUD operations in database.

Types of Statements:

1. Data Definition Language (DDL):
    1) CREATE
    2) RENAME
    3) ALTER
    4) TRUNCATE
    5) DROP

2. Data Manipulation Language (DML):
    1) INSERT
    2) UPDATE
    3) DELETE

3. Transaction Control Language (TCL):
    1) COMMIT
    2) SAVEPOINT
    3) ROLLBACK

4. Data Control Language (DCL):
    1) GRANT
    2) REVOKE

5. Data Query Language (DQL):
    1) SELECT
    2) PROJECTION
    3) SELECTION
    4) JOINS

Q25. Difference between TRUNCATE, DROP & DELETE.

Ans:

TRUNCATE	DROP	DELETE
DDL Command.	1. DDL Command.	1. DML Command.
1. Removes all the rows exists in the table.	2. Removes whole table from database.	2. Deletes removes all/single/multiple rows from the table.
2. Where clause cannot be used.	3. Where clause cannot be used.	3. Where clause can be used.