

Basic SQL Interview Questions

Note: *These questions are very important in interview point of view and were asked in companies like myntra, shaadi.com, affine, flutura, musigma, navi, capgemini, ford, flipkart etc.*

Question: What is the difference between DBMS and RDBMS?

Answer: DBMSs are software applications that help you build and maintain databases. RDBMS is a subset of DBMS, and it is a database management system based on the relational model of the DBMS.

Question: Can we embed PL/SQL in SQL? Justify your answers

Answer: PL/SQL is a procedural language, and it has one or more SQL statements in it. So SQL can be embedded in a PL/SQL block; however, PL/SQL cannot be embedded in SQL as SQL executes a single query at a time.

```
DECLARE                                /* this is a PL/SQL block */
```

```
    qty_on_hand  NUMBER(5);
```

```
BEGIN
```

```
    SELECT quantity INTO qty_on_hand FROM inventory    /* this is the SQL  
statement embedded in the PL/SQL block */
```

```
    WHERE product = 'TENNIS RACKET';
```

```
END;
```

Question: What do you mean by data manipulation language - DML?

Answer: DML includes the most common SQL statements to store, modify, delete, and retrieve data. They are SELECT, UPDATE, INSERT, and DELETE.

```
INSERT INTO table_name /*      Insert is a DML statement */
```

```
VALUES (value, value, value ...)
```

```
INSERT INTO customers /*      data being inserted in the table customers      */
```

```
VALUES ('George' , 'Washington' , 'Connecticut')
```

Question: What is a join in SQL? What are the types of joins?

Answer: A join is used to query data from multiple tables based on the relationship between the fields.

There are four types of joins:

- Inner Join

Rows are returned when there is at least one match of rows between the tables.

```
select first_name, last_name, order_date, order_amount
```

```
from customers c
```

```
inner join orders o
```

```
on c.customer_id = o.customer_id
```

```
/*      customers and orders are two tables. Data will be displayed from the  
two tables where the customer_id from customer table matches
```

```
The customer_id from the orders table.      */
```

- Right Join

Right, join returns all rows from the right table and those which are shared between the tables. If there are no matching rows in the left table, it will still return all the rows from the right table.

```
select first_name, last_name, order_date, order_amount
```

```
from customers c
```

```
left join orders o
```

```
on c.customer_id = o.customer_id
```

```
/* customers and orders are two tables. All rows from the Orders table  
is returned with matching rows from the Customers table if any */
```

- Left Join

Left join returns all rows from the Left table and those which are shared between the tables. If there are no matching rows in the right table, it will still return all the rows from the left table.

```
select first_name, last_name, order_date, order_amount
```

```
from customers c
```

```
left join orders o
```

```
on c.customer_id = o.customer_id
```

```
/* customers and orders are two tables. All rows from the customers  
table is returned with matching rows from the orders table if any */
```

- Full Join

Full join return rows when there are matching rows in any one of the tables. This means it returns all the rows from the left-hand side table and all the rows from the right-hand side table.

```
select first_name, last_name, order_date, order_amount
```

```
from customers c
```

```
full join orders o
```

```
on c.customer_id = o.customer_id
```

```
/* customers and orders are two tables. All rows from the Orders table  
and customer table are returned */
```

Question: What is the difference between CHAR and VARCHAR2 datatype in SQL?

Answer: CHAR is used to store fixed-length character strings, and VARCHAR2 is used to store variable-length character strings.

For example, suppose you store the string 'Database' in a CHAR(20) field and a VARCHAR2(20) field.

The CHAR field will use 22 bytes (2 bytes for leading length).

The VARCHAR2 field will use 10 bytes only (8 for the string, 2 bytes for leading length).

Question: Explain SQL constraints?

Answer: Constraints are used to specify the rules of data type in a table.

They can be specified while creating and altering the table.

The following are the constraints in SQL:

- **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.

Question: What is a primary key, a foreign key, and a unique key

Answer:

- The primary key is a field in the table which uniquely identifies a row. It cannot be NULL

Primary Key →

Table : Employee	
Employee_ID	Employee_Name
1	Jhon
2	Alex
3	James
4	Roy
5	Kay

- A foreign key is a field in one table, which is a primary key in another table. A relationship is created between the two tables by referencing the foreign key of one table with the primary key of another table.

In the example below, the employee_id_ref in the salary table is the foreign key.

Table : Employee		Table : Salary			
Employee_ID	Employee_Name	Employee_ID_Ref	Year	Month	Salary
1	Jhon	1	2012	April	30000
2	Alex	1	2012	May	31000
3	James	1	2012	June	32000
4	Roy	2	2012	April	40000
5	Kay	2	2012	May	41000
		2	2012	June	42000

- Unique Key uniquely identifies a record in a table. There can be many unique key constraints defined on a table.

EMP_ID	EMP_NAME	Government_ID
222	Harry	111-203-987

333

Stephan

789-456-123

444

Lan

745-562-321

#BBD0E0 »

In the table above Emp_id is the primary key however Government_id is the unique key. You may want the Government_id to be unique for every employee. Since the data belongs to the government, you may not want it to be the primary key.

Question: What is a trigger?

Answer: Triggers are stored programs that get automatically executed when an event such as INSERT, DELETE, UPDATE(DML) statement occurs. Triggers can also be evoked in response to Data definition statements(DDL) and database operations, for example, SERVER ERROR, LOGON.

```
create trigger dbtrigger
```

```
on database
```

```
for
```

```
create_table,alter_table,drop_table
```

```
as
```

```
print'you can not create ,drop and alter table in this database'
```

```
rollback;
```

```
create trigger emptrigger
```

```
on emp
```

```
for
```

```
insert,update,delete
```

```
as
```

```
print'you can not insert,update and delete this table i'
rollback;
```

Intermediate Level Interview Questions

Question: What are ACID properties in a transaction

Answer: In order to maintain consistency in a database 'before and after' transactions, certain properties are followed. They are

- **Atomicity:** This means the transaction must happen fully and cannot be left midway.
- **Consistency:** To maintain integrity constraints hence valid data enters the database
- **Isolation:** Controls Concurrency
- **Durability:** Once a transaction is committed it remains committed

Question: What is Dateadd SQL?

Answer: Dateadd is a function that is used to add a number to specified part of the date and returns the modified date. Syntax- DATEADD (date_part, value, input_date); Date part can be as below-

date_part	abbreviations
Year	yy, yyyy
Quarter	qq, q
Month	mm, m
dayofyear	dy, y

Day	dd, d
Week	wk, ww
Hour	hh
Minute	mi, n
Second	ss, s
millisecond	ms
microsecond	mcs
nanosecond	ns

Question: What is SAVEPOINT in transaction control?

Answer: A SAVEPOINT is a point in a transaction when you can roll the transaction back to a certain point without rolling back the entire transaction.

```
SQL> SAVEPOINT A
```

```
SQL> INSERT INTO TEST VALUES (1, 'Savepoint A');
```

```
1 row inserted.
```

```
SQL> SAVEPOINT B
```

```
SQL> INSERT INTO TEST VALUES (2, 'Savepoint B');
```

```
1 row inserted.
```

```
SQL> ROLLBACK TO B;
```

```
Rollback complete.
```



```
SQL> SELECT * FROM TEST;
```

```
ID MSG
```

```
-----
```

```
1 Savepoint A
```

Question: What is a Natural Join?

Answer: A Natural join by default is an Inner Join that creates an implicit join based on the common columns in the two tables being joined:

A NATURAL JOIN can be an INNER join, a LEFT OUTER join, or a RIGHT OUTER join. The default is INNER join.

If the tables COUNTRIES and CITIES have two common columns named COUNTRY and COUNTRY_ISO_CODE, the following two SELECT statements are equivalent:

```
SELECT * FROM COUNTRIES NATURAL JOIN CITIES
```

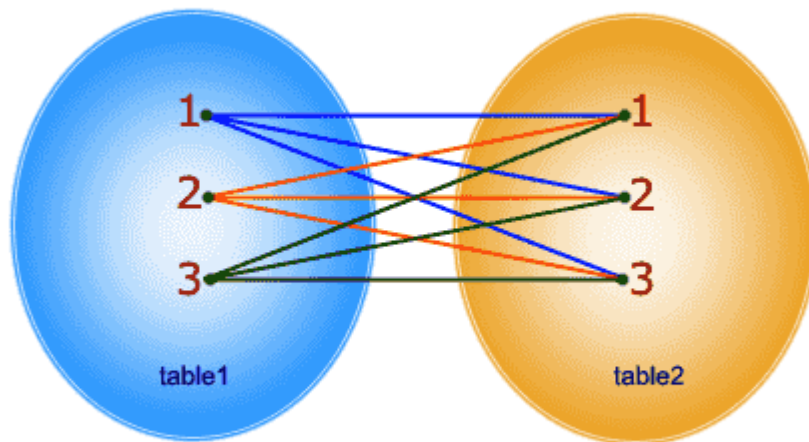
```
SELECT * FROM COUNTRIES JOIN CITIES
```

```
USING (COUNTRY, COUNTRY_ISO_CODE)
```

Question: What is a Cross Join?

Answer: In a SQL cross join, a combination of every row from the two tables is included in the result set. This is also called Cross Product Join. For example, if table A has ten rows and table B has 20 rows, the result set will have $10 * 20 = 200$ rows provided there is NOWHERE clause in the SQL statement.

`SELECT * FROM table1 CROSS JOIN table2;`



In CROSS JOIN, each row from 1st table joins with all the rows of another table.
If 1st table contain x rows and y rows in 2nd one the result set will be $x * y$ rows.

Question: What are the subsets of SQL?

Answer:

The following are the subsets of SQL

1. **DDL(Data Definition Language):** Includes SQL commands like CREATE, ALTER, and DELETE.
2. **DML(Data Manipulation Language):** Accesses and manipulates data
Uses INSERT, UPDATE commands
3. **DCL(Data Control Language):** Controls access to the database. Uses commands like GRANT and REVOKE.

Question: What are the scalar functions in SQL? Give an example?

Answer: Scalar Functions are used to return a single value based on the input values. Scalar Functions are as follows:

- **UCASE():** Converts the specified field in upper case

```
SELECT UCASE("SQL Tutorial is FUN!") AS UppercaseText;
```

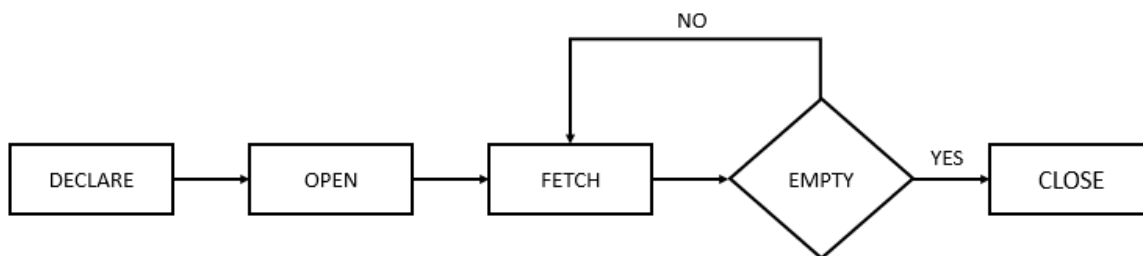
```
UppercaseText
```

```
SQL TUTORIAL IS FUN!
```

- **LCASE():** Converts the specified field in lower case

Question: What is a cursor, and when do you use it?

Answer: A cursor is a database object which is used to manipulate data by traversing row by row in a result set. A cursor is used when you need to retrieve data, one row at a time from a result set and when you need to update records one row at a time.



```
DECLARE @CustomerId INT
```

```
,@Name VARCHAR(100)
```

```
,@Country VARCHAR(100)
```

```
--DECLARE AND SET COUNTER.
```

```
DECLARE @Counter INT
```

```
SET @Counter = 1
```

```
--DECLARE THE CURSOR FOR A QUERY.
```

```
DECLARE PrintCustomers CURSOR READ_ONLY
```

```
FOR
```

```
SELECT CustomerId, Name, Country
```

```
FROM Customers
```

```

--OPEN CURSOR.

OPEN PrintCustomers

--FETCH THE RECORD INTO THE VARIABLES.

FETCH NEXT FROM PrintCustomers INTO

@CustomerId, @Name, @Country

--LOOP UNTIL RECORDS ARE AVAILABLE.

WHILE @@FETCH_STATUS = 0

BEGIN

    IF @Counter = 1

        BEGIN

            PRINT 'CustomerID' + CHAR(9) + 'Name' + CHAR(9) +
CHAR(9) + CHAR(9) + 'Country'

            PRINT '-----'

        END

        --PRINT CURRENT RECORD.          PRINT CAST(@CustomerId AS
VARCHAR(10)) + CHAR(9) + CHAR(9) + CHAR(9) + @Name + CHAR(9) + @Country

        --INCREMENT COUNTER.

        SET @Counter = @Counter + 1

        --FETCH THE NEXT RECORD INTO THE VARIABLES.

        FETCH NEXT FROM PrintCustomers INTO

@CustomerId, @Name, @Country

    END

--CLOSE THE CURSOR.

```

```
CLOSE PrintCustomers
```

```
DEALLOCATE PrintCustomers
```

Question: What is a set-based solution?

Answer: Cursors operate on individual rows, and in the case of a set, it works on a resultant set of data, which could be a table/view or a join of both. The resultant set is an output of a SQL query.

Question: What is a forward cursor?

Answer: Forward cursors support fetching of rows from start to end from a result set. You cannot go to the previous row in the result set.

Question: State one situation where the set-based solution is advantageous over the cursor-based solution

Answer: Set-based solutions provide better performance as they work on a result set and not on one row at a time. They are concise and more readable.

Question: What is normalization and what are the normal forms

Answer: [Normalization](#) is a process in database design to minimize data redundancy and dependency. The database is divided into two or more tables, and relationships are defined between them.

- **First Normal Form:** Every record is unique in a table and is identified by a primary or a composite key

```
StudID  Name  Phonenum
```

```
-----
```

```
1      John  9176612345,9176645698
```

```
2      Susie  9176645789
```

```
3      Jim   9176696325
```

In the above table the field 'phonenum' is a multi-valued attribute, so it is not in 1NF.

Below Table is in 1NF as there is no multi-valued attribute

StudiD	Name	Phonenum
--------	------	----------

1	John	9176612345
---	------	------------

1	John	9176645698
---	------	------------

2	Susie	9176645789
---	-------	------------

3	Jim	9176696325
---	-----	------------

- **Second Normal Form:** The table must be in First Normal Form, and it should have a single column as its primary key.

2NF tries to reduce the redundant data getting stored in memory. To bring the above table in 2NF we split the table into two tables

StudiD	Name
/* student table */	

1. John

2. Susie

3. Jim

StudiD	Phonenum
/* studentphonenum table */	

1	9176612345
---	------------

1	9176645698
---	------------

2	9176645789
---	------------

- **Third Normal Form:** The table must be in Second Normal Form and must have no transitive functional dependencies. I.e., a non-key column must not be dependent on another non-key column within the same table.'

Consider the EMPLOYEE_DETAIL table: This table is not in the third normal form because the fields emp_state

and emp_city depend on emp_zip and not on the primary key emp_id.

EMP_ID	EMP_NAME	EMP_ZIP	EMP_STATE	EMP_CITY
222	Harry	201010	CT	Monroe
333	Stephan	02228	TX	Dallas
444	Lan	060007	IL	Chicago

The above table is split into 2 tables and now the tables are in the third normal form.

EMPLOYEE table:

EMP_ID	EMP_NAME	EMP_ZIP
222	Harry	201010
333	Stephan	02228
444	Lan	060007

EMPLOYEE_ZIP table:

EMP_ZIP	EMP_STATE	EMP_CITY
201010	CT	Monroe
02228	TX	Dallas
060007	IL	Chicago

Question: What is de-normalization, and when do you go for it?

Answer: De-normalization is a technique sometimes used to improve performance so the table design allows redundant data to avoid complex joins. If the application involves heavy read operations, then de-normalization is used at the expense of the write operations performance.

Question: What are clustered indexes and non-clustered indexes?

Answer: A table can have only one clustered index. In this type of index, it reorders the table based on the key values and physically stores them in that order.

The non-clustered index does not have the physical ordering of the data in the table it has a logical order.

```
CREATE CLUSTERED INDEX IX_tblStudent_Gender_Score
```

```
ON student(gender ASC, total_score DESC)
```

The above script creates a clustered index named “IX_tblStudent_Gender_Score” on the student table. This index is created on the “gender” and “total_score” columns. An index that is created on more than one column is called the “composite index”.

A non-clustered index doesn’t sort the physical data inside the table. A non-clustered index is stored in one place, and table data is stored in another place. This allows for more than one non-clustered index per table.


```
CREATE NONCLUSTERED INDEX IX_tblStudent_Name
```

```
ON student (name ASC)
```

The above script creates a non-clustered index on the “name” column of the student table — the index sorts by name in ascending order. The table data and index will be stored in different places.

Question: What is T-SQL?

Answer: It is an extension of SQL(Structured Query Language) developed by Sybase and used by Microsoft.

Question: What are system functions and give one example

Answer: System functions are operations performed on the database server, and values are returned accordingly. Example @@ERROR - Returns 0 if the previous Transact-SQL statement encountered no errors. Otherwise returns an error number.

```
@@ERROR - Returns 0 if the previous Transact-SQL statement encountered no errors.
```

Otherwise returns an error number.

Question: What is a transaction log?

Answer: A log is an audit trail file where the history of actions executed by the DBMS is stored.

Results		Messages					
	Current LSN	Operation	Transaction ID	Parent Transaction ID	Begin Time	Transaction Name	Transaction SID
1	00000020:00000055:0001	LOP_BEGIN_XACT	0000:000002cf	NULL	2013/05/06 12:54:04:977	Backup.CommitDifferentialBase	0x010500000000000515
2	00000020:0000005d:0002	LOP_BEGIN_XACT	0000:000002d0	NULL	2013/05/06 12:54:05:037	CREATE TABLE	0x010500000000000515
3	00000020:0000005d:0003	LOP_BEGIN_XACT	0000:000002d1	0000:000002d0	2013/05/06 12:54:05:037	CMEDCatYukonObject::PreAllocateO	0x010500000000000515
4	00000020:0000005f:000e	LOP_BEGIN_XACT	0000:000002d2	0000:000002d0	2013/05/06 12:54:05:050	Split Page	0x010500000000000515
5	00000020:0000005f:003f	LOP_BEGIN_XACT	0000:000002d3	0000:000002d0	2013/05/06 12:54:05:050	Split Page	0x010500000000000515
6	00000020:0000005f:0058	LOP_BEGIN_XACT	0000:000002d4	0000:000002d0	2013/05/06 12:54:05:053	Split Page	0x010500000000000515
7	00000020:0000005f:0074	LOP_BEGIN_XACT	0000:000002d5	0000:000002d0	2013/05/06 12:54:05:053	Split Page	0x010500000000000515
8	00000020:0000009b:0002	LOP_BEGIN_XACT	0000:000002d6	NULL	2013/05/06 12:54:05:070	INSERT	0x010500000000000515
9	00000020:0000009b:0004	LOP_BEGIN_XACT	0000:000002d7	NULL	2013/05/06 12:54:05:070	Allocate Root	0x010500000000000515
10	00000020:0000009b:0006	LOP_BEGIN_XACT	0000:000002d8	0000:000002d7	2013/05/06 12:54:05:070	AllocFirstPage	0x010500000000000515
11	00000020:000000a0:0002	LOP_BEGIN_XACT	0000:000002d9	NULL	2013/05/06 12:54:05:073	INSERT	0x010500000000000515
12	00000020:000000a0:1:0001	LOP_BEGIN_XACT	0000:000002da	NULL	2013/05/06 12:54:05:077	INSERT	0x010500000000000515
13	00000020:000000a2:0001	LOP_BEGIN_XACT	0000:000002db	NULL	2013/05/06 12:54:05:077	INSERT	0x010500000000000515
14	00000020:000000a3:0001	LOP_BEGIN_XACT	0000:000002dc	NULL	2013/05/06 12:54:05:080	INSERT	0x010500000000000515
15	00000020:000000a4:0001	LOP_BEGIN_XACT	0000:000002dd	NULL	2013/05/06 12:54:05:090	user_transaction	0x010500000000000515
16	00000020:000000a7:0001	LOP_BEGIN_XACT	0000:000002de	NULL	2013/05/06 12:54:05:107	UpdateQPStats	0x010500000000000515
17	00000020:000000aa:0001	LOP_BEGIN_XACT	0000:000002df	NULL	2013/05/06 12:54:05:113	DELETE	0x010500000000000515

Question: How do you maintain database integrity where deletions from one table will automatically cause deletions in another table?

Answer: ON DELETE CASCADE is a command that is used when deletions happen in the parent table, and all child records are automatically deleted, and the child table is referenced by the foreign key in the parent table.

```
CREATE TABLE products
( product_id INT PRIMARY KEY,
  product_name VARCHAR(50) NOT NULL,
  category VARCHAR(25)
);
```

```
CREATE TABLE inventory
( inventory_id INT PRIMARY KEY,
  product_id INT NOT NULL,
  quantity INT,
  min_level INT,
  max_level INT,
  CONSTRAINT fk_inv_product_id
  FOREIGN KEY (product_id)
  REFERENCES products (product_id)
  ON DELETE CASCADE
);
```

The Products table is the parent table and the inventory table is the child table. If a productid is deleted from the parent table all the inventory records for that productid will be deleted from the child table

Question: What is the difference between SQL and MySQL?

Answer: SQL is a structured query language used to access the DBMS whereas MYSQL is an Open Source Relational DBMS.

Question: Can we use TRUNCATE with a WHERE clause?

Answer: No, we cannot use TRUNCATE with the WHERE clause.

Question: Define COMMIT

Answer: When a COMMIT is used in a transaction all changes made in the transaction are written into the database permanently.

```
BEGIN TRANSACTION;
```

```
DELETE FROM HumanResources.JobCandidate
```

```
WHERE JobCandidateID = 13;
```

```
COMMIT TRANSACTION;
```

The above example deletes a job candidate in a SQL server.

Question: What does CHECK CONSTRAINT do?

Answer: Check Constraint limits the values that can enter a column in a database table. It is used as an integrity constraint check.

The following SQL creates a CHECK constraint on the "Age" column when the "Persons" table is created. The CHECK constraint ensures that you can not have any person below 18 years:

The syntax below is in MySQL.

```
CREATE TABLE Persons (
```

```
    ID int NOT NULL,
```

```
    LastName varchar(255) NOT NULL,
```

```
    FirstName varchar(255),
```

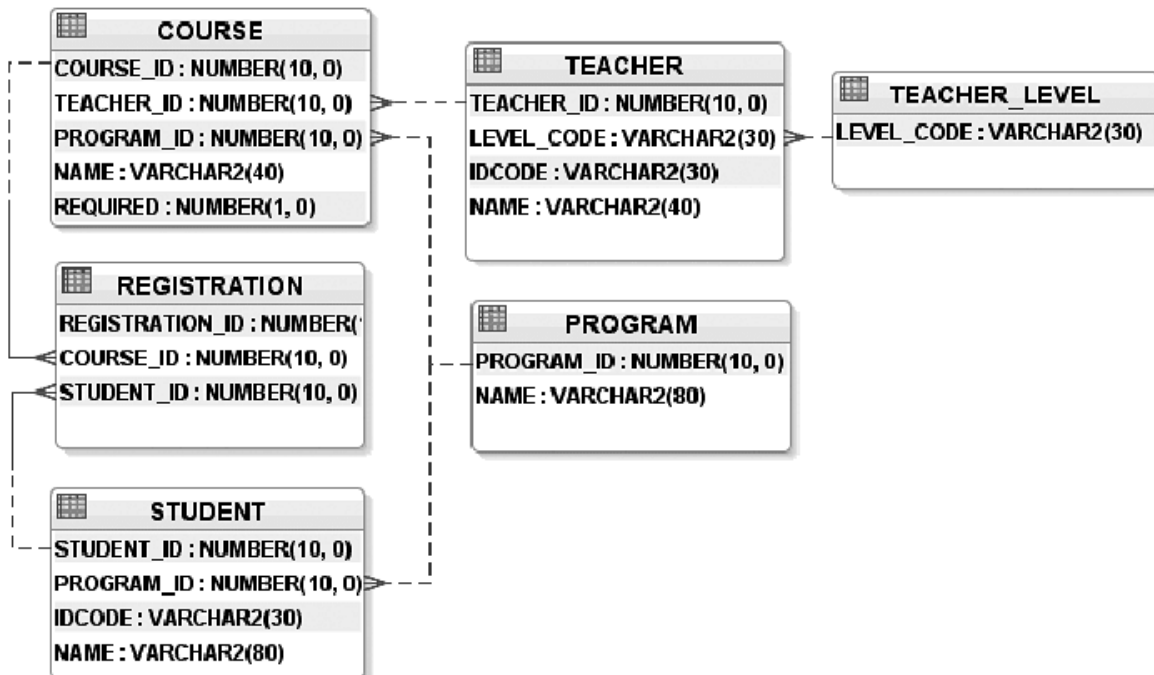
```
    Age int,
```

```
CHECK (Age>=18)
```

```
);
```

Question: What is a schema?

Answer: A schema is a collection of database objects in a database for a particular user/owner. Objects can be tables, views, indices and so on.



Question: How can you create an empty table from an existing table?

Answer:

```
CREATE TABLE NEW_TABLE_NAME AS SELECT [column1, column2 .....column]
FROM EXISTING_TABLE_NAME [WHERE ]
```

Question: What is a composite key?

Answer: When more than one column is used to define the primary key, it is called a composite key.

Here is a SQL syntax to create a composite key in MySQL

```
CREATE TABLE SAMPLE_TABLE
```

```
(COL1 integer,
```

```
COL2 varchar(30),
```

```
COL3 varchar(50),
```

```
PRIMARY KEY (COL1, COL2));
```

Question: How do you sort records in a table?

Answer: The ORDER BY Clause is used to sort records in a table.

```
SELECT * FROM Emp ORDER BY salary;
```

By default, the records are returned in ascending order.

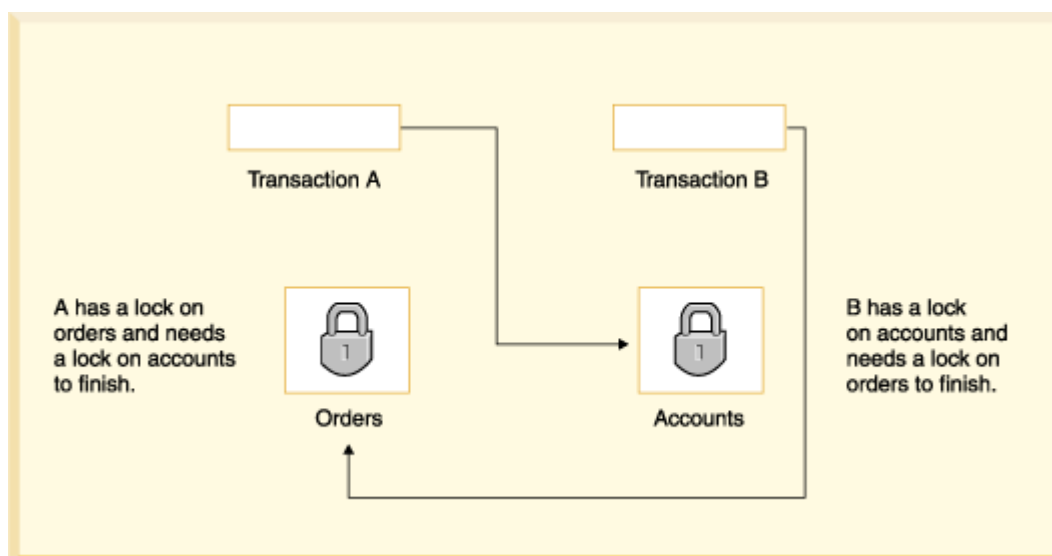
Question: What is a shared lock?

Answer: When two transactions are granted read access to the same data, they are given a shared lock. This enables reading the same data, and data is not updated until the shared lock is released.

Question: What is a deadlock?

Answer: It is an unwanted situation where two or more transactions are waiting indefinitely for one another to release the locks.

Below is an example of a deadlock situation



Question: What is lock escalation?

Answer: Lock escalation is a process of converting row or page locks into table locks. It is an optimization technique used by RDBMS like SQL Server dynamically.

Question: What is SQL injection?

Answer: SQL injection is a code injection technique used to hack data-driven applications.

Question: What are views, and why are they used?

Answer: SQL views are virtual tables created from one or more tables. Views are a subset of data; hence, it can limit the degree of exposure of data from the tables.

The following SQL creates a view that shows all customers from Brazil:

```
CREATE VIEW Brazil_Customers_view AS  
  
SELECT CustomerName, ContactName  
  
FROM Customers  
  
WHERE Country = "Brazil";
```

You can query the view above as follows:

```
SELECT * FROM Brazil_Customers_view;
```

Question: How do we avoid getting duplicate entries in a query?

Answer: The Select DISTINCT is used to get distinct data from tables using a query

The following SQL statement selects only the DISTINCT values from the "Country" column in the "Customers" table:

```
SELECT DISTINCT Country FROM Customers;
```

Question: Give an example of a comparison operator in SQL

Answer: EQUAL TO written as = is used to compare data values

Question: What is a Subquery?

Answer: A subQuery is a SQL query nested into a larger Query.

```
SELECT
    employee_id, first_name, last_name
FROM
    employees
WHERE
    department_id IN (SELECT
        department_id
    FROM
        departments
    WHERE
        location_id = 1700)
ORDER BY first_name , last_name;
```

The query placed within the parentheses is called a subquery. It is also known as an inner query or inner select. The query that contains the subquery is called an outer query or an outer select.

Question: What is a Non-correlated subquery

Answer: A Non-Correlated subquery is an independent query, and the output of subquery is substituted in the main query.

Advance Interview Questions

Question: What is a SYSTEM Privilege?

Answer: Rights are given to a user, usually by the DBA, to perform a particular action on the database schema objects like creating tablespaces.

The following are examples of system privileges that can be granted to users:

- **CREATE TABLE** allows a grantee to create tables in the grantee's schema.
- **CREATE USER** allows a grantee to create users in the database.
- **CREATE SESSION** allows a grantee to connect to an Oracle database to create a user session.

Question: What are Object Privileges?

Answer: An object-level privilege is a permission granted to a database user account or role to perform some action on a database object. These object privileges include **SELECT**, **INSERT**, **UPDATE**, **DELETE**, **ALTER**, **INDEX** on tables, and so on.

The following examples are object privileges that can be granted to users:

- **SELECT ON hr.employees TO myuser**
- **INSERT ON hr.employees TO myuser**

Question: What does the BCP command do?

Answer: The BCP (Bulk Copy) is a utility or a tool that exports/imports data from a table into a file and vice versa

Question: What is a NULL Value field?

Answer: A NULL value is a field with No Value.

Question: What does the VARIANCE function do?

Answer: This function returns the VARIANCE of a set of numbers:

```
CREATE TABLE EMP (EMPNO NUMBER(4) NOT NULL,  
  
                   ENAME VARCHAR2(10) ,  
  
                   JOB VARCHAR2(9) ,  
  
                   SAL NUMBER(7, 2) ,  
  
                   DEPTNO NUMBER(2)) ;  
  
INSERT INTO EMP VALUES (1, 'SMITH', 'CLERK',      800, 20) ;  
  
INSERT INTO EMP VALUES (2, 'ALLEN', 'SALESMAN', 1600, 30) ;  
  
INSERT INTO EMP VALUES (3, 'WARD', 'SALESMAN', 1250, 30) ;  
  
INSERT INTO EMP VALUES (4, 'JONES', 'MANAGER', 2975, 20) ;  
  
INSERT INTO EMP VALUES (5, 'MARTIN', 'SALESMAN', 1250, 30) ;  
  
INSERT INTO EMP VALUES (6, 'BLAKE', 'MANAGER', 2850, 30) ;  
  
INSERT INTO EMP VALUES (7, 'CLARK', 'MANAGER', 2850, 10) ;  
  
INSERT INTO EMP VALUES (8, 'SCOTT', 'ANALYST', 3000, 20) ;  
  
INSERT INTO EMP VALUES (9, 'KING', 'PRESIDENT', 3000, 10) ;  
  
INSERT INTO EMP VALUES (10, 'TURNER', 'SALESMAN', 1500, 30) ;  
  
INSERT INTO EMP VALUES (11, 'ADAMS', 'CLERK', 1500, 20) ;  
  
SQL> SELECT VARIANCE(sal)  
  
2 FROM emp;  
  
VARIANCE (SAL)  
  
-----
```

759056.818

Question: What is the role of GRANT and REVOKE commands?

Answer: The GRANT command enables privileges on the database objects and the REVOKE command removes them. They are DCL commands

```
GRANT CREATE ANY TABLE TO username
```

```
GRANT sysdba TO username
```

```
GRANT DROP ANY TABLE TO username
```

```
REVOKE CREATE TABLE FROM username
```

Question: What is a UNION operator?

Answer: The UNION operator combines the results of two or more Select statements by removing duplicate rows. The columns and the data types must be the same in the SELECT statements.

```
SELECT City FROM Customers
```

```
UNION
```

```
SELECT City FROM Suppliers
```

```
ORDER BY City;
```

Question: Where are stored procedures stored in the database?

Answer: [Stored Procedures](#) are stored in the Data Dictionary of the Database.

Question: Can we call Stored Procedure inside Stored Procedure?

Answer: Yes, we can call a stored procedure from another stored procedure. For example, Procedure2 is the procedure which is called Procedure1. Both Procedure1 and Procedure2 can have business logic implemented in it.

```
Create PROCEDURE Procedure1
```

```
AS BEGIN
```

```
Exec Procedure2
```

```
END
```

Question: Does the data stored in the stored procedure increase access time or execution time? Explain

Answer: Data stored in stored procedures can be retrieved much faster than the data stored in the SQL database. Data can be precompiled and stored in stored procedures. This reduces the time gap between query and compiling as the data has been pre-compiled and stored in the procedure.

Question: Can a Stored Procedure contain a return value?

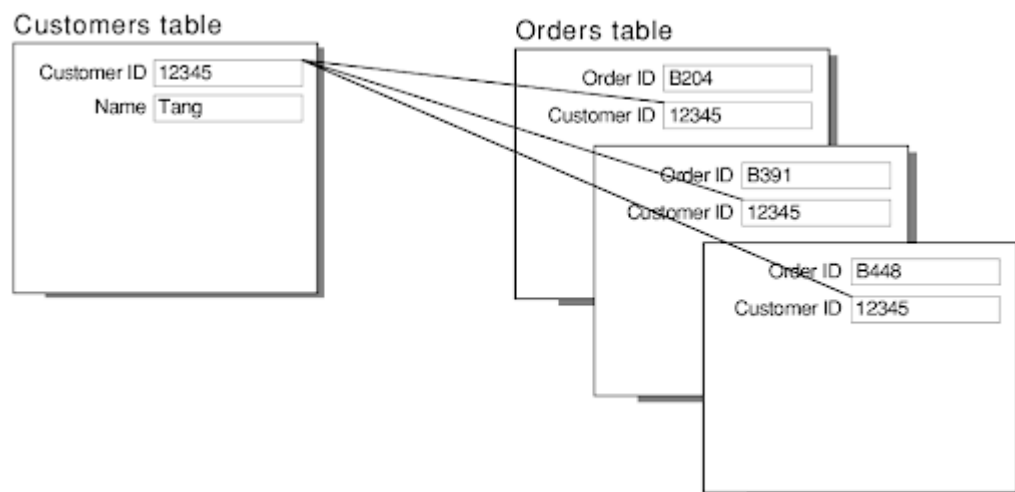
Answer: Procedures may or may not return values.

Question: Can a View be active if the Base table is dropped?

Ans: No, the view cannot be active if the parent table is dropped.

Question: What is a One-Many Relationship in SQL?

Answer: In a One-Many relationship, a record in One Table can be associated or related to Many records in another table.



Question: Distinguish between a table and a field in SQL.

Answer: The collection of data organized in the form of columns and rows refers to the table. The number of columns in a table refers to the field.

Table: Employee_Details

Fields: Emp_id, Emp_name, Emp_dept, Emp_salary

Question: What is data integrity?

Answer: Data integrity defines the accuracy, consistency, and reliability of data that is stored in the database.

There are four kinds of data integrity:

1. Row integrity
2. Column integrity
3. Referential integrity
4. User-defined integrity

Question: What are entities and relationship

Answer:

- **Entity:** A person, place, or any real-world thing that can be represented as a table is called an entity.

Example: Employee table represents the details of an employee in an organization.

- **Relationship:** Relationship defines the dependency that entities share amongst each other.

Example: Employee name, id, salary might belong to the same or different tables.

Question: Write the SQL query to get the current date.

Answer:

```
SELECT CURDATE () ;
```

Question: What is the TRUNCATE command? How is it different from the DELETE command?

Answer:

DELETE	TRUNCATE
DML command	DDL command
We can use WHERE clause	We cannot use WHERE clause
Deletes a row from the table.	Deletes all rows from the table.
We can rollback.	We cannot rollback.

Question: What is the difference between null, zero and blank space?

Answer: NULL refers to a value that is unknown, not available, unapplicable or unassigned. While zero is a number and blank space is treated as a character.

Question: Which function is used to return the remainder in a division operator in SQL?

Answer: The MOD function returns the remainder in the division operation.

Question: What are case manipulation functions?

Answer: Case manipulation functions converts the existing data in the table to lower, upper or mixed case characters.

Question: What are the different case manipulation functions in SQL?

Answer:

- 1. LOWER:** converts all the characters to lowercase.
- 2. UPPER:** converts all the characters to uppercase.
- 3. INITCAP:** converts initial character of each word to uppercase

Question: What are the character manipulation functions?

Answer: Character manipulation functions alter, extract and change the character string.

Question: What are the different character manipulation functions?

Answer:

1. **CONCAT:** joins two or more string values.
2. **SUBSTR:** extracts string of a specific length.
3. **LENGTH:** returns the length of the string
4. **INSTR:** returns the position of the specific character.
5. **LPAD:** padding of the left-side character value for right-justified value.
6. **RPAD:** padding of right-side character value for left-justified value.
7. **TRIM:** removes the defined character from beginning and end or both.
8. **REPLACE:** replaces a specific sequence of characters with another sequence of characters.

Question: Define inconsistent dependency.

Answer: The difficulty of accessing data as the path may be broken or missing defines inconsistent dependency. Inconsistent dependency enables users to search for data in the wrong different table which afterward results in an error as an output.

Question: What are the different operators in SQL?

Answer:

1. **Arithmetic**
2. **Comparison**
3. **Logical**

Question: What are GROUP functions? Why do we need them?

Answer: Group functions work on a set of rows and return a single result per group. The popularly used group functions are AVG, MAX, MIN, SUM, VARIANCE, COUNT

Question: Distinguish between BETWEEN and IN conditional operators.

Answer: BETWEEN- Displays the rows based on range of values

IN- Checks for values contained in a specific set of values.

Example:

```
SELECT * FROM Students where ROLL_NO BETWEEN 10 AND 50;
```

```
SELECT * FROM students where ROLL_NO IN (8,15,25);
```

Question: What is the MERGE statement?

Answer: The statement enables conditional updates or inserts into the table. It updates the row if it exists or inserts the row if it does not exist.