Database/SQL Interview Questions and Answers

1. What is DBMS?

The database management system is a collection of programs that enables user to store, retrieve, update and delete information from a database.

---

2. What is RDBMS?

Relational Database Management system (RDBMS) is a database management system (DBMS) that is based on the relational model. Data from relational database can be accessed or reassembled in many different ways without having to reorganize the database tables. Data from relational database can be accessed using an API, Structured Query Language (SQL).

---

3. What is SQL?

Structured Query Language (SQL) is a language designed specifically for communicating with databases. SQL is an ANSI (American National Standards Institute) standard.

---

4. What are the different types of SQL's statements?

This is one of the most frequently asked SQL Interview Questions for freshers. SQL statements are broadly classified into three. They are

1. **DDL** – **Data Definition Language:** DDL is used to define the structure that holds the data. For example, Create, Alter, Drop and Truncate table.
2. **DML**– **Data Manipulation Language:** DML is used for manipulation of the data itself. Typical operations are Insert, Delete, Update and retrieving the data from the table. The Select statement is considered as a limited version of the DML, since it can't change the data in the database. But it can perform operations on data retrieved from the DBMS, before the results are returned to the calling function.
3. **DCL**– **Data Control Language:** DCL is used to control the visibility of data like granting database access and set privileges to create tables, etc. Example - Grant, Revoke access permission to the user to access data in the database.

Types of SQL Commands			
DDL	**DML**	**DCL**	**TCL**
CREATE	SELECT		
ALTER	INSERT	GRANT	COMMIT
DROP	UPDATE	REVOKE	ROLLBACK
TRUNCATE	DELETE		SAVEPOINT
RENAME	MERGE		

---

5. What are the Advantages of SQL?

1. **SQL is not a proprietary language** used by specific database vendors. Almost every major DBMS supports SQL, so learning this one language will enable programmers to interact with any database like ORACLE, SQL, MYSQL etc.
2. **SQL is easy to learn**. The statements are all made up of descriptive English words, and there aren't that many of them.
3. SQL is actually a very powerful language and by using its language elements you can perform very **complex and sophisticated database operations**.

---

6. what is a field in a database?

A field is an area within a record reserved for a specific piece of data. **Examples**: Employee Name, Employee ID, etc.

---

7. What is a Record in a database?

A record is the collection of values / fields of a specific entity: i.e. an Employee, Salary etc.

---

8. What is a Table in a database?

A table is a collection of records of a specific type. For example, employee table, salary table etc.

---

9. What is a database transaction?

Database transaction takes database from one consistent state to another. At the end of the transaction the system must be in the prior state if the transaction fails or the status of the system should reflect the successful completion if the transaction goes through.

---

10. What are properties of a transaction?

Expect this SQL Interview Questions as a part of an any interview, irrespective of your experience. Properties of the transaction can be summarized as ACID Properties.

1. Atomicity

A transaction consists of many steps. When all the steps in a transaction get completed, it will get reflected in DB or if any step fails, all the transactions are rolled back.

2. Consistency

The database will move from one consistent state to another, if the transaction succeeds and remain in the original state, if the transaction fails.

3. Isolation

Every transaction should operate as if it is the only transaction in the system.

4. Durability

Once a transaction has completed successfully, the updated rows/records must be available for all other transactions on a permanent basis.

---

11. What is a Database Lock?

DB lock tells a transaction, if the data item in questions is currently being used by other transactions.

---

12. What are the type of locks?

1. Shared Lock

When a shared lock is applied on data item, other transactions can only read the item, but can't write into it.

2. Exclusive Lock

When an exclusive lock is applied on data item, other transactions can't read or write into the data item.

Database Normalization Interview Questions

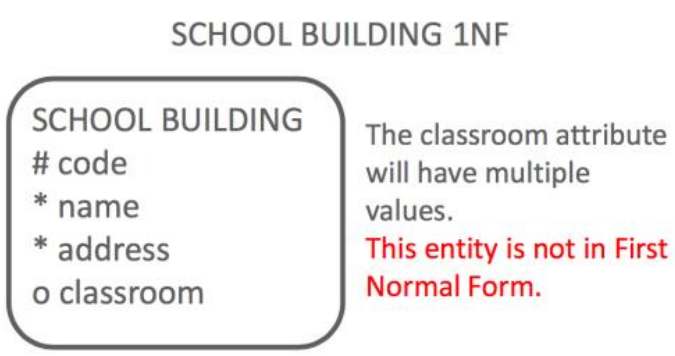
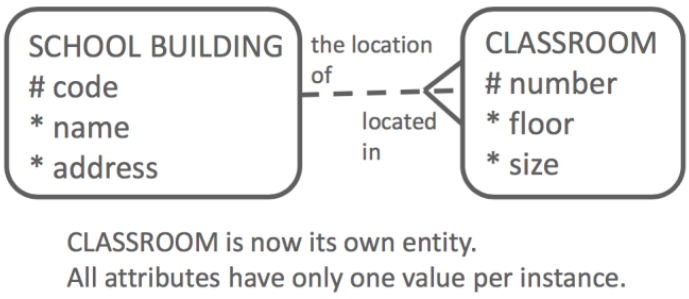
13. What are the different types of normalization?

In database design, we start with one single table, with all possible columns. A lot of redundant data would be present since it's a single table. **The process of removing the redundant data, by splitting up the table in a well-defined fashion is called normalization.**

1. First Normal Form (1NF)

For a table to be in the First Normal Form, it should follow the following 4 rules:

1. It should only have single(atomic) valued attributes/columns.
2. Values stored in a column should be of the same domain
3. All the columns in a table should have unique names.
4. And the order in which data is stored, does not matter.

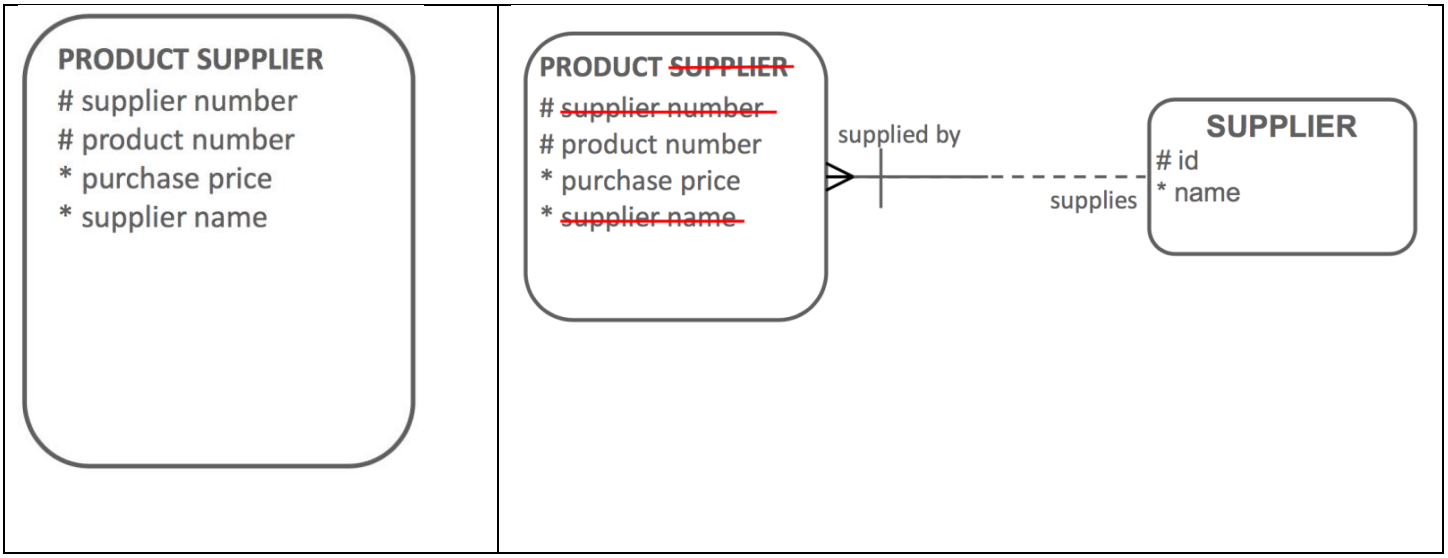
Without 1NF	With 1NF
	

2. Second Normal Form (2NF)

For a table to be in the Second Normal Form,

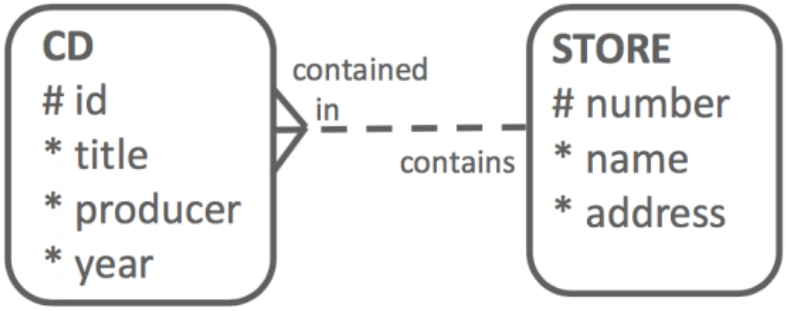
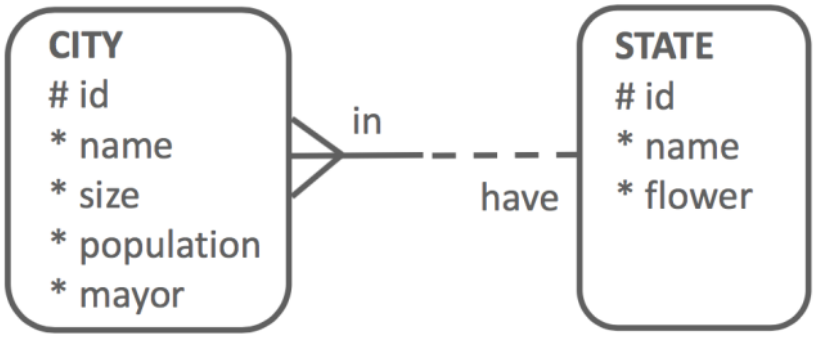
1. It should be in the First Normal form.
2. And, it should not have Partial Dependency.

Without 2NF	With 2NF



3. Third Normal Form (3NF)
A table is said to be in the Third Normal Form when,

1. It is in the Second Normal form.
2. And, it doesn't have Transitive Dependency.

No.	Without 3NF	With 3NF
Example 1		
Example 2		

Database Keys and Constraints SQL Interview Questions

14. What is a primary key?

A primary key is a column whose values **uniquely identify every row** in a table. Primary key values can never be reused. If a row is deleted from the table, its primary key may not be assigned to any

new rows in the future. To define a field as primary key, following conditions had to be met:
1. No two rows can have the same primary key value.
2. Every row must have a primary key value.
3. The primary key field cannot be null.
4. Value in a primary key column can never be modified or updated, if any foreign key refers to that primary key.

---

15. What is a Composite Key?

A Composite primary key is a type of candidate key, which represents a set of columns whose values uniquely identify every row in a table.
For example - if "Employee_ID" and "Employee Name" in a table is combined to uniquely identify a row it's called a Composite Key.

16. What is a Composite Primary Key?

A Composite primary key is a set of columns whose values uniquely identify every row in a table. What it means is that, a table which contains composite primary key will be indexed based on the columns specified in the primary key. This key will be referred in Foreign Key tables.
For example - if the combined effect of columns, "Employee_ID" and "Employee Name" in a table is required to uniquely identify a row, it's called a Composite Primary Key. In this case, both the columns will be represented as primary key.

---

17. What is a Foreign Key?

When a "one" table's primary key field is added to a related "many" tables in order to create the common field which relates the two tables, it is called a foreign key in the "many" tables.
For example, the salary of an employee is stored in salary table. The relation is established via foreign key column "Employee_ID_Ref" which refers "Employee_ID" field in the Employee table.

---

18. What is a Unique Key?

Unique key is same as primary with the difference being the existence of null. Unique key field allows one value as NULL value.

SQL Insert, Update and Delete Commands Interview Questions

19. Define SQL Insert Statement?

SQL INSERT statement is used to add rows to a table. For a full row insert, SQL Query should start with "insert into "statement followed by table name and values command, followed by the values that need to be inserted into the table. The insert can be used in several ways:
1. To insert a single complete row. **2. To insert a single partial row.**

Example:

```
INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)
VALUES ('Cardinal', 'Tom B. Erichsen', 'Skagen 21', 'Stavanger', '4006', 'Norway');
```

---

20. Define SQL Update Statement?

SQL Update is used to update data in a row or set of rows specified in the filter condition. The basic format of an SQL UPDATE statement is, Update command followed by table to be updated and SET

command followed by column names and their new values followed by filter condition that determines which rows should be updated.

1. **Example:**
```
UPDATE Supplier SET City = 'Oslo', Phone = '(0)1-953530' WHERE Id = 15
```

---

21. Define SQL Delete Statement?

SQL Delete is used to delete a row or set of rows specified in the filter condition. The basic format of an SQL DELETE statement is, DELETE FROM command followed by table name followed by filter condition that determines which rows should be updated.

Example:
```
DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';
```

---

22. What are wild cards used in database for Pattern Matching?

SQL Like operator is used for pattern matching. SQL 'Like' command takes more time to process. So before using "like" operator, consider suggestions given below on when and where to use wild card search.

1) Don't overuse wild cards. If another search operator will do, use it instead.

2) When you do use wild cards, try not to use them at the beginning of the search pattern, unless absolutely necessary. Search patterns that begin with wild cards are the slowest to process.

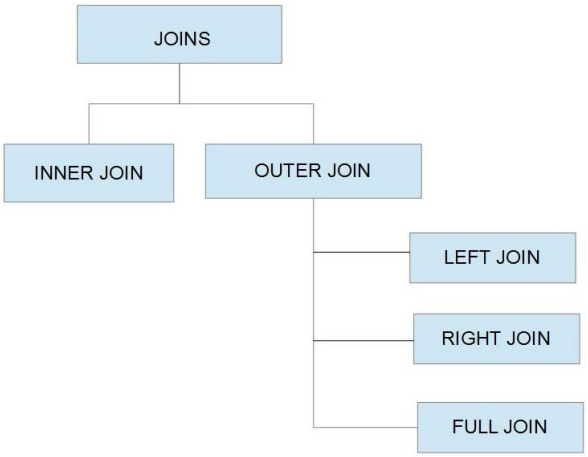
3) Pay careful attention to the placement of the wild card symbols. If they are misplaced, you might not return the data you intended.

Example:
```
SELECT * FROM Customers WHERE CustomerName LIKE '%or%';
```

SQL Joins Interview Questions and answers

23. Define Join and explain different type of joins?

Another frequently asked SQL Interview Questions on Joins. In order to avoid data duplication, data is stored in related tables. **Join** keyword is used to fetch data from related tables. "Join" return rows when there is at least one match in both tables. Type of joins are



Right Join

Return all rows from the right table, even if there are no matches in the left table.

Consider the following two tables, **Table 1** − CUSTOMERS Table is as follows.

```
+----+----------+-----+-----------+----------+
| ID | NAME     | AGE | ADDRESS   | SALARY   |
+----+----------+-----+-----------+----------+
|  1 | Ramesh   |  32 | Ahmedabad |  2000.00 |
|  2 | Khilan   |  25 | Delhi     |  1500.00 |
|  3 | kaushik  |  23 | Kota      |  2000.00 |
|  4 | Chaitali |  25 | Mumbai    |  6500.00 |
|  5 | Hardik   |  27 | Bhopal    |  8500.00 |
|  6 | Komal    |  22 | MP        |  4500.00 |
|  7 | Muffy    |  24 | Indore    | 10000.00 |
+----+----------+-----+-----------+----------+
```

Table 2 − ORDERS Table is as follows.

```
+-----+---------------------+-------------+--------+
|OID  | DATE                | CUSTOMER_ID | AMOUNT |
+-----+---------------------+-------------+--------+
| 102 | 2009-10-08 00:00:00 |           3 |   3000 |
| 100 | 2009-10-08 00:00:00 |           3 |   1500 |
| 101 | 2009-11-20 00:00:00 |           2 |   1560 |
| 103 | 2008-05-20 00:00:00 |           4 |   2060 |
+-----+---------------------+-------------+--------+
```

Now, let us join these two tables using the RIGHT JOIN as follows.

```
SELECT ID, NAME, AMOUNT, DATE   FROM CUSTOMERS
RIGHT JOIN ORDERS
ON CUSTOMERS.ID = ORDERS.CUSTOMER_ID;
```

This would produce the following result −

```
+------+----------+--------+---------------------+
| ID   | NAME     | AMOUNT | DATE                |
+------+----------+--------+---------------------+
|    3 | kaushik  |   3000 | 2009-10-08 00:00:00 |
|    3 | kaushik  |   1500 | 2009-10-08 00:00:00 |
|    2 | Khilan   |   1560 | 2009-11-20 00:00:00 |
|    4 | Chaitali |   2060 | 2008-05-20 00:00:00 |
+------+----------+--------+---------------------+
```

Left Join

Return all rows from the left table, even if there are no matches in the right table.

Consider two tables CUSTOMERS and ORDERS from previous example,

Now, let us join these two tables using the LEFT JOIN as follows.

```
SELECT ID, NAME, AMOUNT, DATE
FROM CUSTOMERS
LEFT JOIN ORDERS
ON CUSTOMERS.ID = ORDERS.CUSTOMER_ID;
```

This would produce the following result −

```
+----+----------+--------+---------------------+
| ID | NAME     | AMOUNT | DATE                |
+----+----------+--------+---------------------+
|  1 | Ramesh   |   NULL | NULL                |
|  2 | Khilan   |   1560 | 2009-11-20 00:00:00 |
|  3 | kaushik  |   3000 | 2009-10-08 00:00:00 |
|  3 | kaushik  |   1500 | 2009-10-08 00:00:00 |
|  4 | Chaitali |   2060 | 2008-05-20 00:00:00 |
|  5 | Hardik   |   NULL | NULL                |
|  6 | Komal    |   NULL | NULL                |
|  7 | Muffy    |   NULL | NULL                |
+----+----------+--------+---------------------+
```

Full Join

Return rows when there is a match in one of the tables.

Consider two tables CUSTOMERS and ORDERS from previous example,

Now, let us join these two tables using FULL JOIN as follows.

```
SELECT ID, NAME, AMOUNT, DATE   FROM CUSTOMERS
FULL JOIN ORDERS
ON CUSTOMERS.ID = ORDERS.CUSTOMER_ID;
```

This would produce the following result −

```
+------+----------+--------+---------------------+
| ID   | NAME     | AMOUNT | DATE                |
+------+----------+--------+---------------------+
|    1 | Ramesh   |   NULL | NULL                |
|    2 | Khilan   |   1560 | 2009-11-20 00:00:00 |
|    3 | kaushik  |   3000 | 2009-10-08 00:00:00 |
|    3 | kaushik  |   1500 | 2009-10-08 00:00:00 |
|    4 | Chaitali |   2060 | 2008-05-20 00:00:00 |
|    5 | Hardik   |   NULL | NULL                |
|    6 | Komal    |   NULL | NULL                |
|    7 | Muffy    |   NULL | NULL                |
|    3 | kaushik  |   3000 | 2009-10-08 00:00:00 |
|    3 | kaushik  |   1500 | 2009-10-08 00:00:00 |
|    2 | Khilan   |   1560 | 2009-11-20 00:00:00 |
|    4 | Chaitali |   2060 | 2008-05-20 00:00:00 |
+------+----------+--------+---------------------+
```

If your Database does not support FULL JOIN (MySQL does not support FULL JOIN), then you can use **UNION ALL** clause to combine these two JOINS as shown below.

```
SELECT ID, NAME, AMOUNT, DATE    FROM CUSTOMERS
LEFT JOIN ORDERS
ON CUSTOMERS.ID = ORDERS.CUSTOMER_ID
UNION ALL
SELECT ID, NAME, AMOUNT, DATE   FROM CUSTOMERS
RIGHT JOIN ORDERS
ON CUSTOMERS.ID = ORDERS.CUSTOMER_ID
```

https://stackoverflow.com/questions/38549/what-is-the-difference-between-inner-join-and-outer-join

24. What is Self-Join?

Self-join is query used to **join a table to itself**. Aliases should be used for the same table comparison.

```
SELECT a.emp_id AS "Emp_ID", a.emp_name AS "Employee Name", b.emp_id AS
"Supervisor ID", b.emp_name AS "Supervisor Name" FROM employee a, employee b
WHERE a.emp_supv = b.emp_id;
```

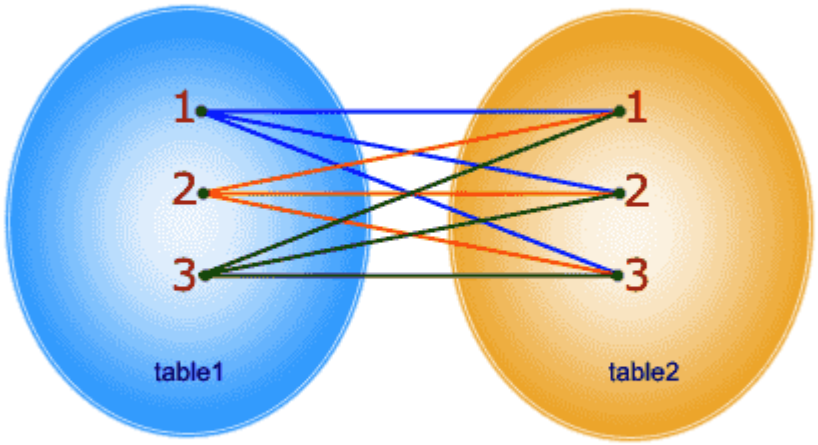
EMP_ID	EMP_NAME	DT_OF_JOIN	EMP_SUPV
20051	Vijes Setthi	15-JUN-09	-
20073	Unnath Nayar	09-AUG-10	20051
20064	Rakesh Patel	23-OCT-09	20073
20069	Anant Kumar	03-DEC-08	20051
20055	Vinod Rathor	27-NOV-09	20051
20075	Mukesh Singh	25-JAN-11	20073

Emp_ID	Employee Name	Supervisor ID	Supervisor Name
20055	Vinod Rathor	20051	Vijes Setthi
20069	Anant Kumar	20051	Vijes Setthi
20073	Unnath Nayar	20051	Vijes Setthi
20075	Mukesh Singh	20073	Unnath Nayar
20064	Rakesh Patel	20073	Unnath Nayar

25. What is Cross Join?

Cross Join will return all records where each row from the first table is combined with each row from the second table. https://www.w3resource.com/sql/joins/cross-join.php

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.

Database Views Interview Questions

26. What is a view?

The views are virtual tables. Unlike tables that contain data, views simply contain queries that dynamically retrieve data when used.

---

27. What is a materialized view?

Materialized views are also a view but are disk based. **Materialized views** get updates on specific duration, based upon the interval specified in the query definition. We can index materialized view.

---

28. What are the advantages and disadvantages of views in a database?

Advantages:

1. Views don't store data in a physical location.
2. The view can be used to hide some of the columns from the table.
3. Views can provide Access Restriction, since data insertion, update and deletion are not possible with the view.

Disadvantages:

1. When a table is dropped, associated view become irrelevant.
2. Since the view is created when a query requesting data from view is triggered, it’s a bit slow.
3. When views are created for large tables, it occupies more memory.

---

29. What is a stored procedure?

Stored Procedure is a function which contains a collection of SQL Queries. The procedure can take inputs, process them and send back output.

---

30. What are the advantages of a stored procedure?

Stored Procedures are precompiled and stored in the database. This enables the database to execute the queries much faster. Since many queries can be included in a stored procedure, round trip time to execute multiple queries from source code to database and back is avoided.

---

31. What is a trigger?

Database triggers are sets of commands that get executed when an event (Before Insert, After Insert, On Update, on delete of a row) occurs on a table, views.

---

32. Explain the difference between DELETE, TRUNCATE and DROP commands?

Once **delete operation** is performed, Commit and Rollback can be performed to retrieve data.
Once the **truncate** statement is executed, Commit and Rollback statement cannot be performed.
Where condition can be used along with delete statement but it can't be used with truncate statement.
Drop command is used to drop the table or keys like primary, foreign from a table.

```
DELETE FROM Customers WHERE OrderId > 1000;

TRUNCATE TABLE Customers;

DROP TABLE Customers;
```

33. What is the difference between Cluster and Non-cluster Index?

A **clustered index** reorders the way records in the table are physically stored. There can be only one clustered index per table. It makes data retrieval faster.

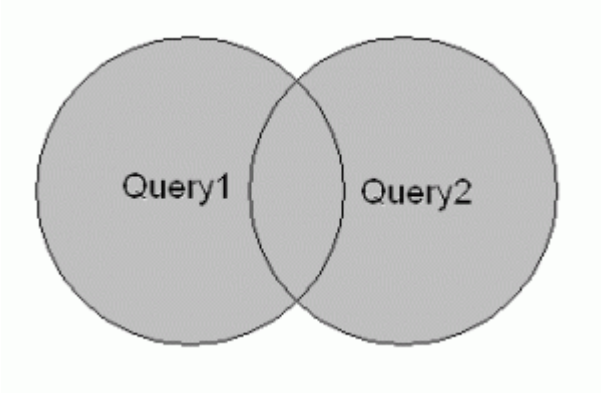
A **non-clustered index** does not alter the way it was stored but creates a completely separate object within the table. As a result, insert and update command will be faster.

---

34. What is Union, minus and Interact commands?

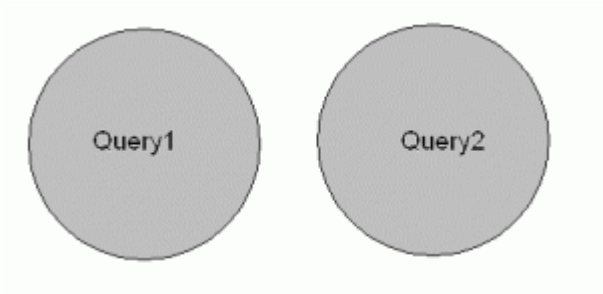
MINUS operator is used to return rows from the first query but not from the second query.
NTERSECT operator is used to return rows returned by both the queries.

UNION: It returns a union of two select statements. It is returning unique (distinct) values of them. It removes duplicate values.



SELECT * FROM table1 UNION SELECT * FROM table2;

UNION ALL Similar to UNION just that UNION ALL returns also the duplicated values.

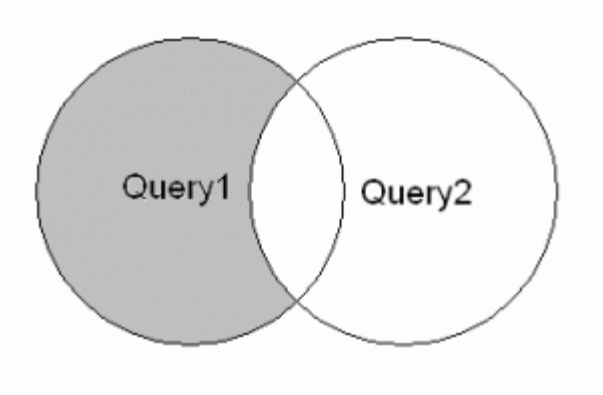


SELECT * FROM table1 UNION SELECT * FROM table2;

When using UNION and UNION ALL columns in SELECT statements need to match. This would return an error:

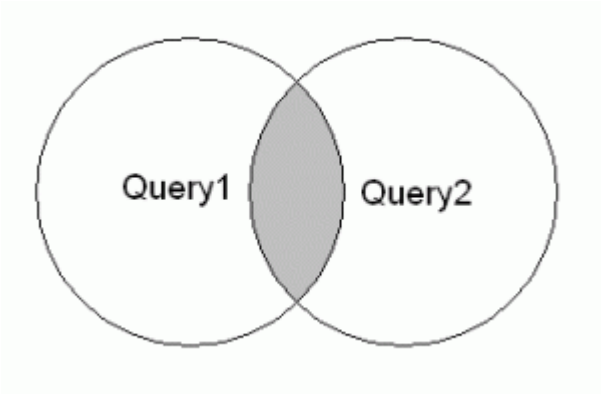
SELECT column1 FROM table1 UNION SELECT * FROM table2;

MINUS: MINUS (also known as EXCEPT) returns the difference between the first and second SELECT statement. It is the one where we need to be careful which statement will be put first, because we will get only those results that are in the first SELECT statement and not in the second.



SELECT * FROM table1 MINUS SELECT * FROM table2;

INTERSECT: INTERSECT is opposite from MINUS as it returns us the results that are both to be found in first and second SELECT statement.



SELECT * FROM table1 INTERSECT SELECT * FROM table2;

EXAMPLE: There is an interesting example that us use SET OPERATORS to compare whether two tables have identical values, testing symmetric difference. If result of this entire query returns no rows, it will mean that they are identical.

(SELECT * FROM table1 MINUS SELECT * FROM table2) UNION (SELECT * FROM table2 MINUS SELECT * FROM table1);
