UNION CLAUSE

The SQL UNION clause/operator is used to combine the results of two or more SELECT statements without returning any duplicate rows.

To use UNION, each SELECT must have the same number of columns selected, the same number of column expressions, the same data type, and have them in the same order but they do not have to be the same length.

```
SELECT column1 [, column2 ]
FROM table1 [, table2 ]
 [WHERE condition]

UNION

SELECT column1 [, column2 ]
FROM table1 [, table2 ]
[WHERE condition]
```

UNION ALL CLAUSE

The SQL UNION ALL clause/operator is used to combine the results of two SELECT statements ***including*** duplicate rows.

The same rules that apply to UNION apply to the UNION ALL operator.

```
SELECT column1 [, column2 ]
FROM table1 [, table2 ]
 [WHERE condition]

UNION ALL

SELECT column1 [, column2 ]
FROM table1 [, table2 ]
[WHERE condition]
```

INTERSECT CLAUSE

The SQL INTERSECT clause/operator is used to combine two SELECT statements, but returns rows only from the first SELECT statement that are identical to a row in the second SELECT statement. This means INTERSECT returns only common rows returned by the two SELECT statements.

The same rules that apply to UNION apply to the INTERSECT operator.

```
SELECT column1 [, column2 ]
FROM table1 [, table2 ]
 [WHERE condition]

INTERSECT

SELECT column1 [, column2 ]
FROM table1 [, table2 ]
[WHERE condition]
```

EXCEPT CLAUSE

The SQL EXCEPT clause/operator is used to combine two SELECT statements and returns rows from the first SELECT statement that are not returned by the second SELECT statement. This means EXCEPT returns only rows which are not available in second SELECT statement.

The same rules that apply to UNION apply to the EXCEPT operator.

```
SELECT column1 [, column2 ]
FROM table1 [, table2 ]
 [WHERE condition]

EXCEPT

SELECT column1 [, column2 ]
FROM table1 [, table2 ]
[WHERE condition]
```

JOin will works with multiple tables
where as
union will work with multiple queries

ASSIGNMENT

ASSIGNMENT – 5

IN THE EMP TABLE DISPLAY :

1) EID NAME CITY DOJ DEPT DESI SALARY OF THE DELHI EMPLOYEES

2) DETAILS OF ALL THE EMPLOYEES WHOSE SALARY DETAILS ARE NOT AVAILABLE.

IN THE INVENTORY STRUCTURE DISPLAY :

1) PID, PDESC, CATEGORY, SNAME, SCITY

2) DISPLAY OID , ODATE , CNAME, CADDRESS, CPHONE, PDESC, PRICE,OQTY, AMT

INDEXES

Indexes

sql has to update lookup table also

Indexes are special lookup tables that the database search engine can use to speed up data retrieval.

An index helps speed up SELECT queries and WHERE clauses, but it slows down data input, with UPDATE and INSERT statements. Indexes can be created or dropped with no effect on the data.

https://www.youtube.com/watch?v=SxHX1T53n_A
https://www.youtube.com/watch?v=O-Mbn6VI1zc
https://www.youtube.com/watch?v=L-THExvsv0s
https://www.youtube.com/watch?v=mbjE4WsWYCA
https://www.youtube.com/watch?v=54gUz7QqE4o&t=405s
https://www.youtube.com/watch?v=SoYYwKXtCC0
https://www.youtube.com/watch?v=Tmbv15xiIPo
https://www.youtube.com/watch?v=6_94Fm_yNAM

The CREATE INDEX Command:

Explicit index

```
CREATE INDEX index_EID ON table_EID (column_EID);
```

Composite Indexes:

```
CREATE INDEX index_EID on table_EID (column1, column2);
```

Implicit Indexes: Implicit indexes are indexes that are automatically created by the database server when an object is created. Indexes are automatically created for primary key constraints and unique constraints.

DROP INDEX Command:

```
DROP INDEX index_EID ON table_EID;
```

select * from emp where city='delhi' and address='dwaraka'
if this query need benefits of indexing i should create an index based on this 2 fields

create index index_city_addr on emp (city,address). >> composite index
but below index wont make benefit for this query

create index index_city on emp (city)
create index index_addr on emp (address)

Non Clustered index: here sql will keep the data itself and create an look up table of it self
clustered index: here sql will reshuffle data(in hard drive) and create lookup table for better retrival.

clustered vs non clustered check more on it

create clustered index indx_name on tbl_name (column)

clustered index can only created with only one column in a table , multiple clustered index can’ t be created.

by default clustered index will be created with primary key automatically by sql.

SQL VIEWS

VIEWS

A view is nothing more than a SQL statement that is stored in the database with an associated ID.

A view can contain all rows of a table or select rows from a table. A view can be created from one or many tables which depends on the written SQL query to create a view.

Views which are kind of virtual tables, allow users to do the following:

- Structure data in a way that users or classes of users find natural or intuitive.
- Restrict access to the data such that a user can see and (sometimes) modify exactly what they need and no more.
- Summarize data from various tables which can be used to generate reports.

VIEWS

every time we run a view its actually rerun the query statement and give data to us.

deleting the data from table will effect the view and if we delete data from view it will also effect the table also.

but if we create a view by combining multiple table or group by statement etc then we cant manipulate view.

```
CREATE VIEW view_EID AS
(SELECT column1, column2.....
FROM table_EID
WHERE [condition]
);
```

The WITH CHECK OPTION:

The WITH CHECK OPTION is a CREATE VIEW statement option. The purpose of the WITH CHECK OPTION is to ensure that all UPDATE and INSERTs satisfy the condition(s) in the view definition.

```
CREATE VIEW view_EID AS
SELECT column1, column2.....
FROM table_EID
WHERE [condition]
WITH CHECK OPTION
;
```

suppose i create a view to fetch emp from delhi.

```
create view delhi_emp_view
as
 select * from emp
        where city ='delhi'
```

here even thought i cant see but i can still insert city detail of 'gurgaon' using this view.coz usually view dont check condition when manupulating a view.
if i use WITH CHECK OPTION the even in the time of manupulation(update and insert) condition in statement of the view has to be meet.