INDEXES

Indexes

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.

The CREATE INDEX Command:

```
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;
```

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

```
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
;
```

Updating a Views

A view can be updated under certain conditions:

- The SELECT clause may not contain the keyword DISTINCT.
- The SELECT clause may not contain summary functions.
- The SELECT clause may not contain set operators.
- The FROM clause may not contain multiple tables.
- The query may not contain GROUP BY or HAVING.
- Calculated columns may not be updated.
- All NOT NULL columns from the base table must be included in the view in order for the INSERT query to function.
- The SELECT clause may not contain an ORDER BY clause.

Dropping Views

DROP VIEW view_EID;

SQL HAVING CLAUSE

SQL HAVING CLAUSE

The HAVING clause enables we to specify conditions that filter which group results appear in the final results

The WHERE clause places conditions on the selected columns, whereas the HAVING clause places conditions on groups created by the GROUP BY clause

The following is the position of the HAVING clause in a query

SELECT

FROM

WHERE

GROUP BY

HAVING

ORDER BY

SQL HAVING CLAUSE

The HAVING clause must follow the GROUP BY clause in a query and must also precede the ORDER BY clause if used.

The following is the syntax of the SELECT statement, including the HAVING clause:

```
SELECT column1, column2
FROM table1, table2
WHERE [ conditions ]
GROUP BY column1, column2
HAVING [ conditions ]
ORDER BY column1, column2
```

e to Think!!

Create a View for the below queries:

From the employee salary table, we need to see the total salary as “TOTAL COST” for each department arranged in the descending order of total salary .

Also just show only those departments where “TOTAL COST” is greater than 50000.

ASSIGNMENT

ASSIGNMENT – 6

1) CREATE A VIEW *EMP_SAL_DETAILS* TO GET EID NAME DOJ DEPT DESI SALARY AS BASIC. ALSO CALCULATE HRA (15% OF BASIC), PF (9% OF BASIC), NET(BASIC+HRA+PF), GROSS(NET-PF).

2) CREATE A VIEW TO DISPLAY EID,NAME, DOJ, DESI, DEPT OF ALL THE MANAGERS JOINED IN 2019.

3) CREATE A VIEW TO HOW MANY TEAM MEMBERS ARE THERE IN EACH DEPARTMENTS IN EACH CITY, ALONG WITH THERE TOTAL & AVERAGE SALARY.

4) IN THE INVENTORY STRUCTURE GENERATE A VIEW *BILL*. IT SHOULD DISPLAY OID,ODATE,CNAME,ADDRESS,PHONE,PDESC, PRICE, OQTY, AMOUNT