MIS 301 Relational Database Management System

DATABASE MANAGEMENT SYSTEM

Structured Query Language(SQL)-4, Database Security & Authorization (concept of GRANT / REVOKE)

Lecture 15 & 16

ORDER BY CLAUSE

- The SQL **ORDER BY** clause is used to sort the data in ascending or descending order, based on one or more columns.
- The syntax of this clause is

```
select attrib1, attrib2, … from <table name>
where <condition>
order by attrib1 (asc/desc), attrib2 (asc/desc), …..;
```

- asc stands for ascending while desc stands for descending. Default is ascending.
- Example :

```
select * from student where city=‘Durgapur’
order by marks desc, roll_no asc;

select roll_no, name from student
order by city;
```

GROUP FUNCTIONS

- ***GROUP FUNCTION*s** generate a single value for a group of records.
- AVG calculates the average of the specified column from a set of rows
- COUNT calculates the number of rows in a set.
- MAX calculates the maximum from a set of values
- MIN calculates the minimum from a set of values
- STDDEV, calculates the standard deviation of a set of values
- SUM calculates the sum of a set of values
- VARIANCE calculates the variance of a set of values

GROUP FUNCTIONS

- ***Example :***

    select sum(marks), max(marks), avg(marks), count(*), count(distinct city) from student;

- In absence of the group by clause, the group functions treat the entire table as a single group
- ***Distinct*** picks up different values ignoring duplication like

    select distinct city from student;

- **DISTINCT** keyword is used in conjunction with the SELECT statement to eliminate all the duplicate records and fetch only unique records.
- Count(*) counts all records in the group
- Count(attribname) counts the number of records with ***not null*** values in the given attribute.

Group By Clause

- **GROUP BY** clause is used along with the SELECT statement to arrange identical data into groups.
- The GROUP BY clause follows the WHERE clause in a SELECT statement and comes before the ORDER BY clause.
- ***Group by*** clause puts all records with the same value for the *group field/attribute* in a single group.
- The syntax is

```
select group_attrib, group_func(attrib2), group_func(attrib3), ….
from <table name>
where condition
group by attrib1, attrib2,…..
order by group_attrib/group_func(attrib),; ← optional
```

Group By Clause

- Example :

```
select city, avg(marks) from student group by city;
```

- Example :

```
select city, count(*) from student
where marks>=60
group by city
order by count(*) desc, city asc;
```

HAVING CLAUSE

- HAVING clause is used in combination with the GROUP BY clause to restrict the groups of returned rows to only those that satisfy the condition following having.
- Just as ***where*** applies conditions on individual records, ***having*** applies conditions on groups of records.
- ***Having*** clause follows the *group by* clause and can not be used in absence of the *group by* clause.
- It precedes the *order by* clause, if *order by* is present.
- Having accepts and rejects groups of records based on a group condition.

Having Clause

- Example :

```
select city, count(*), max(marks)
from student
where stream in (‘marketing’,’finance’,’MIS’,’HR’)
group by city
having avg(marks)>50
order by count(*) desc, city asc;
```

Views in SQL

- A view is a virtual table with no physical existence.
- It draws values from one or more tables based on an SQL defined on it
- It reflects all changes made to the underlying tables.
- It contains rows and columns like a real table.
- Example :

```
create view citywise
as select city, count(*), max(marks)
from student
where stream in (‘marketing’,’finance’,’MIS’,’HR’)
group by city
having avg(marks)>50
order by count(*) desc, city asc;
```

DATA CONTROL LANGUAGE-DCL

- DCL commands are used to control privileges in Database.
- DCL commands can be used for granting permissions for querying and creating/removing/updating data definitions like tables, etc.
- It can also be used for revoking such permissions.
- There are two DCL commands, namely GRANT and REVOKE.
- Grant is used for giving privileges and Revoke for removing the privileges.
- Example: grant create table to mba;

  where mba is a user of the database
- Example: revoke create table from mba;
- ***sysdba*** stands for all permissions
- grant sysdba to mba; →grants all permissions to the user mba

TRANSACTION CONTROL LANGUAGE-TCL

- These commands are used to manage transactions in the database.
- They allow statements to be grouped together into logical transactions which are either executed together or not executed at all.
- The ***commit*** command is used for saving a transaction to the database permanently.
- Syntax: commit;
- The ***rollback*** command is used to undo all DML commands issued since the last commit.
- Syntax: rollback;

Transaction Control Language-TCL

- ***savepoint*** is a TCL command which saves a point in the ongoing DML commands being issued, so that the transaction can be rolled back to that point.
- Example: savepoint upto_this;

  rollback to upto_this;
- In this example rollback undoes all DMLs issued after upto_this.

- TILL WE MEET AGAIN IN THE NEXT CLASS……….