MIS 301 RELATIONAL DATABASE MANAGEMENT SYSTEM

DATABASE MANAGEMENT SYSTEM

Structured Query Language(SQL)-2

LECTURE 10,11,12

TYPES OF SQL COMMANDS

- DDL or Data Definition Language Command
- DML or Data Manipulation Language Command
- DCL or Data Control Language Command
- TCL or Transaction Control Language Command
- DQL or Data Query Language Command

DATA DEFINITION LANGUAGE

- DDL is used for creating table structures, modifying existing structures as well as for removing such structures
- DDL commands are auto-committed i.e. changes made by these commands are permanently saved in the database.
- DDL commands are CREATE, ALTER, DROP, TRUNCATE

DDL COMMAND CREATE

- This command creates a new table in the database.
- Syntax:

```
create table <table name>
( attrib1        datatype(size),
  attrib2        datatype(size),
        :                :
  attribn        datatype(size));
```

- Example:

```
create table student
( roll_no        char(10),
  name           char(20)
  dt_of_birth    date,
  stream         char(10)
  city           char(10),
  marks          number(5,1));
```

DDL COMMAND CREATE

The following constraints are commonly used with the *CREATE TABLE* command

- **NOT NULL** - Ensures that a column cannot have a NULL value
- **UNIQUE** - Ensures that all values in a column are different
- **PRIMARY KEY** - A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
- **FOREIGN KEY** - Uniquely identifies a row/record in another table
- **CHECK** - Ensures that all values in a column satisfies a specific condition
- **DEFAULT** - Sets a default value for a column when no value is specified

Example:

```
create table student
( roll_no          char(10) not null primary key,
  name             char(20) not null,
  dt_of_birth      date,
   mob_no          number(10) unique,
  stream           char(10) check (stream in ('Marketing','MIS','Finance','HR')
  city_code        char(10) foreign key  references city(city_code)  default 'Durgapur',
  marks            number(5,1));
```

DDL COMMAND ALTER

- *Alter* command is used for changing the table structure.
- *Alter* command with the *add* clause adds new attribute definitions to an existing table.
- Syntax with add clause

```
alter table <table name>
add ( attrib1        datatype(size),
                :                          :
        attribn      datatype(size));
```

- Example

```
alter table student
add ( fname          char(20),
        result       bool);
```

DDL COMMAND ALTER

- *Alter* command is used for changing the table structure.
- *Alter* command with the *modify* clause changes the type and size of existing attributes.
- Syntax with modify clause

```
alter table <table name>
modify ( attrib1          datatype(size),
                :                    :
         attribn          datatype(size));
```

- Example

```
alter table student
modify(    roll_no     number(10),
           name        char(30));
```

DDL COMMAND DROP

- *Drop* command is used for removing the table structure along with contents.
- Syntax

  drop table <table name>;

- Example

  drop table student;

DDL COMMAND TRUNCATE

- *Truncate* command is used for removing the table contents and not the table definition.
- Syntax

  truncate table <table name>;

- Example

  truncate table student;

DATA MANIPULATION LANGUAGE

- DML is used for changing contents of the database
- DML commands are not auto-committed i.e. changes made by these commands are not permanently saved in the database and can be rolled back.
- DML commands are INSERT, UPDATE AND DELETE.

DML COMMAND INSERT

- The ***insert*** command is used for adding a row of data or record into a table
- There are more or less three variations of the insert command

  as follows:

1. insert into <tablename>

   values(val1, val2, ……, valn); →here the field values must be provided sequentially. No field value may be omitted though it may be replaced by NULL(representing absence of a value)

Example:

insert into student values(123,'John','25-Mar-97', 'marketing', 'Durgapur', 81.5, 'Robbins', TRUE);

DML COMMAND INSERT

- The ***insert*** command is used for adding a row of data or record into a table
- There are more or less three variations of the insert command as follows:

2. insert into <tablename> (attrib1, attrib2,…, attribn)

values(val1, val2, ……, valn); →here the field values must be provided only for mentioned attributes in the mentioned sequence.
Example:

insert into student(name,marks,result,roll_no) values('Nikhil', 75.0,TRUE,212);

DML COMMAND INSERT

- The ***insert*** command is used for adding a row of data or record into a table
- There are more or less three variations of the insert command as follows:

3. insert into <tablename> (attrib1, attrib2,…, attribn)
   values(&val1, &val2, ……, &valn); →here the field values are input at runtime after prompting and then the accepted entries are inserted as a record.

Example:

```
insert into student(name,marks,result,roll_no) values('&name', &marks, &result, &roll_no);
Enter value for name: Girish
Enter value for marks: 25.5
Enter value for result: FALSE
Enter value for roll_no: 102
```

❖ Every time this command(macro substitution) is repeated, a new record can be input by the user with new values. The first 2 formats of ***insert***, if repeated, insert duplicate records.

DML COMMAND UPDATE

- The ***update*** command is used for modifying/changing the content of a table
- In absence of the **where** clause , update command updates all the records
- The ***set*** clause is used to assign a constant value or an expression(after evaluation) to an attribute in a tuple.
- Syntax: update <table name> set attrib1=value/expr, attrib2=value/expr, ……;
- Example: update student set stream='Finance', marks=marks+10;

Stream for all students are set to Finance irrespective of their previous stream and everyone's marks are increased by 10

- In presence of the where clause, only the records satisfying the where clause are updated.
- **Syntax:** update <table name> set attrib1=value/expr, attrib2=value/expr, … where condition;
- **Example:** update student set stream='Finance', marks=marks+10 where city='Durgapur' and marks<33;

Stream for students residing in Durgapur and having obtained less than 33 marks are set to Finance irrespective of their previous stream and their marks are increased by 10

DML COMMAND DELETE

- The ***delete*** command is used for removing entire records/tuples from a table
- In absence of the **where** clause , delete removes all the records from the table.
- If **where** clause is used then the records satisfying the where clause only, are deleted.
- Syntax: delete from <table name>;
- Example: delete from student;

All student records are deleted leaving the empty table structure/definition.

- In presence of the where clause, only the records satisfying the where clause are updated.
- **Syntax:** delete from <table name> where condition;
- **Example:** delete from student where marks<33;

Records of students securing less than 33 are deleted. Rest of the records remain intact.

- **delete either deletes entire records or not at all . Part of the record i.e. some of the attributes can not be deleted using this command**

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