CST 204 Database Management Systems

B.Tech. CSE

Semester IV

Viswajyothi College of Engineering and Technology

Sindhu Jose, AP, CSE, VJCET

MODULE 2

Relational Model

Syllabus of Module 2

- Structure of Relational Databases - Integrity Constraints, Synthesizing ER diagram to relational schema
- Introduction to Relational Algebra - select, project, cartesian product operations, join - Equi-join, natural join. query examples
- Introduction to Structured Query Language (SQL), Data Definition Language (DDL), Table definitions and operations – CREATE, DROP, ALTER, INSERT, DELETE, UPDATE.

Introduction to SQL

- SQL (Structured Query Language)is a database sub language for querying and modifying relational databases. It was developed by IBM Research in the mid70's and standardized by ANSI in1986.
- In the relational model, data is stored in structures called relations or tables. Each table has one or more attributes or columns that describe the table.

SQL language consists of three categories of statements:

- **Data Definition Language(DDL):** used to create, alter and drop schema objects such as tables and indexes etc.
- **Data Manipulation Language(DML):** used to manipulate the data within those schema objects.
- **Data control language (DCL):** It includes commands such as GRANT and REVOKE which mainly deals with rigths , permissions and other controls of database system.

Data Types in SQL Statements

a) **CHAR(n)** -Character string, fixed length n

b) **VARCHAR(n)** - Variable length character string, maximum length n.

c) **NUMBER(p)** - Integer numbers with precision p.

d) **BOOLEAN** - Boolean variable used to store TRUE, FALSE, NULL

e) **DATE** - Stores year, month, day, hour, minute and second values

f) **BLOB** - Binary Large Object (the data type in Oracle for storing binary files like executables, images etc.)

Data Definition Language (DDL):

- The SQL data-definition language(DDL) allows the specification of information about relations, including:
  - ▶ The schema for each relation.
  - ▶ The domain of values associated with each attribute.
  - ▶ Integrity constraints
- The common DDL commands are

  a. CREATE TABLE:

  b. ALTER TABLE

  c. DROP TABLE

  d. TRUNCATE

  e. RENAME

1) CREATE

- There are two CREATE statements available in SQL:
  - ▶ CREATE DATABASE
  - ▶ CREATE TABLE
- The **CREATE DATABASE** statement is used to create a new database in SQL.
- Syntax:

  CREATE DATABASE database_name;
  - Example

    CREATE DATABASE university;
- The **CREATE TABLE** statement is used to create a new table.

Syntax:

CREATE TABLE <table name> (column1 datatype(size), column2 datatype(size), column3 datatype(size),.....);

Constraints

- Constraint specifications add additional restrictions on the contents of the table.They are automatically enforced by the DBMS.

  a) Primary Key: Specifies the columns that uniquely identify a row in a table.One or more columns could be part of a primary key.

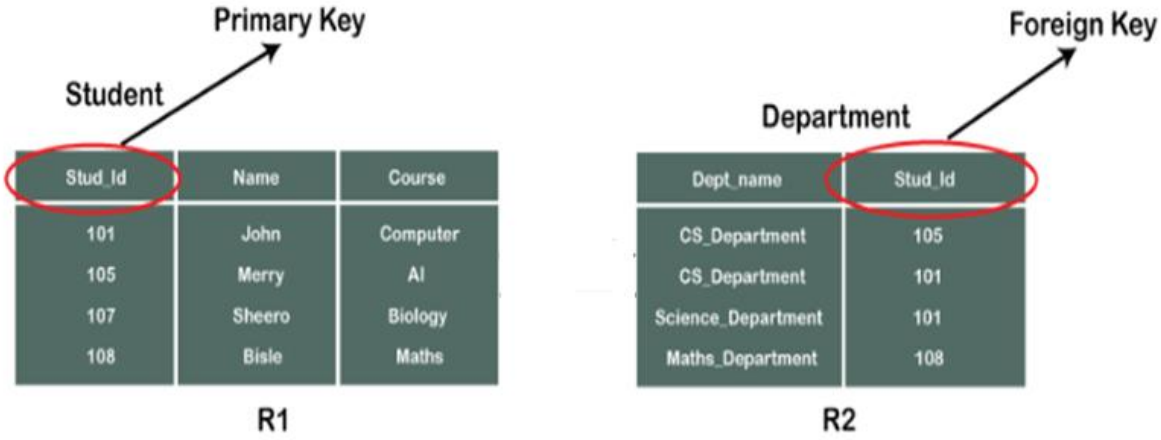
Syntax:

CREATE TABLE <table name> (column1 datatype(size), column2 datatype(size), column3 datatype(size),……, **PRIMARY KEY**(coulmn name));

OR

CREATE TABLE <table name> (column1 datatype(size) **PRIMARY KEY**, column2 datatype(size), column3 datatype(size),……);

Example: create table Student (Stud_Id number(5) **not null**, Name varchar(10),Course varchar(10), **primary key**(Stud_Id));



b) NOT NULL: Specifies that the column that can't be set to null. Normally, primary key columns are declared as NOT NULL(as shown in above Example).

Syntax:

CREATE TABLE <table name> (column1 datatype(size) **NOT NULL**, column2 datatype(size), column3 datatype(size) ,.....);

c) UNIQUE: Specifies that this column has a unique value or null for all rows of the table.

Syntax:

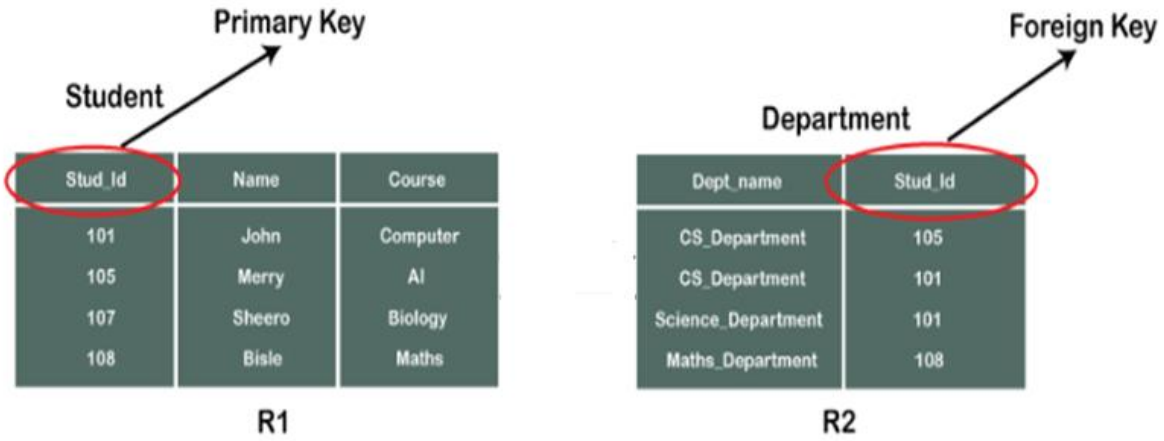
CREATE TABLE <table name> (column1 datatype(size) **UNIQUE**, column2 datatype(size), column3 datatype (size),.....);

d)Foreign Key: Specifies the set of columns in a table that is used to establish the relationship with another table in the database.

Syntax:

CREATE TABLE <table name> (column1 datatype(size) **PRIMARY KEY**, column2 datatype(size), column3 datatype(size),.....**FOREIGN KEY**(column-name) **REFERENCES** < referenced-table>(column name);

Example: create table Department (Dept_name varchar(15) **not null**, Stud_Id number(10), **foreign key** (Stud_Id) **references** STUDENT(Stud_Id));



e)CHECK: Specifies a user defined constraint, known as a check condition. The CHECK specifier is followed by a condition enclosed in parentheses.

Syntax:

CREATE TABLE <table name> (column1 datatype(size) **CHECK**(Condition), column2 datatype(size), column3 datatype(size),……);

2) ALTER TABLE (Use ADD, DROP, MODIFY, RENAME with ALTER as shown below)

- The ALTER TABLE statement is used to change the table definition.

To add new column

ALTER TABLE<table name> **ADD** column name datatype(size);

To add Primary key

ALTER TABLE <table name> **ADD PRIMARY KEY**(column name);

To add Foreign key

ALTER TABLE <table name> **ADD FOREIGN KEY** (column name) **REFERENCES** <table name> (column name);

To remove a Foreign key

ALTER TABLE <table name> **DROP** <foreign key name>;

To modify the existing columns in a table

ALTER TABLE <table name> **MODIFY** column_name column_type;

To rename a table

ALTER TABLE <table name> **RENAME TO** <new table name>;

Example 1: Alter(To add new column)

ROLL_NO	NAME
1	Ram
2	Abhi
3	Rahul
4	Tanu

To ADD 2 columns AGE and COURSE to table Student.

ALTER TABLE Student ADD (AGE number(3),COURSE varchar(40));

ROLL_NO	NAME	AGE	COURSE
1	Ram		
2	Abhi		
3	Rahul		
4	Tanu		

Example 2 :Alter (To drop a column)

ROLL_NO	NAME	AGE	COURSE
1	Ram		
2	Abhi		
3	Rahul		
4	Tanu		

- ALTER TABLE Student DROP COLUMN COURSE;

ROLL_NO	NAME	AGE
1	Ram	
2	Abhi	
3	Rahul	
4	Tanu	

Example 3: Alter

To add primary key to a table

ALTER TABLE Student **ADD PRIMARY KEY** (Stud_Id);

To add Foreign key to a table

ALTER TABLE Department **ADD FOREIGN KEY** (Stud_Id)

REFERENCES Student(Stud_Id);

To remove a Foreign key

ALTER TABLE Department **DROP FOREIGN KEY** Stud_Id;

To modify the existing columns in a table

ALTER TABLE Student **MODIFY** COURSE varchar(20);

To rename a table

ALTER TABLE Department **RENAME TO** Dept;

3) DROP TABLE

- DROP is used to delete a whole database or just a table.
- The DROP statement destroys the objects like an existing database, table, index, or view.

Syntax:

To drop a table

DROP TABLE <table name>;

To drop a database

DROP DATABASE <database_name>;

4)TRUNCATE TABLE COMMAND

- The SQL TRUNCATE TABLE command is used to delete complete data from an existing table.(empty for reuse).

Syntax:

TRUNCATE TABLE <table name>;

DROP Vs TRUNCATE

- You can also use DROP TABLE command to delete complete table **but** it would **remove complete table structure** form the database and you would need to re-create this table once again if you wish you store some data.
- Truncate preserves the structure of the table for future use, unlike drop table where the table is deleted with its full structure.
- Truncate is normally ultra-fast than DROP and its ideal for deleting data from a temporary table.
- Table or Database deletion using DROP or TRUNCATE statement cannot be rolled back, so it must be used wisely.

DML Statements

- The DML statements are used to manipulate (store and maintain) the data in database tables. Common DML statements are:

  a. INSERT

  b. UPDATE

  c. DELETE

  d. SELECT

a. INSERT

- The INSERT INTO statement is used to insert new records in a table.
- It is possible to write the INSERT INTO statement in two ways:

1. Specify both the column names and the values to be inserted:

Syntax :

INSERT INTO table_name (column1, column2, column3, ...) **VALUES**(value1, value2, value3, ...);

2. If you are adding values for all the columns of the table, you do not need to specify the column names in the SQL query. Make sure that the order of the values is in the same order as the columns in the table.

Syntax:

INSERT INTO< table_name> **VALUES** (value1, value2, value3, ...value n);

Example 1: Insert

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
89	White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
91	Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland

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

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
89	White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
91	Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland
92	Cardinal	Tom B. Erichsen	Skagen 21	Stavanger	4006	Norway

Example 2:

- Insert Data Only in Specified Columns
  - It is also possible to only insert data in specific columns.
  - Syntax
    - INSERT INTO Customers (CustomerName, City, Country) VALUES ('Cardinal', 'Stavanger', 'Norway');
    - The selection from the "Customers" table will now look like this:

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
89	White Clover Markets	Karl Jablonski	305 - 14th Ave. S. Suite 3B	Seattle	98128	USA
90	Wilman Kala	Matti Karttunen	Keskuskatu 45	Helsinki	21240	Finland
91	Wolski	Zbyszek	ul. Filtrowa 68	Walla	01-012	Poland
92	Cardinal	null	null	Stavanger	null	Norway

Example 3: Insert

- To insert multiple rows in a table using Single SQL Statement

```
INSERT INTO table_name(Column1,Column2,Column3,.....)
VALUES (Value1, Value2,Value3,.....),
       (Value1, Value2,Value3,.....),
       (Value1, Value2,Value3,.....),
       ..... ;
```

- Example

```
INSERT INTO STUDENT(ID, NAME,AGE,GRADE,CITY)
VALUES(1,"AMIT KUMAR",15,10,"DELHI"),
      (2,"GAURI RAO",18,12,"BANGALORE"),
      (3,"MANAV BHATT",17,11,"NEW DELHI"),
      (4,"RIYA KAPOOR",10,5,"UDAIPUR");
```

b. UPDATE

- Updating allows us to change some values in a tuple without necessarily changing all.
- We can update single columns as well as multiple columns using UPDATE statement as per our requirement.

Syntax

UPDATE <table_name> **SET** column1 = value1, column2 = value2, …**WHERE** <condition>;

Note: The WHERE clause specifies which record(s) that should be updated. If you omit the WHERE clause, all records in the table will be updated!

Example: Update the date of birth of employee 1090 to15-Oct-1979.

UPDATE EMPLOYEE **SET** DATE_OF_BIRTH='15-OCT-1979' **WHERE** EMPLOYEE_ID=1090;

Example 1 : Update

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

```
UPDATE Customers
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'
WHERE CustomerID = 1;
```

Result

```
UPDATE Customers
SET ContactName = 'Alfred Schmidt', City= 'Frankfurt'
WHERE CustomerID = 1;
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Alfred Schmidt	Obere Str. 57	Frankfurt	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

Example 2: Update

- UPDATE Multiple Records
  - It is the WHERE clause that determines how many records will be updated.

```
UPDATE Customers
SET ContactName='Juan'
WHERE Country='Mexico';
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

Result

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Alfred Schmidt	Obere Str. 57	Frankfurt	12209	Germany
2	Ana Trujillo Emparedados y helados	Juan	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Juan	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Thomas Hardy	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Christina Berglund	Berguvsvägen 8	Luleå	S-958 22	Sweden

- If you omit the WHERE clause, ALL records will be updated!

```
UPDATE Customers
SET ContactName='Juan';
```

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Juan	Obere Str. 57	Frankfurt	12209	Germany
2	Ana Trujillo Emparedados y helados	Juan	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Juan	Mataderos 2312	México D.F.	05023	Mexico
4	Around the Horn	Juan	120 Hanover Sq.	London	WA1 1DP	UK
5	Berglunds snabbköp	Juan	Berguvsvägen 8	Luleå	S-958 22	Sweden

c. DELETE

- The DELETE Statement in SQL is used to delete existing records from a table.
- We can delete a single record or multiple records depending on the condition we specify in the WHERE clause.

Syntax

DELETE FROM <table_name> **WHERE** <condition>;

Note: The WHERE clause specifies which record(s) should be deleted. If you omit the WHERE clause, all records in the table will be deleted!

Delete Specific rows

DELETE FROM <table name> **WHERE** <condition>;

To Delete all rows

DELETE * FROM <tablename>;

OR

DELETE FROM <tablename>;

Example 1: Delete

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

- DELETE FROM Customers WHERE CustomerName='Alfreds Futterkiste';

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

d. SELECT

- The SQL SELECT statement queries data from tables in the database. The statement begins with the SELECT keyword. The basic SELECT statement has 3 clauses:

  1. SELECT – specifies the table columns retrieved
  2. FROM - specifies the tables to be accessed
  3. WHERE – specifies which rows in the FROM tables to use

Syntax

SELECT <attribute list> **FROM** <table name> **WHERE** <condition>;

To select the whole table

SELECT * FROM <table name>;

Example 1: Select

Student

ID	First_name	Last_name	Age	Subject	Hobby
1	Amar	Sharma	20	Maths	Cricket
2	Akbar	Khan	22	Biology	Football
3	Anthony	Milton	25	Commerce	Gambling

- **SELECT** first_name, last_name **FROM** Student;

Amar	Sharma
Akbar	Khan
Anthony	Milton

Example 2: Select

- **SELECT * FROM** Customers **WHERE** City='Berlin';

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

- Output

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany

Example 3: Select

- **SELECT * FROM** Customers;
- Customers

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico

- Output

CustomerID	CustomerName	ContactName	Address	City	PostalCode	Country
1	Alfreds Futterkiste	Maria Anders	Obere Str. 57	Berlin	12209	Germany
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Avda. de la Constitución 2222	México D.F.	05021	Mexico
3	Antonio Moreno Taquería	Antonio Moreno	Mataderos 2312	México D.F.	05023	Mexico