CF4CRT10 DATABASE MANAGEMENT SYSTEMS AND SECURITY

Module: - 2

Structured Query Language (SQL)

SQL is a standard query language used to access and manipulate data in databases. SQL stands for Structured Query Language. We can create, update, delete, and retrieve data in databases like MySQL, Oracle, PostgreSQL, etc. Overall, SQL is a query language that communicates with databases.

1. DDL Commands (Data Definition Language Commands)

1. Create table command

This command is used to create a new table in SQL. Syntax:

CREATE TABLE table_name (Attribute1 datatype, Attribute2 datatype, .. Attribute n datatype, primary key(key_name), CONSTRAINT const_name FOREIGN KEY(key_name) REFERENCES table_name(key_name));

Example:1

Create a table employee with **attributes eid,ename, age, salary and dept id(foreign key)**

Also

Create another table department with **attribute deptid,dname and location**

CREATE TABLE department (deptid int, dname varchar(20),location varchar(30),primary key(deptid));

CREATE TABLE employee (eid int primary key,ename varchar(10),age int, salary int, deptid int, CONSTRAINT pk_id FOREIGN KEY(deptid) REFERENCES department(deptid));

Example:2

student

Reg_no	Sname	Address	Age	mark

CREATE TABLE student (reg_no int primary key, sname varchar(10), address varchar(30),age int, mark int);

<u>Specifying constraints in SQL</u>

CREATE TABLE employee (eno int primary key NOT NULL,ename varchar(20), age int DEFAULT ‘19’, dno int,phone int UNIQUE, CONSTRAINT pk_id FOREIGN KEY (DNO) REFERENCES department (dno));

From the above table definition constraints are

1. **Primary key** – it is used to identify each record of a table uniquely. It is called key constraint
2. **NOT NULL** – this constraint is used to specify NULL values is not permitted for a particular attribute
3. **DEFAULT -** by using this constraint we can specify a default value for an attribute. The default value is included within single quotes.
4. **FOREIGN KEY -** it is called referential integrity constraint. It is specified when the given table has a foreign key
5. **UNIQUE -** This constraint is used to specify alternate key or secondary key for a table

2. DROP COMMAND

It is used to destroy the created table. After the execution of this command the table along all of its data will be destroyed

Syntax:

DROP TABLE table_name;

Example : drop table student;

3. RENAME COMMAND

It is used to rename the existing table

Syntax:

RENAME oldtable_name TO newtable_name;

Example

If we want to change the table name employee to customer, use the following code

RENAME employee TO customer

4. ALTER COMMAND

There are 3 uses for the alter command

1. To add new columns
2. To remove any column from the table
3. To modify the data type of existing table

1. Add new columns Syntax:

ALTER TABLE table_name ADD (attribute1 datatype, Attribute 2 datatype……);

Example:

From the above table employee add new attribute or column named designation

ALTER TABLE employee ADD(designation varchar(20));

2. Remove any column

Syntax:

ALTER TABLE table_name DROP COLUMN column_name;

Example:

From the table employee remove the column age ALTER TABLE employee DROP COLUMN (age);

3. Modify the data type existing table

This command is used to modify the data type of the existing column of a table

Syntax:

ALTER TABLE table_name MODIFY (column_name new_datatype);

Example:

Change the datatype of age to number in the table employee ALTER TABLE employee MODIFY(age number(3));

<u>Restrictions of ALTER command</u>

By using the ALTER command we cannot perform the following tasks

- Change the name of the table
- Change the name of the column
- Decreases the size of the column if table data exists

2. DML Commnads (Data Manipulation Commands)

1. Insert Command

This command is used to insert values to the table. The values must be inserted in the order in which the columns are created in the table

Syntax:

INSERT INTO table_name VALUES(value1,value2,…….,value n); OR

INSERT INTO table_name (column_name1,column_name2…) VALUES (value1,value2….)

Example:

Create the following table and insert the values STUDENT

SID	SNAME	AGE	MARK
1	Anu	18	200
2	Bindhu	17	289
3	Veena	18	250

CREATE TABLE student(sid int primary key, sname varchar(20), age int, mark int);

INSERT INTO student VALUES (1,’Anu’,18,200);

INSERT INTO student VALUES (2,’Bindhu’,17,289);

INSERT INTO student VALUES (3,’Veena’,18,250);

2. Select Command

It is used for viewing or retrieving data from the table. We can use logic operators as well as relational operators with in the select command.

Syntax

SELECT <attribute list>
FROM <table list> WHERE
<condition>

<attribute list > is the list of attribute names to be retrieved from the query.

<table list> is the list table names in which these attributes belongs.

<condition> It is a condition that identifies the tuples to be retrieved by the query

Example:

Create the following table , insert the values and do the quires Employee

ENO	ENAME	GENDER	PLACE	PHONE	AGE	SALARY
1	Anu	female	Pala	22435346	30	30000
2	Meenu	female	kottayam	46474585	40	20000
3	Aravind	male	kottayam	35665768	38	30000
4	Veena	female	ponkunnam	12567869	35	25000

5	Alen	male	Pala	12345679	48	45000

CREATE TABLE employee (eno int primary key, ename varchar(20),gender varchar(20),place varchar(20),phone int,age int,salary int)

INSERT INTO employee VALUES(1,’anu’,’female’,’pala’,
22435346,30,30000);

INSERT INTO employee VALUES(2,’meenu’,’female’,’kottayam’, 46474585,40,20000);

INSERT INTO employee VALUES(3,’aravind’,’male’,’kottayam’, 35665768,38,30000);

INSERT INTO employee VALUES(4,’veena’,’female’,’ponkunnam’, 12567869,35,25000);

INSERT INTO employee VALUES(5,’alan’,’male’,’pala’,
12345679,48,45000);

Queries

1. Retrieve all employees who have the age below 40 SELECT
   ename
   From employee
   WHRE age<40
2. Retrieve all employees
   SELECT *
   From employee
3. Retrieve name and age of all male employees SELECT
   ename,age

FROM employee WHERE
gender=’male’

ALL and DISTINCT

4. Select all employee salary from the employee table SELECT
ALL salary From employee

Output
30000
20000
30000
25000
45000

5. Select distinct employee salary from the employee table SELECT
DISTINCT salary
From employee Output
30000
20000
25000
45000

6. Retrieve all employee details whose age less than 40and are coming from ‘pala’
SELECT * FROM employee WHERE
age<40 and place=’pala’

7. Retrieve eno and name of all employees who are either coming from ‘pala’ or salary greater than 25000
SELECT eno,ename
FROM employee
WHERE place=’pala’ OR salary>25000

8. Retrieve eno and name of all female employees who are either coming from ‘kottayam’

and age greater than 20

```
SELECT eno,ename
FROM employee
WHERE place=’kottayam’ and gender=’female’ and age>20
```

Renaming of attributes in SQL

It is possible to rename any attribute that appears in the result of a query. To do that use the qualifier ‘as’ followed by new attribute name

Example:

Display all employee names working in sales department and display the result with the name ‘sales employees’

```
SELECT e.ename AS sales employees FROM
employee e, department d
WHERE e.Dno = d.Dno and d.dname=’sales’
```

Use of Arithmetic operators in query

We can use basic arithmetic operators (+,-,*, /) on attributes to perform any of the arithmetic operations on attribute values

Example: write a query to increase the salary of all employees working in sales department to 10% and show the results as ‘new salary’

```
SELECT e.ename , e.esal, e.esal+(e.esal*.1) AS new salary FROM
employee e, department d
WHERE e.Dno = d.Dno and d.dname=’sales’;
```

Delete Command In SQL

The delete command is used to remove tuples from the tables based on a condition. We can delete several number of tuples based a condition by using a single delete command.

Syntax:

DELETE FROM table_name WHERE <condition>

Q . delete all employees whose name starts with v.

DELETE FROM employee WHERE ename LIKE ‘v%’

Q. delete all employees in sales department **DELETE FROM employee e, department d WHERE e.dno=d.dno and d.dname=’sales’**

Update Command in SQL

The update command is used to modify the values of one or more selected tuples depending on a condition. We cannot modify the primary key values

Syntax:

UPADTE tablename SET attribute_name=new value WHERE <condition>

Q. modify the name of the employee as ‘bijil’ with the employee no 3

UPDATE employee SET ename=’bijil’ WHERE eno=3

Q. modify the salary of the employee ‘veena’ to 30000

UPDATE employee SET esal=30000 WHERE ename=’veena’;

Aggregate functions in SQL

These functions are used to perform mathematical operations on tuples. Aggregate functions are used to summarize information from multiple tuples. There are 5 type of aggregate functions

- Count
- Sum

- Max
- Min
- Avg

COUNT function returns the no: of tuples specified in the query Example: count total number of employees working in company Ans: **SELECT COUNT(*) FROM employee;**

Q:select number of employees whose salary between 20000 and 40000

Ans: **SELECT COUNT(ename) FROM employee WHERE esal BETWEEN 20000 and 40000;**

Q:select number employees working in sales department

Ans: **SELECT COUNT(e.ename) FROM employee e, department d WHERE e.dno=d.dno and d.dname=’sales’;**

SUM,MAX,MIN,AVG can be applied to set of values and returns respectively Sum, maximum, minimum and average of those values

Q: display maximum and minimum salary of employees

Ans: **SELECT MAX(esal),MIN(esal) FROM employee;**

Q: display average salary of employees in sales department

Ans: **SELECT AVG(e.esal) FROM employee e, department d WHERE e.dno=d.dno and d.dname=’sales’;**