22AIE303 DBMS

LABSHEET 2

Date:23/9/2024

Aniketh Vijesh

Question 1

department

Column name	Datatype	Size	Constraint
deptno	Integer		PK
dname	Varchar	14	Not null
loc	Varchar	20	

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

1. Create the tables with suitable constraints.

```
CREATE TABLE department(
        deptno INT PRIMARY KEY,
        dname VARCHAR(14) NOT NULL,
        loc VARCHAR (20)
);
```

2. Insert data in the two tables.

```
INSERT INTO department VALUES
(10 , 'ACCOUNTING' , 'NEW YORK'),
(20 , 'RESEARCH' , 'DALLAS'),
(30 , 'SALES' , 'CHICAGO'),
(40 , 'OPERATIONS' , 'BOSTON');
```

employee

Column name	Datatype	Size	Constraint
empno	Integer		PK
ename	Varchar	20	Not null
job	Varchar	10	
mgr_id	Integer		
hired_date	date		
basic_sal	Numeric	(6,2)	Default value 1000
incentive	Numeric	(6,2)	Should not be greater than basic_sal
deptno	Integer		Refers to deptno of dept table

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	INC	DEPTNO
7369	SMITH	CLERK	7902	17/12/1980	6800		20
7499	ALLEN	SALESMAN	7698	20/02/1981	11600	300	30
7521	WARD	SALESMAN	7698	22/02/1981	11250	500	30
7566	JONES	MANAGER	7839	02/04/1981	22975		20
7654	MARTIN	SALESMAN	7698	28/09/1981	11250	1400	30
7698	BLAKE	MANAGER	7839	01/05/1981	22850		30
7782	CLARK	MANAGER	7839	09/06/1981	22450		10
7788	SCOTT	ANALYST	7566	09/12/1982	13000		20

1. Create the tables with suitable constraints.

```
CREATE TABLE employee(
        empno INT PRIMARY KEY,
        ename VARCHAR(20) NOT NULL,
        job VARCHAR(10),
        mgr_id INT,
        hired_date DATE,
        basic_sal NUMERIC(6,2) DEFAULT 1000,
        incentive NUMERIC(6,2),
        deptno INT,
        CONSTRAINT incentive_check CHECK(incentive <= basic_sal),
        FOREIGN KEY (deptno) REFERENCES department(deptno)
);
```

2. Insert data in the two tables.

```
INSERT INTO employee VALUES
(7499,'ALLEN','SALESMAN',7698,'20/02/1981',11600,300,30),
(7521,'WARD','SALESMAN',7698,'22/02/1981',11250,500,30),
(7654,'MARTIN','SALESMAN',7698,'28/09/1981',11250,1400,30)
;

INSERT INTO employee
(empno,ename,job,mgr_id,hired_date,basic_sal,deptno)
VALUES
(7369,'SMTIH','CLERK',7902,'17/12/1980',6800,20),
(7566,'JONES','MANAGER',7839,'02/04/1981',22975,20),
(7698,'BLAKE','MANAGER',7839,'01/05/1981',22850,30),
(7782,'CLARK','MANAGER',7839,'09/06/1981',22450,10),
(7788,'SCOTT','ANALYST',7566,'09/12/1982',13000,20)
;
```

3. Select all data from the DEPARTMENT table.

```
SELECT * FROM department;
```

4. Get the details of all the employees.

```
SELECT * FROM employee;
```

5. Show the details of employee 'BLAKE'.

```
SELECT * FROM employee WHERE ename = 'BLAKE';
```

6. Get employee number, employee name of employees who are managers.

```
SELECT empno,ename FROM employee WHERE job = 'MANAGER';
```

7. Display unique jobs with second letter as 'a' from the EMPLOYEE table.

```
SELECT DISTINCT(job) FROM employee WHERE job LIKE '_A%';
```

8. Display the names of employees concatenated with their jobs.

```
SELECT ename||job FROM employee;
```

9. Display all the names, department numbers and hired dates from the EMPLOYEE table.

```
SELECT ename,deptno,hired_date from employee;
```

10. Display employees in the ascending order of their names.

```
SELECT * FROM employee ORDER BY ename;
```

11. Find the names of all employees that begin with 'S' or 'J'

```
SELECT ename FROM employee WHERE ename LIKE 'S%' OR ename LIKE 'J
%';
```

12. Get the highest salary from the EMPLOYEE table.

```
SELECT max(basic_sal) from employee;
```

13. Display the names, deptno of all employees who receive salary between 10000 and 25000.

```
SELECT ename,deptno FROM employee WHERE basic_sal BETWEEN 10000
and 25000;
```

14. List department number and count of employees in each department ordered by department number.

```
SELECT deptno,COUNT(empno) FROM employee GROUP BY deptno;
```

15. List the names and hired date of managers and clerks without incentives.

```
SELECT ename,hired_date FROM employee
WHERE (job IN ('MANAGER','CLERK')) AND incentive IS NULL;
```

16. Delete the records with deptno '10' from the EMPLOYEE table.

```
DELETE FROM employee WHERE deptno = 10
```

17. Print the names and jobs of all employees except 'analyst'.

```
SELECT ename,job FROM employee WHERE job <> 'ANALYST';
```

18. Print the name of employees whose salaries are greater than the value 21000.

SELECT ename FROM employee WHERE basic_sal > 21000;

19. Find the names of employees who have a salary equal to Rs 13000.

SELECT ename from employee WHERE basic_sal = 13000;

20. Display the empname, deptno, hired date information in the dept '20' and '30'.

SELECT ename, deptno, hired_date from employee WHERE deptno IN (20,30);

Question 2

Create a table with the following columns :

Column name	Data type
Empno	vachar
Deptno	varchar
Name	varchar
Desig	varchar
Basic	numeric
Join_date	date
gender	character

1. Set the composite key as empno and deptno.

```
ALTER TABLE employees
ADD CONSTRAINT prime PRIMARY KEY(empno,deptno);
```

2. Add 3 rows into the table.

```
INSERT INTO employees VALUES
(22001,'D1','THARUN','MANAGER',80000,'12-07-2023','M'),
(22002,'D2','ABHIRAM','ANALYST',75000,'27-04-2024','M'),
(22003,'D3','ADITHYA','CONSULTANT',78000,'16-10-2022','M');
```

3. Display all the records from the above table.

```
SELECT * FROM employees;
```

4. Display the empno, name, designation and basic salary of all the employees.

```
SELECT empno,emp_name,desig,basic_sal FROM employees;
```

5. Display empno and name of all the employees from department no. 2

```
SELECT empno,emp_name FROM employees WHERE deptno = 'D2';
```

6. Display empno, name, desig, department no., and basic salary in the descending order of basic pay.

```
SELECT empno,emp_name,desig,deptno,basic_sal FROM employee
ORDER BY basic_sal;
```

7. Display all designations without duplicate values.

```
SELECT DISTINCT(desig) from employees;
```

8. Display empno,name,desig, and basic salary in the descending order of basic pay and in the ascending order of names.

```
SELECT empno,emp_name,desig,basic_sal from employees
ORDER BY basic_sal DESC, emp_name;
```

9. Sort the table in the order of basic salary.

SELECT * FROM employees ORDER BY basic_sal;

10.Delete the records of employees whose basic is less than 5000.

DELETE FROM employees WHERE basic_sal < 5000;

Question 3

Create the following tables
Category_ details (category_id integer (2), category_name varchar (10))
Sub_category_details (sub_category_id integer(2), category_id integer(2),sub_category_name varchar(10))
Product_details (Product_id integer (6), category_id integer(2),sub_category_id integer(2), product_name varchar(10))

CREATE TABLE category_details(

category_id INT(2),

category_name VARCHAR(10),

```
);

CREATE TABLE category_details(
        category_id INT,
        category_name VARCHAR(10),
);

CREATE TABLE sub_category_details(
        sub_category_id INT,
        category_id INT,
        sub_category_name VARCHAR(10)
);
```

1) Add a primary key constraint (without any constraint name) on column category_id of category_details table.

```
ALTER TABLE category_details
ADD PRIMARY KEY(category_id);
```

2) Add a primary key constraint with a constraint name on column sub_category_id of sub_category_details table.

```
ALTER TABLE sub_category_details
ADD CONSTRAINT category_prime PRIMARY KEY(sub_category_id);
```

3) Add a foreign key constraint with constraint name on column category_id of sub_category_details table referencing category_id of category_details table.

```
ALTER TABLE sub_category_details
ADD CONSTRAINT fk_category FOREIGN KEY (category_id) REFERENCES
category_details (category_id);
```

4) For product_details table add primary key constraint on product_id. Also add foreign key constraint on category_id and sub_category_id columns referencing category_details(category_id) and sub_category_details (sub_category_id). Give appropriate names for all constraints.

```
ALTER TABLE product_details
```

ADD PRIMARY KEY (product_id),

ADD CONSTRAINT fk_category_id FOREIGN KEY (category_id) REFERENCES category_details (category_id),

ADD CONSTRAINT fk_sub_category_id FOREIGN KEY (sub_category_id) REFERENCES sub_category_details (sub_category_id);

5) Add a new column (price numeric(6,2)) to product_details table

ALTER TABLE product_details

ADD COLUMN price NUMERIC(6,2);

6) Insert four tuples in the table. (With valid data)

INSERT INTO category_details (category_id, category_name) VALUES (1, 'Electronics'), (2, 'Clothing');

INSERT INTO sub_category_details (sub_category_id, category_id, sub_category_name) VALUES (1, 1, 'Mobiles'), (2, 2, 'Dresses');

INSERT INTO product_details (product_id, category_id, sub_category_id, product_name, price) VALUES

(101, 1, 1, 'iPhone', 999.99),

(102, 1, 1, 'Android', 599.99),

(201, 2, 2, 'Skirt', 49.99),

(202, 2, 2, 'Jeans', 89.99);

7) Add a new column BRANDNAME varchar(20) NOT NULL

ALTER TABLE product_details

ADD COLUMN BRANDNAME VARCHAR(20) NOT NULL;

8) Rename Category_details table to Cat_dt .

ALTER TABLE category_details

RENAME TO Cat_dt;