Database Lab-2

Definition of the database of a company is specified below.

Required **DDL** (table creating and structural operations) commands for this system are included in: "**company-schema**" file; **DML** (data insertion operations) commands are included in "**company-data**" file.

Definition of each table in the system:

1. DEPARTMENT table is the one keeps information of the departments in the company. This table contains department number, name, social security number of the manager and starting date of the manager's management. Column names and types are illustrated below.

DEPARTMENT

DNAME	NOT NULL	VARCHAR(25)
DNUMBER	NOT NULL	NUMERIC
MGRSSN	NOT NULL	CHAR(9)
MGRSTARTDATE		DATE

2. PROJECT table contains projects' informations developed in the company. This table might contain project number, name, where it's carried out, which department in charge of the project. Column names and types are illustrated below.

PROJECT

PNAME	NOT NULL	VARCHAR(25)
PNUMBER	NOT NULL	NUMERIC
PLOCATION		VARCHAR(15)
DNUM	NOT NULL	NUMERIC

3. DEPT_LOCATIONS table is the table which contains information about locations of company departments. Department number and location information are stored in this table. Column names and types are illustrated below.

DEPT_LOCATIONS

DNUMBER	NOT NULL	NUMERIC
DLOCATION	NOT NULL	VARCHAR(15)

4. EMPLOYEE table are constructed for employees working under company. This table might contain each employee's social security number, first name, last name, birth date, address, sex, salary, supervisor's social security number. Column names and types are illustrated below.

EMPLOYEE

FNAME	NOT NULL	VARCHAR(15)
MINIT		VARCHAR(1)
LNAME	NOT NULL	VARCHAR(15)
SSN	NOT NULL	CHAR(9)
BDATE		DATE

ADDRESS		VARCHAR(50)
SEX		CHAR(1)
SALARY		NUMERIC
SUPERSSN		CHAR(9)
DNO		NUMERIC

5. WORKS_ON table describes which project and how many hours each employee works on. This table might contain each employee's social security number, worked on project's number and how many hours the employee works. Column names and types are illustrated below.

WORKS_ON

ESSN	NOT NULL	CHAR(9)
PNO	NOT NULL	NUMERIC
HOURS		NUMERIC

6. DEPENDENT table stores information of each employee's family members. This table might contain each employee's social security number, each dependent's name, sex, birth date and relationship type (e.g. daughter, mother etc.). Column names and types are illustrated below.

DEPENDENT

ESSN	NOT NULL	CHAR(9)
DEPENDENT_NAME	NOT NULL	VARCHAR(15)
SEX		CHAR(1)
BDATE		DATE
RELATIONSHIP		VARCHAR(8)

Query examples:
<u>Question 1:</u> Write down the sql query that lists first and last name of the employees working under department no 5.

Query:
select fname, lname from employee where dno=5;

Result is stated as this:

FNAME	LNAME
Franklin	Wong
John	Smith
Ramesh	Narayan
Joyce	English

<u>Question 2:</u>Write down the query that finds **department name** of the employee named "Jared James" working for.

Table employee includes employees' names and **department number** they work for, but **department names** does not exist in the table employee. Therefore department name data should be gathered from "**department"** table; while whether employee name is "Jared James" or not is checked from "**employee**" table. Accordingly there will be two tables in our query:

Query:
select dname from employee e, department d where fname = 'Jared' and lname='James' and e.dno = d.dnumber;

Because only the department name data is asked in query, after select, ***dname*** is selected; as condition it is stated that desired row is the one includes "***Jared James***" named entry; and two tables are ***join***ed. **dno** in employee table refers to **dnumber** in department table (In other words dno data in employee table are related to dnumber data in department table). Hence matching is done on these two columns.

NOTE: The reason for adding "e" after employee table and "d" after department table is the will to state used tables in short form. "d" and "e" are the variable names named by user. In this way desired shortening can be given for the table names:
It can be used in the form of ***... from table name shortening...***

Result is stated as this:

DNAME
Software

Question 3: Write down the query locates in which city(ies) there are offices of company's "Sales" department.

In **dept_locations** tables, **location** data of the company departments are stated, however **department names** are not. Location (city) data should be attained from "dept_locations" table; and whether department name is "Sales" or not should be checked from "**department**" table. In this case again there will two tables in the query:

Query:
select dlocation from department d, dept_locations d_l where d.dname = 'Sales' and d.dnumber=d_l.dnumber;

In the query, since location data is just asked, after keyword select, ***dlocation*** is selected; as condition it is stated that department named "***Sales***" is concerned; two tables are ***join***ed. **dnumber** in Dlocation table refers to **dnumber** in department table. Hence matching is done on these two columns.

Result:

DLOCATION
Chicago
Dallas
Miami
Philadephia
Seattle

Query 4: Write down the query finds employee(s)' first/last name data and department name(s) they work for, who live(s) in the city named "Houston".

Employees table contains employees' fisrt/last names and **address** data, but **department names** are not included**.** Also, address field involves city information however it is not just consist of this data. Therefore checking if **address** field contains **Houston** data or not (***address like '%Houston%'***) should be done on the **Employee** table; and **department names** should be attained from **department** table. Here again there will two tables in the query:

Query:

```
select fname,lname, dname from department d, employee e where e.address like '%Houston%' and
d.dnumber=e.dno;
```

Since first/last name and department name are asked in the query, after select ***fname***, ***lname***, ***dname*** are selected; as condition it is specified that we are interested in the departments of the employees who lives in "***Houston***", work for; two tables are joined. **dno** in the employee table refers to **dnumber** in the department table. Therefore matching is done through these two columns.

Result:

FNAME	LNAME	DNAME
Franklin	Wong	Research
James	Borg	Headquarters
John	Smith	Research
Joyce	English	Research
Ahmad	Jabbar	Administration