Program Name : Masters of Computer Application

Semester : 2

Paper Title : Database Management System(DBMS)

Submitted by : Rishi Raj Chaurasia

Examination Roll Number : 19234757031

Assignment 3 : SQL Queries

Q1. Toy Database.

1.1 Create the database tables for Product, Manufacturer, Supplier, Inventory, Manufacturers and Supplies with above attributes.

```
CREATE DATABASE toy_store;

USE toy_store;

CREATE TABLE product(
pid INTEGER NOT NULL
name VARCHAR(20),
min_age INTEGER,
PRIMARY KEY(pid)
);


CREATE TABLE manufacturer(
mid INTEGER NOT NULL,
name VARCHAR(20),
address VARCHAR(50),
PRIMARY KEY(mid)
);


CREATE TABLE supplier(
sid INTEGER NOT NULL,
name VARCHAR(20),
address VARCHAR(50),
PRIMARY KEY(sid)
);


CREATE TABLE inventory(
AUTO_INCREMENT,
AUTO_INCREMENT,
AUTO_INCREMENT,
pid INTEGER,
stock INTEGER,
PRIMARY KEY(pid),
FOREIGN KEY(pid) REFERENCES product(pid) ON DELETE CASCADE ON UPDATE
CASCADE
);


CREATE TABLE manufactures(
mid INTEGER,
pid INTEGER,
PRIMARY KEY(mid, pid),
FOREIGN KEY(mid) REFERENCES manufacturer(mid) ON DELETE CASCADE ON
UPDATE CASCADE,
FOREIGN KEY(pid) REFERENCES product(pid) ON DELETE CASCADE ON UPDATE
CASCADE
);


CREATE TABLE supplies(
sid INTEGER,
pid INTEGER,
```

```
PRIMARY KEY(sid, pid),
FOREIGN KEY(sid) REFERENCES supplier(sid) ON DELETE CASCADE ON UPDATE
CASCADE,
FOREIGN KEY(pid) REFERENCES product(pid) ON DELETE CASCADE ON UPDATE
CASCADE );
```

1.2 Insert values into Product, Manufacturer, Supplier, Inventory, Manufacturers and Supplies tables.

```
INSERT INTO product(name,min_age)
VALUES('Ben 10 car', 5), ('Avengers Ship', 8), ('Doraemon Ball', 6),
('Shinchan Tatoo', 14), ('Power Rangers 3D', 12), ('Barbie Doll', 4),
('Mickey Mouse Sticker', 5), ('Cars', 10),
('Bay Blade', 16),('Pokemon Ball', 8);

INSERT INTO manufacturer(name, address)
VALUES('Fun Zoo Toys', '49, Greater Noida, Gautam Budh Nagar'), ('Acctu
Toys', 'B-30 Katju Nagar, Ground Floor, Jadavpur, Kolkata'), ('Pals Plush',
'260 Alstonia Drive, Sri City Sez, Chittoor'), ('Jumboo', 'B/42, Corporate
House, SG Highway, Ahmedabad'),
('Dimpy Stuff', 'RZ 116-118/B Narsingh Industrial Area, New Delhi');

INSERT INTO supplier(name, address)
VALUES('Natkhat', 'Plot No. 155 & 156, Toy City, Greater Noida'),
('Amusement Games', 'Plot No.B-3, Chehar Estate Ahmedabad, Gujarat'),
('Model Makers', 'B-2, 127/1, Gem Wellington, Bengaluru'),
('Toys India', '2nd Floor, Hathi Khana, Chandni Chowk,Delhi'), ('Choice
Creation', 'Shop No. 3, J.B. Shah Market, Mumbai');

INSERT INTO inventory(pid, stock)
VALUES(1, 12), (2, 10), (3, 7), (4, 6), (5, 15), (6, 4), (7, 8), (8, 20),
(9, 1), (10, 10);

INSERT INTO manufactures(mid, pid)
VALUES(1, 1), (2, 2), (3, 3), (4, 4), (5, 5), (5, 6), (4, 7), (3, 8), (2,
9), (1, 10);

INSERT INTO supplies(sid, pid)
VALUES(1, 1), (4, 1), (5, 2), (2, 3), (3, 3), (5, 4), (4, 4), (2, 5), (1,
6), (4, 7), (3, 7), (2, 7),
(5, 8), (1, 8), (2, 9), (4, 10), (2, 10);
```

1.3 List the ids and names of all products whose inventory is below 5.

```
SELECT pid, name
FROM product
NATURAL JOIN inventory WHERE stock<5;
```

1.4 List the ids and names of all suppliers for products manufactured by "manufacturer_2". The id and name of each supplier should appear only once.

```
SELECT sid, name
FROM supplier
NATURAL JOIN supplies NATURAL JOIN manufactures WHERE mid = 2;
```

1.5 List the ids, names, and number in stock of all products in inventory. Order the list by decreasing number in stock and decreasing product ids.

```
SELECT pid, name, stock
FROM product
NATURAL JOIN inventory
ORDER BY stock DESC, pid DESC;
```

1.6 List the ids and names of all products for whom there is only one supplier.

```
SELECT pid, name FROM product
NATURAL JOIN supplies GROUP BY pid

HAVING COUNT(*) = 1;
```

1.7 Find the ids and names of the products with the lowest inventory. Do NOT assume these are always products with an inventory of zero.

```
SELECT pid, name
FROM product
NATURAL JOIN inventory
where stock = (SELECT MIN(stock) FROM inventory);
```

1.8 List the id and name of each supplier along with the total number of products it supplies.

```
SELECT sid, name, count(*) AS total_product FROM supplier
NATURAL JOIN supplies
GROUP BY sid;
```

1.9 Find the id and name of the manufacturer who produces toys on average for the youngest Children.

```
SELECT mid, name FROM manufacturer m NATURAL JOIN
(SELECT mid, AVG(min_age) AS avg_age FROM product p JOIN manufactures ms on
ms.pid=p.pid
GROUP BY mid
ORDER BY avg_age
LIMIT 1) A;
```

Q. 2. Consider Company database schema from chapter 5(according to 5th edition). Write SQL queries for the following and executes them in Oracle.

```
CREATE DATABASE COMPANY;

USE COMPANY;

CREATE TABLE EMPLOYEE( Fname VARCHAR(20),

Minit CHAR(1),
Lname VARCHAR(20),
Ssn INTEGER UNSIGNED,
Bdate DATE,
Address VARCHAR(50),
Sex CHAR(1),
Salary MEDIUMINT UNSIGNED,
Super_ssn INTEGER UNSIGNED,
Dno TINYINT UNSIGNED,
PRIMARY KEY(Ssn),
FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn) ON DELETE SET NULL
```

```
);

CREATE TABLE DEPARTMENT( Dname VARCHAR(20),
Dnumber TINYINT UNSIGNED,
Mgr_ssn INTEGER UNSIGNED,
Mgr_start_date DATE,
PRIMARY KEY(Dnumber),
FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn) ON DELETE SET NULL

);

CREATE TABLE DEPT_LOCATIONS(
Dnumber TINYINT UNSIGNED,
Dlocation VARCHAR(20),
PRIMARY KEY(Dnumber, Dlocation),
FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber) ON DELETE
CASCADE );

CREATE TABLE PROJECT( Pname VARCHAR(20),
Pnumber TINYINT UNSIGNED,
Plocation VARCHAR(20),
Dnum TINYINT UNSIGNED,
PRIMARY KEY(Pnumber),
FOREIGN KEY (Dnum) REFERENCES DEPT_LOCATIONS(Dnumber) ON DELETE SET NULL );

CREATE TABLE WORKS_ON(
Essn INTEGER UNSIGNED,
Pno TINYINT UNSIGNED,
Hours DECIMAL(4,1),
PRIMARY KEY(Essn, Pno),
FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn) ON DELETE CASCADE,
FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber) ON DELETE CASCADE

);

CREATE TABLE DEPENDENT( Essn INTEGER UNSIGNED,

Dependent_name VARCHAR(20),

Sex CHAR(1),
Bdate DATE,
Relationship VARCHAR(20),
PRIMARY KEY(Essn, Dependent_name),
FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn) ON DELETE CASCADE

);
SET FOREIGN_KEY_CHECKS = 0;

INSERT INTO EMPLOYEE(Fname, Minit, Lname, Ssn, Bdate, Address, Sex, Salary,
Super_ssn, Dno)
VALUES('John', 'B', 'Smith', 123456789, '1965-01-09', '731 Fondren,
Houston, TX', 'M', 30000, 333445555, 5),
('Franklin', 'T', 'Wong', 333445555, '1955-12-08', '638 Voss, Houston, TX',
'M', 40000, 888665555, 5),
('Alicia', 'J', 'Zelaya', 999887777, '1968-01-19', '3321 Castle, Spring,
TX', 'F', 25000, 987654321, 4),
('Jennifer', 'S', 'Wallace', 987654321, '1941-06-20', '291 Berry, Bellaire,
TX', 'F', 43000, 888665555, 4),
('Ramesh', 'K', 'Narayan', 666884444, '1962-09-15', '975 Fire Oak, Humble,
```

```
TX', 'M', 38000, 333445555, 5),
('Joyce', 'A', 'English', 453453453, '1972-07-31', '5631 Rice, Houston,
TX', 'F', 25000, 333445555, 5),
('Ahmad', 'V', 'Jabbar', 987987987, '1969-03-29', '980 Dallas, Houston,
TX', 'M', 25000, 987654321, 4),
('James', 'E', 'Borg', 888665555, '1937-11-10', '450 Stone, Houston, TX',
'M', 55000, NULL, 1);

SET FOREIGN_KEY_CHECKS = 1;

INSERT INTO DEPARTMENT(Dname, Dnumber, Mgr_ssn, Mgr_start_date)
VALUES('Research', 5, 333445555, '1988-05-22'), ('Administration', 4,
987654321, '1995-01-01'), ('Headquarters', 1, 888665555, '1981-06-19');

INSERT INTO DEPT_LOCATIONS(Dnumber, Dlocation)
VALUES(1, 'Houston'), (4, 'Stafford'), (5, 'Bellaire'), (5, 'Sugarland'),
(5, 'Houston');

INSERT INTO PROJECT(Pname, Pnumber, Plocation, Dnum)
VALUES('ProductX', 1, 'Bellaire', 5), ('ProductY', 2, 'Sugarland', 5),
('ProductZ', 3, 'Houston', 5), ('Computerization', 10, 'Stafford', 4),
('Reorganization', 20, 'Houston', 1), ('Newbenefites', 30, 'Stafford', 4);

INSERT INTO WORKS_ON(Essn, Pno, Hours)
VALUES(123456789, 1, 32.5), (123456789, 2, 7.5), (666884444, 3, 40.0),
(453453453, 1, 20.0), (453453453, 2, 20.0), (333445555, 2, 10.0),
(333445555, 3, 10.0), (333445555, 10, 10.0), (333445555, 20, 10.0),
(999887777, 30, 30.0), (999887777, 10, 10.0), (987987987, 10, 35.0),
(987987987, 30, 5.0), (987654321, 30, 20.0), (987654321, 20, 15.0),
(888665555, 20, NULL);

INSERT INTO DEPENDENT(Essn, Dependent_name, Sex, Bdate, Relationship)
VALUES(333445555, 'Alice', 'F', '1986-04-05', 'Daughter'), (333445555,
'Theodore', 'M', '1983-10-25', 'Son'),
(333445555, 'Joy', 'F', '1958-05-03', 'Spouse'),
(987654321, 'Abner', 'M', '1942-02-28', 'Spouse'), (123456789, 'Michael',
'M', '1988-01-04', 'Son'), (123456789, 'Alice', 'F', '1988-12-30',
'Daughter'), (123456789, 'Elizabeth', 'F', '1967-05-05', 'Spouse');
```

2.1. Retrieve the first and last names and department number and name of all employees directly supervised by James Borg. Show results in ascending alpha order (by last name and then first name).

```
SELECT Fname, Lname, Dno, Dname
FROM EMPLOYEE JOIN DEPARTMENT ON Dno = Dnumber
WHERE Super_ssn = (SELECT Ssn FROM EMPLOYEE WHERE Fname = 'James' AND Lname
= 'Borg')
ORDER BY Lname, Fname;
```

2.2 Retrieve the name and number of departments which have employees who do not work on at least one project. Show results in ascending alpha order. (NOTE: a department should appear on this list if it has an employee who does not work on any project at all.)

```
SELECT DISTINCT(d.Dnumber),d.Dname FROM DEPARTMENT d, EMPLOYEE e WHERE
d.Dnumber=e.Dno AND e.Ssn IN (SELECT Essn FROM WORKS_ON
GROUP BY Essn
HAVING COUNT(Pno)<(SELECT COUNT(Pnumber) FROM PROJECT)) ORDER BY
d.Dname,d.Dnumber;
```

2.3 For each department, list the department name and the total number of hours assigned to projects controlled by the department (irrespective of the employee to whom they are assigned) and the total number of hours assigned to employees of the department (irrespective of the project involved). Show results in ascending alpha order.

```
SELECT Dname,hours_project,hours_employee
FROM DEPARTMENT D JOIN (SELECT Dnum,SUM(Hours) hours_project FROM
PROJECT,WORKS_ON
WHERE Pnumber=Pno
GROUP BY Dnum) P ON D.Dnumber = P.Dnum
JOIN (SELECT Dno,SUM(Hours) hours_employee
FROM EMPLOYEE,WORKS_ON
WHERE Ssn=Essn
GROUP BY Dno) E ON D.Dnumber = E.Dno
ORDER BY D.Dname;
```

2.4 Retrieve the names of departments which have at least one project which employs every one of the employees of the department that controls the project. Also show the name of the project. Show results in ascending alpha order.

```
SELECT D.Dname,Pname
FROM DEPARTMENT D, (SELECT Pname, COUNT(Essn) emp_proj,Dnum
FROM PROJECT,WORKS_ON,EMPLOYEE
WHERE Pnumber = Pno AND Dno=Dnum AND Essn=Ssn
GROUP BY Dnum,Pnumber) P, (SELECT COUNT(Ssn) emp_dept,Dno
FROM EMPLOYEE
GROUP BY Dno) E
WHERE P.emp_proj = E.emp_dept AND P.Dnum = E.Dno AND D.Dnumber = E.Dno;
```

2.5 Retrieve the first and last names of employees who work on projects which are not controlled by their departments. Also show the names of the projects, the employee's department number, and the number of the project's controlling department. (All of this should be shown in the same result table.) Show results in ascending alpha order (by the last name and then the first name and then project name).

```
SELECT e.Fname, e.Lname, p.Pname project_name ,p.Dnum
controlling_dept,e.Dno dept_number
FROM EMPLOYEE e, PROJECT p, WORKS_ON w
WHERE e.Ssn=w.Essn AND w.Pno=p.Pnumber AND Dno!=Dnum ORDER BY
e.Lname,e.Fname,p.Pname;
```

2.6 Retrieve the first and last names of employees who work on more than the average number of projects. (Note: employees who do not work on any project are to be included in the average.) Display their names, the number of projects they work on, and the average number of projects. (The same average should be repeated in each row.) Show results in ascending alpha order (by the last name and then the first name). [The average number of projects is the average number of projects worked on per employee].

```
SELECT DISTINCT e.Fname, e.Lname, r.proj_count AS project_count ,
s.avg_proj AS average_project
FROM EMPLOYEE e, WORKS_ON w, (SELECT AVG(q.Pnum) AS avg_proj
FROM (SELECT COUNT(Pno) AS Pnum FROM WORKS_ON GROUP BY Essn) AS q) AS s,
(SELECT COUNT(Pno) AS proj_count,Essn FROM WORKS_ON GROUP BY Essn) AS r
WHERE e.Ssn = w.Essn AND r.proj_count>s.avg_proj AND w.Essn = r.Essn ORDER
BY e.Lname,e.Fname;
```

2.7 Retrieve the name and number of the project which uses the most employees. Also, show the total number of employees for that project. If there is more than one project that has gained that maximum, list them all. Show results in ascending alpha order.

```
SELECT p.Pname, p.Pnumber, b.emp_count
FROM PROJECT AS p, (SELECT Pno,COUNT(*) AS emp_count FROM WORKS_ON GROUP BY
Pno
HAVING COUNT(Essn) = (SELECT MAX(a.emp_count) FROM (SELECT COUNT(*) AS
emp_count FROM WORKS_ON
GROUP BY pno) AS a)) AS b
WHERE b.Pno = p.Pnumber
ORDER BY p.Pname;
```

2.8 Do any departments have a location in which they have no projects? Retrieve the names of departments which have at least one location which is not the same as any of the location of the department's projects. Show results in ascending alpha order. [This means that one department location is different from every location of every project of that department.]

```
SELECT d.Dname,l.Dlocation
FROM DEPARTMENT AS d,dept_locations AS l
WHERE l.Dlocation NOT IN (SELECT Plocation
FROM PROJECT AS p WHERE p.Dnum=l.Dnumber) AND d.Dnumber=l.Dnumber ORDER BY
Dname;
```

2.9 List the names of dependents that have the same first name as an employee of whom they are not the dependent. Also, show the ssn of the employee with the same first name and the ssn of the employee on whom the dependent is dependent (dependent.essn). (All of this should be shown in the same table.) Show results in ascending alpha order.

```
SELECT Dependent_name, Ssn AS same_name_emp, Essn AS depend_upon FROM
DEPENDENT, EMPLOYEE
WHERE Dependent_name = Fname and Ssn!=Essn
ORDER BY Dependent_name;
```

2.10 Retrieve the first and last names of employees whose supervisor works on any project outside the employee's department. Show results in ascending alpha order (by the last name and then the first name). [Note that you are to retrieve the employee's name, not the supervisor's.]

```
SELECT DISTINCT (Fname),Lname
FROM EMPLOYEE,PROJECT,WORKS_ON
WHERE Dno != Dnum AND Super_ssn = Essn AND Pno = Pnumber ORDER BY Lname
,Fname;
```
