

Homework 7

Your Name: Naga Satya Silpa Annadevara

Student ID: A20517818

Q1. Assume we have a database which stores product and equipment information, as shown below. Use subqueries to answer the following questions [30]

```
Product(maker, model, type)
PC(model, speed, ram, hd, price)
Laptop(model, speed, ram, hd, screen, price)
Printer(model, color, type, price)
```

- a) Find the makers of PC's with a speed of at least 3.0.
- b) Find the printers with the highest price.
- c) Find the laptops whose speed is slower than that of any PC.
- d) Find the model number of the item (PC, laptop, or printer) with the highest price.
- e) Find the maker of the color printer with the lowest price.
- f) Find the maker(s) of the PC(s) with the fastest processor among all those PC's that have the smallest amount of RAM.

SOLUTION:

```
a)
SELECT DISTINCT maker
FROM Product
WHERE model IN (SELECT model FROM PC WHERE speed >= 3);
```

```
b)
SELECT model, price
FROM Printer
WHERE price IN (SELECT MAX(price) FROM Printer);
```

c)

```
SELECT model
FROM Laptop
WHERE Laptop.speed < (SELECT MIN(speed) FROM PC);
```

d)

```
WITH Model_Price AS (SELECT model, price
FROM PC
UNION
SELECT model, price
FROM Laptop
UNION
SELECT model, price
FROM Printer)
SELECT model
FROM Model_Price
WHERE price = (SELECT MAX(price) FROM Model_Price);
```

e)

```
SELECT DISTINCT p.maker
FROM product p , printer r
WHERE p.model=r.model
AND r.price=(SELECT min(price) FROM printer);
```

Note: If the color column in the table specifies the type of print (i.e. color = true for a color printer and color = false for a black and white printer, we can include color=true in the where clause to select minimum price of only color printers.

f)

```
SELECT DISTINCT maker
FROM Product
WHERE model IN (SELECT model FROM PC WHERE speed=(SELECT
MAX(speed) FROM PC WHERE ram= (SELECT MIN(ram) FROM PC)));
```

Q2. Use appropriate SQL (regular, subqueries, joins, etc.) to answer the following questions by using the same database in Q1. Note, the “manufacturer” is referred as the “maker” in the table “product”. [30]

- a) Find the average speed of PC's.
- b) Find the average speed of laptops costing over \$1000.
- c) Find the average price of PC's made by manufacturer "A."
- d) Find the average price of PC's and laptops made by manufacturer "D."
- e) Find, for each different speed, the average price of a PC.
- f) Find for each manufacturer, the average screen size of its laptops.
- g) Find the manufacturers that make at least three different models of PC.
- h) Find for each manufacturer who sells PC's the maximum price of a PC.
- i) Find, for each speed of PC above 2.0, the average price.
- j) Find the average hard disk size of a PC for all those manufacturers that make printers.

SOLUTION:

a)
 SELECT AVG(speed) FROM PC;

b)
 SELECT AVG(speed) FROM Laptop WHERE price>1000;

c)
 SELECT AVG(price) AS "Avg_Price"
 FROM Product p, PC c
 WHERE p.model = c.model
 AND p.maker='A';

d)
 SELECT AVG(price) AS "Avg_Price"
 FROM (SELECT PC.price FROM PC
 INNER JOIN Product
 ON PC.model = Product.model

```
WHERE maker = 'D'  
UNION ALL  
SELECT Laptop.price  
FROM Laptop  
INNER JOIN product  
ON Laptop.model = Product.model  
WHERE maker = 'D');
```

e)

```
SELECT speed, AVG(price) AS "Avg_Price"  
FROM PC  
GROUP BY speed;
```

f)

```
SELECT p.maker, AVG(l.screen) AS "Avgscreen_size"  
FROM product p  
INNER JOIN laptop l  
ON p.model=l.model  
GROUP BY p.maker;
```

g)

```
SELECT maker, COUNT(DISTINCT model) AS "Number of models"  
FROM product  
WHERE type = 'PC'  
GROUP BY maker  
HAVING COUNT(DISTINCT model) >= 3;
```

h)

```
SELECT Product.maker, MAX(PC.price) AS "Max Price"  
FROM Product  
INNER JOIN PC  
ON Product.model = PC.model  
GROUP BY Product.maker ;
```

i)

```
SELECT speed, AVG(price) AS "Avg_Price"  
FROM PC  
WHERE speed >2.0  
GROUP BY speed;
```

j)

```
SELECT p.maker, AVG(c.hd) AS "Average hardDisk Size"
```

```
FROM product p
JOIN PC c
ON p.model = c.model
WHERE p.maker IN (SELECT p.maker FROM product WHERE type='printer')
GROUP BY p.maker;
```

Q3. Using SQL to answer the following questions, by using the database in Q1 [20]

- a) Using two **INSERT** statements, store in the database the fact that PC model 1100 is made by manufacturer C, has speed 3.2, RAM 1024, hard disk 180, and sells for \$2499.
- b) Insert the facts that for every PC there is a laptop with the same manufacturer, speed, RAM, and hard disk, a 17-inch screen, a model number 1100 greater, and a price \$500 more.
- c) Delete all PC's with less than 100 gigabytes of hard disk.
- d) Delete all laptops made by a manufacturer that doesn't make printers.
- e) Manufacturer A buys manufacturer B. Change all products made by B so they are now made by A.

SOLUTION:

a)

```
INSERT INTO Product VALUES ('C', 1100, 'PC')
INSERT INTO PC VALUES (1100 , 3.2, 1024, 180, 2499);
```

b)

```
INSERT INTO
  Product(maker, model, type) (
  SELECT
    maker,
    model + 1100,
    'Laptop'
  FROM
    Product
```

```
WHERE
    type = 'PC'
);
```

```
INSERT INTO
    Laptop(model, speed, ram, hd, screen, price) (
        SELECT
            model + 1000,
            speed,
            ram,
            hd,
            17,
            price + 500
        FROM
            PC
    );
```

c)

-- Delete records from both PRODUCT and PC Tables

```
DELETE *
FROM PRODUCT
WHERE MODEL IN (
    SELECT MODEL
    FROM PC WHERE hd<100
) AND TYPE = 'PC';
```

```
DELETE * FROM PC WHERE hd<100;
(Assuming that the hard disc can store the data in gigabytes)
```

d)

```
DELETE * FROM Laptop
WHERE model IN
(SELECT p.model FROM product p, laptop l
WHERE p.model = l.model
AND maker IN
((SELECT DISTINCT maker FROM product)
EXCEPT
(SELECT DISTINCT maker FROM product
WHERE type = 'printer')));
```

e)

```
UPDATE Product  
SET maker = 'A'  
WHERE maker = 'B';
```

Q4. Give SQL for the following questions, and run them on the HR schema, give outputs in Oracle [20]

- Get employee ID and the number of his/her job histories, rank the outputs by the number of job histories in a descending way
- Return employee ID, name, emails, if he or she has more than 1 job history
- Return employee ID, name, and his or her all job titles
- Get department ID, name, and the number of employees in each department, sort records by the number of employees in descending order
- Get manager ID, name, and the number of employees he or she supervised

SOLUTION:

a)

```
SELECT EMPLOYEE_ID, COUNT(*) AS "Number_Of_Jobs"  
FROM JOB_HISTORY  
GROUP BY EMPLOYEE_ID  
ORDER BY COUNT(*) DESC;
```

The screenshot shows the Oracle SQL Developer interface. The 'Connections' pane on the left shows the 'hr' schema with various tables. The 'Worksheet' pane contains the following SQL query:

```
SELECT EMPLOYEE_ID, COUNT(*) AS "Number_Of_Jobs"
FROM JOB_HISTORY
GROUP BY EMPLOYEE_ID
ORDER BY COUNT(*) DESC;
```

The 'Query Result' pane displays the results of the query, showing 7 rows. The status bar indicates 'All Rows Fetched: 7 in 0.001 seconds'.

EMPLOYEE_ID	Number_Of_Jobs
1	2
2	2
3	2
4	1
5	1
6	1
7	1

b)

```
SELECT e.EMPLOYEE_ID, e.FIRST_NAME, e.LAST_NAME, e.EMAIL,
COUNT(e.EMPLOYEE_ID) "History"
FROM EMPLOYEE e, JOB_HISTORY jh
WHERE e.EMPLOYEE_ID = jh.EMPLOYEE_ID
HAVING COUNT (e.EMPLOYEE_ID) >1
GROUP BY e.EMPLOYEE_ID, e.FIRST_NAME, e.LAST_NAME, e.EMAIL;
```

The screenshot shows the Oracle SQL Developer interface. The 'Connections' pane on the left shows the 'hr' schema with various tables. The 'Worksheet' pane contains the following SQL query:

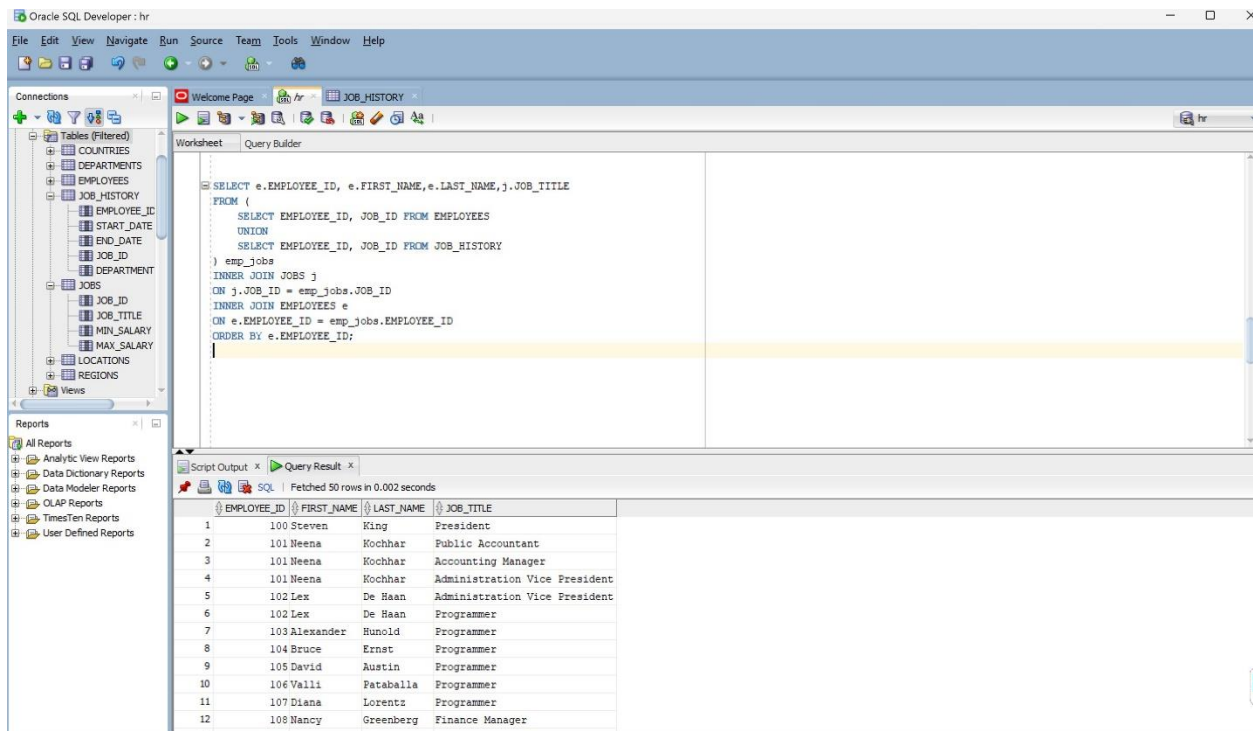
```
SELECT e.EMPLOYEE_ID, e.FIRST_NAME, e.LAST_NAME, e.EMAIL, COUNT(e.EMPLOYEE_ID) "History" FROM EMPLOYEE e, JOB_HISTORY jh WHERE e.EMPLOYEE_ID = jh.EMPLOYEE_ID HAVING COUNT(e.EMPLOYEE_ID) >1 GROUP BY e.EMPLOYEE_ID, e.FIRST_NAME, e.LAST_NAME, e.EMAIL;
```

The 'Query Result' pane displays the results of the query, showing 3 rows. The status bar indicates 'All Rows Fetched: 3 in 0.004 seconds'.

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	History	
1	101	Weena	Kochhar	WKOCHHAR	2
2	176	Jonathan	Taylor	JTAYLOR	2
3	200	Jennifer	Whalen	JWHALEN	2

c)

```
SELECT e.EMPLOYEE_ID, e.FIRST_NAME, e.LAST_NAME, JOB_TITLE
FROM (
    SELECT EMPLOYEE_ID, JOB_ID FROM EMPLOYEES
    UNION
    SELECT EMPLOYEE_ID, JOB_ID FROM JOB_HISTORY
) emp_jobs
INNER JOIN JOBS j
ON j.JOB_ID = emp_jobs.JOB_ID
INNER JOIN EMPLOYEE e
ON e.EMPLOYEE_ID = emp_jobs.EMPLOYEE_ID
ORDER BY e.EMPLOYEE_ID;
```



d)

```
SELECT d.DEPARTMENT_ID, d.DEPARTMENT_NAME,
COUNT(e.EMPLOYEE_ID) AS "Number_Of_Employees"
FROM DEPARTMENTS d
LEFT OUTER JOIN EMPLOYEES e
ON d.DEPARTMENT_ID, d.DEPARTMENT_NAME
ORDER BY COUNT(e.EMPLOYEE_ID) DESC;
```

Oracle SQL Developer : hr

File Edit View Navigate Run Source Team Tools Window Help

Connections

Tables (Filtered)

- COUNTRIES
- DEPARTMENTS
 - DEPARTMENT_ID
 - DEPARTMENT_NAME
 - MANAGER_ID
 - LOCATION_ID
- EMPLOYEES
 - EMPLOYEE_ID
 - FIRST_NAME
 - LAST_NAME
 - EMAIL
 - PHONE_NUMBER
 - HIRE_DATE
 - JOB_ID
 - SALARY
 - COMMISSION_PCT
 - MANAGER_ID
 - DEPARTMENT_ID

Reports

- All Reports
- Analytic View Reports
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- TimesTen Reports
- User Defined Reports

Worksheet

```

SELECT d.DEPARTMENT_ID,d.DEPARTMENT_NAME, COUNT(e.EMPLOYEE_ID) AS "Number_of_Employees"
FROM DEPARTMENTS d
LEFT OUTER JOIN EMPLOYEES e
ON d.DEPARTMENT_ID = e.DEPARTMENT_ID
GROUP BY d.DEPARTMENT_ID,d.DEPARTMENT_NAME
ORDER BY COUNT(e.EMPLOYEE_ID) DESC;

```

Script Output x Query Result x

SQL | All Rows Fetched: 27 in 0.002 seconds

DEPARTMENT_ID	DEPARTMENT_NAME	Number_of_Employees
50	Shipping	45
80	Sales	34
30	Purchasing	6
100	Finance	6
60	IT	5
90	Executive	3
110	Accounting	2
20	Marketing	2
40	Human Resources	1
70	Public Relations	1
10	Administration	1
230	IT Helpdesk	0
240	Government Sales	0
250	Retail Sales	0
260	Recruiting	0
220	NOC	0
210	IT Support	0
200	Operations	0
190	Contracting	0
180	Construction	0

e)

```

SELECT emp.MANAGER_ID mgr.FIRST_NAME, mgr.LAST_NAME , COUNT(*)
AS "num_of_employees"
FROM EMPLOYEES emp, EMPLOYEES mgr
WHERE emp.MANAGER_ID = mgr.EMPLOYEE_ID
GROUP BY emp.MANAGER_ID, mgr.FIRST_NAME, mgr.LAST_NAME
ORDER BY emp.MANAGER-ID ASC;

```

Oracle SQL Developer : hr

File Edit View Navigate Run Source Team Tools Window Help

Connections

Oracle Connections

- connection
- hr
 - Tables (Filtered)
 - COUNTRIES
 - DEPARTMENTS
 - EMPLOYEES
 - JOB_HISTORY
 - JOBS
 - LOCATIONS
 - REGIONS
 - Views
 - Indexes
 - Packages
 - Procedures
 - Functions
 - Operators
 - Queues

Reports

- All Reports
- Analytic View Reports
- Data Dictionary Reports
- Data Modeler Reports
- OLAP Reports
- TimesTen Reports
- User Defined Reports

Worksheet

```

SELECT emp.MANAGER_ID,mgr.FIRST_NAME, mgr.LAST_NAME, COUNT(*) AS "num_of_employees"
FROM EMPLOYEES emp, EMPLOYEES mgr
WHERE emp.MANAGER_ID = mgr.EMPLOYEE_ID
GROUP BY emp.MANAGER_ID,mgr.FIRST_NAME, mgr.LAST_NAME
ORDER BY emp.MANAGER-ID ASC;

```

Query Result x

SQL | All Rows Fetched: 18 in 0.003 seconds

MANAGER_ID	FIRST_NAME	LAST_NAME	num_of_employees
100	Steven	King	14
101	Neena	Kochhar	5
102	Lex	De Haan	1
103	Alexander	Hunold	4
108	Nancy	Greenberg	5
114	Den	Raphaely	5
120	Matthew	Weiss	8
121	Adam	Fripp	8
122	Payam	Kaufling	8
123	Shanta	Vollman	8
124	Kevin	Mourgos	8
145	John	Russell	6
146	Karen	Partners	6
147	Alberto	Errazuriz	6
148	Gerald	Cambraut	6
149	Eleni	Zlotkey	6
201	Michael	Hartstein	1
205	Shelley	Higgins	1

Feedback and grade:
97 marks

G

HW7

3. d. -3 Incorrect syntax. The asterisk symbol (*) should not be included in the DELETE statement. The EXCEPT operator is not supported in Oracle. Instead, you should use the MINUS operator to achieve the same result.