DEPARTMENT OF COMPUTER SCIENCE

The University of Chittagong

Student Name:	Sumaiya Tabassum
Student ID:	23701025
Session:	2022-2023
Course Title:	Database Systems Lab
Course Code:	CSE 414
Task:	Assignment 03 - Oracle Database 9i: SQL I, II Chap 08, 09, 10, 18 Exercise
Submitted to:	Rudra Pratap Deb Nath, Associate Professor, Department of Computer Science and Engineering
Submission Date:	June 14, 2025

Contents

Chapter 08

Problem 1

Q: Run the statement in the `lab8_1.sql` script to build the `MY_EMPLOYEE` table to be used for the lab.

A:

`lab8_1.sql`

```
CREATE TABLE my_employee
   (id NUMBER(4) CONSTRAINT my_employee_id_nn NOT NULL,
   last_name VARCHAR2(25),
   first_name VARCHAR2(25),
   userid VARCHAR2(8),
   salary NUMBER(9,2));
```

Problem 2

Q: Describe the structure of the `MY_EMPLOYEE` table to identify the column names.

Name	Null?	Type
ID	NOT NULL	NUMBER(4)
LAST_NAME		VARCHAR2(25)
FIRST_NAME		VARCHAR2(25)
USERID		VARCHAR2(8)
SALARY		NUMBER(9,2)

A:

```
DESCRIBE my_employee;
```

```
Name       Null?    Type
---------- -------- ------------
ID         NOT NULL NUMBER(4)
LAST_NAME           VARCHAR2(25)
FIRST_NAME          VARCHAR2(25)
USERID              VARCHAR2(8)
SALARY              NUMBER(9,2)
```

Problem 3

Q: Add the first row of data to the MY_EMPLOYEE table from the following sample data. Do not list the columns in the INSERT clause.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860
3	Biri	Ben	bbiri	1100
4	Newman	Chad	cnewman	750
5	Ropeburn	Audrey	aropebur	1550

A:

```
INSERT INTO my_employee
VALUES (1, 'Patel', 'Ralph', 'rpatel', 895);
```

Problem 4

Q: Populate the MY_EMPLOYEE table with the second row of sample data from the preceding list. This time, list the columns explicitly in the INSERT clause.

A:

```
INSERT INTO my_employee (id, last_name, first_name, userid, salary)
VALUES (2, 'Dancs', 'Betty', 'bdancs', 860);
```

Problem 5

Q: Confirm your addition to the table.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860

A:

```
SELECT *
FROM my_employee;
```

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860

Problem 6

Q: Write an `INSERT` statement in a text file named `loademp.sql` to load rows into the `MY_EMPLOYEE` table. Concatenate the first letter of the first name and the first seven characters of the last name to produce the user ID.

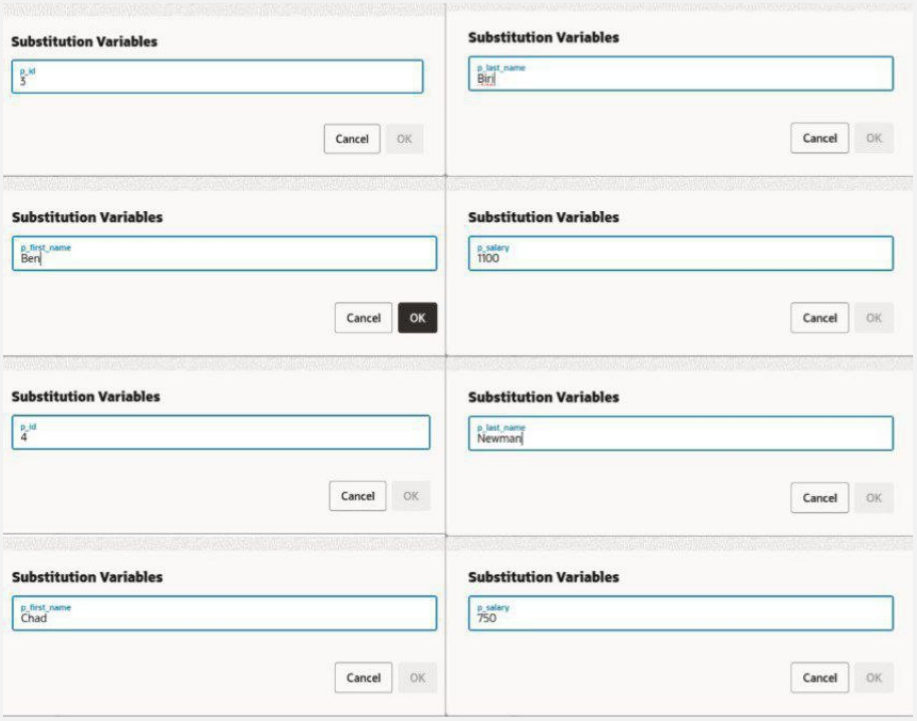
A:

```
INSERT INTO my_employee
VALUES (&p_id, '&p_last_name', '&p_first_name',
lower(substr('&p_first_name', 1, 1) ||
substr('&p_last_name', 1, 7)), &p_salary);
```

Problem 7

Q: Populate the table with the next two rows of sample data by running the `INSERT` statement in the script that you created.

A:



Problem 8

Q: Confirm your additions to the table.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860
3	Biri	Ben	bbiri	1100
4	Newman	Chad	cnewman	750

A:

```
SELECT *
FROM my_employee
ORDER BY id;
```

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	895
2	Dancs	Betty	bdancs	860
3	Biri	Ben	bbiri	1100
4	Newman	Chad	cnewman	750

Problem 9

Q: Confirm your additions to the table.

A:

```
COMMIT;
```

Problem 10

Q: Change the last name of employee 3 to Drexler.

A:

```
UPDATE my_employee
SET last_name = 'Drexler'
WHERE id = 3;
```

Problem 11

Q: Change the salary to 1000 for all employees with a salary less than 900.

A:

```
UPDATE my_employee
SET salary = 1000
WHERE salary < 900;
```

Problem 12

Q: Verify your changes to the table.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
2	Dancs	Betty	bdancs	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000

A:

```
SELECT *
FROM my_employee
ORDER BY id;
```

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
2	Dancs	Betty	bdancs	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000

Problem 13

Q: Delete Betty Dancs from the MY_EMPLOYEE table.

A:

```
DELETE FROM my_employee
WHERE last_name = 'Dancs';
```

Problem 14

Q: Confirm your changes to the table.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000

A:

```
SELECT *
FROM my_employee
ORDER BY id;
```

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000

Problem 15

Q: Commit all pending changes.

A:

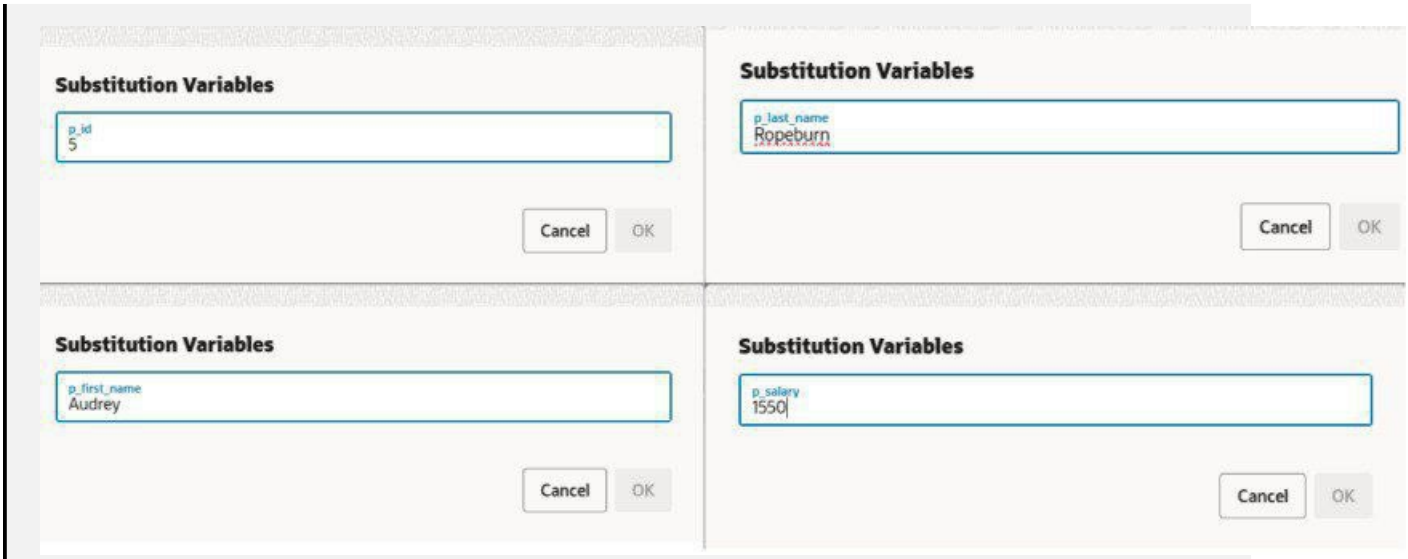
```
COMMIT;
```

Problem 16

Q: Populate the table with the last row of sample data by modifying the statements in the script that you created in step 6. Run the statements in the script.

A:

```
INSERT INTO my_employee
VALUES (&p_id, '&p_last_name', '&p_first_name',
lower(substr('&p_first_name', 1, 1) ||
substr('&p_last_name', 1, 7)), &p_salary);
```



Problem 17

Q: Confirm your addition to the table.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000
5	Ropeburn	Audrey	aropebur	1550

A:

```
SELECT *
FROM my_employee
ORDER BY id;
```

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000
5	Ropeburn	Audrey	aropebur	1550

Problem 18

Q: Mark an intermediate point in the processing of the transaction.

A:

```
SAVEPOINT step_18;
```

Problem 19

Q: Empty the entire table.

A:

```
DELETE
FROM my_employee;
```

Problem 20

Q: Confirm that the table is empty.

A:

```
SELECT *
FROM my_employee;
```

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
No items to display.				

Problem 21

Q: Discard the most recent `DELETE` operation without discarding the earlier `INSERT` operation.

A:

```
ROLLBACK TO step_18;
```

Problem 22

Q: Confirm that the new row is still intact.

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000
5	Ropeburn	Audrey	aropebur	1550

A:

```
SELECT *
FROM my_employee;
```

ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	Patel	Ralph	rpatel	1000
3	Drexler	Ben	bbiri	1100
4	Newman	Chad	cnewman	1000
5	Ropeburn	Audrey	aropebur	1550

Problem 23

Q: Make the data addition permanent.

A:

```
COMMIT;
```

Chapter 09

Problem 1

Q: Create the `DEPT` table based on the following table instance chart. Place the syntax in a script called `lab9_1.sql`, then execute the statement in the script to create the table.
Confirm that the table is created.

Column Name	ID	NAME
Key Type		
Nulls/Unique		
FK Table		
FK Column		
Data type	NUMBER	VARCHAR2
Length	7	25

Name	Null?	Type
ID		NUMBER(7)
NAME		VARCHAR2(25)

A:

`lab9_1.sql`

```
CREATE TABLE DEPT
   (id NUMBER(7),
   name VARCHAR(25));

DESCRIBE dept;
```

```
Name Null? Type
---- ----- ------------
ID         NUMBER(7)
NAME       VARCHAR2(25)
```

Problem 2

Q: Populate the `DEPT` table with data from the `DEPARTMENTS` table. Include only columns that you need.

A:

```
INSERT INTO dept
   SELECT department_id, department_name
   FROM hr.departments;
```

```
Name Null? Type
---- ----- ------------
ID         NUMBER(7)
NAME       VARCHAR2(25)
```

Problem 3

Q: Create the `EMP` table based on the following table instance chart. Place the syntax in a script called `lab9_3.sql`, and then execute the statement in the script to create the table. Confirm that the table is created.

Column Name	ID	LAST_NAME	FIRST_NAME	DEPT_ID
Key Type				
Nulls/Unique				
FK Table				
FK Column				
Data type	NUMBER	VARCHAR2	VARCHAR2	NUMBER
Length	7	25	25	7

Name	Null?	Type
ID		NUMBER(7)
LAST_NAME		VARCHAR2(25)
FIRST_NAME		VARCHAR2(25)
DEPT_ID		NUMBER(7)

A:

`lab9_3.sql`

```
CREATE TABLE EMP
   (id NUMBER(7),
   last_name VARCHAR2(25),
   first_name VARCHAR2(25),
   dept_id NUMBER(7));

DESCRIBE emp;
```

```
Name       Null? Type
---------- ----- ------------
ID                NUMBER(7)
LAST_NAME         VARCHAR2(25)
FIRST_NAME        VARCHAR2(25)
DEPT_ID           NUMBER(7)
```

Problem 4

Q: Modify the EMP table to allow for longer employee last names. Confirm your modification.

Name	Null?	Type
ID		NUMBER(7)
LAST_NAME		VARCHAR2(50)
FIRST_NAME		VARCHAR2(25)
DEPT_ID		NUMBER(7)

A:

```
ALTER TABLE emp
MODIFY (last_name VARCHAR2(50));

DESCRIBE emp;
```

```
Name       Null? Type
---------- ----- ------------
ID                NUMBER(7)
LAST_NAME         VARCHAR2(50)
FIRST_NAME        VARCHAR2(25)
DEPT_ID           NUMBER(7)
```

Problem 5

Q: Confirm that both the DEPT and EMP tables are stored in the data dictionary. (Hint: USER_TABLES)

TABLE_NAME
DEPT
EMP

A:

```
SELECT table_name
FROM user_tables
WHERE table_name IN ('DEPT', 'EMP');
```

	TABLE_NAME
1	DEPT
2	EMP

Problem 6

Q: Create the EMPLOYEES2 table based on the structure of the EMPLOYEES table. Include only the EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY, and DEPARTMENT_ID columns. Name the columns in your new table ID, FIRST_NAME, LAST_NAME, SALARY , and DEPT_ID, respectively.

A:

```
CREATE TABLE employees2 AS
SELECT employee_id id, first_name, last_name, salary, department_id
    dept_id
FROM hr.employees;
```

Problem 7

Q: Drop the EMP table.

A:

```
DROP TABLE emp;
```

Problem 8

Q: Rename the EMPLOYEES2 table as EMP.

A:

```
RENAME employees2 TO emp;
```

Problem 9

Q: Add a comment to the DEPT and EMP table definitions describing the tables. Confirm your additions in the data dictionary.

A:

```
COMMENT ON TABLE emp IS 'Employee Information';
COMMENT ON TABLE dept IS 'Department Information';
SELECT *
FROM user_tab_comments
WHERE table_name = 'DEPT'
OR table_name = 'EMP';
```

	TABLE_NAME	TABLE_TYPE	COMMENTS	ORIGIN_CON_ID
1	DEPT	TABLE	Department Informat	3
2	EMP	TABLE	Employee Informatio	3

Problem 10

Q: Drop the FIRST_NAME column from the EMP table. Confirm your modification by checking the description of the table.

A:

```
ALTER TABLE emp
DROP COLUMN first_name;

DESCRIBE emp;
```

```
Name      Null?    Type
--------- -------- ------------
ID                 NUMBER(6)
LAST_NAME NOT NULL VARCHAR2(25)
SALARY             NUMBER(8,2)
DEPT_ID            NUMBER(4)
```

Problem 11

Q: In the EMP table, mark the DEPT_ID column in the EMP table as UNUSED. Confirm your modification by checking the description of the table.

A:

```
ALTER TABLE emp
SET UNUSED (dept_id);

DESCRIBE emp;
```

```
Name      Null?    Type
--------- -------- ------------
ID                 NUMBER(6)
LAST_NAME NOT NULL VARCHAR2(25)
SALARY             NUMBER(8,2)
```

Problem 12

Q: Drop all the `UNUSED` columns from the `EMP` table. Confirm your modification by checking the description of the table.

A:

```
ALTER TABLE emp
DROP UNUSED COLUMNS;

DESCRIBE emp;
```

```
Name      Null?    Type
--------- -------- ------------
ID                 NUMBER(6)
LAST_NAME NOT NULL VARCHAR2(25)
SALARY             NUMBER(8,2)
```

Chapter 10

Problem 1

Q: Add a table-level `PRIMARY KEY` constraint to the `EMP` table on the `ID` column. The constraint should be named at creation. Name the constraint `my_emp_id_pk`.
Hint: The constraint is enabled as soon as the `ALTER TABLE` command executes successfully.

A:

```
ALTER TABLE emp
ADD CONSTRAINT my_emp_id_pk PRIMARY KEY (id);
```

Problem 2

Q: Create a `PRIMARY KEY` constraint to the `DEPT` table using the `ID` column. The constraint should be named at creation. Name the constraint `my_deptid_pk`.
Hint: The constraint is enabled as soon as the `ALTER TABLE` command executes successfully.

A:

```
ALTER TABLE dept
ADD CONSTRAINT my_deptid_pk PRIMARY KEY(id);
```

Problem 3

Q: Add a column `DEPT_ID` to the `EMP` table. Add a foreign key reference on the `EMP` table that ensures that the employee is not assigned to a nonexistent department. Name the constraint `my_emp_dept_id_fk`.

A:

```
ALTER TABLE emp
ADD (dept_id NUMBER(7));

ALTER TABLE emp
ADD CONSTRAINT my_emp_dept_id_fk
FOREIGN KEY (dept_id) REFERENCES dept(id);
```

Problem 4

Q: Confirm that the constraints were added by querying the `USER_CONSTRAINTS` view. Note the types and names of the constraints. Save your statement text in a file called `lab10_4.sql`.

CONSTRAINT_NAME	C
MY_DEPT_ID_PK	P
SYS_C002541	C
MY_EMP_ID_PK	P
MY_EMP_DEPT_ID_FK	R

A:

`lab10_4.sql`

```
SELECT constraint_name, constraint_type
FROM user_constraints
WHERE table_name IN ('EMP', 'DEPT');
```

CONSTRAINT_NAME	CONSTRAINT_TYPE
SYS_C002248208	C
MY_EMP_ID_PK	P
MY_DEPTID_PK	P
MY_EMP_DEPT_ID_FK	R

Problem 5

Q: Display the object names and types from the `USER_OBJECTS` data dictionary view for the `EMP` and `DEPT` tables. Notice that the new tables and a new index were created.

A:

```
SELECT object_name, object_type
FROM user_objects
WHERE object_name LIKE 'EMP%'
OR object_name LIKE 'DEPT%';
```

OBJECT_NAME	OBJECT_TYPE
DEPT	TABLE
EMP	TABLE

Problem 6

Q: Modify the `EMP` table. Add a `COMMISSION` column of `NUMBER` data type, precision 2, scale 2. Add a constraint to the commission column that ensures that a commission value is greater than zero.

A:

```
ALTER TABLE EMP
ADD commission NUMBER(2,2)
CONSTRAINT my_emp_comm_ck CHECK (commission > 0);
```

Chapter 18

Problem 1

Q: Write a query to display the last name, department number, and salary of any employee whose department number and salary both match the department number and salary of any employee who earns a commission.

LAST_NAME	DEPARTMENT_ID	SALARY
Taylor	80	8600
Zlotkey	80	10500
Abel	80	11000

A:

```
SELECT last_name, department_id, salary
FROM hr.employees
WHERE (salary, department_id) IN
      (SELECT salary, department_id
      FROM hr.employees
      WHERE commission_pct IS NOT NULL);
```

LAST_NAME	DEPARTMENT_ID	SALARY
Singh	80	14000
Partners	80	13500
Errazuriz	80	12000
Cambrault	80	11000

Problem 2

Q: Display the last name, department name, and salary of any employee whose salary and commission match the salary and commission of any employee located in location ID 1700.

LAST_NAME	DEPARTMENT_NAME	SALARY
Whalen	Administration	4400
Gietz	Accounting	8300
Higgins	Accounting	12000
Kochhar	Executive	17000
De Haan	Executive	17000
King	Executive	24000

A:

```
SELECT last_name, department_name, salary
FROM hr.employees e, hr.departments d
WHERE e.department_id = d.department_id
AND (salary, NVL(commission_pct,0)) IN
      (SELECT salary, NVL(commission_pct,0)
      FROM hr.employees e, hr.departments d
      WHERE e.department_id = d.department_id
      AND d.location_id = 1700);
```

LAST_NAME	DEPARTMENT_NAME	SALARY
King	Executive	24000
Yang	Executive	17000
Garcia	Executive	17000
Gruenberg	Finance	12008

Problem 3

Q: Create a query to display the last name, hire date, and salary for all employees who have the same salary and commission as Kochhar.
Note: Do not display Kochhar in the result set.

LAST_NAME	HIRE_DATE	SALARY
De Haan	13-JAN-93	17000

A:

```
SELECT last_name, hire_date, salary
FROM hr.employees
WHERE (salary, NVL(commission_pct,0)) IN
      (SELECT salary, NVL(commission_pct,0)
      FROM hr.employees
      WHERE last_name = 'Taylor')
AND last_name != 'Taylor';
```

LAST_NAME	HIRE_DATE	SALARY
Nayer	7/16/2015, 12:00:00 AM	3200
Stiles	10/26/2015, 12:00:00 AM	3200
McLeod	7/1/2016, 12:00:00 AM	3200

Problem 4

Q: Create a query to display the employees who earn a salary that is higher than the salary of all of the sales managers (`JOB_ID = SA_MAN`). Sort the results on salary from highest to lowest.

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000
Kochhar	AD_VP	17000
De Haan	AD_VP	17000
Hartstein	MK_MAN	13000
Higgins	AC_MGR	12000
Abel	SA_REP	11000

A:

```
SELECT last_name, job_id, salary
FROM hr.employees
WHERE salary > ALL
      (SELECT salary
      FROM hr.employees
      WHERE job_id = 'SA_MAN')
ORDER BY salary DESC;
```

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000
Garcia	AD_VP	17000
Yang	AD_VP	17000

Problem 5

Q: Display the details of the employee ID, last name, and department ID of those employees who live in cities whose name begins with T.

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
201	Hartstein	20
202	Fay	20

A:

```
SELECT employee_id, last_name, department_id
FROM hr.employees
WHERE department_id IN (SELECT department_id
                  FROM hr.departments
                  WHERE location_id IN
                     (SELECT location_id
                     FROM hr.locations
                     WHERE city LIKE 'T%'));
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
201	Martinez	20
202	Davis	20

Problem 6

Q: Write a query to find all employees who earn more than the average salary in their departments. Display last name, salary, department ID, and the average salary for the department. Sort by average salary. Use aliases for the columns retrieved by the query as shown in the sample output.

ENAME	SALARY	DEPTNO	DEPT_AVG
Mourgos	5800	50	3500
Hunold	9000	60	6400
Hartstein	13000	20	9500
Abel	11000	80	10033.3333
Zlotkey	10500	80	10033.3333
Higgins	12000	110	10150
King	24000	90	19333.3333

A:

```
SELECT e.last_name ename, e.salary salary,
e.department_id deptno, ROUND(AVG(a.salary)) dept_avg
FROM hr.employees e, hr.employees a
WHERE e.department_id = a.department_id
AND e.salary > (SELECT ROUND(AVG(salary))
            FROM hr.employees
            WHERE department_id = e.department_id )
GROUP BY e.last_name, e.salary, e.department_id
ORDER BY ROUND(AVG(a.salary));
```

ENAME	SALARY	DEPTNO	DEPT_AVG
Bull	4100	50	3476
Bell	4000	50	3476
Everett	3900	50	3476
Dilly	3600	50	3476
Chung	3800	50	3476

Problem 7

Q: Find all employees who are not supervisors.

a. First do this using the `NOT EXISTS` operator.

b. Can this be done by using the `NOT IN` operator? How, or why not?

LAST_NAME
Ernst
Lorentz
Rajs
Davies
Matos
Vargas
Abel
Taylor
Grant
Whalen
Fay
Gietz

A: a.

```
SELECT e.last_name ename, e.salary salary,
e.department_id deptno, ROUND(AVG(a.salary)) dept_avg
FROM hr.employees e, hr.employees a
WHERE e.department_id = a.department_id
AND e.salary > (SELECT ROUND(AVG(salary))
            FROM hr.employees
            WHERE department_id = e.department_id )
GROUP BY e.last_name, e.salary, e.department_id
ORDER BY ROUND(AVG(a.salary));
```

LAST_NAME
Abel
Ande
Atkinson
Baida
Banda
Bates

b.

```
      SELECT outer.last_name
FROM hr.employees outer
WHERE outer.employee_id
NOT IN (SELECT inner.manager_id
      FROM hr.employees inner);
```

This is not a good solution. The subquery includes a `NULL`, so the whole query gives no results. That's because comparing anything with `NULL` gives `NULL`, not true or false. So, if there's a chance of `NULL` values, don't use `NOT IN` — use `NOT EXISTS` instead.

Problem 8

Q: Write a query to display the last names of the employees who earn less than the average salary in their departments

LAST_NAME
Kochhar
De Haan
Ernst
Lorentz
Davies
Matos
Vargas
Taylor
Fay
Gietz

A:

```
SELECT last_name
FROM hr.employees outer
WHERE outer.salary < (SELECT AVG(inner.salary)
                      FROM hr.employees inner
                      WHERE inner.department_id
                           = outer.department_id);
```

LAST_NAME
Yang
Garcia
Williams
Jackson
Nguyen
Chen

Problem 9

Q: Write a query to display the last names of the employees who have one or more coworkers in their departments with later hire dates but higher salaries.

LAST_NAME
Rajs
Davies
Matos
Vargas
Taylor

A:

```
SELECT last_name
FROM hr.employees outer
WHERE EXISTS (SELECT 'X'
           FROM hr.employees inner
           WHERE inner.department_id =
                outer.department_id
           AND inner.hire_date > outer.hire_date
           AND inner.salary > outer.salary);
```

LAST_NAME
Garcia
Williams
Jackson
Nguyen
Faviet
Sciarra

Problem 10

Q: Write a query to display the employee ID, last names, and department names of all employees. **Note:** Use a scalar subquery to retrieve the department name in the `SELECT` statement.

EMPLOYEE_ID	LAST_NAME	DEPARTMENT
205	Higgins	Accounting
206	Gietz	Accounting
200	Whalen	Administration
100	King	Executive
101	Kochhar	Executive
102	De Haan	Executive
103	Hunold	IT
104	Ernst	IT
107	Lorentz	IT
201	Hartstein	Marketing
202	Fay	Marketing
149	Zlotkey	Sales
176	Taylor	Sales
174	Abel	Sales
EMPLOYEE_ID	LAST_NAME	DEPARTMENT
124	Mourgos	Shipping
141	Rajs	Shipping
142	Davies	Shipping
143	Matos	Shipping
144	Vargas	Shipping
178	Grant	

A:

```
SELECT employee_id, last_name,
      (SELECT department_name
      FROM hr.departments d
      WHERE e.department_id =
          d.department_id ) department
FROM hr.employees e
ORDER BY department;
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT
205	Higgins	Accounting
206	Gietz	Accounting
200	Whalen	Administration
100	King	Executive
101	Yang	Executive
102	Garcia	Executive

Problem 11

Q: Write a query to display the department names of those departments whose total salary cost is above one eighth (1/8) of the total salary cost of the whole company. Use the WITH clause to write this query. Name the query `SUMMARY`.

EMPLOYEE_ID	LAST_NAME	DEPARTMENT
205	Higgins	Accounting
206	Gietz	Accounting
200	Whalen	Administration
100	King	Executive
101	Kochhar	Executive
102	De Haan	Executive
103	Hunold	IT
104	Ernst	IT
107	Lorentz	IT
201	Hartstein	Marketing
202	Fay	Marketing
149	Zlotkey	Sales
176	Taylor	Sales
174	Abel	Sales
EMPLOYEE_ID	LAST_NAME	DEPARTMENT
124	Mourgos	Shipping
141	Rajs	Shipping
142	Davies	Shipping
143	Matos	Shipping
144	Vargas	Shipping
178	Grant	

A:

```
WITH
   summary AS (
   SELECT department_name, SUM(salary) AS dept_total
   FROM hr.employees e, hr.departments d
   WHERE e.department_id =
        d.department_id
   GROUP BY department_name)
   SELECT department_name, dept_total
   FROM summary
   WHERE dept_total > (
                  SELECT SUM(dept_total) * 1/8
                  FROM summary)
   ORDER BY dept_total DESC;
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT
205	Higgins	Accounting
206	Gietz	Accounting
200	Whalen	Administration
100	King	Executive
101	Yang	Executive
102	Garcia	Executive