UNIVERSITY OF CHITTAGONG

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Program: **B.Sc.** (Engineering)
Session: 2022–2023
4th Semester

Lab_task_03

Course Title: DataBase Systems Lab
Course Code: CSE-414

Submitted To

Dr. Rudra Pratap Deb Nath
Associate Professor
Department of Computer Science & Engineering
University of Chittagong

Submitted By

Subha Shesgin
ID: 23701036
Department of Computer Science & Engineering

Date of Submission: 31 May, 2025

Contents

Introduction to Oracle9i: SQL
(SQL Practice Exercises)

CHAPTER 8
Manipulating Data

Practice 8

1. Insert data into the MY_EMPLOYEE table..
   Run the statement in the lab8_1.sql script to build the MY_EMPLOYEE table to be used for the lab.

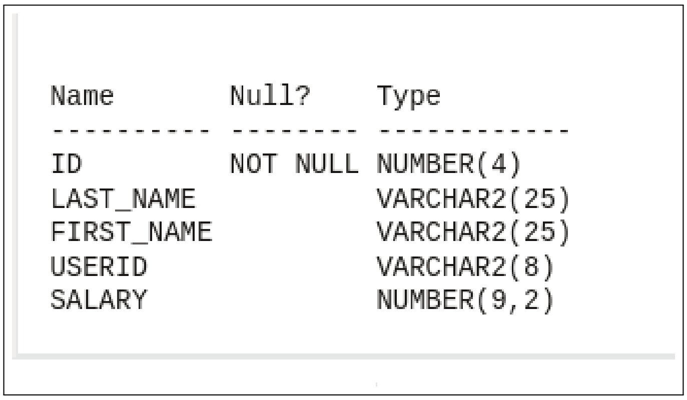
   Answer :
   Query :

```
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)
);
```

2. Describe the structure of the MY_EMPLOYEE table to identify the column names.

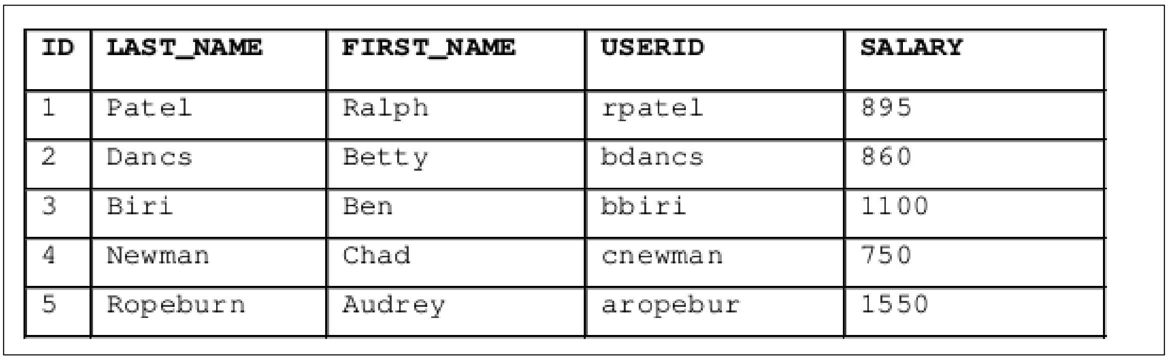
   Answer :
   Query :

```
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)
```

3. 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

Answer :

Query :

```sql
INSERT into MY_EMPLOYEE
VALUES (1 ,'Patel', 'ralph', 'rpatel',895);
```

4. 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.

Answer :

Query :

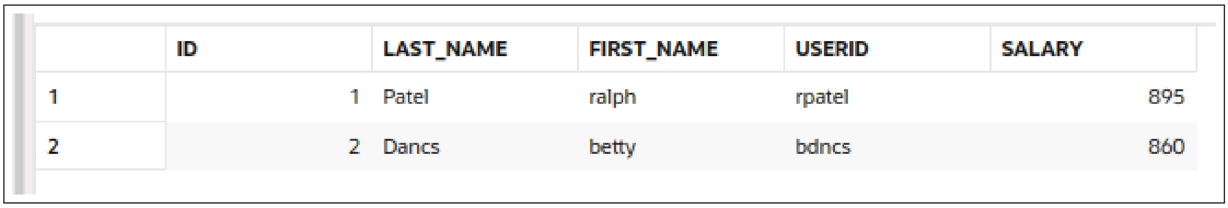
```sql
INSERT into MY_EMPLOYEE(ID,LAST_NAME,FIRST_NAME, USERID,SALARY)
VALUES (2,'Dancs', 'betty', 'bdncs',860);
```

5. Confirm your addition to the table.

Answer :

Query :

```sql
SELECT *
from MY_EMPLOYEE;
```



	ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	1	Patel	ralph	rpatel	895
2	2	Dancs	betty	bdncs	860

6. 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.

   Answer :

   Query :

```
INSERT INTO MY_EMPLOYEE
VALUES (3, ’Biri’, ’Ben’, LOWER(SUBSTR('Ben’, 1, 1) || SUBSTR('
   Biri’, 1, 7)), 1100);
INSERT INTO MY_EMPLOYEE
VALUES (4, ’Newman’, ’Chad’, LOWER(SUBSTR('Chad’, 1, 1) ||
   SUBSTR('Newman’, 1, 7)), 750);
```

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

   Answer :

   Query :

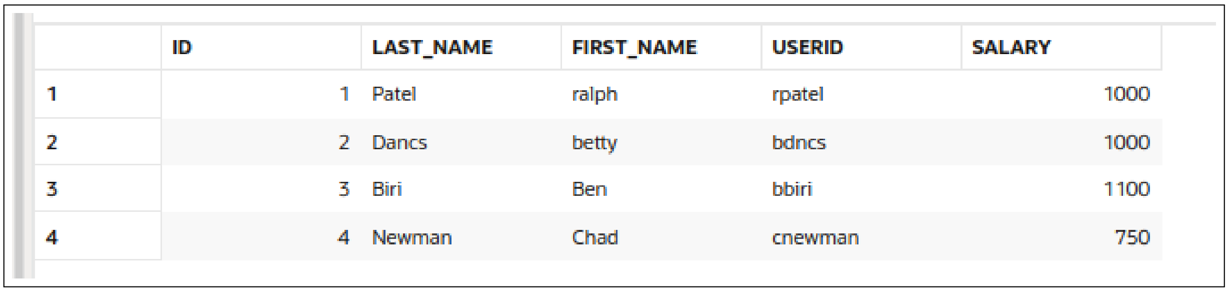
```
INSERT INTO MY_EMPLOYEE
VALUES (3, ’Biri’, ’Ben’, LOWER(SUBSTR('Ben’, 1, 1) || SUBSTR('
   Biri’, 1, 7)), 1100);
INSERT INTO MY_EMPLOYEE
VALUES (4, ’Newman’, ’Chad’, LOWER(SUBSTR('Chad’, 1, 1) ||
   SUBSTR('Newman’, 1, 7)), 750);
```

8. Confirm your additions to the table.

   Answer :

   Query :

```
SELECT *
from MY_EMPLOYEE;
```



	ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	1	Patel	ralph	rpatel	1000
2	2	Dancs	betty	bdncs	1000
3	3	Biri	Ben	bbiri	1100
4	4	Newman	Chad	cnewman	750

9. Make the data additions permanent.

Answer :

Query :

```
COMMIT;
```

Update and delete data in the MY_EMPLOYEE table.

10. Change the last name of employee 3 to Drexler

Answer :

Query :

```
update MY_EMPLOYEE
set LAST_NAME = 'Drexler'
WHERE id = 3;
```

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

Answer :

Query :

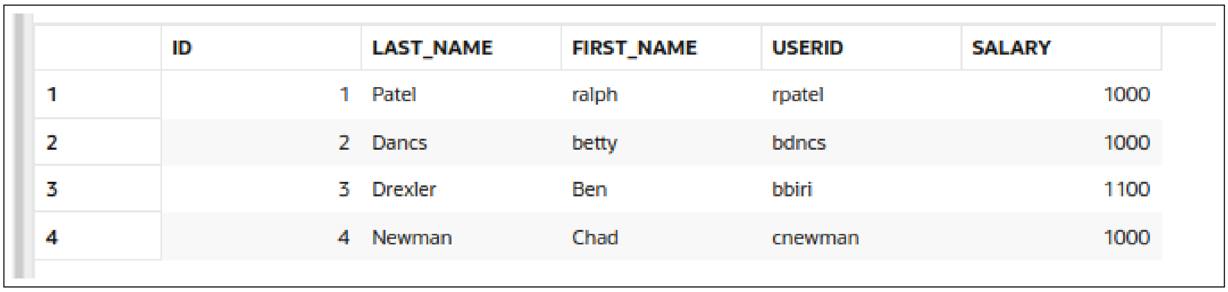
```
update MY_EMPLOYEE
set salary = 1000
WHERE salary < 900;
```

12. Verify your changes to the table.

Answer :

Query :

```
SELECT *
from MY_EMPLOYEE;
```



	ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	1	Patel	ralph	rpatel	1000
2	2	Dancs	betty	bdncs	1000
3	3	Drexler	Ben	bbiri	1100
4	4	Newman	Chad	cnewman	1000

13. Delete Betty Dancs from the MY_EMPLOYEE table.

Answer :

Query :

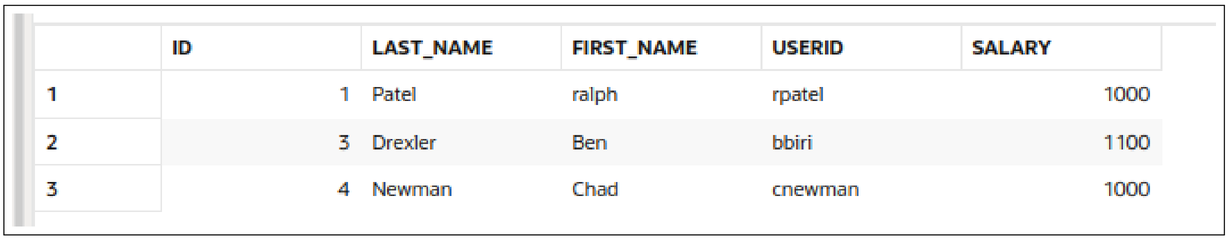
```
DELETE FROM MY_EMPLOYEE
WHERE LAST_NAME = ’Dancs’ AND FIRST_NAME = ’betty’;
```

14. Confirm your changes to the table.

Answer :

Query :

```
SELECT *
from MY_EMPLOYEE;
```



	ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	1	Patel	ralph	rpatel	1000
2	3	Drexler	Ben	bbiri	1100
3	4	Newman	Chad	cnewman	1000

15. Commit all pending changes.

Answer :

Query :

```
commit;
```

Control data transaction to the MY_EMPLOYEE table.

16. 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.

Answer :

Query :

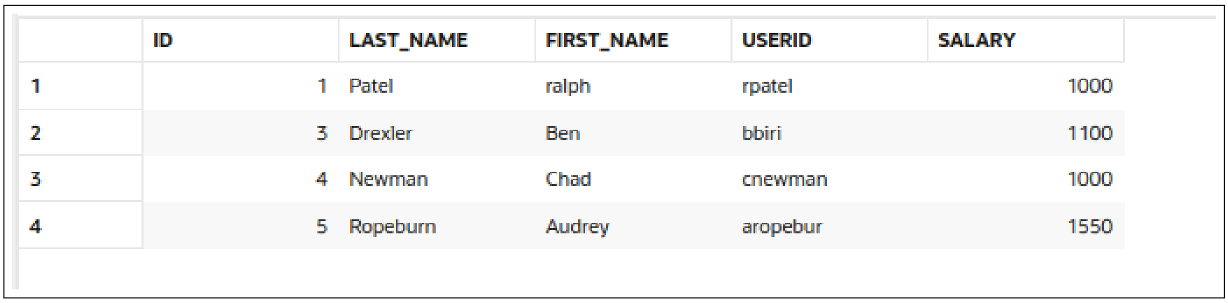
```
INSERT INTO MY_EMPLOYEE
VALUES (5, ’Ropeburn’, ’Audrey’, LOWER(SUBSTR(’Audrey’, 1, 1) ||
    SUBSTR(’Ropeburn’, 1, 7)), 1550);
```

17. Confirm your addition to the table.

Answer :

Query :

```
SELECT *
from MY_EMPLOYEE
order by id asc;
```



	ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	1	Patel	ralph	rpatel	1000
2	3	Drexler	Ben	bbiri	1100
3	4	Newman	Chad	cnewman	1000
4	5	Ropeburn	Audrey	aropebur	1550

18. Mark an intermediate point in the processing of the transaction.

Answer :

Query :

```
SAVEPOINT step_18;
```

19. Empty the entire table.

Answer :

Query :

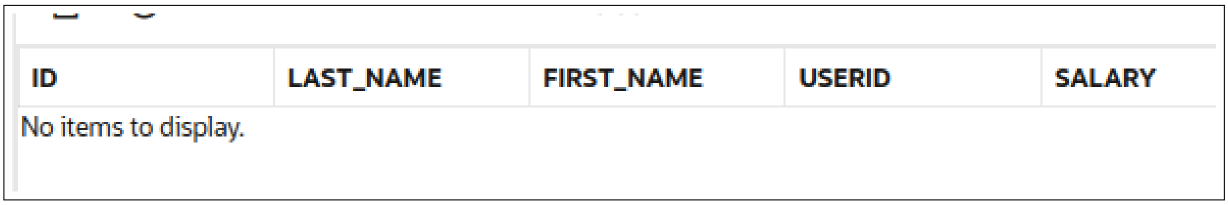
```
delete from MY_EMPLOYEE;
```

20. Confirm your addition to the table.

Answer :

Query :

```
SELECT *
from MY_EMPLOYEE;
```



ID	LAST_NAME	FIRST_NAME	USERID	SALARY

No items to display.

21. Discard the most recent DELETE operation without discarding the earlier INSERT operation.

    Answer :

    Query :

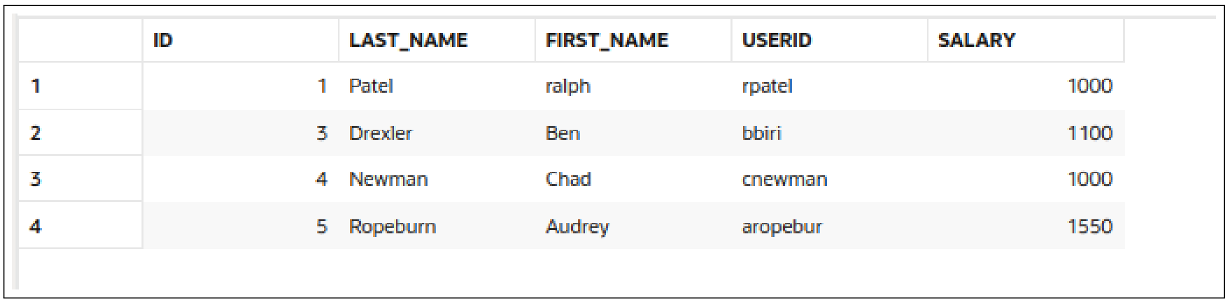
```
ROLLBACK to step_18;
```

22. Confirm that the new row is still intact.

    Answer :

    Query :

```
SELECT *
from MY_EMPLOYEE
order by id asc;
```



	ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	1	Patel	ralph	rpatel	1000
2	3	Drexler	Ben	bbiri	1100
3	4	Newman	Chad	cnewman	1000
4	5	Ropeburn	Audrey	aropebur	1550

23. Make the data addition permanent.

    Answer :

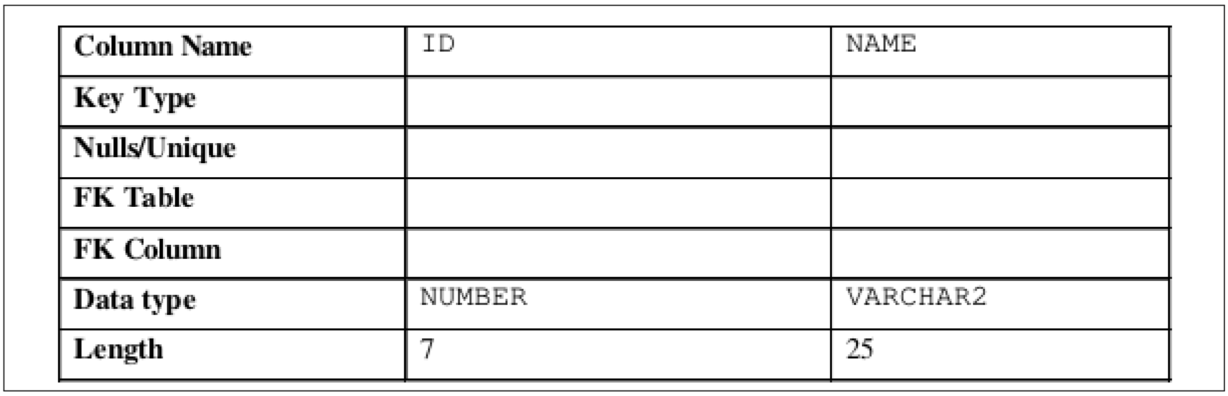
    Query :

```
commit;
```

CHAPTER 9
Creating and Managing Tables

Practice 9

1. 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.

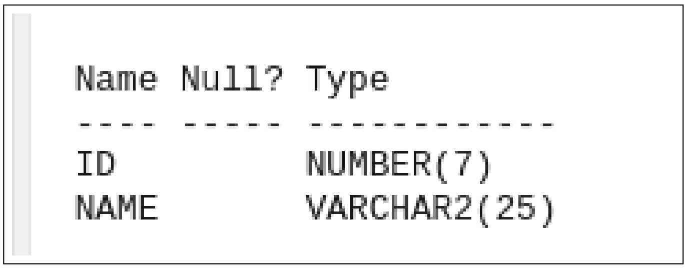


Column Name	ID	NAME
Key Type		
Nulls/Unique		
FK Table		
FK Column		
Data type	NUMBER	VARCHAR2
Length	7	25

Answer :

Query :

```
CREATE TABLE dept (
  id NUMBER(7),
  name VARCHAR2(25)
);

describe dept;
```



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

2. Populate the DEPT table with data from the DEPARTMENTS table. Include only columns that you need.

Answer :

Query :

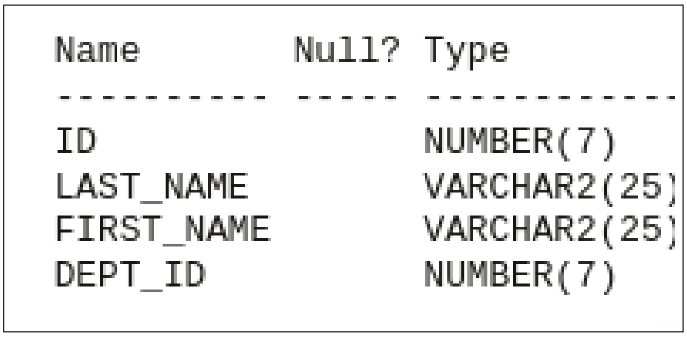
```
insert into dept
select department_id , department_name
from hr.DEPARTMENTS;
```

3. 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.

   Answer :

   Query :

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

describe EMP;
```



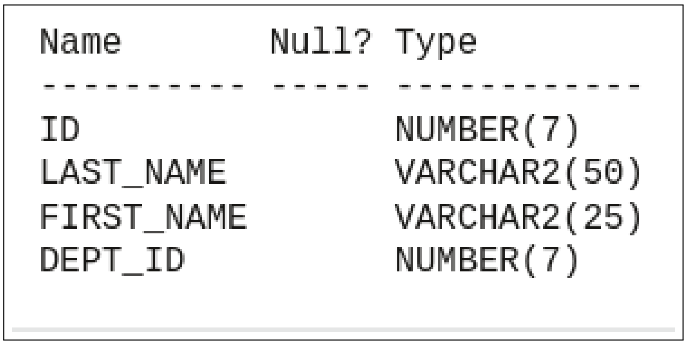
```
Name        Null? Type
----------- ----- ------------
ID                NUMBER(7)
LAST_NAME         VARCHAR2(25)
FIRST_NAME        VARCHAR2(25)
DEPT_ID           NUMBER(7)
```

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

   Answer :

   Query :

```
ALTER table EMP
MODIFY (last_name VARCHAR2(50));

DESCRIBE emp;
```



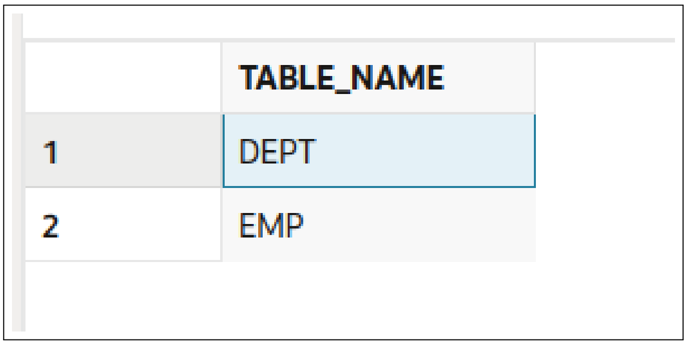
```
Name        Null? Type
----------- ----- ------------
ID                NUMBER(7)
LAST_NAME         VARCHAR2(50)
FIRST_NAME        VARCHAR2(25)
DEPT_ID           NUMBER(7)
```

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

   Answer :

   Query :

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



	TABLE_NAME
1	DEPT
2	EMP

6. 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.

   Answer :

   Query :

```sql
create table EMPLOYEES2 as
select employee_id id, first_name, last_name,salary,
   department_id dept_id
from hr.EMPLOYEES;
```

7. Drop the EMP table.

   Answer :

   Query :

```sql
drop table emp;
```

8. Rename the EMPLOYEES2 table as EMP.

   Answer :

   Query :

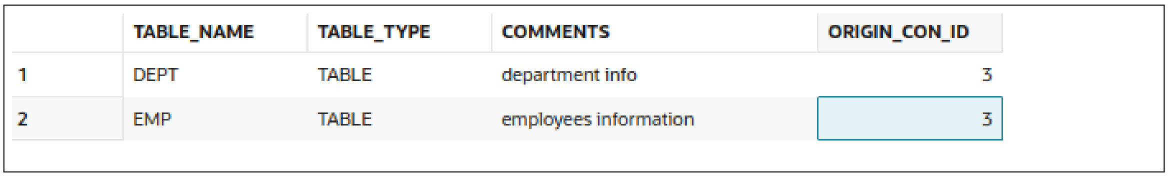
```sql
rename EMPLOYEES2 to EMP;
```

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

   Answer :

   Query :

```
COMMENT ON TABLE emp IS 'employees information';

COMMENT ON TABLE dept IS 'department info';

select *
from user_tab_comments
where table_name ='DEPT' or table_name = 'EMP';
```



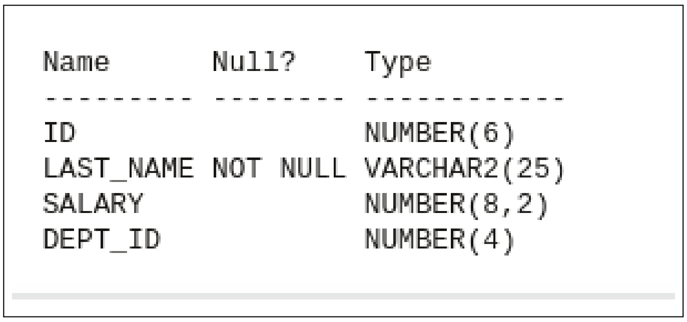
	TABLE_NAME	TABLE_TYPE	COMMENTS	ORIGIN_CON_ID
1	DEPT	TABLE	department info	3
2	EMP	TABLE	employees information	3

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

    Answer :

    Query :

```
alter table emp
drop COLUMN first_name;

DESCRIBE emp;
```



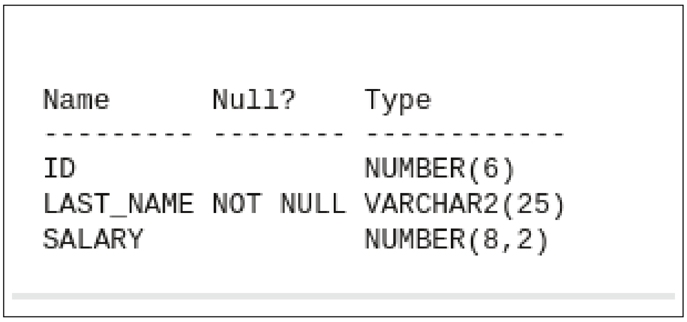
```
Name      Null?    Type
--------- -------- ------------
ID                 NUMBER(6)
LAST_NAME NOT NULL VARCHAR2(25)
SALARY             NUMBER(8,2)
DEPT_ID            NUMBER(4)
```

11. 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.

Answer :

Query :

```
alter table emp
set UNUSED (dept_id);

DESCRIBE emp;
```



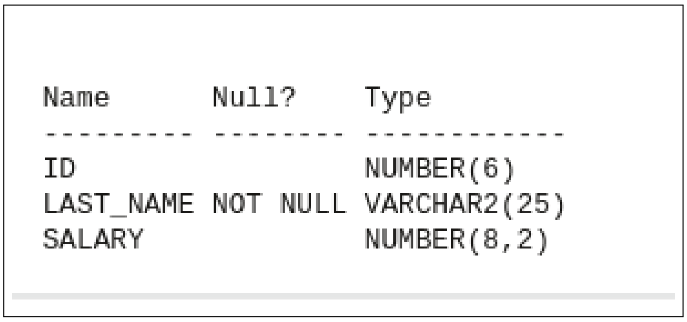
```
Name      Null?    Type
--------- -------- ------------
ID                 NUMBER(6)
LAST_NAME NOT NULL VARCHAR2(25)
SALARY             NUMBER(8,2)
```

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

Answer :

Query :

```
alter table emp
drop UNUSED COLUMNS;

DESCRIBE emp;
```



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

CHAPTER10
Including Constraints

Practice 10

1. 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

   Answer :

   Query :

```sql
alter table EMP
add constraint my_emp_id_pk primary key (id);
```

2. 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.

   Answer :

   Query :

```sql
alter table dept
add constraint my_deptid_pk primary key(id);
```

3. 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..

   Answer :

   Query :

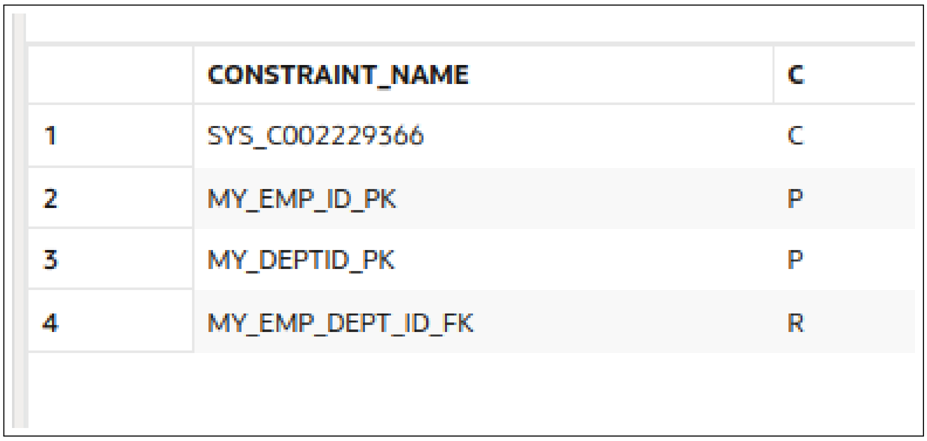
```sql
alter table emp
add (dept_id NUMBER(7));

alter table emp
add constraint my_emp_dept_id_fk
FOREIGN key (dept_id) REFERENCES dept(id);
```

4. 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.

Answer :

Query :

```sql
SELECT constraint_name, constraint_type AS "C"
FROM user_constraints
WHERE table_name IN (’EMP’, ’DEPT’);
```



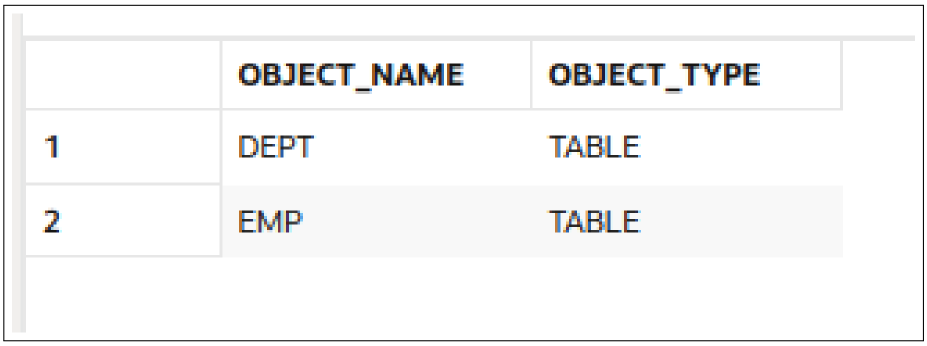
	CONSTRAINT_NAME	C
1	SYS_C002229366	C
2	MY_EMP_ID_PK	P
3	MY_DEPTID_PK	P
4	MY_EMP_DEPT_ID_FK	R

5. 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.

Answer :

Query :

```sql
select object_name,object_type
from USER_OBJECTS
where OBJECT_NAME like ’EMP%’
or OBJECT_NAME like ’DEPT%’;
```



	OBJECT_NAME	OBJECT_TYPE
1	DEPT	TABLE
2	EMP	TABLE

6. 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.

   Answer :

   Query :

```sql
alter table EMP
add commission NUMBER(2,2)
CONSTRAINT my_emp_com_ck check (commission >= 0);
```

CHAPTER18
Advanced Subqueries

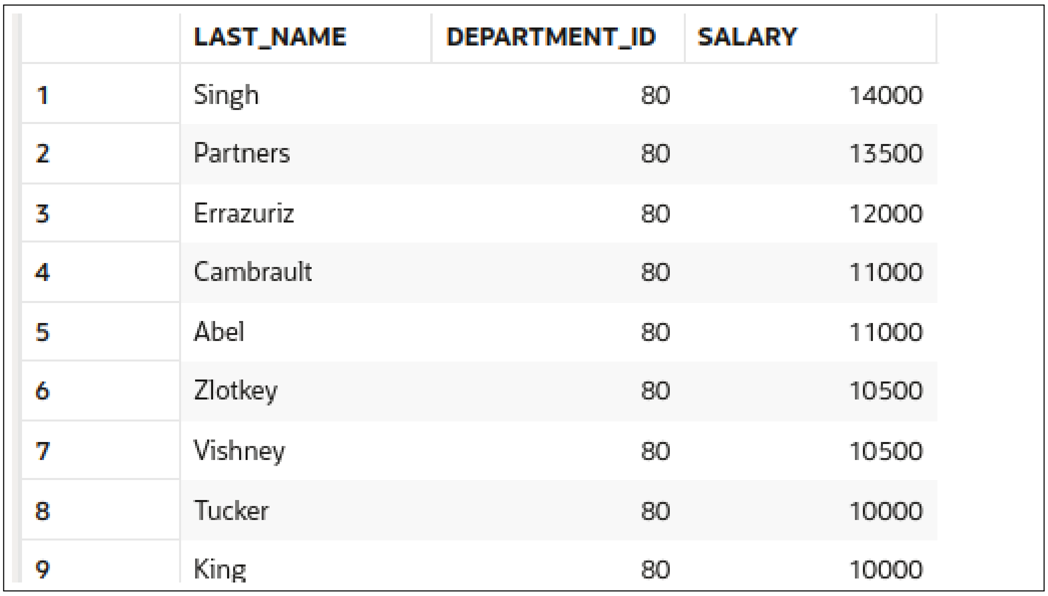
Practice 18

1. 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.

   Answer :

   Query :

```
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);
```



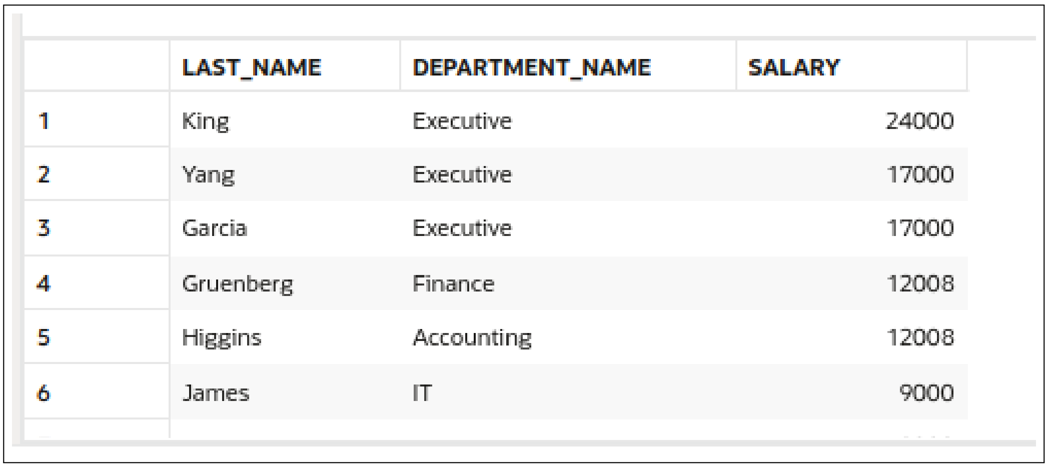
	LAST_NAME	DEPARTMENT_ID	SALARY
1	Singh	80	14000
2	Partners	80	13500
3	Errazuriz	80	12000
4	Cambrault	80	11000
5	Abel	80	11000
6	Zlotkey	80	10500
7	Vishney	80	10500
8	Tucker	80	10000
9	King	80	10000

2. 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.

Answer :

Query :

```
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);
```



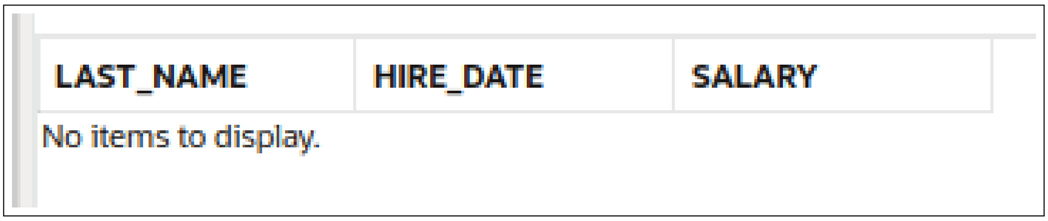
	LAST_NAME	DEPARTMENT_NAME	SALARY
1	King	Executive	24000
2	Yang	Executive	17000
3	Garcia	Executive	17000
4	Gruenberg	Finance	12008
5	Higgins	Accounting	12008
6	James	IT	9000

3. 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.

Answer :

Query :

```
 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 = ’Kochhar’)
and last_name != ’kochhar’;
```



LAST_NAME	HIRE_DATE	SALARY

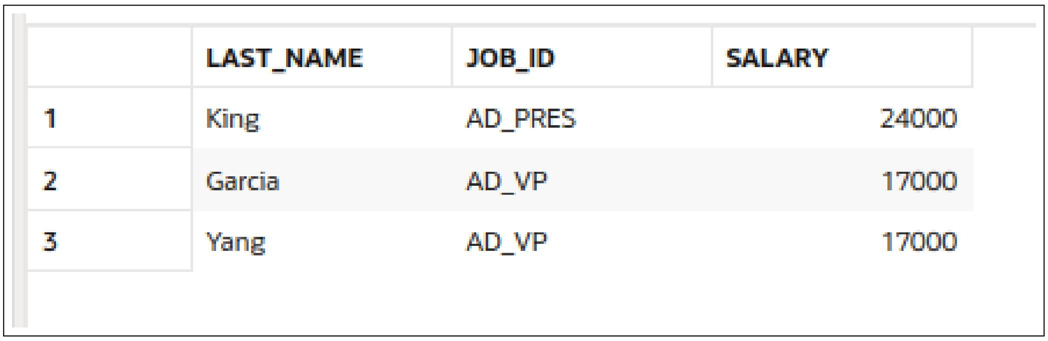
No items to display.

4. 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.

Answer :

Query :

```
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;
```



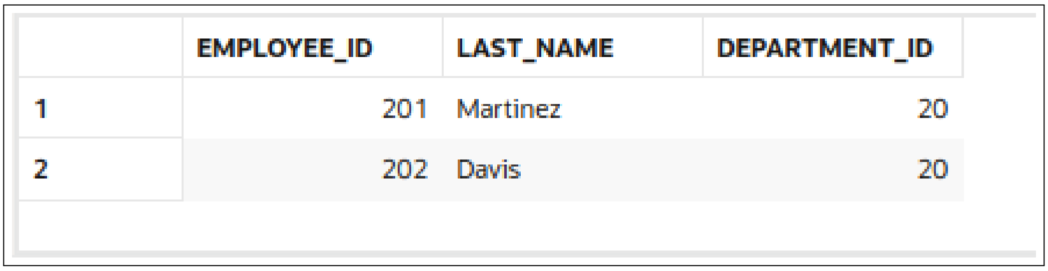
	LAST_NAME	JOB_ID	SALARY
1	King	AD_PRES	24000
2	Garcia	AD_VP	17000
3	Yang	AD_VP	17000

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

Answer :

Query :

```
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%’));
```



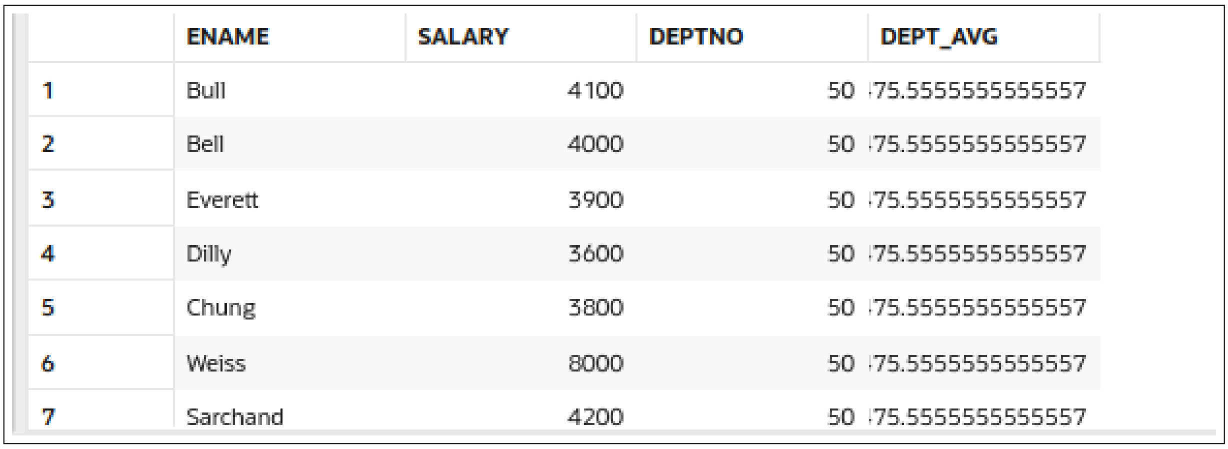
	EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
1	201	Martinez	20
2	202	Davis	20

6. 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.

Answer :

Query :

```sql

select e.last_name ename , e.salary salary ,
       e.department_id deptno , avg(a.salary) dept_avg
from hr.employees e , hr.employees a
where e.department_id = a.department_id
and e.salary > (select avg(salary)
            from hr.employees
            where department_id = e.department_id)
group by e.last_name , e.salary , e.department_id
order by avg(a.salary);
```



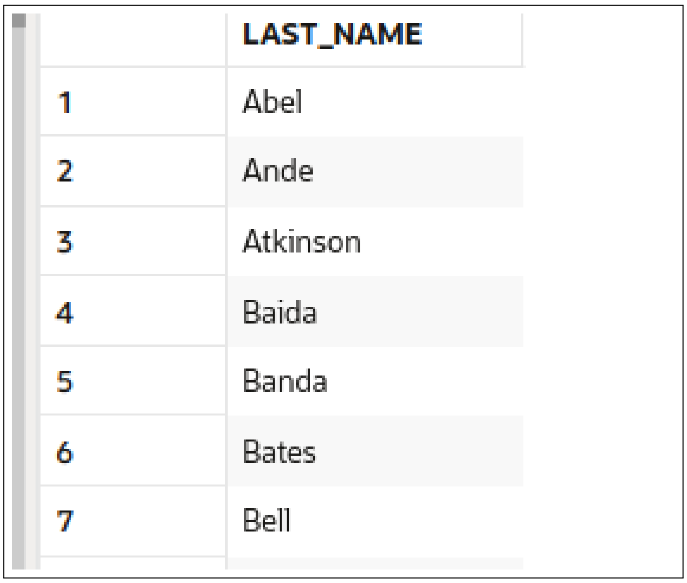
	ENAME	SALARY	DEPTNO	DEPT_AVG
1	Bull	4100	50	75.5555555555557
2	Bell	4000	50	75.5555555555557
3	Everett	3900	50	75.5555555555557
4	Dilly	3600	50	75.5555555555557
5	Chung	3800	50	75.5555555555557
6	Weiss	8000	50	75.5555555555557
7	Sarchand	4200	50	75.5555555555557

7. Find all employees who are not supervisors.
   a. First do this using the NOT EXISTS operator.

Answer :

Query :

```sql

SELECT e.last_name
FROM hr.employees e
WHERE NOT EXISTS (
  SELECT 'X'
  FROM hr.employees m
  WHERE m.manager_id = e.employee_id
);
```



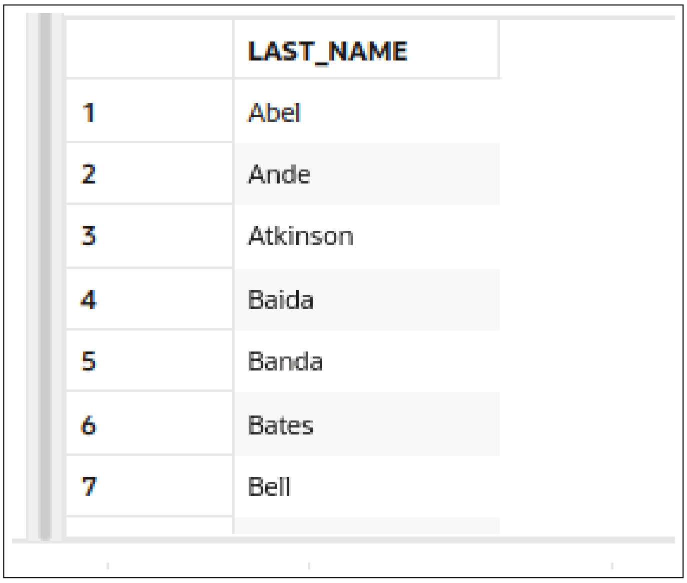
	LAST_NAME
1	Abel
2	Ande
3	Atkinson
4	Baida
5	Banda
6	Bates
7	Bell

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

Answer :

Query :

```sql

SELECT e.last_name
FROM hr.employees e
WHERE e.employee_id NOT IN (
  SELECT a.manager_id
  FROM hr.employees a
  WHERE a.manager_id IS NOT NULL
);
```



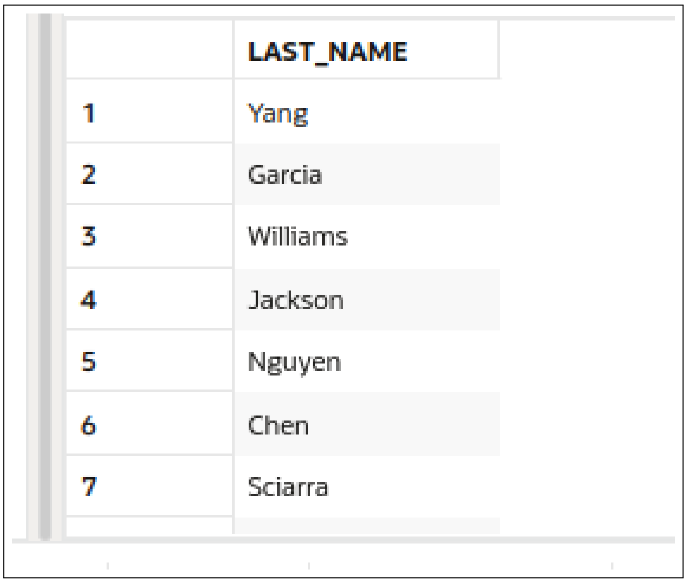
	LAST_NAME
1	Abel
2	Ande
3	Atkinson
4	Baida
5	Banda
6	Bates
7	Bell

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

Answer :

Query :

```
select last_name
from hr.employees e
where e.salary <(select avg(a.salary)
            from hr.employees a
            where a.DEPARTMENT_ID = e.DEPARTMENT_ID );
```



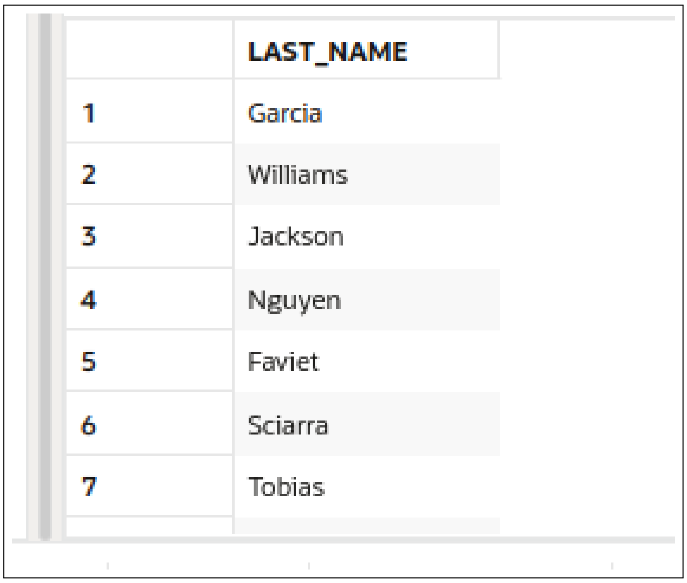
	LAST_NAME
1	Yang
2	Garcia
3	Williams
4	Jackson
5	Nguyen
6	Chen
7	Sciarra

9. 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.

   Answer :

   Query :

```
select last_name
from hr.employees e
where exists (select ’X’
            from hr.employees a
            where a.DEPARTMENT_ID = e.DEPARTMENT_ID
            and a.hire_date > e.HIRE_DATE
            and a.SALARY > e.SALARY );
```



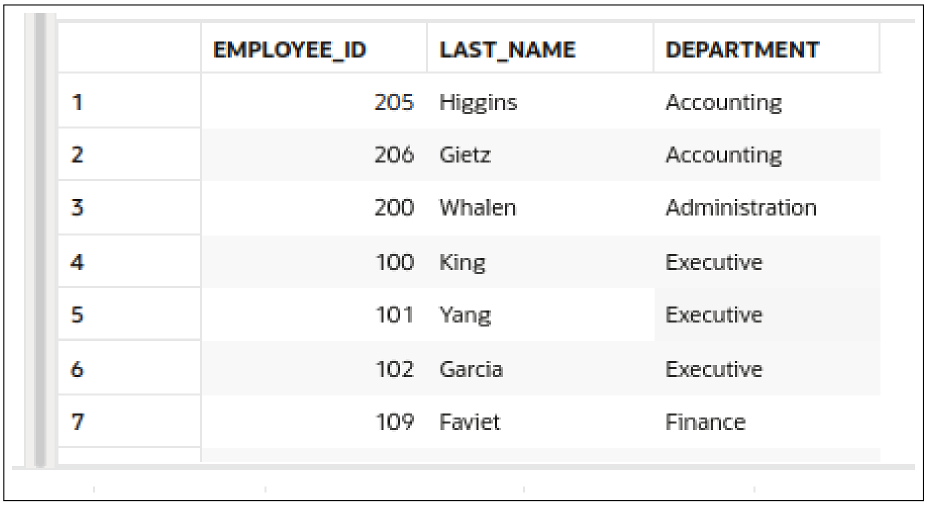
	LAST_NAME
1	Garcia
2	Williams
3	Jackson
4	Nguyen
5	Faviet
6	Sciarra
7	Tobias

10. 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..

    Answer :

    Query :

```
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;
```



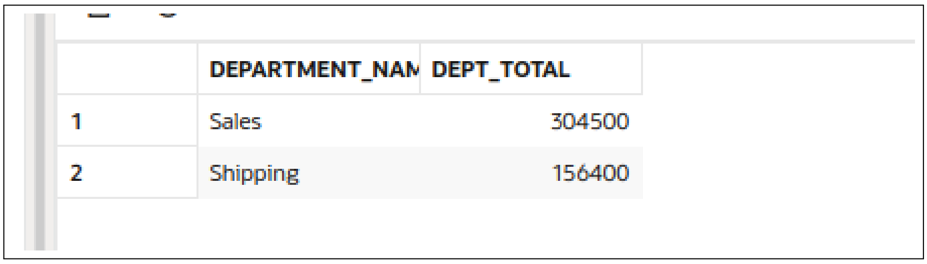
	EMPLOYEE_ID	LAST_NAME	DEPARTMENT
1	205	Higgins	Accounting
2	206	Gietz	Accounting
3	200	Whalen	Administration
4	100	King	Executive
5	101	Yang	Executive
6	102	Garcia	Executive
7	109	Faviet	Finance

11. 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.

Answer :

Query :

```
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;
```



	DEPARTMENT_NAM	DEPT_TOTAL
1	Sales	304500
2	Shipping	156400