



UNIVERSITY OF CHITTAGONG

Department of Computer Science & Engineering

Program: **B.Sc. (Engineering)**

Session: 2022-2023

4th Semester

Lab_Task_03

Topic: SQL Practice Exercises

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:

Nilanjana Das Jui

ID: 23701011

Dept. of Computer Science & Engineering

Date of submission: June 14, 2025

Contents

• Chapter - 8: Manipulating Data	2
– Practice - 8	2
• Chapter - 9: Creating and Managing Tables	9
– Practice - 9	9
• Chapter-10: Including Constraints	14
– Practice - 10	14
• Chapter-18: Advanced Subqueries	17
– Practice - 18	17

Chapter - 8

Manipulating Data

Practice - 8

1. Run the statement in the lab8_1.sql script to build the MY_EMPLOYEE table that will be used for the lab.

Answer:

```
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.

```
SQL> 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)
```

Answer:

```
DESCRIBE my_employee;
```

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:



```
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:

```
INSERT INTO MY_EMPLOYEE (ID, LAST_NAME, FIRST_NAME,
USERID, SALARY)
VALUES (2, 'Dancs', 'Betty', 'bdancs', 860);
```

5. Confirm your addition to the table.

		Download ▾	Execution time: 0.005 seconds		
	ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	1	Patel	Ralph	rpatel	895
2	2	Dancs	Betty	bdancs	860

Answer:

```
SELECT *
FROM MY_EMPLOYEE;
```

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:

```
SET ECHO OFF
SET VERIFY OFF
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);
SET VERIFY ON
SET ECHO ON
```

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

Answer:

In 'loademp.sql' file,

```
INSERT INTO my_employee
VALUES (3, 'Biri', 'Ben', 'bbiri', 1100);

INSERT INTO my_employee
VALUES (4, 'Newman', 'Chad', 'cnewman', 750);
```

In 'Lab8_7.sql' file,

```
SET ECHO OFF
SET VERIFY OFF
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);
SET VERIFY ON
SET ECHO ON
```

8. Confirm your additions to the table.

Download ▾ Execution time: 0.005 seconds					
	ID	LAST_NAME	FIRST_NAME	USERID	SALARY
1	1	Patel	Ralph	rpatel	895
2	2	Dancs	Betty	bdancs	860
3	3	Biri	Ben	bbiri	1100
4	4	Newman	Chad	cnewman	750

Answer:

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

9. Make the data additions permanent.

Answer:

```
COMMIT
```

Update and delete data in the MY_EMPLOYEE table.

10. Change the last name of employee 3 to Drexler.


```
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:

```
UPDATE MY_EMPLOYEE  
SET SALARY = 1000  
WHERE SALARY < 900;
```

12. Verify your changes to the table.





Download

▼

Execution time: 0.001 seconds

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

Answer:

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

13. Delete Betty Dancs from the MY_EMPLOYEE table.

Answer:

```
DELETE FROM MY_EMPLOYEE
WHERE LAST_NAME = 'Dancs';
```

14. Confirm your changes to the table.

🗑️ ⓘ Download ▾	Execution time: 0.001 seconds				
	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

Answer:

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

15. Commit all pending changes.

Answer:

```
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:



In 'loademp.sql' script,

```
INSERT INTO my_employee
VALUES (5, 'Ropeburn', 'Adrey', 'aropebur', 1500);
```

In 'Lab8_16.sql' file,

```
SET ECHO OFF
SET VERIFY OFF
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);
SET VERIFY ON
SET ECHO ON
```

17. Confirm your addition to the table.

		Download ▾	Execution time: 0.001 seconds		
	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	Adrey	aropebur	1550

Answer:

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

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

Answer:

```
SAVEPOINT STEP_18;
```

19. Empty the entire table.

Answer:

```
DELETE FROM MY_EMPLOYEE;
```


20. Confirm that the table is empty.

Answer:

```
SELECT *
FROM MY_EMPLOYEE;
```





Download 

Execution time: 0.001 seconds


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:

```
ROLLBACK TO STEP_18;
```

22. Confirm that the new row is still intact.





Download 

Execution time: 0.001 seconds

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

Answer:

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

23. Make the data addition permanent.

Answer:

```
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

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

Answer:

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

```
DESCRIBE dept;
```

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

Answer:

```
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.

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)

```

Answer:

```

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

```

```

DESCRIBE emp;

```

4. 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)

```

Answer:

```

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



```

```

DESCRIBE emp;

```

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

		Download ▼	Execution time: 0.043 seconds
	TABLE_NAME		
1	DEPT		
2	EMP		

Answer:

```
SELECT table_name
FROM user_tables
WHERE table_name IN ('DEPT', '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:

```
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:

```
DROP TABLE emp;
```

8. Rename the EMPLOYEES2 table as EMP.

Answer:

```
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:

```
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';
```

Download ▾ Execution time: 0.006 seconds				
	TABLE_NAME	TABLE_TYPE	COMMENTS	ORIGIN_CON_ID
1	DEPT	TABLE	Department Informa	3
2	EMP	TABLE	Employee Informatio	3

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

Answer:

```
ALTER TABLE emp
DROP COLUMN FIRST_NAME;

DESCRIBE emp;
```

```

Name          Null?     Type
-----
ID            NOT NULL  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:

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

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:

```
ALTER TABLE emp
DROP UNUSED COLUMNS;
```

```
DESCRIBE emp;
```

SQL> DESCRIBE emp		
Name	Null?	Type

ID		NUMBER(6)
LAST_NAME	NOT NULL	VARCHAR2(25)
SALARY		NUMBER(8,2)

Chapter - 10

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:

```
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:

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

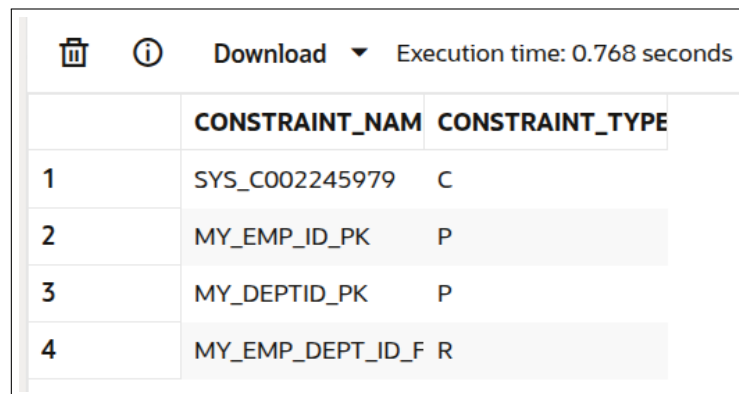
3. Add a column DEPT_ID to the EMP table. Add a foreign key reference on the EMP table that ensures that the employee is not assigned to a nonexistent department. Name the constraint my_emp_dept_id_fk.

Answer:

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

ALTER TABLE textcolorblackemp
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.



The screenshot shows a database query result with the following table structure and data:

	CONSTRAINT_NAME	CONSTRAINT_TYPE
1	SYS_C002245979	C
2	MY_EMP_ID_PK	P
3	MY_DEPTID_PK	P
4	MY_EMP_DEPT_ID_F	R

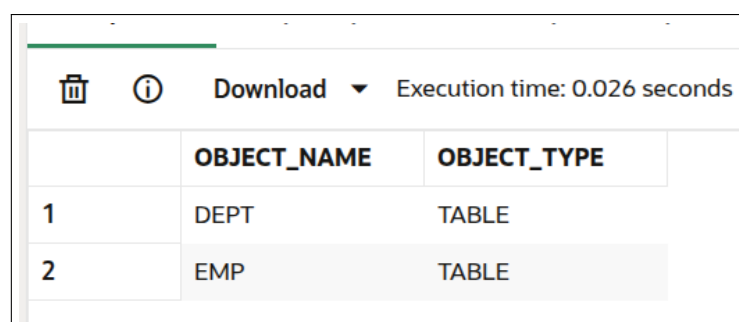
Answer:

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

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:

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



The screenshot shows a database query result with the following table structure and data:

	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:

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

Chapter - 18

Advanced Subqueries

Practice - 18

- 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
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
10	Bloom	80	10000

Answer:

```
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);
```

- 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
1	King	Executive	24000
2	Yang	Executive	17000
3	Garcia	Executive	17000
4	Gruenberg	Finance	12008
5	Higgins	Accounting	12008
6	James	IT	9000
7	Faviet	Finance	9000
8	Fripp	Shipping	8200
9	Chen	Finance	8200
10	Sciarra	Finance	7700

Answer:

```
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);
```

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.

Download ▾ Execution time: 0.005 seconds			
	LAST_NAME	HIRE_DATE	SALARY
1	Yang	21-SEP-15	17000
2	Garcia	13-JAN-11	17000

Answer:

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

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.

Download ▾ Execution time: 0.005 seconds			
	LAST_NAME	JOB_ID	SALARY
1	King	AD_PRES	24000
2	Garcia	AD_VP	17000
3	Yang	AD_VP	17000

Answer:

```
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;
```

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.

Download ▾ Execution time: 0.006 seconds			
	EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
1	201	Martinez	20
2	202	Davis	20

Answer:

```
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%'));
```

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.

Download ▾ Execution time: 0.021 seconds				
	ENAME	SALARY	DEPTNO	DEPT_AVG
1	Bull	4100	50	3475.5556
2	Bell	4000	50	3475.5556
3	Everett	3900	50	3475.5556
4	Dilly	3600	50	3475.5556
5	Chung	3800	50	3475.5556
6	Weiss	8000	50	3475.5556
7	Sarchand	4200	50	3475.5556
8	Fripp	8200	50	3475.5556

Answer:

```
SELECT e.last_name ename, e.salary salary,
       e.department_id deptno,
       ROUND(AVG(a.salary), 4) 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);
```

7. Find all employees who are not supervisors.

a. First do this using the NOT EXISTS operator.

Download ▾ Execution time: 0.007 seconds	
	LAST_NAME
1	Abel
2	Ande
3	Atkinson
4	Baida
5	Banda
6	Bates
7	Bell
8	Bernstein
9	Bissot
10	Bloom

Answer:

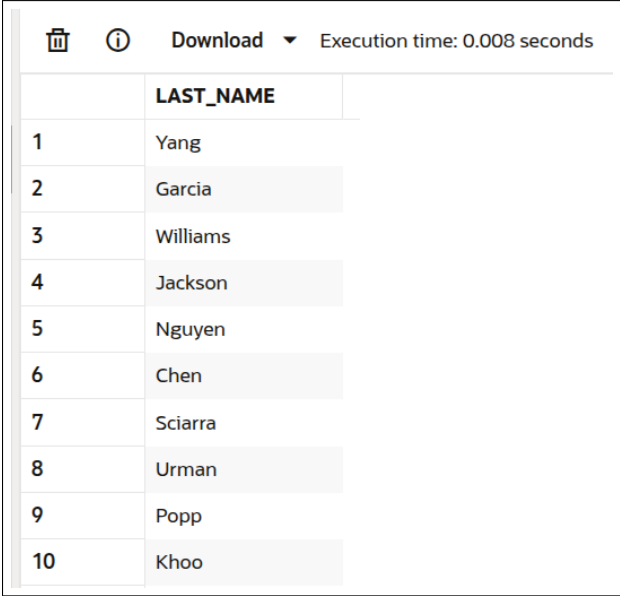
```
SELECT e1.last_name
FROM hr.employees e1
WHERE NOT EXISTS (SELECT 'X'
                  FROM hr.employees e2
                  WHERE e2.manager_id = e1.employee_id);
```

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

Answer:

Using **NOT IN** here isn't a good idea if there's a chance the subquery includes **NULL** values. That's because when SQL tries to compare something with **NULL**, the result is always unknown, and the condition won't work as expected. So even if most values match, just one **NULL** in the list can cause the entire query to return nothing. That's why, as students, we're usually told to avoid **NOT IN** in these cases and stick to **NOT EXISTS**, which handles **NULLs** more reliably.

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



The screenshot shows a database query result with a table containing 10 rows. The table has two columns: an index from 1 to 10 and a column labeled 'LAST_NAME'. The last names listed are Yang, Garcia, Williams, Jackson, Nguyen, Chen, Sciarra, Urman, Popp, and Khoo. Above the table, there are icons for a trash can and information, a 'Download' button, and the text 'Execution time: 0.008 seconds'.

	LAST_NAME
1	Yang
2	Garcia
3	Williams
4	Jackson
5	Nguyen
6	Chen
7	Sciarra
8	Urman
9	Popp
10	Khoo

Answer:

```
SELECT last_name
FROM hr.employees e1
WHERE e1.salary < (SELECT ROUND(AVG(e2.salary), 4)
                  FROM hr.employees e2
                  WHERE e2.department_id = e1.department_id);
```

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.

	LAST_NAME
1	Bates
2	Hutton
3	Taylor
4	Livingston
5	Tuvault
6	Marvins
7	Johnson

Answer:

```
SELECT last_name
FROM hr.employees e1
WHERE EXISTS (SELECT 'X'
              FROM hr.employees e2
              WHERE e2.department_id = e1.department_id
              AND e2.hire_date > e1.hire_date
              AND e2.salary > e1.salary);
```

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.



	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
8	108	Gruenberg	Finance
9	112	Urman	Finance
10	111	Sciarra	Finance

	EMPLOYEE_ID	LAST_NAME	DEPARTMENT
11	110	Chen	Finance
12	113	Popp	Finance
13	203	Jacobs	Human Resources
14	107	Nguyen	IT
15	106	Jackson	IT
16	105	Williams	IT
17	103	James	IT
18	104	Miller	IT
19	201	Martinez	Marketing
20	202	Davis	Marketing

Answer:

```
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;
```

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.

  Download ▾ Execution time: 0.002 seconds		
	DEPARTMENT_NAME	DEPT_TOTAL
1	Sales	304500
2	Shipping	156400

Answer:

```
WITH summary AS (
    SELECT department_name, SUM(salary) AS dept_total
    FROM hr.employees, hr.departments
    WHERE employees.department_id =
           departments.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;
```