

# HR SCHEMA

**USE HR;**

**SHOW TABLES;**

**SELECT \* FROM EMP\_DETAILS\_VIEW;**

**1)- /\*\*WRITE A QUERY TO DISPLAY THE NAMES (FIRST\_NAME, LAST\_NAME) USING ALIAS NAME "FIRST NAME", "LAST NAME" /\*\***

**SELECT**

**CONCAT(FIRST\_NAME,' ',LAST\_NAME) AS NAME FROM EMP\_DETAILS\_VIEW;**

The screenshot shows the SQL Developer interface with a query window and a results grid. The query window contains the following SQL code:

```
1 • USE HR;
2 • SHOW TABLES;
3 • SELECT * FROM emp_details_view;
4 • SELECT
5   CONCAT(first_name,' ',last_name) as NAME from emp_details_view;
6
7
8
```

The results grid displays the output of the query, showing a single column named 'NAME' with 8 rows of data:

| NAME              |
|-------------------|
| Alexander Hunsold |
| Bruce Ernst       |
| David Austin      |
| Valli Pataballa   |
| Diana Lorentz     |
|                   |
|                   |
|                   |

The bottom section of the interface shows the 'Action Output' window, which displays the execution details of the SQL statements:

| # | Time     | Action  | Message             | Duration / Fetch      |
|---|----------|---|---------------------|-----------------------|
| 1 | 21:12:52 | USE HR  | 0 row(s) affected   | 0.000 sec             |
| 2 | 21:12:55 | SHOW TABLES   | 8 row(s) returned   | 0.094 sec / 0.000 sec |
| 3 | 21:13:14 | SELECT CONCAT(first_name,' ',last_name) as NAME from emp_details_view LIMIT 0, 1000 | 106 row(s) returned | 0.062 sec / 0.000 sec |

## 2) WRITE A QUERY TO GET UNIQUE DEPARTMENT ID FROM EMPLOYEE TABLE

**SELECT DISTINCT DEPARTMENT\_ID FROM EMP\_DETAILS\_VIEW;**

The screenshot displays the SQL Developer environment. The left sidebar shows the 'SCHEMAS' tree with 'hr' selected. The main editor window shows the following SQL query:

```
3 • SELECT * FROM emp_details_view;  
4 • SELECT  
5   CONCAT(first_name, ' ', last_name) as NAME from emp_details_view;  
6 • SELECT DISTINCT department_id FROM emp_details_view;  
7  
8  
9  
10
```

The 'Result Grid' shows the output of the third query, displaying a single column 'department\_id' with the following values:

| department_id |
|---------------|
| 60            |
| 50            |
| 10            |
| 30            |
| 90            |

The 'Output' window at the bottom shows the execution log:

| #   | Time     | Action  | Message             | Duration / Fetch      |
|-----|----------|---|---------------------|-----------------------|
| ✓ 1 | 21:12:52 | USE HR  | 0 row(s) affected   | 0.000 sec             |
| ✓ 2 | 21:12:55 | SHOW TABLES   | 8 row(s) returned   | 0.094 sec / 0.000 sec |
| ✓ 3 | 21:13:14 | SELECT CONCAT(first_name, ' ', last_name) as NAME from emp_details_view LIMIT 0, 1000 | 106 row(s) returned | 0.062 sec / 0.000 sec |
| ✓ 4 | 21:21:49 | SELECT DISTINCT department_id FROM emp_details_view LIMIT 0, 1000                     | 11 row(s) returned  | 0.015 sec / 0.000 sec |

### 3) WRITE A QUERY TO GET ALL EMPLOYEE DETAILS FROM THE EMPLOYEE TABLE ORDER BY FIRST NAME, DESCENDING

**SELECT \* FROM EMPLOYEES ORDER BY FIRST\_NAME DESC;**

The screenshot displays the SQL Developer interface. The left sidebar shows the 'SCHEMAS' pane with a tree view containing 'hr', 'jimbh', and 'newschema'. The 'hr' schema is selected, showing its tables, views, stored procedures, and functions. The main window shows a SQL script with the following queries:

```
5 CONCAT(first_name, ' ', last_name) as NAME from emp_details_view;
6 SELECT DISTINCT department_id FROM emp_details_view;
7 SELECT * FROM employees;
8 SELECT * FROM employees ORDER BY first_name DESC;
9
10
11
12 SELECT first_name, last_name, salary, salary*.15 PF
```

The 'Result Grid' shows the results of the query 'SELECT \* FROM employees ORDER BY first\_name DESC;'. The results are as follows:

| employee_id | first_name | last_name | email    | phone_number       | hire_date  | job_id     | salary  | commission_pct | manager_id | department_id |
|-------------|------------|-----------|----------|--------------------|------------|------------|---------|----------------|------------|---------------|
| 180         | Winston    | Taylor    | WTAYLOR  | 650.507.9876       | 1998-01-24 | SH_CLERK   | 3200.00 | NULL           | 120        | 50            |
| 171         | William    | Smith     | WSMITH   | 011.44.1343.629268 | 1999-02-23 | SA_REP     | 7400.00 | 0.15           | 148        | 80            |
| 206         | William    | Gietz     | WGIEZT   | 51hr5.123.8181     | 1994-06-07 | AC_ACCOUNT | 8300.00 | NULL           | 205        | 110           |
| 195         | Vance      | Jones     | VJONES   | 650.501.4876       | 1999-03-17 | SH_CLERK   | 2800.00 | NULL           | 123        | 50            |
| 106         | Valli      | Pataballa | VPATABAL | 590.423.4560       | 1998-02-05 | IT_PROG    | 4800.00 | NULL           | 103        | 60            |

The 'Output' pane shows the execution log with the following entries:

| # | Time     | Action  | Message             | Duration / Fetch      |
|---|----------|---|---------------------|-----------------------|
| 1 | 21:12:52 | USE HR  | 0 row(s) affected   | 0.000 sec             |
| 2 | 21:12:55 | SHOW TABLES   | 8 row(s) returned   | 0.094 sec / 0.000 sec |
| 3 | 21:13:14 | SELECT CONCAT(first_name, ' ', last_name) as NAME from emp_details_view LIMIT 0, 1000 | 106 row(s) returned | 0.062 sec / 0.000 sec |
| 4 | 21:21:49 | SELECT DISTINCT department_id FROM emp_details_view LIMIT 0, 1000                     | 11 row(s) returned  | 0.015 sec / 0.000 sec |
| 5 | 21:23:50 | SELECT * FROM employees ORDER BY first_name DESC LIMIT 0, 1000                        | 107 row(s) returned | 0.016 sec / 0.015 sec |

**4)WRITE A QUERY TO GET THE NAMES (FIRST\_NAME, LAST\_NAME), SALARY, PF OF ALL THE EMPLOYEES (PF IS CALCULATED AS 15% OF SALARY)**

**SELECT FIRST\_NAME, LAST\_NAME, SALARY, SALARY\*.15 PF**  
**FROM EMPLOYEES;**

The screenshot displays the SQL Developer interface. On the left, the 'SCHEMAS' pane shows the 'hr' schema selected. The main window, titled 'SQL File 1', contains the following SQL query:

```
5  CONCAT(first_name, ',last_name) as NAME from emp_details_view;
6  SELECT DISTINCT department_id FROM emp_details_view;
7  SELECT * FROM employees;
8  SELECT * FROM employees ORDER BY first_name DESC;
9  SELECT first_name, last_name, salary, salary*.15 PF
```

Below the query editor, the 'Result Grid' shows the results of the query. The columns are 'first\_name', 'last\_name', 'salary', and 'PF'. The data is as follows:

| first_name | last_name | salary   | PF        |
|------------|-----------|----------|-----------|
| Steven     | King      | 24000.00 | 3600.0000 |
| Neena      | Kochhar   | 17000.00 | 2550.0000 |
| Lex        | De Haan   | 17000.00 | 2550.0000 |
| Alexander  | Hunold    | 9000.00  | 1350.0000 |
| Bruce      | Ernst     | 6000.00  | 900.0000  |

At the bottom, the 'Output' pane shows the 'Action Output' table, which tracks the execution of the SQL statements:

| # | Time     | Action  | Message             | Duration / Fetch      |
|---|----------|---|---------------------|-----------------------|
| 1 | 21:12:52 | USE HR  | 0 row(s) affected   | 0.000 sec             |
| 2 | 21:12:55 | SHOW TABLES   | 8 row(s) returned   | 0.094 sec / 0.000 sec |
| 3 | 21:13:14 | SELECT CONCAT(first_name,',last_name) as NAME from emp_details_view LIMIT 0, 1000 | 106 row(s) returned | 0.062 sec / 0.000 sec |
| 4 | 21:21:49 | SELECT DISTINCT department_id FROM emp_details_view LIMIT 0, 1000                 | 11 row(s) returned  | 0.015 sec / 0.000 sec |
| 5 | 21:23:50 | SELECT * FROM employees ORDER BY first_name DESC LIMIT 0, 1000                    | 107 row(s) returned | 0.016 sec / 0.015 sec |
| 6 | 21:24:59 | SELECT first_name, last_name, salary, salary*.15 PF FROM employees LIMIT 0, 1000  | 107 row(s) returned | 0.016 sec / 0.000 sec |

**5)WRITE A QUERY TO GET THE EMPLOYEE ID, NAMES (FIRST\_NAME, LAST\_NAME), SALARY IN ASCENDING ORDER OF SALARY**

**SELECT EMPLOYEE\_ID , FIRST\_NAME, LAST\_NAME, SALARY FROM EMPLOYEES ORDER BY SALARY ASC;**

The screenshot shows the SQL Developer interface. The left pane displays the 'SCHEMAS' tree with 'hr' selected. The main editor shows a SQL query: `SELECT * FROM employees ORDER BY first_name DESC;`. The 'Result Grid' shows the following data:

| first_name | last_name | salary   | PF        |
|------------|-----------|----------|-----------|
| Steven     | King      | 24000.00 | 3600.0000 |
| Neena      | Kochhar   | 17000.00 | 2550.0000 |
| Lex        | De Haan   | 17000.00 | 2550.0000 |
| Alexander  | Hunold    | 9000.00  | 1350.0000 |
| Bruce      | Ernst     | 6000.00  | 900.0000  |

The 'Output' pane shows the execution log with the following entries:

| # | Time     | Action   | Message             | Duration / Fetch      |
|---|----------|--|---------------------|-----------------------|
| 2 | 21:12:55 | SHOW TABLES  | 8 row(s) returned   | 0.094 sec / 0.000 sec |
| 3 | 21:19:14 | SELECT CONCAT(first_name,' '  last_name) as NAME from emp_details_view LIMIT 0, 1000 | 106 row(s) returned | 0.062 sec / 0.000 sec |
| 4 | 21:21:49 | SELECT DISTINCT department_id FROM emp_details_view LIMIT 0, 1000                    | 11 row(s) returned  | 0.015 sec / 0.000 sec |
| 5 | 21:23:50 | SELECT * FROM employees ORDER BY first_name DESC LIMIT 0, 1000                       | 107 row(s) returned | 0.016 sec / 0.015 sec |
| 6 | 21:24:59 | SELECT first_name, last_name, salary, salary*.15 PF FROM employees LIMIT 0, 1000     | 107 row(s) returned | 0.016 sec / 0.000 sec |
| 7 | 21:27:05 | SELECT first_name, last_name, salary, salary*.15 PF FROM employees LIMIT 0, 1000     | 107 row(s) returned | 0.000 sec / 0.000 sec |

The status bar at the bottom indicates 'Query Completed'.

## 6. WRITE A QUERY TO GET THE TOTAL SALARIES PAYABLE TO EMPLOYEES

**SELECT SUM(SALARY) FROM EMPLOYEES;**

The screenshot shows the SQL Developer interface. The left pane displays the 'SCHEMAS' tree with 'hr', 'jimbjh', and 'newschema' schemas. The 'hr' schema is expanded, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'Tables' folder is selected, and the 'EMPLOYEES' table is highlighted. The main editor shows the following SQL query:

```
SELECT SUM(SALARY) FROM EMPLOYEES;
```

The 'Result Grid' tab is active, showing the query results. The results are displayed in a table with the following data:

| SUM(salary) |
|-------------|
| 691400.00   |

The bottom pane shows the 'Output' tab with the 'Action Output' view. It displays a list of actions and their results:

| # | Time     | Action  | Message             | Duration / Fetch      |
|---|----------|---|---------------------|-----------------------|
| 3 | 21:19:14 | SELECT CONCAT(first_name,'last_name') as NAME from emp_details_view LIMIT 0, 1000 | 106 row(s) returned | 0.062 sec / 0.000 sec |
| 4 | 21:21:49 | SELECT DISTINCT department_id FROM emp_details_view LIMIT 0, 1000                 | 11 row(s) returned  | 0.015 sec / 0.000 sec |
| 5 | 21:23:50 | SELECT * FROM employees ORDER BY first_name DESC LIMIT 0, 1000                    | 107 row(s) returned | 0.016 sec / 0.015 sec |
| 6 | 21:24:59 | SELECT first_name, last_name, salary, salary*.15 PF FROM employees LIMIT 0, 1000  | 107 row(s) returned | 0.016 sec / 0.000 sec |
| 7 | 21:27:05 | SELECT first_name, last_name, salary, salary*.15 PF FROM employees LIMIT 0, 1000  | 107 row(s) returned | 0.000 sec / 0.000 sec |
| 8 | 21:28:24 | SELECT SUM(salary) FROM employees LIMIT 0, 1000                                   | 1 row(s) returned   | 0.000 sec / 0.000 sec |

## 7. WRITE A QUERY TO GET THE MAXIMUM AND MINIMUM SALARY FROM EMPLOYEES TABLE

**SELECT MAX(salary), MIN(salary) FROM employees;**

The screenshot displays the SQL Developer interface with a query window titled 'SQL File 1'. The query is as follows:

```
8 • SELECT * FROM employees ORDER BY first_name DESC;
9 • SELECT first_name, last_name, salary, salary*.15 PF
10 FROM employees;
11 • SELECT employee_id, first_name, last_name, salary FROM employees ORDER BY salary ASC;
12 • SELECT SUM(salary) FROM employees;
13 • SELECT MAX(salary), MIN(salary) FROM employees;
14
15
```

The 'Result Grid' shows the output of the query, with columns 'MAX(salary)' and 'MIN(salary)'. The values are 24000.00 and 2100.00 respectively.

| MAX(salary) | MIN(salary) |
|-------------|-------------|
| 24000.00    | 2100.00     |

The 'Output' window shows the execution details of the query, including the time taken and the number of rows returned.

| # | Time     | Action   | Message             | Duration / Fetch      |
|---|----------|--|---------------------|-----------------------|
| 4 | 21:21:49 | SELECT DISTINCT department_id FROM emp_details_view LIMIT 0, 1000                | 11 row(s) returned  | 0.015 sec / 0.000 sec |
| 5 | 21:23:50 | SELECT * FROM employees ORDER BY first_name DESC LIMIT 0, 1000                   | 107 row(s) returned | 0.016 sec / 0.015 sec |
| 6 | 21:24:59 | SELECT first_name, last_name, salary, salary*.15 PF FROM employees LIMIT 0, 1000 | 107 row(s) returned | 0.016 sec / 0.000 sec |
| 7 | 21:27:05 | SELECT first_name, last_name, salary, salary*.15 PF FROM employees LIMIT 0, 1000 | 107 row(s) returned | 0.000 sec / 0.000 sec |
| 8 | 21:28:24 | SELECT SUM(salary) FROM employees LIMIT 0, 1000                                  | 1 row(s) returned   | 0.000 sec / 0.000 sec |
| 9 | 21:30:24 | SELECT MAX(salary), MIN(salary) FROM employees LIMIT 0, 1000                     | 1 row(s) returned   | 0.016 sec / 0.000 sec |

## 8. WRITE A QUERY TO GET THE AVERAGE SALARY AND NUMBER OF EMPLOYEES IN THE EMPLOYEES TABLE

**SELECT AVG(salary) ,COUNT(\*) FROM employees;**

The screenshot displays the SQL Developer interface. On the left, the 'SCHEMAS' pane shows the 'hr' schema selected. The main editor window contains the following SQL queries:

```
11 • SELECT employee_id , first_name, last_name, salary FROM employees ORDER BY salary ASC;
12 • SELECT SUM(salary) FROM employees;
13 • SELECT MAX(salary), MIN(salary) FROM employees;
14 • SELECT AVG(salary) ,COUNT(*) FROM employees;
```

Below the editor, the 'Result Grid' shows the execution results for the selected query (Query 9):

|   | AVG(salary) | COUNT(*) |
|---|-------------|----------|
| ▶ | 6461.682243 | 107      |

The 'Output' pane at the bottom shows the 'Action Output' table, which details the execution of the query:

| #  | Time     | Action  | Message             | Duration / Fetch      |
|----|----------|---|---------------------|-----------------------|
| 5  | 21:23:50 | SELECT * FROM employees ORDER BY first_name DESC LIMIT 0, 1000                  | 107 row(s) returned | 0.016 sec / 0.015 sec |
| 6  | 21:24:59 | SELECT first_name, last_name, salary, salary*15 PF FROM employees LIMIT 0, 1000 | 107 row(s) returned | 0.016 sec / 0.000 sec |
| 7  | 21:27:05 | SELECT first_name, last_name, salary, salary*15 PF FROM employees LIMIT 0, 1000 | 107 row(s) returned | 0.000 sec / 0.000 sec |
| 8  | 21:28:24 | SELECT SUM(salary) FROM employees LIMIT 0, 1000                                 | 1 row(s) returned   | 0.000 sec / 0.000 sec |
| 9  | 21:30:24 | SELECT MAX(salary), MIN(salary) FROM employees LIMIT 0, 1000                    | 1 row(s) returned   | 0.016 sec / 0.000 sec |
| 10 | 21:31:43 | SELECT AVG(salary),COUNT(*) FROM employees LIMIT 0, 1000                        | 1 row(s) returned   | 0.000 sec / 0.000 sec |

The status bar at the bottom indicates 'Query Completed'.



## 9. WRITE A QUERY TO GET THE NUMBER OF EMPLOYEES WORKING WITH THE COMPANY

**SELECT COUNT(\*) FROM EMPLOYEES;**

The screenshot shows the SQL Developer interface with a query window titled 'SQL File 1\*' containing the following SQL code:

```
14 • SELECT AVG(salary) ,COUNT(*) FROM employees ;
15 • SELECT COUNT(*) FROM employees ;
16
17
18
19
20
21
```

The 'Result Grid' tab is active, displaying the results of the second query. The grid shows a single row with the value 107.

| COUNT(*) |
|----------|
| 107      |

The 'Output' tab is also visible, showing a list of actions and their results:

| #  | Time     | Action  | Message             | Duration / Fetch      |
|----|----------|---|---------------------|-----------------------|
| 6  | 21:24:59 | SELECT first_name, last_name, salary, salary*15 PF FROM employees LIMIT 0, 1000 | 107 row(s) returned | 0.016 sec / 0.000 sec |
| 7  | 21:27:05 | SELECT first_name, last_name, salary, salary*15 PF FROM employees LIMIT 0, 1000 | 107 row(s) returned | 0.000 sec / 0.000 sec |
| 8  | 21:28:24 | SELECT SUM(salary) FROM employees LIMIT 0, 1000                                 | 1 row(s) returned   | 0.000 sec / 0.000 sec |
| 9  | 21:30:24 | SELECT MAX(salary), MIN(salary) FROM employees LIMIT 0, 1000                    | 1 row(s) returned   | 0.016 sec / 0.000 sec |
| 10 | 21:31:43 | SELECT AVG(salary), COUNT(*) FROM employees LIMIT 0, 1000                       | 1 row(s) returned   | 0.000 sec / 0.000 sec |
| 11 | 21:33:04 | SELECT COUNT(*) FROM employees LIMIT 0, 1000                                    | 1 row(s) returned   | 0.219 sec / 0.000 sec |

## 10. WRITE A QUERY TO GET THE NUMBER OF JOBS AVAILABLE IN THE EMPLOYEES TABLE

**SELECT COUNT(DISTINCT Job\_ID) FROM EMPLOYEES;**

The screenshot shows the SQL Developer interface. The left pane displays the 'SCHEMAS' tree with 'hr' selected. The main editor shows a list of SQL queries, with the 16th query highlighted: `SELECT count(DISTINCT Job_id) FROM employees;`. The 'Result Grid' tab is active, showing a single row with the value 19. The 'Output' pane at the bottom shows the execution log for the 12th query, which is the same as the one in the editor. The log entry is: `12 21:34:26 SELECT count(DISTINCT Job_id) FROM employees LIMIT 0, 1000`, with a message of '1 row(s) returned' and a duration of '0.000 sec / 0.000 sec'.

| #  | Time     | Action  | Message             | Duration / Fetch      |
|----|----------|---|---------------------|-----------------------|
| 7  | 21:27:05 | SELECT first_name, last_name, salary, salary*15 PF FROM employees LIMIT 0, 1000 | 107 row(s) returned | 0.000 sec / 0.000 sec |
| 8  | 21:28:24 | SELECT SUM(salary) FROM employees LIMIT 0, 1000                                 | 1 row(s) returned   | 0.000 sec / 0.000 sec |
| 9  | 21:30:24 | SELECT MAX(salary), MIN(salary) FROM employees LIMIT 0, 1000                    | 1 row(s) returned   | 0.016 sec / 0.000 sec |
| 10 | 21:31:43 | SELECT AVG(salary), COUNT(*) FROM employees LIMIT 0, 1000                       | 1 row(s) returned   | 0.000 sec / 0.000 sec |
| 11 | 21:33:04 | SELECT COUNT(*) FROM employees LIMIT 0, 1000                                    | 1 row(s) returned   | 0.219 sec / 0.000 sec |
| 12 | 21:34:26 | SELECT count(DISTINCT Job_id) FROM employees LIMIT 0, 1000                      | 1 row(s) returned   | 0.000 sec / 0.000 sec |

## 11. WRITE A QUERY GET ALL FIRST NAME FROM EMPLOYEES TABLE IN UPPER CASE

**SELECT UPPER(FIRST\_NAME) FROM EMPLOYEES;**

The screenshot shows the SQL Developer interface. The 'Schemas' pane on the left shows the 'hr' schema selected. The main editor displays a SQL query: `SELECT UPPER(first_name) FROM employees;`. The 'Result Grid' shows the output of the query, which is a list of first names in uppercase: STEVEN, NEENA, LEX, ALEXANDER, and BRUCE. The 'Output' pane at the bottom shows the execution log, indicating that the query was successful and returned 107 rows.

Schema: hr

Result Grid

| UPPER(first_name) |
|-------------------|
| STEVEN            |
| NEENA             |
| LEX               |
| ALEXANDER         |
| BRUCE             |

Result 12 x

Output

Action Output

| #  | Time     | Action   | Message             | Duration / Fetch      |
|----|----------|--|---------------------|-----------------------|
| 8  | 21:28:24 | SELECT SUM(salary) FROM employees LIMIT 0, 1000              | 1 row(s) returned   | 0.000 sec / 0.000 sec |
| 9  | 21:30:24 | SELECT MAX(salary), MIN(salary) FROM employees LIMIT 0, 1000 | 1 row(s) returned   | 0.016 sec / 0.000 sec |
| 10 | 21:31:43 | SELECT AVG(salary), COUNT(*) FROM employees LIMIT 0, 1000    | 1 row(s) returned   | 0.000 sec / 0.000 sec |
| 11 | 21:33:04 | SELECT COUNT(*) FROM employees LIMIT 0, 1000                 | 1 row(s) returned   | 0.219 sec / 0.000 sec |
| 12 | 21:34:26 | SELECT count(DISTINCT Job_id) FROM employees LIMIT 0, 1000   | 1 row(s) returned   | 0.000 sec / 0.000 sec |
| 13 | 21:35:57 | SELECT UPPER(first_name) FROM employees LIMIT 0, 1000        | 107 row(s) returned | 0.000 sec / 0.000 sec |

Query Completed

## 12. WRITE A QUERY TO GET THE FIRST 3 CHARACTERS OF FIRST NAME FROM EMPLOYEES

**SELECT SUBSTRING(FIRST\_NAME,1,3) FROM EMPLOYEES;**

The screenshot shows the SQL Developer interface with the following components:

- Navigator:** Shows the database structure with schemas `hr`, `jimbih`, and `newschema`. The `hr` schema is selected, showing tables, views, stored procedures, and functions.
- SQL Editor:** Contains the query: `SELECT SUBSTRING(first_name,1,3) FROM employees;`
- Result Grid:** Displays the results of the query. The first column is labeled `SUBSTRING(first_name,1,3)`. The results are: `Ste`, `Nee`, `Lex`, `Ale`, and `Bru`.
- Output:** Shows the execution log with the following entries:

| #  | Time     | Action  | Message             | Duration / Fetch      |
|----|----------|---|---------------------|-----------------------|
| 9  | 21:30:24 | SELECT MAX(salary), MIN(salary) FROM employees LIMIT 0, 1000  | 1 row(s) returned   | 0.016 sec / 0.000 sec |
| 10 | 21:31:43 | SELECT AVG(salary), COUNT(*) FROM employees LIMIT 0, 1000     | 1 row(s) returned   | 0.000 sec / 0.000 sec |
| 11 | 21:33:04 | SELECT COUNT(*) FROM employees LIMIT 0, 1000                  | 1 row(s) returned   | 0.219 sec / 0.000 sec |
| 12 | 21:34:26 | SELECT count(DISTINCT Job_id) FROM employees LIMIT 0, 1000    | 1 row(s) returned   | 0.000 sec / 0.000 sec |
| 13 | 21:35:57 | SELECT UPPER(first_name) FROM employees LIMIT 0, 1000         | 107 row(s) returned | 0.000 sec / 0.000 sec |
| 14 | 21:36:45 | SELECT SUBSTRING(first_name,1,3) FROM employees LIMIT 0, 1000 | 107 row(s) returned | 0.000 sec / 0.000 sec |

### 13. WRITE A QUERY TO GET FIRST NAME FROM EMPLOYEES TABLE AFTER REMOVING WHITE SPACES FROM BOTH SIDE

**SELECT TRIM(FIRST\_NAME) FROM EMPLOYEES;**

The screenshot shows the SQL Developer interface. The left pane displays the 'SCHEMAS' tree with 'hr' selected. The main editor shows a query: `SELECT TRIM(first_name) FROM employees;`. The 'Result Grid' shows the first five results: Steven, Neena, Lex, Alexander, and Bruce. The 'Output' pane shows the execution log for the query.

| #    | Time     | Action   | Message             | Duration / Fetch      |
|------|----------|--|---------------------|-----------------------|
| ✓ 11 | 21:33:04 | SELECT COUNT(*) FROM employees LIMIT 0, 1000                             | 1 row(s) returned   | 0.219 sec / 0.000 sec |
| ✓ 12 | 21:34:26 | SELECT count(DISTINCT Job_id) FROM employees LIMIT 0, 1000               | 1 row(s) returned   | 0.000 sec / 0.000 sec |
| ✓ 13 | 21:35:57 | SELECT UPPER(first_name) FROM employees LIMIT 0, 1000                    | 107 row(s) returned | 0.000 sec / 0.000 sec |
| ✓ 14 | 21:36:45 | SELECT SUBSTRING(first_name,1,3) FROM employees LIMIT 0, 1000            | 107 row(s) returned | 0.000 sec / 0.000 sec |
| ✓ 15 | 21:37:39 | SELECT LTRIM(first_name), rtrim(first_name) FROM employees LIMIT 0, 1000 | 107 row(s) returned | 0.000 sec / 0.000 sec |
| ✓ 16 | 21:38:14 | SELECT TRIM(first_name) FROM employees LIMIT 0, 1000                     | 107 row(s) returned | 0.000 sec / 0.000 sec |

#### 14. WRITE A QUERY TO GET THE LENGTH OF THE EMPLOYEE NAMES (FIRST\_NAME, LAST\_NAME) FROM EMPLOYEES TABLE

**SELECT LENGTH(FIRST\_NAME), LENGTH(LAST\_NAME) FROM EMPLOYEES;**

The screenshot shows the SQL Developer interface. The left pane displays the 'SCHEMAS' tree with 'hr' selected. The main editor shows a list of queries, with the 21st query highlighted: `SELECT LENGTH(first_name), length(last_name) FROM employees;`. Below the editor, the 'Result Grid' shows the output of this query. The grid has two columns: 'LENGTH(first\_name)' and 'length(last\_name)'. The results are as follows:

| LENGTH(first_name) | length(last_name) |
|--------------------|-------------------|
| 6                  | 4                 |
| 5                  | 7                 |
| 3                  | 7                 |
| 9                  | 6                 |
| 5                  | 5                 |

Below the result grid, the 'Output' pane shows the 'Action Output' table, which includes details about the execution of the queries, such as the time taken and the number of rows returned.

| #  | Time     | Action   | Message  | Duration / Fetch      |
|----|----------|--|--|-----------------------|
| 17 | 21:40:33 | SELECT LENGTH(first_name), length(last_name) FROM employees LIMIT 0, 1000                  | 107 row(s) returned  | 0.016 sec / 0.000 sec |
| 18 | 21:41:01 | SELECT * FROM employees WHERE first_name REGEXP '[0-9]' LIMIT 0, 1000                      | 0 row(s) returned  | 0.078 sec / 0.000 sec |
| 19 | 21:41:19 | SELECT first_name, last_name, LENGTH(first_name) + Length(last_name) 'length of names F... | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.031 sec             |
| 20 | 21:41:28 | SELECT first_name, last_name, LENGTH(first_name) + Length(last_name) 'length of names F... | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 21 | 21:42:09 | SELECT LENGTH(first_name), length(last_name) FROM employees LIMIT 0, 1000                  | 107 row(s) returned  | 0.000 sec / 0.000 sec |
| 22 | 21:55:52 | SELECT LENGTH(first_name), length(last_name) FROM employees LIMIT 0, 1000                  | 107 row(s) returned  | 0.000 sec / 0.000 sec |

```
SELECT * FROM EMPLOYEES WHERE FIRST_NAME REGEXP '(0-9)';
```

Navigator

SCHEMAS

Filter objects

hr

Tables

Views

Stored Procedures

Functions

jiminh

Tables

Views

Stored Procedures

Functions

newschema

Administration

Schemas

Information

Schema: hr

SQL File 1\*

Limit to 1000 rows

16

•

SELECT count(DISTINCT Job\_id) FROM employees;

17

•

SELECT UPPER(first\_name) FROM employees;

18

•

SELECT SUBSTRING(first\_name,1,3) FROM employees;

19

•

SELECT LTRIM(first\_name), rtrim(first\_name) FROM employees;

20

•

SELECT TRIM(first\_name) FROM employees;

21

•

SELECT LENGTH(first\_name), length(last\_name) FROM employees;

22

•

SELECT \* FROM employees WHERE first\_name REGEXP '(0-9)';

23

•

Result Grid

Filter Rows:

Edit

Export/Import

Wrap Cell Content

|   | employee_id | first_name | last_name | email | phone_number | hire_date | job_id | salary | commission_pct | manager_id | department_id |
|---|-------------|------------|-----------|-------|--------------|-----------|--------|--------|----------------|------------|---------------|
| 1 | NULL        | NULL       | NULL      | NULL  | NULL         | NULL      | NULL   | NULL   | NULL           | NULL       | NULL          |

employees 23

Apply

Output

Action Output

| #    | Time     | Action   | Message   | Duration / Fetch      |
|------|----------|--|---|-----------------------|
| ✓ 22 | 21:55:52 | SELECT LENGTH(first_name), length(last_name) FROM employees LIMIT 0, 1000      | 107 row(s) returned   | 0.000 sec / 0.000 sec |
| ✓ 23 | 22:09:01 | SELECT first_name FROM employees WHERE first_name REGEXP '(0-9)' LIMIT 0, 1000 | 0 row(s) returned   | 0.063 sec / 0.000 sec |
| ✗ 24 | 22:10:08 | SELECT FROM employees WHERE first_name REGEXP '(0-9)'                          | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to... | 0.000 sec             |
| ✓ 25 | 22:10:21 | SELECT * FROM employees WHERE first_name REGEXP '(0-9)' LIMIT 0, 1000          | 0 row(s) returned   | 0.000 sec / 0.000 sec |
| ✓ 26 | 22:11:04 | SELECT * FROM employees WHERE first_name REGEXP '(0-9)' LIMIT 0, 1000          | 0 row(s) returned   | 0.000 sec / 0.000 sec |
| ✓ 27 | 22:11:18 | SELECT * FROM employees WHERE first_name REGEXP '(0-9)' LIMIT 0, 1000          | 0 row(s) returned   | 0.000 sec / 0.000 sec |

Object Info

Session

Query Completed

**16. WRITE A QUERY TO DISPLAY THE NAME (FIRST\_NAME, LAST\_NAME) AND SALARY FOR ALL EMPLOYEES WHOSE SALARY IS NOT IN THE RANGE \$10,000 THROUGH \$15,000**

**SELECT FIRST\_NAME, LAST\_NAME, SALARY**

**FROM EMPLOYEES WHERE SALARY NOT BETWEEN 10000 AND 15000 ORDER BY SALARY;**

The screenshot shows the SQL Developer interface. The left pane displays the 'SCHEMAS' tree with 'hr' selected. The main editor shows a SQL script with the following queries:

```
20 • SELECT TRIM(first_name) FROM employees;
21 • SELECT LENGTH(first_name), length(last_name) FROM employees;
22 • SELECT * FROM employees WHERE first_name REGEXP '(0-9)';
23 • SELECT first_name, last_name, salary
24 FROM employees WHERE salary NOT BETWEEN 10000 AND 15000 ORDER BY SALARY;
25
26
```

The 'Result Grid' shows the results of query 24:

| first_name | last_name  | salary  |
|------------|------------|---------|
| TJ         | Olson      | 2100.00 |
| Hazel      | Philtanker | 2200.00 |
| Steven     | Markle     | 2200.00 |
| Ki         | Gee        | 2400.00 |
| James      | Landry     | 2400.00 |

The 'Output' pane shows the execution log:

| #  | Time     | Action   | Message  | Duration / Fetch      |
|----|----------|--|--|-----------------------|
| 23 | 22:09:01 | SELECT first_name FROM employees WHERE first_name REGEXP '[0-9]' LIMIT 0, 1000     | 0 row(s) returned  | 0.063 sec / 0.000 sec |
| 24 | 22:10:08 | SELECT FROM employees WHERE first_name REGEXP '[0-9]'                              | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 25 | 22:10:21 | SELECT * FROM employees WHERE first_name REGEXP '[0-9]' LIMIT 0, 1000              | 0 row(s) returned  | 0.000 sec / 0.000 sec |
| 26 | 22:11:04 | SELECT * FROM employees WHERE first_name REGEXP '[0-9]' LIMIT 0, 1000              | 0 row(s) returned  | 0.000 sec / 0.000 sec |
| 27 | 22:11:18 | SELECT * FROM employees WHERE first_name REGEXP '[0-9]' LIMIT 0, 1000              | 0 row(s) returned  | 0.000 sec / 0.000 sec |
| 28 | 22:19:23 | SELECT first_name, last_name, salary FROM employees WHERE salary NOT BETWEEN 10... | 91 row(s) returned   | 0.047 sec / 0.000 sec |

Query Completed



**17. WRITE A QUERY TO DISPLAY THE NAME (FIRST\_NAME, LAST\_NAME) AND DEPARTMENT ID OF ALL EMPLOYEES IN DEPARTMENTS 30 OR 100 IN ASCENDING ORDER**

**SELECT FIRST\_NAME, LAST\_NAME, DEPARTMENT\_ID**

**FROM EMPLOYEES WHERE DEPARTMENT\_ID IN(30,100) ORDER BY DEPARTMENT\_ID;**

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
22 • SELECT * FROM employees WHERE first_name REGEXP '(0-9)';
23 • SELECT first_name, last_name, salary
24 FROM employees WHERE salary NOT BETWEEN 10000 AND 15000 ORDER BY SALARY;
25 • SHOW TABLES;
26 • SELECT * FROM departments;
27 • SELECT first_name, last_name, department_id
28 FROM employees WHERE department_id IN(30,100) ORDER BY department_id;
29
```

The results pane displays the output of the query in a table format:

| first_name | last_name | department_id |
|------------|-----------|---------------|
| Den        | Raphaely  | 30            |
| Alexander  | Khoo      | 30            |
| Shelli     | Baida     | 30            |
| Sigal      | Tobias    | 30            |
| Guy        | Himuro    | 30            |

The bottom pane shows the execution log with the following entries:

| #  | Time     | Action  | Message  | Duration / Fetch      |
|----|----------|---|--|-----------------------|
| 29 | 22:21:15 | SELECT employees LIMIT 0, 1000  | Error Code: 1054. Unknown column 'employees' in 'field list'                                   | 0.016 sec             |
| 30 | 22:22:04 | SELECT * FROM employees LIMIT 0, 1000   | 107 row(s) returned  | 0.000 sec / 0.000 sec |
| 31 | 22:24:08 | SHOW TABLES   | 8 row(s) returned  | 0.015 sec / 0.000 sec |
| 32 | 22:24:54 | SELECT * FROM departments   | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 33 | 22:25:08 | SELECT * FROM departments LIMIT 0, 1000   | 27 row(s) returned   | 0.016 sec / 0.000 sec |
| 34 | 22:30:20 | SELECT first_name, last_name, department_id FROM employees WHERE department_id IN(... | 12 row(s) returned   | 0.062 sec / 0.000 sec |

**18. WRITE A QUERY TO DISPLAY THE NAME (FIRST\_NAME, LAST\_NAME) AND SALARY FOR ALL EMPLOYEES WHOSE SALARY IS NOT IN THE RANGE \$10,000 THROUGH \$15,000 AND ARE IN DEPARTMENT 30 OR 100**

**SELECT FIRST\_NAME, LAST\_NAME, SALARY, DEPARTMENT\_ID**

**FROM EMPLOYEES WHERE SALARY NOT BETWEEN 10000 AND 15000 AND DEPARTMENT\_ID IN(30,100);**

The screenshot shows the SQL Developer interface with a query window and a results grid. The query is as follows:

```
27 • SELECT first_name, last_name, department_id
28 FROM employees WHERE department_id IN(30,100) ORDER BY department_id;
29 • SELECT first_name, last_name, salary, department_id
30 FROM employees WHERE salary NOT BETWEEN 10000 AND 15000 AND department_id IN(30,100);
31
32
33
```

The results grid displays the following data:

| first_name | last_name  | salary  | department_id |
|------------|------------|---------|---------------|
| Alexander  | Khoo       | 3100.00 | 30            |
| Shelli     | Baida      | 2900.00 | 30            |
| Sigal      | Tobias     | 2800.00 | 30            |
| Guy        | Himuro     | 2600.00 | 30            |
| Karen      | Colmenares | 2500.00 | 30            |

The output window shows the following messages:

| #  | Time     | Action   | Message   | Duration / Fetch      |
|----|----------|--|---|-----------------------|
| 31 | 22:24:08 | SHOW TABLES  | 8 row(s) returned   | 0.015 sec / 0.000 sec |
| 32 | 22:24:54 | SELECT * departments   | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to... | 0.000 sec             |
| 33 | 22:25:08 | SELECT * FROM departments LIMIT 0, 1000  | 27 row(s) returned  | 0.016 sec / 0.000 sec |
| 34 | 22:30:20 | SELECT first_name, last_name, department_id FROM employees WHERE department_id IN(...  | 12 row(s) returned  | 0.062 sec / 0.000 sec |
| 35 | 22:38:23 | SELECT first_name, last_name, salary, department_id FROM employees WHERE salary NOT... | 10 row(s) returned  | 0.062 sec / 0.000 sec |
| 36 | 22:39:36 | SELECT first_name, last_name, salary, department_id FROM employees WHERE salary NOT... | 10 row(s) returned  | 0.015 sec / 0.000 sec |

Query Completed

## 19. WRITE A QUERY TO DISPLAY THE NAME (FIRST\_NAME, LAST\_NAME) AND HIRE DATE FOR ALL EMPLOYEES WHO WERE HIRED IN 1987

**SELECT FIRST\_NAME, LAST\_NAME, HIRE\_DATE**

**FROM EMPLOYEES WHERE YEAR(HIRE\_DATE) LIKE '1987%';**

The screenshot shows the SQL Developer interface. The left sidebar displays the 'SCHEMAS' tree with 'hr' selected. The main window shows a SQL script with the following queries:

```
27 • SELECT first_name, last_name, department_id
28 FROM employees WHERE department_id IN(30,100) ORDER BY department_id;
29 • SELECT first_name, last_name, salary, department_id
30 FROM employees WHERE salary NOT BETWEEN 10000 AND 15000 AND department_id IN(30,100);
31 • SELECT first_name, last_name, hire_date
32 FROM employees WHERE YEAR(hire_date) LIKE '1987%';
33
34
```

The 'Result Grid' shows the results of the query in line 31:

| first_name | last_name | hire_date  |
|------------|-----------|------------|
| Steven     | King      | 1987-06-17 |
| Jennifer   | Whalen    | 1987-09-17 |

The 'Action Output' pane shows the execution log:

| #    | Time     | Action   | Message  | Duration / Fetch      |
|------|----------|--|--|-----------------------|
| ✓ 35 | 22:38:23 | SELECT first_name, last_name, salary, department_id FROM employees WHERE salary NOT... | 10 row(s) returned   | 0.062 sec / 0.000 sec |
| ✓ 36 | 22:39:36 | SELECT first_name, last_name, salary, department_id FROM employees WHERE salary NOT... | 10 row(s) returned   | 0.015 sec / 0.000 sec |
| ✗ 37 | 22:42:16 | SELECT first_name, last_name, hire_date WHERE hire_date YEAR(1987)                     | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| ✗ 38 | 22:43:05 | SELECT first_name, last_name, hire_date FROM employees WHERE hire_date YEAR(1987)      | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| ✗ 39 | 22:46:09 | SELECT first_name, last_name, hire_date FROM employees WHERE hire_date YEAR(hire_d...  | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| ✓ 40 | 22:46:41 | SELECT first_name, last_name, hire_date FROM employees WHERE YEAR(hire_date) LIKE '... | 2 row(s) returned  | 0.047 sec / 0.000 sec |

At the bottom, a status bar indicates 'Query Completed'.

## 20. WRITE A QUERY TO DISPLAY THE FIRST\_NAME OF ALL EMPLOYEES WHO HAVE BOTH "B" AND "C" IN THEIR FIRST NAME

**SELECT first\_name FROM employees WHERE first\_name LIKE '%B%' AND first\_name LIKE '%c%';**

The screenshot shows the SQL Developer interface with a query window titled 'SQL File 1'. The query is as follows:

```
31 • SELECT first_name, last_name, hire_date
32 FROM employees WHERE YEAR(hire_date) LIKE '1987%';
33 • SELECT first_name FROM employees
34 where first_name LIKE '%b%'
35 and first_name LIKE '%c%';
36
37
```

The 'Result Grid' tab is active, showing the results of the query. The results are as follows:

| first_name |
|------------|
| Bruce      |

The 'Output' tab is also active, showing the execution log. The log contains the following entries:

| #  | Time     | Action  | Message  | Duration / Fetch      |
|----|----------|---|--|-----------------------|
| 47 | 22:58:50 | SELECT first_name FROM employees where first_name LIKE 'b%' and where first_name LIK... | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 48 | 22:59:16 | SELECT first_name FROM employees where first_name LIKE '%b%' and where first_name LI... | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 49 | 22:59:36 | SELECT first_name FROM employees where first_name LIKE '%b%' and first_name LIKE '%c... | 1 row(s) returned  | 0.000 sec / 0.000 sec |
| 50 | 22:59:59 | SELECT first_name FROM employees where first_name LIKE '%b%' and first_name LIKE '%d... | 0 row(s) returned  | 0.000 sec / 0.000 sec |
| 51 | 23:00:05 | SELECT first_name FROM employees where first_name LIKE '%b%' and first_name LIKE '%b... | 5 row(s) returned  | 0.000 sec / 0.000 sec |
| 52 | 23:00:09 | SELECT first_name FROM employees where first_name LIKE '%b%' and first_name LIKE '%c... | 1 row(s) returned  | 0.000 sec / 0.000 sec |

The 'Object Info' tab is also active, showing the schema 'hr'.

**21. WRITE A QUERY TO DISPLAY THE LAST NAME, JOB, AND SALARY FOR ALL EMPLOYEES WHOSE JOB IS THAT OF A PROGRAMMER OR A SHIPPING CLERK, AND WHOSE SALARY IS NOT EQUAL TO \$4,500, \$10,000, OR \$15,000**

**SELECT LAST\_NAME , JOB\_ID, SALARY**

**FROM EMPLOYEES WHERE JOB\_ID IN('IT\_PROG', 'SH\_CLERK') AND SALARY NOT IN(4500.10000,15000) ORDER BY JOB\_ID;**

The screenshot shows the SQL Developer interface. The left pane displays the 'SCHEMAS' tree with 'hr' selected. The main editor shows a query with line numbers 33 to 40. The query is: `SELECT first_name FROM employees` (line 33), `where first_name LIKE '%b%'` (line 34), `and first_name LIKE '%c%';` (line 35), `SELECT * FROM employees;` (line 36), `SELECT last_name , job_id, salary` (line 37), `FROM employees WHERE job_id IN('IT_PROG', 'SH_CLERK') and SALARY NOT IN(4500.10000,15000) ORDER BY job_id;` (line 38). The 'Result Grid' shows the results of the query, with columns 'last\_name', 'job\_id', and 'salary'. The results are: 

| last_name | job_id  | salary  |
|-----------|---------|---------|
| Hunold    | IT_PROG | 9000.00 |
| Ernst     | IT_PROG | 6000.00 |
| Austin    | IT_PROG | 4800.00 |
| Pataballa | IT_PROG | 4800.00 |
| Lorentz   | IT_PROG | 4200.00 |

. The 'Output' pane shows the execution of the query, with a message: 'Error Code: 1054. Unknown column 'IT\_PROG' in 'where clause''. The 'Query Completed' status is shown at the bottom.

SQL File 1\* x

Limit to 1000 rows

33 • SELECT first\_name FROM employees

34 where first\_name LIKE '%b%'

35 and first\_name LIKE '%c%';

36 • SELECT \* FROM employees;

37 • SELECT last\_name , job\_id, salary

38 FROM employees WHERE job\_id IN('IT\_PROG', 'SH\_CLERK') and SALARY NOT IN(4500.10000,15000) ORDER BY job\_id;

39

40

Result Grid

| last_name | job_id  | salary  |
|-----------|---------|---------|
| Hunold    | IT_PROG | 9000.00 |
| Ernst     | IT_PROG | 6000.00 |
| Austin    | IT_PROG | 4800.00 |
| Pataballa | IT_PROG | 4800.00 |
| Lorentz   | IT_PROG | 4200.00 |

employees 23 employees 41 x

Output

Action Output

| #  | Time     | Action  | Message  | Duration / Fetch      |
|----|----------|---|--|-----------------------|
| 56 | 23:15:18 | SELECT last_name , job_id, salary FROM employees WHERE job_id IN('IT_PROG', 'SH_CLERK'... | Error Code: 1054. Unknown column 'IT_PROG' in 'where clause' | 0.047 sec             |
| 57 | 23:15:42 | SELECT last_name , job_id, salary FROM employees WHERE job_id IN('IT_PROG', 'SH_CLERK'... | 25 row(s) returned   | 0.094 sec / 0.000 sec |
| 58 | 23:16:59 | SELECT last_name , job_id, salary FROM employees WHERE job_id IN('IT_PROG', 'SH_CLERK'... | 25 row(s) returned   | 0.016 sec / 0.000 sec |
| 59 | 23:17:40 | SELECT last_name , job_id, salary FROM employees WHERE job_id IN('IT_PROG', 'SH_CLERK'... | 25 row(s) returned   | 0.000 sec / 0.000 sec |
| 60 | 23:17:54 | SELECT * FROM employees LIMIT 0, 1000   | 107 row(s) returned  | 0.000 sec / 0.000 sec |
| 61 | 23:17:58 | SELECT last_name , job_id, salary FROM employees WHERE job_id IN('IT_PROG', 'SH_CLERK'... | 25 row(s) returned   | 0.000 sec / 0.000 sec |

Object Info Session

Query Completed

## 22. WRITE A QUERY TO DISPLAY THE LAST NAME OF EMPLOYEES WHOSE NAMES HAVE EXACTLY 6 CHARACTERS

**SELECT LAST\_NAME FROM EMPLOYEES WHERE LAST\_NAME LIKE '\_\_\_\_\_';**

The screenshot shows the SQL Developer interface. The left pane displays the 'SCHEMAS' tree with 'hr' selected. The main editor shows a SQL query with line numbers 36 to 42. The query is:

```
36 • SELECT * FROM employees;  
37 • SELECT last_name , job_id, salary  
38 FROM employees WHERE job_id IN('IT_PROG', 'SH_Clerk') and SALARY NOT IN(4500.10000,15000) ORDER BY job_id;  
39 • SELECT last_name FROM employees WHERE last_name LIKE '_____';  
40  
41  
42
```

The 'Result Grid' shows the results of the query. The first column is 'last\_name'. The results are:

| last_name |
|-----------|
| Hunold    |
| Austin    |
| Faviet    |
| Tobias    |
| Himuro    |

The bottom pane shows the 'Output' window with a table of execution details:

| #  | Time     | Action  | Message  | Duration / Fetch      |
|----|----------|---|--|-----------------------|
| 58 | 23:16:59 | SELECT last_name , job_id, salary FROM employees WHERE job_id IN('IT_PROG', 'SH_Clerk'... | 25 row(s) returned   | 0.016 sec / 0.000 sec |
| 59 | 23:17:40 | SELECT last_name , job_id, salary FROM employees WHERE job_id IN('IT_PROG', 'SH_Clerk'... | 25 row(s) returned   | 0.000 sec / 0.000 sec |
| 60 | 23:17:54 | SELECT * FROM employees LIMIT 0,1000  | 107 row(s) returned  | 0.000 sec / 0.000 sec |
| 61 | 23:17:58 | SELECT last_name , job_id, salary FROM employees WHERE job_id IN('IT_PROG', 'SH_Clerk'... | 25 row(s) returned   | 0.000 sec / 0.000 sec |
| 62 | 23:19:29 | SELECT last_name WHERE IN(6)  | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 63 | 23:21:13 | SELECT last_name FROM employees WHERE last_name LIKE '_____ ' LIMIT 0,1000                | 28 row(s) returned   | 0.016 sec / 0.000 sec |

## 23. WRITE A QUERY TO DISPLAY THE LAST NAME OF EMPLOYEES HAVING 'E' AS THE THIRD CHARACTER

**SELECT LAST\_NAME FROM EMPLOYEES WHERE LAST\_NAME LIKE '\_\_E%';**

The screenshot shows the SQL Developer interface. The left sidebar displays the 'SCHEMAS' tree with 'hr' selected. The main editor shows a SQL query: `SELECT last_name FROM employees WHERE last_name LIKE '__E%';`. The 'Result Grid' tab is active, showing a list of employee last names: Greenberg, Chen, Gee, McEwen, and Greene. The 'Output' tab is also visible, showing a table of execution results.

| #  | Time     | Action  | Message  | Duration / Fetch      |
|----|----------|---|--|-----------------------|
| 66 | 23:24:08 | SELECT last_name FROM employees WHERE last_name LIKE('__') LIMIT 0, 1000  | Error Code: 1054. Unknown column 'e' in 'where clause'   | 0.000 sec             |
| 67 | 23:24:39 | SELECT last_name FROM employees WHERE last_name LIKE('__e%')              | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 68 | 23:25:02 | SELECT last_name FROM employees WHERE last_name LIKE('__e%'               | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 69 | 23:25:31 | SELECT last_name FROM employees WHERE last_name LIKE '__e%' LIMIT 0, 1000 | 0 row(s) returned  | 0.000 sec / 0.000 sec |
| 70 | 23:25:37 | SELECT last_name FROM employees WHERE last_name LIKE '__e%' LIMIT 0, 1000 | 13 row(s) returned   | 0.016 sec / 0.000 sec |
| 71 | 23:25:47 | SELECT last_name FROM employees WHERE last_name LIKE '__e%' LIMIT 0, 1000 | 13 row(s) returned   | 0.000 sec / 0.000 sec |

Query Completed

**24. WRITE A QUERY TO GET THE JOB\_ID AND RELATED EMPLOYEE'S ID PARTIAL OUTPUT OF THE QUERY: JOB\_ID EMPLOYEES ID**

**AC\_ACCOUNT 206**

**AC\_MGR 205**

**AD\_ASST 200**

**AD\_PRES 100**

**AD\_VP 101,102**

**FI\_ACCOUNT 110,113,111,109,112**

**SELECT JOB\_ID, GROUP\_CONCAT(EMPLOYEE\_ID, ' ') 'EMPLOYEES ID'**

**FROM EMPLOYEES GROUP BY JOB\_ID;**

The screenshot shows the SQL Developer interface. The left pane displays the 'SCHEMAS' tree with 'hr' and 'jimpjh' schemas expanded. The main editor contains the following SQL query:

```
SELECT job_id, GROUP_CONCAT(employee_id, ' ') 'Employees ID'
FROM employees GROUP BY job_id;
```

The 'Result Grid' shows the output of the query:

| job_id     | Employees ID |
|------------|--------------|
| AC_ACCOUNT | 206          |
| AC_MGR     | 205          |
| AD_ASST    | 200          |
| AD_PRES    | 100          |
| AD_VP      | 101,102      |

The 'Output' pane shows the execution log with the following entries:

| #  | Time     | Action   | Message  | Duration / Fetch      |
|----|----------|--|--|-----------------------|
| 46 | 22:52:21 | UPDATE employees SET phone_number = REPLACE(phone_number,'123','999') WHERE...     | 0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0                                       | 0.000 sec             |
| 47 | 22:52:24 | SELECT * FROM employees LIMIT 0,1000   | 107 row(s) returned  | 0.000 sec / 0.000 sec |
| 48 | 22:52:38 | UPDATE employees SET phone_number = REPLACE(phone_number,'124','999') WHERE...     | 0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0                                       | 0.000 sec             |
| 49 | 22:52:40 | SELECT * FROM employees LIMIT 0,1000   | 107 row(s) returned  | 0.000 sec / 0.000 sec |
| 50 | 22:53:24 | UPDATE employees substring(phone_number = REPLACE(phone_number,'124','999') W...   | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 51 | 22:58:07 | SELECT job_id, GROUP_CONCAT(employee_id, ' ') 'Employees ID' FROM employees GRO... | 19 row(s) returned   | 0.016 sec / 0.000 sec |



**25. WRITE A QUERY TO UPDATE THE PORTION OF THE PHONE\_NUMBER IN THE EMPLOYEES TABLE, WITHIN THE PHONE NUMBER THE SUBSTRING '124' WILL BE REPLACED BY '999'**

**UPDATE EMPLOYEES**

**SET PHONE\_NUMBER = REPLACE('PHONE\_NUMBER','124','999')**

**WHERE PHONE\_NUMBER LIKE ('124');**

The screenshot shows the SQL Developer interface with a query window titled 'SQL File 1'. The query is as follows:

```
40 • SELECT last_name FROM employees WHERE last_name LIKE '_e%';
41 • SHOW TABLES;
42
43 • SELECT job_id, GROUP_CONCAT(employee_id, ' ') 'Employees ID'
44 FROM employees GROUP BY job_id;
45 • UPDATE employees
46 SET phone_number = REPLACE('phone_number','124','999')
47 WHERE phone_number LIKE ('124');
48 • SELECT * FROM employees;
49
50
51
52
53
```

The 'Output' window shows the execution results for the queries:

| #  | Time     | Action  | Message  | Duration / Fetch      |
|----|----------|---|--|-----------------------|
| 45 | 22:51:16 | SELECT * FROM employees LIMIT 0,1000  | 107 row(s) returned  | 0.000 sec / 0.016 sec |
| 46 | 22:52:21 | UPDATE employees SET phone_number = REPLACE(phone_number,'123','999') WHERE...    | 0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0                                       | 0.000 sec             |
| 47 | 22:52:24 | SELECT * FROM employees LIMIT 0,1000  | 107 row(s) returned  | 0.000 sec / 0.000 sec |
| 48 | 22:52:38 | UPDATE employees SET phone_number = REPLACE(phone_number,'124','999') WHERE...    | 0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0                                       | 0.000 sec             |
| 49 | 22:52:40 | SELECT * FROM employees LIMIT 0,1000  | 107 row(s) returned  | 0.000 sec / 0.000 sec |
| 50 | 22:53:24 | UPDATE employees substring(phone_number = REPLACE(phone_number,'124','999')) W... | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |

The 'Object Info' window shows 'No object selected'.

## 26. WRITE A QUERY TO GET THE DETAILS OF THE EMPLOYEES WHERE THE LENGTH OF THE FIRST NAME GREATER THAN OR EQUAL TO 8

**SELECT \* FROM EMPLOYEES WHERE**

**LENGTH(FIRST\_NAME) >=8;**

The screenshot shows the SQL Developer interface. The left pane displays the 'SCHEMAS' tree with 'hr' selected. The main editor shows a query with line numbers 43 to 49. The query is:

```
43 SELECT job_id, GROUP_CONCAT(employee_id, ' ') 'Employees ID'
44 FROM employees GROUP BY job_id;
45 SELECT * FROM employees WHERE
46 length(first_name) >=8;
47
48
49
```

The 'Result Grid' shows the following data:

| employee_id | first_name  | last_name | email   | phone_number       | hire_date  | job_id     | salary  | commission_pct | manager_id | department_id |
|-------------|-------------|-----------|---------|--------------------|------------|------------|---------|----------------|------------|---------------|
| 103         | Alexander   | Hunold    | AHUNOLD | 590.423.4567       | 1990-01-03 | IT_PROG    | 9000.00 | NULL           | 102        | 60            |
| 112         | Jose Manuel | Urman     | JMURMAN | 515.124.4469       | 1998-03-07 | FI_ACCOUNT | 7800.00 | NULL           | 108        | 100           |
| 115         | Alexander   | Khoo      | AKHOO   | 515.127.4562       | 1995-05-18 | PU_CLERK   | 3100.00 | NULL           | 114        | 30            |
| 153         | Christopher | Olsen     | COLSEN  | 011.44.1344.498718 | 1998-03-30 | SA_REP     | 8000.00 | 0.20           | 145        | 80            |
| 163         | Danielle    | Greene    | DGREENE | 011.44.1346.229268 | 1999-03-19 | SA_REP     | 9500.00 | 0.15           | 147        | 80            |

The bottom pane shows the 'Output' tab with 'Action Output' selected. It displays a list of actions and their messages:

| #  | Time     | Action   | Message  | Duration / Fetch      |
|----|----------|--|--|-----------------------|
| 50 | 22:53:24 | UPDATE employees substring(phone_number = REPLACE(phone_number,'124','999')) W...  | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 51 | 22:58:07 | SELECT job_id, GROUP_CONCAT(employee_id, ' ') 'Employees ID' FROM employees GRO... | 19 row(s) returned   | 0.016 sec / 0.000 sec |
| 52 | 23:04:38 | SELECT * FROM employees WHERE Len(first_name, >=8) ORDER BY first_name             | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 53 | 23:05:17 | SELECT * FROM employees WHERE first_name Len(first_name, >=8) ORDER BY first_name  | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 54 | 23:06:28 | SELECT * FROM employees WHERE length(first_name) >=8) ORDER BY first_name          | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 55 | 23:06:40 | SELECT * FROM employees WHERE length(first_name) >=8 LIMIT 0,1000                  | 11 row(s) returned   | 0.000 sec / 0.000 sec |

The status bar at the bottom indicates 'Query Completed'.

## 27. WRITE A QUERY TO APPEND '@EXAMPLE.COM' TO EMAIL FIELD

**UPDATE EMPLOYEES SET EMAIL = CONCAT(EMAIL, '@EXAMPLE.COM');**

**SELECT EMAIL FROM EMPLOYEES;**

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'SCHEMAS' tree with 'hr', 'jinh', 'newschema', 'northwind', and 'akila' schemas. The 'hr' schema is expanded, showing 'Tables', 'Views', 'Stored Procedures', and 'Functions'. The 'jinh' schema is also expanded. The 'newschema' schema is selected. The 'northwind' schema is selected. The 'akila' schema is selected. The 'Administration' tab is selected. The 'Schemas' tab is selected. The 'Information' pane shows 'No object selected'. The 'SQL File 1' window displays the following SQL script:

```
47 • UPDATE employees
48   SET phone_number = REPLACE('phone_number', '124', '999')
49   WHERE phone_number LIKE ('124');
50 • SELECT * FROM employees;
51 • UPDATE employees SET email = CONCAT(email, '@example.com');
52 • select email from employees;
```

The 'Result Grid' shows the results of the 'select email from employees' query. The results are as follows:

| email                |
|----------------------|
| SKING@example.com    |
| NKOCHHAR@example.com |
| LDEHAAN@example.com  |
| AHUNOLD@example.com  |
| BERNST@example.com   |

The 'Output' pane shows the 'Action Output' for the query execution. The results are as follows:

| #  | Time     | Action  | Message  | Duration / Fetch      |
|----|----------|---|--|-----------------------|
| 54 | 23:06:28 | SELECT * FROM employees WHERE length(first_name) >= 8 ORDER BY first_name | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 55 | 23:06:40 | SELECT * FROM employees WHERE length(first_name) >= 8 LIMIT 0, 1000       | 11 row(s) returned   | 0.000 sec / 0.000 sec |
| 56 | 23:09:58 | SELECT * FROM employees LIMIT 0, 1000                                     | 107 row(s) returned  | 0.015 sec / 0.000 sec |
| 57 | 23:13:20 | UPDATE employees SET email = CONCAT(email, '@example.com')                | 107 row(s) affected Rows matched: 107 Changed: 107 Warnings: 0                                 | 0.938 sec             |
| 58 | 23:17:37 | UPDATE employees SET email = CONCAT(email, '@example.com')                | Error Code: 1406. Data too long for column 'email' at row 1                                    | 0.047 sec             |
| 59 | 23:18:03 | select email from employees LIMIT 0, 1000                                 | 107 row(s) returned  | 0.000 sec / 0.000 sec |

The 'Query Completed' status is shown at the bottom left.

## 28. WRITE A QUERY TO EXTRACT THE LAST 4 CHARACTER OF PHONE NUMBERS

**SELECT RIGHT(phone\_number, 4) FROM employees;**

The screenshot displays the SQL Server Enterprise Manager interface. The left pane shows the 'SCHEMAS' tree with 'hr' and 'jinh' schemas expanded. The main pane shows a query window with the following SQL code:

```
48 SET phone_number = REPLACE('phone_number', '124', '999' )
49 WHERE phone_number LIKE ('124');
50 SELECT * FROM employees;
51 UPDATE employees SET email = CONCAT(email, '@example.com');
52 select email from employees;
53 SELECT RIGHT(phone_number, 4) FROM employees;
54
```

The 'Result Grid' shows the results of the query:

| RIGHT(phone_number, 4) |
|------------------------|
| 9878                   |
| 9879                   |
| 1876                   |
| 2876                   |
| ....                   |

The 'Output' pane shows the 'Action Output' table:

| #  | Time     | Action   | Message  | Duration / Fetch      |
|----|----------|--|--|-----------------------|
| 70 | 23:26:21 | SELECT SUBSTRING(phone_number, Right(4)) FROM employees                          | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 71 | 23:26:28 | SELECT SUBSTRING(phone_number, Right(4)) FROM employees                          | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 72 | 23:28:39 | SELECT SUBSTRING(phone_number) Right(4) FROM employees                           | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 73 | 23:28:52 | SELECT RIGHT(phone_number, 4) FROM employees LIMIT 0, 1000                       | 107 row(s) returned  | 0.000 sec / 0.000 sec |
| 74 | 23:29:33 | SELECT RIGHT(phone_number, 4) FROM employees ORDER BY phone_number LIMIT 0, 1... | 107 row(s) returned  | 0.015 sec / 0.000 sec |
| 75 | 23:29:49 | SELECT RIGHT(phone_number, 4) FROM employees LIMIT 0, 1000                       | 107 row(s) returned  | 0.000 sec / 0.000 sec |

The bottom status bar indicates 'Query Completed'.

## 29. WRITE A QUERY TO GET THE LAST WORD OF THE STREET ADDRESS

**SELECT \* FROM locations;**

**SELECT location\_id, street\_address,**

**SUBSTRING\_INDEX(REPLACE(REPLACE(REPLACE(street\_address, ',', ' '), ' '), ' '), ' ', -1)**

**AS 'Last--word-of-street\_address'**

**FROM locations;**

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```
54 • SHOW TABLES;
55 • SELECT * FROM locations;
56 • SELECT location_id, street_address,
57 • SUBSTRING_INDEX(REPLACE(REPLACE(REPLACE(street_address, ',', ' '), ' '), ' '), ' ', -1)
58 • AS 'Last--word-of-street_address'
59 • FROM locations;
```

The Results window displays the following data:

| location_id | street_address          | Last--word-of-street_address |
|-------------|-------------------------|------------------------------|
| 1000        | 1297 Via Cola di Rie    | Rie                          |
| 1100        | 93091 Calle della Testa | Testa                        |
| 1200        | 2017 Shinjuku-ku        | Shinjuku-ku                  |
| 1300        | 9450 Kamiya-cho         | Kamiya-cho                   |
| 1400        | 2014 Jabberwocky Rd     | Rd                           |

The Action Output window shows the following log:

| #  | Time     | Action   | Message             | Duration / Fetch      |
|----|----------|--|---------------------|-----------------------|
| 74 | 23:29:33 | SELECT RIGHT(phone_number, 4) FROM employees ORDER BY phone_number LIMIT 0, 1... | 107 row(s) returned | 0.015 sec / 0.000 sec |
| 75 | 23:29:49 | SELECT RIGHT(phone_number, 4) FROM employees LIMIT 0, 1000                       | 107 row(s) returned | 0.000 sec / 0.000 sec |
| 76 | 23:31:37 | SELECT * FROM employees LIMIT 0, 1000  | 107 row(s) returned | 0.016 sec / 0.000 sec |
| 77 | 23:31:56 | SHOW TABLES  | 8 row(s) returned   | 0.078 sec / 0.000 sec |
| 78 | 23:35:16 | SELECT * FROM locations LIMIT 0, 1000  | 23 row(s) returned  | 0.078 sec / 0.000 sec |
| 79 | 23:38:50 | SELECT location_id, street_address, SUBSTRING_INDEX(REPLACE(REPLACE(REPLACE(...  | 23 row(s) returned  | 0.000 sec / 0.000 sec |

### 30. WRITE A QUERY TO GET THE LOCATIONS THAT HAVE MINIMUM STREET LENGTH

**SELECT \* FROM locations WHERE LENGTH(STREET\_ADDRESS) <= (SELECT MIN(LENGTH(STREET\_ADDRESS)) FROM locations);**

The screenshot shows the SQL Developer interface with a query editor and a results grid. The query editor contains the following SQL code:

```
52 SUBSTRING_INDEX(REPLACE(REPLACE(REPLACE(street_address, ',' , ' '), ' ' , ' '), ' ' , ' '), '-1', -1)
53 AS 'Last--word-of-street_address'
54 FROM locations;
55 SELECT * FROM locations WHERE LENGTH(street_address) <= (SELECT MIN(LENGTH(street_address)) FROM locations);
56
57
58
59
```

The results grid displays the following data:

| location_id | street_address | postal_code | city            | state_province | country_id |
|-------------|----------------|-------------|-----------------|----------------|------------|
| 1600        | 2007 Zagora St | 50090       | South Brunswick | New Jersey     | US         |
| 2400        | 8204 Arthur St | NULL        | London          | NULL           | UK         |
| ...         | ...            | ...         | ...             | ...            | ...        |

The bottom panel shows the Action Output log with the following entries:

| #  | Time     | Action  | Message  | Duration / Fetch      |
|----|----------|---|--|-----------------------|
| 39 | 07:43:45 | SELECT FROM locations WHERE LENGTH(street_address) < (SELECT MIN(LENGTH(street... | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 40 | 07:44:18 | SELECT * FROM locations WHERE LENGTH(street_address) <= (SELECT MIN(LENGTH(str... | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 41 | 07:44:28 | SELECT * FROM locations WHERE LENGTH(street_address) <= (SELECT MIN(LENGTH(str... | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 42 | 07:44:54 | SELECT * FROM locations WHERE LENGTH(street_address) <= (SELECT MIN(LENGTH(st...  | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 43 | 07:45:12 | SELECT * FROM locations WHERE LENGTH(street_address) <= (SELECT MIN(LENGTH(st...  | 2 row(s) returned  | 0.000 sec / 0.000 sec |
| 44 | 07:45:48 | SELECT * FROM locations WHERE LENGTH(street_address) <= (SELECT MIN(LENGTH(str... | 2 row(s) returned  | 0.000 sec / 0.000 sec |

### 31. WRITE A QUERY TO DISPLAY THE FIRST WORD FROM THOSE JOB TITLES WHICH CONTAINS MORE THAN ONE WORDS

**SELECT \* FROM JOBS;**

**SELECT JOB\_TITLE, SUBSTR(JOB\_TITLE,1, INSTR(JOB\_TITLE, ' ')-1) FROM JOBS;**

The screenshot shows the SQL Developer interface with a query window titled 'Final Project working23RD.doc'. The query is as follows:

```
52 SUBSTRING_INDEX(REPLACE(REPLACE(REPLACE(street_address, ','), ' '), ' '), ' '), -1)
53 AS 'Last--word-of-street_address'
54 FROM locations;
55 • SELECT * FROM locations WHERE LENGTH(street_address) <= (SELECT MIN(LENGTH(street_address)) FROM locations);
56 • SELECT * FROM jobs;
57 • SELECT job_title, SUBSTR(job_title,1, INSTR(job_title, ' ')-1) FROM jobs;
58
```

The 'Result Grid' shows the following data:

| job_title                | SUBSTR(job_title,1, INSTR(job_title, ' ')-1) |
|--------------------------|--|
| Public Accountant        | Public                                       |
| Accounting Manager       | Accounting                                   |
| Administration Assistant | Administration                               |
| President                |  |

The 'Output' window shows the execution log:

| #  | Time     | Action   | Message  | Duration / Fetch      |
|----|----------|--|--|-----------------------|
| 42 | 07:44:54 | SELECT * FROM locations WHERE LENGTH(street_address) <= (SELECT MIN(LENGTH(st...       | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 43 | 07:45:12 | SELECT * FROM locations WHERE LENGTH(street_address) <= (SELECT MIN(LENGTH(st...       | 2 row(s) returned  | 0.000 sec / 0.000 sec |
| 44 | 07:45:48 | SELECT * FROM locations WHERE LENGTH(street_address) <= (SELECT MIN(LENGTH(st...       | 2 row(s) returned  | 0.000 sec / 0.000 sec |
| 45 | 07:47:50 | SELECT * FROM jobs LIMIT 0, 1000   | 19 row(s) returned   | 0.000 sec / 0.000 sec |
| 46 | 07:52:57 | SELECT job_title WHERE SUBSTR(job_title,1, INSTR(job_title, ' ')-1) FROM jobs          | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 47 | 07:53:31 | SELECT job_title, SUBSTR(job_title,1, INSTR(job_title, ' ')-1) FROM jobs LIMIT 0, 1000 | 19 row(s) returned   | 0.031 sec / 0.000 sec |

## 32. WRITE A QUERY TO DISPLAY THE LENGTH OF FIRST NAME FOR EMPLOYEES WHERE LAST NAME CONTAIN CHARACTER 'C' AFTER 2ND POSITION

**SELECT FIRST\_NAME,  
LAST\_NAME FROM EMPLOYEES WHERE INSTR(LAST\_NAME,'C') > 2;**

The screenshot shows the SQL Developer interface with a query window titled 'Final Project working:23RD dec'. The query is as follows:

```
54 FROM locations;
55 • SELECT * FROM locations WHERE LENGTH(street_address) <= (SELECT MIN(LENGTH(street_address)) FROM locations);
56 • SELECT * FROM jobs;
57 • SELECT job_title, SUBSTR(job_title,1, INSTR(job_title, '-')-1) FROM jobs;
58 • SELECT * FROM employees;
59 • SELECT first_name,
60 last_name FROM employees WHERE INSTR(last_name,'C') > 2;
61
```

The 'Result Grid' shows the following data:

| first_name | last_name |
|------------|-----------|
| Neena      | Kochhar   |
| Peter      | Tucker    |
| Nandita    | Sarchand  |

The 'Output' pane shows the execution log with the following entries:

| #  | Time     | Action   | Message   | Duration / Fetch      |
|----|----------|--|---|-----------------------|
| 45 | 07:47:50 | SELECT * FROM jobs LIMIT 0, 1000   | 19 row(s) returned  | 0.000 sec / 0.000 sec |
| 46 | 07:52:57 | SELECT job_title WHERE SUBSTR(job_title,1, INSTR(job_title, '-')-1) FROM jobs          | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to... | 0.000 sec             |
| 47 | 07:53:31 | SELECT job_title, SUBSTR(job_title,1, INSTR(job_title, '-')-1) FROM jobs LIMIT 0, 1000 | 19 row(s) returned  | 0.031 sec / 0.000 sec |
| 48 | 07:57:32 | SELECT employees LIMIT 0, 1000   | Error Code: 1054. Unknown column 'employees' in 'field list'                                    | 0.016 sec             |
| 49 | 07:57:46 | SELECT * FROM employees LIMIT 0, 1000  | 107 row(s) returned   | 0.062 sec / 0.000 sec |
| 50 | 07:58:50 | SELECT first_name, last_name FROM employees WHERE INSTR(last_name,'C') > 2 LIMIT 0...  | 3 row(s) returned   | 0.000 sec / 0.000 sec |



**33. WRITE A QUERY THAT DISPLAYS THE FIRST NAME AND THE LENGTH OF THE FIRST NAME FOR ALL EMPLOYEES WHOSE NAME STARTS WITH THE LETTERS 'A', 'J' OR 'M'. GIVE EACH COLUMN AN APPROPRIATE LABEL. SORT THE RESULTS BY THE EMPLOYEES' FIRST NAMES**

```
SELECT first_name AS "Name",  
  
LENGTH(first_name) AS "Length"  
  
FROM employees  
  
WHERE first_name LIKE 'J%'  
  
OR first_name LIKE 'M%'  
  
OR first_name LIKE 'A%'  
  
ORDER BY first_name ;
```

The screenshot shows the SQL Developer interface. The query editor contains the following SQL query:

```
64 SELECT first_name as "Name",  
65 LENGTH(first_name) as "Length"  
66 FROM employees  
67 WHERE first_name LIKE 'J%'  
68 OR first_name LIKE 'M%'  
69 OR first_name LIKE 'A%'  
70 ORDER BY first_name ;  
71
```

The Results pane displays the following data:

| Name      | Length |
|-----------|--------|
| Adam      | 4      |
| Alana     | 5      |
| Alberto   | 7      |
| Alexander | 9      |
| Alexander | 9      |

The Output pane shows the execution log with the following entries:

| #  | Time     | Action   | Message             | Duration / Fetch      |
|----|----------|--|---------------------|-----------------------|
| 51 | 08:01:35 | SELECT first_name, LENGTH(first_name) FROM employees WHERE first_name LIKE 'J%' O... | 32 row(s) returned  | 0.000 sec / 0.000 sec |
| 52 | 08:01:52 | SELECT first_name "Name", LENGTH(first_name) "Length" FROM employees WHERE first_... | 32 row(s) returned  | 0.000 sec / 0.000 sec |
| 53 | 08:02:01 | SELECT first_name, LENGTH(first_name) FROM employees WHERE first_name LIKE 'J%' O... | 32 row(s) returned  | 0.000 sec / 0.000 sec |
| 54 | 08:02:38 | SELECT first_name as "Name", LENGTH(first_name) as "Length" FROM employees WHER...   | 32 row(s) returned  | 0.000 sec / 0.000 sec |
| 55 | 08:03:13 | SELECT first_name, LENGTH(first_name) FROM employees LIMIT 0, 1000                   | 107 row(s) returned | 0.000 sec / 0.000 sec |
| 56 | 08:03:42 | SELECT first_name as "Name", LENGTH(first_name) as "Length" FROM employees WHER...   | 32 row(s) returned  | 0.000 sec / 0.000 sec |

**34. WRITE A QUERY TO DISPLAY THE FIRST NAME AND SALARY FOR ALL EMPLOYEES. FORMAT THE SALARY TO BE 10 CHARACTERS LONG, LEFT-PADDED WITH THE \$ SYMBOL. LABEL THE COLUMN SALARY**

**SELECT FIRST\_NAME,**

**LPAD(SALARY, 10, '\$') SALARY**

**FROM EMPLOYEES;**

The screenshot shows the SQL Developer interface. The left pane displays the 'SCHEMAS' tree with 'hr' and 'jimpjh' schemas expanded. The main editor shows a SQL query with line numbers 77 to 83. The query is as follows:

```
77 • SELECT employee_id,first_name,last_name,hire_date
78 FROM employees
79 WHERE POSITION("07" IN DATE_FORMAT(hire_date, '%d %m %Y'))>0;
80 • SELECT first_name,
81 LPAD(salary, 10, '$') SALARY
82 FROM employees;
83
```

Below the editor, the 'Result Grid' shows the output of the query. It has two columns: 'first\_name' and 'SALARY'. The data is as follows:

| first_name | SALARY        |
|------------|---------------|
| Steven     | \$\$24000.00  |
| Neena      | \$\$17000.00  |
| Lex        | \$\$17000.00  |
| Alexander  | \$\$\$9000.00 |
| Bruce      | \$\$\$6000.00 |

The bottom pane shows the 'Output' tab with 'Action Output' selected. It displays a log of SQL actions and their results:

| #  | Time     | Action  | Message  | Duration / Fetch      |
|----|----------|---|--|-----------------------|
| 45 | 18:34:14 | SELECT first_name,LENGTH(first_name) FROM employees LIMIT 0, 1000                         | 107 row(s) returned  | 0.000 sec / 0.000 sec |
| 46 | 21:37:14 | SELECT left(first_name,8),REPEAT('\$',FLOOR(salary/1000)) 'SALARY(\$)', salary FROM em... | 107 row(s) returned  | 0.047 sec / 0.000 sec |
| 47 | 21:45:48 | SELECT * FROM employees LIMIT 0, 1000   | 107 row(s) returned  | 0.000 sec / 0.000 sec |
| 48 | 21:45:54 | SELECT employee_id,first_name,last_name,hire_date FROM employees WHERE POSITIO...         | 16 row(s) returned   | 0.031 sec / 0.000 sec |
| 49 | 21:59:37 | SELECT first_name,LPAD(salary, 10, '\$') ORDER BY DESC SALARY FROM employees              | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 50 | 22:00:30 | SELECT first_name,LPAD(salary, 10, '\$') SALARY FROM employees LIMIT 0, 1000              | 107 row(s) returned  | 0.000 sec / 0.000 sec |

The status bar at the bottom indicates 'Query Completed'.

**35. WRITE A QUERY TO DISPLAY THE FIRST EIGHT CHARACTERS OF THE EMPLOYEES' FIRST NAMES AND INDICATES THE AMOUNTS OF THEIR SALARIES WITH '\$' SIGN. EACH '\$' SIGN SIGNIFIES A THOUSAND DOLLARS. SORT THE DATA IN DESCENDING ORDER OF SALARY**

```
SELECT LEFT(FIRST_NAME, 8),  
REPEAT('$', FLOOR(SALARY/1000))  
'SALARY($)', SALARY  
FROM EMPLOYEES  
ORDER BY SALARY DESC;
```

The screenshot shows the SQL Developer interface. The SQL Editor window contains the following query:

```
73 ORDER BY DESC SALARY FROM employees;  
74 SELECT left(first_name, 8),  
75 REPEAT('$', FLOOR(salary/1000))  
76 'SALARY($)', salary  
77 FROM employees  
78 ORDER BY salary DESC;  
79
```

The Results window displays the query results in a grid format:

| left(first_name, 8) | SALARY(\$) | salary   |
|---------------------|------------|----------|
| Steven              | #####      | 24000.00 |
| Neena               | #####      | 17000.00 |
| Lex                 | #####      | 17000.00 |
| John                | #####      | 14000.00 |
| ...                 | .....      | .....    |

The Output window shows the execution log with the following entries:

| #  | Time     | Action   | Message             | Duration / Fetch      |
|----|----------|--|---------------------|-----------------------|
| 41 | 18:33:52 | SELECT job_title, SUBSTR(job_title, INSTR(job_title, '-')-1) FROM jobs LIMIT 0, 1000         | 19 row(s) returned  | 0.000 sec / 0.000 sec |
| 42 | 18:33:53 | SELECT * FROM employees LIMIT 0, 1000  | 107 row(s) returned | 0.015 sec / 0.000 sec |
| 43 | 18:33:54 | SELECT first_name, last_name FROM employees WHERE INSTR(last_name, 'C') > 2 LIMIT 0...       | 3 row(s) returned   | 0.000 sec / 0.000 sec |
| 44 | 18:34:08 | SELECT first_name, last_name FROM employees WHERE INSTR(last_name, 'C') > 2 LIMIT 0...       | 3 row(s) returned   | 0.000 sec / 0.000 sec |
| 45 | 18:34:14 | SELECT first_name, LENGTH(first_name) FROM employees LIMIT 0, 1000                           | 107 row(s) returned | 0.000 sec / 0.000 sec |
| 46 | 21:37:14 | SELECT left(first_name, 8), REPEAT('\$', FLOOR(salary/1000)) 'SALARY(\$)', salary FROM em... | 107 row(s) returned | 0.047 sec / 0.000 sec |

The status bar at the bottom indicates "Query Completed".

**36. WRITE A QUERY TO DISPLAY THE EMPLOYEES WITH THEIR CODE, FIRST NAME, LAST NAME AND HIRE DATE WHO HIRED EITHER ON SEVENTH DAY OF ANY MONTH OR SEVENTH MONTH IN ANY YEAR**

**SELECT \* FROM EMPLOYEES;**

**SELECT EMPLOYEE\_ID,FIRST\_NAME,LAST\_NAME,HIRE\_DATE**

**FROM EMPLOYEES**

**WHERE POSITION("07" IN DATE\_FORMAT(HIRE\_DATE, '%D %M %Y'))>0;**

The screenshot shows a SQL IDE interface with a query editor, a result grid, and an action output pane.

**Query Editor:**

```
78 ORDER BY salary DESC;
79 • SELECT * FROM employees;
80 • SELECT employee_id,first_name,last_name,hire_date
81 FROM employees
82 WHERE POSITION("07" IN DATE_FORMAT(hire_date, '%d %m %Y'))>0;
83
84
```

**Result Grid:**

| employee_id | first_name  | last_name | hire_date  |
|-------------|-------------|-----------|------------|
| 107         | Diana       | Lorentz   | 1999-02-07 |
| 112         | Jose Manuel | Urman     | 1998-03-07 |
| 113         | Luis        | Popp      | 1999-12-07 |
| 114         | Den         | Raphaely  | 1994-12-07 |
| 117         | Sigal       | Tobias    | 1997-07-24 |

**Action Output:**

| #  | Time     | Action  | Message             | Duration / Fetch      |
|----|----------|---|---------------------|-----------------------|
| 43 | 18:33:54 | SELECT first_name, last_name FROM employees WHERE INSTR(last_name,'C') > 2 LIMIT 0...       | 3 row(s) returned   | 0.000 sec / 0.000 sec |
| 44 | 18:34:08 | SELECT first_name, last_name FROM employees WHERE INSTR(last_name,'C') > 2 LIMIT 0...       | 3 row(s) returned   | 0.000 sec / 0.000 sec |
| 45 | 18:34:14 | SELECT first_name, LENGTH(first_name) FROM employees LIMIT 0, 1000                          | 107 row(s) returned | 0.000 sec / 0.000 sec |
| 46 | 21:37:14 | SELECT left(first_name, 8), REPEAT('\$',FLOOR(salary/1000)) 'SALARY(\$)', salary FROM em... | 107 row(s) returned | 0.047 sec / 0.000 sec |
| 47 | 21:45:48 | SELECT * FROM employees LIMIT 0, 1000   | 107 row(s) returned | 0.000 sec / 0.000 sec |
| 48 | 21:45:54 | SELECT employee_id,first_name,last_name,hire_date FROM employees WHERE POSITIO...           | 16 row(s) returned  | 0.031 sec / 0.000 sec |

Query Completed

# NORTHWIND DATABASE EXERCISES

## 1) WRITE A QUERY TO GET PRODUCT NAME AND QUANTITY/UNIT

**USE NORTHWIND;**

**SHOW TABLES;**

**SELECT \* FROM PRODUCTS;**

**SELECT ProductName, QuantityPerUnit FROM PRODUCTS;**

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the 'SCHEMAS' tree with the 'northwind' database selected. The central pane shows a query window with the following SQL statements:

```
1 • USE northwind;
2 • SHOW TABLES;
3 • SELECT * FROM products;
4 • SELECT ProductName, QuantityPerUnit FROM products;
```

The 'Result Grid' at the bottom displays the results of the fourth query, showing a list of products with their names and quantities per unit:

| ProductName                  | QuantityPerUnit     |
|------------------------------|---------------------|
| Chai                         | 10 boxes x 20 bags  |
| Chang                        | 24 - 12 oz bottles  |
| Aniseed Syrup                | 12 - 550 ml bottles |
| Chef Anton's Cajun Seasoning | 48 - 6 oz jars      |
| Chef Anton's Gumbo Mix       | 36 boxes            |

The bottom pane shows the 'Output' window with a table of query execution results:

| #  | Time     | Action  | Message   | Duration / Fetch      |
|----|----------|---|---|-----------------------|
| 5  | 18:04:30 | SELECT * FROM order details                                     | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to... | 0.140 sec             |
| 6  | 18:04:39 | SELECT * FROM order details                                     | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to... | 0.000 sec             |
| 7  | 18:05:59 | SELECT * FROM products LIMIT 0, 1000                            | Error Code: 1146. Table 'northwind.products' doesn't exist                                      | 0.391 sec             |
| 8  | 18:06:02 | SELECT * FROM products LIMIT 0, 1000                            | 77 row(s) returned  | 0.125 sec / 0.000 sec |
| 9  | 18:07:44 | SELECT ProductName, QuantityPerUnit * FROM products             | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to... | 0.000 sec             |
| 10 | 18:08:22 | SELECT ProductName, QuantityPerUnit FROM products LIMIT 0, 1000 | 77 row(s) returned  | 0.000 sec / 0.000 sec |

## 2. WRITE A QUERY TO GET CURRENT PRODUCT LIST (PRODUCT ID AND NAME)

**SELECT ProductID, ProductName FROM products;**

The screenshot displays the SQL Server Enterprise Manager interface. The left pane shows the 'SCHEMAS' tree with the 'northwind' database selected. The central pane shows a query window with the following SQL script:

```
1 USE northwind;
2 SHOW TABLES;
3 SELECT * FROM products;
4 SELECT ProductName, QuantityPerUnit FROM products;
5 SELECT ProductID, ProductName FROM products;
```

The 'Result Grid' tab is active, showing the results of the last query. The results are as follows:

| ProductID | ProductName                  |
|-----------|------------------------------|
| 1         | Chai                         |
| 2         | Chang                        |
| 3         | Aniseed Syrup                |
| 4         | Chef Anton's Cajun Seasoning |
| 5         | Chef Anton's Gumbo Mix       |

The 'Output' pane at the bottom shows the execution log with the following entries:

| #  | Time     | Action  | Message   | Duration / Fetch      |
|----|----------|---|---|-----------------------|
| 7  | 18:05:59 | SELECT * FROM product LIMIT 0, 1000                             | Error Code: 1146. Table 'northwind.products' doesn't exist                                      | 0.391 sec             |
| 8  | 18:06:02 | SELECT * FROM products LIMIT 0, 1000                            | 77 row(s) returned  | 0.125 sec / 0.000 sec |
| 9  | 18:07:44 | SELECT ProductName, QuantityPerUnit * FROM products             | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to... | 0.000 sec             |
| 10 | 18:08:22 | SELECT ProductName, QuantityPerUnit FROM products LIMIT 0, 1000 | 77 row(s) returned  | 0.000 sec / 0.000 sec |
| 11 | 18:10:18 | SELECT * FROM products LIMIT 0, 1000                            | 77 row(s) returned  | 0.000 sec / 0.000 sec |
| 12 | 18:11:21 | SELECT ProductID, ProductName FROM products LIMIT 0, 1000       | 77 row(s) returned  | 0.000 sec / 0.000 sec |

The status bar at the bottom indicates 'Query Completed'.

### 3. WRITE A QUERY TO GET DISCONTINUED PRODUCT LIST (PRODUCT ID AND NAME)

**SELECT PRODUCTID, PRODUCTNAME**

**FROM PRODUCTS**

**WHERE DISCONTINUED = 1;**

The screenshot displays the SQL Server Enterprise Manager interface. The query editor at the top contains the following SQL query:

```
3 • SELECT * FROM products;
4 • SELECT ProductName, QuantityPerUnit FROM products;
5 • SELECT ProductID, ProductName FROM products;
6 • SELECT ProductID, ProductName
7 FROM Products
8 WHERE Discontinued = 1;
9
```

Below the query editor, the 'Result Grid' shows the results of the query. The grid has two columns: 'ProductID' and 'ProductName'. The results are as follows:

| ProductID | ProductName            |
|-----------|------------------------|
| 5         | Chef Anton's Gumbo Mix |
| 9         | Mishi Kobe Niku        |
| 17        | Alice Mutton           |
| 24        | Guaraná Fantástica     |
| 28        | Rössle Sauerkraut      |

Below the result grid, the 'Output' pane shows the execution log. The log contains the following entries:

| #    | Time     | Action  | Message  | Duration / Fetch      |
|------|----------|---|--|-----------------------|
| ✓ 16 | 18:14:26 | SELECT ProductID, ProductName FROM products LIMIT 0, 1000                         | 77 row(s) returned   | 0.000 sec / 0.000 sec |
| ✓ 17 | 18:18:30 | SELECT ProductID, ProductName FROM Products WHERE discontinued = 1 ORDER BY Pr... | 8 row(s) returned  | 0.015 sec / 0.000 sec |
| ✓ 18 | 18:18:39 | SELECT ProductID, ProductName FROM Products WHERE Discontinued = 1 LIMIT 0, 1000  | 8 row(s) returned  | 0.000 sec / 0.000 sec |
| ✗ 19 | 18:19:21 | FROM Products   | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| ✗ 20 | 18:19:31 | SELECT ProductID, ProductName WHERE Discontinued = 1 FROM Products                | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.016 sec             |
| ✓ 21 | 18:19:44 | SELECT ProductID, ProductName FROM Products WHERE Discontinued = 1 LIMIT 0, 1000  | 8 row(s) returned  | 0.016 sec / 0.000 sec |

#### 4. WRITE A QUERY TO GET MOST EXPENSE AND LEAST EXPENSIVE PRODUCT LIST (NAME AND UNIT PRICE)

**SELECT PRODUCTNAME, UNITPRICE**

**FROM PRODUCTS**

**ORDER BY UNITPRICE DESC;**

The screenshot shows the SQL Server Enterprise Manager interface. The query window contains the following SQL code:

```
6 • SELECT ProductID, ProductName
7 FROM Products
8 WHERE Discontinued = 1;
9 ✖ SELECT ProductName, UnitPrice Max(UnitPrice) , Min(UnitPrice) FROM Products;
10 • SELECT ProductName, UnitPrice
11 FROM Products
12 ORDER BY UnitPrice DESC;
13
```

Below the query window, the Results pane shows a table with the following data:

| ProductName             | UnitPrice |
|-------------------------|-----------|
| Côte de Blaye           | 263.5000  |
| Thüringer Rostbratwurst | 123.7900  |
| Mishi Kobe Niku         | 97.0000   |
| Sir Rodney's Marmalade  | 81.0000   |
| Carnarvon Tigers        | 62.5000   |

The bottom pane shows the Action Output window with the following log:

| # | Time        | Action   | Message  | Duration / Fetch      |
|---|-------------|--|--|-----------------------|
| ✖ | 25 18:30:39 | SELECT ProductName, UnitPrice WHERE Max(UnitPrice) , Min(UnitPrice) FROM Products      | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| ✖ | 26 18:31:02 | SELECT ProductName, UnitPrice Max(UnitPrice) , Min(UnitPrice) FROM Products            | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| ✓ | 27 18:31:24 | SELECT Max(UnitPrice) , Min(UnitPrice) FROM Products LIMIT 0, 1000                     | 1 row(s) returned  | 0.000 sec / 0.000 sec |
| ✓ | 28 18:32:05 | SELECT Max(UnitPrice) , Min(UnitPrice) FROM Products ORDER BY ProductName, UnitPric... | 1 row(s) returned  | 0.016 sec / 0.000 sec |
| ✖ | 29 18:35:14 | SELECT ProductName, UnitPrice Max(UnitPrice) , Min(UnitPrice) FROM Products            | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| ✓ | 30 18:35:45 | SELECT ProductName, UnitPrice FROM Products ORDER BY UnitPrice DESC LIMIT 0, 1000      | 77 row(s) returned   | 0.000 sec / 0.000 sec |



## 5. WRITE A QUERY TO GET PRODUCT LIST (ID, NAME, UNIT PRICE) WHERE CURRENT PRODUCTS COST LESS THAN \$20

**SELECT ProductID, ProductName, UnitPrice FROM Products WHERE (UnitPrice<20) AND ('Discontinued' = False) ORDER BY UnitPrice DESC;**

The screenshot shows a database query editor with a SQL query and its results. The query is as follows:

```
7 FROM Products
8 WHERE Discontinued = 1;
9 SELECT ProductName, UnitPrice Max(UnitPrice) , Min(UnitPrice) FROM Products;
10 SELECT ProductName, UnitPrice
11 FROM Products
12 ORDER BY UnitPrice DESC;
13 SELECT ProductID, ProductName, UnitPrice WHERE UnitPrice<20 * FROM Products;
14 SELECT ProductID, ProductName, UnitPrice FROM Products WHERE (UnitPrice<20) AND ('Discontinued' = False) ORDER BY UnitPrice DESC
```

The results are displayed in a table with the following columns: ProductID, ProductName, and UnitPrice. The table contains 5 rows of data:

| ProductID | ProductName      | UnitPrice |
|-----------|------------------|-----------|
| 57        | Ravioli Angelo   | 19.5000   |
| 44        | Gula Malacca     | 19.4500   |
| 2         | Chang            | 19.0000   |
| 36        | Inlagd Sill      | 19.0000   |
| 40        | Boston Crab Meat | 18.4000   |

The bottom section of the screenshot shows the 'Action Output' table, which contains the following rows:

| #  | Time     | Action   | Message   | Duration / Fetch      |
|----|----------|--|---|-----------------------|
| 37 | 18:56:55 | SELECT * FROM products LIMIT 0, 1000   | 77 row(s) returned  | 0.016 sec / 0.000 sec |
| 38 | 19:03:59 | SELECT ProductID, ProductName, UnitPrice FROM Products WHERE (UnitPrice<20) AND (...)  | Error Code: 1054. Unknown column 'Discontinued' in 'where clause' | 0.000 sec             |
| 39 | 19:04:17 | SELECT ProductID, ProductName, UnitPrice FROM Products WHERE (UnitPrice<20) AND (...)  | 39 row(s) returned  | 0.078 sec / 0.000 sec |
| 40 | 19:04:24 | SELECT ProductID, ProductName, UnitPrice FROM Products WHERE ((UnitPrice<20) AND (...) | 37 row(s) returned  | 0.000 sec / 0.000 sec |
| 41 | 19:04:31 | SELECT ProductID, ProductName, UnitPrice FROM Products WHERE (UnitPrice<20) AND (...)  | 39 row(s) returned  | 0.031 sec / 0.000 sec |
| 42 | 19:04:42 | SELECT ProductID, ProductName, UnitPrice FROM Products WHERE (UnitPrice<20) AND (...)  | 39 row(s) returned  | 0.000 sec / 0.000 sec |

Limit to 1000 rows

```

7 FROM Products
8 WHERE Discontinued = 1;
9 SELECT ProductName, UnitPrice, Max(UnitPrice), Min(UnitPrice) FROM Products;
10 SELECT ProductName, UnitPrice
11 FROM Products
12 ORDER BY UnitPrice DESC;
13 SELECT ProductID, ProductName, UnitPrice WHERE UnitPrice < 20 FROM Products;
14 SELECT ProductID, ProductName, UnitPrice FROM Products WHERE (UnitPrice < 20) AND ('Discontinued' = False) ORDER BY UnitPrice DESC

```

**Result Grid** | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | **Result Grid**

|   | ProductID | ProductName      | UnitPrice |
|---|-----------|------------------|-----------|
| ▶ | 57        | Ravioli Angelo   | 19.5000   |
|   | 44        | Gula Malacca     | 19.4500   |
|   | 2         | Chang            | 19.0000   |
|   | 36        | Inlagd Sill      | 19.0000   |
|   | 40        | Boston Crab Meat | 18.4000   |

Products 25 x Apply

Output

Action Output

| #    | Time     | Action   | Message   | Duration / Fetch      |
|------|----------|--|---|-----------------------|
| ✓ 37 | 18:56:55 | SELECT * FROM products LIMIT 0, 1000   | 77 row(s) returned  | 0.016 sec / 0.000 sec |
| ✗ 38 | 19:03:59 | SELECT ProductID, ProductName, UnitPrice FROM Products WHERE (UnitPrice < 20) AND (...)  | Error Code: 1054. Unknown column 'Discontinued' in 'where clause' | 0.000 sec             |
| ✓ 39 | 19:04:17 | SELECT ProductID, ProductName, UnitPrice FROM Products WHERE (UnitPrice < 20) AND (...)  | 39 row(s) returned  | 0.078 sec / 0.000 sec |
| ✓ 40 | 19:04:24 | SELECT ProductID, ProductName, UnitPrice FROM Products WHERE ((UnitPrice < 20) AND (...) | 37 row(s) returned  | 0.000 sec / 0.000 sec |
| ✓ 41 | 19:04:31 | SELECT ProductID, ProductName, UnitPrice FROM Products WHERE (UnitPrice < 20) AND (...)  | 39 row(s) returned  | 0.031 sec / 0.000 sec |
| ✓ 42 | 19:04:42 | SELECT ProductID, ProductName, UnitPrice FROM Products WHERE (UnitPrice < 20) AND (...)  | 39 row(s) returned  | 0.000 sec / 0.000 sec |

6. WRITE A QUERY TO GET PRODUCT LIST (ID, NAME, UNIT PRICE) WHERE PRODUCTS COST BETWEEN \$15 AND \$25

SELECT PRODUCTID, PRODUCTNAME, UNITPRICE

FROM PRODUCTS

WHERE UNITPRICE BETWEEN 15 AND 25;

The screenshot displays the SQL Server Enterprise Manager interface. The top pane shows a SQL query in the 'SQL File 1' window. The query is as follows:

```
10 SELECT ProductName, UnitPrice
11 FROM Products
12 ORDER BY UnitPrice DESC;
13 SELECT ProductID, ProductName, UnitPrice WHERE UnitPrice < 20 * FROM Products;
14 SELECT ProductID, ProductName, UnitPrice FROM Products WHERE (UnitPrice < 20) AND ('Discontinued' = False) ORDER BY UnitPrice DESC;
15 SELECT ProductID, ProductName, UnitPrice
16 FROM products
17 WHERE UnitPrice BETWEEN 15 AND 25;
```

The bottom pane shows the 'Result Grid' with the following data:

| ProductID | ProductName                  | UnitPrice |
|-----------|------------------------------|-----------|
| 1         | Chai                         | 18.0000   |
| 2         | Chang                        | 19.0000   |
| 4         | Chef Anton's Cajun Seasoning | 22.0000   |
| 5         | Chef Anton's Gumbo Mix       | 21.3500   |
| 6         | Grandma's Boysenberry Spread | 25.0000   |

Below the result grid, the 'Output' pane shows the 'Action Output' table, which includes the execution time, action, message, and duration for each query execution.

| #  | Time     | Action   | Message  | Duration / Fetch      |
|----|----------|--|--|-----------------------|
| 48 | 19:24:13 | SELECT ProductID, ProductName, UnitPrice FROM products WHERE UnitPrice BETWEEN...        | 25 row(s) returned   | 0.016 sec / 0.000 sec |
| 49 | 19:24:43 | SELECT ProductName, UnitPrice FROM Products WHERE (((UnitPrice) >= 15 And (UnitPrice)... | 24 row(s) returned   | 0.016 sec / 0.000 sec |
| 50 | 19:24:55 | SELECT ProductID, ProductName, UnitPrice FROM products WHERE UnitPrice BETWEEN...        | 25 row(s) returned   | 0.000 sec / 0.000 sec |
| 51 | 19:25:32 | SELECT ProductID, ProductName, UnitPrice FROM products WHERE UnitPrice BETWEEN...        | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.031 sec             |
| 52 | 19:25:36 | SELECT ProductID, ProductName, UnitPrice FROM products WHERE UnitPrice BETWEEN...        | 25 row(s) returned   | 0.000 sec / 0.000 sec |
| 53 | 19:25:55 | SELECT ProductID, ProductName, UnitPrice FROM products WHERE UnitPrice BETWEEN...        | 25 row(s) returned   | 0.000 sec / 0.000 sec |

## 7. WRITE A QUERY TO GET PRODUCT LIST (NAME, UNIT PRICE) OF ABOVE AVERAGE PRICE

**SELECT DISTINCT PRODUCTNAME, UNITPRICE FROM PRODUCTS**

**WHERE UNITPRICE > (SELECT AVG(UNITPRICE) FROM PRODUCTS)**

**ORDER BY UNITPRICE;**

The screenshot shows a SQL IDE interface. The top pane contains a SQL query with line numbers 13 to 20. The query is as follows:

```
13 FROM PRODUCTS
14 ORDER BY UNITPRICE DESC;
15 • SELECT PRODUCTID, PRODUCTNAME, UNITPRICE FROM PRODUCTS WHERE (UNITPRICE < 20) AND ('DISCONTINUED' = FALSE) ORDER BY UNITPRICE DESC;
16 ❌ SELECT ProductName, UnitPrice WHERE (UnitPrice > AveragePrice) FROM PRODUCTS;
17 • SELECT DISTINCT ProductName, UnitPrice FROM products
18 WHERE UnitPrice > (SELECT AVG(UnitPrice) FROM Products)
19 ORDER BY UnitPrice;
20
```

The bottom pane shows the 'Result Grid' with a table of product names and unit prices:

| ProductName                     | UnitPrice |
|---------------------------------|-----------|
| Uncle Bob's Organic Dried Pears | 30.0000   |
| Ikura                           | 31.0000   |
| Gumbär Gummibärchen             | 31.2300   |
| Mascarpone Fabioli              | 32.0000   |
| Perth Pasties                   | 32.8000   |

Below the result grid is an 'Output' pane showing a table of execution actions and messages:

| #  | Time     | Action  | Message  | Duration / Fetch      |
|----|----------|---|--|-----------------------|
| 5  | 09:51:28 | SELECT ProductName, UnitPrice WHERE (UnitPrice > AveragePrice) FROM PRODUCTS          | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.047 sec             |
| 6  | 09:55:14 | SELECT DISTINCT ProductName, UnitPrice WHERE UnitPrice > AVG(UnitPrice) FROM Produ... | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 7  | 09:55:49 | SELECT DISTINCT ProductName, UnitPrice WHERE UnitPrice > AVG(UnitPrice) FROM Produ... | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 8  | 09:56:33 | SELECT DISTINCT ProductName, UnitPrice WHERE UnitPrice > (SELECT AVG(UnitPrice) F...  | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 9  | 09:56:57 | SELECT DISTINCT ProductName, UnitPrice WHERE UnitPrice > (SELECT AVG(UnitPrice) F...  | Error Code: 1054. Unknown column 'ProductName' in 'field list'                                 | 0.000 sec             |
| 10 | 09:57:21 | SELECT DISTINCT ProductName, UnitPrice FROM products WHERE UnitPrice > (SELECT ...    | 25 row(s) returned   | 0.031 sec / 0.000 sec |

## 8. WRITE A QUERY TO GET PRODUCT LIST (NAME, UNIT PRICE) OF TEN MOST EXPENSIVE PRODUCTS

**SELECT DISTINCT PRODUCTNAME, UNITPRICE FROM PRODUCTS**

**WHERE COUNT(UNITPRICE) >=10 ORDER BY UNITPRICE;**

The screenshot shows the SQL Server Enterprise Manager interface. The query editor contains the following SQL code:

```
8 • SELECT PRODUCTID, PRODUCTNAME
9 FROM PRODUCTS
10 WHERE DISCONTINUED = 1;
11
12 • SELECT PRODUCTNAME, UNITPRICE
13 FROM PRODUCTS
14 ORDER BY UNITPRICE DESC;
15 • SELECT PRODUCTID, PRODUCTNAME, UNITPRICE FROM PRODUCTS WHERE (UNITPRICE<20) AND ('DISCONTINUED' = FALSE) ORDER BY UNITPRICE DESC;
16 SELECT ProductName, UnitPrice WHERE (UnitPrice >AveragePrice) FROM PRODUCTS;
17 • SELECT DISTINCT ProductName, UnitPrice FROM products
18 WHERE UnitPrice> (SELECT AVG(UnitPrice) From Products)
19 ORDER BY UnitPrice;
20 • SELECT DISTINCT ProductName, UnitPrice FROM Products
21 WHERE COUNT(UnitPrice) >=10 ORDER BY UnitPrice;
22
```

The output window shows the following execution results:

| #  | Time     | Action   | Message  | Duration / Fetch      |
|----|----------|--|--|-----------------------|
| 6  | 09:55:14 | SELECT DISTINCT ProductName, UnitPrice WHERE UnitPrice> AVG(UnitPrice) From Produ... | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 7  | 09:55:49 | SELECT DISTINCT ProductName, UnitPrice WHERE UnitPrice> AVG(UnitPrice) From Produ... | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 8  | 09:56:33 | SELECT DISTINCT ProductName, UnitPrice WHERE UnitPrice> (SELECT AVG(UnitPrice) F...  | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds t... | 0.000 sec             |
| 9  | 09:56:57 | SELECT DISTINCT ProductName, UnitPrice WHERE UnitPrice> (SELECT AVG(UnitPrice) F...  | Error Code: 1054. Unknown column 'ProductName' in 'field list'                                 | 0.000 sec             |
| 10 | 09:57:21 | SELECT DISTINCT ProductName, UnitPrice FROM products WHERE UnitPrice> (SELECT ...    | 25 row(s) returned   | 0.031 sec / 0.000 sec |
| 11 | 10:04:11 | SELECT DISTINCT ProductName, UnitPrice FROM Products WHERE COUNT(UnitPrice) >=...    | Error Code: 1111. Invalid use of group function  | 0.000 sec             |

## 9. WRITE A QUERY TO COUNT CURRENT AND DISCONTINUED PRODUCTS

**SELECT \* FROM PRODUCTS;**

**SELECT COUNT(ProductName) FROM Products GROUP BY Discontinued;**

The screenshot shows the SQL Server Enterprise Manager interface. The top pane displays a query window with the following SQL code:

```
18 WHERE UnitPrice> (SELECT AVG(UnitPrice) FROM Products)
19 ORDER BY UnitPrice;
20 • SELECT DISTINCT ProductName, UnitPrice FROM Products
21 WHERE COUNT(UnitPrice) >=10 ORDER BY UnitPrice;
22 • SELECT COUNT(ProductName) FROM Products GROUP BY Discontinued;
23 • Select * From Products;
24 • SELECT COUNT(ProductName) FROM Products GROUP BY Discontinued;
```

The bottom pane shows the results grid for the query. The first column is labeled 'COUNT(ProductName)'. The results are as follows:

| COUNT(ProductName) |
|--------------------|
| 69                 |
| 8                  |

The bottom pane also shows the 'Output' window with the following data:

| #  | Time     | Action  | Message  | Duration / Fetch      |
|----|----------|---|--|-----------------------|
| 9  | 09:56:57 | SELECT DISTINCT ProductName, UnitPrice > (SELECT AVG(UnitPrice) F...              | Error Code: 1054. Unknown column 'ProductName' in 'field list'       | 0.000 sec             |
| 10 | 09:57:21 | SELECT DISTINCT ProductName, UnitPrice FROM products WHERE UnitPrice> (SELECT ... | 25 row(s) returned   | 0.031 sec / 0.000 sec |
| 11 | 10:04:11 | SELECT DISTINCT ProductName, UnitPrice FROM Products WHERE COUNT(UnitPrice) >=... | Error Code: 1111. Invalid use of group function                      | 0.000 sec             |
| 12 | 10:08:09 | SELECT COUNT(ProductName) FROM Products GROUP BY Discontinued LIMIT 0, 1000       | Error Code: 1054. Unknown column 'Discontinued' in 'group statement' | 0.000 sec             |
| 13 | 10:09:26 | Select * From Products LIMIT 0, 1000  | 77 row(s) returned   | 0.000 sec / 0.000 sec |
| 14 | 10:09:48 | SELECT COUNT(ProductName) FROM Products GROUP BY Discontinued LIMIT 0, 1000       | 2 row(s) returned  | 0.000 sec / 0.000 sec |

**10. WRITE A QUERY TO GET PRODUCT LIST (NAME, UNITS ON ORDER , UNITS IN STOCK) OF STOCK IS LESS THAN THE QUANTITY ON ORDER**

**SELECT ProductName, UNITSONORDER, UNITSINSTOCK**

**FROM PRODUCTS**

**WHERE ((DISCONTINUED)=FALSE) AND ((UNITSINSTOCK)<(UNITSONORDER));**

The screenshot displays the SQL Enterprise Manager interface. The left pane shows the 'SCHEMAS' tree with 'northwind' selected. The central pane shows the following SQL query:

```
83 • use northwind;
84 • SHOW TABLES;
85 • SELECT * FROM products;
86 • SELECT ProductName, UNITSONORDER, UNITSINSTOCK
87 FROM products
88 WHERE ((discontinued)=false) AND ((UnitsInStock)<(UnitsonOrder));
89
```

Below the query editor, the 'Result Grid' shows the following data:

| ProductName         | UNITSONORDER | UNITSINSTOCK |
|---------------------|--------------|--------------|
| Chang               | 40           | 17           |
| Aniseed Syrup       | 70           | 13           |
| Queso Cabrales      | 30           | 22           |
| Sir Rodney's Scones | 40           | 3            |
| Gorgonzola Telino   | 70           | 0            |

The bottom pane shows the 'Output' window with the 'Action Output' tab selected. It displays a list of actions and their results:

| #  | Time     | Action   | Message   | Duration / Fetch      |
|----|----------|--|---|-----------------------|
| 54 | 22:07:46 | SHOW TABLES  | 8 row(s) returned   | 0.000 sec / 0.000 sec |
| 55 | 22:09:12 | SELECT * FROM products LIMIT 0, 1000   | 77 row(s) returned  | 0.078 sec / 0.000 sec |
| 56 | 22:14:44 | SELECT NAME, UNITSONORDER, UNITSINSTOCK FROM products WHERE ((discontinue... | Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to... | 0.015 sec             |
| 57 | 22:15:28 | SELECT NAME, UNITSONORDER, UNITSINSTOCK FROM products WHERE ((discontinue... | Error Code: 1054. Unknown column 'NAME' in 'field list'   | 0.000 sec             |
| 58 | 22:15:52 | SELECT PRODCUTNAME, UNITSONORDER, UNITSINSTOCK FROM products WHERE ((...     | Error Code: 1054. Unknown column 'PRODCUTNAME' in 'field list'                                  | 0.000 sec             |
| 59 | 22:16:17 | SELECT ProductName, UNITSONORDER, UNITSINSTOCK FROM products WHERE ((disc... | 14 row(s) returned  | 0.000 sec / 0.000 sec |