

**Name:** Pushkaraj Yadav

**Roll No.:** BTB43

**PRN:** 2122000755

**Subject:** Advanced Database Systems Labs

**Experiment No.:** 3

**Problem Statement**

Create a global conceptual schema emp (eno, ename, city, salary) wit eno as a primary key and insert 10 records.

**Horizontal Fragmentation:**

Divide emp into horizontal fragments using the condition that emph1 contains the tuples with salary<=15000 and emph2 with salary>15000.

**Vertical Fragmentation:**

Divide emp into vertical fragments using the condition that empv1 contains the attributes (eno, ename) and empv2 contains the attributes (eno, city, salary)

**CODE:**

```
CREATE TABLE emp (  
    eno INT PRIMARY KEY,  
    ename VARCHAR(50),  
    city VARCHAR(50),  
    salary DECIMAL(10, 2)  
);
```

```
INSERT INTO emp (eno, ename, city, salary) VALUES (1, 'Pushkaraj', 'Pune', 12000);
```

```
INSERT INTO emp (eno, ename, city, salary) VALUES (2, 'Aryan', 'Mumbai', 18000);
```

```
INSERT INTO emp (eno, ename, city, salary) VALUES (3, 'Rohan', 'Bangalore', 9000);
```

```
INSERT INTO emp (eno, ename, city, salary) VALUES (4, 'Ankita', 'Hyderabad', 22000);
```

```

INSERT INTO emp (eno, ename, city, salary) VALUES (5, 'Kiran', 'Chennai', 15000);
INSERT INTO emp (eno, ename, city, salary) VALUES (6, 'Avdhut', 'Pune', 17000);
INSERT INTO emp (eno, ename, city, salary) VALUES (7, 'Ritesh', 'Kolkata', 13000);
INSERT INTO emp (eno, ename, city, salary) VALUES (8, 'Dhairyasheel', 'Jaipur',
25000);
INSERT INTO emp (eno, ename, city, salary) VALUES (9, 'Ishaan', 'Lucknow', 8000);
INSERT INTO emp (eno, ename, city, salary) VALUES (10, 'Atharv', 'Ahmedabad',
16000);

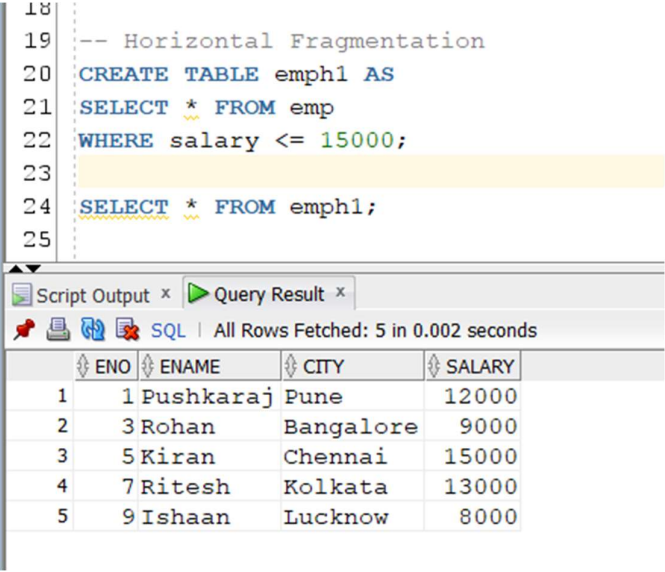
```

### -- Horizontal Fragmentation

```

CREATE TABLE emph1 AS
SELECT * FROM emp
WHERE salary <= 15000;
SELECT * FROM emph1;

```



```

18
19 -- Horizontal Fragmentation
20 CREATE TABLE emph1 AS
21 SELECT * FROM emp
22 WHERE salary <= 15000;
23
24 SELECT * FROM emph1;
25

```

ENO	ENAME	CITY	SALARY
1	1 Pushkaraj	Pune	12000
2	3 Rohan	Bangalore	9000
3	5 Kiran	Chennai	15000
4	7 Ritesh	Kolkata	13000
5	9 Ishaan	Lucknow	8000

```

CREATE TABLE emph2 AS
SELECT * FROM emp
WHERE salary > 15000;

```

SELECT \* FROM emph2;

```
25
26 CREATE TABLE emph2 AS
27 SELECT * FROM emp
28 WHERE salary > 15000;
29
30 SELECT * FROM emph2;
31
```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.002 seconds

	ENO	ENAME	CITY	SALARY
1	2	Aryan	Mumbai	18000
2	4	Ankita	Hyderabad	22000
3	6	Avdhut	Pune	17000
4	8	Dhairyasheel	Jaipur	25000
5	10	Atharv	Ahmedabad	16000

## -- Vertical Fragmentation

CREATE TABLE empv1 AS

SELECT eno, ename FROM emp;

SELECT \* FROM empv1;

```
33 -- Vertical Fragmentation
34 CREATE TABLE empv1 AS
35 SELECT eno, ename FROM emp;
36
37 SELECT * FROM empv1;
```

Script Output x Query Result x

SQL | All Rows Fetched: 10 in 0.005 seconds

	ENO	ENAME
1	1	Pushkaraj
2	2	Aryan
3	3	Rohan
4	4	Ankita
5	5	Kiran
6	6	Avdhut
7	7	Ritesh
8	8	Dhairyasheel
9	9	Ishaan
10	10	Atharv

CREATE TABLE empv2 AS

SELECT eno, city, salary FROM emp;

SELECT \* FROM empv2;

```
38
39 CREATE TABLE empv2 AS
40 SELECT eno, city, salary FROM emp;
41
42 SELECT * FROM empv2;
```

Script Output x Query Result x

SQL | All Rows Fetched: 10 in 0.012 seconds

	ENO	CITY	SALARY
1	1	Pune	12000
2	2	Mumbai	18000
3	3	Bangalore	9000
4	4	Hyderabad	22000
5	5	Chennai	15000
6	6	Pune	17000
7	7	Kolkata	13000
8	8	Jaipur	25000
9	9	Lucknow	8000
10	10	Ahmedabad	16000

-- Q1 Find salary of all employees

SELECT eno, salary FROM emph1

UNION

SELECT eno, salary FROM emph2;

```
44 -- Q1 Find salary of all employees
45 SELECT eno, salary FROM emph1
46 UNION
47 SELECT eno, salary FROM emph2;
48
```

Script Output x Query Result x

SQL | All Rows Fetched: 10 in 0.002 seconds

	ENO	SALARY
1	1	12000
2	3	9000
3	5	15000
4	7	13000
5	9	8000
6	2	18000
7	4	22000
8	6	17000
9	8	25000
10	10	16000

-- Q2 Find the name of all employees where salary = 15000

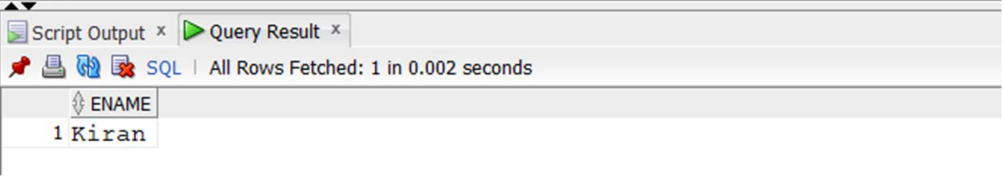
SELECT empv1.ename

FROM empv1

JOIN emph1 ON empv1.eno = emph1.eno

WHERE emph1.salary = 15000;

```
49 -- Q2 Find the name of all employees where salary = 15000
50 SELECT empv1.ename
51 FROM empv1
52 JOIN emph1 ON empv1.eno = emph1.eno
53 WHERE emph1.salary = 15000;
54
```



Script Output x Query Result x

SQL | All Rows Fetched: 1 in 0.002 seconds

ENAME
1 Kiran

-- Q3 Find the employee's name and city where employee salary is between 15000 to 25000

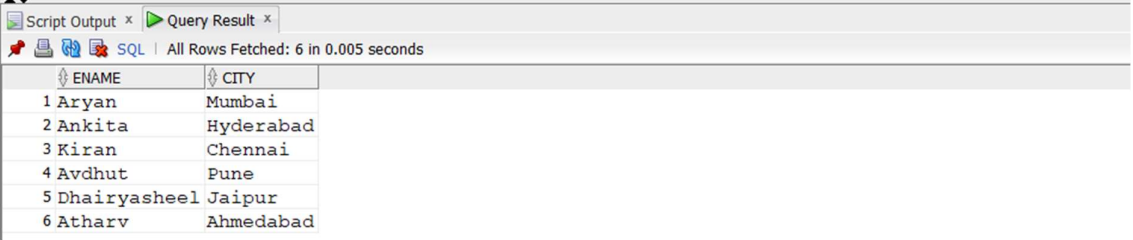
SELECT empv1.ename, empv2.city

FROM empv1

JOIN empv2 ON empv1.eno = empv2.eno

WHERE empv2.salary BETWEEN 15000 AND 25000;

```
55 -- Q3 Find the employee's name and city where employee salary is between 15000 to 25000
56 SELECT empv1.ename, empv2.city
57 FROM empv1
58 JOIN empv2 ON empv1.eno = empv2.eno
59 WHERE empv2.salary BETWEEN 15000 AND 25000;
60
```



Script Output x Query Result x

SQL | All Rows Fetched: 6 in 0.005 seconds

ENAME	CITY
1 Aryan	Mumbai
2 Ankita	Hyderabad
3 Kiran	Chennai
4 Avdhut	Pune
5 Dhairyasheel	Jaipur
6 Atharv	Ahmedabad

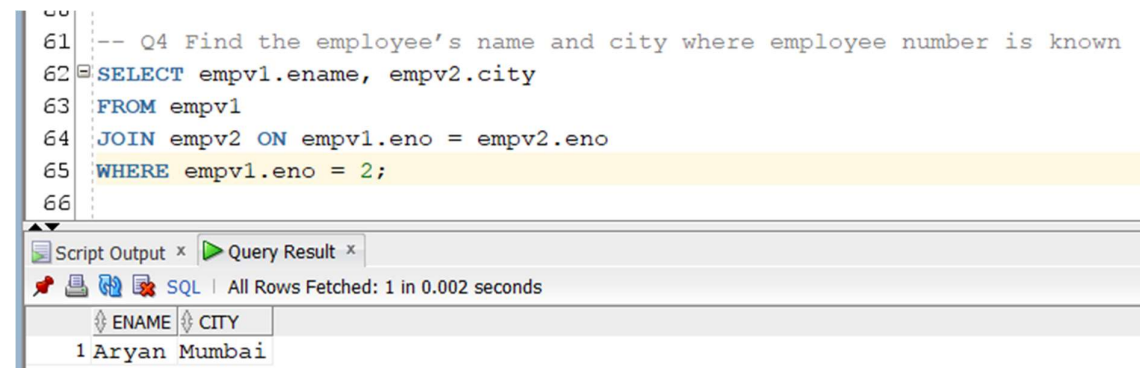
-- Q4 Find the employee's name and city where employee number is known

SELECT empv1.ename, empv2.city

FROM empv1

JOIN empv2 ON empv1.eno = empv2.eno

WHERE empv1.eno = 2;



The screenshot shows a SQL IDE interface. The top pane displays a SQL query with line numbers 61 to 66. The query is:   
61 -- Q4 Find the employee's name and city where employee number is known  
62 SELECT empv1.ename, empv2.city  
63 FROM empv1  
64 JOIN empv2 ON empv1.eno = empv2.eno  
65 WHERE empv1.eno = 2;  
66  
The bottom pane is titled 'Query Result' and shows the execution output. It indicates 'All Rows Fetched: 1 in 0.002 seconds'. Below this, a table with two columns, 'ENAME' and 'CITY', is displayed. The table contains one row with the values 'Aryan' and 'Mumbai'.

```
-- Q4 Find the employee's name and city where employee number is known
SELECT empv1.ename, empv2.city
FROM empv1
JOIN empv2 ON empv1.eno = empv2.eno
WHERE empv1.eno = 2;
```

ENAME	CITY
1 Aryan	Mumbai