B27

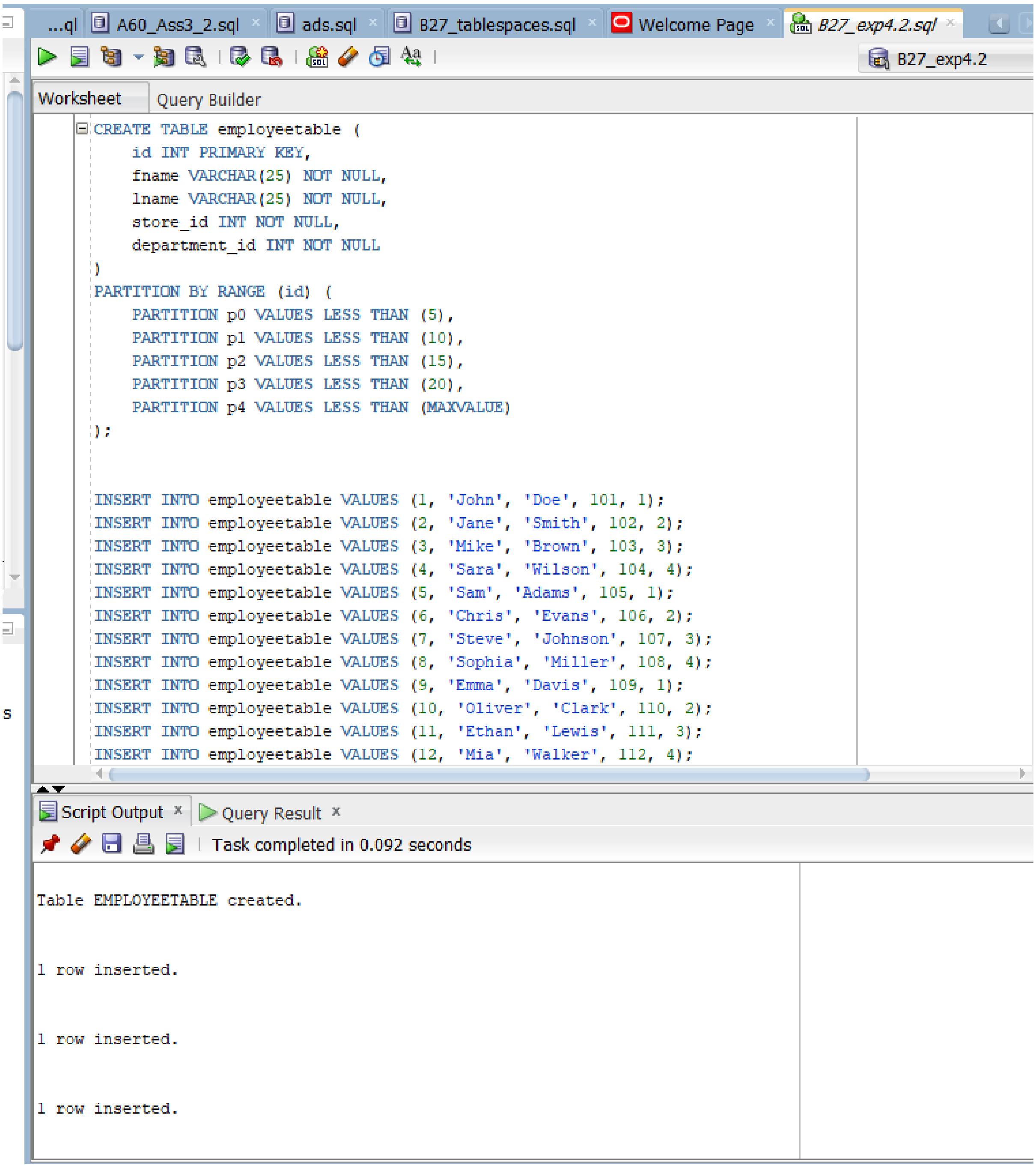
Name: Prachi Mahavir Patil.

PRN: 2122000442

EXPERIMENT NO. 05

---

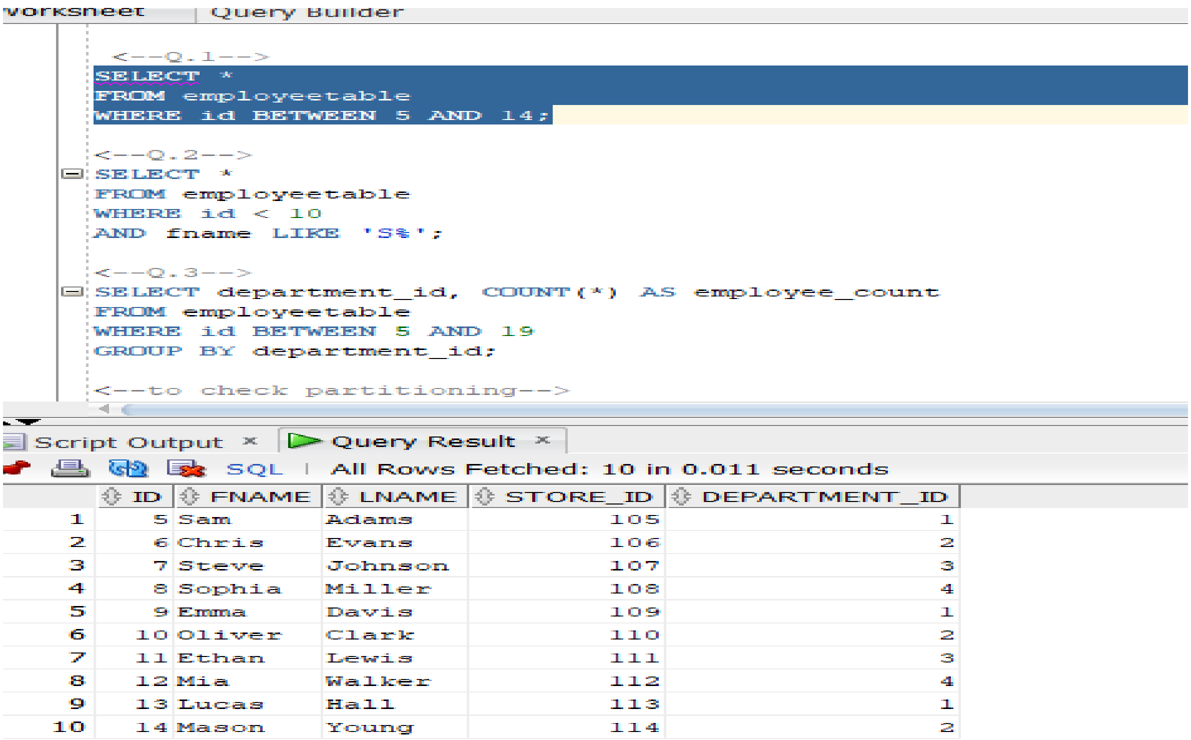
1. Range Partitioning:



```
CREATE TABLE employeetable (
    id INT PRIMARY KEY,
    fname VARCHAR(25) NOT NULL,
    lname VARCHAR(25) NOT NULL,
    store_id INT NOT NULL,
    department_id INT NOT NULL
)
PARTITION BY RANGE (id) (
    PARTITION p0 VALUES LESS THAN (5),
    PARTITION p1 VALUES LESS THAN (10),
    PARTITION p2 VALUES LESS THAN (15),
    PARTITION p3 VALUES LESS THAN (20),
    PARTITION p4 VALUES LESS THAN (MAXVALUE)
);


INSERT INTO employeetable VALUES (1, 'John', 'Doe', 101, 1);
INSERT INTO employeetable VALUES (2, 'Jane', 'Smith', 102, 2);
INSERT INTO employeetable VALUES (3, 'Mike', 'Brown', 103, 3);
INSERT INTO employeetable VALUES (4, 'Sara', 'Wilson', 104, 4);
INSERT INTO employeetable VALUES (5, 'Sam', 'Adams', 105, 1);
INSERT INTO employeetable VALUES (6, 'Chris', 'Evans', 106, 2);
INSERT INTO employeetable VALUES (7, 'Steve', 'Johnson', 107, 3);
INSERT INTO employeetable VALUES (8, 'Sophia', 'Miller', 108, 4);
INSERT INTO employeetable VALUES (9, 'Emma', 'Davis', 109, 1);
INSERT INTO employeetable VALUES (10, 'Oliver', 'Clark', 110, 2);
INSERT INTO employeetable VALUES (11, 'Ethan', 'Lewis', 111, 3);
INSERT INTO employeetable VALUES (12, 'Mia', 'Walker', 112, 4);
```

Task completed in 0.092 seconds

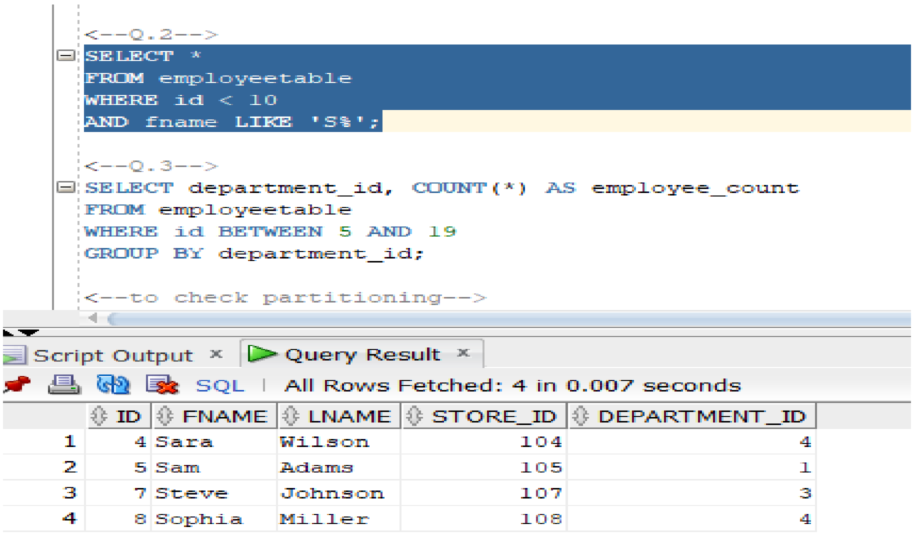
```
Table EMPLOYEETABLE created.


1 row inserted.


1 row inserted.


1 row inserted.
```



```
<--Q.1-->
SELECT *
FROM employeetable
WHERE id BETWEEN 5 AND 14;

<--Q.2-->
SELECT *
FROM employeetable
WHERE id < 10
AND fname LIKE 'S%';

<--Q.3-->
SELECT department_id, COUNT(*) AS employee_count
FROM employeetable
WHERE id BETWEEN 5 AND 19
GROUP BY department_id;

<--to check partitioning-->
```

All Rows Fetched: 10 in 0.011 seconds

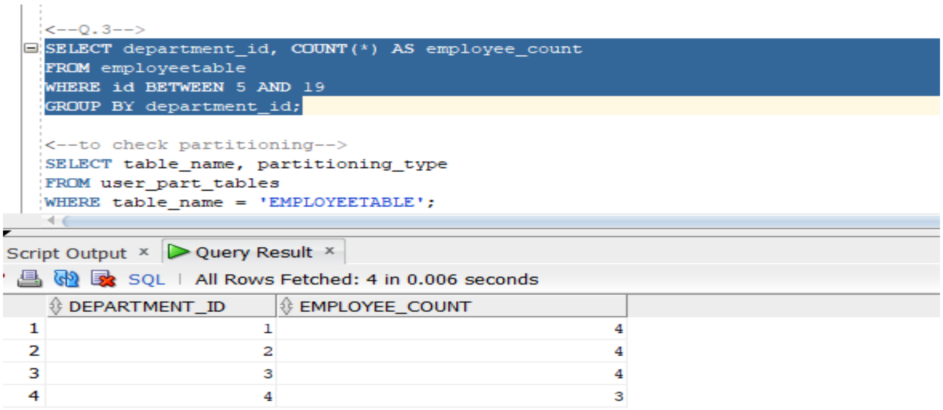
	ID	FNAME	LNAME	STORE_ID	DEPARTMENT_ID
1	5	Sam	Adams	105	1
2	6	Chris	Evans	106	2
3	7	Steve	Johnson	107	3
4	8	Sophia	Miller	108	4
5	9	Emma	Davis	109	1
6	10	Oliver	Clark	110	2
7	11	Ethan	Lewis	111	3
8	12	Mia	Walker	112	4
9	13	Lucas	Hall	113	1
10	14	Mason	Young	114	2



```
<--Q.2-->
SELECT *
FROM employeetable
WHERE id < 10
AND fname LIKE 'S%';

<--Q.3-->
SELECT department_id, COUNT(*) AS employee_count
FROM employeetable
WHERE id BETWEEN 5 AND 19
GROUP BY department_id;

<--to check partitioning-->
```

All Rows Fetched: 4 in 0.007 seconds

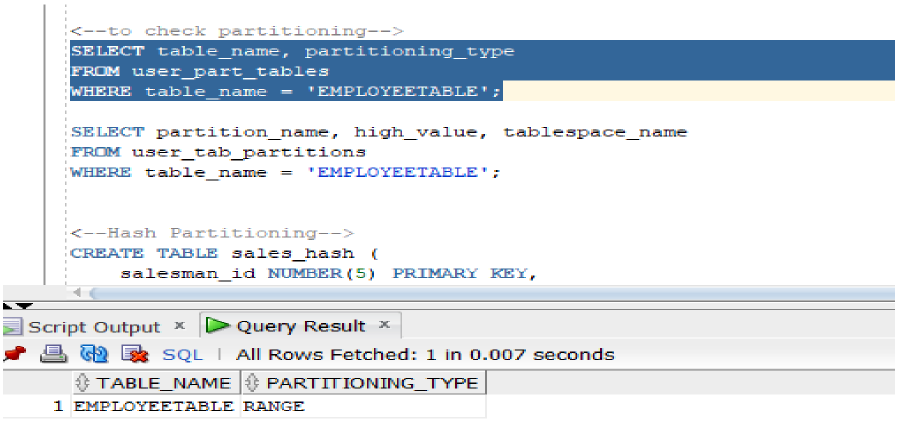
	ID	FNAME	LNAME	STORE_ID	DEPARTMENT_ID
1	4	Sara	Wilson	104	4
2	5	Sam	Adams	105	1
3	7	Steve	Johnson	107	3
4	8	Sophia	Miller	108	4



```
<--Q.3-->
SELECT department_id, COUNT(*) AS employee_count
FROM employeetable
WHERE id BETWEEN 5 AND 19
GROUP BY department_id;

<--to check partitioning-->
SELECT table_name, partitioning_type
FROM user_part_tables
WHERE table_name = 'EMPLOYEETABLE';
```

All Rows Fetched: 4 in 0.006 seconds

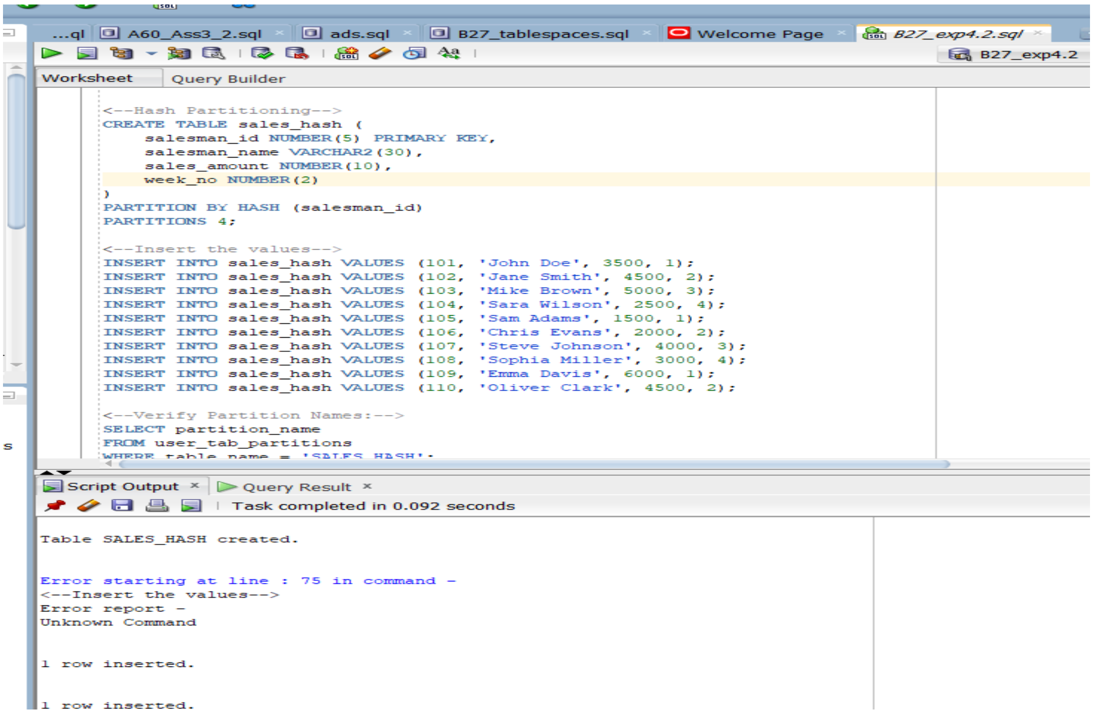
	DEPARTMENT_ID	EMPLOYEE_COUNT
1	1	4
2	2	4
3	3	4
4	4	3



```
<--to check partitioning-->
SELECT table_name, partitioning_type
FROM user_part_tables
WHERE table_name = 'EMPLOYEETABLE';

SELECT partition_name, high_value, tablespace_name
FROM user_tab_partitions
WHERE table_name = 'EMPLOYEETABLE';


<--Hash Partitioning-->
CREATE TABLE sales_hash (
    salesman_id NUMBER(5) PRIMARY KEY,
```

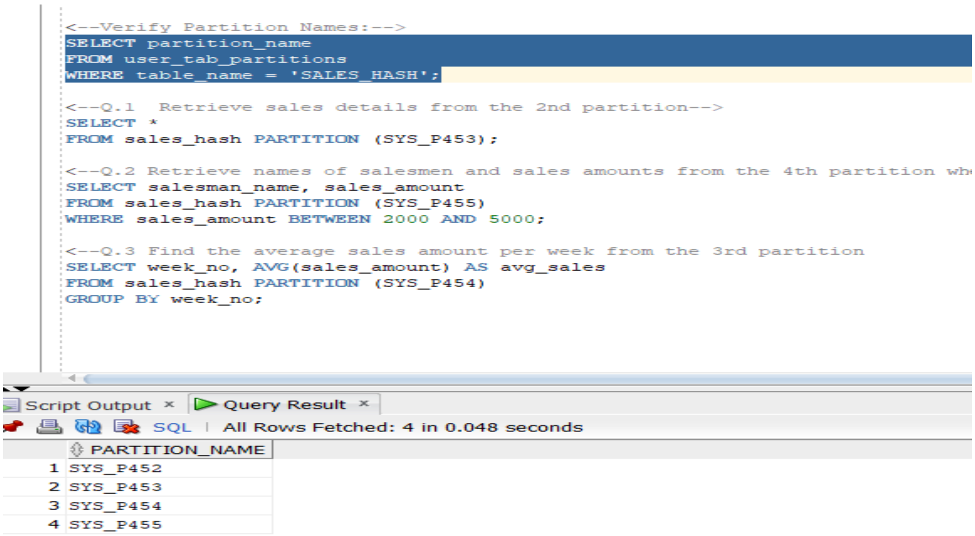
	TABLE_NAME	PARTITIONING_TYPE
1	EMPLOYEETABLE	RANGE

2. Hash Partitioning:



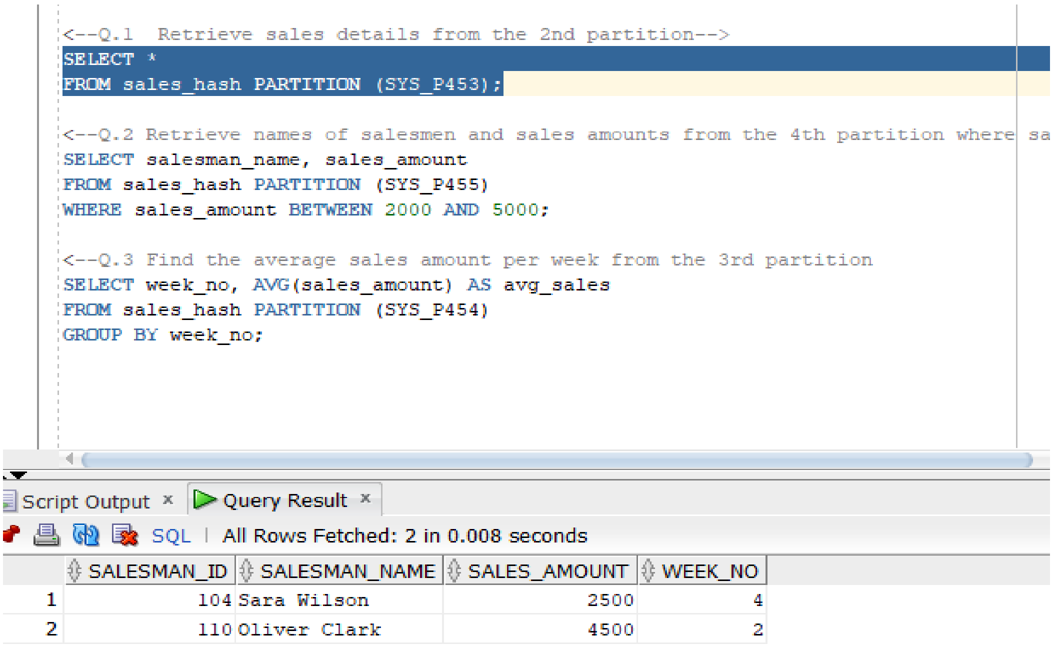
```
<--Hash Partitioning-->
CREATE TABLE sales_hash (
    salesman_id NUMBER(5) PRIMARY KEY,
    salesman_name VARCHAR2(30),
    sales_amount NUMBER(10),
    week_no NUMBER(2)
)
PARTITION BY HASH (salesman_id)
PARTITIONS 4;

<--Insert the values-->
INSERT INTO sales_hash VALUES (101, 'John Doe', 3500, 1);
INSERT INTO sales_hash VALUES (102, 'Jane Smith', 4500, 2);
INSERT INTO sales_hash VALUES (103, 'Mike Brown', 5000, 3);
INSERT INTO sales_hash VALUES (104, 'Sara Wilson', 2500, 4);
INSERT INTO sales_hash VALUES (105, 'Sam Adams', 1500, 1);
INSERT INTO sales_hash VALUES (106, 'Chris Evans', 2000, 2);
INSERT INTO sales_hash VALUES (107, 'Steve Johnson', 4000, 3);
INSERT INTO sales_hash VALUES (108, 'Sophia Miller', 3000, 4);
INSERT INTO sales_hash VALUES (109, 'Emma Davis', 6000, 1);
INSERT INTO sales_hash VALUES (110, 'Oliver Clark', 4500, 2);

<--Verify Partition Names:-->
SELECT partition_name
FROM user_tab_partitions
WHERE table_name = 'SALES_HASH';
```

```
Table SALES_HASH created.

Error starting at line : 75 in command -
<--Insert the values-->
Error report -
Unknown Command


1 row inserted.


1 row inserted.
```



```
<--Verify Partition Names:-->
SELECT partition_name
FROM user_tab_partitions
WHERE table_name = 'SALES_HASH';

<--Q.1  Retrieve sales details from the 2nd partition-->
SELECT *
FROM sales_hash PARTITION (SYS_P453);

<--Q.2 Retrieve names of salesmen and sales amounts from the 4th partition whe
SELECT salesman_name, sales_amount
FROM sales_hash PARTITION (SYS_P455)
WHERE sales_amount BETWEEN 2000 AND 5000;

<--Q.3 Find the average sales amount per week from the 3rd partition
SELECT week_no, AVG(sales_amount) AS avg_sales
FROM sales_hash PARTITION (SYS_P454)
GROUP BY week_no;
```

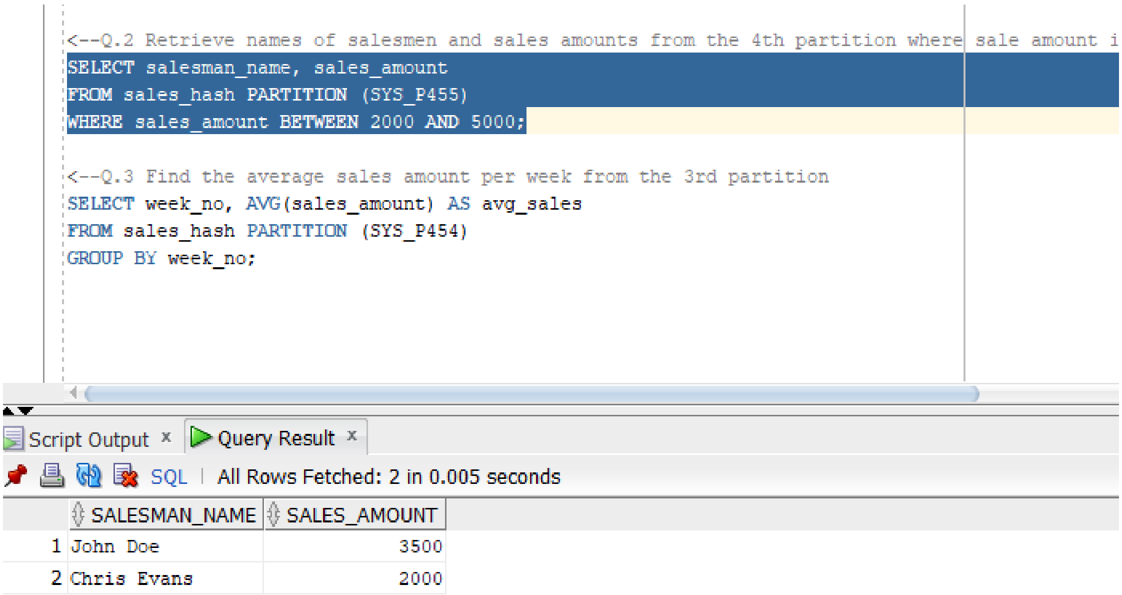
	PARTITION_NAME
1	SYS_P452
2	SYS_P453
3	SYS_P454
4	SYS_P455



```
<--Q.1  Retrieve sales details from the 2nd partition-->
SELECT *
FROM sales_hash PARTITION (SYS_P453);

<--Q.2 Retrieve names of salesmen and sales amounts from the 4th partition where sa
SELECT salesman_name, sales_amount
FROM sales_hash PARTITION (SYS_P455)
WHERE sales_amount BETWEEN 2000 AND 5000;

<--Q.3 Find the average sales amount per week from the 3rd partition
SELECT week_no, AVG(sales_amount) AS avg_sales
FROM sales_hash PARTITION (SYS_P454)
GROUP BY week_no;
```

All Rows Fetched: 2 in 0.008 seconds

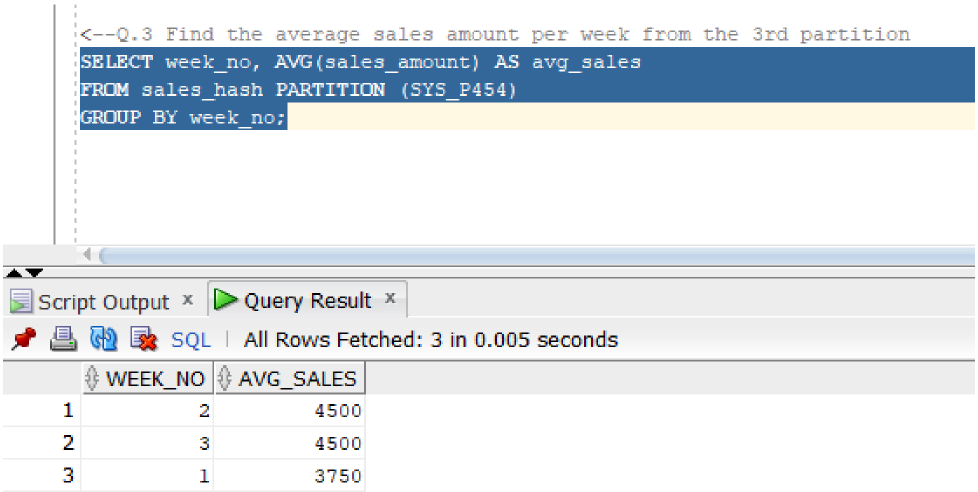
	SALESMAN_ID	SALESMAN_NAME	SALES_AMOUNT	WEEK_NO
1	104	Sara Wilson	2500	4
2	110	Oliver Clark	4500	2



```
<--Q.2 Retrieve names of salesmen and sales amounts from the 4th partition where sale amount i
SELECT salesman_name, sales_amount
FROM sales_hash PARTITION (SYS_P455)
WHERE sales_amount BETWEEN 2000 AND 5000;

<--Q.3 Find the average sales amount per week from the 3rd partition
SELECT week_no, AVG(sales_amount) AS avg_sales
FROM sales_hash PARTITION (SYS_P454)
GROUP BY week_no;
```

All Rows Fetched: 2 in 0.005 seconds

	SALESMAN_NAME	SALES_AMOUNT
1	John Doe	3500
2	Chris Evans	2000



```
<--Q.3 Find the average sales amount per week from the 3rd partition
SELECT week_no, AVG(sales_amount) AS avg_sales
FROM sales_hash PARTITION (SYS_P454)
GROUP BY week_no;
```

All Rows Fetched: 3 in 0.005 seconds

	WEEK_NO	AVG_SALES
1	2	4500
2	3	4500
3	1	3750