B30 Shruti Ghate ADS experiment 5

Implementation Range and Hash Partition

1)Range Partition:

```
CREATE TABLE employees (
   id INT NOT NULL 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)
);

INSERT INTO employees VALUES (1, 'John', 'Doe', 101, 1);
INSERT INTO employees VALUES (2, 'Sarah', 'Smith', 102, 2);
INSERT INTO employees VALUES (3, 'Alice', 'Brown', 101, 1);
INSERT INTO employees VALUES (4, 'Steve', 'Johnson', 102, 3);
INSERT INTO employees VALUES (5, 'Bob', 'Davis', 103, 4);
INSERT INTO employees VALUES (6, 'Tom', 'Jones', 101, 1);
INSERT INTO employees VALUES (7, 'Susan', 'Clark', 102, 2);
INSERT INTO employees VALUES (8, 'Mike', 'Anderson', 103, 3);
INSERT INTO employees VALUES (9, 'Peter', 'White', 101, 4);
INSERT INTO employees VALUES (10, 'Rick', 'Martin', 102, 1);
INSERT INTO employees VALUES (11, 'Paul', 'Taylor', 103, 2);
INSERT INTO employees VALUES (12, 'Nancy', 'Moore', 101, 3);
INSERT INTO employees VALUES (13, 'Linda', 'Thomas', 102, 4);
INSERT INTO employees VALUES (14, 'George', 'Harris', 103, 1);
INSERT INTO employees VALUES (15, 'Sandra', 'Garcia', 101, 2);
INSERT INTO employees VALUES (16, 'Kevin', 'Martinez', 102, 3);
INSERT INTO employees VALUES (17, 'Betty', 'Robinson', 103, 4);
INSERT INTO employees VALUES (18, 'Lisa', 'Rodriguez', 101, 1);
INSERT INTO employees VALUES (19, 'James', 'Lewis', 102, 2);

Select * from employees;
```

Script Output | Query Result

SQL | All Rows Fetched: 19 in 0.015 seconds

	ID	FNAME	LNAME	STORE_ID	DEPARTMENT_ID
3	3	Alice	Brown	101	1
4	4	Steve	Johnson	102	3
5	5	Bob	Davis	103	4
6	6	Tom	Jones	101	1
7	7	Susan	Clark	102	2
8	8	Mike	Anderson	103	3
9	9	Peter	White	101	4
10	10	Rick	Martin	102	1
11	11	Paul	Taylor	103	2
12	12	Nancy	Moore	101	3
13	13	Linda	Thomas	102	4
14	14	George	Harris	103	1
15	15	Sandra	Garcia	101	2
16	16	Kevin	Martinez	102	3
17	17	Betty	Robinson	103	4
18	18	Lisa	Rodriguez	101	1
19	19	James	Lewis	102	2

Query 1: Retrieve employee details from partition P1 and P2.
SELECT * FROM employees WHERE id >= 5 AND id < 15;

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.011 seconds

	ID	FNAME	LNAME	STORE_ID	DEPARTMENT_ID
1	5	Bob	Davis	103	4
2	6	Tom	Jones	101	1
3	7	Susan	Clark	102	2
4	8	Mike	Anderson	103	3
5	9	Peter	White	101	4
6	10	Rick	Martin	102	1
7	11	Paul	Taylor	103	2
8	12	Nancy	Moore	101	3
9	13	Linda	Thomas	102	4
10	14	George	Harris	103	1

Query 2: Retrieve employee details from partition P0 and P1 where `fname` begins with 'S'.
SELECT * FROM employees WHERE id < 10 AND fname LIKE 'S%';

Script Output | Query Result

SQL | All Rows Fetched: 3 in 0.008 seconds

	ID	FNAME	LNAME	STORE_ID	DEPARTMENT_ID
1	2	Sarah	Smith	102	2
2	4	Steve	Johnson	102	3
3	7	Susan	Clark	102	2

Query 3: Count the number of employees from each department from P1, P2, and P3.

```
SELECT department_id, COUNT(*) AS employee_count
FROM employees
WHERE id >= 5 AND id < 20
GROUP BY department_id;
```

Script Output | Query Result

SQL | All Rows Fetched: 4 in 0.007 seconds

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

---

2)Hash Partition:

```
CREATE TABLE sales_hash (
   salesman_id NUMBER(5) PRIMARY KEY,
   salesman_name VARCHAR2(30) NOT NULL,
   sales_amount NUMBER(10),
   week_no NUMBER(2)
)
PARTITION BY HASH (salesman_id)
PARTITIONS 4;

INSERT INTO sales_hash VALUES (101, 'John', 2500, 1);
INSERT INTO sales_hash VALUES (102, 'Sarah', 4000, 2);
INSERT INTO sales_hash VALUES (103, 'Alice', 3500, 3);
INSERT INTO sales_hash VALUES (104, 'Steve', 6000, 1);
INSERT INTO sales_hash VALUES (105, 'Bob', 1500, 2);
INSERT INTO sales_hash VALUES (106, 'Tom', 2000, 3);
INSERT INTO sales_hash VALUES (107, 'Susan', 5000, 1);
```

```
INSERT INTO sales_hash VALUES (108, 'Mike', 7000, 2);
INSERT INTO sales_hash VALUES (109, 'Peter', 3000, 3);
INSERT INTO sales_hash VALUES (110, 'Rick', 1000, 1);

select * from sales_hash;
```

Script Output | Query Result

SQL | All Rows Fetched: 10 in 0.01 seconds

	SALESMAN_ID	SALESMAN_NAME	SALES_AMOUNT	WEEK_NO
1	108	Mike	7000	2
2	104	Steve	6000	1
3	110	Rick	1000	1
4	102	Sarah	4000	2
5	103	Alice	3500	3
6	105	Bob	1500	2
7	107	Susan	5000	1
8	109	Peter	3000	3
9	101	John	2500	1
10	106	Tom	2000	3

```
SELECT partition_name
FROM all_tab_partitions
WHERE table_name = 'SALES_HASH';
```

Script Output | Query Result

SQL | All Rows Fetched: 4 in 0.152 seconds

	PARTITION_NAME
1	SYS_P468
2	SYS_P469
3	SYS_P470
4	SYS_P471

Query 1: Retrieve sales details from the 2nd partition.
SELECT * FROM sales_hash PARTITION (SYS_P469);

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.007 seconds

	SALESMAN_ID	SALESMAN_NAME	SALES_AMOUNT	WEEK_NO
1	104	Steve	6000	1
2	110	Rick	1000	1

Query 2: Retrieve names of salesmen and sales amounts from the 4th partition where the sale amount is between 2000 and 5000.
SELECT salesman_name, sales_amount
FROM sales_hash PARTITION (SYS_P471)
WHERE sales_amount BETWEEN 2000 AND 5000;

Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.006 seconds

	SALESMAN_NAME	SALES_AMOUNT
1	John	2500
2	Tom	2000

Query 3: Find the average sale amount per week from the 3rd partition.
SELECT AVG(sales_amount) AS avg_sales
FROM sales_hash PARTITION (SYS_P470);

Script Output | Query Result

SQL | All Rows Fetched: 1 in 0.006 seconds

	AVG_SALES
1	3400