

Experiment No. 05

Implement Range and Hash Partitioning

Name: Samiksha R. Devardekar

Div.: B

Roll No.: 05

PRN: 2122000098

Problem Statement

Range Partitioning

Consider a table named **employees** with schema emp (id int, fname varchar(25) not null, lname varchar(25) not null, store_id int not null, department_id int not null) with id as a primary key and insert 20 records with id ranges from 1 to 20.

Make 4 partitions by range:

P0: id < 5

P1: id < 10

P2: id < 15

P3: id < 20 or Maxvalue.

```
create table employees(  
    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)  
);
```

```
insert into employees values (2, 'Harnish', 'Savsani', 111, 301);  
insert into employees values (3, 'Pranav', 'Kulkarni', 112, 301);  
insert into employees values(4, 'Suraj', 'Patil', 113, 302);  
insert into employees values (5, 'Pundalik', 'Masurkar', 114, 302);  
insert into employees values(6, 'Virat', 'Kohli', 115, 303);  
insert into employees values (7, 'Mahinder', 'Dhoni', 116, 303);  
Insert into employees values(8, 'Rohit', 'Sharma', 117, 303);  
insert into employees values(9, 'Ravinder', 'Jadeja', 118, 303);  
insert into employees values (10, 'Harshal', 'Patel', 119, 304);  
INSERT INTO employees VALUES (11, 'Prasant', 'Patil', 100, 304);  
INSERT INTO employees VALUES (12, 'Ranjit', 'Yadav', 101,305);  
INSERT INTO employees VALUES(13, 'Yash', 'Patel', 102,306);  
INSERT INTO employees VALUES (14, 'Neel','Savsani', 103,306);  
INSERT INTO employees VALUES (15, 'Dev', 'Hirani',104,306);  
INSERT INTO employees VALUES (16, 'Pranav', 'Kulkarni', 105, 306);  
INSERT INTO employees VALUES (17, 'Kishan', 'Dixit', 106, 307);  
INSERT INTO employees VALUES (18, 'Steve', 'Jobs', 107, 308);  
INSERT INTO employees VALUES(19,'Elon','Musk',108,308);
```

--1. Retrieve employee details from partition P1 and P2.

```
select * from employees partition(p1) union select * from employees partition(p2);
```

Query Result x					
SQL All Rows Fetched: 10 in 0.002 seconds					
	ID	FNAME	LNAME	STORE_ID	DEPARTMENT_ID
1	5	Pundalik	Masurkar	114	302
2	6	Virat	Kohli	115	303
3	7	Mahinder	Dhoni	116	303
4	8	Rohit	Sharma	117	303
5	9	Ravinder	Jadeja	118	303
6	10	Harshal	Patel	119	304
7	11	Prasant	Patil	100	304
8	12	Ranjit	Yadav	101	305
9	13	Yash	Patel	102	306
10	14	Neel	Savsani	103	306

--2. Retrieve employee details from partition P0 and P1 where fname begin with 'S'.

select * from employees partition(p0) where fname like 'S%' union select * from employees partition(p1) where fname like 'S%';

Query Result x					
SQL All Rows Fetched: 1 in 0.004 seconds					
	ID	FNAME	LNAME	STORE_ID	DEPARTMENT_ID
1	4	Suraj	Patil	113	302

--3. Count number of employees from each department from p1, p2 and p3.

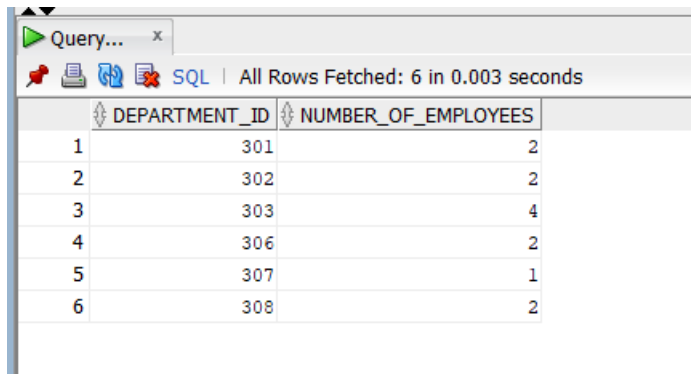
select department_id,count(*) Number_Of_Employees from

(select * from employees minus select * from employees partition(p1)) group by department_id;

Query Result x		
SQL All Rows Fetched: 7 in 0.004 seconds		
	DEPARTMENT_ID	NUMBER_OF_EMPLOYEES
1	301	2
2	302	1
3	304	2
4	305	1
5	306	4
6	307	1
7	308	2

select department_id,count(*) Number_Of_Employees from

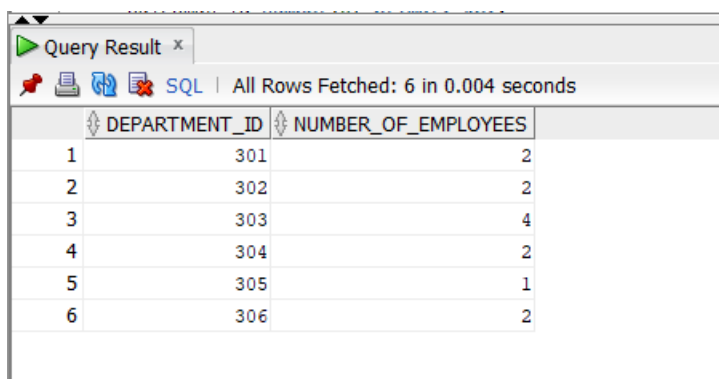
(select * from employees minus select * from employees partition(p2)) group by department_id;



	DEPARTMENT_ID	NUMBER_OF_EMPLOYEES
1	301	2
2	302	2
3	303	4
4	306	2
5	307	1
6	308	2

select department_id,count(*) Number_Of_Employees from

(select * from employees minus select * from employees partition(p3)) group by department_id;



	DEPARTMENT_ID	NUMBER_OF_EMPLOYEES
1	301	2
2	302	2
3	303	4
4	304	2
5	305	1
6	306	2

Hash Partitioning

Consider a table named sales_hash with schema (salesman_id number(5), salesman_name varchar2(30), sales_amount number(10), week_no number(2)) with salesman_id as primary key and insert at least 10 records.

Create 4 partitions using hash partitioning.

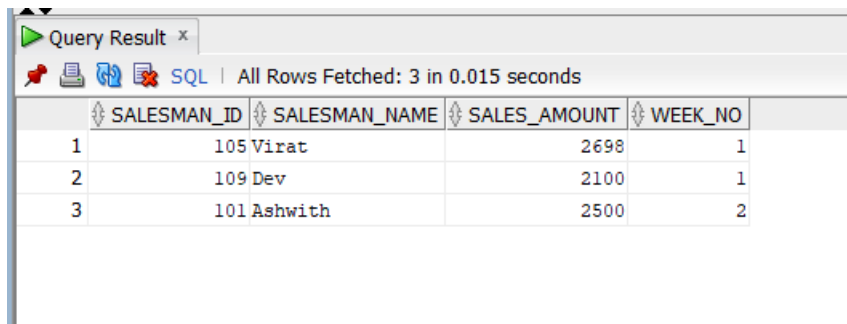
```
create table sales_hash(
    salesman_id number(5) primary key,
    salesman_name varchar2(30),
    sales_amount number(10),
    week_no number(2)
)
PARTITION BY HASH(salesman_id)
```

PARTITIONS 4;

```
insert into sales_hash values(101, 'Ashwith', 2500, 2);
insert into sales_hash values(102, 'Harnish', 1500, 1);
insert into sales_hash values (103, 'Suraj', 3200, 4);
insert into sales_hash values (104, 'Pranav', 9500, 2);
insert into sales_hash values (105, 'Virat', 2698, 1);
Insert INTO sales_hash values (106, 'Rahul', 2500,5);
Insert INTO sales_hash values (107, 'Sai', 3000,3);
Insert INTO sales_hash values (108, 'Kushal', 6000,6);
Insert INTO sales_hash values (109, 'Dev', 2100,1);
Insert INTO sales_hash values (110, 'Vallab',1500,2);
```

--1. Retrieve sales details from 2nd partition.

```
select * from sales_hash where mod(salesman_id,4)=1;
```



The screenshot shows a 'Query Result' window with the following data:

	SALESMAN_ID	SALESMAN_NAME	SALES_AMOUNT	WEEK_NO
1	105	Virat	2698	1
2	109	Dev	2100	1
3	101	Ashwith	2500	2

--2. Retrieve name of sales mans and amount from 4th partition where sale amount between 2000 and 5000.

```
select salesman_name, sales_amount from sales_hash
where mod(salesman_id,4)=3 and sales_amount between 2000 and 5000;
```

Query Result x		
SQL All Rows Fetched: 2 in 0.002 seconds		
	SALESMAN_NAME	SALES_AMOUNT
1	Suraj	3200
2	Sai	3000

--3. Find average sale amount per week from 3rd partition.

```
select week_no, avg(sales_amount) as avg_sales_amount from sales_hash
where mod(salesman_id,4)=2
group by week_no;
```

Query... x		
SQL All Rows Fetched: 3 in 0.001 seconds		
	WEEK_NO	AVG_SALES_AMOUNT
1	2	1500
2	1	1500
3	5	2500