Assignment 4.2

--Range Partition: create table Emp (

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 Emp (id, fname, lname, store_id, department_id) values (1, 'john', 'doe', 1, 101); insert

into Emp (id, fname, lname, store_id, department_id) values (2, 'jane', 'smith', 1, 102); insert into

Emp (id, fname, lname, store_id, department_id) values (3, 'paul', 'adams', 2, 103); insert into

Emp (id, fname, lname, store_id, department_id) values (4, 'sophie', 'lee', 2, 101); insert into Emp

(id, fname, lname, store_id, department_id) values (5, 'sam', 'jackson', 3, 102); insert into Emp (id,

fname, lname, store_id, department_id) values (6, 'sara', 'connor', 3, 101); insert into Emp (id,

fname, lname, store_id, department_id) values (7, 'mike', 'jordan', 4, 103); insert into Emp (id,

fname, lname, store_id, department_id) values (8, 'chris', 'taylor', 4, 102); insert into Emp (id, fname,

lname, store_id, department_id) values (9, 'susan', 'brown', 5, 101); insert into Emp (id, fname,

lname, store_id, department_id) values (10, 'tom', 'hardy', 5, 103); insert into Emp (id, fname,

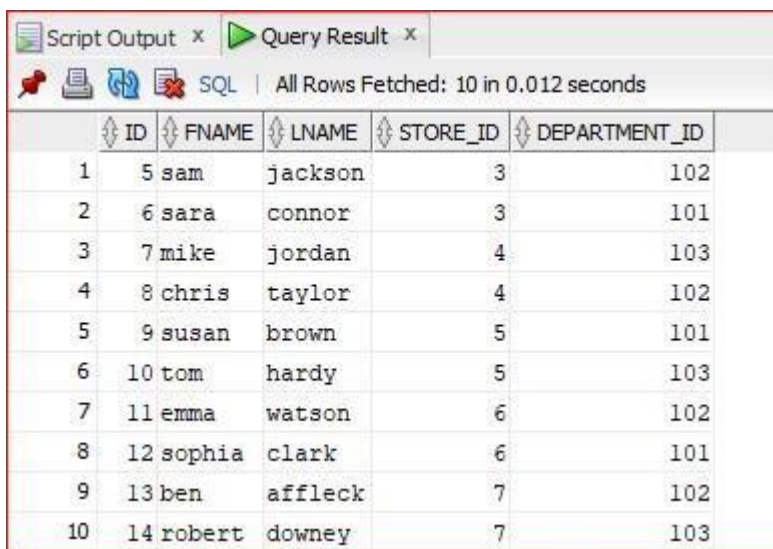lname, store_id, department_id) values (11, 'emma', 'watson', 6, 102); insert into Emp (id, fname,

lname, store_id, department_id) values (12, 'sophia', 'clark', 6, 101); insert into Emp (id, fname,

lname, store_id, department_id) values (13, 'ben', 'affleck', 7, 102); insert into Emp (id, fname,

lname, store_id, department_id) values (14, 'robert', 'downey', 7, 103); insert into Emp (id, fname,

lname, store_id, department_id) values (15, 'anna', 'williams', 8, 101); insert into Emp (id, fname,

lname, store_id, department_id) values (16, 'steve', 'johnson', 8, 102); insert into Emp (id, fname,

lname, store_id, department_id) values (17, 'rachel', 'green', 9, 103); insert into Emp (id, fname,

lname, store_id, department_id) values (18, 'phoebe', 'buffay', 9, 101); insert into Emp (id, fname,

lname, store_id, department_id) values (19, 'ross', 'geller', 10, 102); insert into Emp (id, fname,

lname, store_id, department_id) values (20, 'monica', 'geller', 10, 101);

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

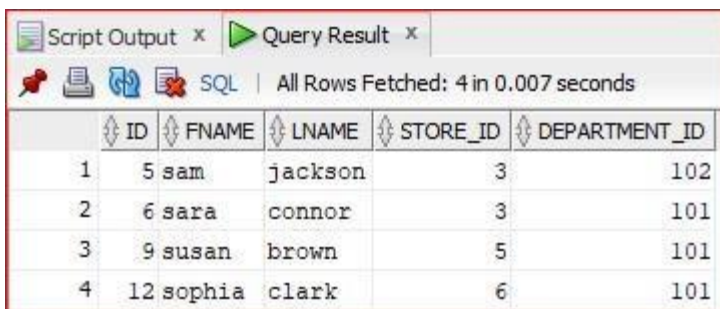Emp partition(p1) union select * from Emp partition(p2);

	ID	FNAME	LNAME	STORE_ID	DEPARTMENT_ID
1	5	sam	jackson	3	102
2	6	sara	connor	3	101
3	7	mike	jordan	4	103
4	8	chris	taylor	4	102
5	9	susan	brown	5	101
6	10	tom	hardy	5	103
7	11	emma	watson	6	102
8	12	sophia	clark	6	101
9	13	ben	affleck	7	102
10	14	robert	downey	7	103

Script Output × Query Result ×
SQL | All Rows Fetched: 10 in 0.012 seconds

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

select * from Emp partition(p1) where fname like 's%' union select * from Emp partition(p2)

where fname like 's%';

	ID	FNAME	LNAME	STORE_ID	DEPARTMENT_ID
1	5	sam	jackson	3	102
2	6	sara	connor	3	101
3	9	susan	brown	5	101
4	12	sophia	clark	6	101

Script Output × Query Result ×
SQL | All Rows Fetched: 4 in 0.007 seconds

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

count(*) as Emp_count from

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

select department_id, count(*) as Emp_count from

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

select department_id, count(*) as Emp_count from

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

--Hash Partition:

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 (salesman_id, salesman_name, sales_amount, week_no) values (1, 'alice', 3000, 1);

insert into sales_hash (salesman_id, salesman_name, sales_amount, week_no) values (2, 'bob', 4500, 1);

insert into sales_hash (salesman_id, salesman_name, sales_amount, week_no) values (3, 'charlie', 2500, 2);

insert into sales_hash (salesman_id, salesman_name, sales_amount, week_no) values (4, 'david', 5000, 2);

insert into sales_hash (salesman_id, salesman_name, sales_amount, week_no) values (5, 'eve', 6000, 3);
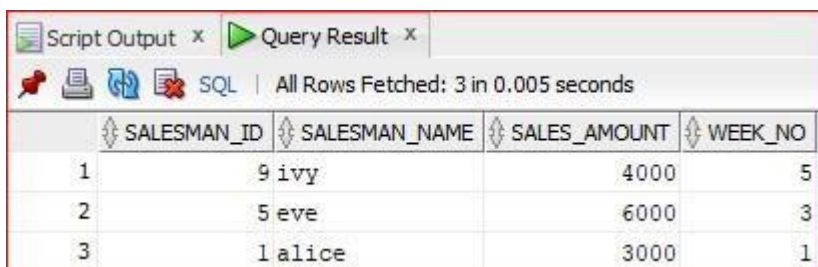
insert into sales_hash (salesman_id, salesman_name, sales_amount, week_no) values (6, 'frank', 7000, 3);

insert into sales_hash (salesman_id, salesman_name, sales_amount, week_no) values (7, 'grace', 2000, 4);

insert into sales_hash (salesman_id, salesman_name, sales_amount, week_no) values (8, 'hank', 1000, 4);

insert into sales_hash (salesman_id, salesman_name, sales_amount, week_no) values (9, 'ivy', 4000, 5);

insert into sales_hash (salesman_id, salesman_name, sales_amount, week_no) values (10, 'jack', 5500, 5);

select partition_name from user_tab_partitions where table_name = 'sales_hash';

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

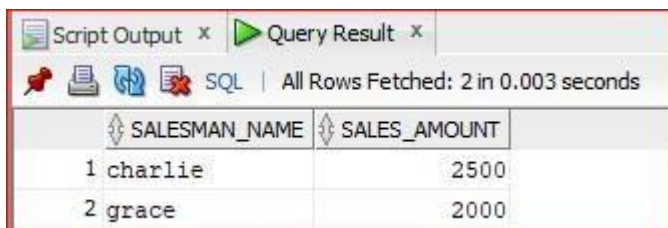sales_hash where mod(salesman_id, 4) = 1;

Script Output × | Query Result ×

SQL | All Rows Fetched: 3 in 0.005 seconds

	SALESMAN_ID	SALESMAN_NAME	SALES_AMOUNT	WEEK_NO
1	9	ivy	4000	5
2	5	eve	6000	3
3	1	alice	3000	1

--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

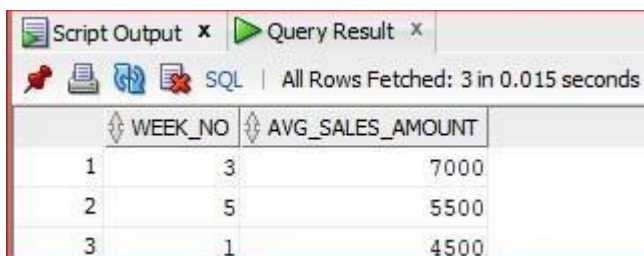sales_amount between 2000 and 5000;

Script Output × | Query Result ×

SQL | All Rows Fetched: 2 in 0.003 seconds

	SALESMAN_NAME	SALES_AMOUNT
1	charlie	2500
2	grace	2000

--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;

Script Output × | Query Result ×

SQL | All Rows Fetched: 3 in 0.015 seconds

	WEEK_NO	AVG_SALES_AMOUNT
1	3	7000
2	5	5500
3	1	4500