Student Name : Santosh Acharya
Student Id : C0930325
Program Code : CSD 2206

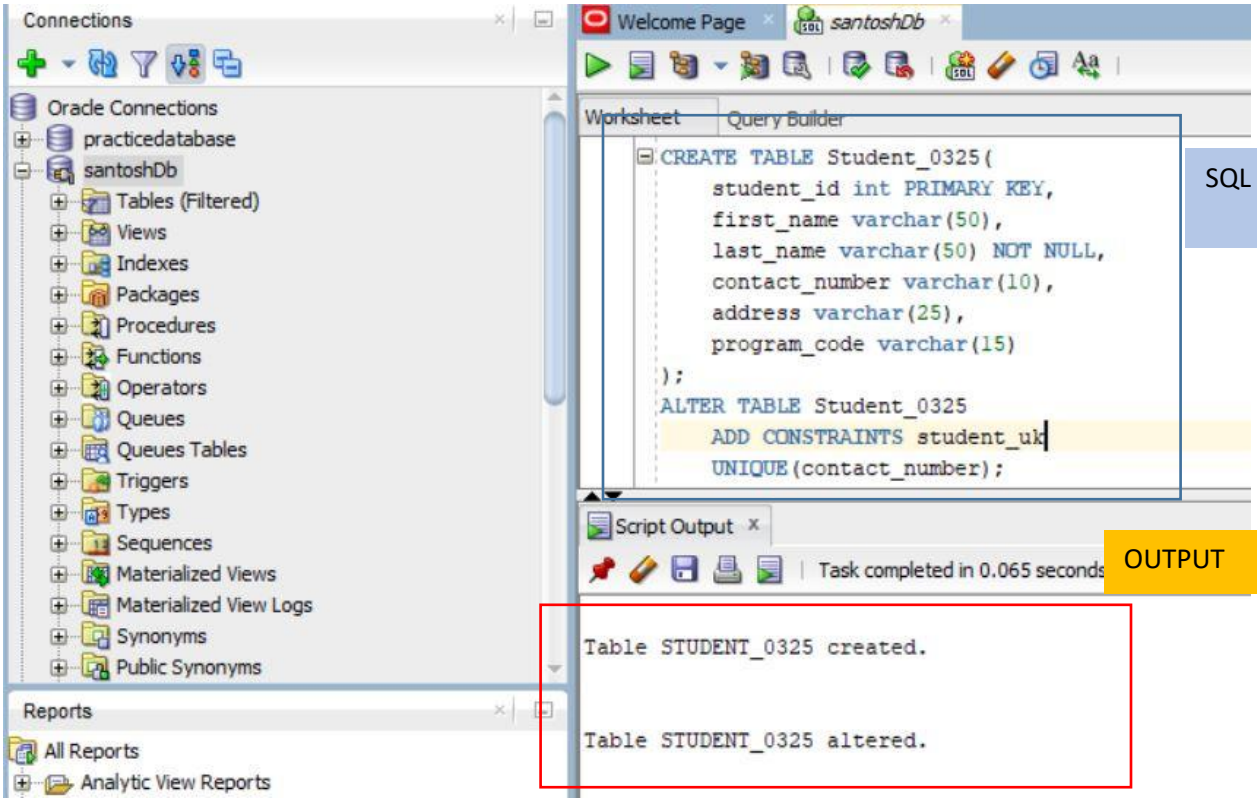
Individual Project

Table Content

PART I

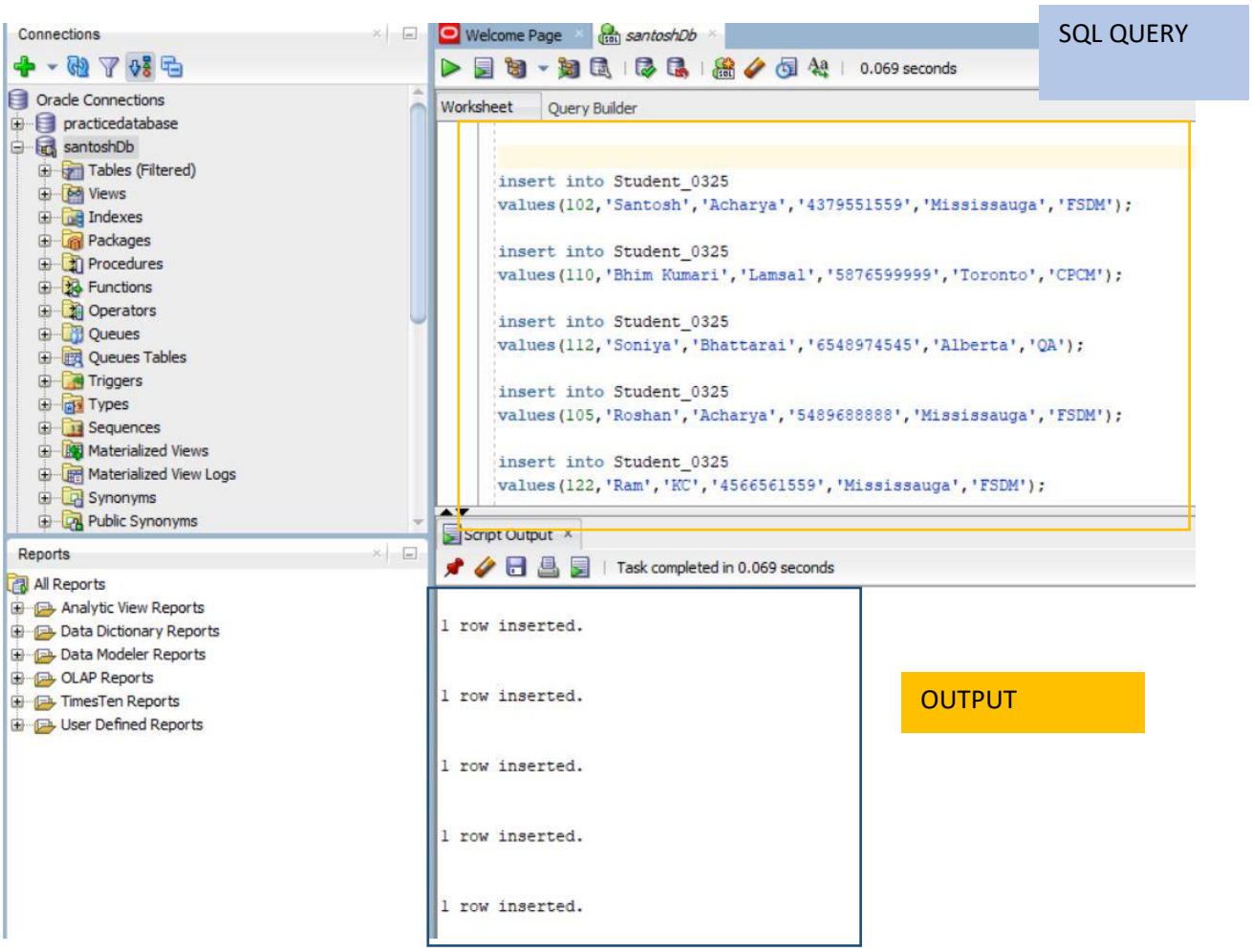
- DDL stands for Data Definition Language.
- DDL is the part of the SQL that helps to create, alerts and delete database.
- DML stands for Data Manipulation Language.
- DML is the SQL that helps to insert, update and delete database.
- CREATE is a syntax in the sql used to create the new table with in the database.
- Alter is use to change the table values after the table is created. It is table level query.
- Column level Constraint are used PRIMARY KEY and NOT NULL
- UNIQUE is used as the table level constraints
- SQL Query:

```
CREATE TABLE Student_0325(
        student_id int PRIMARY KEY,
        first_name varchar(50),
        last_name varchar(50) NOT NULL,
        contact_number varchar(10),
        address varchar(25),
        program_code varchar(15)
);
ALTER TABLE Student_0325
        ADD CONSTRAINTS student_uk
        UNIQUE(contact_number);
```



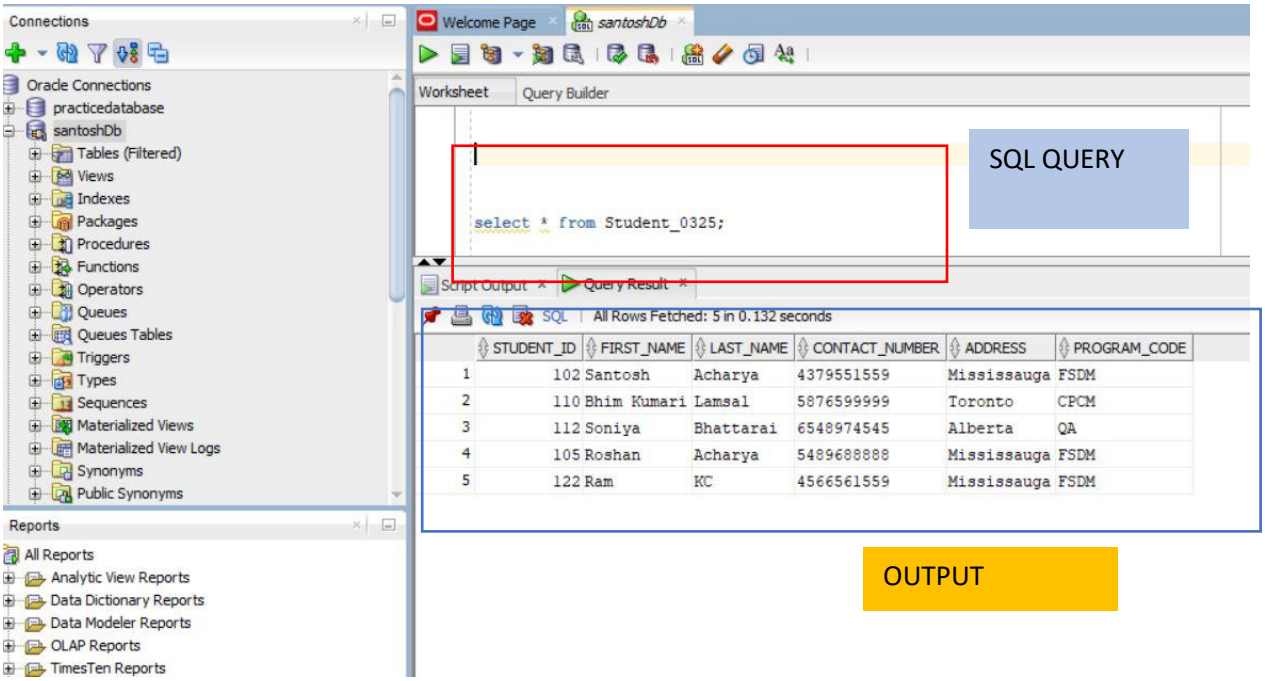
- Insert is the DDL query used to enter the data in the database.
- We can insert the multiple value in the table using the insert.
- SQL Query

```
insert into Student_0325
Values(102,'Santosh','Acharya','4379551559','Mississauga','FSDM');

insert into Student_0325
Values(110,'Bhim Kumari','Lamsal','5876599999','Toronto','CPCM');

insert into Student_0325
Values(112,'Soniya','Bhattarai','6548974545','Alberta','QA');

insert into Student_0325
Values(105,'Roshan','Acharya','5489688888','Mississauga','FSDM');

insert into Student_0325
Values(122,'Ram','KC','4566561559','Mississauga','FSDM');
```



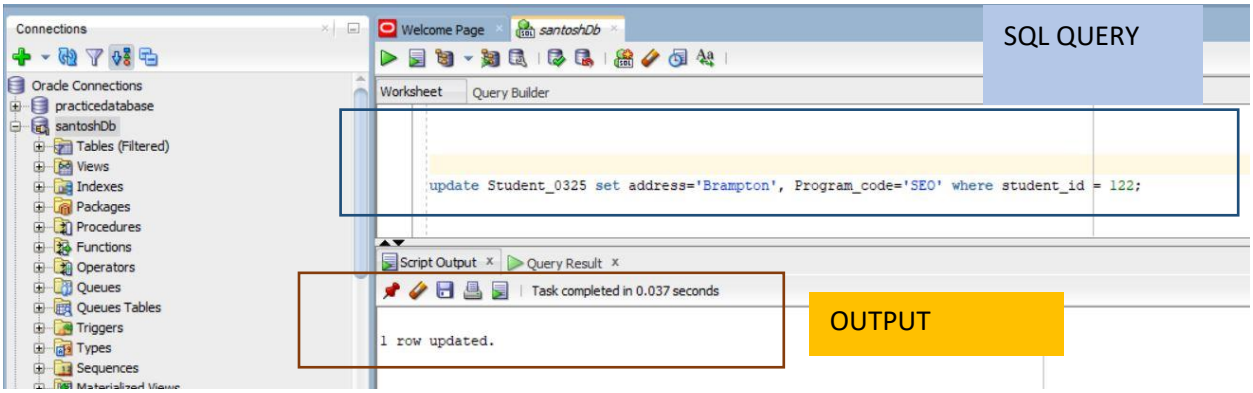
- Select is use to display the value from the table.
- It helps to get the value from the table.
- SQL Query

```
select * from Student_0325;
```



- Update is the query used to update the value in the table.
- It need to specified the value which it want to update with the value that will be updated.
- SQL Query

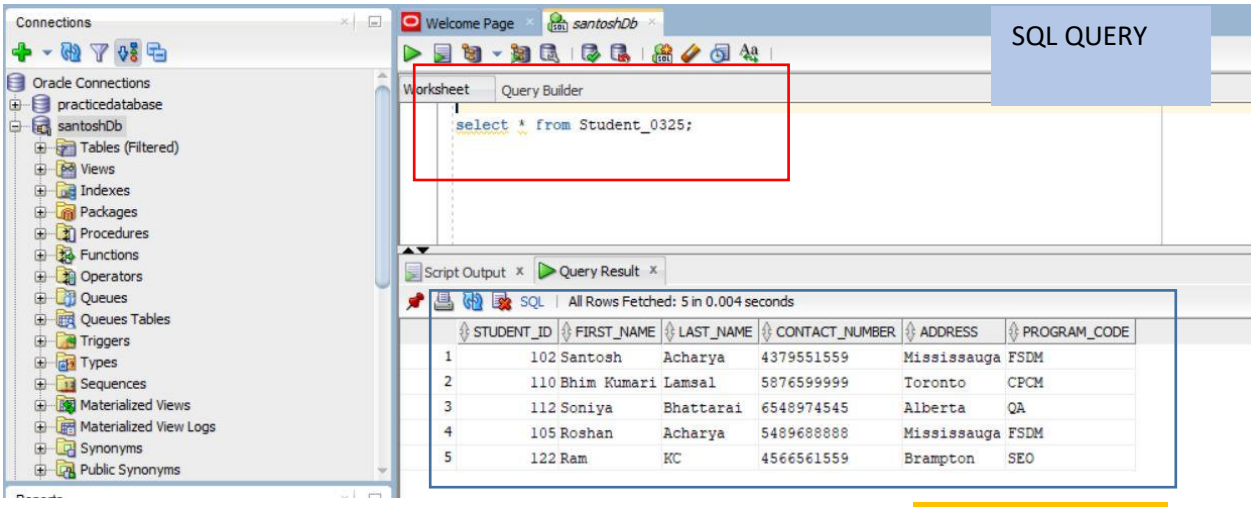
```
update Student_0325 set address='Brampton', Program_code ='SEO' where student_id = 122;
```



View of the table after Update

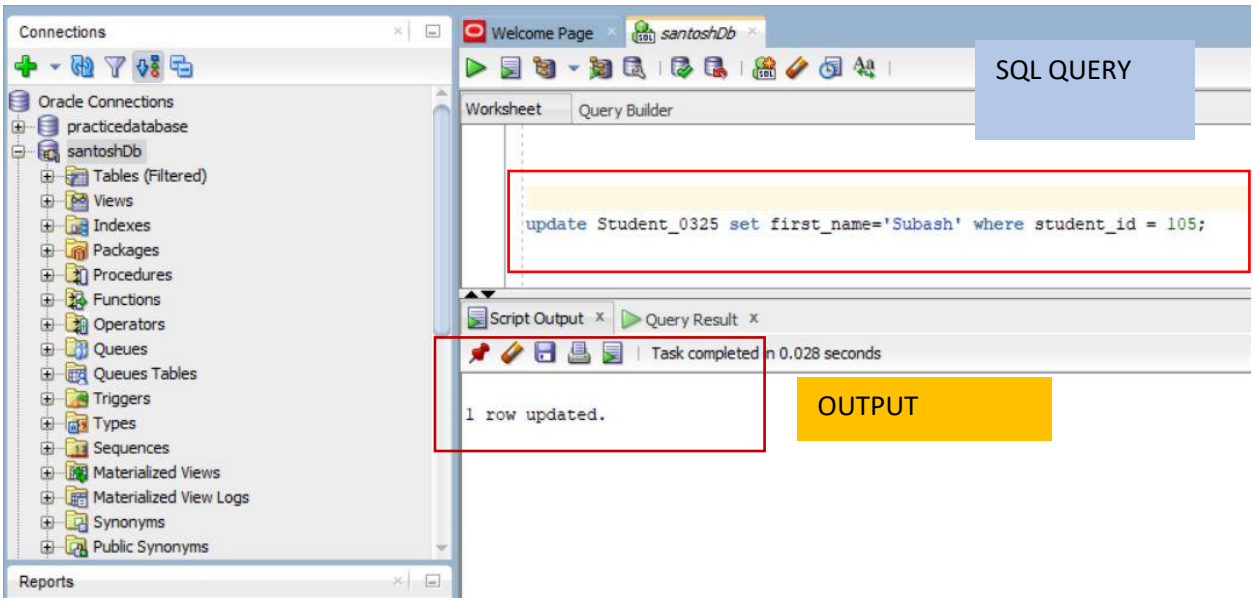
- SQL Query

```
select * from Student_0325;
```



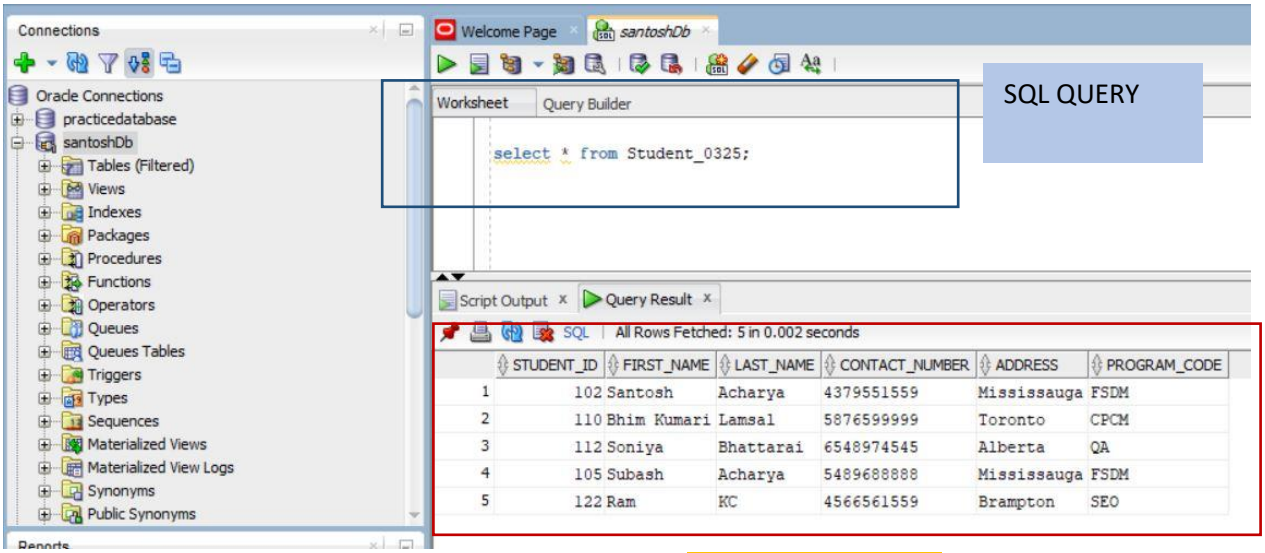
- SQL Query to update next data

```
Update Student_0325 set first_name =’Subash’ where student_id = 105;
```



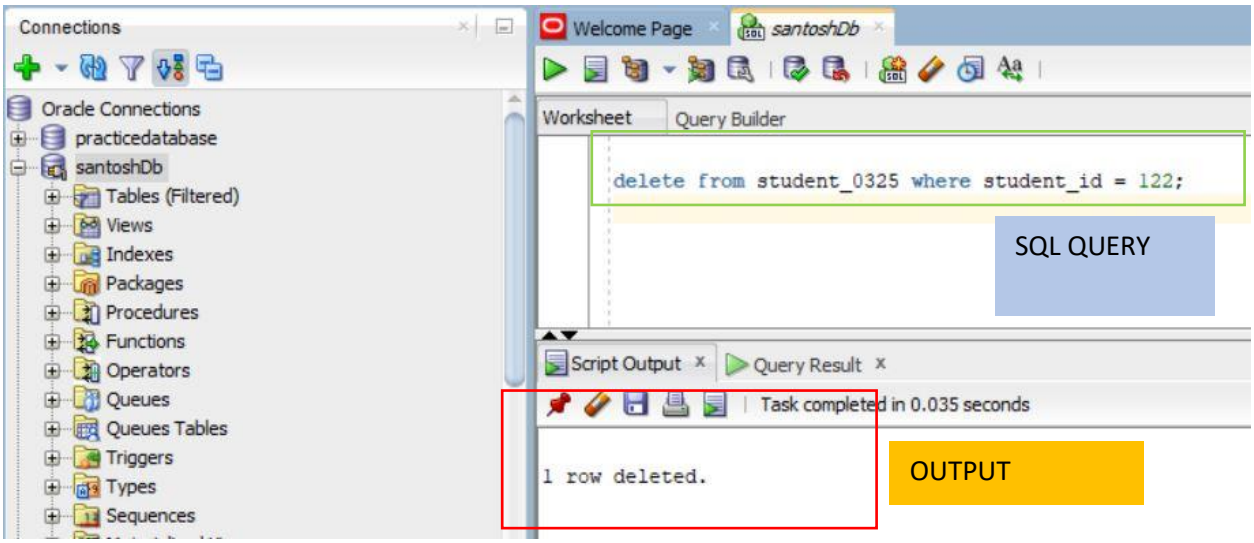
- SQL Query to view the data

Select * from Student_0325;



- Delete is use to remove the value from the table in the SQL query.
- Delete can be used to remove the single or multiple data in the table.
- SQL Query

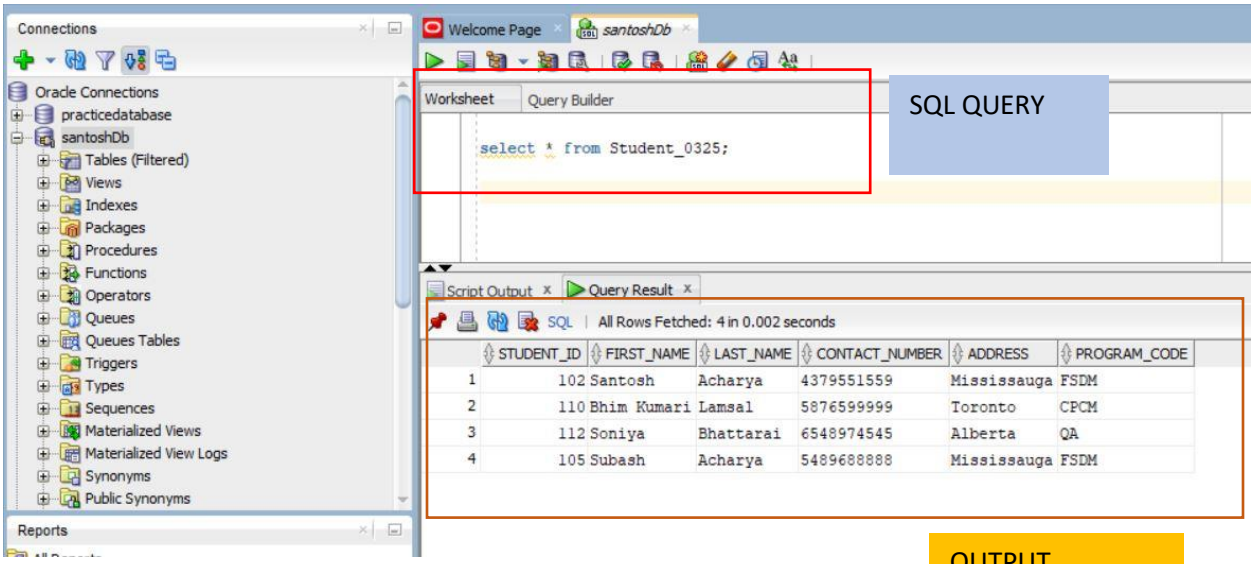
Delete from student_0325 where student_id = 122;



View of the table

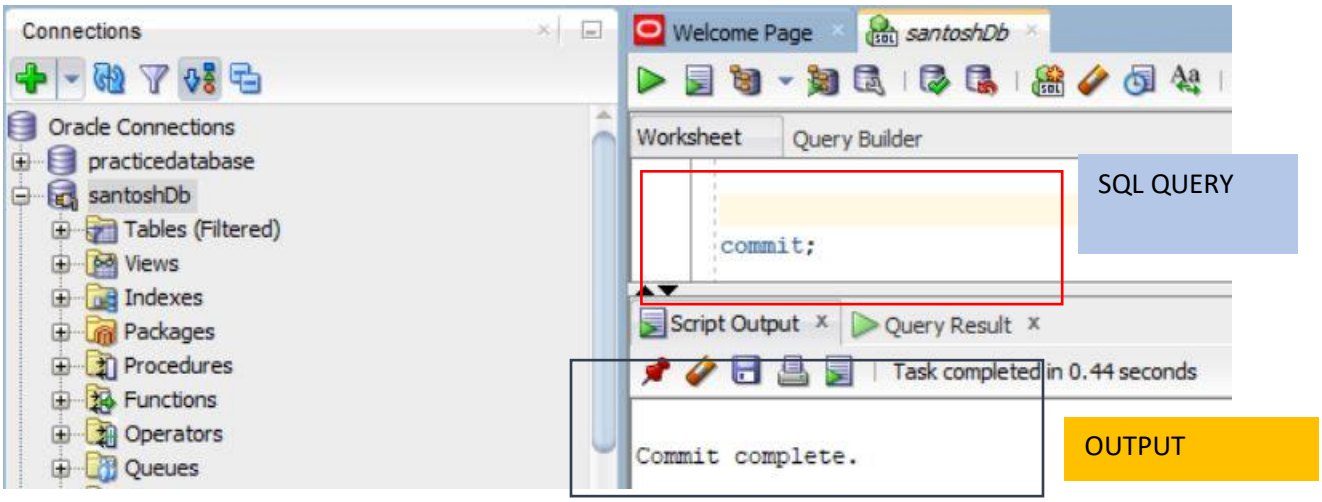
- SQL Query

Select * from Student_0325;



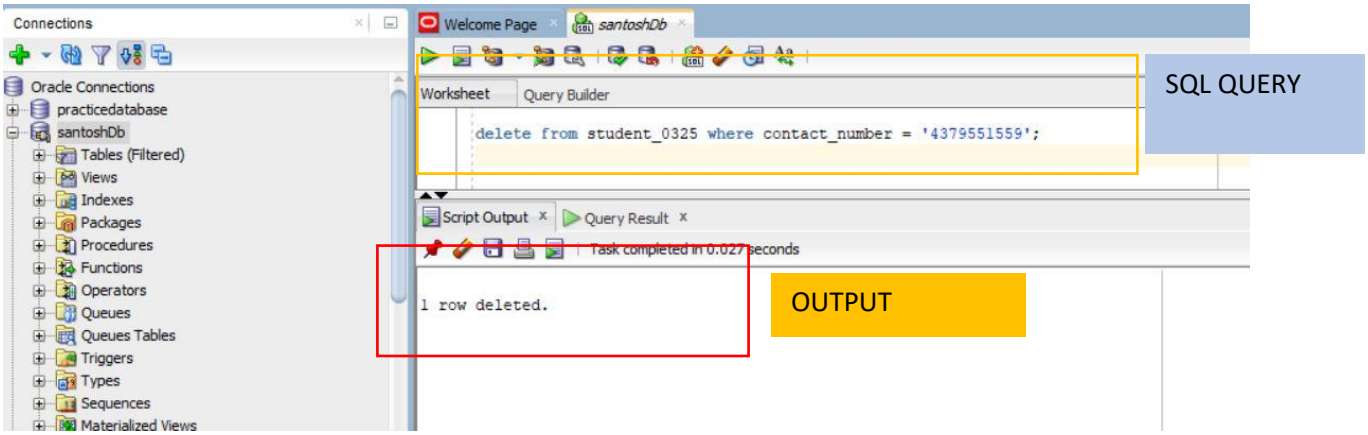
- Commit is use to save the checkpoint in the table
- It is a point where the table will rollback in the sql.
- After deleting one data commit was used to save the table.
- SQL Query

Commit;



❖ SQL Query

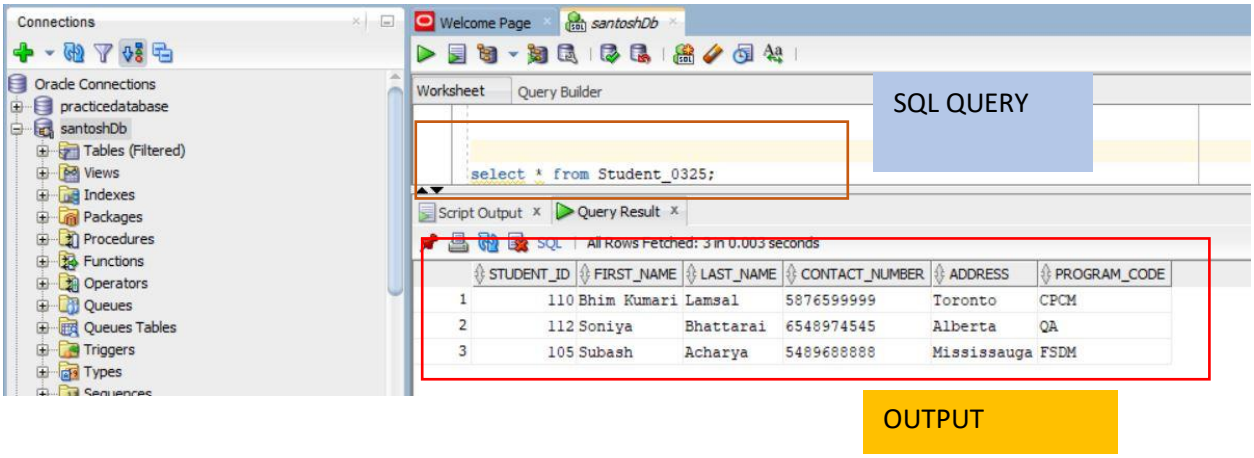
Delete from student_0325 where contact_number = '4379551559';



View table after delete

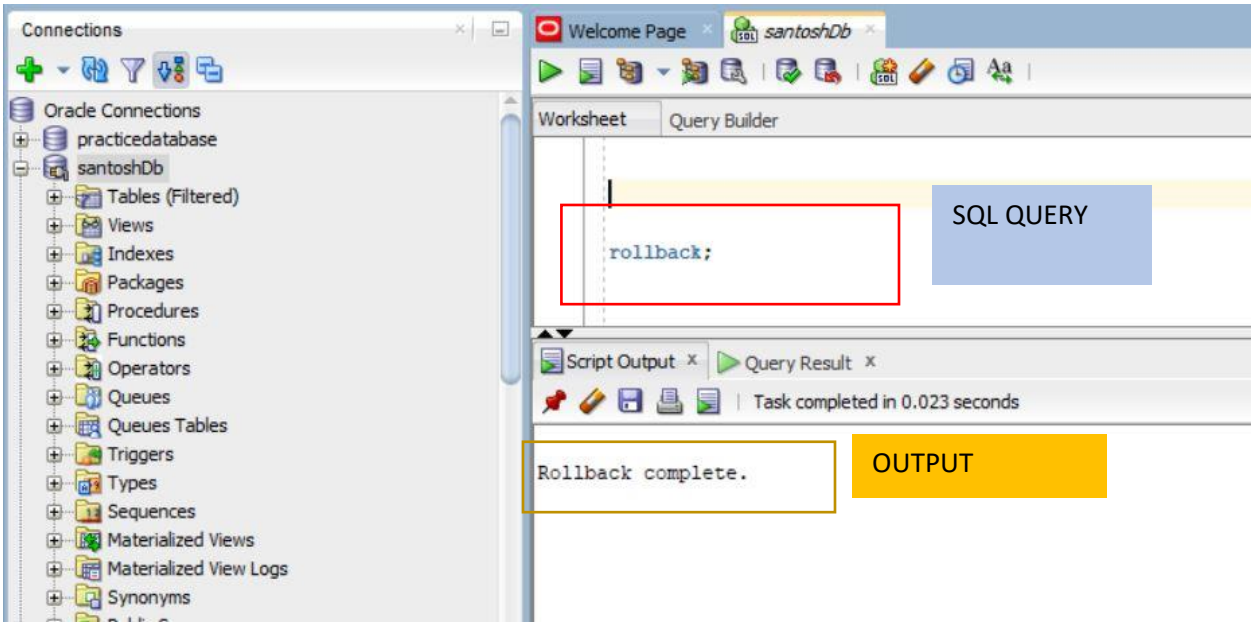
❖ SQL Query

Select * from Student_0325



- Rollback is use to move the table to the previous commit point.
- It helps to recover the data in case there is any error.
- SQL Query

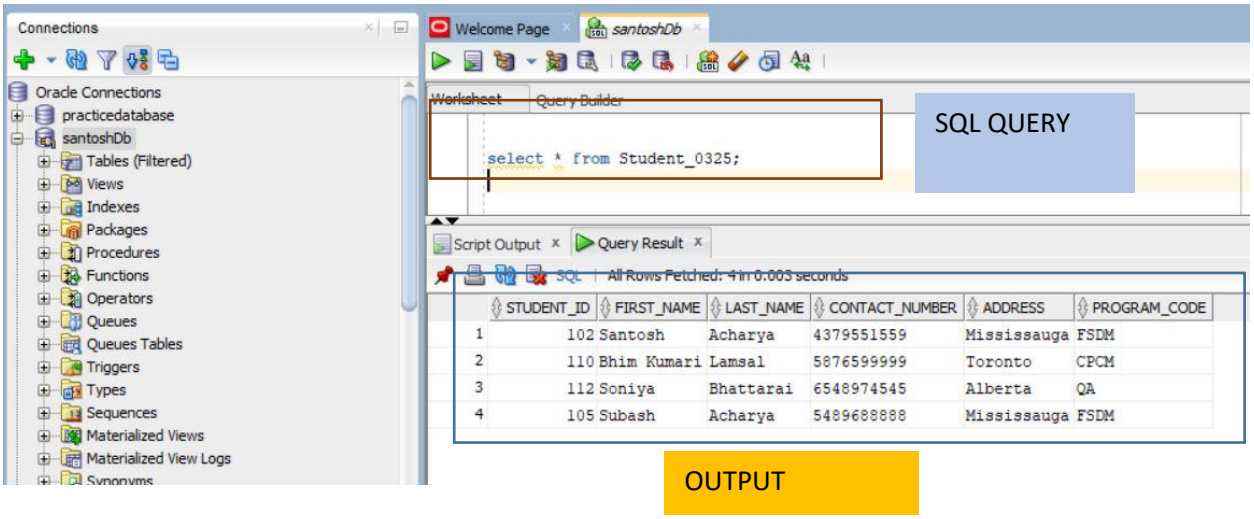
Rollback;



View after the rollback

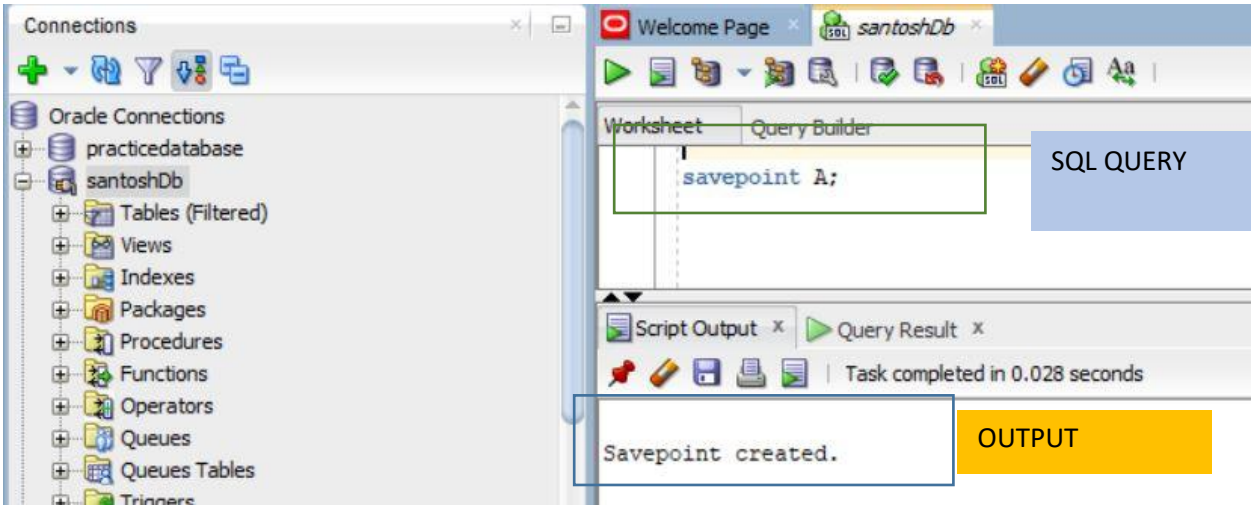
- SQL Query

Select * from Student_0325;



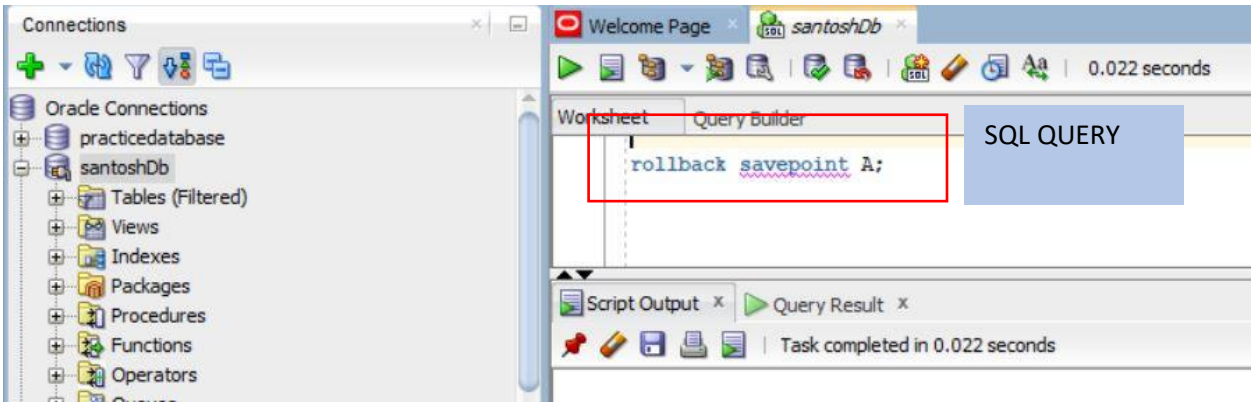
- Savepoint is similar to commit that is the check point that is used to save the different point in the table.
- SQL Query:

Savepoint A;



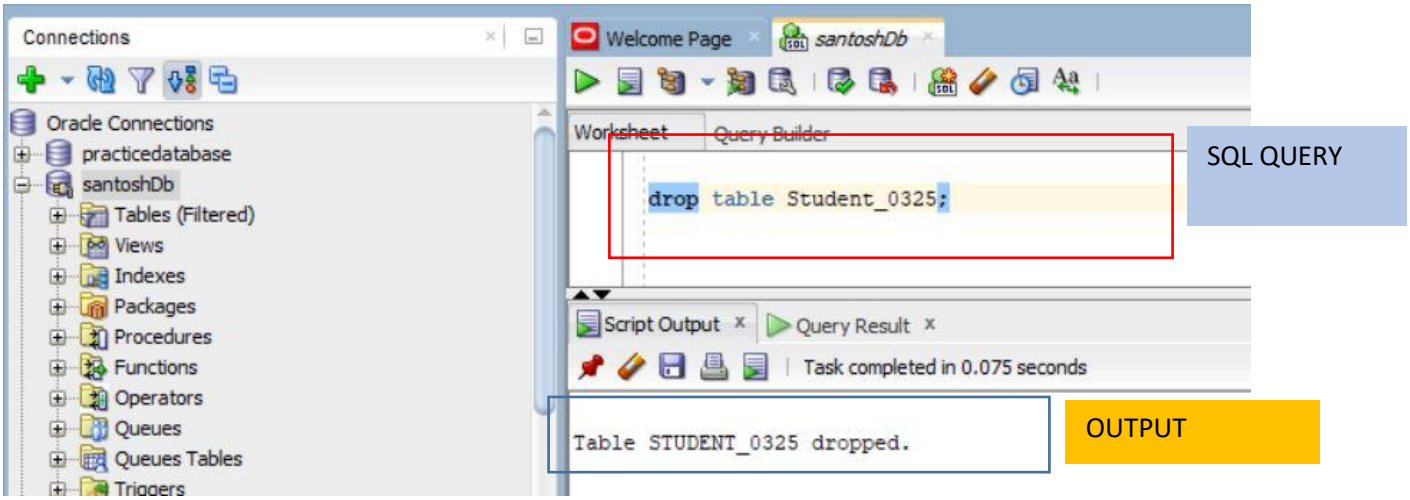
- SQL Query to rollback

Rollback savepoint A;



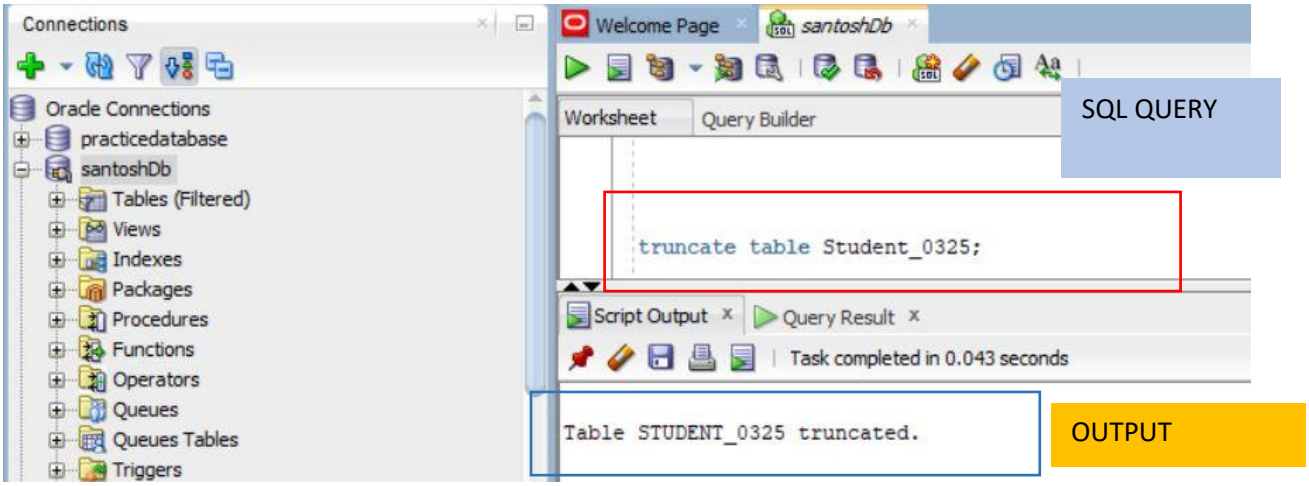
- Drop is used to delete the whole table from the database.
- It will not only remove the data but also the all structure of the table in the database.
- SQL Query

Drop table Student_0325;



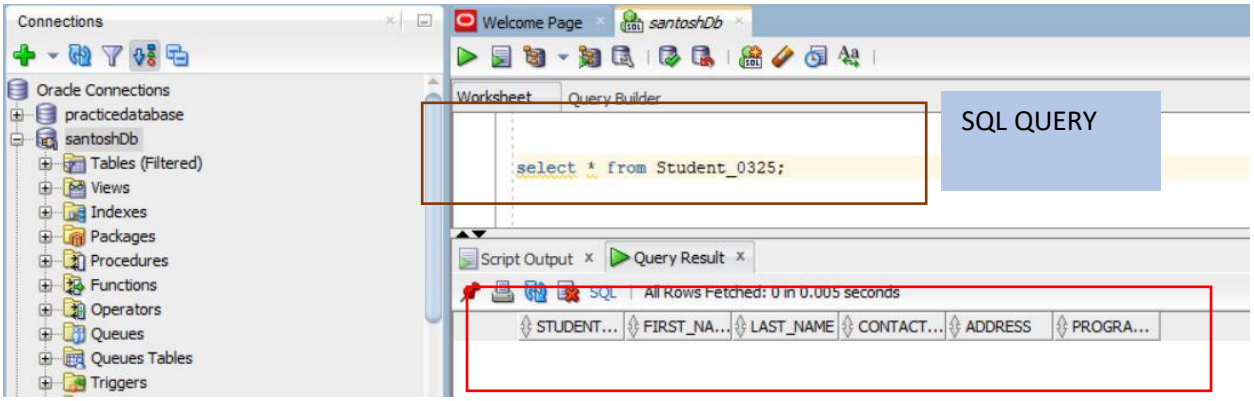
- Truncate is the query which will remove the data from the table but not the structure of the table.
- It will remove all the data.

Truncate table Student_0325



View After the Truncate Query

Select * from Student_0325;



PART II

ALL table that has been used in the Query

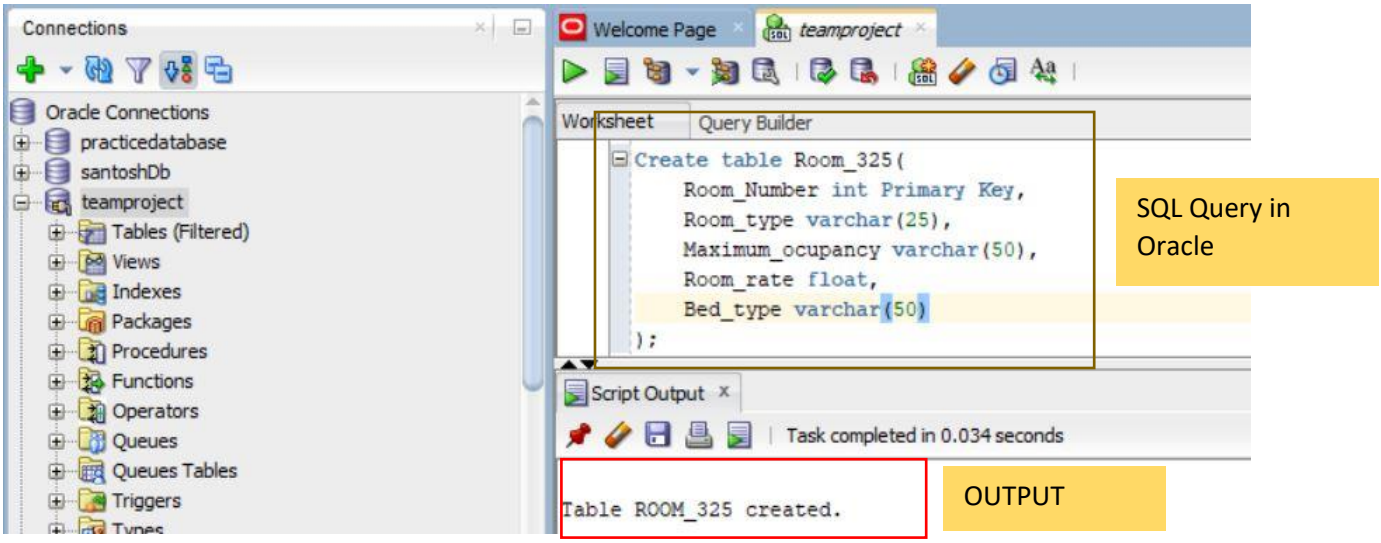
- The Create is use to create the table.
- Create is the sql query used to add the new table in the database.
- It is DDL based query in the oracle.

- SQL Query

```
Create table Room_325(
   Room_Number int Primary Key,
   Room_type varchar(25),
   Maximum_ocupancy varchar(50),
   Room_rate float,
   Bed_type varchar(50)
);
```

- SQL Query Explanation

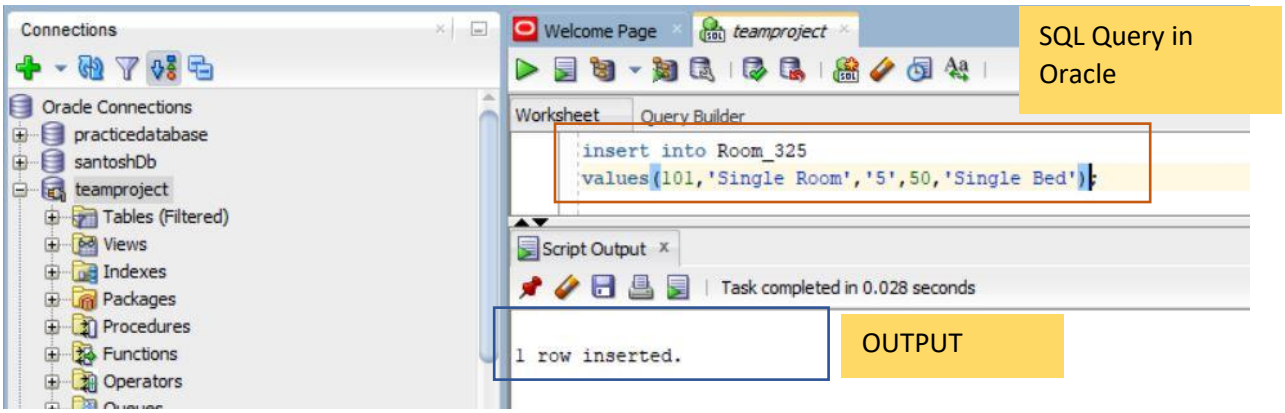
Create is used to make the table name Room_325 with Room_number as the primary key.

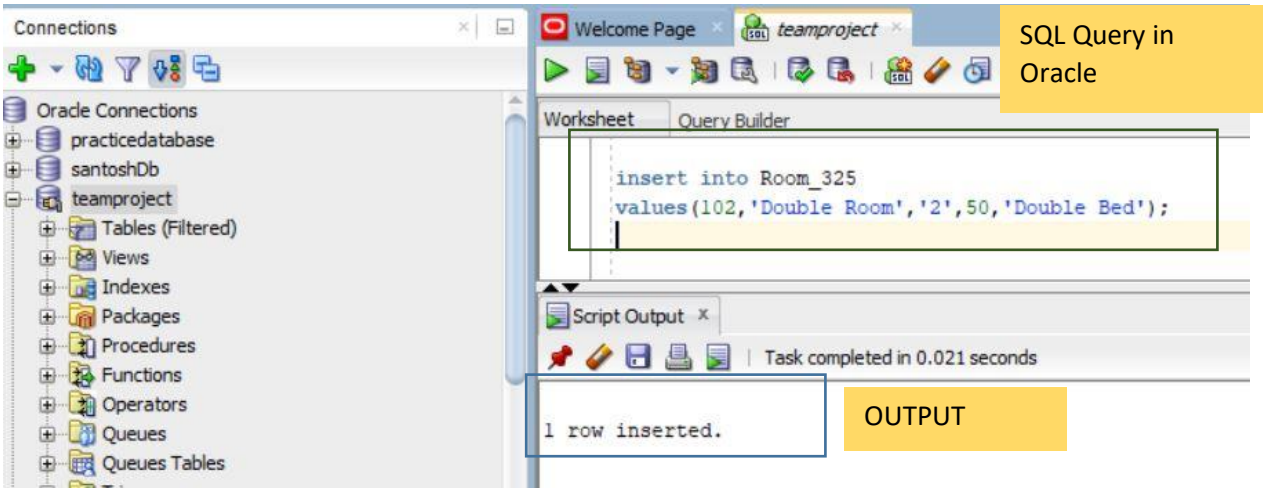


- Inserted into Room
- Insert is used enter the data into the database.
- It is DDL based query

- SQL Query

```
insert into Room_325
values(101,'Single Room','5',50,'Single Bed');
insert into Room_325
values(102,'Double Room','2',50,'Double Bed');
```





Customer_325 table

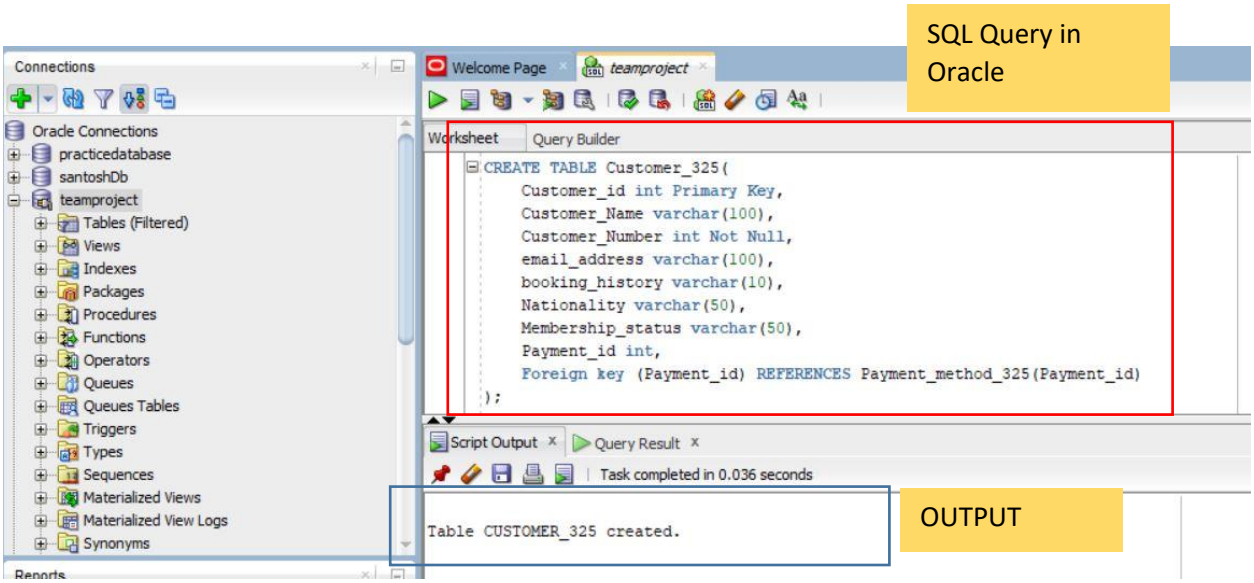
- ❖ Foreign key is used to get the column that is been assigned to the other table in the same database.
- ❖ References is used to refer the table from where the foreign key is selected.

- ❖ SQL Query Explanation :

Customer_325 with the primary key customer_id , customer_number is also not null and Payment_id is the foreign key used from the payment_method_325.

- ❖ SQL Query

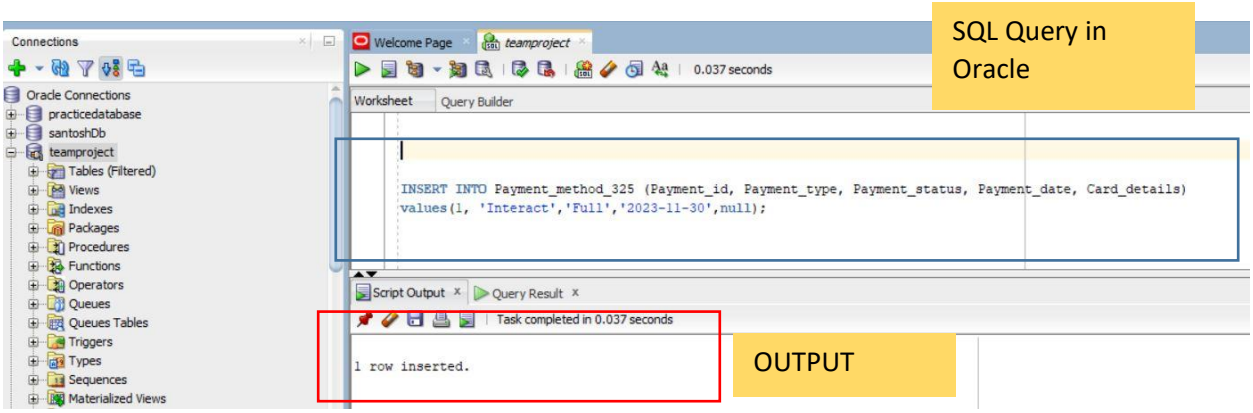
```
CREATE TABLE Customer_325(
   Customer_id int Primary Key,
   Customer_Name varchar(100),
   Customer_Number int Not Null,
   email_address varchar(100),
   booking_history varchar(10),
   Nationality varchar(50),
   Membership_status varchar(10),
   Payment_id int,
   Foreign key (Payment_id) REFERENCES Payment_method_325(Payment_id)
);
```



Inserted into Customer_325

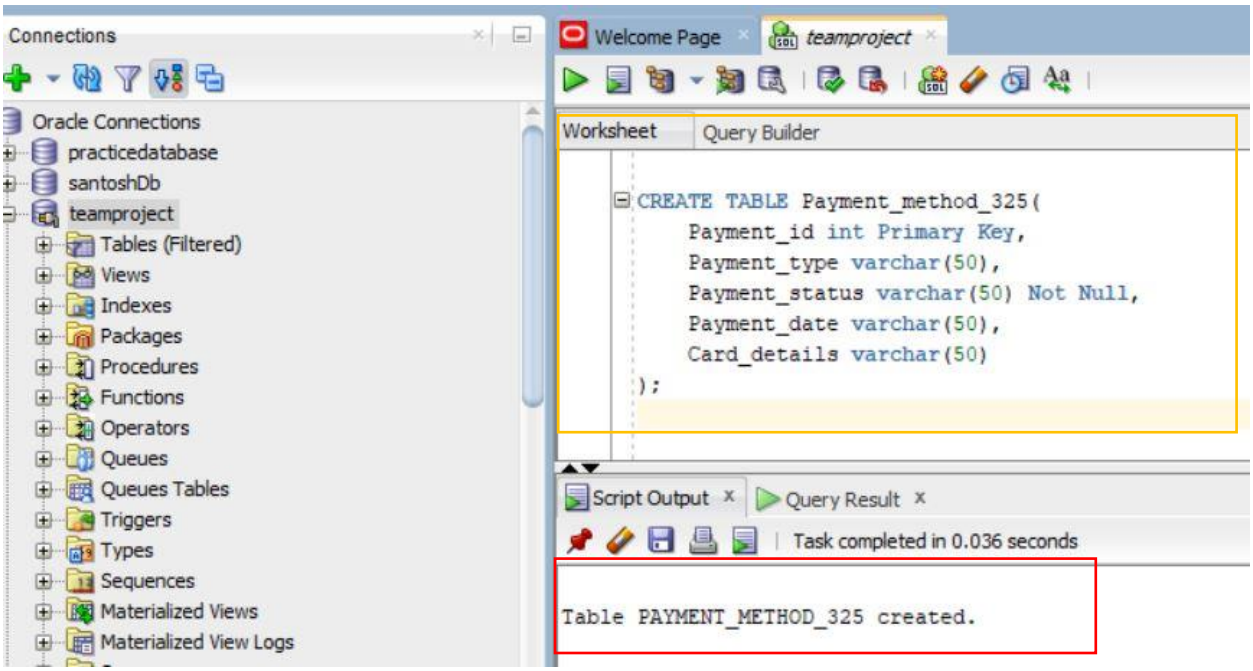
- Valued are inserted according to the data types that has been created using the table.
- SQL Query

```
insert into Customer_325
values(201,'Santosh Acharya',4379551559,'acharyaanish16@gmail.com',null,'Nepali','Full Member',1);
```



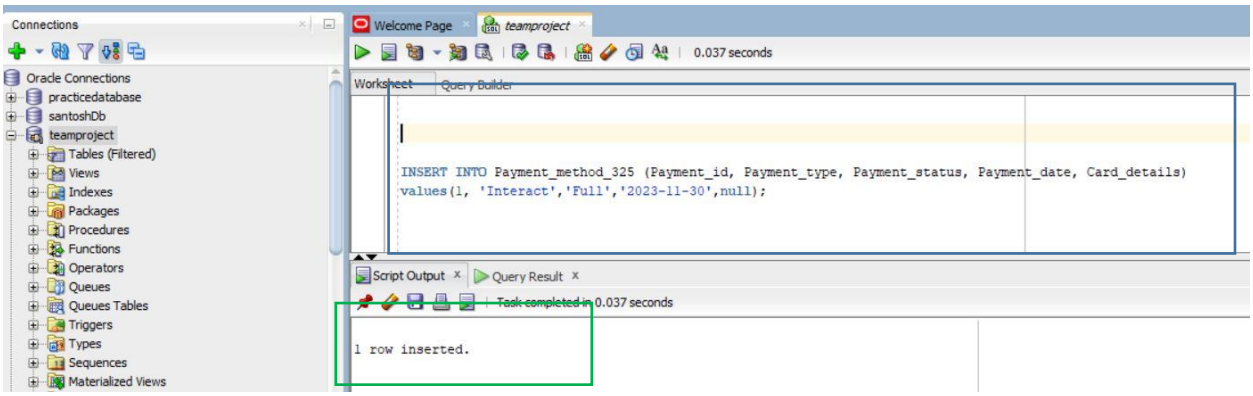
Payment_method_325

```
CREATE TABLE Payment_method_325(
   Payment_id int Primary Key,
   Payment_type varchar(50),
   Payment_status varchar(50) Not Null,
   Payment_date date,
   Card_details varchar(50)
);
```

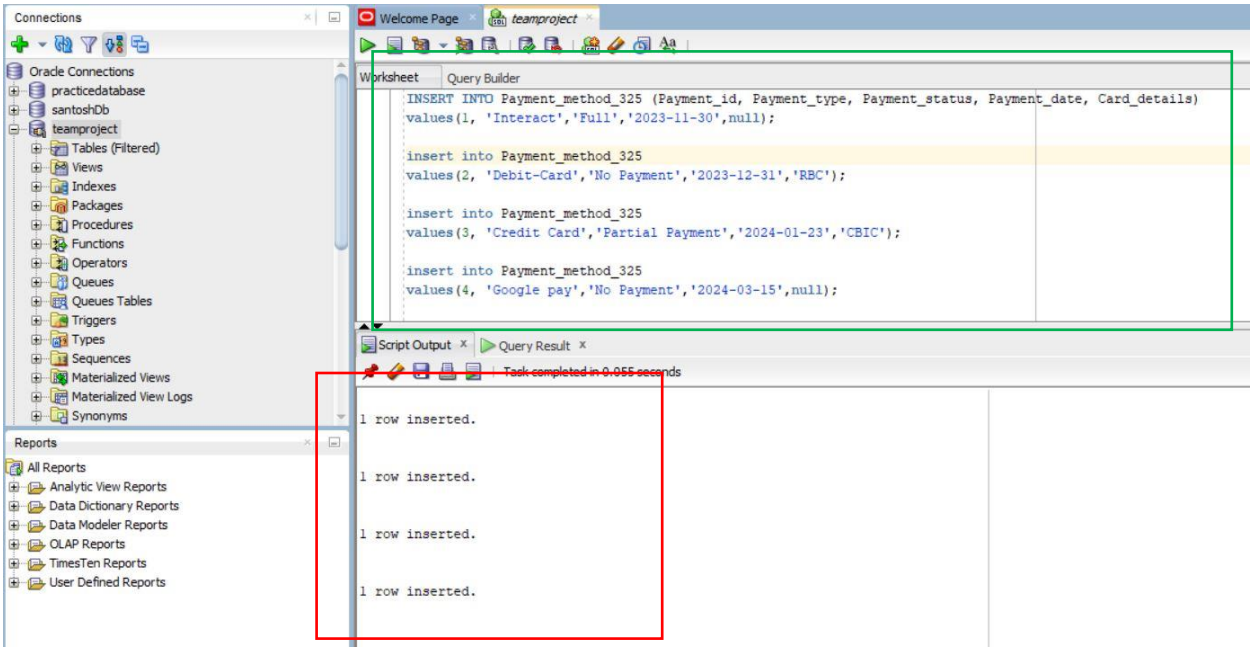


Inserted into payment_method_325

```
insert into Payment_method_325
values(1, 'Interact','Full Payment','2023-11-31',null);
```



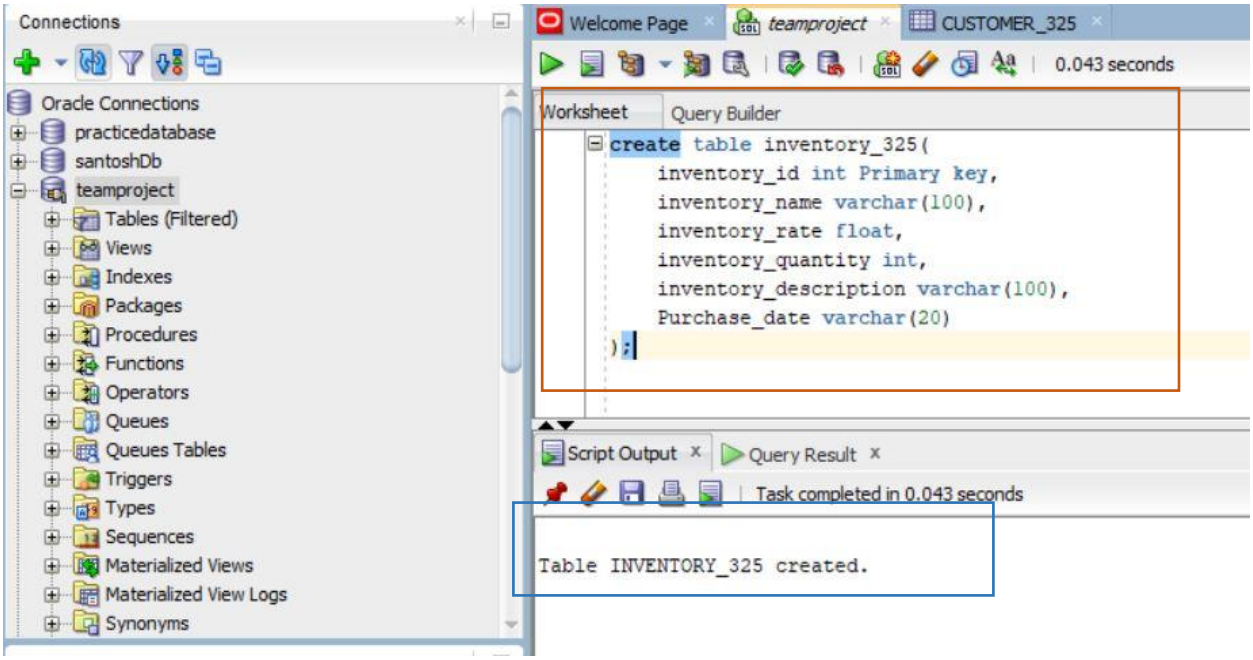

```
insert into Payment_method_325
values(2, 'Debit Card','No Payment','2023-12-31','RBC');
insert into Payment_method_325
values(3, 'Credit Card','Partial Payment','2024-01-23','CBIC');
insert into Payment_method_325
values(4, 'Google pay','No Payment','2024-03-15',null);
insert into Payment_method_325
values(5, 'Bank Transfer','Partial Payment','2024-01-23','TD');
```



Inventory_325

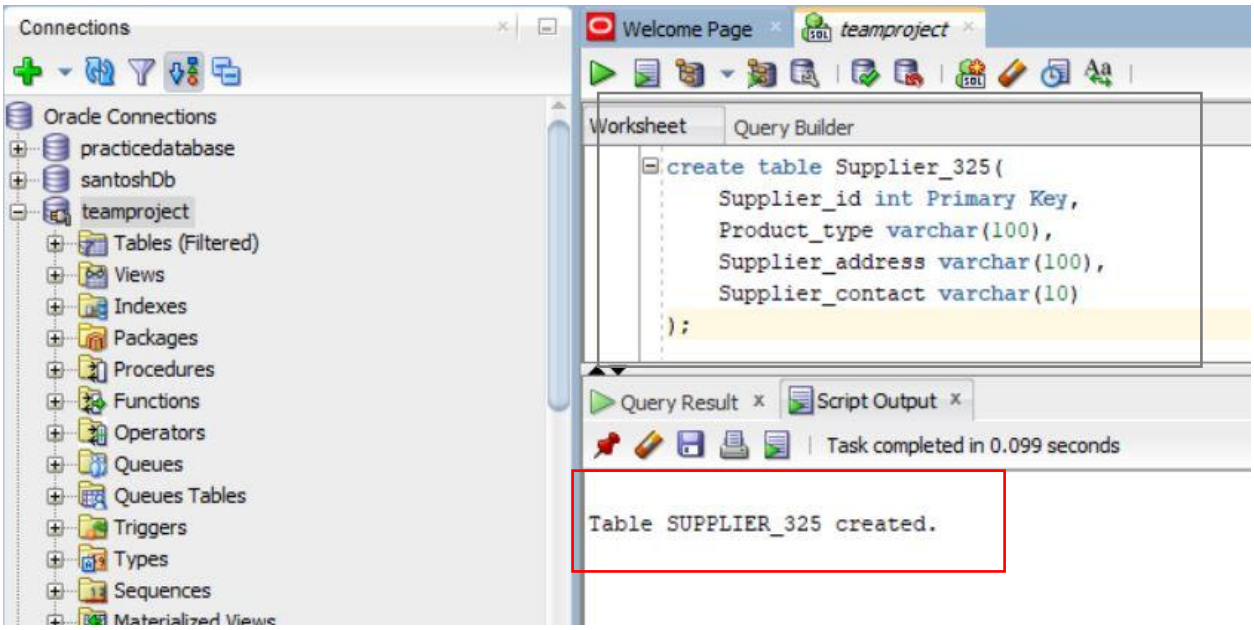
- ❖ It is the table used to store the inventory related information in the hotel reservation system.
- ❖ SQL Query :

```
create table inventory_325(
        inventory_id int Primary key,
        inventory_name varchar(100),
        inventory_rate float,
        inventory_quantity int,
        inventory_description varchar(100),
        Purchase_date varchar(100)

);
```



Supplier_325

```
create table Supplier_325(
        Supplier_id int Primary Key,
        Product_type varchar(100),
        Supplier_address varchar(100),
        Supplier_contact varchar(10)
);
```



Category A

1

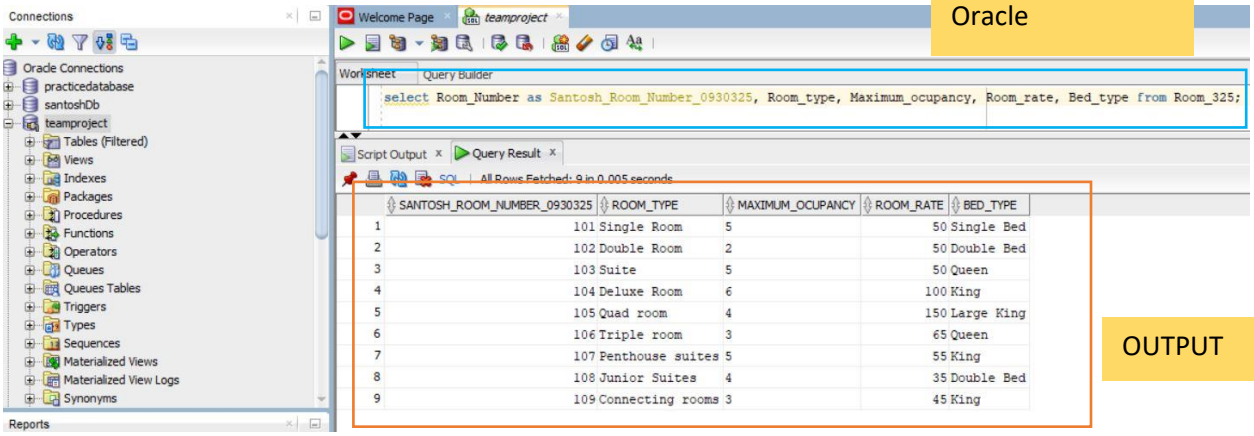
- Select is the query used to view the data in the table. It is the used to select and view the particular table.
- SQL Query

```
select Room_Number as Santosh_Room_Number_0930325, Room_type, Maximum_ocupancy, Room_rate, Bed_type from Room_325;
```

- SQL Query Explanation

In the sql query room_number, room type, maximum_ocupancy, room_rate, bed_type is selected from the room_325 table.

SQL Query in Oracle



OUTPUT

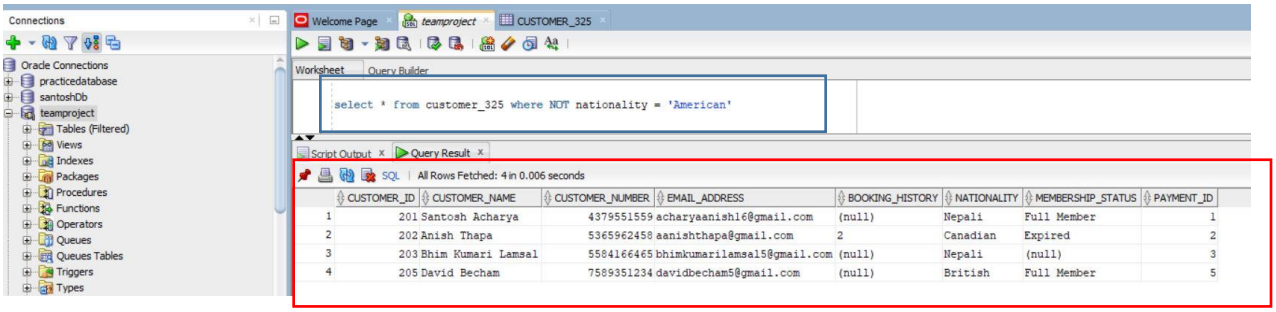
2

- NOT is the operator used to denote the select query that will select all other value except the assigned value.
- SQL Query :

```
Select * from Customer_325 where NOT Nationality = “American”
```

- SQL Query Explanation :

In the query the table will show the value except the value that contain Nationality = “American”

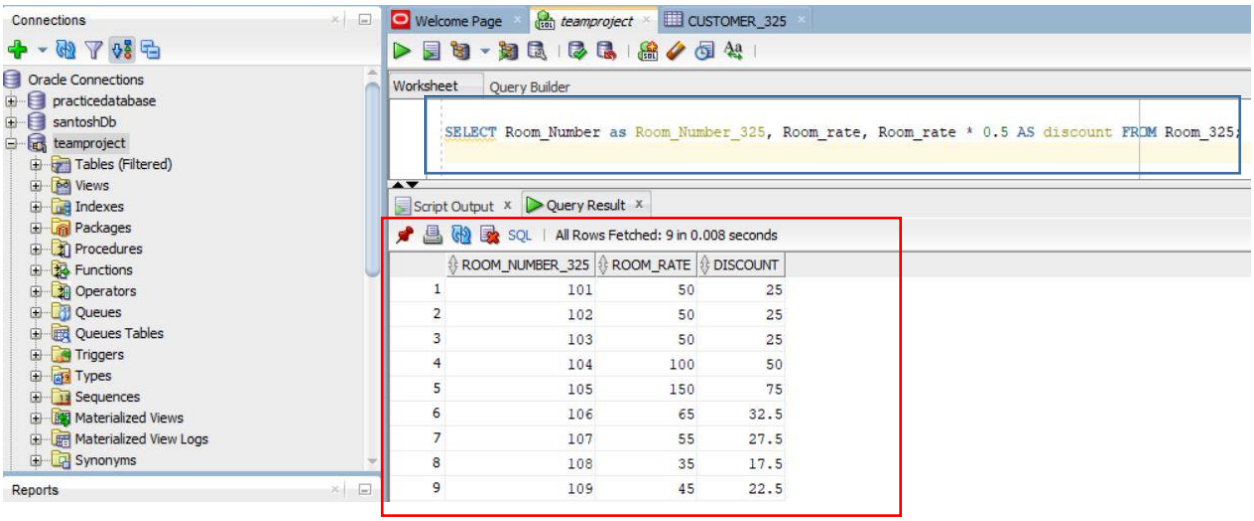


- Multiplication is a mathematical operation used to multiply numeric values, typically within SELECT statements or as part of UPDATE or INSERT statements.
- SQL Query:

```
SELECT Room_Number as Room_Number_325, Room_rate, Room_rate * 0.5 AS discount FROM Room_325;
```

- SQL Query Explanation:

In the query the multiplication is used to get the discount rate of 50% to the room rate from room_325



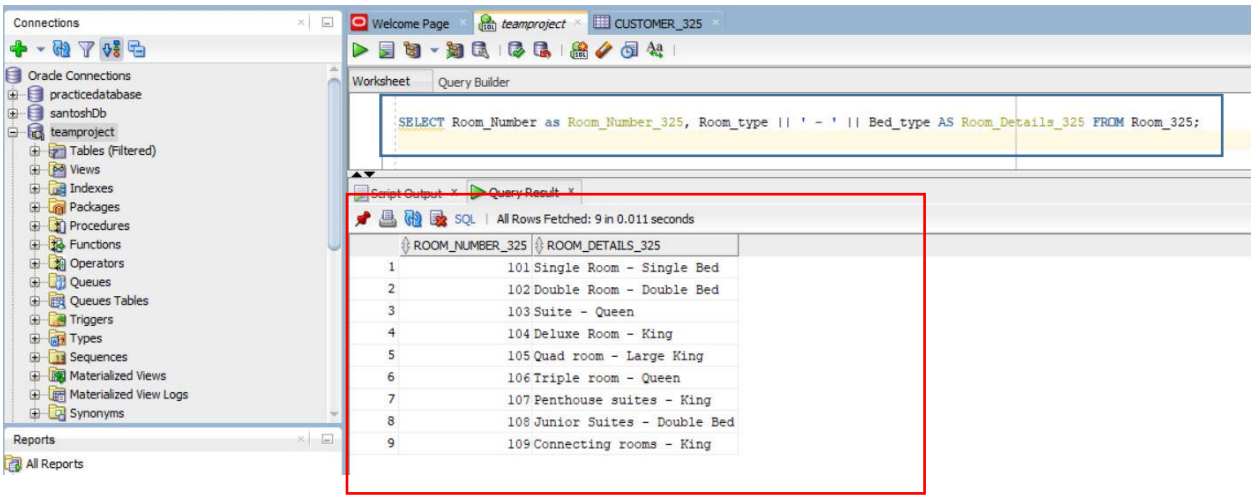
4

- Concatenation is used to combine the string together. It is useful to join the string
- SQL Query :

```
SELECT Room_Number as Room_Number_325, Room_type || ' - ' || Bed_type AS
Room_Details_325 FROM Room_325;
```

- SQL query explanation:

In the query, room type and bed type is connected with each other with the '– ' sign.



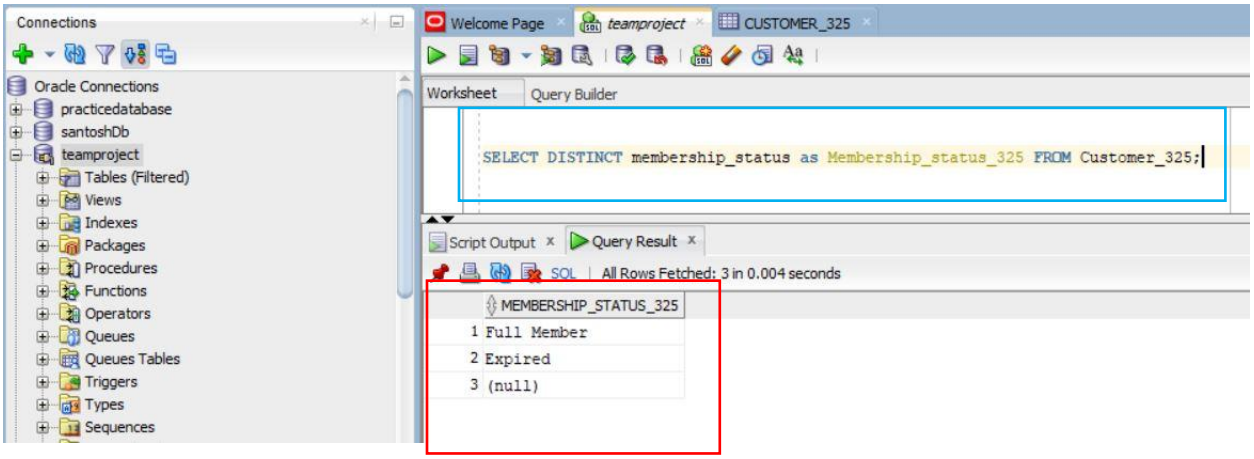
5

- DISTINCT is use to return the different value.
- SQL Query

```
SELECT DISTINCT membership_status as Membership_status_325 FROM Customer_325;
```

- SQL Explanation:

In the query, the membership status with different value is selected.



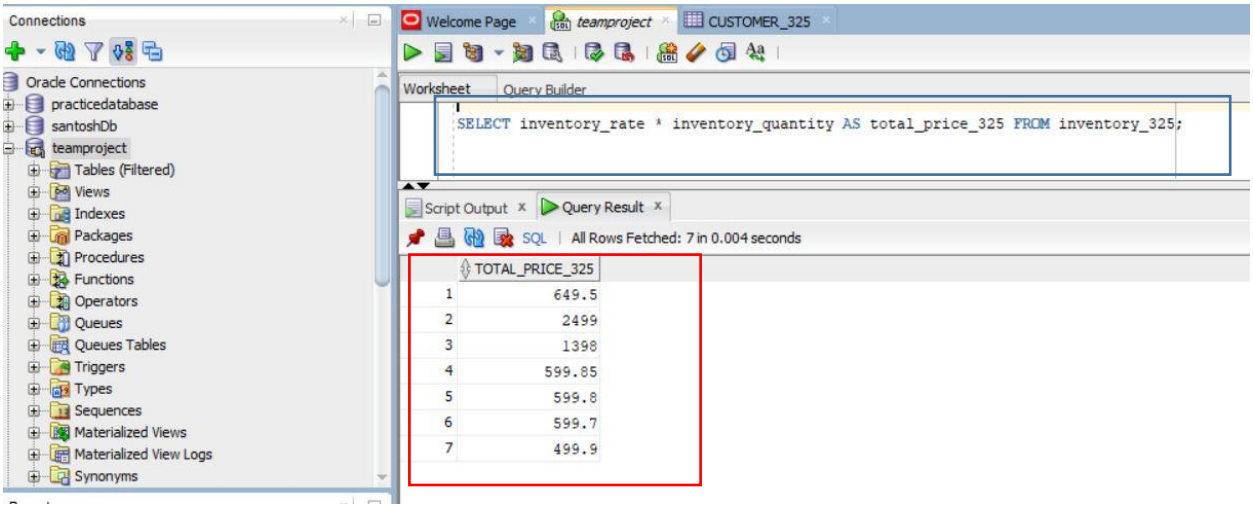
6

- Addition, Subtraction , Multiplication is the mathematical operand used with the select in the sql query.
- SQL Query

```
Select inventory_rate * inventory_quantity as total_price_325 from inventory_325;
```

- SQL Query Explanation:

In the query the inventory rate is multiplied with the inventory quantity to get the total_price from the inventory_325 table.



Category B

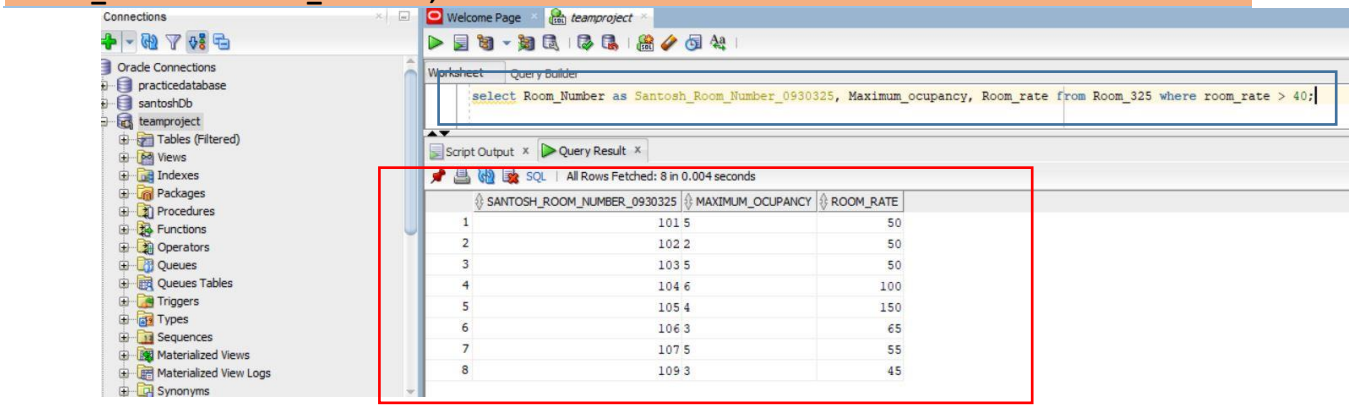
1

- ❖ Comparison operator helps to know the certain condition in the database and select the data according to the condition.
- ❖ Greater than sign will compare the value and determine if the value is greater than the given condition
- ❖ SQL query explanation :

  In the query the room rate greater than 40 will be selected from the room_325 table.

- ❖ SQL Query

```
select Room_Number as Santosh_Room_Number_0930325, Maximum_ocupancy, Room_rate from Room_325 where room_rate > 40;
```



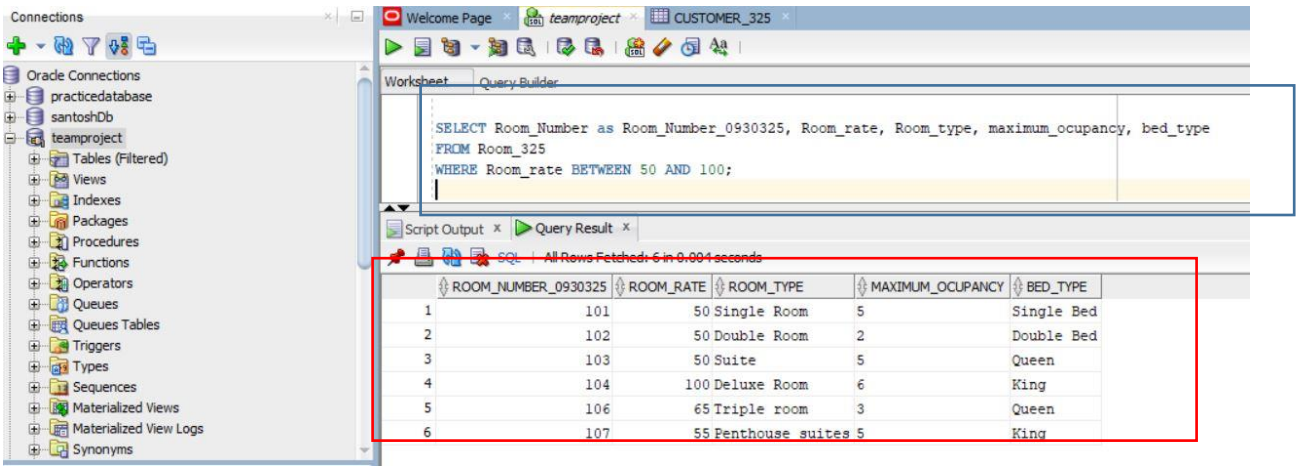
2

- ❖ Between Operands will help to get the value within a given range.
- ❖ SQL Query

```
SELECT Room_Number as Room_Number_0930325, Room_rate, Room_type, maximum_ocupancy, bed_type FROM Room_325 WHERE Room_rate BETWEEN 50 AND 100;
```

- ❖ SQL Query explanation:

In the query the Room Number, Room rate, Room type , maximum occupancy and bed type is selected from the room_325 table where room rate is between 50 and 100.



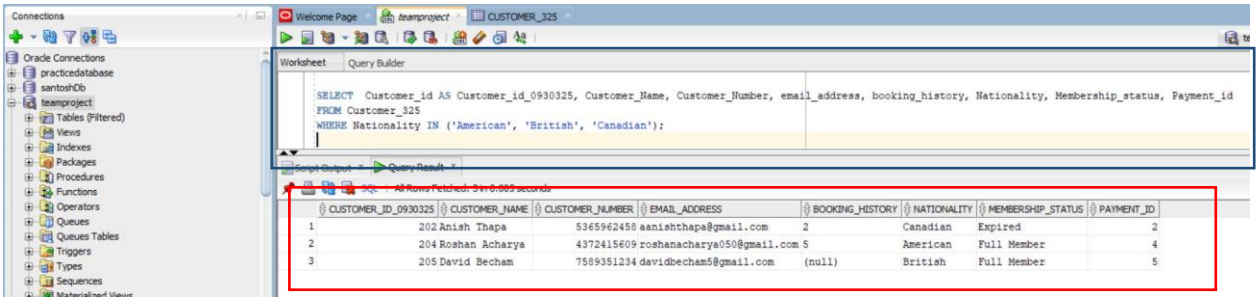
3

- IN operands helps to specify multiple values in a Where clause. It is similar of using multiple OR condition
- SQL Query

```
SELECT  Customer_id AS Customer_id_0930325, Customer_Name, Customer_Number,
email_address, booking_history, Nationality, Membership_status, Payment_id
FROM Customer_325
WHERE Nationality IN ('American', 'British', 'Canadian');
```

- SQL Explanation :

In the query the customer id, customer name, customer number, email address, booking history , membership status and payment id is selected when nationality is either American, british or Canadian.



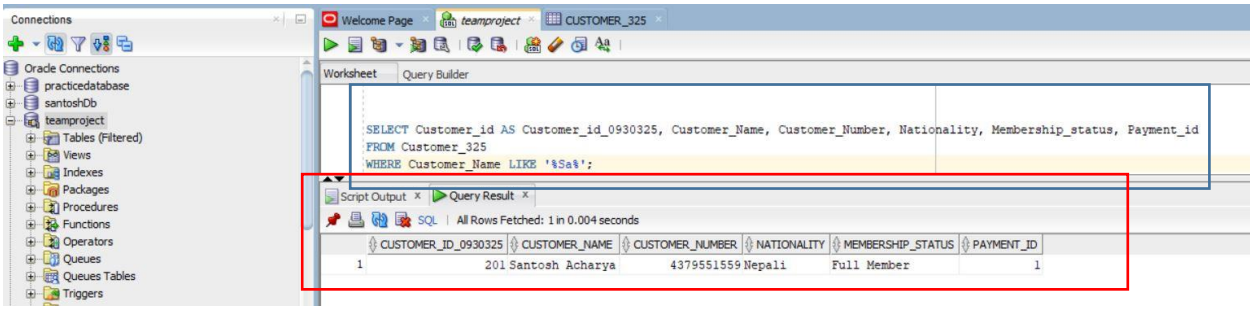
4

- Like operand is used to search the similar pattern in the table. Percentage % and Underscore _ are often used with like operand.
- SQL Query

```
SELECT Customer_id AS Customer_id_0930325, Customer_Name, Customer_Number, Nationality,
Membership_status, Payment_id
FROM Customer_325
WHERE Customer_Name LIKE '%Sa%';
```

- SQL Explanation:

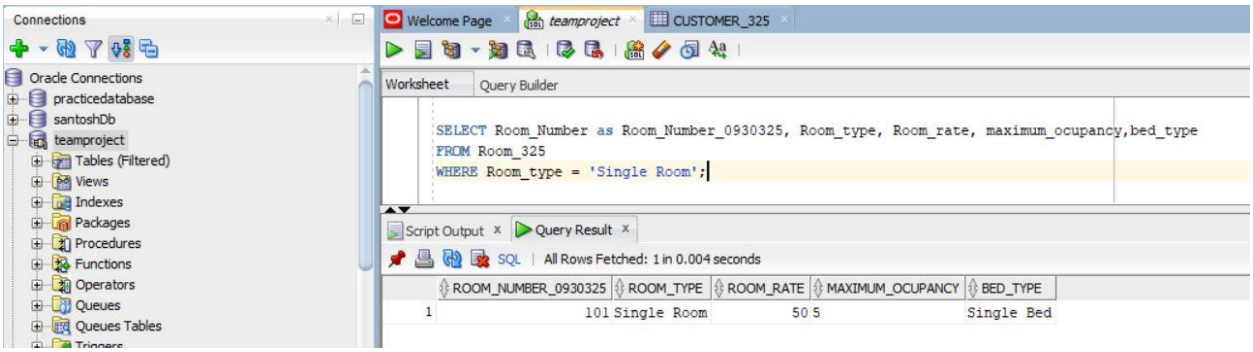
In the query, customer id,customer name, customer number, nationality, membership status and payment id is selected from the customer_325 where customer name is like '%Sa%'.



5

- = sign is used to compare the exact value within the table.
- It will provide the value that match the condition.
- SQL Query

```
SELECT Room_Number as Room_Number_0930325, Room_type, Room_rate,
maximum_ocupancy,bed_type
FROM Room_325
WHERE Room_type = 'Single Room';
```

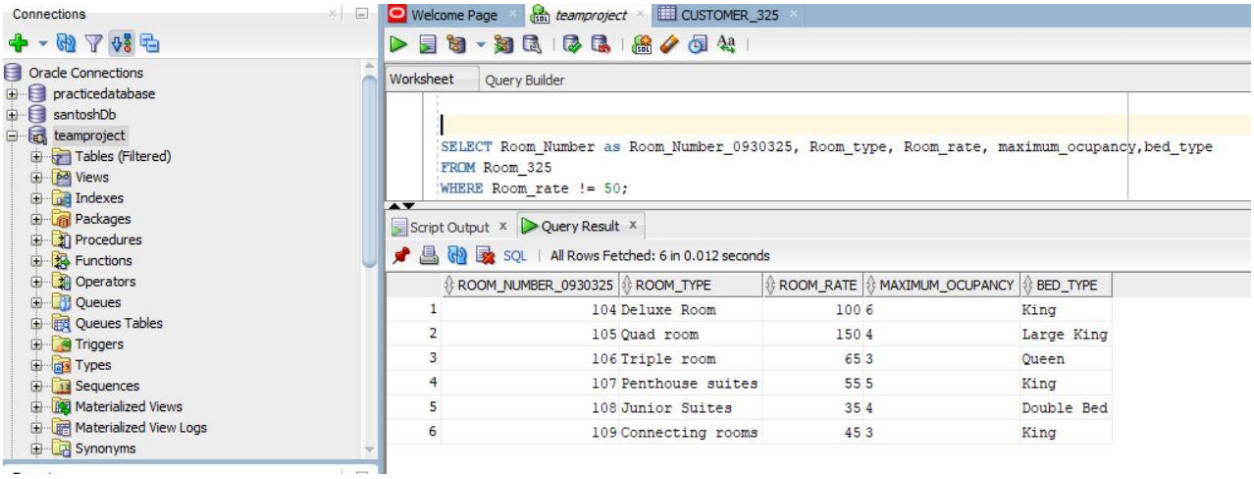


- != also Know as Not Equal to is used select the value that is not equal to the given value.
- It is the comparison operand used in the sql query.
- SQL Query

```
SELECT Room_Number as Room_Number_0930325, Room_type, Room_rate,
maximum_ocupancy,bed_type
FROM Room_325
WHERE Room_rate != 50;
```

- SQL Query Explanation:

IN the query the given value is used to select the room number, room type, room rate, maximum occupancy and bed type from room_325 table that is not equal to 50.



Category C

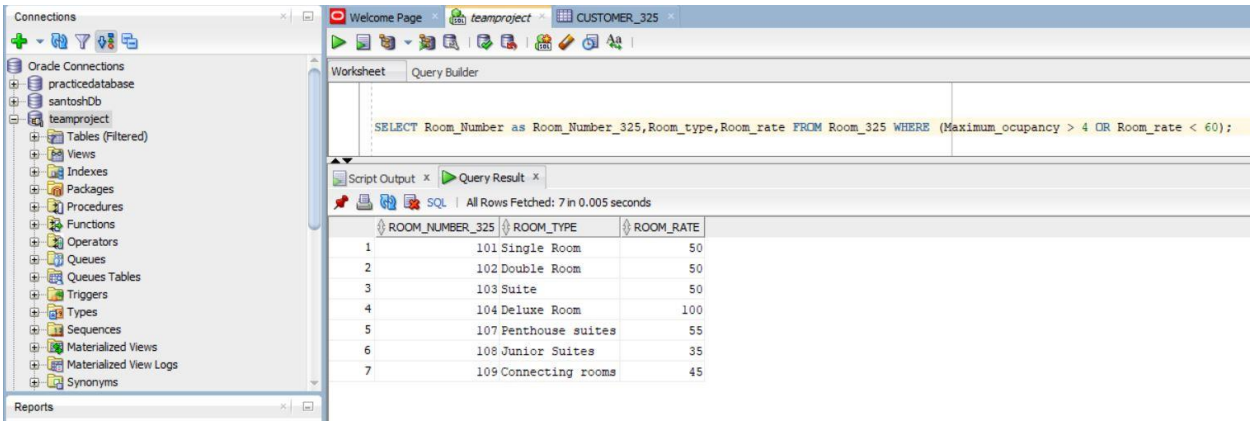
1

- Logical Operator are the operator test for the truth of some condition.
- SQL Query

```
SELECT Room_Number as Room_Number_325,Room_type,Room_rate FROM Room_325 WHERE (Maximum_ocupancy > 4 OR Room_rate < 60);
```

- SQL query Explanation:

In the query, the room number, room type, room rate from room_325 where maximum occupancy is greater than 4 or room rate is less than 60.



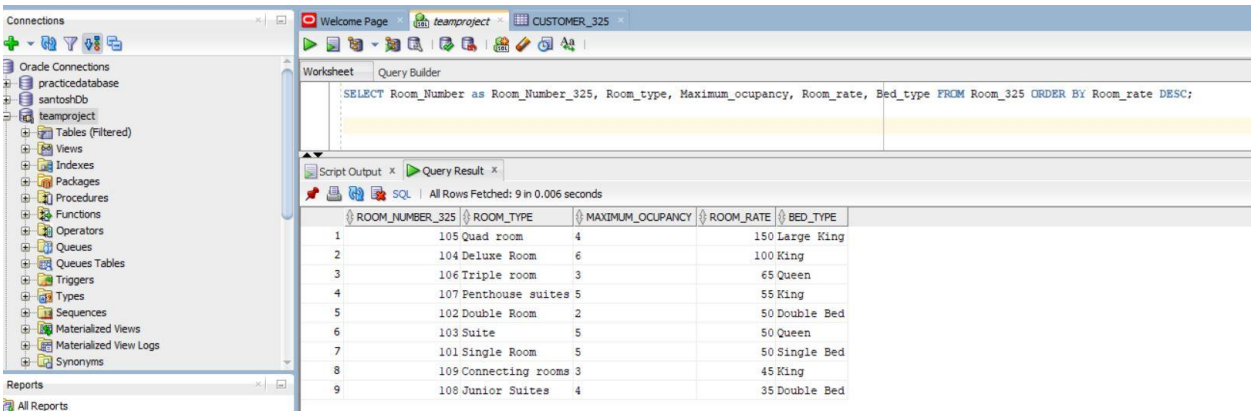
2

- Order By is use to sort the result in ascending or descending order.
- It will display the result in the either from low value to high or vice-versa.
- SQL Query

```
SELECT Room_Number as Room_Number_325, Room_type, Maximum_ocupancy, Room_rate, Bed_type FROM Room_325 ORDER BY Room_rate DESC;
```

- SQL Query Explanation:

In the query, the selected table will be display the value with the descending order by the room rate.



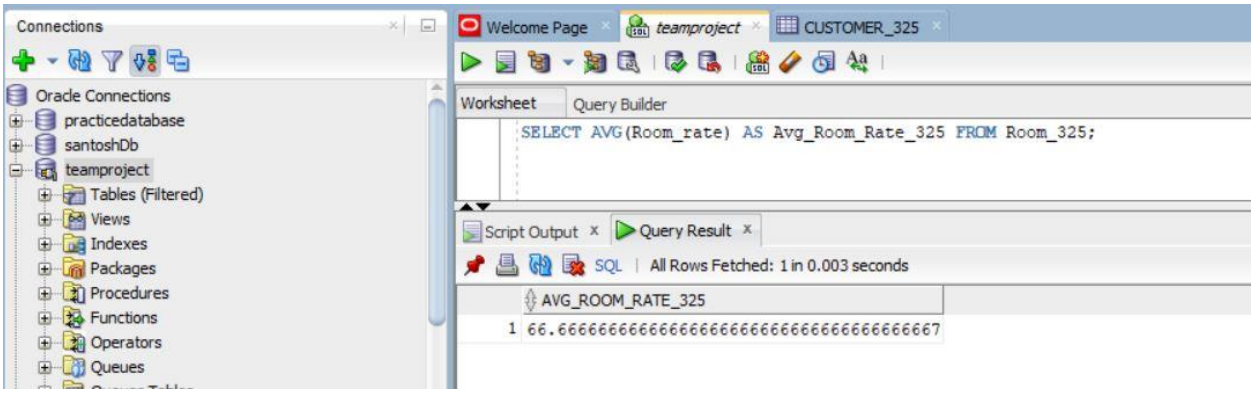
3

- Average also known as AVG is function that will return the average value of the column.
- SQL Query

```
SELECT AVG(Room_rate) AS Avg_Room_Rate_325 FROM Room_325;
```

- SQL Query Explanation:

In the query the average of the room rate is displayed from the room_325



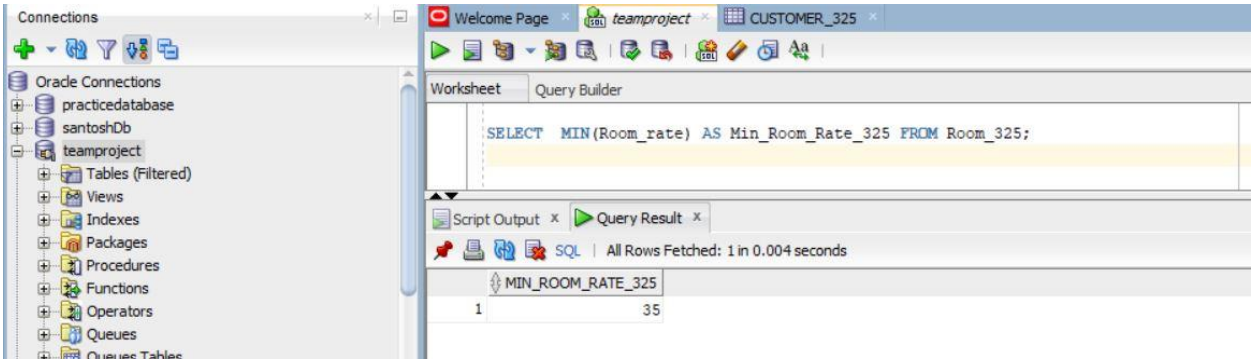
4

- Min is the function that will return the minimum value of the column.
- SQL Query

```
SELECT  MIN(Room_rate) AS Min_Room_Rate_325 FROM Room_325;
```

- SQL Query Explanation:

In the query the the room rate minimum value is return from the room_325



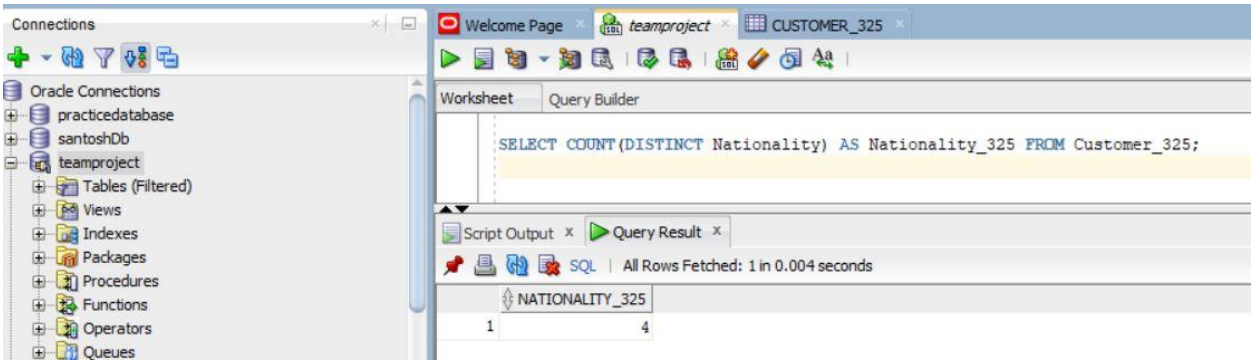
5

- Count is the function is the used to return the count of the value of the column.
- SQL Query

```
SELECT COUNT(DISTINCT Nationality) AS Nationality_325 FROM Customer_325;
```

- SQL Query Explanation:

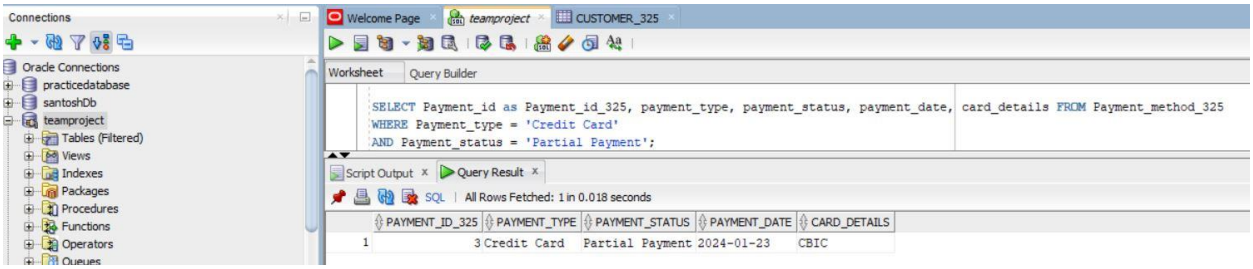
In the query the distinct nationality that count from the customer_325



6

- ❖ And operator is used to determine the value that will match all the case before returning the value.
- ❖ SQL Query

```
SELECT Payment_id as Payment_id_325, payment_type, payment_status, payment_date,
card_details FROM Payment_method_325 WHERE Payment_type = 'Credit Card'
AND Payment_status = 'Partial Payment';
```

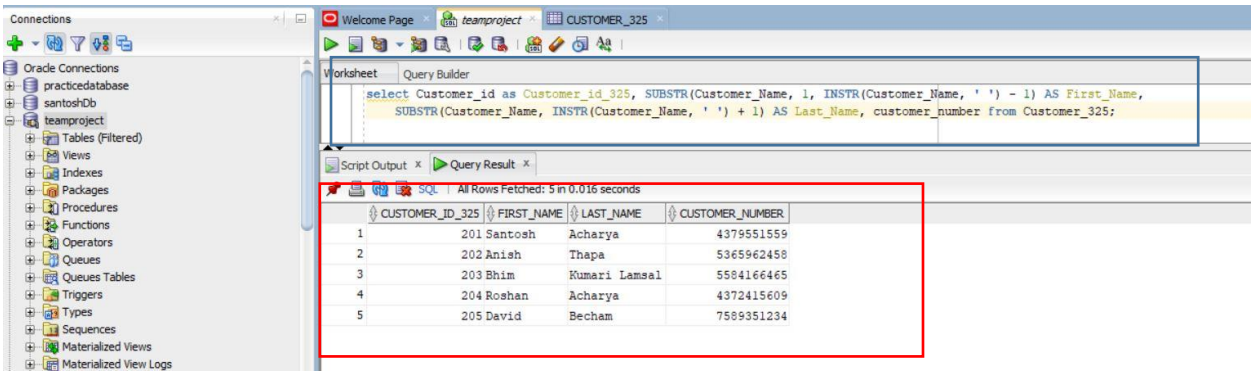


Category D

1

- INSTR is the function that return the first occurance position of a string in another string.
- SUBSTR is the function that extract substring from a string.
- SQL Query

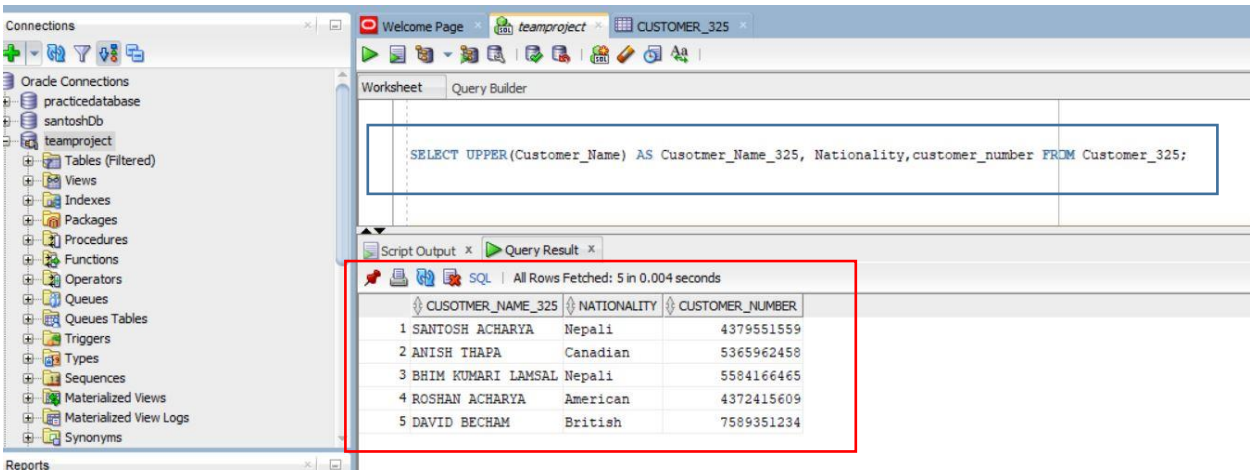
```
select Customer_id as Customer_id_325, SUBSTR(Customer_Name, 1, INSTR(Customer_Name, ' ') - 1) AS First_Name, SUBSTR(Customer_Name, INSTR(Customer_Name, ' ') + 1) AS Last_Name, customer_number from Customer_325;
```



2

- Upper is the function that will return the all the string in the column with the upper case.
- SQL Query

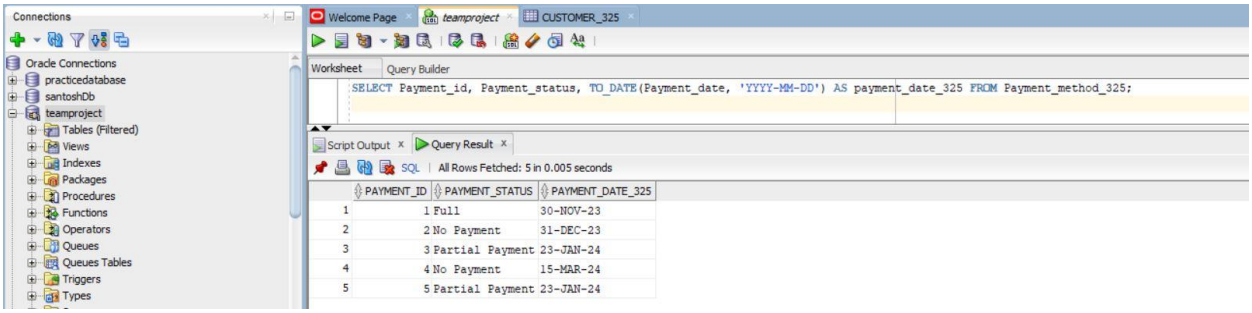
```
SELECT UPPER(Customer_Name) AS Cusotmer_Name_325, Nationality,customer_number FROM Customer_325;
```



3

- TO_DATE is the function that will convert the string to the date format in the SQL.
- SQL Query

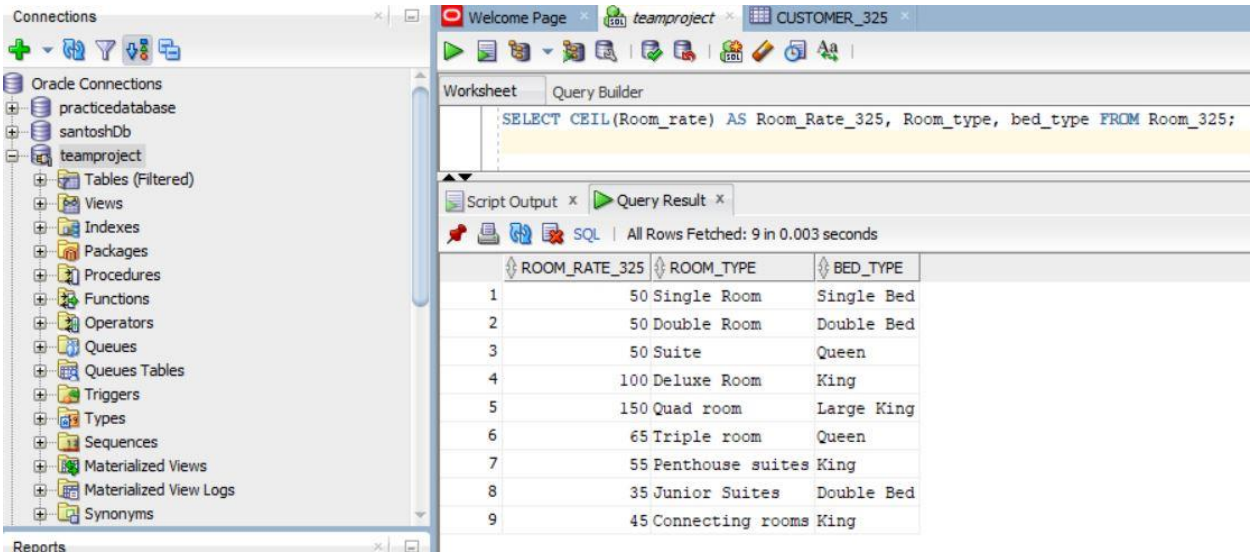
```
SELECT Payment_id, Payment_status, TO_DATE(Payment_date, 'YYYY-MM-DD') AS payment_date_325 FROM Payment_method_325;
```



4

- CEIL is the function that will return the nearest small value that is greater than or less than the number.
- SQL Query

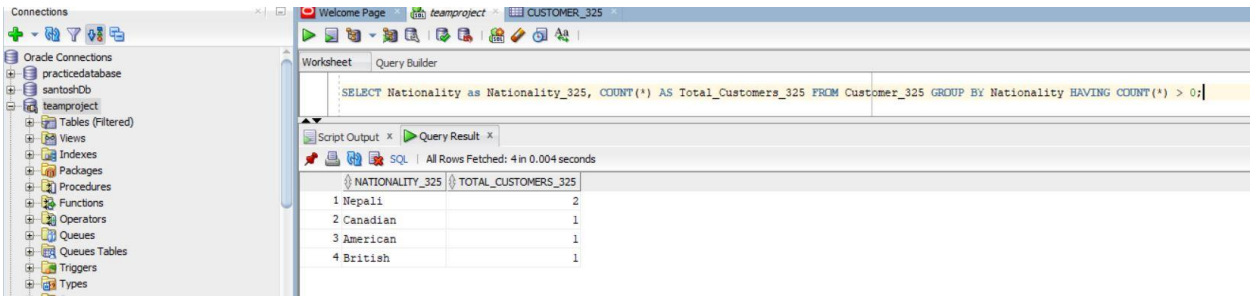
```
SELECT CEIL(Room_rate) AS Room_Rate_325, Room_type, bed_type FROM Room_325;
```



5

- ❖ COUNT is thr function that is used to count the value in the table.
- ❖ Having is used in the case when the when clause can't be use in the query.
- ❖ SQL Query

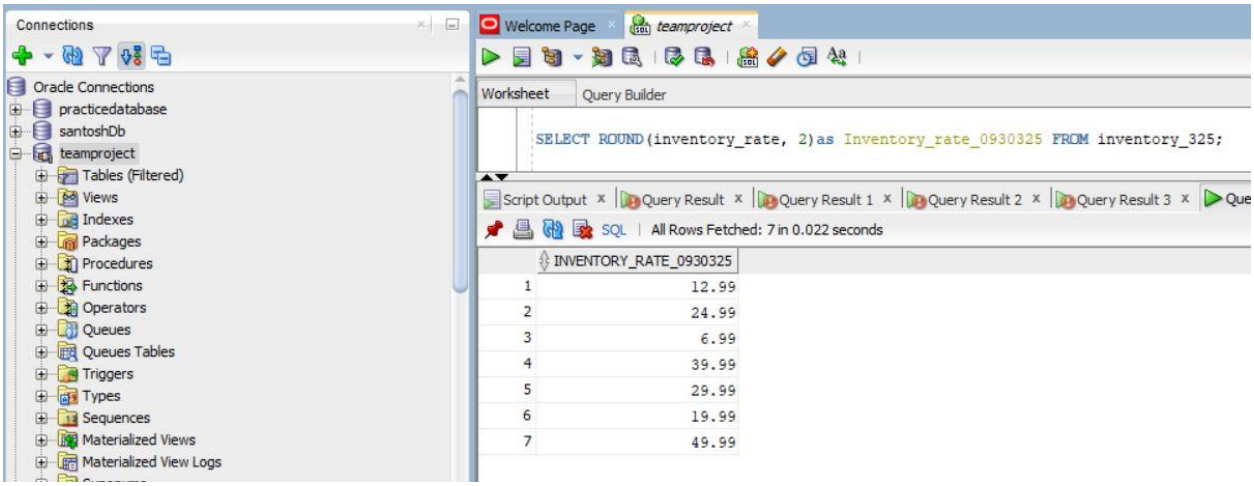
```
SELECT Nationality as Nationality_325, COUNT(*) AS Total_Customers_325 FROM Customer_325
GROUP BY Nationality HAVING COUNT(*) > 0;
```



6

- ❖ Round is the function that is used to round of the value to a specified number of decimal places.
- ❖ SQL Query

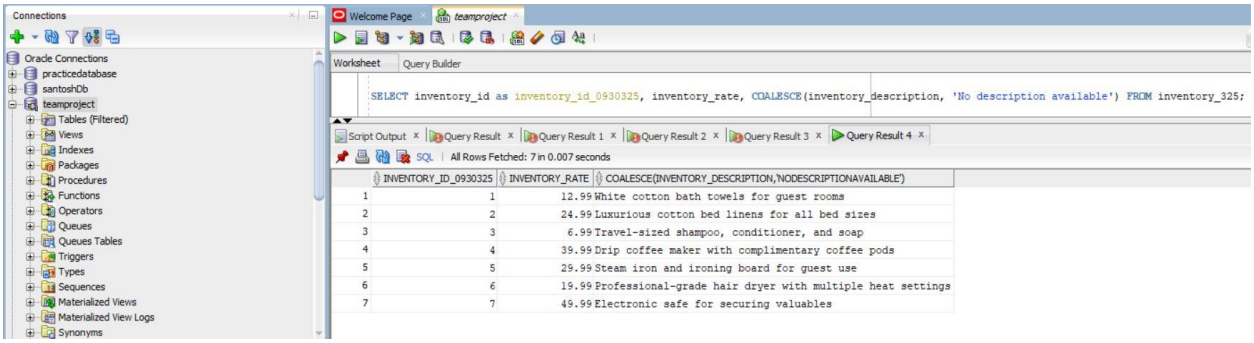
```
SELECT ROUND(inventory_rate, 2)as Inventory_rate_0930325 FROM inventory_325;
```



7

- ❖ COALESCE is used to return the first non-value in the list.
- ❖ It used two argument within it.
- ❖ SQL Query

```
SELECT inventory_id as inventory_id_0930325, inventory_rate, COALESCE(inventory_description,
'No description available') FROM inventory_325;
```



Category E

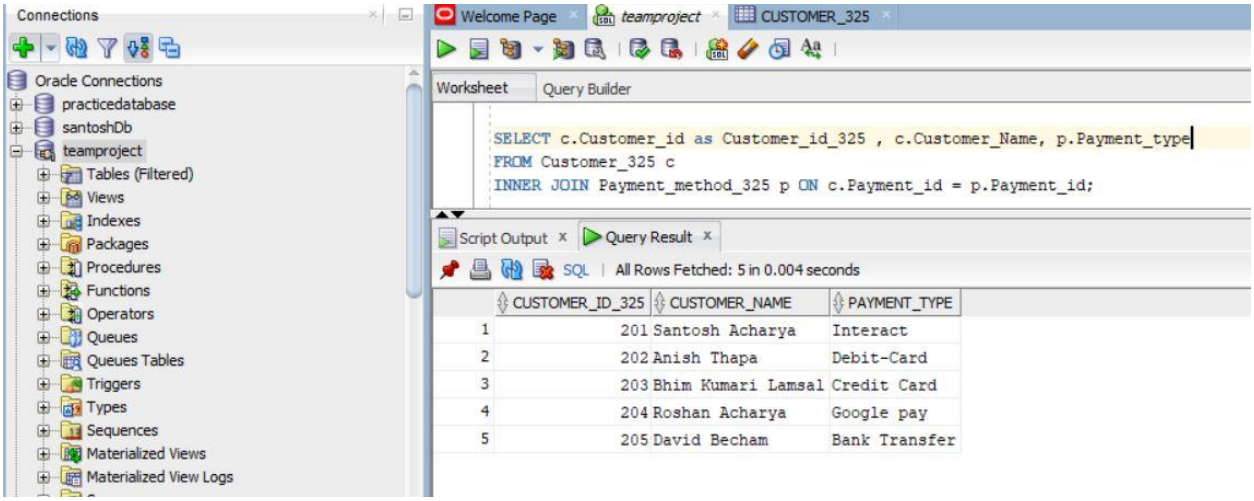
1

- Inner join is the function that is used to join the table when both the table has some matching values in a field that is common to both the table.
- SQL Query

```
SELECT c.Customer_id as Customer_id_325 , c.Customer_Name, p.Payment_type
FROM Customer_325 c
INNER JOIN Payment_method_325 p ON c.Payment_id = p.Payment_id;
```

- SQL Explanation:

In the query, the Customer_325 is join with thr Payment_method_325 where customer.payemntId is equal to payment.paymentid. c denote as the alias for the table customer and p for the payment.



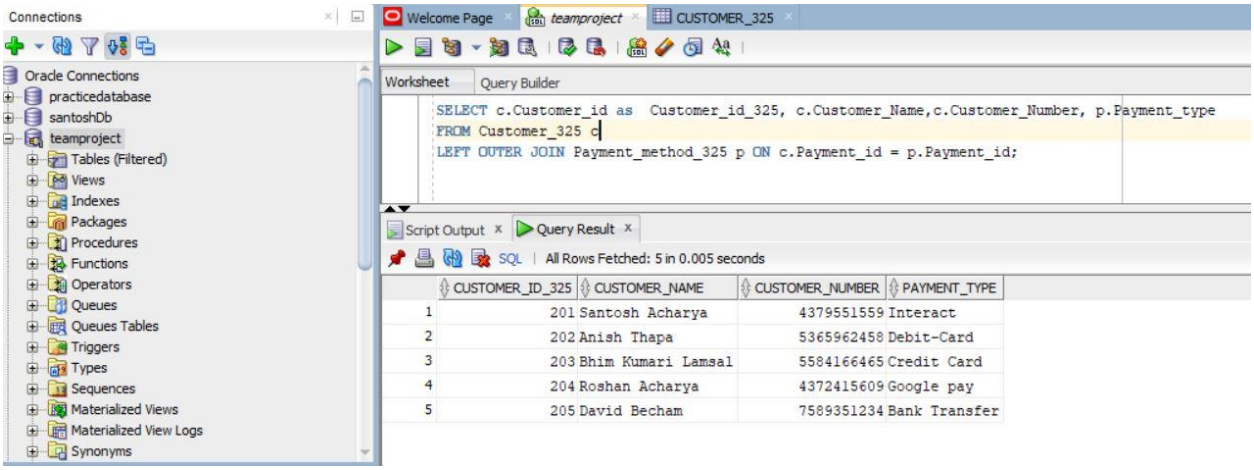
2

- Left outer join will return all the record from the left table and the matching record from the right table in the sql.
- SQL Query

```
SELECT c.Customer_id as  Customer_id_325, c.Customer_Name,c.Customer_Number,
p.Payment_type FROM Customer_325 c
LEFT OUTER JOIN Payment_method_325 p ON c.Payment_id = p.Payment_id;
```

- SQL Query Explanation:

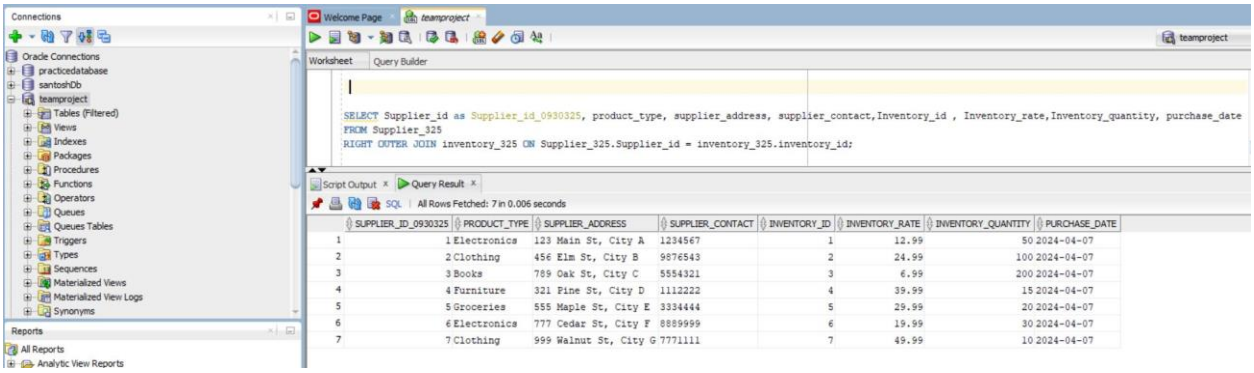
In the query , the table will return all the record from the customer and matching record from the payment method.



3

- Right outer join is similar to the left outer join but in Right outer join all the record from the right table is recorded and all the matching record from the left table is extracted.
- SQL Query

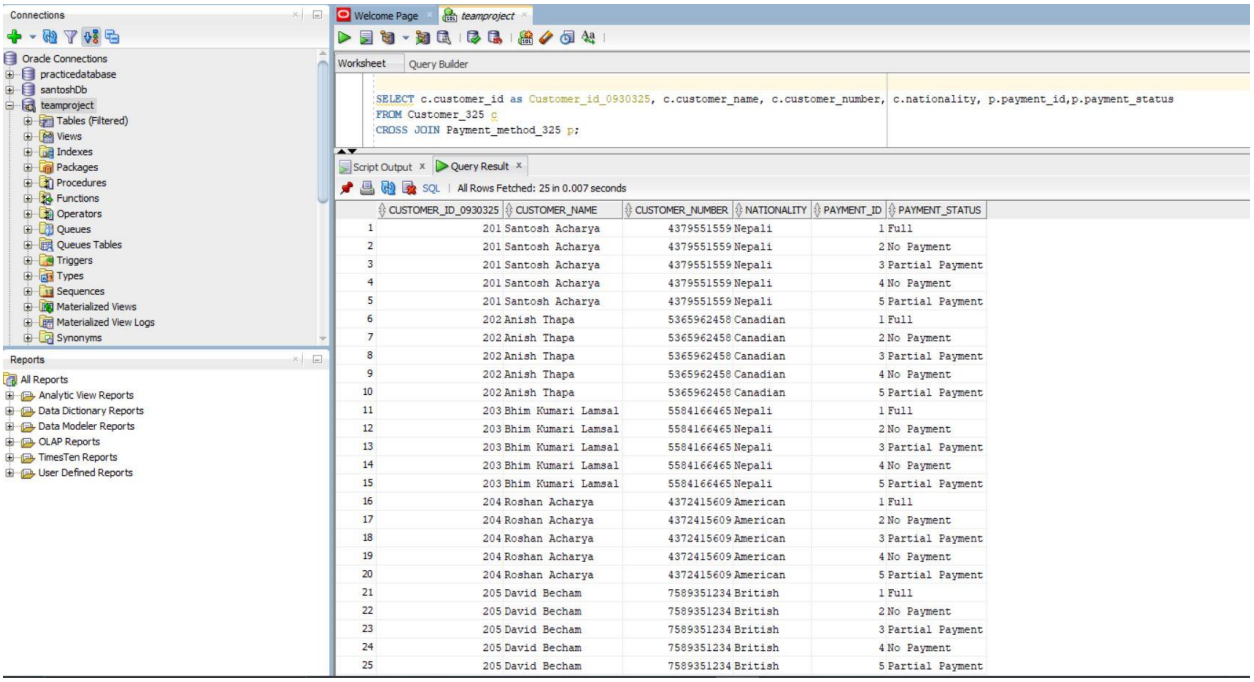
```
SELECT Supplier_id as Supplier_id_0930325, product_type, supplier_address,
supplier_contact,Inventory_id , Inventory_rate,Inventory_quantity, purchase_date
FROM Supplier_325
RIGHT OUTER JOIN inventory_325 ON Supplier_325.Supplier_id = inventory_325.inventory_id;
```



4

- Cross join is used to combine the each row of one table to each row of another table.
- It return the Cartesian product of the sets of rows.
- SQL Query

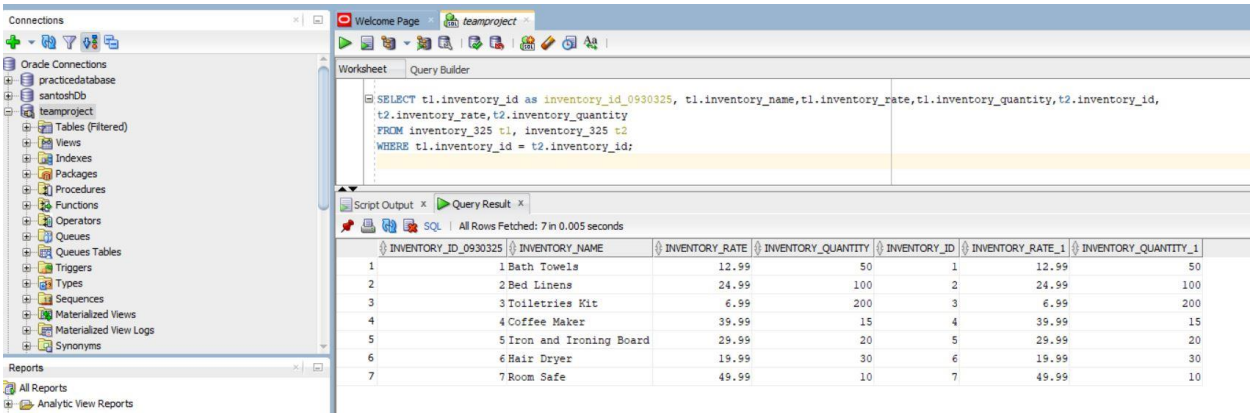
```
SELECT c.customer_id as Customer_id_0930325, c.customer_name, c.customer_number,
c.nationality, p.payment_id,p.payment_status FROM Customer_325 c
CROSS JOIN Payment_method_325 p;
```



5

- Self join is the query in which table is combined with itself in the sql.
- Here all the value is combined with itself in the query.
- SQL Query

```
SELECT t1.inventory_id as inventory_id_0930325,
t1.inventory_name,t1.inventory_rate,t1.inventory_quantity,t2.inventory_id,
t2.inventory_rate,t2.inventory_quantity FROM inventory_325 t1, inventory_325 t2
WHERE t1.inventory_id = t2.inventory_id;
```



Category F

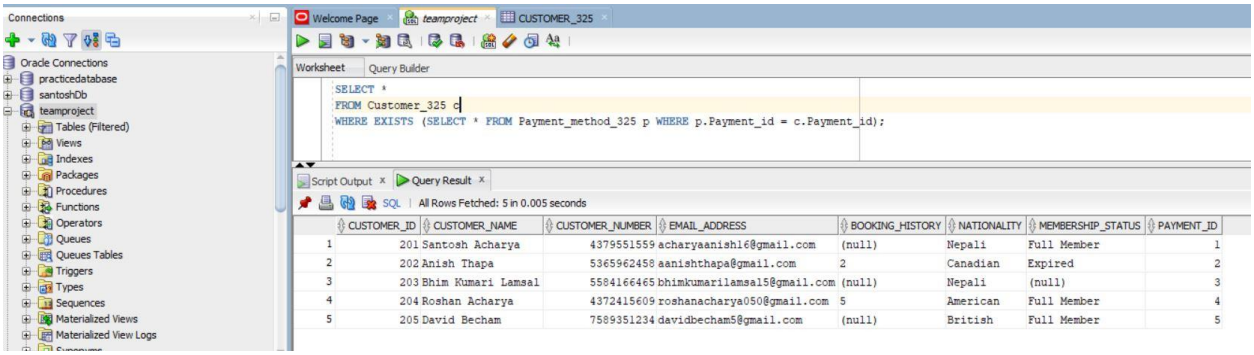
1

- Nested query is the query that has query inside the other query.
- Exists is the query in the sql which will return the value in the given condition exists in the query
- SQL Query

```
SELECT *
FROM Customer_325 c
WHERE EXISTS (SELECT * FROM Payment_method_325 p WHERE p.Payment_id = c.Payment_id);
```

- SQL Explanation:

In the query the customer table will return the value if the nested query full fill the requirement where paymentid of payment method table is equal to the paymentid of customer table.



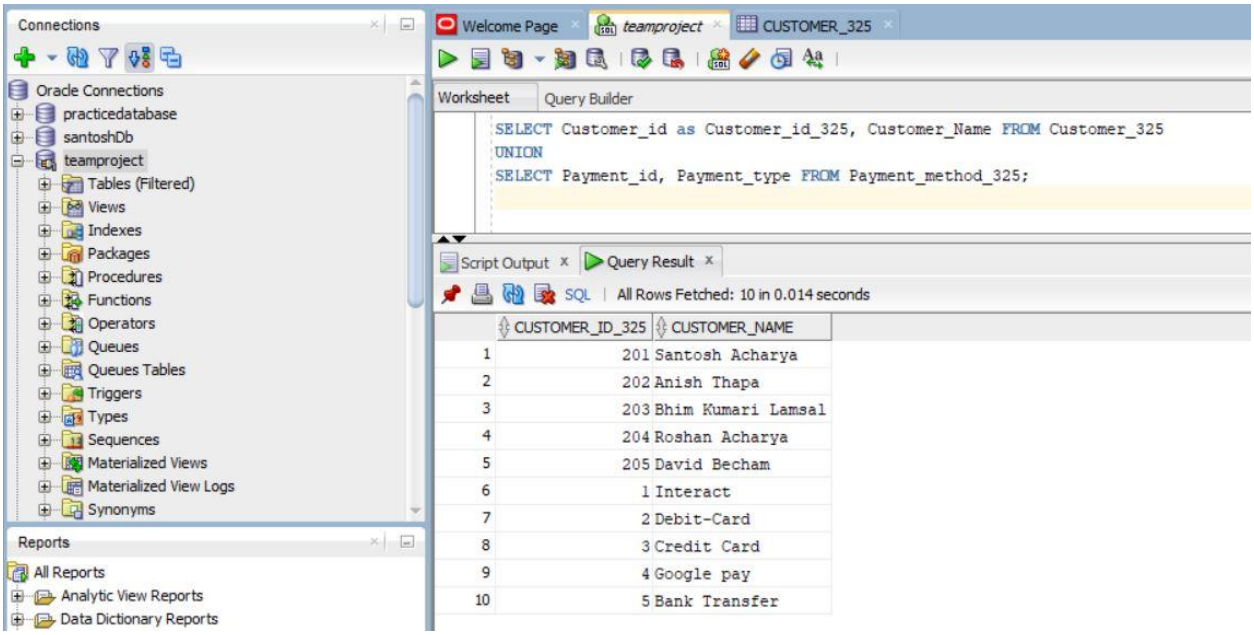
2

- UNION is use to combine the result set of two or more select statement.
- SQL Query

```
SELECT Customer_id as Customer_id_325, Customer_Name FROM Customer_325
UNION
SELECT Payment_id, Payment_type FROM Payment_method_325;
```

- SQL Explanation:

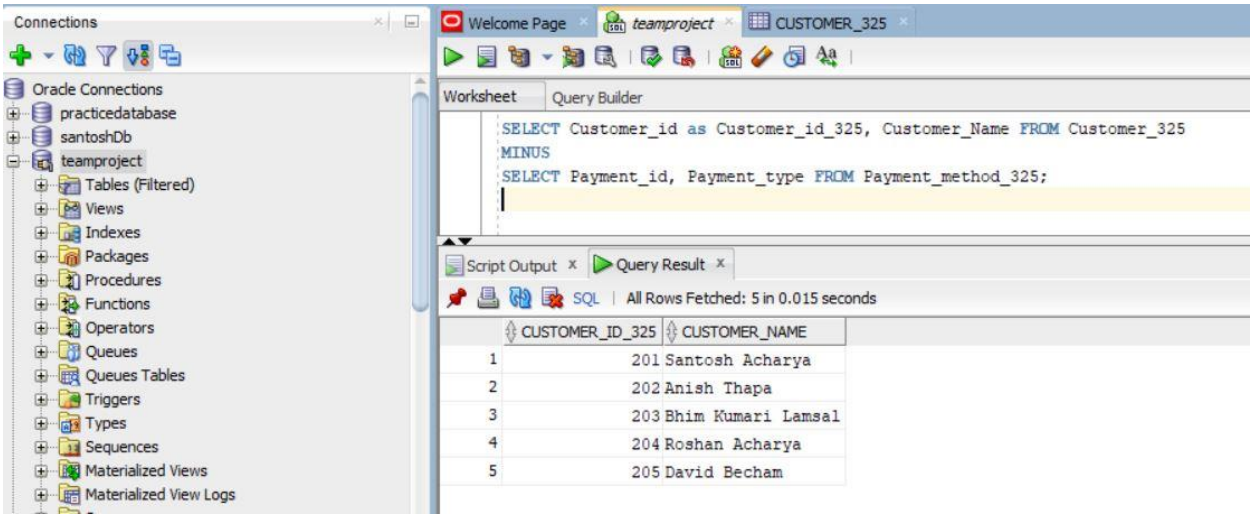
In the query the customer_325 table is combine with the payement_method_325 table.



3

- MINUS is the function used to return all the rows in the first select statement that are not returned by the second.
- It will return the unique rows.
- SQL Query

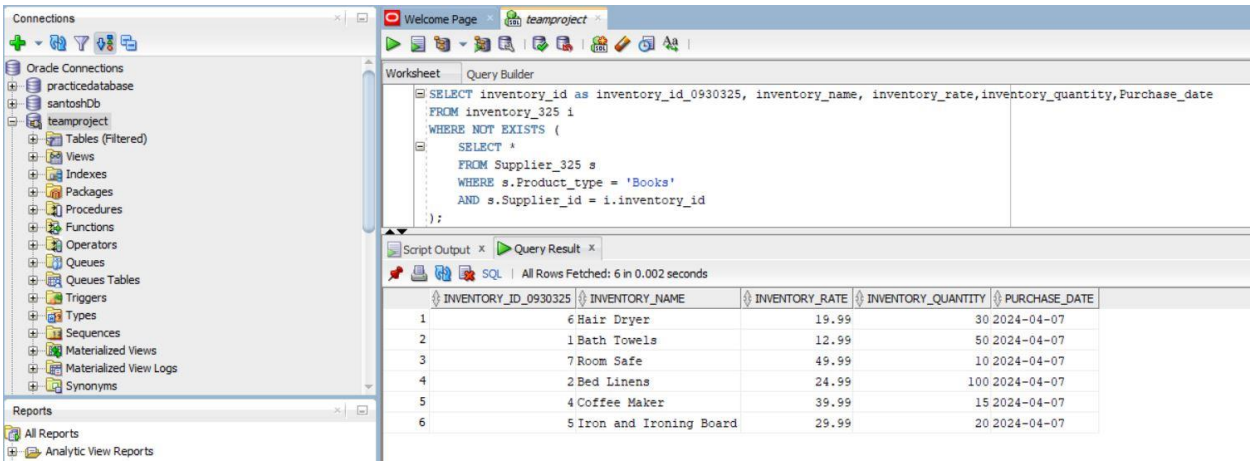
```
SELECT Customer_id as Customer_id_325, Customer_Name FROM Customer_325
MINUS
SELECT Payment_id, Payment_type FROM Payment_method_325;
```



4

- ❖ NOT EXISTS is the query that test the non existing value in the table.
- ❖ It is used when to ignore the logic of exists.
- ❖ SQL Query

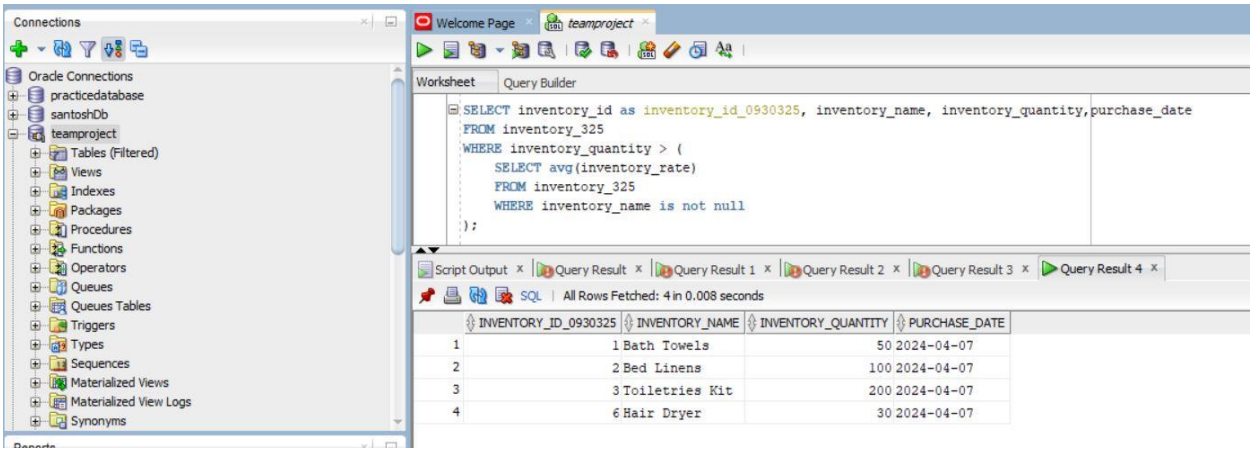
```
SELECT inventory_id as inventory_id_0930325, inventory_name,
inventory_rate,inventory_quantity,Purchase_date FROM inventory_325 i
WHERE NOT EXISTS (
SELECT * FROM Supplier_325 s
        WHERE s.Product_type = 'Bath Towels'
        AND s.Supplier_id = i.inventory_id
);
```



5

- ❖ Not null is the function that will return the value that aren't null or have some value in the column or table.
- ❖ SQL Query

```
SELECT inventory_id as inventory_id_0930325, inventory_name, inventory_quantity,purchase_date
FROM inventory_325
WHERE inventory_quantity > (
SELECT avg(inventory_rate) FROM inventory_325  WHERE inventory_name is not null
);
```



Advance SQL

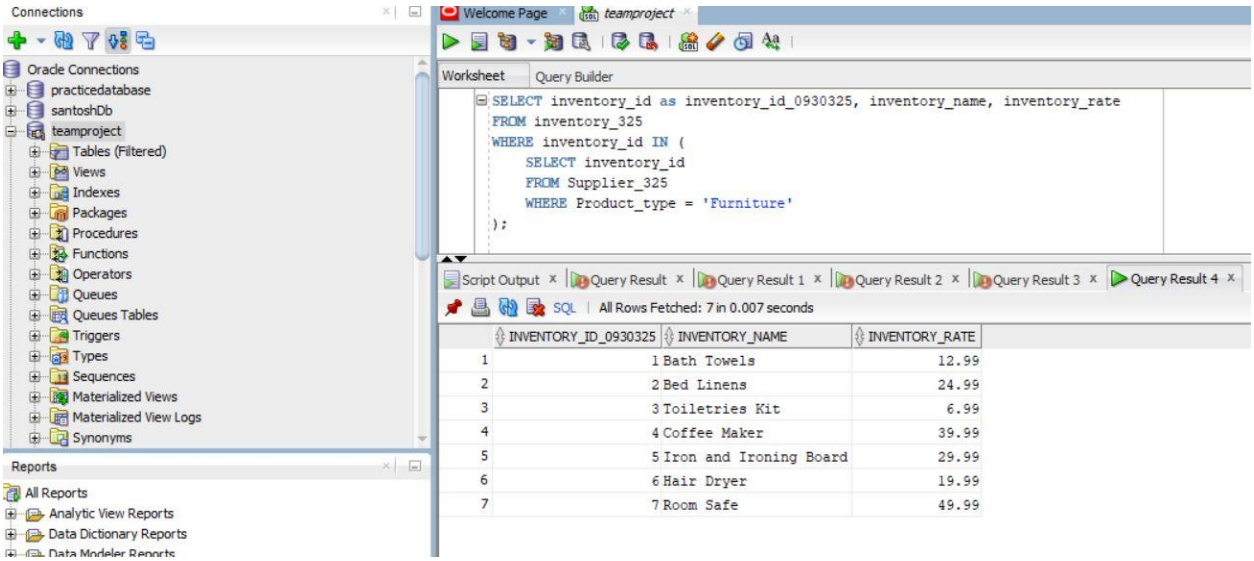
1

- ❖ In will allow to accept multiple value . It operate as the OR operator in the sql
- ❖ SQL Query

```
SELECT inventory_id as inventory_id_0930325, inventory_name, inventory_rate
FROM inventory_325 WHERE inventory_id IN (
SELECT inventory_id FROM Supplier_325 WHERE Product_type = 'Furniture'
);
```

- ❖ SQL Query Explanation:

In the query first the inventory id is selected from the suppliers table which match the furniture and then all the number that match in the inventory table is displayed.



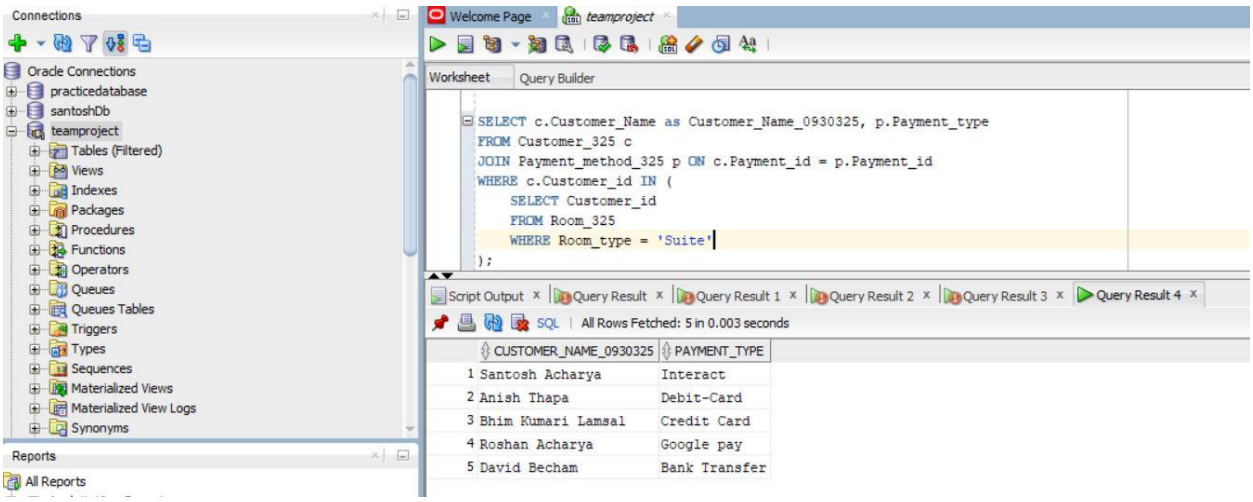
2

- JOIN is used to combine the two or more table related to there column.
- SQL Query

```
SELECT c.Customer_Name as Customer_Name_0930325, p.Payment_type
FROM Customer_325 c
JOIN Payment_method_325 p ON c.Payment_id = p.Payment_id
WHERE c.Customer_id IN ( SELECT Customer_id FROM Room_325 WHERE Room_type = 'Suite'
);
```

- SQL Explanation:

In the Query the customerid is selected from the room table and according to the value the customer table and payment method table is combined.



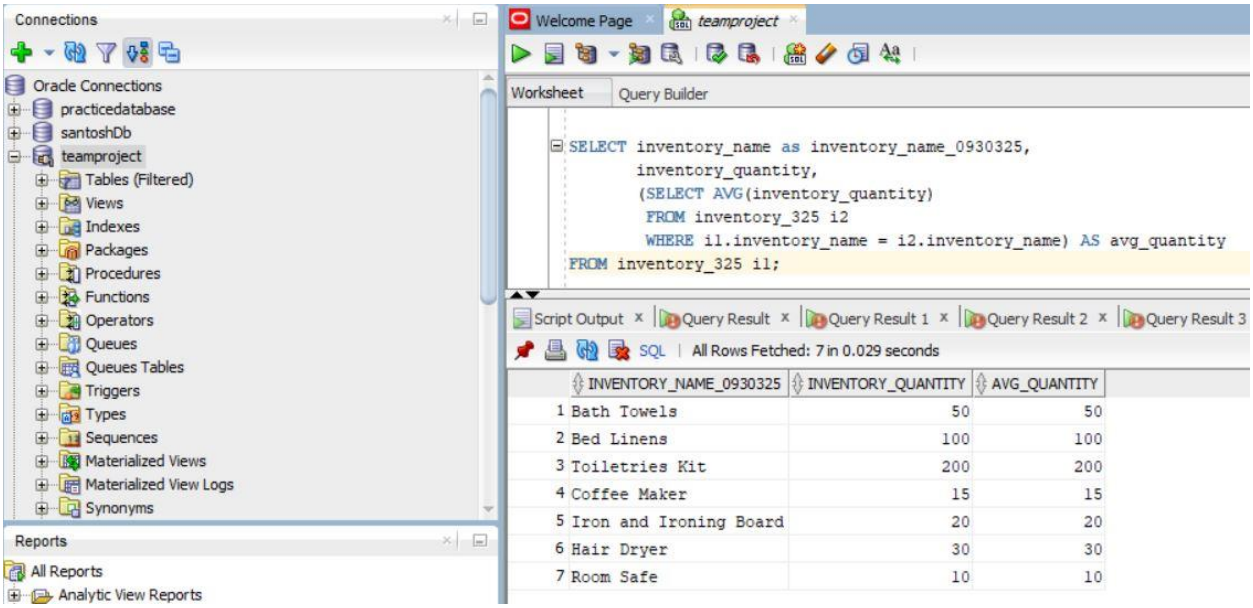
3

- Average function will return the average value of the column.
- SQL Query

```
SELECT inventory_name as inventory_name_0930325, inventory_quantity,
(SELECT AVG(inventory_quantity) FROM inventory_325 i2
WHERE i1.inventory_name = i2.inventory_name) AS avg_quantity
FROM inventory_325 i1;
```

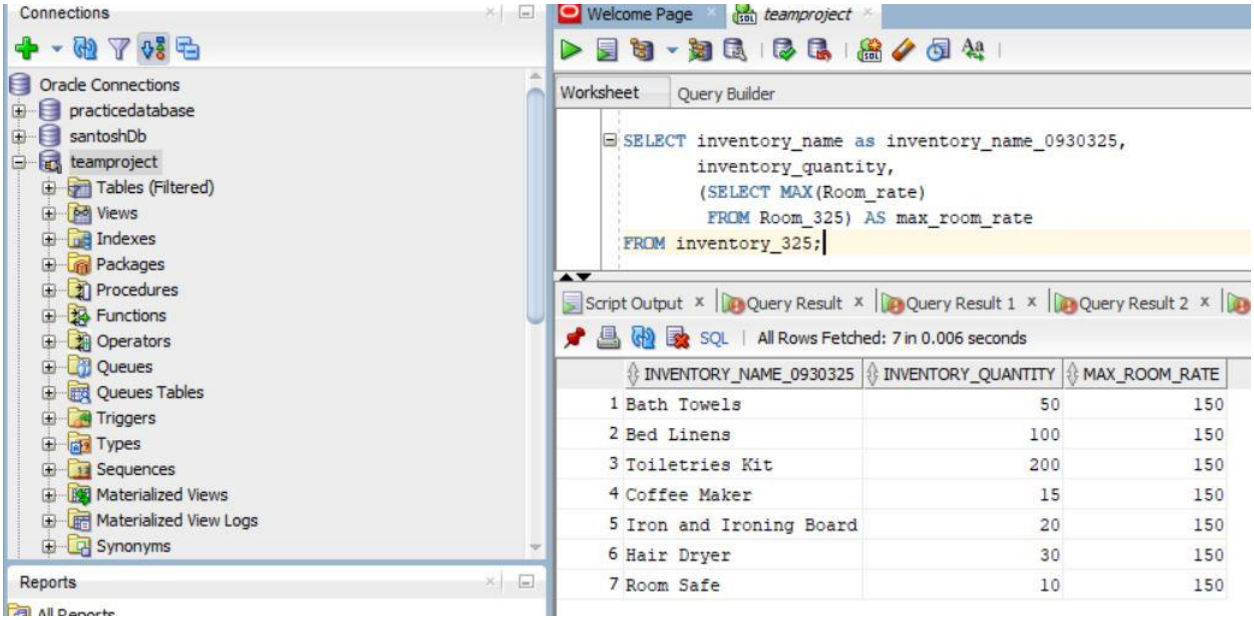
- SQL Query Explanation:

In the query the average of inventory quantity is taken when the inventory_name of the table is equal to the inventory name of the other table and the output is displayed.



- ❖ MAX function will return the maximum value of the column.
- ❖ It is used to get the highest value with in the column.
- ❖ SQL Query

```
SELECT inventory_name as inventory_name_0930325, inventory_quantity,
(SELECT MAX(Room_rate) FROM Room_325) AS max_room_rate
FROM inventory_325;
```



6

- ❖ Exists function will return the value when the condition is meet.
- ❖ It works only if there is one value that will be return.
- ❖ SQL Query

```
SELECT Customer_id as Customer_id_0930325, customer_name,
customer_number,membership_status,payment_id
FROM Customer_325 c WHERE EXISTS (
SELECT * FROM Payment_method_325 p WHERE c.Payment_id = p.Payment_id
AND p.Payment_status = 'Full'
);
```