**B\_18 DAYAL ADITYA KISHOR**

**Assignment No. 5**

**Title:**

Implementation of different types of Joins

* Inner Join
* Outer Join
* Natural Join etc..

**Problem Statement:**

Create tables with primary key, foreign key constraint as shown in above schema, salesman\_id is foreign in customer table

Salesman

|  |  |  |  |
| --- | --- | --- | --- |
| salesman\_id | name | city | commission |
| 5001 | James Hoog | New York | 0.15 |
| 5002 | Nail Knite | Paris | 0.13 |
| 5005 | Pit Alex | London | 0.11 |
| 5006 | Mc Lyon | Paris | 0.14 |
| 5007 | Paul Adam | Rome | 0.13 |
| 5003 | Lauson Hen | San Jose | 0.12 |

Customer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| customer\_id | cust\_name | city | grade | salesman\_id |
| 3002 | Nick Rimando | New York | 100 | 5001 |
| 3007 | Brad Davis | New York | 200 | 5001 |
| 3005 | Graham Zusi | California | 200 | 5002 |
| 3008 | Julian Green | London | 300 | 5002 |
| 3004 | Fabian Johnson | Paris | 300 | 5006 |
| 3009 | Geoff Cameron | Berlin | 100 | 5003 |
| 3003 | Jozy Altidor | Moscow | 200 | 5007 |
| 3001 | Brad Guzan | London |  | 5003 |

Orders

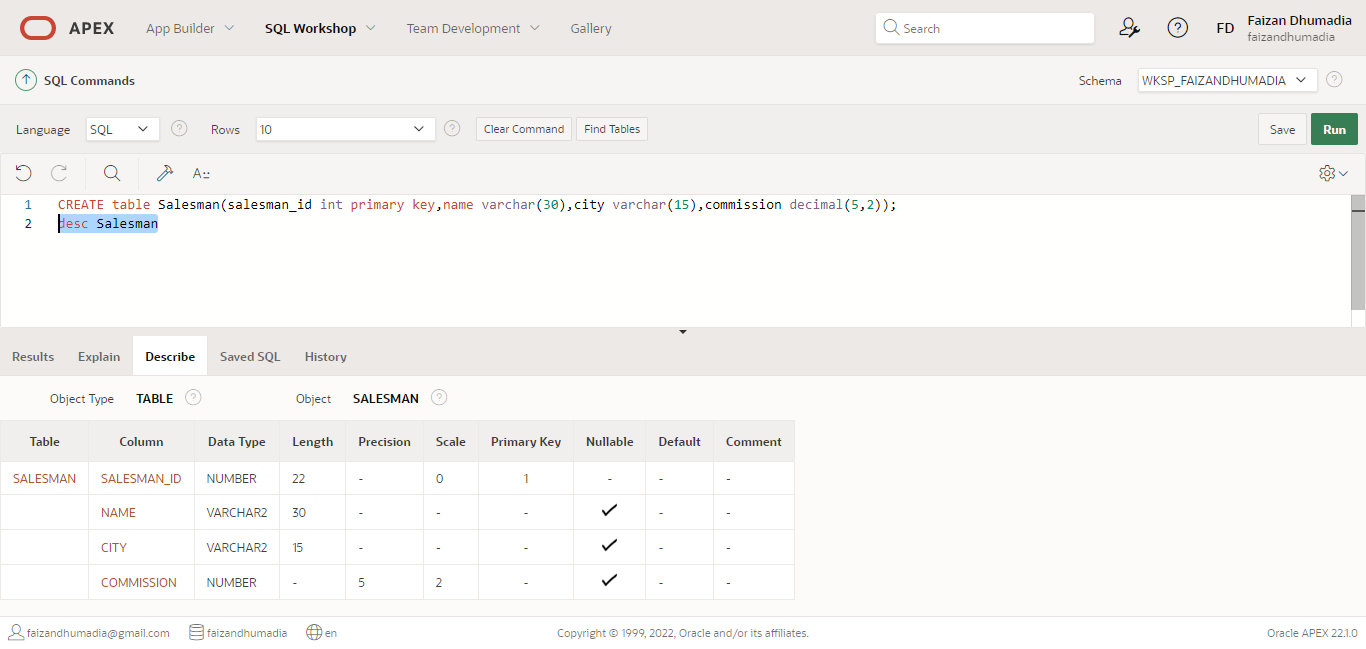
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ord\_no | purch\_amt | ord\_date | customer\_id | salesman\_id |
| 70001 | 150.5 | 2012-10-05 | 3005 | 5002 |
| 70009 | 270.65 | 2012-09-10 | 3001 | 5003 |
| 70002 | 65.26 | 2012-10-05 | 3002 | 5001 |
| 70004 | 110.5 | 2012-08-17 | 3009 | 5003 |
| 70007 | 948.5 | 2012-09-10 | 3005 | 5002 |
| 70005 | 2400.6 | 2012-07-27 | 3007 | 5001 |
| 70008 | 5760 | 2012-09-10 | 3002 | 5001 |
| 70010 | 1983.43 | 2012-10-10 | 3004 | 5006 |
| 70003 | 2480.4 | 2012-10-10 | 3009 | 5003 |
| 70012 | 250.45 | 2012-06-27 | 3008 | 5002 |
| 70011 | 75.29 | 2012-08-17 | 3003 | 5007 |
| 70013 | 3045.6 | 2012-04-25 | 3002 | 5001 |

Queries:

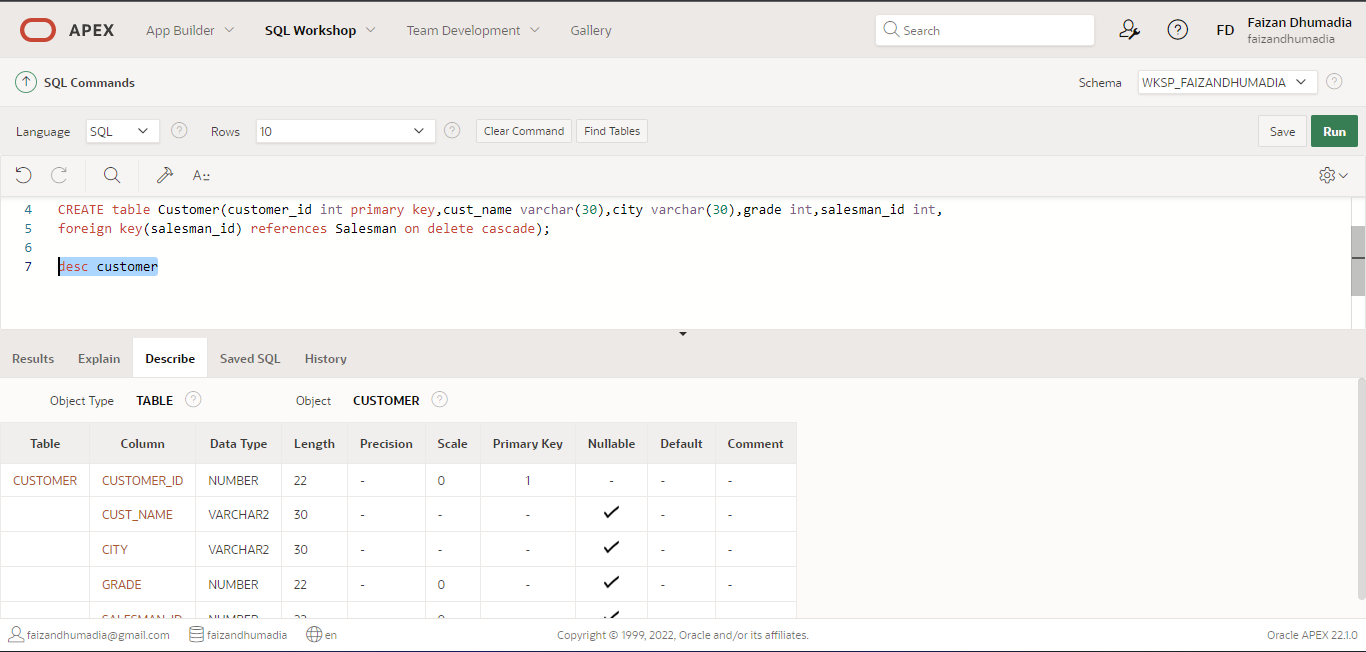
1. Find the salesman and customers with their name and cities, who belongs to the same city. (Use Cross join/ cartesian product)
2. Prepare a list for which salesman are working for which customer along with city and commissions earned by the salesman. (use inner join/join … on condition)
3. Write a query to make a join on the tables salesman and orders in such a form that the same column of each table will appear once and only the relational rows will come.
4. Prepare a list for salesman who works either for one or more customer or not yet join under any of the customer (use right outer join)
5. Prepare a list in ascending order for the salesman who works either for one or more customer or not yet join under any of the customer ( Use left outer join)

**Answer:**

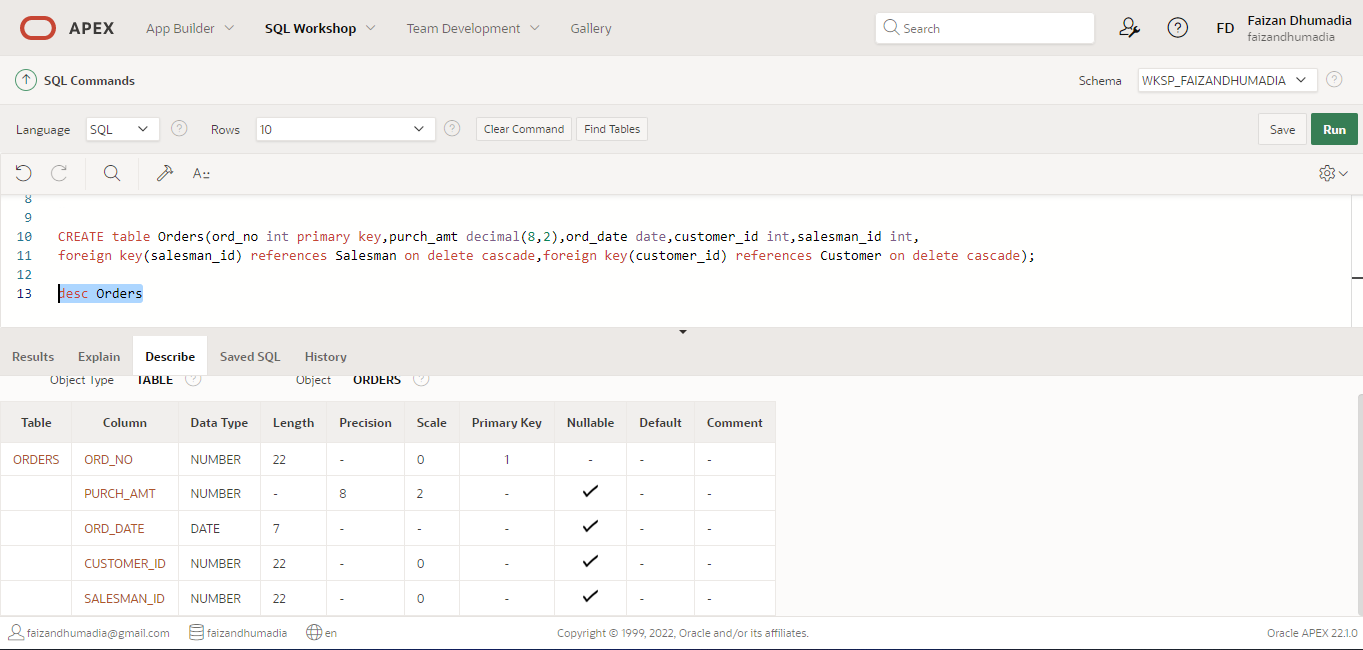
Create Tables Salesman, Customer, Orders  
CREATE table Salesman(salesman\_id int primary key,name varchar(30),city varchar(15),commission decimal(5,2));



CREATE table Customer(customer\_id int primary key,cust\_name varchar(30),city varchar(30),grade int,salesman\_id int, foreign key(salesman\_id) references Salesman on delete cascade);



CREATE table Orders(ord\_no int primary key,purch\_amt decimal(8,2),ord\_date date,customer\_id int,salesman\_id int,foreign key(salesman\_id) references Salesman on delete cascade,foreign key(customer\_id) references Customer on delete cascade);



Insert Records into Salesman Tables

Insert into Salesman values(5001,'James Hoog', 'New York',0.15);

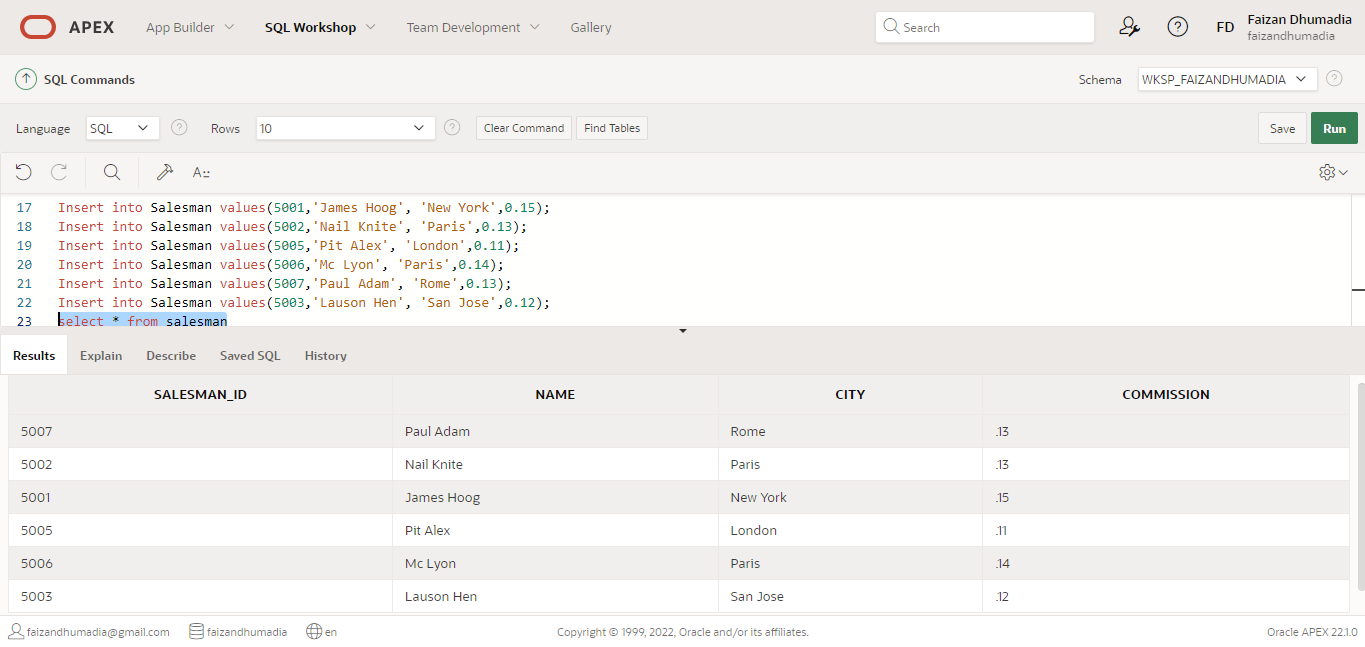
Insert into Salesman values(5002,'Nail Knite', 'Paris',0.13);

Insert into Salesman values(5005,'Pit Alex', 'London',0.11);

Insert into Salesman values(5006,'Mc Lyon', 'Paris',0.14);

Insert into Salesman values(5007,'Paul Adam', 'Rome',0.13);

Insert into Salesman values(5003,'Lauson Hen', 'San Jose',0.12);



Insert Records into Customer Tables

Insert into Customer values(3002, 'Nick Rimando','New York',100,5001);

Insert into Customer values(3007, 'Brad Davis','New York',200,5001);

Insert into Customer values(3005, 'Graham Zusi','California',200,5002);

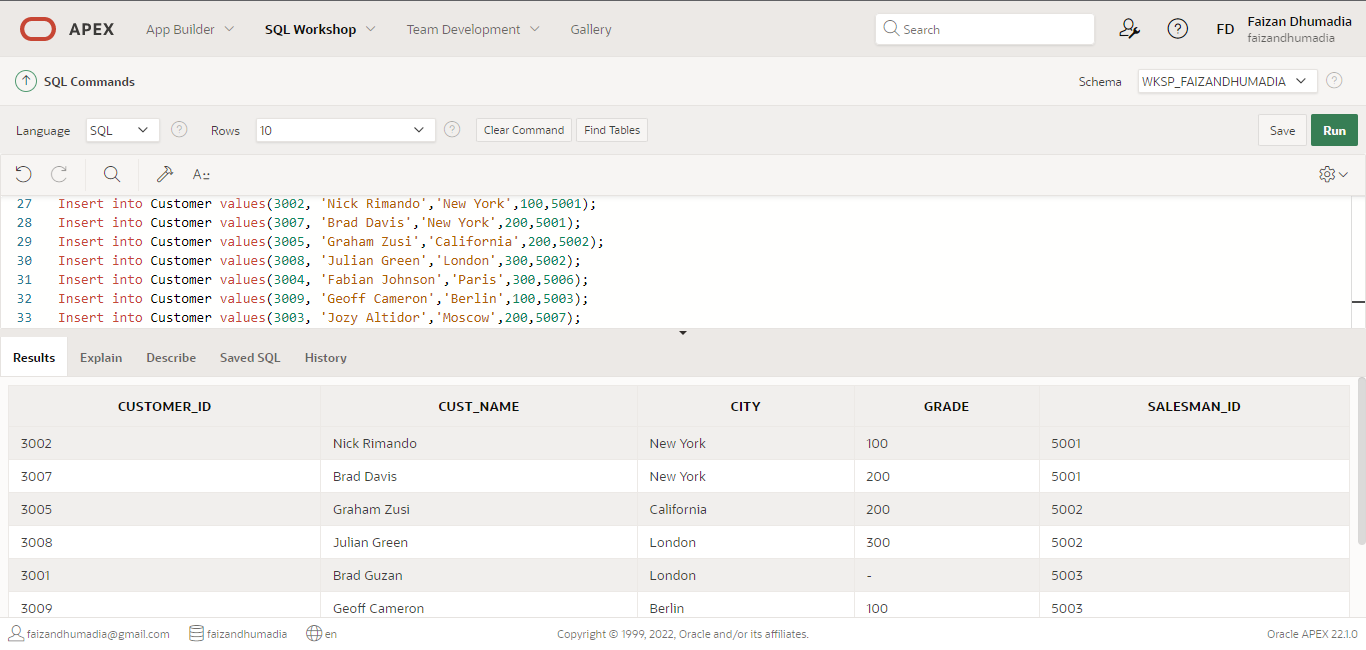
Insert into Customer values(3008, 'Julian Green','London',300,5002);

Insert into Customer values(3004, 'Fabian Johnson','Paris',300,5006);

Insert into Customer values(3009, 'Geoff Cameron','Berlin',100,5003);

Insert into Customer values(3003, 'Jozy Altidor','Moscow',200,5007);

Insert into Customer values(3001, 'Brad Guzan','London','',5003);



Insert Records into Orders Tables

insert into Orders values(70001,150.5 ,'10-05-2012',3005,5002);

insert into Orders values(70009,270.65 ,'09-10-2012',3001,5003);

insert into Orders values(70002,65.26 ,'10-05-2012',3002,5001);

insert into Orders values(70004,110.5 ,'08-17-2012',3009,5003);

insert into Orders values(70007,948.5 ,'09-10-2012',3005,5002);

insert into Orders values(70005,2400.6 ,'07-27-2012',3007,5001);

insert into Orders values(70008,5760 ,'09-10-2012',3002,5001);

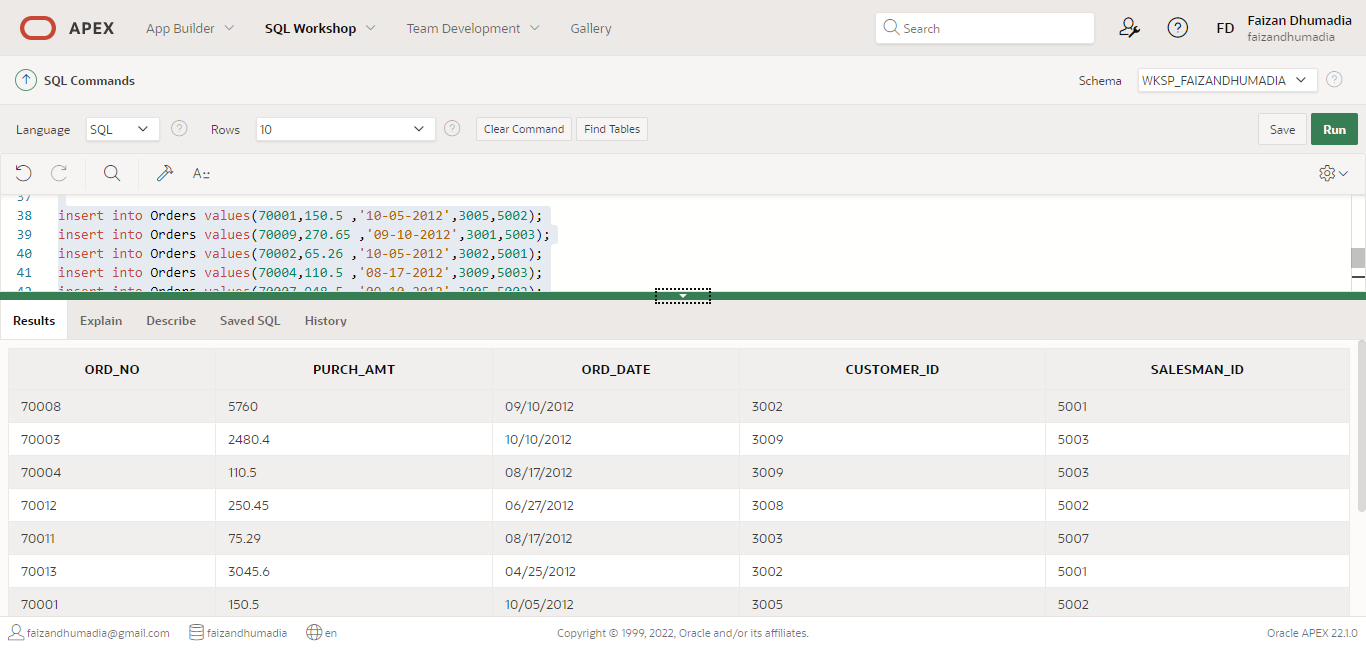
insert into Orders values(70010,1983.43 ,'10-10-2012',3004,5006);

insert into Orders values(70003,2480.4,'10-10-2012',3009,5003);

insert into Orders values(70012,250.45,'06-27-2012',3008,5002);

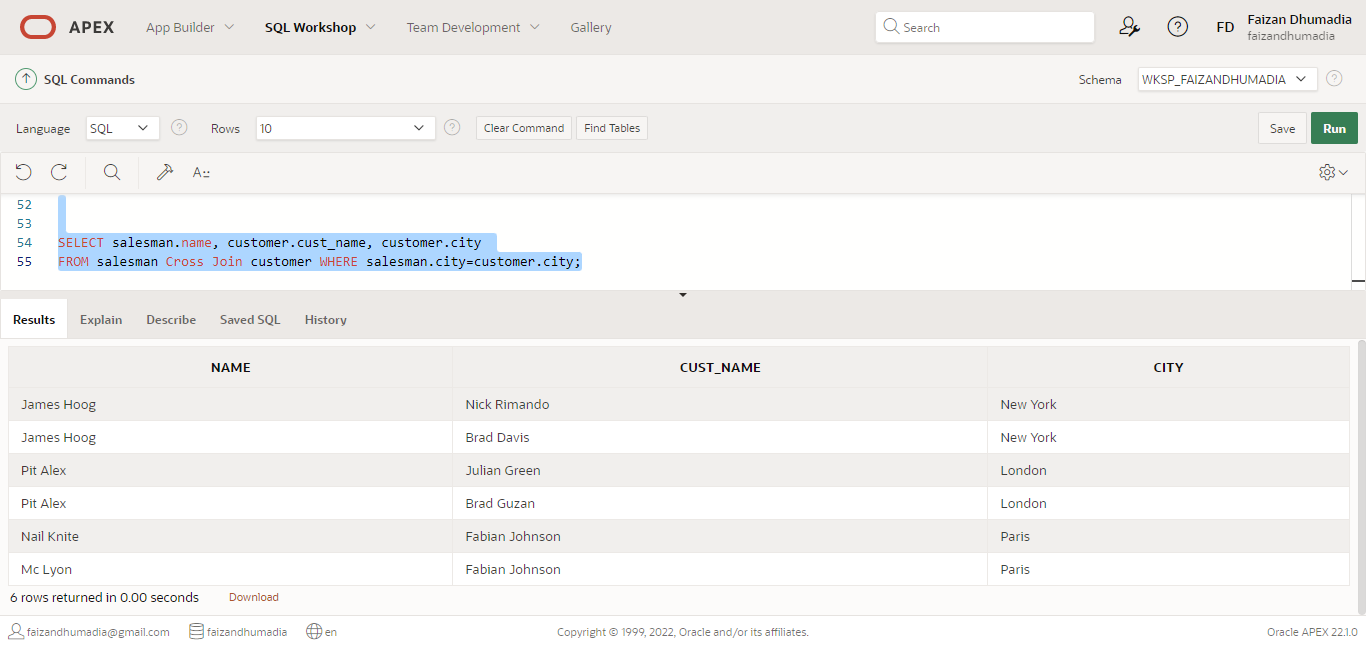
insert into Orders values(70011,75.29, '08-17-2012',3003,5007);

insert into Orders values(70013,3045.6, '04-25-2012',3002,5001);



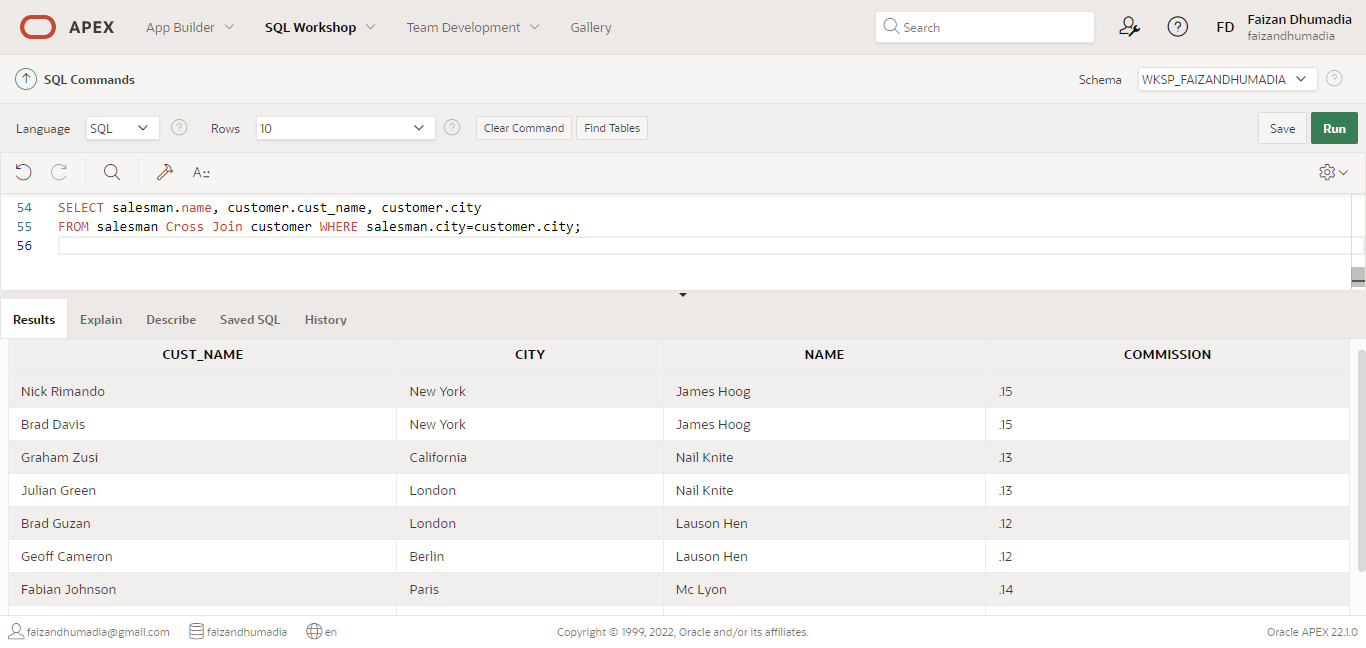
1. Find the salesman and customers with their name and cities, who belongs to the same city. (Use Cross join/ cartesian product)

SELECT salesman.name, customer.cust\_name, customer.city   
FROM salesman Cross Join customer WHERE salesman.city=customer.city;



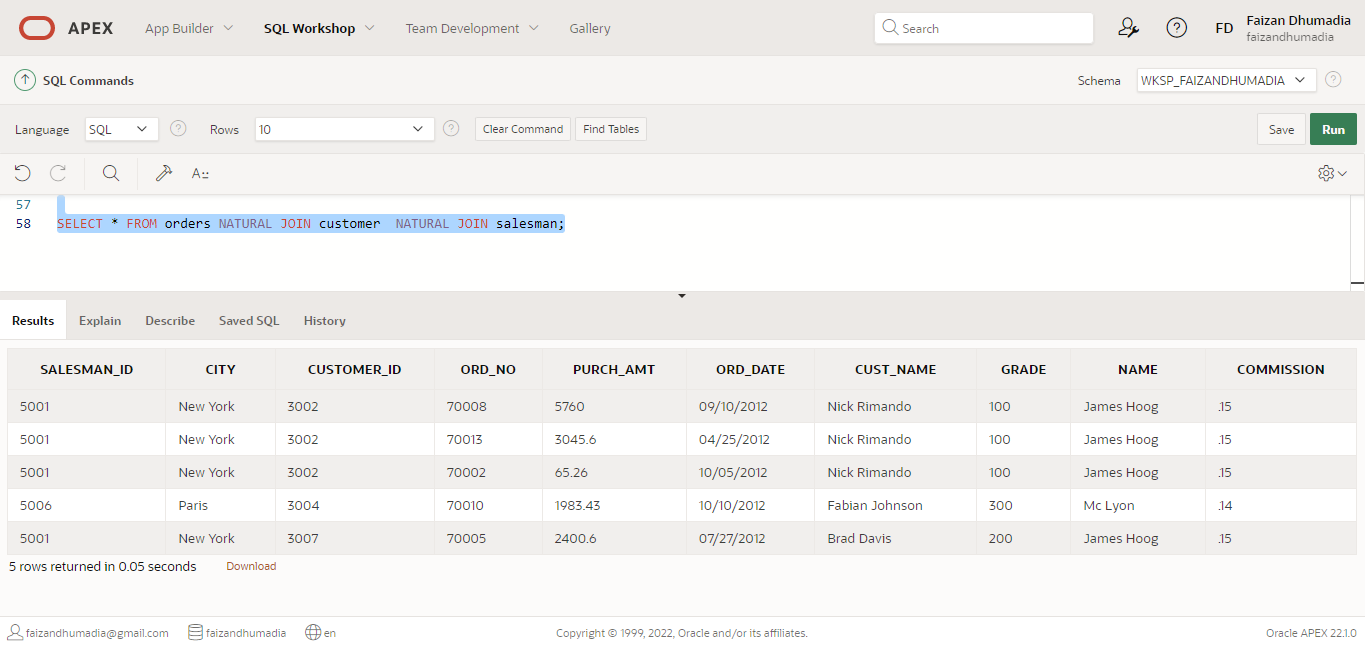
2. Prepare a list for which salesman are working for which customer along with city and commissions earned by the salesman. (use inner join/join … on condition)

SELECT salesman.name, customer.cust\_name, customer.city   
FROM salesman Cross Join customer WHERE salesman.city=customer.city;



3. Write a query to make a join on the tables salesman and orders in such a form that the same column of each table will appear once and only the relational rows will come.

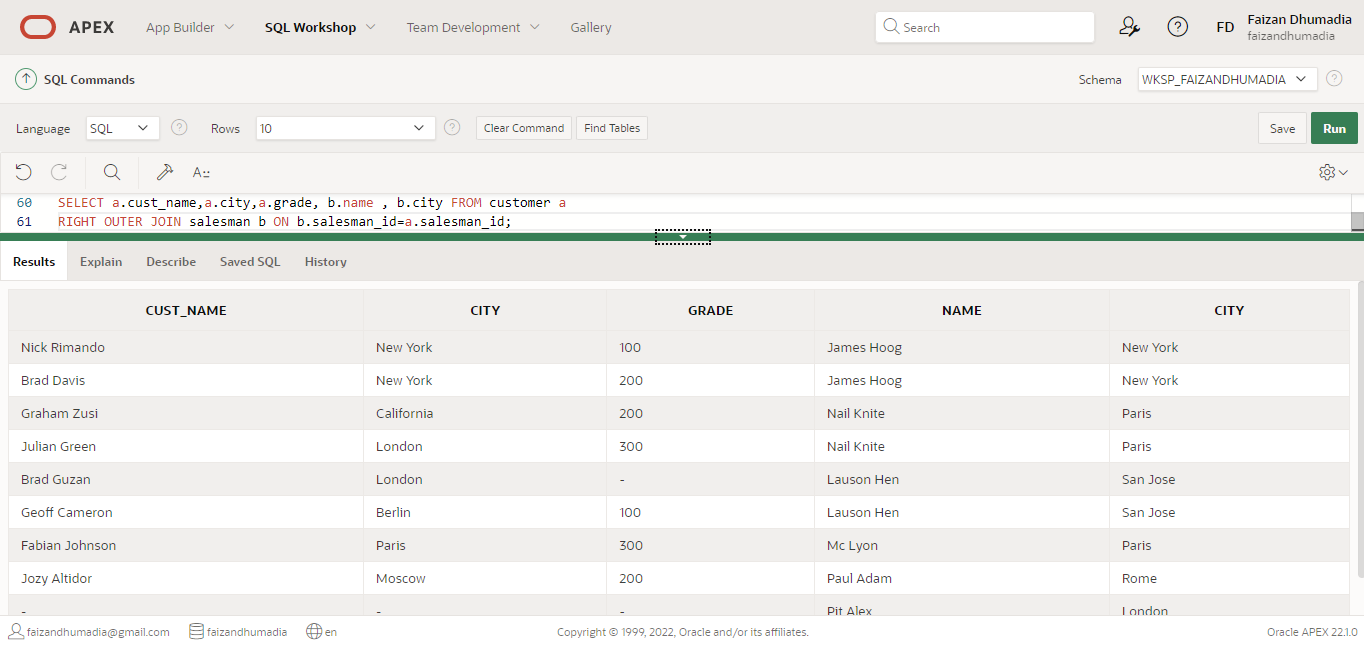
SELECT \* FROM orders NATURAL JOIN customer NATURAL JOIN salesman;



4. Prepare a list for salesman who works either for one or more customer or not yet join under any of the customer (use right outer join)

SELECT a.cust\_name,a.city,a.grade, b.name , b.city FROM customer a

RIGHT OUTER JOIN salesman b ON b.salesman\_id=a.salesman\_id;



5. Prepare a list in ascending order for the salesman who works either for one or more customer or not yet join under any of the customer ( Use left outer join)

SELECT a.cust\_name,a.city,a.grade, b.name , b.city FROM customer a

LEFT OUTER JOIN salesman b ON b.salesman\_id=a.salesman\_id ORDER BY a.cust\_name;

