Answers of Assignment 3

Part I: Table Creation:

The table creation accounts for 40 points (5’ * 8 tables). We create the 8 required tables with the following statements.

- product table

```
CREATE TABLE Product(
product_id varchar2(8),
product_name varchar2(80) NOT NULL,
price number(3),
product_date number(8),
PRIMARY KEY (product_id)
);
```

- manu table

```
CREATE TABLE Manu(
manu_id number(8),
manu_first_name varchar2(80),
manu_last_name varchar2(80),
PRIMARY KEY (manu_ID)
);
```

- manu_address table

```
CREATE TABLE Manu_address(
manu_id number(8),
address varchar2(80),
namu_phone number(8),
FOREIGN KEY (manu_id) REFERENCES Manu(manu_id) ON DELETE CASCADE
);
```

- prerequisite table

```
CREATE TABLE Prerequisite(
main_product_id varchar2(8),
prereq_product_id varchar2(8),
PRIMARY KEY (main_product_id, prereq_product_id),
FOREIGN KEY (main_product_id) REFERENCES Product(product_id) ON DELETE CASCADE,
FOREIGN KEY (prereq_product_id) REFERENCES Product(product_id) ON DELETE CASCADE
);
```

- offering table

```
CREATE TABLE Offering(
product_id varchar2(8),
offering_no number(8),
store number(5),
no_of_purchase number(5),
PRIMARY KEY (product_id, offering_no),
FOREIGN KEY (product_id) REFERENCES Product(product_id) ON DELETE CASCADE
);
```

- manu_product table

```
CREATE TABLE Manu_product(
manu_id number(8),
product_ID varchar2(8),
offering_no number(8),
PRIMARY KEY (manu_id, product_ID, offering_no),
FOREIGN KEY (manu_id) REFERENCES Manu(manu_id) ON DELETE CASCADE,
FOREIGN KEY (product_id, offering_no) REFERENCES Offering(product_id, offering_no) ON DELETE CASCADE
);
```

- retailor table

```
CREATE TABLE Retailor(
retailor_id number(8),
retailor_first_name varchar2(80),
retailor_last_name varchar2(80),
region_code number(3),
phone number(8),
product_id varchar2(8) NOT NULL,
offering_no number(8) NOT NULL,
PRIMARY KEY (retailor_id),
FOREIGN KEY (product_id, offering_no) REFERENCES Offering(product_id, offering_no) ON DELETE CASCADE
);
```

- manu_retailor table

```
CREATE TABLE Manu_retailor(
manu_id number(8),
retailor_id number(8),
PRIMARY KEY (manu_id, retailor_id),
FOREIGN KEY (manu_id) REFERENCES Manu(manu_id) ON DELETE CASCADE,
FOREIGN KEY (retailor_id) REFERENCES Retailor(retailor_id) ON DELETE CASCADE
);
```

Grading:

For each table, any error (e.g., missing columns or wrong types) will lead to the deduction of 2'. If there are more than 2 errors or failure of creation, no points will be granted.

Test:

We will test your table with the following sql:

```
insert into Product values ('12345678','Model_A1',3,20200203);
insert into Product values ('22345678','Model_B1',7,20200102);
insert into Product values ('32345678','Model_A2',5,20200103);

insert into Manu values (00000001, 'Mike','James');
insert into Manu values (00000002, 'LeBron','James');
insert into Manu values (00000003, 'Vince','Carter');

insert into Manu_address values (00000001,'address1','57490000');
insert into Manu_address values (00000002,'address2','57490001');
insert into Manu_address values (00000003,'address3','57490002');

insert into Prerequisite values ('32345678','12345678');
insert into Prerequisite values ('22345678','12345678');
insert into Prerequisite values ('22345678','32345678');

insert into Offering values ('12345678',66666666,10000,322);
insert into Offering values ('12345678',66666667,20000,415);
insert into Offering values ('12345678',66666668,30000,412);
insert into Offering values ('12345678',66666669,5000,0);
insert into Offering values ('22345678',66666670,20000,0);
insert into Offering values ('22345678',66666671,10000,0);
```

```
insert into Manu_product values (00000001,'12345678',66666666);
insert into Manu_product values (00000001,'12345678',66666667);
insert into Manu_product values (00000001,'12345678',66666668);
insert into Manu_product values (00000002,'12345678',66666669);
insert into Manu_product values (00000002,'22345678',66666670);
insert into Manu_product values (00000003,'22345678',66666671);

insert into Retailor values (22222222,'Jack','Ma',086,34567890,'12345678',66666666);
insert into Retailor values (44444444,'Huateng','Ma',852,34123456,'12345678',66666667);
insert into Retailor values (33333333,'Qiangdong','Liu',852,67893456,'12345678',66666668);

insert into Manu_retailor values (00000001,22222222);
insert into Manu_retailor values (00000001,33333333);
insert into Manu_retailor values (00000001,44444444);
```

Part II: Queries:

The tables will be initialized with the statement above. After inserting data using the statement, you can execute your query sql. (12' * 5 queries)

Please view the details and expected outputs in your output, if there are any missing attributes or redundant attributes in the output of a query question, 5 marks will be deducted for that query question.

If a query is executable but cannot give the right answer, it will still be granted with 2 points. Queries cannot be executed with be granted with 0 point.

1. Find the product_ID for products manufactured before date "20200202" with the highest price.

```
select product_id
from product
where product_date < 20200202 and price = (
    select max(price)
    from Product
    where product_date < 20200202
);
```

- result

```
22345678
```

2. Find the retailor_ID, retailor_fisrt_name, retailor_last_name of the retailors who have product ID "12345678" and region code "852".

```
select retailor_ID, retailor_first_name,retailor_last_name
from Retailor
where region_code='852' and product_ID='12345678'
```

- result

```
44444444              Huateng                Ma
33333333            Qiangdong                Liu
```

3. Find the manu_ID, last_name, first_name of all manufacturers who has the offerings with store more than 10000 of “12345678”.

```
select manu_ID, manu_last_name, manu_first_name
from Manu
where manu_ID in (
    select manu_ID from Manu_product
    where offering_no in (
        select offering_no from Offering
        where store>10000 and product_ID='12345678'
        )
    );
```

- result

```
1             James           Mike
```

4. Find the manu_ID, last_name, first_name of all the manufacturers who have NOT offered any of the prerequisites of “22345678”.

```
select manu_ID, manu_last_name, manu_first_name
from Manu
where manu_ID not in (
    select manu_ID from Manu_product
    where product_ID in (
        select prereq_product_ID
        from Prerequisite
        where main_product_id='22345678'
        )
    );
```

- result

```
3            Carter          Vince
```

5. Find the manu_ID, manu_first_name of all the manufacturers whose last name is “James” and have produced 'Model_A1' but not 'Model_B1'. (In this place, 'Model_A1' and 'Model_B1' are the sample product names.)

```
select manu_ID, manu_first_name from Manu
where manu_last_name='James' and
manu_ID in (select manu_id from Manu_product
where product_ID=(select product_ID
                    from Product where product_name='Model_A1'))
and manu_ID not in (select manu_ID from Manu_product
where product_ID=(select product_ID
                    from Product where product_name='Model_B1'))
```

- result

```
1            Mike
```