

Group-3

NAME	ID
TAZDIK HOSSAIN	21-45116-2
MD.MESBAH UDDIN	19-40282-1
MOHAMMAD SHAWKAT HOSSAIN	21-45130-2
MD.HARUN OR RASHID	21-44586-1

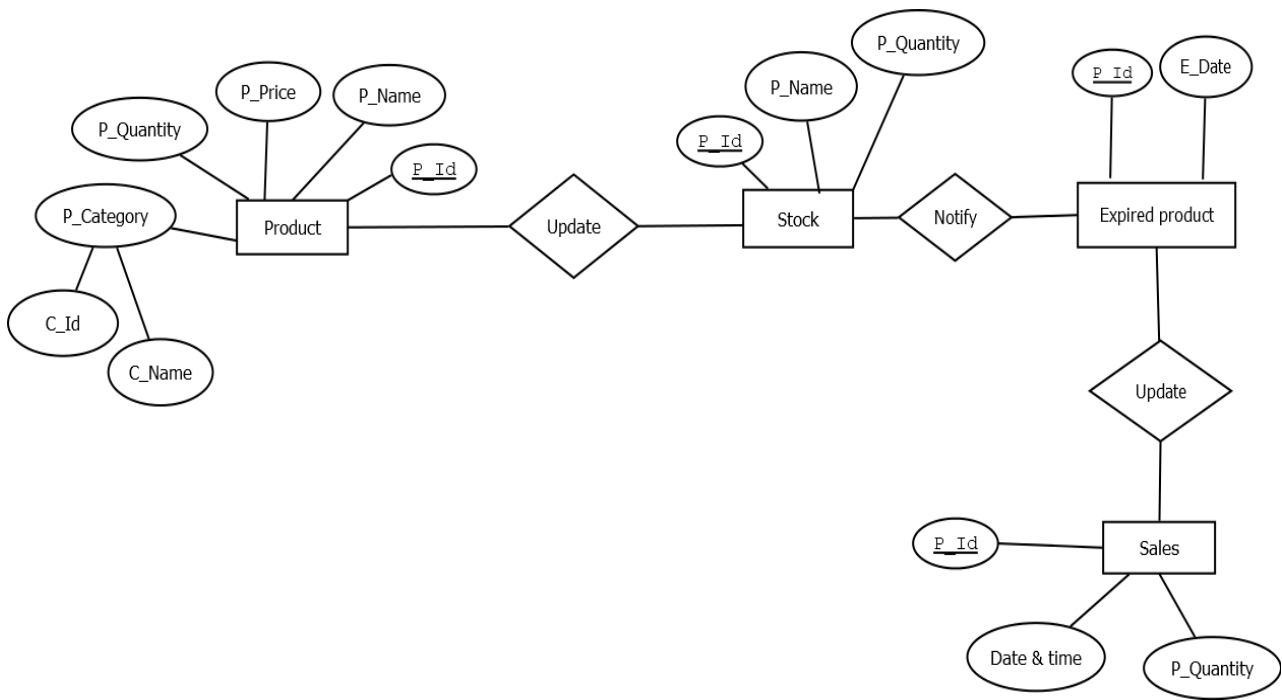
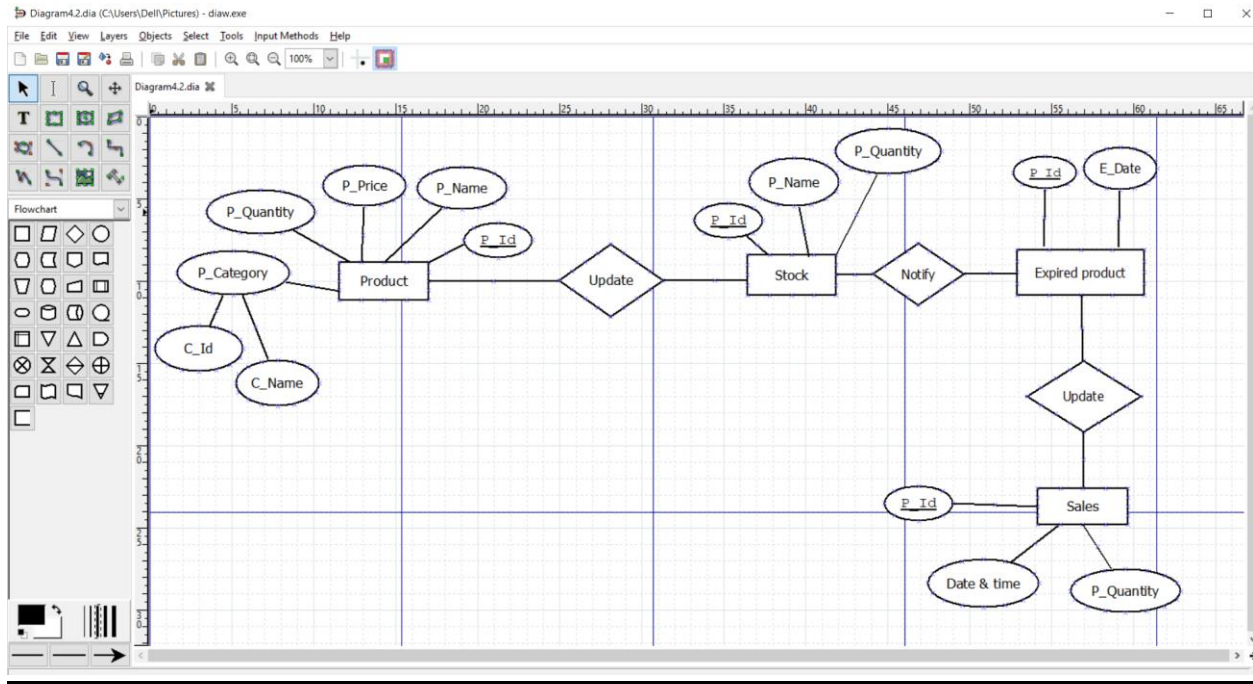
Project Title

Store Product Management System(No expired and spoild product)

Description

In a Store there are many products. This huge amount of product needs to be managed. For a customer we need to manage the products so that they can know everything about the product . Every different store product contains a different ID. Different type of product have a different name. For a different store product the price is different. In a store every product has a particular quantity. In a store management store similar products are placed in same category. For each and every category there is a different ID. This different type of categories have different names. When a customer buy a product the date and time are saved by product ID and name.It also contains the information about how many products are sold.

ER DIAGRAM:



NORMALIZATION:

Update Relation

All attributes are:

P_id, P_Name, P_Price, P_Quantity, C_id, C_Name, P_id, P_Name, P_Quantity.

1st NF:

P_id, P_Name, P_Price, P_Quantity, C_id, C_Name, P_id, P_Name, P_Quantity.

2nd NF:

P_id, P_Name, P_Price, P_Quantity.

C_id, C_Name.

P_id, P_Name, P_Quantity.

3rd NF:

P_id, P_Name, P_Price,

P_id, P_Name, P_Quantity.

C_id, C_Name.

P_id, P_Name, P_Price.

NOTIFY RELATION

All attributes are:

P_id,P_Name,P_Quantity,P_id,E_Date.

1st NF:

P_id,P_Name,P_Quantity,P_id,E_Date.

2nd NF:

P_id,P_Name,P_Quantity

P_id,E_Date

3rd NF:

P_id,P_Name,P_Quantity.

P_id,E_Date.

P_id,P_Name,E_date,

P_id,P_Quantity,E_Date

UPDATE RELATION

All attributes are:

E_Date, P_id, Date&Time, P_Quantity.

1st NF:

E_Date, P_id, Date&Time, P_Quantity.

2nd NF:

P_id, E_Date

P_id, Date&Time, P_Quantity

3rd NF:

P_id, E_Date

P_id, Date&Time, P_Quantity

P_id, E_Date, P_Quantity

P_id, E_Date, Date&Time

FINAL TABLE:

P_id,P_Name,P_Price

P_id,P_Name,P_Quantity

C_id,C_Name

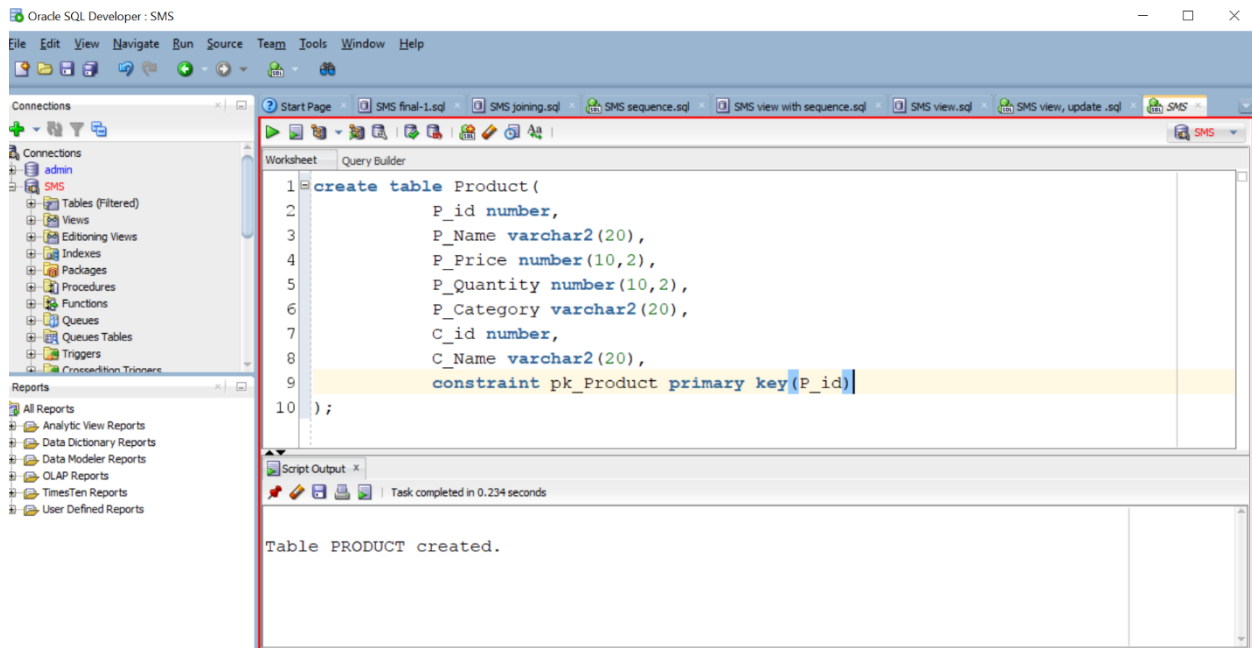
P_id,P_Name,E_Date

P_id,P_Quantity,E_Date

P_id,E_Date,Date&Time

TABLE CREATION WITH DATA INSERTION:

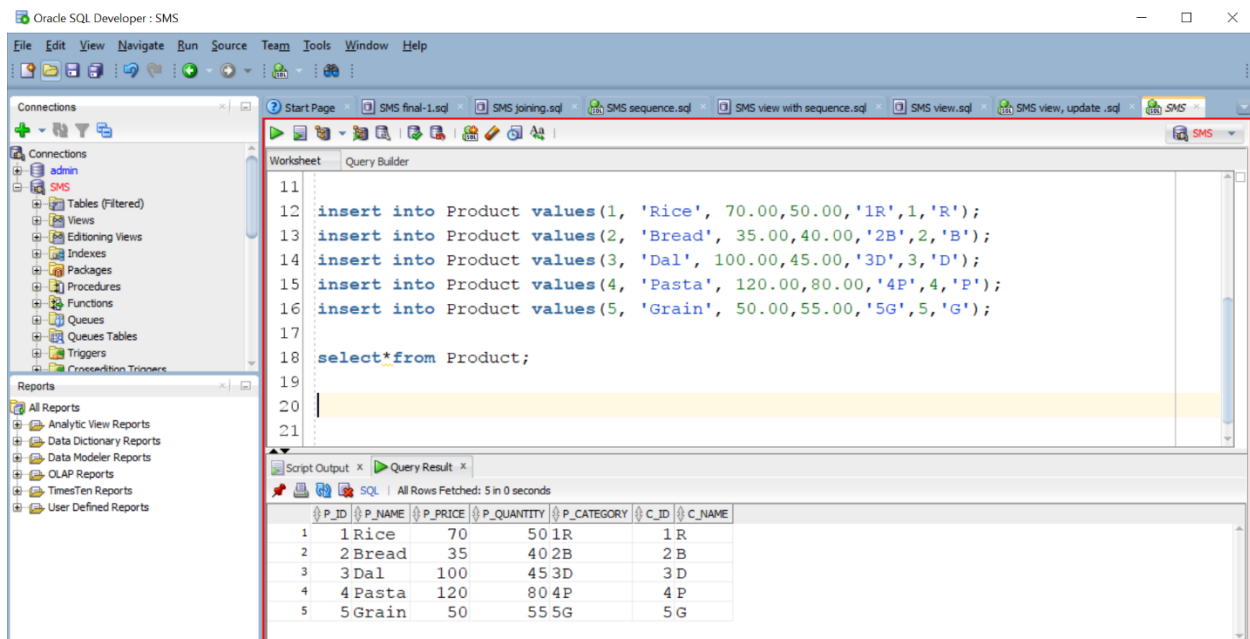
```
create table Product(  
    P_id number,  
    P_Name varchar2(20),  
    P_Price number(10,2),  
    P_Quantity number(10,2),  
    P_Category varchar2(20),  
    C_id number,  
    C_Name varchar2(20),  
    constraint pk_Product primary key(P_id)  
);
```



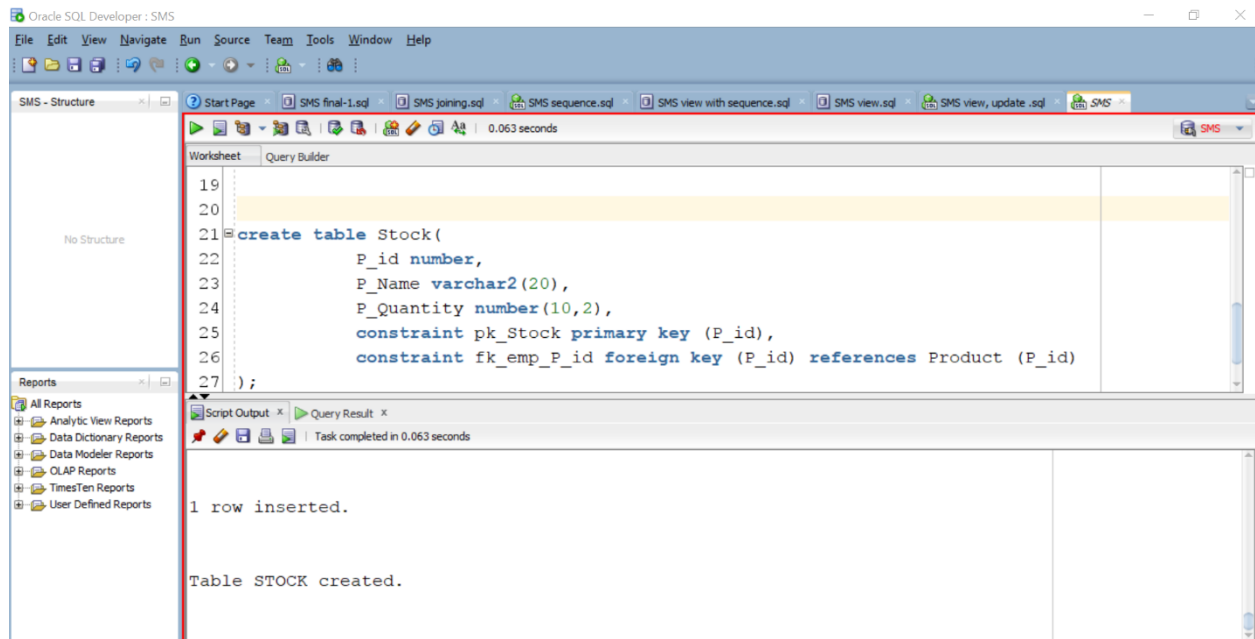
```

insert into Product values(1, 'Rice', 70.00,50.00,'1R',1,'R');
insert into Product values(2, 'Bread', 35.00,40.00,'2B',2,'B');
insert into Product values(3, 'Dal', 100.00,45.00,'3D',3,'D');
insert into Product values(4, 'Pasta', 120.00,80.00,'4P',4,'P');
insert into Product values(5, 'Grain', 50.00,55.00,'5G',5,'G');
select*from Product;

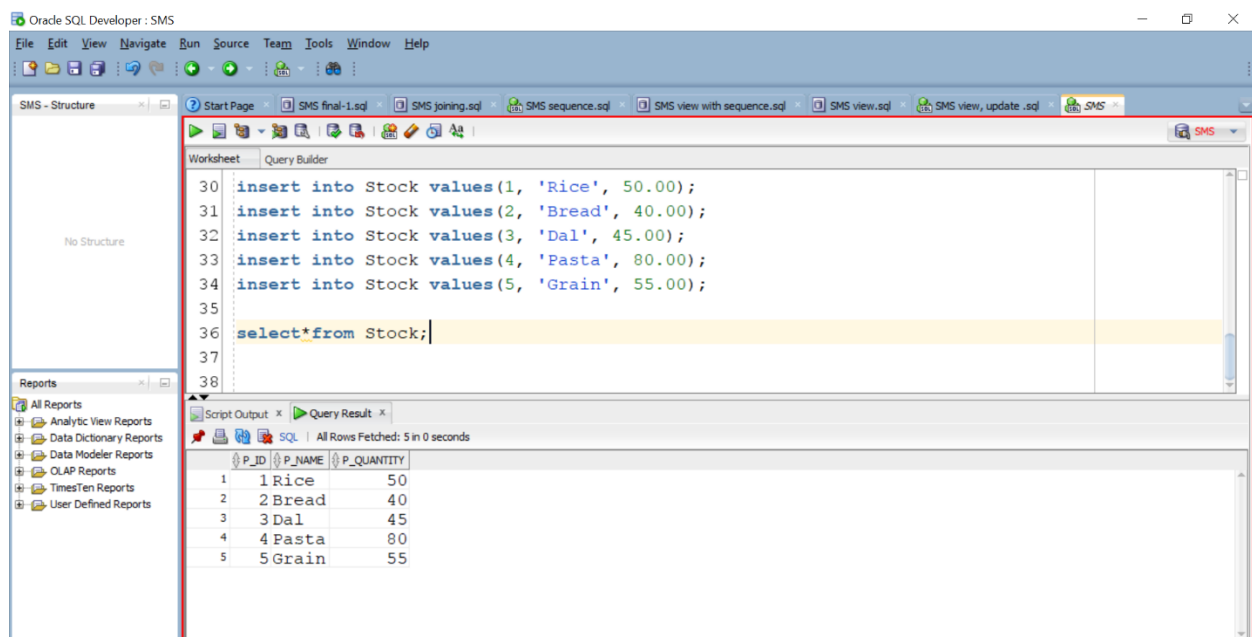
```



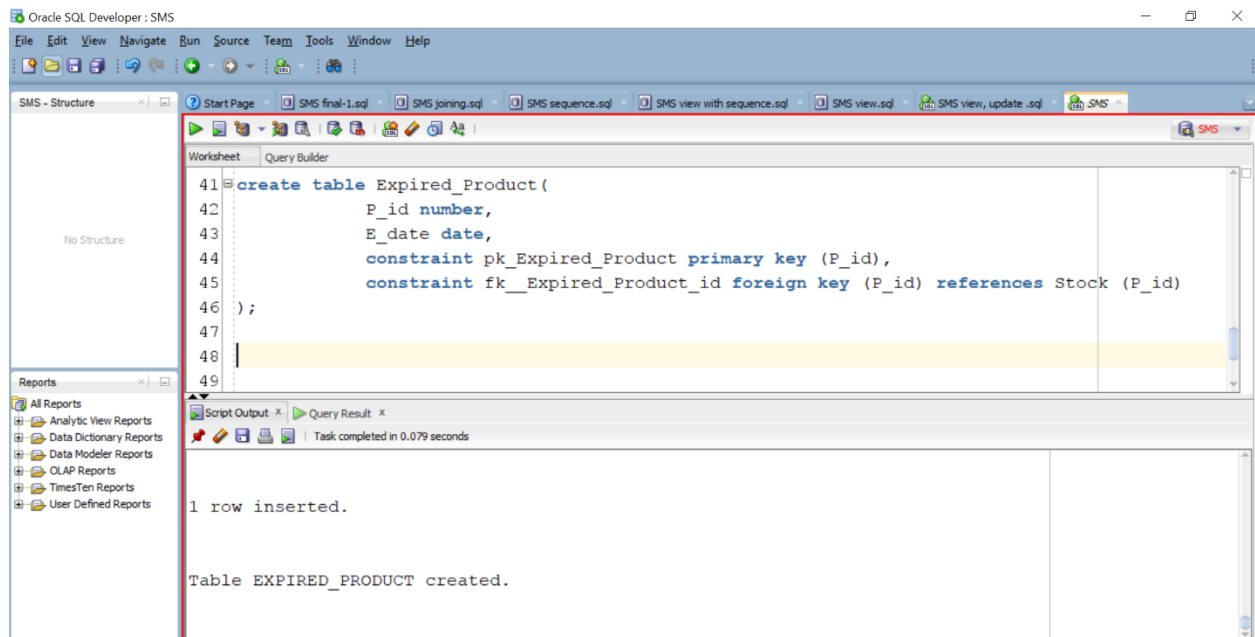

```
create table Stock(  
    P_id number,  
    P_Name varchar2(20),  
    P_Quantity number(10,2),  
    constraint pk_Stock primary key (P_id),  
    constraint fk_emp_P_id foreign key (P_id) references  
Product (P_id)  
);
```



```
insert into Stock values(1, 'Rice', 50.00);  
insert into Stock values(2, 'Bread', 40.00);  
insert into Stock values(3, 'Dal', 45.00);  
insert into Stock values(4, 'Pasta', 80.00);  
insert into Stock values(5, 'Grain', 55.00);  
  
select*from Stock;
```



```
create table Expired_Product(  
    P_id number,  
    E_date date,  
    constraint pk_Expired_Product primary key (P_id),  
    constraint fk__Expired_Product_id foreign key (P_id)  
references Stock (P_id)  
);
```



```
insert into Expired_Product values(1, TO_DATE('2022-02-17','YYYY-MM-DD'));
```

```
insert into Expired_Product values(2, TO_DATE('2022-03-19','YYYY-MM-DD'));
```

```
insert into Expired_Product values(3, TO_DATE('2022-04-20','YYYY-MM-DD'));
```

```
insert into Expired_Product values(4, TO_DATE('2022-05-21','YYYY-MM-DD'));
```

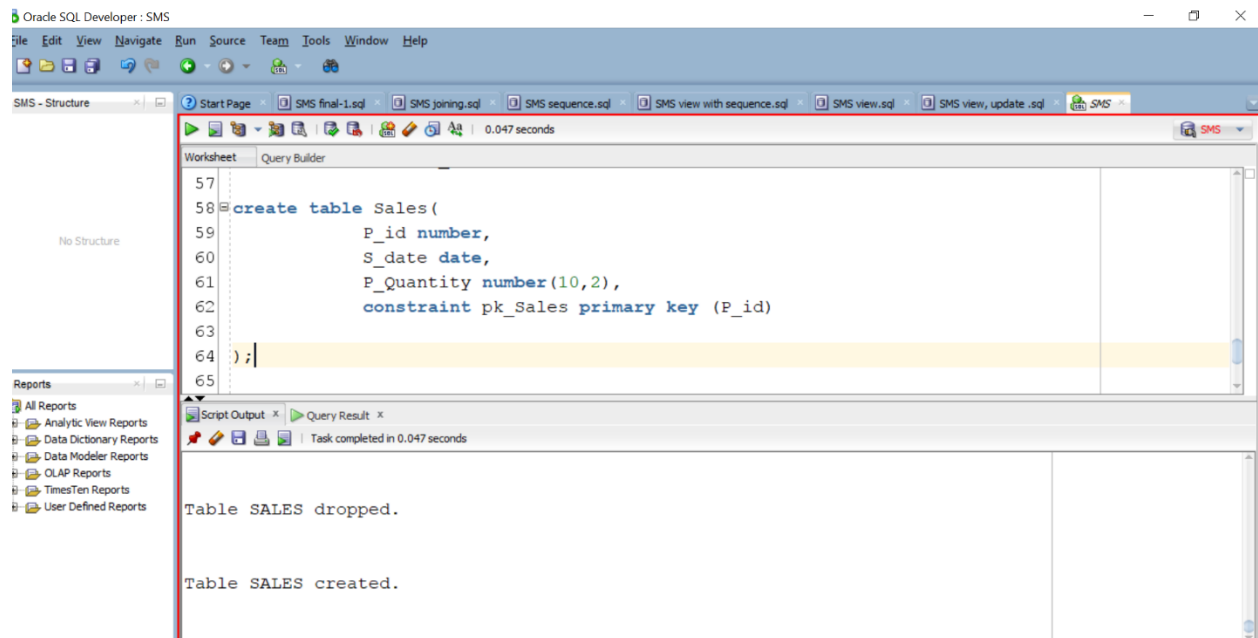
```
insert into Expired_Product values(5, TO_DATE('2022-06-22','YYYY-MM-DD'));
```

```
select*from Expired_Product;
```

The screenshot displays the Oracle SQL Developer interface. The 'Worksheet' tab shows a series of SQL statements: five INSERT statements for the 'Expired_Product' table and a SELECT statement. The 'Script Output' tab shows the execution results, indicating that all rows were fetched successfully. The 'Query Result' tab displays the data returned by the SELECT statement, showing five rows with columns P_ID and E_DATE.

P_ID	E_DATE
1	17-FEB-22
2	19-MAR-22
3	20-APR-22
4	21-MAY-22
5	22-JUN-22

```
create table Sales(  
    P_id number,  
    S_date date,  
    P_Quantity number(10,2),  
    constraint pk_Sales primary key (P_id)  
);
```



insert into Sales values(1, TO_DATE('2022-01-22','YYYY-MM-DD'),10.00);

insert into Sales values(2, TO_DATE('2022-02-3','YYYY-MM-DD'),20.00);

insert into Sales values(3, TO_DATE('2022-03-6','YYYY-MM-DD'),30.00);

insert into Sales values(4, TO_DATE('2022-01-21','YYYY-MM-DD'),15.00);

insert into Sales values(5, TO_DATE('2022-03-11','YYYY-MM-DD'),25.00);

select*from Sales;

The screenshot displays the Oracle SQL Developer interface. The main window shows a SQL script with five insert statements and a select statement. The script is as follows:

```
66 insert into Sales values(1, TO_DATE('2022-01-22','YYYY-MM-DD'),10.00);
67 insert into Sales values(2, TO_DATE('2022-02-3','YYYY-MM-DD'),20.00);
68 insert into Sales values(3, TO_DATE('2022-03-6','YYYY-MM-DD'),30.00);
69 insert into Sales values(4, TO_DATE('2022-01-21','YYYY-MM-DD'),15.00);
70 insert into Sales values(5, TO_DATE('2022-03-11','YYYY-MM-DD'),25.00);
71
72
73 select*from Sales;
74
```

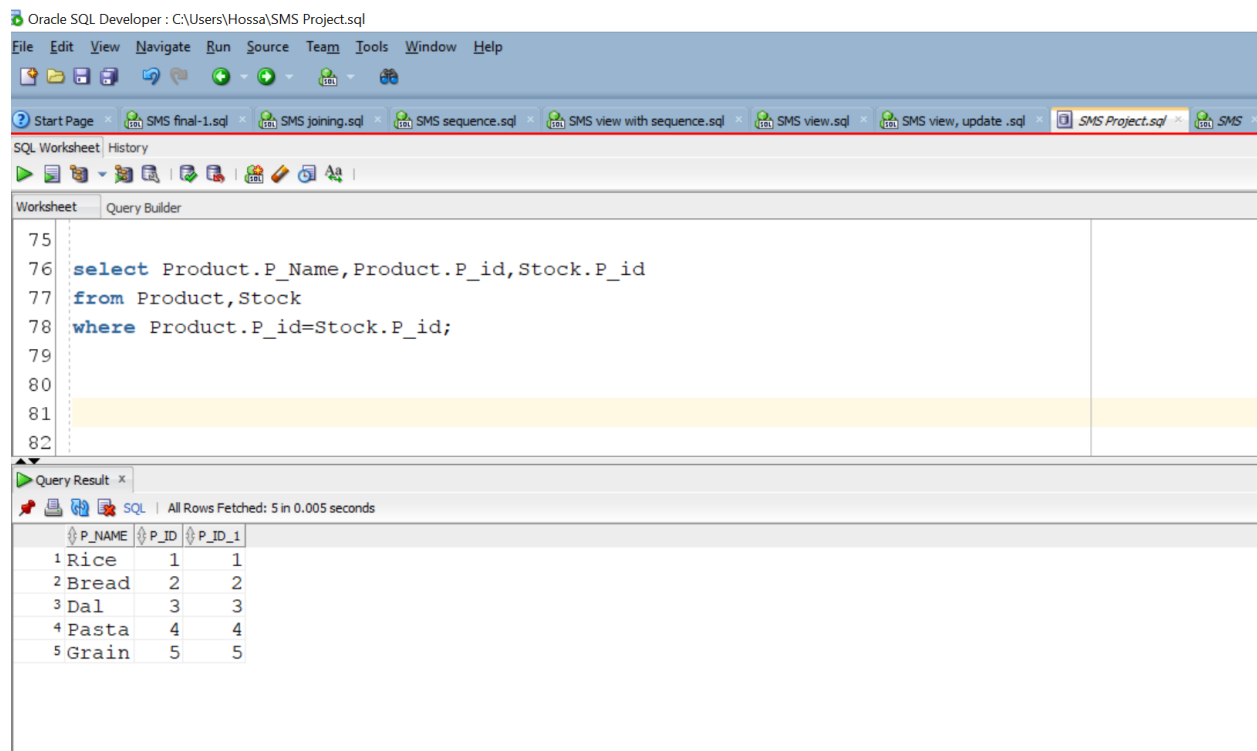
Below the script, the Query Results window shows the output of the select statement. The results are as follows:

P_ID	S_DATE	P_QUANTITY
1	22-JAN-22	10
2	03-FEB-22	20
3	06-MAR-22	30
4	21-JAN-22	15
5	11-MAR-22	25

JOINING:

Join between Product and Stock Table .

```
select Product.P_Name,Product.P_id,Stock.P_id
from Product,Stock
where Product.P_id=Stock.P_id;
```



The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the Worksheet tab. The query is as follows:

```
75
76 select Product.P_Name,Product.P_id,Stock.P_id
77 from Product,Stock
78 where Product.P_id=Stock.P_id;
79
80
81
82
```

Below the query, the Query Result tab is active, showing the results of the query. The results are displayed in a table with 3 columns: P_NAME, P_ID, and P_ID_1. The table contains 5 rows of data:

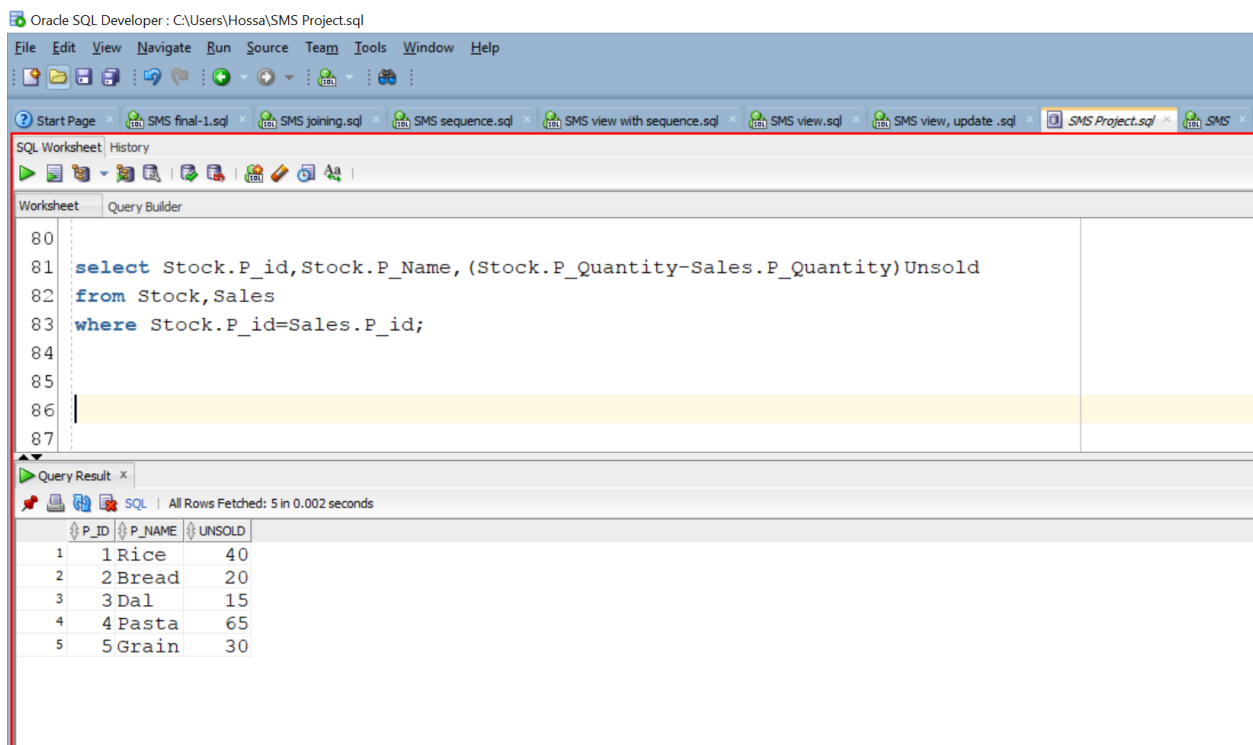
	P_NAME	P_ID	P_ID_1
1	Rice	1	1
2	Bread	2	2
3	Dal	3	3
4	Pasta	4	4
5	Grain	5	5

Eujoin between Stock and Sales table to show the UNSOLD product quantity.

```
select Stock.P_id,Stock.P_Name,(Stock.P_Quantity-  
Sales.P_Quantity)Unsold
```

```
from Stock,Sales
```

```
where Stock.P_id=Sales.P_id;
```



The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the 'Worksheet' tab. The query is as follows:

```
80  
81 select Stock.P_id,Stock.P_Name,(Stock.P_Quantity-Sales.P_Quantity)Unsold  
82 from Stock,Sales  
83 where Stock.P_id=Sales.P_id;  
84  
85  
86  
87
```

Below the query editor, the 'Query Result' tab is active, showing the results of the query. The status bar indicates 'All Rows Fetched: 5 in 0.002 seconds'. The results are displayed in a table with the following columns: P_ID, P_NAME, and UNSOLD.

P_ID	P_NAME	UNSOLD
1	Rice	40
2	Bread	20
3	Dal	15
4	Pasta	65
5	Grain	30

Eujoin between Stock and EXPIRED PRODUct to show the expired product date

select

Stock.P_id,Stock.P_Name,Stock.P_Quantity,Expired_Product.P_id,Expired_Product.E_date

from Stock,Expired_Product

where Stock.P_id=Expired_Product.P_id;

Oracle SQL Developer : C:\Users\Hossa\SMS Project.sql

File Edit View Navigate Run Source Team Tools Window Help

Start Page SMS final-1.sql SMS joining.sql SMS sequence.sql SMS view with sequence.sql SMS view.sql SMS view, update .sql SMS Project.sql SMS

SQL Worksheet History

Worksheet Query Builder

```
85
86 select Stock.P_id,Stock.P_Name,Stock.P_Quantity,Expired_Product.P_id,Expired_Product.E_date
87 from Stock,Expired_Product
88 where Stock.P_id=Expired_Product.P_id;
89
90
91
92
```

Query Result x

SQL All Rows Fetched: 5 in 0.01 seconds

	P_ID	P_NAME	P_QUANTITY	P_ID_1	E_DATE
1	1	Rice	50	1	17-FEB-22
2	2	Bread	40	2	19-MAR-22
3	3	Dal	45	3	20-APR-22
4	4	Pasta	80	4	21-MAY-22
5	5	Grain	55	5	22-JUN-22

Outer join to creat a relation between two table.

select*

from Product,Sales

where Product.P_id=Sales.P_id(+);

Oracle SQL Developer : C:\Users\Hossa\SMS Project.sql

File Edit View Navigate Run Source Team Tools Window Help

Start Page SMS final-1.sql SMS joining.sql SMS sequence.sql SMS view with sequence.sql SMS view.sql SMS view, update .sql SMS Project.sql SMS

SQL Worksheet History

Worksheet Query Builder

```
89
90
91 select*
92 from Product,Sales
93 where Product.P_id=Sales.P_id(+);
94
95
96
```

Query Result x

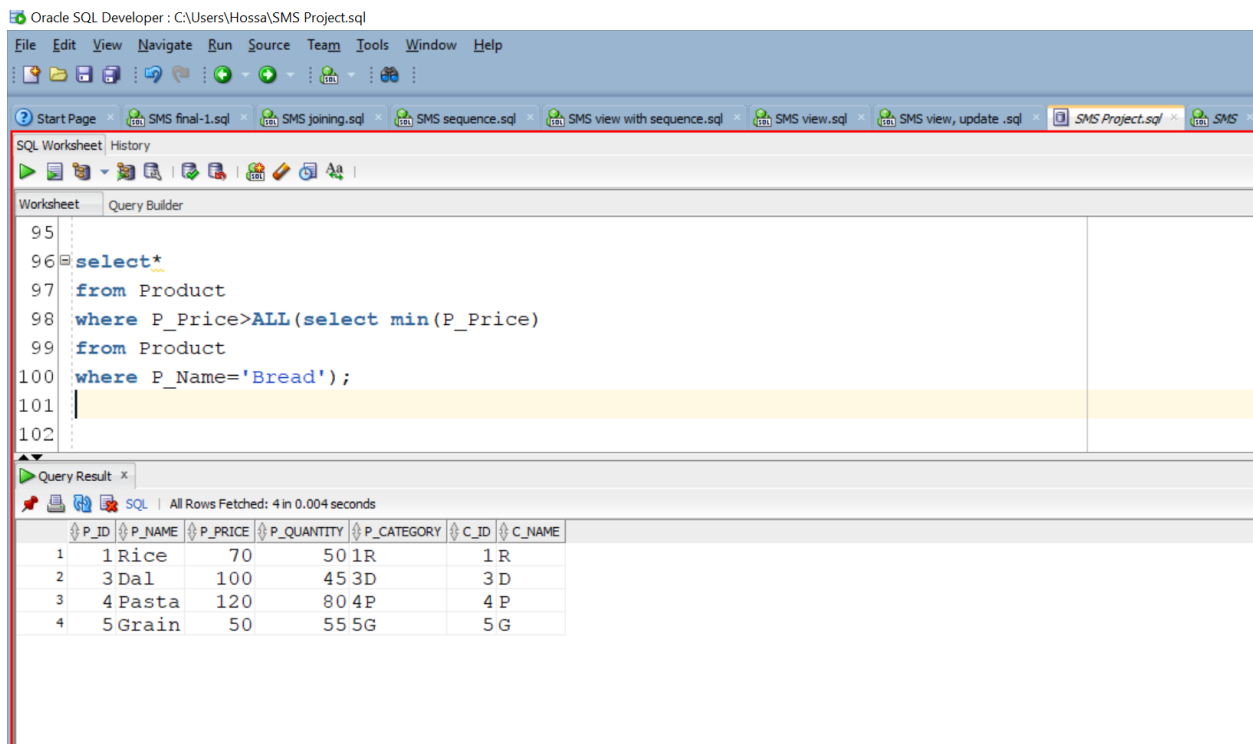
All Rows Fetched: 5 in 0.004 seconds

	P_ID	P_NAME	P_PRICE	P_QUANTITY	P_CATEGORY	C_ID	C_NAME	P_ID_1	S_DATE	P_QUANTITY_1
1	1	Rice	70	50	1R	1	R	1	22-JAN-22	10
2	2	Bread	35	40	2B	2	B	2	03-FEB-22	20
3	3	Dal	100	45	3D	3	D	3	06-MAR-22	30
4	4	Pasta	120	80	4P	4	P	4	21-JAN-22	15
5	5	Grain	50	55	5G	5	G	5	11-MAR-22	25

SUBQUERY:

Subquery to find the product price more than Bread.

```
select*  
from Product  
where P_Price>ALL(select min(P_Price)  
from Product  
where P_Name='Bread');
```



The screenshot shows the Oracle SQL Developer interface. The main window displays a SQL query in the Worksheet tab. The query is as follows:

```
95  
96 select*  
97 from Product  
98 where P_Price>ALL(select min(P_Price)  
99 from Product  
100 where P_Name='Bread');  
101  
102
```

Below the query, the Query Result tab is visible, showing the results of the query. The results are displayed in a table with the following columns: P_ID, P_NAME, P_PRICE, P_QUANTITY, P_CATEGORY, C_ID, and C_NAME. The table contains 4 rows of data:

P_ID	P_NAME	P_PRICE	P_QUANTITY	P_CATEGORY	C_ID	C_NAME
1	Rice	70	50	1R	1	R
2	Dal	100	45	3D	3	D
3	Pasta	120	80	4P	4	P
4	Grain	50	55	5G	5	G

View:

View to show the product and its expiry date.

```
create view Stock11
```

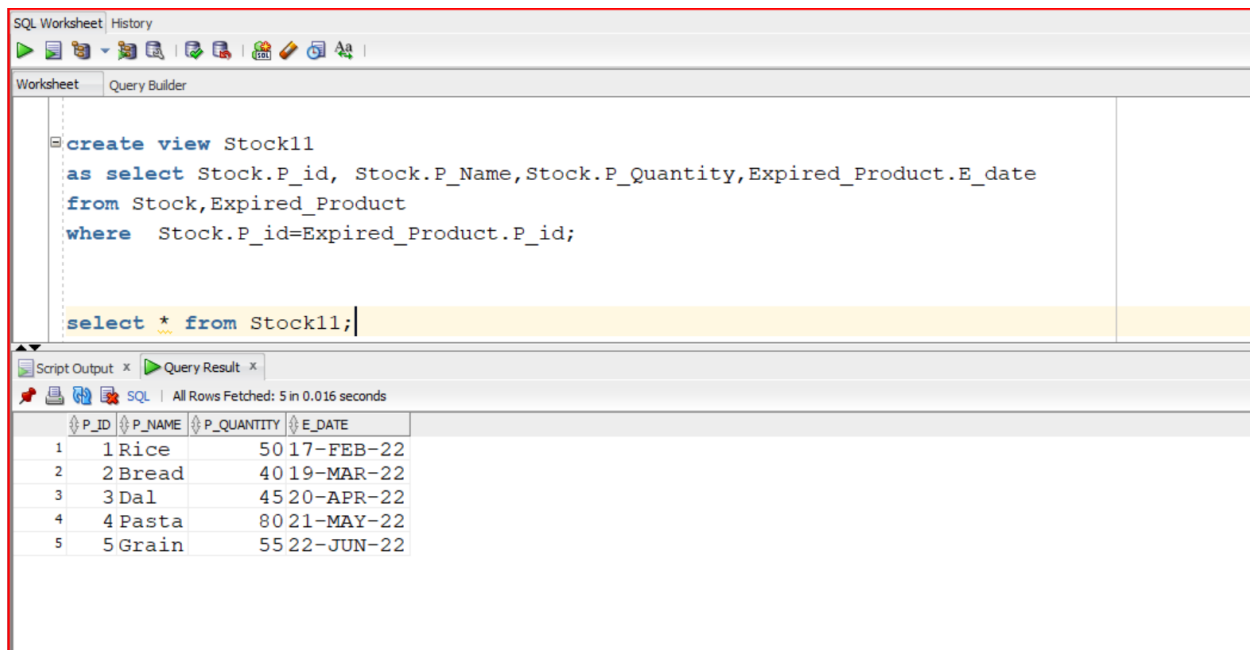
```
as select Stock.P_id,
```

```
Stock.P_Name,Stock.P_Quantity,Expired_Product.E_date
```

```
from Stock,Expired_Product
```

```
where Stock.P_id=Expired_Product.P_id;
```

```
select * from Stock11;
```



SQL Worksheet | History

Worksheet | Query Builder

```
create view Stock11
as select Stock.P_id, Stock.P_Name,Stock.P_Quantity,Expired_Product.E_date
from Stock,Expired_Product
where Stock.P_id=Expired_Product.P_id;

select * from Stock11;
```

Script Output x | Query Result x

SQL | All Rows Fetched: 5 in 0.016 seconds

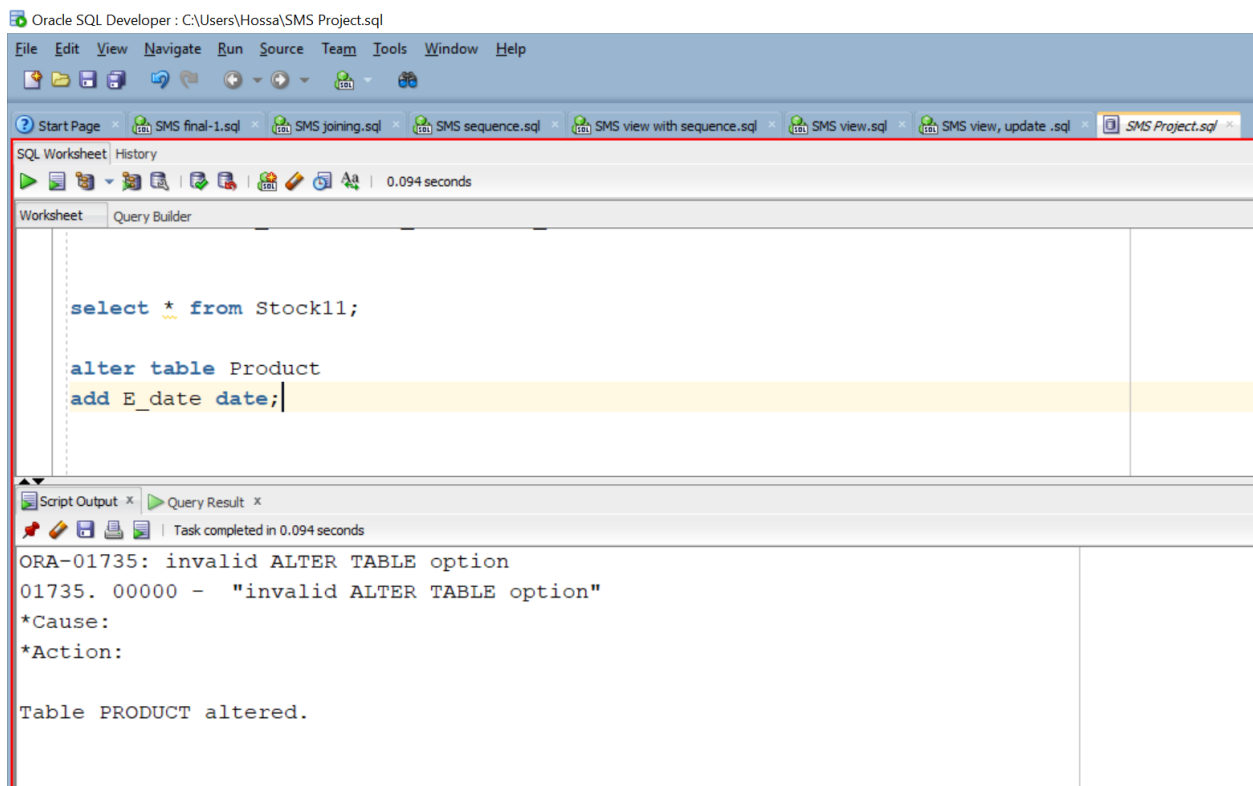
	P_ID	P_NAME	P_QUANTITY	E_DATE
1	1	Rice	50	17-FEB-22
2	2	Bread	40	19-MAR-22
3	3	Dal	45	20-APR-22
4	4	Pasta	80	21-MAY-22
5	5	Grain	55	22-JUN-22

CONSTRAINT:

Add another coloum to the product table using constraint.

alter table Product

add E_date date;



SEQUENCE:

create sequence Product_P_Quantity

increment by 1

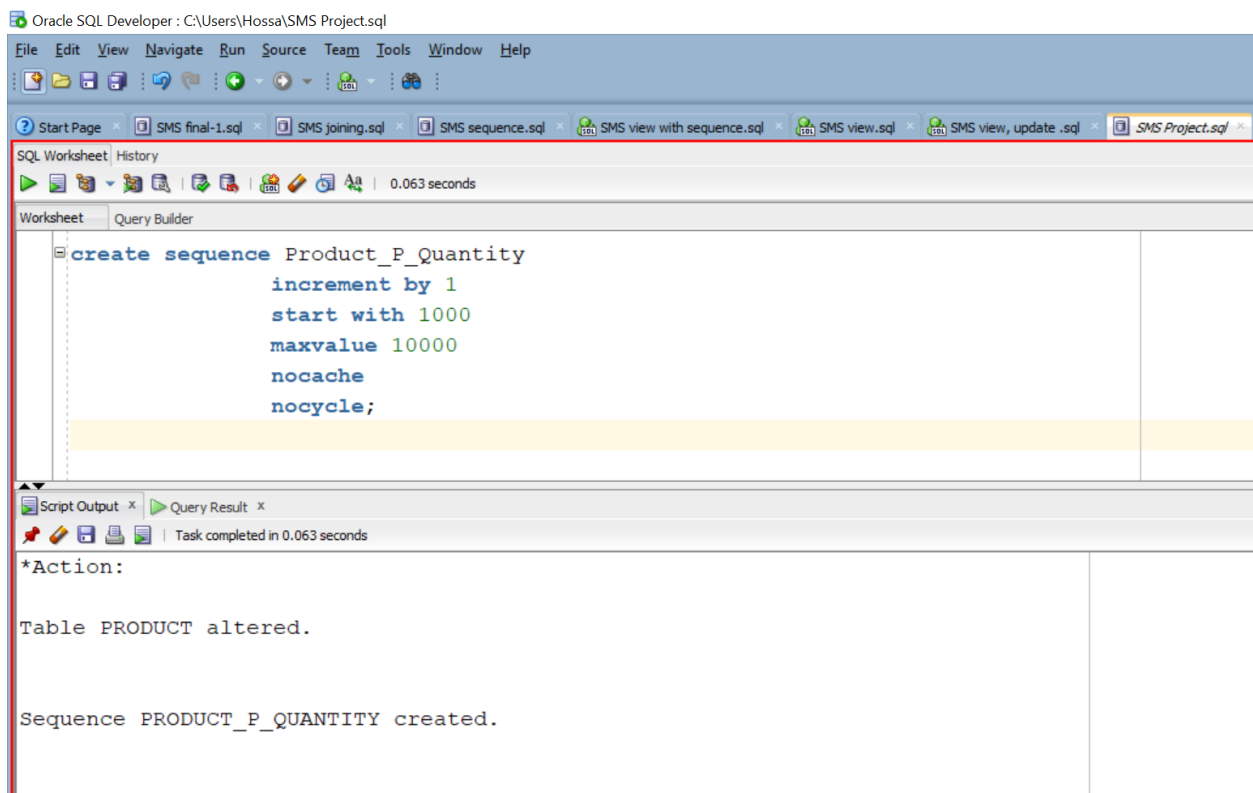
start with 1000

maxvalue 10000

nocache

nocycle;

select * from Product;



Oracle SQL Developer : C:\Users\Hossa\SMS Project.sql

File Edit View Navigate Run Source Team Tools Window Help

Start Page x SMS final-1.sql x SMS joining.sql x SMS sequence.sql x SMS view with sequence.sql x SMS view.sql x SMS view, update .sql x SMS Project.sql x

SQL Worksheet History

Worksheet Query Builder

```

create sequence Product_P_Quantity
    increment by 1
    start with 1000
    maxvalue 10000
    nocache
    nocycle;

select * from Product;

```

Script Output x Query Result x

SQL | All Rows Fetched: 5 in 0.016 seconds

P_ID	P_NAME	P_PRICE	P_QUANTITY	P_CATEGORY	C_ID	C_NAME	E_DATE
1	Rice	70	50	1R	1	R	(null)
2	Bread	35	40	2B	2	B	(null)
3	Dal	100	45	3D	3	D	(null)
4	Pasta	120	80	4P	4	P	(null)
5	Grain	50	55	5G	5	G	(null)

REPORT QUERY:

In a store different kinds of products are displayed. They have different expiry date . Considering the amount and quantity to manage the products we need to know the expiry date every time so that no product is spoild.

1. Eujoin between Product and Stock Table .
2. Eujoin between Stock and Sales table to show the UNSOLD product quantity.
3. Outer join to creat a relation between two table
4. Subquery to find the product price more than Bread.
5. View to show the product and its expiry date.
6. Add another coloum to the product table using constraint.

THANK YOU