

E-Procurement Database Management System

Bora Rajesh (207115)
Paladugu Pruthvi (207152)
Amaballa Srujan Chandra (207106)

Table of Contents

S.no.	Topics	Pg.no.
1.	Problem Statement	3
2.	Database Description	4
3.	ER Diagram	6
4.	Relational Schema	7
5.	Functional Dependencies and Normalisation	8
6.	SQL implementation: Creation of Tables	10
7.	Insertion of Values	15
8.	Displaying Tables	22

Problem Statement

E-Procurement is the purchase and sale of supplies, equipment, works, and services through a web interface or other networked system.

The aim of the project is to make a database for an E-Procurement site to manage the orders on their website and maintain all the orders and transactions made through their site. It is normalised to maximum extent to reduce data redundancy as much as possible to make the database work faster and efficiently.

In this project,

Products are sold by Sellers belonging to different companies through this E-Procurement website

Each Product is classified into Categories

Customer orders products from this E-procurement website

Order is done by payments

Once order is placed, order details and delivery dates are issued

After a successful order, Transaction reports are issued.

Database Description

Customer

{ Primary Key: Customer_ID }

The Customer table stores personal information about each customer, uniquely identified by Customer_ID. This table stores information like First_Name, Last_Name of the Customer, Address information like H.No, Street, City, State, Pin_code, District, Country, Contact information like Email id and phone number.

Payment

{ Primary Key: Payment_ID; Foreign Key: Category_ID }

This table consists of payment information like Payment date, How much amount has been paid or to be paid, Status of the payment and the type of payment. Each payment is uniquely identified by Payment_ID.

Category

{ Primary Key: Category_ID }

This table consists of categories of the particular item's information like What is the name of the category and what type of category it is. Every category is uniquely identified by category_id.

Seller

{ Primary Key: Seller_ID }

The Seller table stores personal information about each Seller, uniquely identified by Seller_ID. This table consists of information about the seller like his first_name, last_name, name of his company, city, country and pincode.

Deliveries

{ Primary Key: Delivery_ID; Foreign Key: Customer_ID }

This table consists of Delivery information of a particular order like What is the date of the delivery, Delivery status. Every record is uniquely identified by Delivery_ID.

Products

{ Primary Key: Product_ID; Foreign Key: Category_ID, Seller_ID }

This table consists of the product's information like name of the product, cost, manufacture_date. Every product is uniquely identified by product_id;

Transaction_Reports

{ Primary Key: Report_ID; Foreign Key: Payment_ID, Product_ID, Customer_ID, Order_ID }

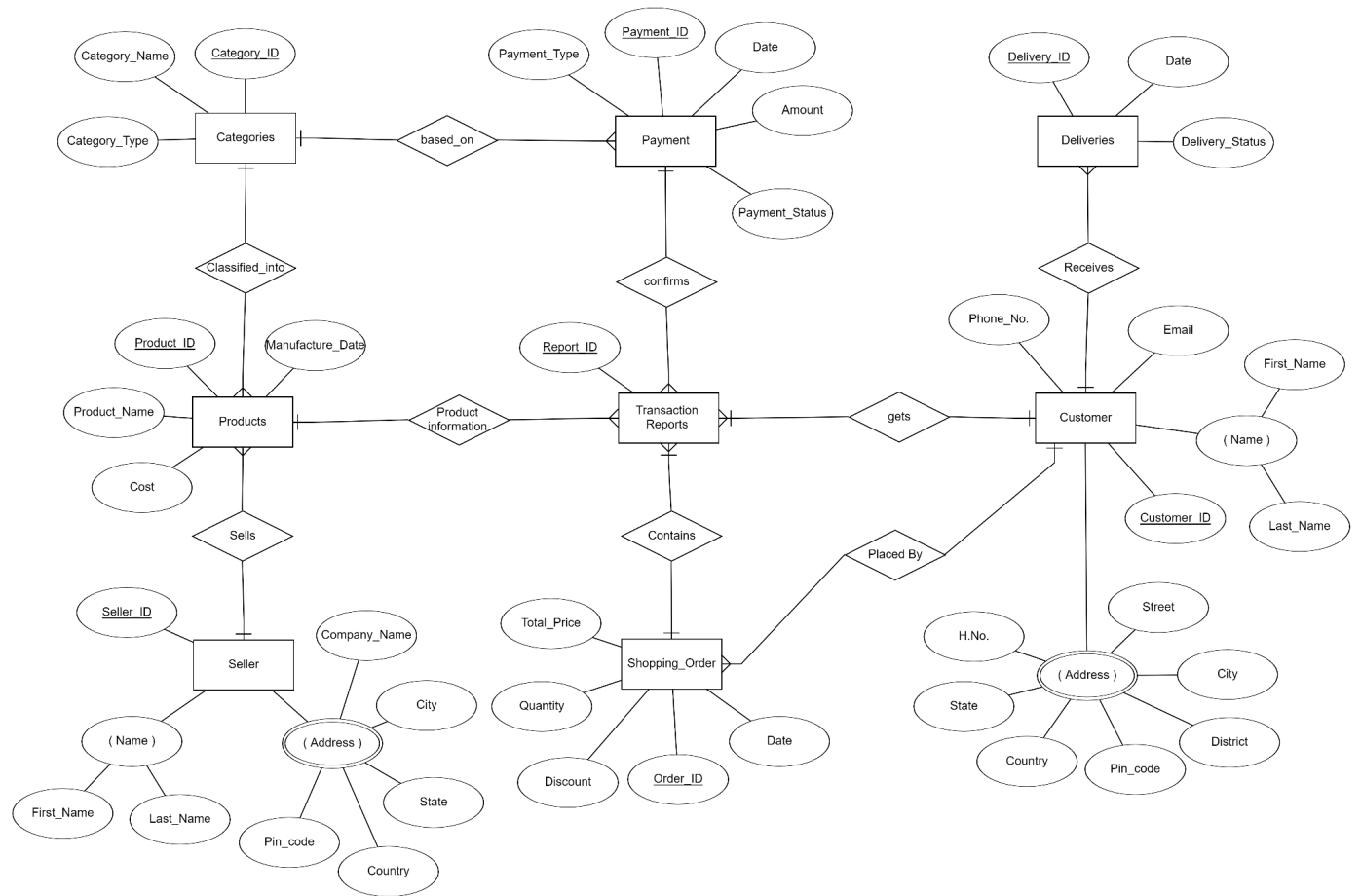
This table consists of the information regarding transactions. It consists of information like payment_id, product_id, customer_id, order_id. Every Transaction report is uniquely identified by report_id.

Shopping_Order

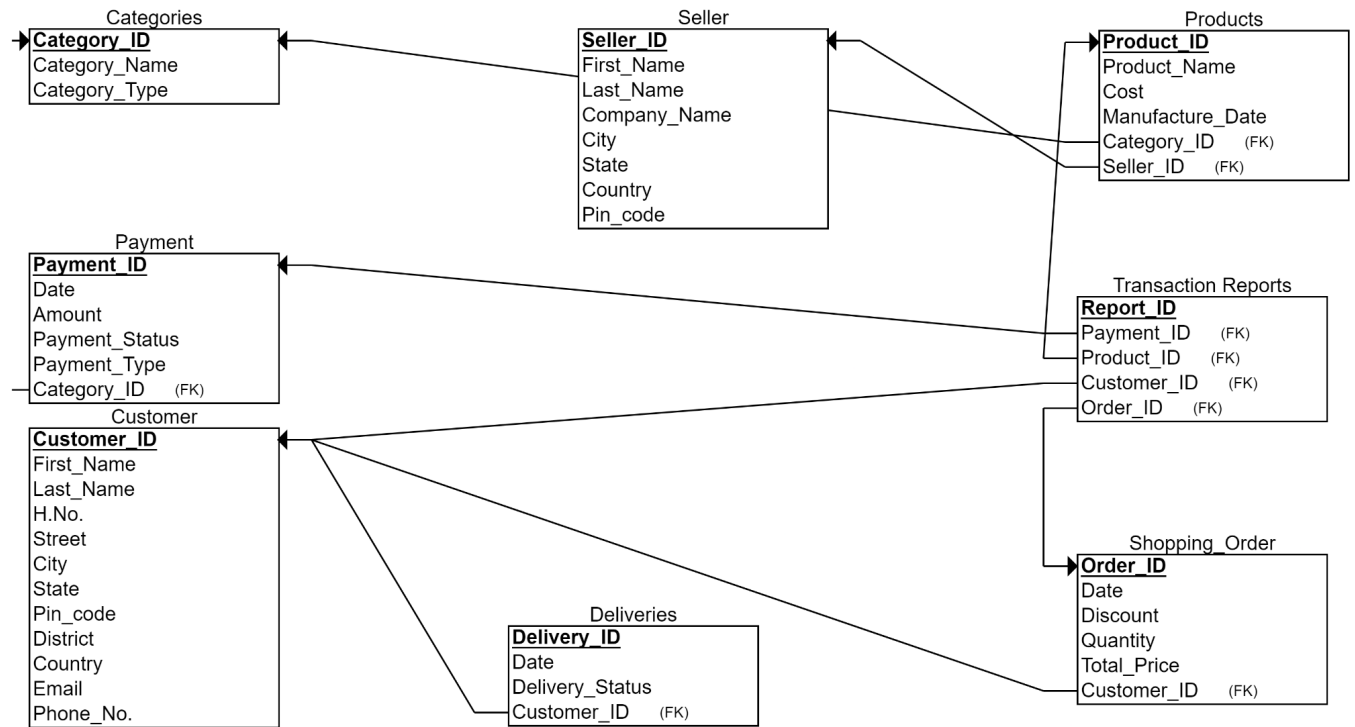
{ Primary Key: Customer_ID; Foreign Key: Customer_ID }

This table consists of information regarding shopping order details like date when order is issued, discount on the product, quantity of items purchased, total price (billing). Each order is uniquely identified by order_id.

ER Diagram of Database



Relational Schema



Functional Dependencies and Normalisation

Customer

Customer_ID → First_Name, Last_Name, H.No., Street, City, State, Pin_code, District, Country, Email, Phone_No.

- All attributes and their values are atomic and therefore in 1NF.
- All attributes are fully dependent on the primary key, Customer_ID. Therefore, the table is in 2NF.
- The relation had no transitive functional dependencies too and in 2 NF which makes the relation in 3NF.

Payment

Payment_ID → Date, Amount, Payment_Status, Payment_Type, Category_ID

- All attributes and their values are atomic and therefore in 1NF.
- All attributes are fully dependent on the primary key, Customer_ID. Therefore, the table is in 2NF.
- The relation had no transitive functional dependencies too and in 2 NF which makes the relation in 3NF.

Category

Category_ID → Category_Name, Category_Type

- All attributes and their values are atomic and therefore in 1NF.
- All attributes are fully dependent on the primary key, Customer_ID. Therefore, the table is in 2NF.
- The relation had no transitive functional dependencies too and in 2 NF which makes the relation in 3NF.

Seller

Seller_ID → First_Name, Last_Name, Company_Name, City, State, Country, Pin_code

- All attributes and their values are atomic and therefore in 1NF.
- All attributes are fully dependent on the primary key, Customer_ID. Therefore, the table is in 2NF.
- The relation had no transitive functional dependencies too and in 2 NF which makes the relation in 3NF.

Deliveries

Delivery_ID → Date, Delivery_Status, Customer_ID

- All attributes and their values are atomic and therefore in 1NF.
- All attributes are fully dependent on the primary key, Customer_ID. Therefore, the table is in 2NF.
- The relation had no transitive functional dependencies too and in 2 NF which makes the relation in 3NF.

Products

Product_ID → Product_Name, Cost, Manufacture_Date, Category_ID, Seller_ID

- All attributes and their values are atomic and therefore in 1NF.
- All attributes are fully dependent on the primary key, Customer_ID. Therefore, the table is in 2NF.
- The relation had no transitive functional dependencies too and in 2 NF which makes the relation in 3NF.

Transaction_Reports

Report_ID → Payment_ID, Product_ID, Customer_ID, Order_ID

- All attributes and their values are atomic and therefore in 1NF.
- All attributes are fully dependent on the primary key, Customer_ID. Therefore, the table is in 2NF.
- The relation had no transitive functional dependencies too and in 2 NF which makes the relation in 3NF.

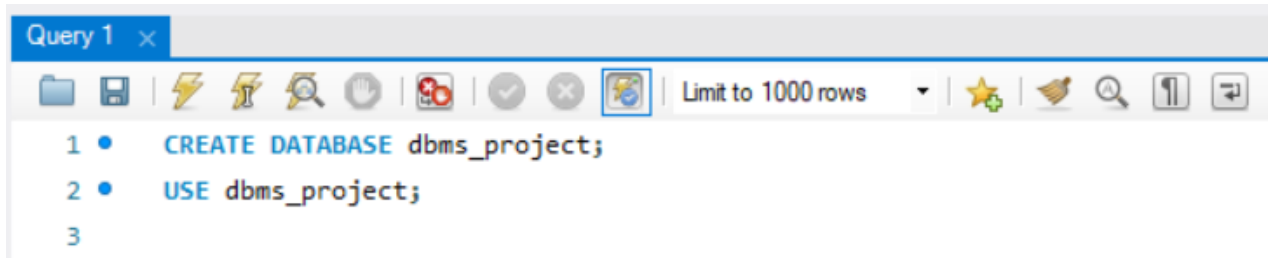
Shopping_Order

Order_ID → Date, Discount, Quantity, Total_Price, Customer_ID

- All attributes and their values are atomic and therefore in 1NF.
- All attributes are fully dependent on the primary key, Customer_ID. Therefore, the table is in 2NF.
- The relation had no transitive functional dependencies too and in 2 NF which makes the relation in 3NF.

SQL Implementation:

Creation of Database

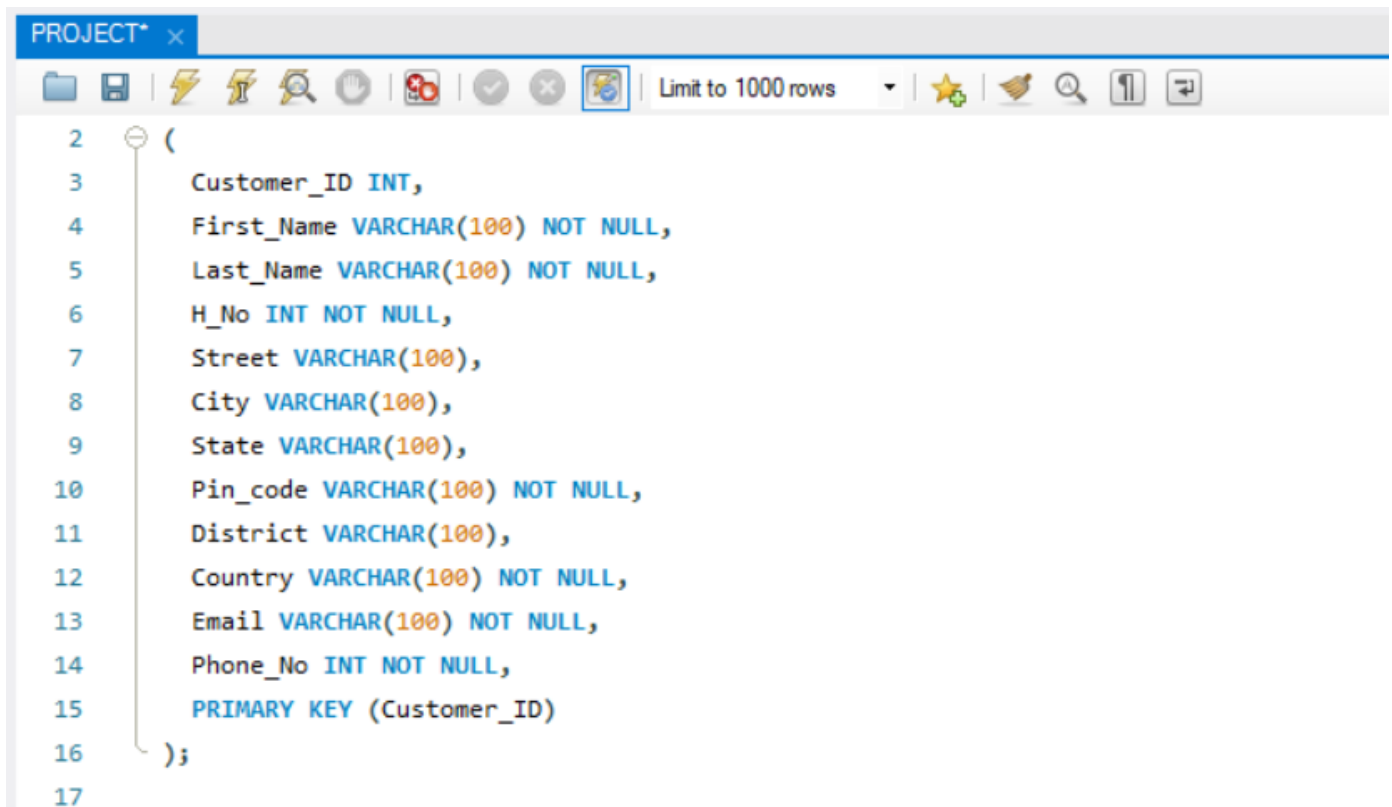


```
1 • CREATE DATABASE dbms_project;
2 • USE dbms_project;
3
```

```
CREATE DATABASE dbms_project;
USE dbms_project;
```

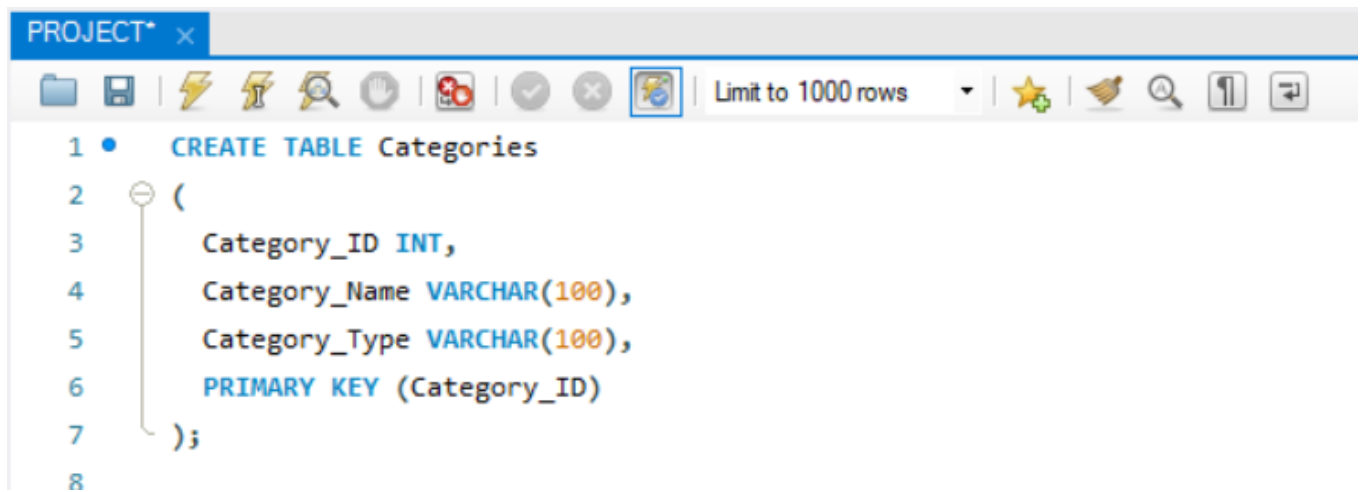
Creation of Tables:

Customer



```
2 (
3     Customer_ID INT,
4     First_Name VARCHAR(100) NOT NULL,
5     Last_Name VARCHAR(100) NOT NULL,
6     H_No INT NOT NULL,
7     Street VARCHAR(100),
8     City VARCHAR(100),
9     State VARCHAR(100),
10    Pin_code VARCHAR(100) NOT NULL,
11    District VARCHAR(100),
12    Country VARCHAR(100) NOT NULL,
13    Email VARCHAR(100) NOT NULL,
14    Phone_No INT NOT NULL,
15    PRIMARY KEY (Customer_ID)
16 );
17
```

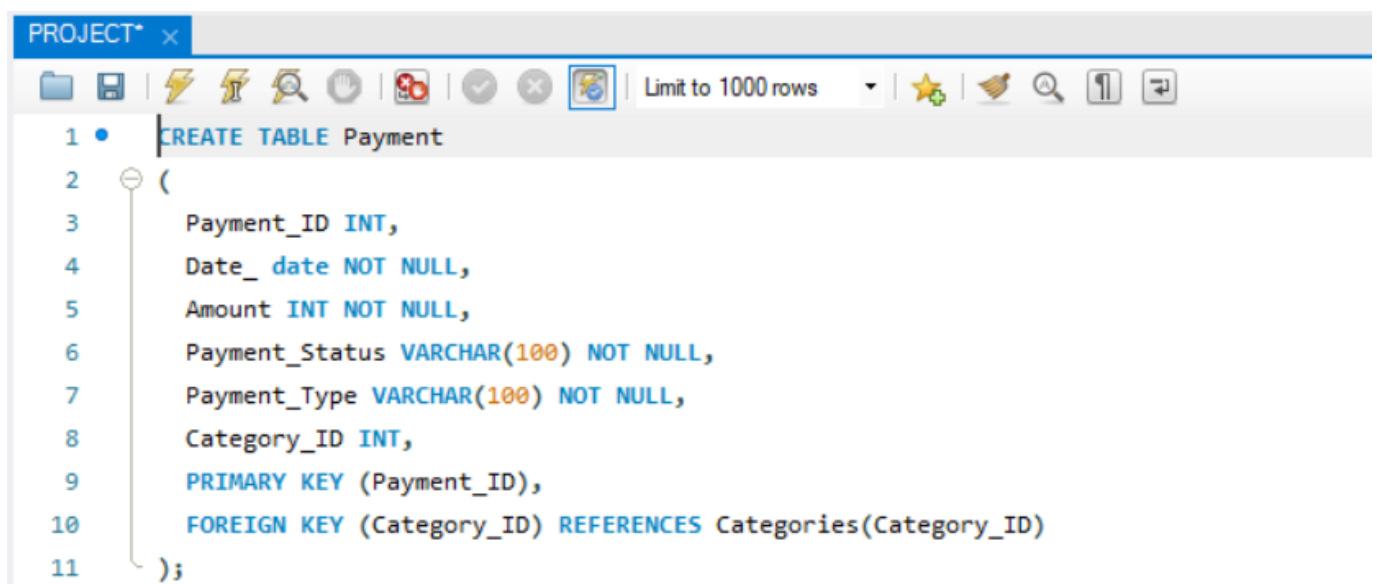
Categories



The screenshot shows a SQL IDE window titled 'PROJECT*'. The toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

```
1 • CREATE TABLE Categories
2 (
3     Category_ID INT,
4     Category_Name VARCHAR(100),
5     Category_Type VARCHAR(100),
6     PRIMARY KEY (Category_ID)
7 );
8
```

Payment



The screenshot shows a SQL IDE window titled 'PROJECT*'. The toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

```
1 • CREATE TABLE Payment
2 (
3     Payment_ID INT,
4     Date_date NOT NULL,
5     Amount INT NOT NULL,
6     Payment_Status VARCHAR(100) NOT NULL,
7     Payment_Type VARCHAR(100) NOT NULL,
8     Category_ID INT,
9     PRIMARY KEY (Payment_ID),
10    FOREIGN KEY (Category_ID) REFERENCES Categories(Category_ID)
11 );
```

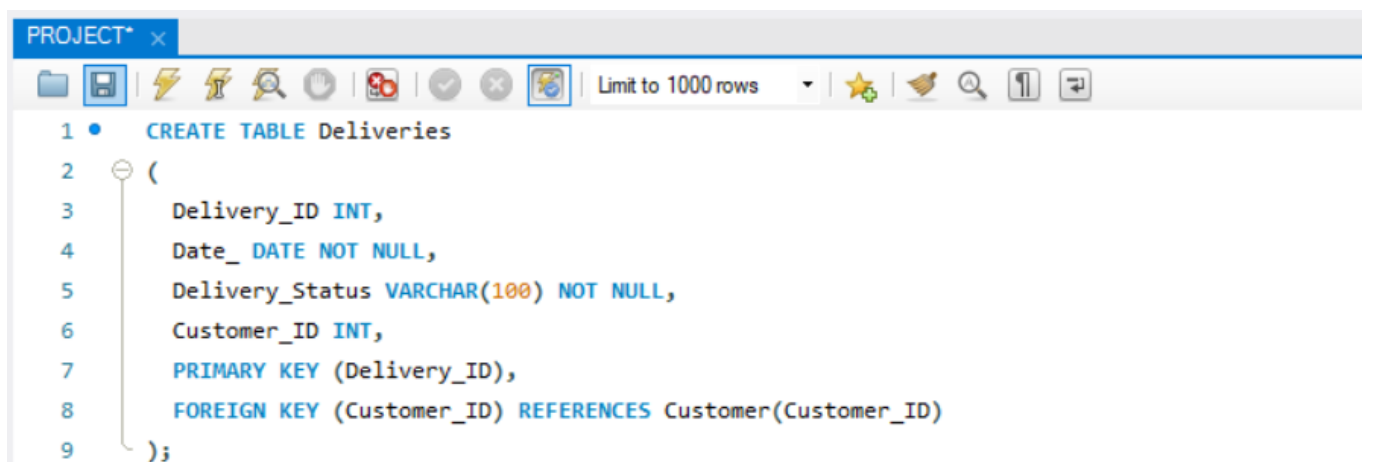
Seller



The screenshot shows a SQL IDE window titled 'PROJECT*'. The toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

```
1 • CREATE TABLE Seller
2 (
3     Seller_ID INT,
4     First_Name VARCHAR(100) NOT NULL,
5     Last_Name VARCHAR(100) NOT NULL,
6     Company_Name VARCHAR(100),
7     City VARCHAR(100),
8     State VARCHAR(100),
9     Country VARCHAR(100) NOT NULL,
10    Pin_code INT NOT NULL,
11    PRIMARY KEY (Seller_ID)
12 );
```

Deliveries



The screenshot shows a SQL IDE window titled 'PROJECT*'. The toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

```
1 • CREATE TABLE Deliveries
2 (
3     Delivery_ID INT,
4     Date_ DATE NOT NULL,
5     Delivery_Status VARCHAR(100) NOT NULL,
6     Customer_ID INT,
7     PRIMARY KEY (Delivery_ID),
8     FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
9 );
```

Products

```
PROJECT* x
Limit to 1000 rows

1 • CREATE TABLE Products
2 (
3     Product_ID INT,
4     Product_Name VARCHAR(100) NOT NULL,
5     Cost INT NOT NULL,
6     Manufacture_Date DATE NOT NULL,
7     Category_ID INT,
8     Seller_ID INT,
9     PRIMARY KEY (Product_ID),
10    FOREIGN KEY (Category_ID) REFERENCES Categories(Category_ID),
11    FOREIGN KEY (Seller_ID) REFERENCES Seller(Seller_ID)
12 );
```

Shopping_Order

```
PROJECT* x
Limit to 1000 rows

1 • CREATE TABLE Shopping_Order
2 (
3     Order_ID INT,
4     Date_ DATE NOT NULL,
5     Discount INT,
6     Quantity INT NOT NULL,
7     Total_Price INT NOT NULL,
8     Customer_ID INT,
9     PRIMARY KEY (Order_ID),
10    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
11 );
```

Transaction_Reports

```
PROJECT* x
Limit to 1000 rows

1 • CREATE TABLE Transaction_Reports
2 (
3     Report_ID INT,
4     Payment_ID INT,
5     Product_ID INT,
6     Customer_ID INT,
7     Order_ID INT,
8     PRIMARY KEY (Report_ID),
9     FOREIGN KEY (Payment_ID) REFERENCES Payment(Payment_ID),
10    FOREIGN KEY (Product_ID) REFERENCES Products(Product_ID),
11    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID),
12    FOREIGN KEY (Order_ID) REFERENCES Shopping_Order(Order_ID)
13 );
14
```

Snapshot of creation of table:

#	Time	Action	Message	Duration / Fetch
✓ 1	15:08:10	CREATE DATABASE dbms_project	1 row(s) affected	0.016 sec
✓ 2	15:08:15	USE dbms_project	0 row(s) affected	0.000 sec
✓ 3	15:08:23	CREATE TABLE Categories (Category_ID INT, Category_Name VARCHAR(100), Category_Type VARCHAR(...	0 row(s) affected	0.110 sec
✓ 4	15:08:29	CREATE TABLE Seller (Seller_ID INT, First_Name VARCHAR(100) NOT NULL, Last_Name VARCHAR(100)...	0 row(s) affected	0.094 sec
✓ 5	15:08:29	CREATE TABLE Payment (Payment_ID INT, Date_date NOT NULL, Amount INT NOT NULL, Payment_St...	0 row(s) affected	0.109 sec
✓ 6	15:08:36	CREATE TABLE Customer (Customer_ID INT, First_Name VARCHAR(100) NOT NULL, Last_Name VARCH...	0 row(s) affected	0.094 sec
✓ 7	15:08:36	CREATE TABLE Deliveries (Delivery_ID INT, Date_DATE NOT NULL, Delivery_Status VARCHAR(100) NO...	0 row(s) affected	0.125 sec
✓ 8	15:08:37	CREATE TABLE Products (Product_ID INT, Product_Name VARCHAR(100) NOT NULL, Cost INT NOT NU...	0 row(s) affected	0.109 sec
✓ 9	15:08:37	CREATE TABLE Shopping_Order (Order_ID INT, Date_DATE NOT NULL, Discount INT, Quantity INT NO...	0 row(s) affected	0.063 sec
✓ 10	15:08:37	CREATE TABLE Transaction_Reports (Report_ID INT, Payment_ID INT, Product_ID INT, Customer_ID IN...	0 row(s) affected	0.125 sec

Values Insertion

Categories:



```
PROJECT* x
Limit to 1000 rows
1 • INSERT INTO Categories VALUES ( 1001, 'Electronics', 'Mobiles' );
2 • INSERT INTO Categories VALUES ( 1002, 'Fashion', 'Shirts' );
3 • INSERT INTO Categories VALUES ( 1003, 'Furniture', 'Beds' );
4 • INSERT INTO Categories VALUES ( 1004, 'Sports', 'Bats' );
5 • INSERT INTO Categories VALUES ( 1005, 'Electronics', 'Laptops' );
6 • INSERT INTO Categories VALUES ( 1006, 'Fashion', 'Shoes' );
7 • INSERT INTO Categories VALUES ( 1007, 'Grocery', 'Soaps' );
8 • INSERT INTO Categories VALUES ( 1008, 'Electronics', 'Headphones' );
9 • INSERT INTO Categories VALUES ( 1009, 'Fashion', 'Bags' );
10 • INSERT INTO Categories VALUES ( 1010, 'Sports', 'Rackets' );
```

```
INSERT INTO Categories VALUES ( 1001, 'Electronics', 'Mobiles' );
INSERT INTO Categories VALUES ( 1002, 'Fashion', 'Shirts' );
INSERT INTO Categories VALUES ( 1003, 'Furniture', 'Beds' );
INSERT INTO Categories VALUES ( 1004, 'Sports', 'Bats' );
INSERT INTO Categories VALUES ( 1005, 'Electronics', 'Laptops' );
INSERT INTO Categories VALUES ( 1006, 'Fashion', 'Shoes' );
INSERT INTO Categories VALUES ( 1007, 'Grocery', 'Soaps' );
INSERT INTO Categories VALUES ( 1008, 'Electronics', 'Headphones' );
INSERT INTO Categories VALUES ( 1009, 'Fashion', 'Bags' );
INSERT INTO Categories VALUES ( 1010, 'Sports', 'Rackets' );
```

Seller:



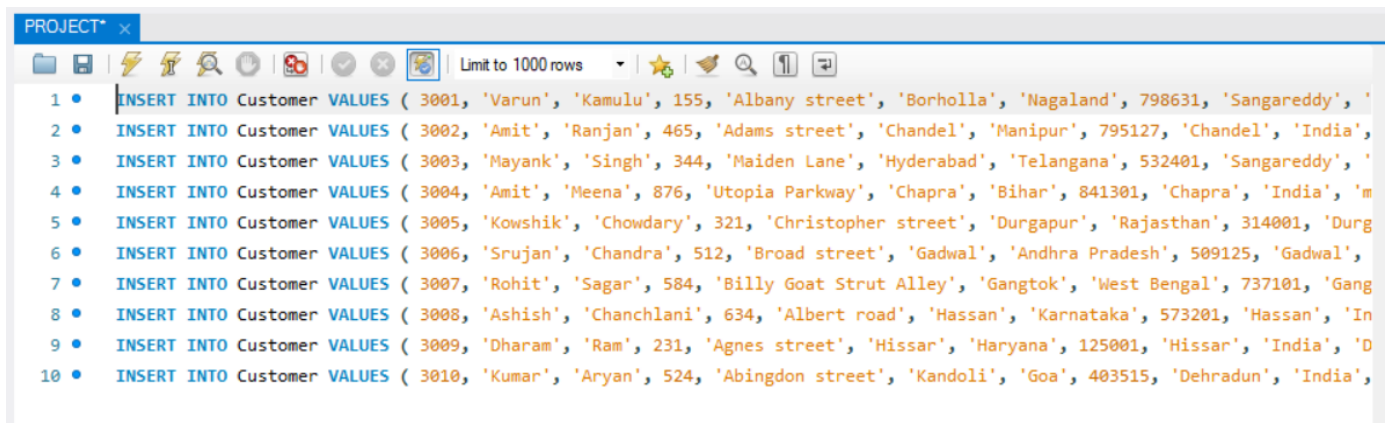
```
PROJECT* x
Limit to 1000 rows
1 • INSERT INTO Seller VALUES ( 2001, 'Akash', 'Yadav', 'Samsung', 'Hyderabad', 'Telangana', 'India', 532401);
2 • INSERT INTO Seller VALUES ( 2002, 'Animesh', 'Singh', 'Fasttrack', 'Surat', 'Gujarat', 'India', 335009);
3 • INSERT INTO Seller VALUES ( 2003, 'Marc', 'Spectre', 'HP', 'Chicago', 'Illinois', 'North America', 60007);
4 • INSERT INTO Seller VALUES ( 2004, 'Paladugu', 'Pruthvi', 'Nike', 'Hyderabad', 'Telangana', 'India', 532404);
5 • INSERT INTO Seller VALUES ( 2005, 'Yogi', 'Nayak', 'Kroger co', 'New Delhi', 'Delhi', 'India', 110001);
6 • INSERT INTO Seller VALUES ( 2006, 'Rallapalle', 'Kumar', 'Fareway', 'Arakkonam', 'Tamil Nadu', 'India', 631001);
7 • INSERT INTO Seller VALUES ( 2007, 'Rithvik', 'Muda', 'Ralph Lauren', 'Bangalore City', 'Karnataka', 'India', 560002);
8 • INSERT INTO Seller VALUES ( 2008, 'Rakshith', 'Ram', 'Hermes', 'Aurangabad', 'Bihar', 'India', 824101);
9 • INSERT INTO Seller VALUES ( 2009, 'Steven', 'Grant', 'Pottery Barn', 'Arakkonam', 'Tamil Nadu', 'India', 631001);
10 • INSERT INTO Seller VALUES ( 2010, 'Jake', 'Lockley', 'Ethan Allen', 'Aurangabad', 'Bihar', 'India', 824101);
```

```

INSERT INTO Seller VALUES ( 2001, 'Akash', 'Yadav', 'Samsung', 'Hyderabad', 'Telangana', 'India', 532401);
INSERT INTO Seller VALUES ( 2002, 'Animesh', 'Singh', 'Fasttrack', 'Surat', 'Gujarat', 'India', 335009);
INSERT INTO Seller VALUES ( 2003, 'Marc', 'Spectre', 'HP', 'Chicago', 'Illinois', 'North America', 60007);
INSERT INTO Seller VALUES ( 2004, 'Paladugu', 'Pruthvi', 'Nike', 'Hyderabad', 'Telangana', 'India', 532404);
INSERT INTO Seller VALUES ( 2005, 'Yogi', 'Nayak', 'Kroger co', 'New Delhi', 'Delhi', 'India', 110001);
INSERT INTO Seller VALUES ( 2006, 'Rallapalle', 'Kumar', 'Fareway', 'Arakkonam', 'Tamil Nadu', 'India', 631001);
INSERT INTO Seller VALUES ( 2007, 'Rithvik', 'Muda', 'Ralph Lauren', 'Bangalore City', 'Karnataka', 'India', 560002);
INSERT INTO Seller VALUES ( 2008, 'Rakshith', 'Ram', 'Hermes', 'Aurangabad', 'Bihar', 'India', 824101);
INSERT INTO Seller VALUES ( 2009, 'Steven', 'Grant', 'Pottery Barn', 'Arakkonam', 'Tamil Nadu', 'India', 631001);
INSERT INTO Seller VALUES ( 2010, 'Jake', 'Lockley', 'Ethan Allen', 'Aurangabad', 'Bihar', 'India', 824101);

```

Customer:



```

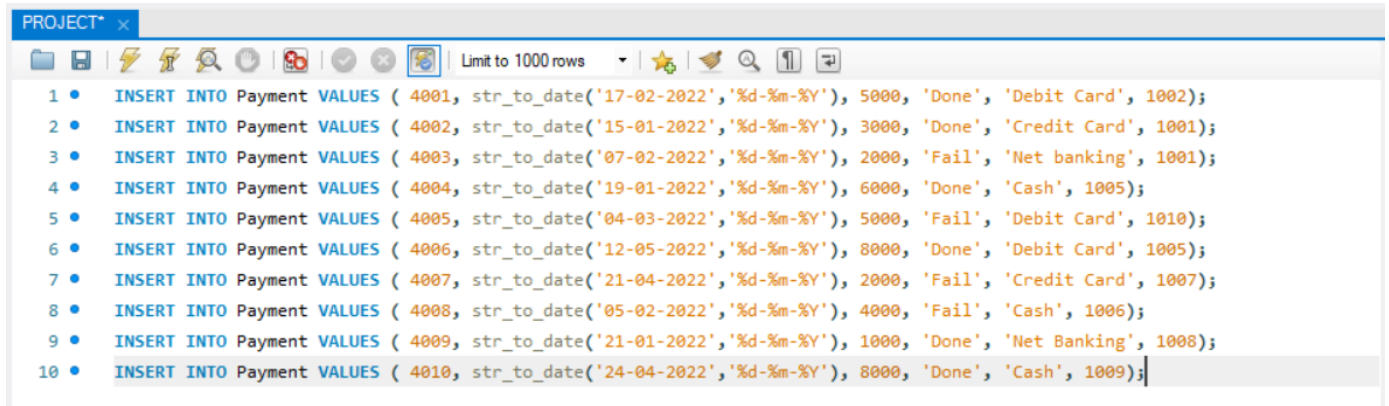
INSERT INTO Customer VALUES ( 3001, 'Varun', 'Kamulu', 155, 'Albany street', 'Borholla', 'Nagaland', 798631, 'Sangareddy', 'India', 'KamuluVarun123@gmail.com', 2313425);
INSERT INTO Customer VALUES ( 3002, 'Amit', 'Ranjan', 465, 'Adams street', 'Chandel', 'Manipur', 795127, 'Chandel', 'India', 'amitrانjan817@gmail.com', 5762347);
INSERT INTO Customer VALUES ( 3003, 'Mayank', 'Singh', 344, 'Maiden Lane', 'Hyderabad', 'Telangana', 532401, 'Sangareddy', 'India', 'mayankbisth111@gmail.com', 6754698);
INSERT INTO Customer VALUES ( 3004, 'Amit', 'Meena', 876, 'Utopia Parkway', 'Chapra', 'Bihar', 841301, 'Chapra', 'India', 'meenaamit111@gmail.com', 9145289);
INSERT INTO Customer VALUES ( 3005, 'Kowshik', 'Chowdary', 321, 'Christopher street', 'Durgapur', 'Rajasthan', 314001, 'Durgapur', 'India', 'Kowshik123@gmail.com', 8265294);
INSERT INTO Customer VALUES ( 3006, 'Srujan', 'Chandra', 512, 'Broad street', 'Gadwal', 'Andhra Pradesh', 509125, 'Gadwal', 'India', 'Srujananchandra153@gmail.com', 2275624);
INSERT INTO Customer VALUES ( 3007, 'Rohit', 'Sagar', 584, 'Billy Goat Strut Alley', 'Gangtok', 'West Bengal', 737101, 'Gangtok', 'India', 'RohitSagar584@gmail.com', 737101);

```


'West Bengal', 737101, 'Gangtok', 'India', 'SagarRohith1827@gmail.com', 6849350);

```
INSERT INTO Customer VALUES ( 3008, 'Ashish', 'Chanchlani', 634, 'Albert road', 'Hassan',  
'Karnataka', 573201, 'Hassan', 'India', 'ChanchalaniAshish@gmail.com', 7565467);  
INSERT INTO Customer VALUES ( 3009, 'Dharam', 'Ram', 231, 'Agnes street', 'Hissar', 'Haryana',  
125001, 'Hissar', 'India', 'DharamRammech7@gmail.com', 6324156);  
INSERT INTO Customer VALUES ( 3010, 'Kumar', 'Aryan', 524, 'Abingdon street', 'Kandoli', 'Goa',  
403515, 'Dehradun', 'India', 'KumarAryanbio99@gmail.com', 7656471);
```


Payment:

A screenshot of a SQL IDE window titled 'PROJECT'. The window shows a list of 10 SQL statements for inserting data into a 'Payment' table. The statements are numbered 1 through 10. Each statement follows the format: INSERT INTO Payment VALUES (payment_id, str_to_date('date', '%d-%m-%Y'), amount, status, card_type, id). The data includes various dates, amounts, statuses (Done, Fail), and card types (Debit Card, Credit Card, Cash, Net banking).

```
1 • INSERT INTO Payment VALUES ( 4001, str_to_date('17-02-2022', '%d-%m-%Y'), 5000, 'Done', 'Debit Card', 1002);  
2 • INSERT INTO Payment VALUES ( 4002, str_to_date('15-01-2022', '%d-%m-%Y'), 3000, 'Done', 'Credit Card', 1001);  
3 • INSERT INTO Payment VALUES ( 4003, str_to_date('07-02-2022', '%d-%m-%Y'), 2000, 'Fail', 'Net banking', 1001);  
4 • INSERT INTO Payment VALUES ( 4004, str_to_date('19-01-2022', '%d-%m-%Y'), 6000, 'Done', 'Cash', 1005);  
5 • INSERT INTO Payment VALUES ( 4005, str_to_date('04-03-2022', '%d-%m-%Y'), 5000, 'Fail', 'Debit Card', 1010);  
6 • INSERT INTO Payment VALUES ( 4006, str_to_date('12-05-2022', '%d-%m-%Y'), 8000, 'Done', 'Debit Card', 1005);  
7 • INSERT INTO Payment VALUES ( 4007, str_to_date('21-04-2022', '%d-%m-%Y'), 2000, 'Fail', 'Credit Card', 1007);  
8 • INSERT INTO Payment VALUES ( 4008, str_to_date('05-02-2022', '%d-%m-%Y'), 4000, 'Fail', 'Cash', 1006);  
9 • INSERT INTO Payment VALUES ( 4009, str_to_date('21-01-2022', '%d-%m-%Y'), 1000, 'Done', 'Net Banking', 1008);  
10 • INSERT INTO Payment VALUES ( 4010, str_to_date('24-04-2022', '%d-%m-%Y'), 8000, 'Done', 'Cash', 1009);
```

```
INSERT INTO Payment VALUES ( 4001, str_to_date('17-02-2022', '%d-%m-%Y'), 5000, 'Done',  
'Debit Card', 1002);  
INSERT INTO Payment VALUES ( 4002, str_to_date('15-01-2022', '%d-%m-%Y'), 3000, 'Done',  
'Credit Card', 1001);  
INSERT INTO Payment VALUES ( 4003, str_to_date('07-02-2022', '%d-%m-%Y'), 2000, 'Fail', 'Net  
banking', 1001);  
INSERT INTO Payment VALUES ( 4004, str_to_date('19-01-2022', '%d-%m-%Y'), 6000, 'Done',  
'Cash', 1005);  
INSERT INTO Payment VALUES ( 4005, str_to_date('04-03-2022', '%d-%m-%Y'), 5000, 'Fail',  
'Debit Card', 1010);  
INSERT INTO Payment VALUES ( 4006, str_to_date('12-05-2022', '%d-%m-%Y'), 8000, 'Done',  
'Debit Card', 1005);  
INSERT INTO Payment VALUES ( 4007, str_to_date('21-04-2022', '%d-%m-%Y'), 2000, 'Fail',  
'Credit Card', 1007);  
INSERT INTO Payment VALUES ( 4008, str_to_date('05-02-2022', '%d-%m-%Y'), 4000, 'Fail',  
'Cash', 1006);  
INSERT INTO Payment VALUES ( 4009, str_to_date('21-01-2022', '%d-%m-%Y'), 1000, 'Done',  
'Net Banking', 1008);  
INSERT INTO Payment VALUES ( 4010, str_to_date('24-04-2022', '%d-%m-%Y'), 8000, 'Done',  
'Cash', 1009);
```

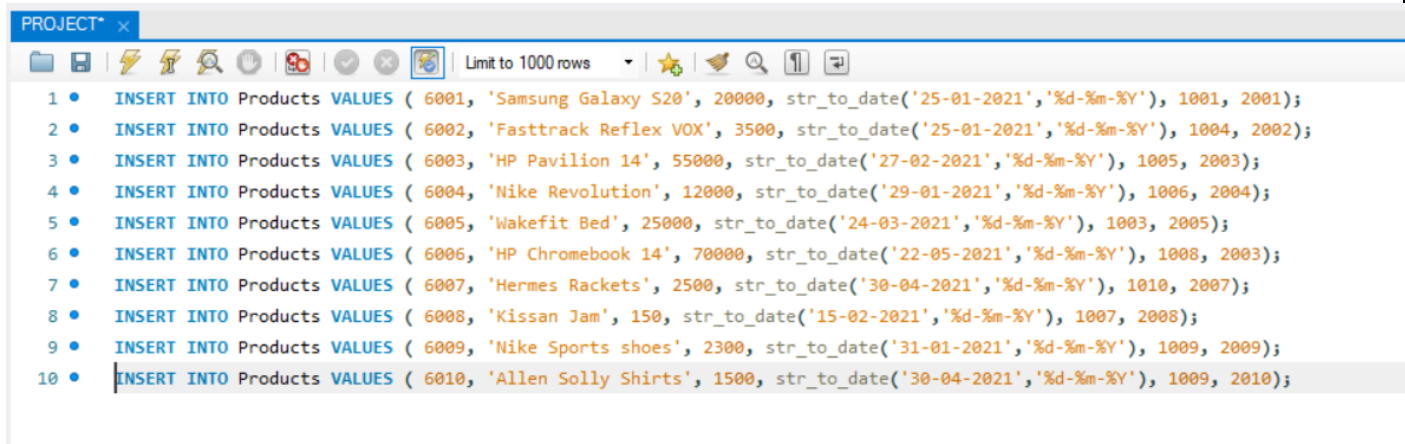
Deliveries:



```
PROJECT* x
Limit to 1000 rows
1 • INSERT INTO Deliveries VALUES ( 5001, str_to_date('27-02-2022','%d-%m-%Y'), 'Done', 3001);
2 • INSERT INTO Deliveries VALUES ( 5002, str_to_date('25-01-2022','%d-%m-%Y'), 'Done', 3002);
3 • INSERT INTO Deliveries VALUES ( 5003, str_to_date('27-02-2022','%d-%m-%Y'), 'Fail', 3003);
4 • INSERT INTO Deliveries VALUES ( 5004, str_to_date('29-01-2022','%d-%m-%Y'), 'Done', 3004);
5 • INSERT INTO Deliveries VALUES ( 5005, str_to_date('24-03-2022','%d-%m-%Y'), 'Fail', 3005);
6 • INSERT INTO Deliveries VALUES ( 5006, str_to_date('22-05-2022','%d-%m-%Y'), 'Done', 3006);
7 • INSERT INTO Deliveries VALUES ( 5007, str_to_date('30-04-2022','%d-%m-%Y'), 'Fail', 3007);
8 • INSERT INTO Deliveries VALUES ( 5008, str_to_date('15-02-2022','%d-%m-%Y'), 'Fail', 3008);
9 • INSERT INTO Deliveries VALUES ( 5009, str_to_date('31-01-2022','%d-%m-%Y'), 'Done', 3009);
10 • INSERT INTO Deliveries VALUES ( 5010, str_to_date('30-04-2022','%d-%m-%Y'), 'Done', 3010);
```

```
INSERT INTO Deliveries VALUES ( 5001, str_to_date('27-02-2022','%d-%m-%Y'), 'Done', 3001);
INSERT INTO Deliveries VALUES ( 5002, str_to_date('25-01-2022','%d-%m-%Y'), 'Done', 3002);
INSERT INTO Deliveries VALUES ( 5003, str_to_date('27-02-2022','%d-%m-%Y'), 'Fail', 3003);
INSERT INTO Deliveries VALUES ( 5004, str_to_date('29-01-2022','%d-%m-%Y'), 'Done', 3004);
INSERT INTO Deliveries VALUES ( 5005, str_to_date('24-03-2022','%d-%m-%Y'), 'Fail', 3005);
INSERT INTO Deliveries VALUES ( 5006, str_to_date('22-05-2022','%d-%m-%Y'), 'Done', 3006);
INSERT INTO Deliveries VALUES ( 5007, str_to_date('30-04-2022','%d-%m-%Y'), 'Fail', 3007);
INSERT INTO Deliveries VALUES ( 5008, str_to_date('15-02-2022','%d-%m-%Y'), 'Fail', 3008);
INSERT INTO Deliveries VALUES ( 5009, str_to_date('31-01-2022','%d-%m-%Y'), 'Done', 3009);
INSERT INTO Deliveries VALUES ( 5010, str_to_date('30-04-2022','%d-%m-%Y'), 'Done', 3010);
```

Products:

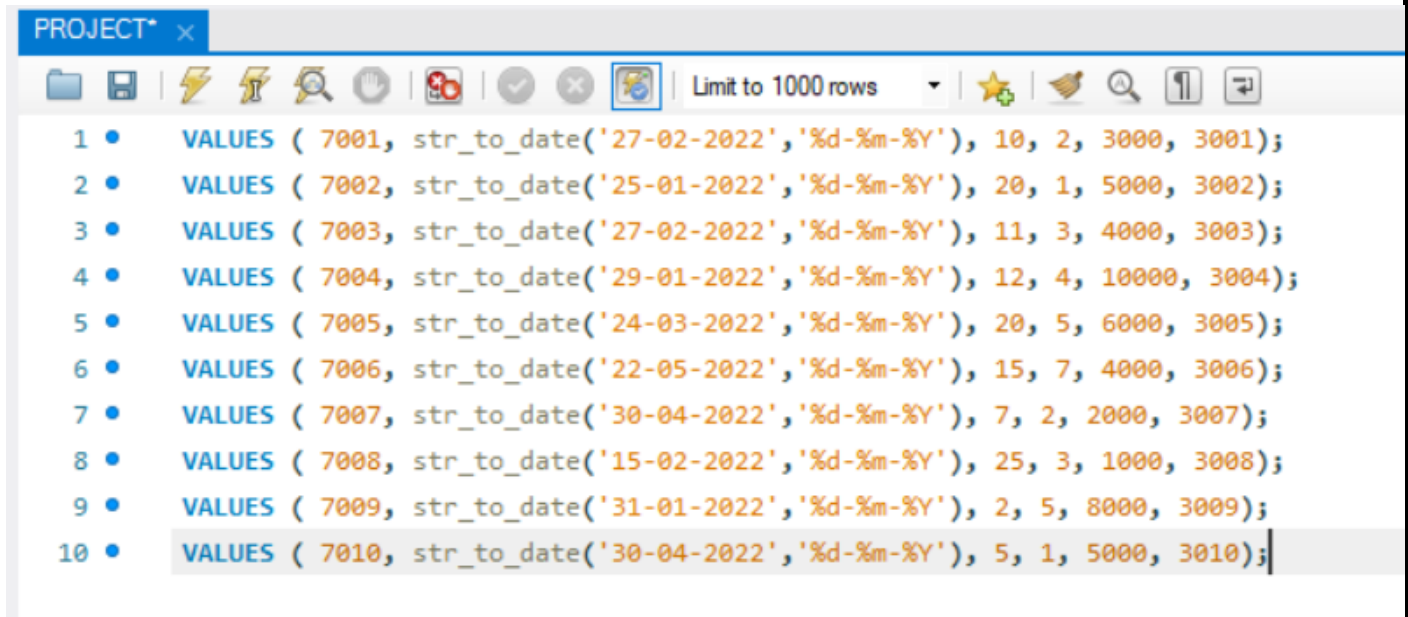


The screenshot shows a SQL IDE window titled 'PROJECT'. The toolbar includes icons for file operations, execution, and search. A dropdown menu indicates 'Limit to 1000 rows'. The SQL editor contains 10 numbered lines, each starting with a blue bullet point and an 'INSERT INTO' statement. The statements insert data into a 'Products' table with columns for product ID, name, price, and two dates. The products listed are: Samsung Galaxy S20, Fasttrack Reflex VOX, HP Pavilion 14, Nike Revolution, Wakefit Bed, HP Chromebook 14, Hermes Rackets, Kissan Jam, Nike Sports shoes, and Allen Solly Shirts.

```
1 • INSERT INTO Products VALUES ( 6001, 'Samsung Galaxy S20', 20000, str_to_date('25-01-2021','%d-%m-%Y'), 1001, 2001);
2 • INSERT INTO Products VALUES ( 6002, 'Fasttrack Reflex VOX', 3500, str_to_date('25-01-2021','%d-%m-%Y'), 1004, 2002);
3 • INSERT INTO Products VALUES ( 6003, 'HP Pavilion 14', 55000, str_to_date('27-02-2021','%d-%m-%Y'), 1005, 2003);
4 • INSERT INTO Products VALUES ( 6004, 'Nike Revolution', 12000, str_to_date('29-01-2021','%d-%m-%Y'), 1006, 2004);
5 • INSERT INTO Products VALUES ( 6005, 'Wakefit Bed', 25000, str_to_date('24-03-2021','%d-%m-%Y'), 1003, 2005);
6 • INSERT INTO Products VALUES ( 6006, 'HP Chromebook 14', 70000, str_to_date('22-05-2021','%d-%m-%Y'), 1008, 2003);
7 • INSERT INTO Products VALUES ( 6007, 'Hermes Rackets', 2500, str_to_date('30-04-2021','%d-%m-%Y'), 1010, 2007);
8 • INSERT INTO Products VALUES ( 6008, 'Kissan Jam', 150, str_to_date('15-02-2021','%d-%m-%Y'), 1007, 2008);
9 • INSERT INTO Products VALUES ( 6009, 'Nike Sports shoes', 2300, str_to_date('31-01-2021','%d-%m-%Y'), 1009, 2009);
10 • INSERT INTO Products VALUES ( 6010, 'Allen Solly Shirts', 1500, str_to_date('30-04-2021','%d-%m-%Y'), 1009, 2010);
```

```
INSERT INTO Products VALUES ( 6001, 'Samsung Galaxy S20', 20000,
str_to_date('25-01-2021','%d-%m-%Y'), 1001, 2001);
INSERT INTO Products VALUES ( 6002, 'Fasttrack Reflex VOX', 3500,
str_to_date('25-01-2021','%d-%m-%Y'), 1004, 2002);
INSERT INTO Products VALUES ( 6003, 'HP Pavilion 14', 55000,
str_to_date('27-02-2021','%d-%m-%Y'), 1005, 2003);
INSERT INTO Products VALUES ( 6004, 'Nike Revolution', 12000,
str_to_date('29-01-2021','%d-%m-%Y'), 1006, 2004);
INSERT INTO Products VALUES ( 6005, 'Wakefit Bed', 25000,
str_to_date('24-03-2021','%d-%m-%Y'), 1003, 2005);
INSERT INTO Products VALUES ( 6006, 'HP Chromebook 14', 70000,
str_to_date('22-05-2021','%d-%m-%Y'), 1008, 2003);
INSERT INTO Products VALUES ( 6007, 'Hermes Rackets', 2500,
str_to_date('30-04-2021','%d-%m-%Y'), 1010, 2007);
INSERT INTO Products VALUES ( 6008, 'Kissan Jam', 150,
str_to_date('15-02-2021','%d-%m-%Y'), 1007, 2008);
INSERT INTO Products VALUES ( 6009, 'Nike Sports shoes', 2300,
str_to_date('31-01-2021','%d-%m-%Y'), 1009, 2009);
INSERT INTO Products VALUES ( 6010, 'Allen Solly Shirts', 1500,
str_to_date('30-04-2021','%d-%m-%Y'), 1009, 2010);
```

Shopping_Order:

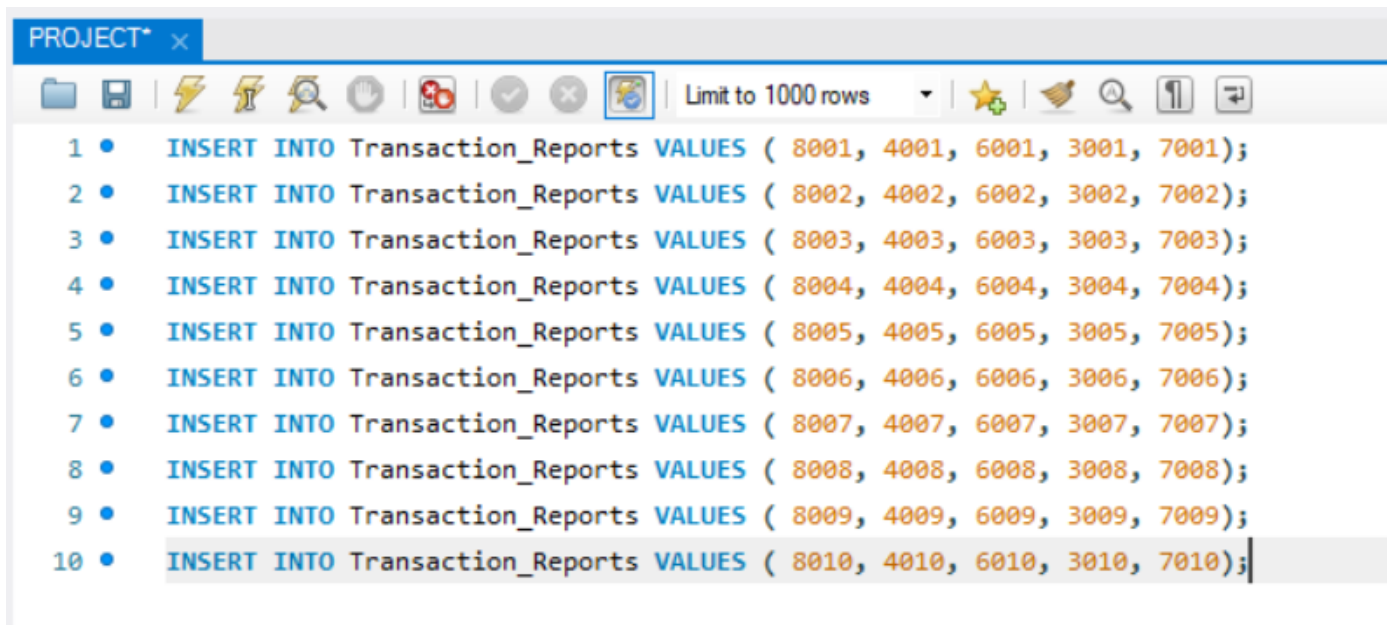


The screenshot shows a SQL IDE window titled 'PROJECT*'. The toolbar includes icons for file operations, execution, and search. A dropdown menu shows 'Limit to 1000 rows'. The main area displays 10 rows of data for the 'Shopping_Order' table, each starting with a blue bullet point and the word 'VALUES' in blue. The data is as follows:

Row	Order ID	Date	Quantity	Price	Product ID
1	7001	2022-02-27	10	3000	3001
2	7002	2022-01-25	20	5000	3002
3	7003	2022-02-27	11	4000	3003
4	7004	2022-01-29	12	10000	3004
5	7005	2022-03-24	20	6000	3005
6	7006	2022-05-22	15	4000	3006
7	7007	2022-04-30	7	2000	3007
8	7008	2022-02-15	25	1000	3008
9	7009	2022-01-31	2	8000	3009
10	7010	2022-04-30	5	5000	3010

```
INSERT INTO Shopping_order VALUES ( 7001, str_to_date('27-02-2022','%d-%m-%Y'), 10, 2, 3000, 3001);
INSERT INTO Shopping_order VALUES ( 7002, str_to_date('25-01-2022','%d-%m-%Y'), 20, 1, 5000, 3002);
INSERT INTO Shopping_order VALUES ( 7003, str_to_date('27-02-2022','%d-%m-%Y'), 11, 3, 4000, 3003);
INSERT INTO Shopping_order VALUES ( 7004, str_to_date('29-01-2022','%d-%m-%Y'), 12, 4, 10000, 3004);
INSERT INTO Shopping_order VALUES ( 7005, str_to_date('24-03-2022','%d-%m-%Y'), 20, 5, 6000, 3005);
INSERT INTO Shopping_order VALUES ( 7006, str_to_date('22-05-2022','%d-%m-%Y'), 15, 7, 4000, 3006);
INSERT INTO Shopping_order VALUES ( 7007, str_to_date('30-04-2022','%d-%m-%Y'), 7, 2, 2000, 3007);
INSERT INTO Shopping_order VALUES ( 7008, str_to_date('15-02-2022','%d-%m-%Y'), 25, 3, 1000, 3008);
INSERT INTO Shopping_order VALUES ( 7009, str_to_date('31-01-2022','%d-%m-%Y'), 2, 5, 8000, 3009);
INSERT INTO Shopping_order VALUES ( 7010, str_to_date('30-04-2022','%d-%m-%Y'), 5, 1, 5000, 3010);
```

Transaction_Reports:



The screenshot shows a SQL IDE window titled 'PROJECT' with a toolbar and a list of 10 SQL statements. The statements are numbered 1 through 10, each preceded by a blue bullet point. The statements are: 1. INSERT INTO Transaction_Reports VALUES (8001, 4001, 6001, 3001, 7001); 2. INSERT INTO Transaction_Reports VALUES (8002, 4002, 6002, 3002, 7002); 3. INSERT INTO Transaction_Reports VALUES (8003, 4003, 6003, 3003, 7003); 4. INSERT INTO Transaction_Reports VALUES (8004, 4004, 6004, 3004, 7004); 5. INSERT INTO Transaction_Reports VALUES (8005, 4005, 6005, 3005, 7005); 6. INSERT INTO Transaction_Reports VALUES (8006, 4006, 6006, 3006, 7006); 7. INSERT INTO Transaction_Reports VALUES (8007, 4007, 6007, 3007, 7007); 8. INSERT INTO Transaction_Reports VALUES (8008, 4008, 6008, 3008, 7008); 9. INSERT INTO Transaction_Reports VALUES (8009, 4009, 6009, 3009, 7009); 10. INSERT INTO Transaction_Reports VALUES (8010, 4010, 6010, 3010, 7010); The toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown.

```
1 • INSERT INTO Transaction_Reports VALUES ( 8001, 4001, 6001, 3001, 7001);
2 • INSERT INTO Transaction_Reports VALUES ( 8002, 4002, 6002, 3002, 7002);
3 • INSERT INTO Transaction_Reports VALUES ( 8003, 4003, 6003, 3003, 7003);
4 • INSERT INTO Transaction_Reports VALUES ( 8004, 4004, 6004, 3004, 7004);
5 • INSERT INTO Transaction_Reports VALUES ( 8005, 4005, 6005, 3005, 7005);
6 • INSERT INTO Transaction_Reports VALUES ( 8006, 4006, 6006, 3006, 7006);
7 • INSERT INTO Transaction_Reports VALUES ( 8007, 4007, 6007, 3007, 7007);
8 • INSERT INTO Transaction_Reports VALUES ( 8008, 4008, 6008, 3008, 7008);
9 • INSERT INTO Transaction_Reports VALUES ( 8009, 4009, 6009, 3009, 7009);
10 • INSERT INTO Transaction_Reports VALUES ( 8010, 4010, 6010, 3010, 7010);
```

```
INSERT INTO Transaction_Reports VALUES ( 8001, 4001, 6001, 3001, 7001);
INSERT INTO Transaction_Reports VALUES ( 8002, 4002, 6002, 3002, 7002);
INSERT INTO Transaction_Reports VALUES ( 8003, 4003, 6003, 3003, 7003);
INSERT INTO Transaction_Reports VALUES ( 8004, 4004, 6004, 3004, 7004);
INSERT INTO Transaction_Reports VALUES ( 8005, 4005, 6005, 3005, 7005);
INSERT INTO Transaction_Reports VALUES ( 8006, 4006, 6006, 3006, 7006);
INSERT INTO Transaction_Reports VALUES ( 8007, 4007, 6007, 3007, 7007);
INSERT INTO Transaction_Reports VALUES ( 8008, 4008, 6008, 3008, 7008);
INSERT INTO Transaction_Reports VALUES ( 8009, 4009, 6009, 3009, 7009);
INSERT INTO Transaction_Reports VALUES ( 8010, 4010, 6010, 3010, 7010);
```

Tables:

Categories:

SELECT * FROM Categories;

Result Grid	Filter Rows:	Ex
Category_ID	Category_Name	Category_Type
1001	Electronics	Mobiles
1002	Fashion	Shirts
1003	Furniture	Beds
1004	Sports	Bats
1005	Electronics	Laptops
1006	Fashion	Shoes
1007	Grocery	Soaps
1008	Electronics	Headphones
1009	Fashion	Bags
1010	Sports	Rackets
NULL	NULL	NULL

Seller:

SELECT * FROM Seller;

Result Grid

Filter Rows:

Edit:


Export/Import:

Wrap Cell Content:

	Seller_ID	First_Name	Last_Name	Company_Name	City	State	Country	Pin_code
▶	2001	Akash	Yadav	Samsung	Hyderabad	Telangana	India	532401
	2002	Animesh	Singh	Fasttrack	Surat	Gujarat	India	335009
	2003	Marc	Spectre	HP	Chicago	Illinois	North America	60007
	2004	Paladugu	Pruthvi	Nike	Hyderabad	Telangana	India	532404
	2005	Yogi	Nayak	Kroger co	New Delhi	Delhi	India	110001
	2006	Rallapalle	Kumar	Fareway	Arakkonam	Tamil Nadu	India	631001
	2007	Rithvik	Muda	Ralph Lauren	Bangalore City	Karnataka	India	560002
	2008	Rakshith	Ram	Hermes	Aurangabad	Bihar	India	824101
	2009	Steven	Grant	Pottery Barn	Arakkonam	Tamil Nadu	India	631001
	2010	Jake	Lockley	Ethan Allen	Aurangabad	Bihar	India	824101
✱	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Customer:

SELECT * FROM Payment;

Result Grid		Filter Rows: <input type="text"/>		Edit:  		Export/Import:  		Wrap Cell Content: <input type="text" value="A"/>				
	Customer_ID	First_Name	Last_Name	H_No	Street	City	State	Pin_code	District	Country	Email	Phone_No
▶	3001	Varun	Kamulu	155	Albany street	Borholla	Nagaland	798631	Sangareddy	India	KamuluVarun123@gmail.com	2313425
	3002	Amit	Ranjan	465	Adams street	Chandel	Manipur	795127	Chandel	India	amitrnanjan817@gmail.com	5762347
	3003	Mayank	Singh	344	Maiden Lane	Hyderabad	Telangana	532401	Sangareddy	India	mayankbisth111@gmail.com	6754698
	3004	Amit	Meena	876	Utopia Parkway	Chapra	Bihar	841301	Chapra	India	meenaamit111@gmail.com	9145289
	3005	Kowshik	Chowdary	321	Christopher street	Durgapur	Rajasthan	314001	Durgapur	India	Kowshik123@gmail.com	8265294
	3006	Srujan	Chandra	512	Broad street	Gadwal	Andhra Pradesh	509125	Gadwal	India	SrujanChandra153@gmail.com	2275624
	3007	Rohit	Sagar	584	Billy Goat Strut Alley	Gangtok	West Bengal	737101	Gangtok	India	SagarRohith1827@gmail.com	6849350
	3008	Ashish	Chanchlani	634	Albert road	Hassan	Karnataka	573201	Hassan	India	ChanchalaniAshish@gmail.com	7565467
	3009	Dharam	Ram	231	Agnes street	Hissar	Haryana	125001	Hissar	India	DharamRammeh7@gmail.com	6324156
	3010	Kumar	Aryan	524	Abingdon street	Kandoli	Goa	403515	Dehradun	India	KumarAryanbio99@gmail.com	7656471
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Payment:

SELECT * FROM Payment;

Result Grid

Filter Rows:

Edit:

Export/Import:

	Payment_ID	Date_	Amount	Payment_Status	Payment_Type	Category_ID
▶	4001	2022-02-17	5000	Done	Debit Card	1002
	4002	2022-01-15	3000	Done	Credit Card	1001
	4003	2022-02-07	2000	Fail	Net banking	1001
	4004	2022-01-19	6000	Done	Cash	1005
	4005	2022-03-04	5000	Fail	Debit Card	1010
	4006	2022-05-12	8000	Done	Debit Card	1005
	4007	2022-04-21	2000	Fail	Credit Card	1007
	4008	2022-02-05	4000	Fail	Cash	1006
	4009	2022-01-21	1000	Done	Net Banking	1008
	4010	2022-04-24	8000	Done	Cash	1009
•	NULL	NULL	NULL	NULL	NULL	NULL

Deliveries:

SELECT * FROM Deliveries;

Result Grid

Filter Rows:


Edit:




	Delivery_ID	Date_	Delivery_Status	Customer_ID
▶	5001	2022-02-27	Done	3001
	5002	2022-01-25	Done	3002
	5003	2022-02-27	Fail	3003
	5004	2022-01-29	Done	3004
	5005	2022-03-24	Fail	3005
	5006	2022-05-22	Done	3006
	5007	2022-04-30	Fail	3007
	5008	2022-02-15	Fail	3008
	5009	2022-01-31	Done	3009
	5010	2022-04-30	Done	3010
✱	NULL	NULL	NULL	NULL



Products:

SELECT * FROM Products;

Result Grid

 Filter Rows:

 Edit:  

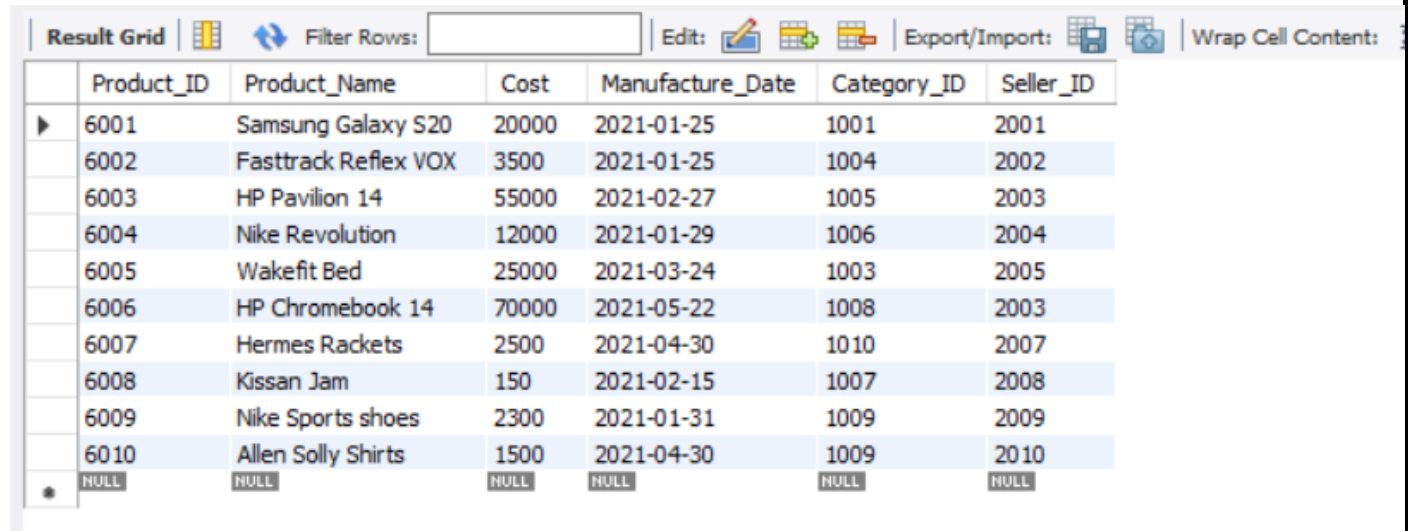
Export/Import:  

Wrap Cell Content

	Product_ID	Product_Name	Cost	Manufacture_Date	Category_ID	Seller_ID
▶	6001	Samsung Galaxy S20	20000	2021-01-25	1001	2001
	6002	Fasttrack Reflex VOX	3500	2021-01-25	1004	2002
	6003	HP Pavilion 14	55000	2021-02-27	1005	2003
	6004	Nike Revolution	12000	2021-01-29	1006	2004
	6005	Wakefit Bed	25000	2021-03-24	1003	2005
	6006	HP Chromebook 14	70000	2021-05-22	1008	2003
	6007	Hermes Rackets	2500	2021-04-30	1010	2007
	6008	Kissan Jam	150	2021-02-15	1007	2008
	6009	Nike Sports shoes	2300	2021-01-31	1009	2009
	6010	Allen Solly Shirts	1500	2021-04-30	1009	2010
•	NULL	NULL	NULL	NULL	NULL	NULL

Shopping_order:

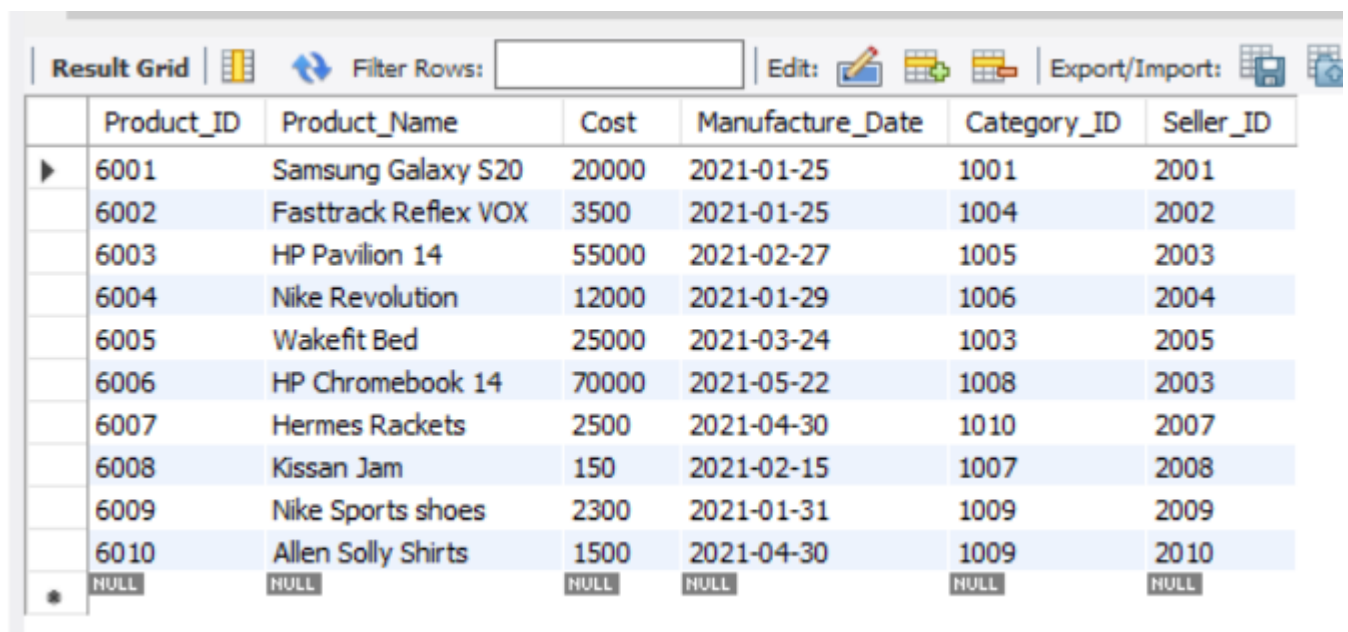
```
SELECT * FROM Shopping_order;
```



	Product_ID	Product_Name	Cost	Manufacture_Date	Category_ID	Seller_ID
▶	6001	Samsung Galaxy S20	20000	2021-01-25	1001	2001
	6002	Fasttrack Reflex VOX	3500	2021-01-25	1004	2002
	6003	HP Pavilion 14	55000	2021-02-27	1005	2003
	6004	Nike Revolution	12000	2021-01-29	1006	2004
	6005	Wakefit Bed	25000	2021-03-24	1003	2005
	6006	HP Chromebook 14	70000	2021-05-22	1008	2003
	6007	Hermes Rackets	2500	2021-04-30	1010	2007
	6008	Kissan Jam	150	2021-02-15	1007	2008
	6009	Nike Sports shoes	2300	2021-01-31	1009	2009
	6010	Allen Solly Shirts	1500	2021-04-30	1009	2010
•	NULL	NULL	NULL	NULL	NULL	NULL

Transaction_Reports:

```
SELECT * FROM Transaction_Reports;
```



	Product_ID	Product_Name	Cost	Manufacture_Date	Category_ID	Seller_ID
▶	6001	Samsung Galaxy S20	20000	2021-01-25	1001	2001
	6002	Fasttrack Reflex VOX	3500	2021-01-25	1004	2002
	6003	HP Pavilion 14	55000	2021-02-27	1005	2003
	6004	Nike Revolution	12000	2021-01-29	1006	2004
	6005	Wakefit Bed	25000	2021-03-24	1003	2005
	6006	HP Chromebook 14	70000	2021-05-22	1008	2003
	6007	Hermes Rackets	2500	2021-04-30	1010	2007
	6008	Kissan Jam	150	2021-02-15	1007	2008
	6009	Nike Sports shoes	2300	2021-01-31	1009	2009
	6010	Allen Solly Shirts	1500	2021-04-30	1009	2010
•	NULL	NULL	NULL	NULL	NULL	NULL