AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

FACULTY OF SCIENCE & TECHNOLOGY

INTRODUCTION TO DATABASE

SPRING 2023-2024

Section: N, Group-06

PROJECT ON

GROCERY STORE MANAGEMENT SYSTEM

Supervised By: SHAHNAJ PARVIN

Submitted By

NAME	ID
FARIA NOURIN	23-50269-1
BARNABAS PRENTICE	23-50274-1
AMRITA BHAWAL DISHA	23-50892-1
NAFIS HASAN	23-50976-1

Table of Contents

Introduction

A Grocery Store Management System simplifies the running of a grocery store by handling tasks like keeping track of products, staff, and customers, and handling payments. It's like a central control center for all store operations. This system makes it easy to manage inventory, track sales, see profits, and keep customers happy. A grocery shop management system is a database project that aims to store and manage the information related to a grocery shop, such as the products, customers, suppliers, orders, payments, etc. Plus, it provides helpful reports and data analysis to guide decision-making and future planning for the store. Ultimately, it helps the store owner use resources efficiently, improve customer experiences, and make smart choices to grow the business.

Case Study

Student ID 1: 23-50269-1 **Name: Faria Nourin**	**Student ID 3: 23-50892-1** **Name: Amrita Bhawal Disha**
Student ID 2: 23-50274-1 **Name: Barnabas Prentice**	**Student ID 4: 23-50976-1** **Name: Nafis Hasan**
CO2: Understand the fundamental concepts underlying database systems and gain hands-on experience with ER diagram Case study.	
PO-c2: Develop process for complex computer science and engineering problems considering cultural and societal factors.	Marks

The management system is like a computer program for running a grocery store. It helps the owner keep track of everything in the store. This includes things like what products are available, who the customers are, orders coming in, and managing payments. The system can also provide a user-friendly interface and support various database management systems and programming languages. The system is made to be easy for anyone to use, with buttons and menus that make sense.

The project can have the following features:

- ✓ ***Managing Products*:** The project can keep track of product details like name, category, price, quantity, and expiry date. With this system, the shop owner can easily add new products to the database, update existing product details like price or quantity, remove products that are no longer in stock, and search for specific products based on their attributes.
- ✓ ***Customer Handling:*** Store customer information such as name, address, phone number, email, and loyalty points. Shop owners can use this feature to add new customers to their records, update existing customer information, remove outdated entries, and search for specific customers when needed.
- ✓ ***Order Tracking:*** The project can record order details like order number, date, time, customer info, products ordered, quantity, total amount, payment method, and order status. Shop owner can efficiently add new orders as they come in, update order statuses as they progress through the fulfillment process and remove canceled or completed orders from the system.
- ✓ ***Payment Records:*** The project can store payment information like payment number, date, time, order details, amount, and payment method. Shop owners can use this feature to add new payment records, update existing payment information if necessary, remove duplicate or incorrect entries, and search for specific payments when required.

ER Diagram


P_Category
P_ID
P_Name
P_Availability
Product
P_Quantity
P_Price
M
of any order
1
Order_ID
Order_Amount
Order_Address
Orders
Order_Date
M
1
can place
1
Customer
C_Name
C_ID
C_Address
C_Email
1
has
1
Customer_ Contact
Customer_ID
C_Phone
Pay_ID
O_Serial
O_ID
Order_Customer
C_ID
1
P_ID
have
1
1
through
1
Pay_ID
Payment
Pay_Amount
M
1
Pay_Date
pays
1
contains
1
Payment_ Details
Pay_Serial
Pay_Method
Payment_ID

Normalization

❖ First Normal Form (1NF):

For a table to be in First Normal Form, it should follow the following 4 rules:

• It should only have single (atomic) valued attributes/columns.

• Values stored in a column should be of the same domain.

• All the columns in a table should have unique names.

• The order in which data in which data is stored, does not matter.

❖ Second Normal Form (2NF):

For a table to be in the Second Normal Form:

• It should be in the First Normal form.

• It should not have Partial Dependency.

❖ Third Normal Form (3NF):

A table is said to be in the Third Normal Form when,

• It is in the Second Normal form.

• It doesn't have Transitive Dependency.

Product Table

Product(P_ID, P_Category, P_Name, P_Quantity, P_Price, P_Availability)

It's in 1NF as it has a primary key (P_ID) and all other attributes contain atomic values. It's in 2NF and 3NF as there are no partial or transitive dependencies

Orders Table

Orders(Order_ID, Order_Address, Order_Date, Order_Amount)

It's in 1NF (primary key is Order_ID, and all other attributes contain atomic values), 2NF, and 3NF (no partial or transitive dependencies).

Customer Table

Customer(C_ID, C_Name, C_Address, C_Email)

It's in 1NF (primary key is C_ID, and all other attributes contain atomic values), 2NF, and 3NF (no partial or transitive dependencies).

Payment Table

Payment(Pay_ID, Pay_Amount, Pay_Date)

It's in 1NF (primary key is Pay_ID, and all other attributes contain atomic values), 2NF, and 3NF (no partial or transitive dependencies).

Payment_Details Table

Payment_Details(Payment_ID, Pay_Serial, Pay_Method)

It's in 1NF (primary key is Pay_Serial, and all other attributes contain atomic values). It's in 2NF and 3NF as Payment_ID is fully dependent on Pay_Serial, and there are no transitive dependencies.

Customer_Contact Table

Customer_Contact(Customer_ID, C_Phone)

It's in 1NF (primary key is Customer_ID, and all other attributes contain atomic values), 2NF, and 3NF (no partial or transitive dependencies).

Order_Customer Table

Order_Customer(O_Serial,O_ID, C_ID,P_ID,Pay_ID)

It's in 1NF (primary key is O_ID, and all other attributes contain atomic values). It's in 2NF and 3NF as C__ID is fully dependent on O_ID, and there are no transitive dependencies.

Finalization

Product Table

Product(P_ID, P_Category, P_Name, P_Quantity, P_Price, P_Availability)

Orders Table

Orders(Order_ID, Order_Address, Order_Date, Order_Amount)

Customer Table

Customer(C_ID, C_Name, C_Address, C_Email)

Payment Table

Payment(Pay_ID, Pay_Amount, Pay_Date)

Payment_Details Table

Payment_Details(Payment_ID, Pay_Serial, Pay_Method)

Customer_Contact Table

Customer_Contact(Customer_ID, C_Phone)

Order_Customer Table

Order_Customer(O_Serial,O_ID, C_ID,P_ID,Pay_ID)

Schema of Tables

Product Table

Schema: Product(P_ID, P_Category, P_Name, P_Quantity, P_Price, P_Availability)

Primary Key: P_ID

Orders Table

Schema: Orders(Order_ID, Order_Address, Order_Date, Order_Amount)

Primary Key: Order_ID

Customer Table

Schema: Customer(C_ID, C_Name, C_Address, C_Email)

Primary Key: C_ID

Payment Table

Schema: Payment(Pay_ID, Pay_Amount, Pay_Date)

Primary Key: Pay_ID

Payment_Details Table

Schema: Payment_Details(Payment_ID, Pay_Serial, Pay_Method)

Foreign Key: Payment_ID references Payment(Pay_ID)

Customer_Contact Table

Schema: Customer_Contact(Customer_ID, C_Phone)

Foreign Key: Customer_ID references Customer(C_ID)

Order_Customer Table

Schema: Order_Customer(O_Serial,O_ID, C_ID,P_ID,Pay_ID)

Foreign Keys: O_ID references Orders(Order_ID), C_ID references Customer(C_ID), Pay_ID references Payment(Pay_ID)

Schema Diagram

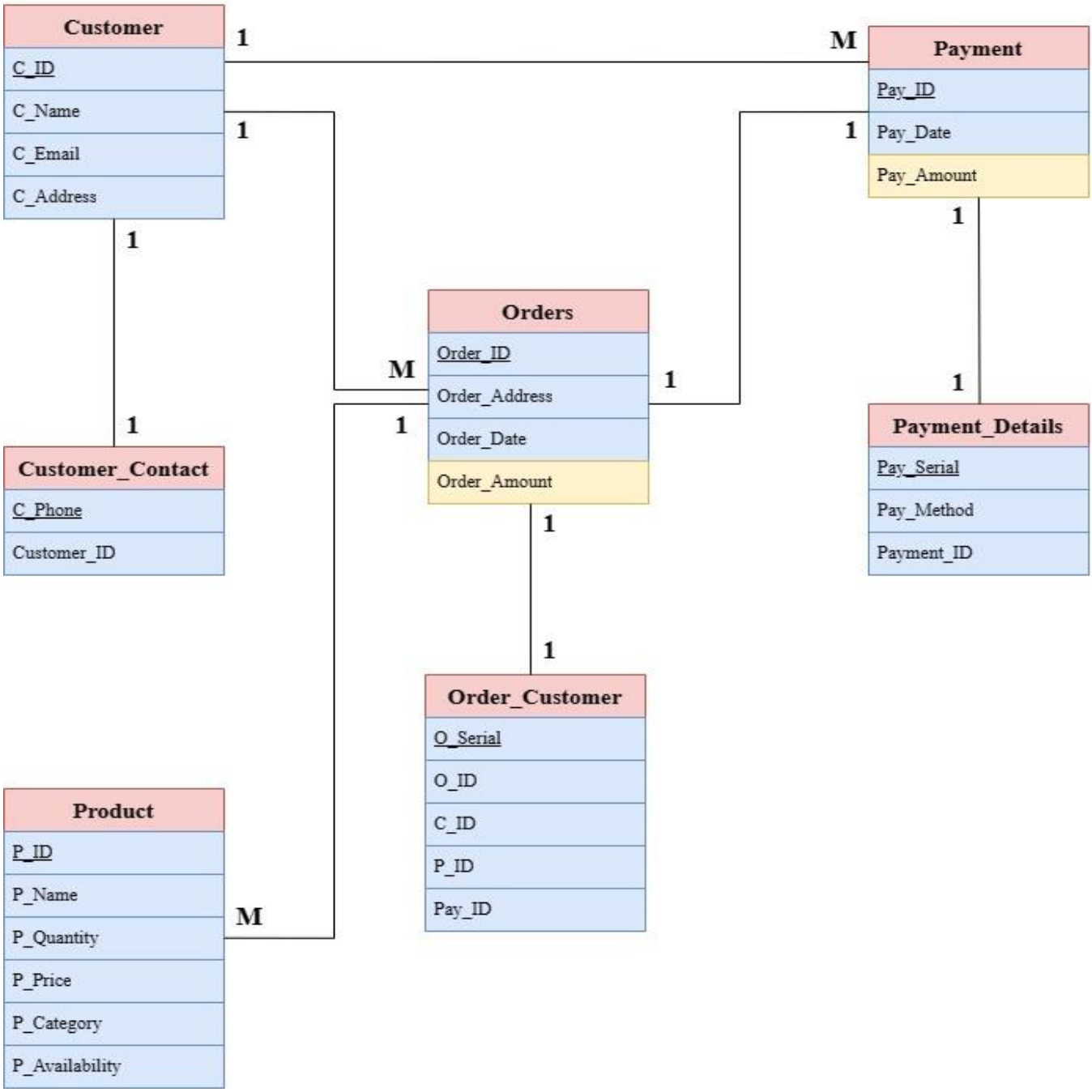


Table Creation & Value Insertion

Student ID 1: 23-50269-1 **Name: Faria Nourin**	**Student ID 3: 23-50892-1** **Name: Amrita Bhawal Disha**
Student ID 2: 23-50274-1 **Name: Barnabas Prentice**	**Student ID 4: 23-50976-1** **Name: Nafis Hasan**
CO4: Creating DML, DDL using Oracle and connection with ODBC/JDBC for existing JAVA application	
PO-e-2: Use modern engineering and IT tools for the prediction and modeling of complex computer science and engineering problem	Marks

Table Name: Customer

```
CREATE TABLE Customer (
C_ID NUMBER PRIMARY KEY NOT NULL,
C_Name VARCHAR2(200) NOT NULL,
C_Address VARCHAR2(200) NOT NULL,
C_Email VARCHAR2(200) NOT NULL
);
DESC Customer;
SELECT * FROM Customer;
CREATE SEQUENCE C_ID_Seq
INCREMENT BY 1
START WITH 1
MAXVALUE 100
NOCYCLE
NOCACHE;
```

ORACLE Database Express Edition

Home Logout Help

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
CREATE TABLE Customer (
C_ID NUMBER PRIMARY KEY NOT NULL,
C_Name VARCHAR2(200) NOT NULL,
C_Address VARCHAR2(200) NOT NULL,
C_Email VARCHAR2(200) NOT NULL
);
DESC Customer;
SELECT * FROM Customer;
CREATE SEQUENCE C_ID_Seq
INCREMENT BY 1
START WITH 1
MAXVALUE 100
NOCYCLE
NOCACHE;

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (1, 'Abir', 'Mirpur-1', 'abir@gmail.com');
```

Results Explain Describe Saved SQL History

```
Table created.
```

ORACLE Database Express Edition

Home Logout Help

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
CREATE TABLE Customer (
C_ID NUMBER PRIMARY KEY NOT NULL,
C_Name VARCHAR2(200) NOT NULL,
C_Address VARCHAR2(200) NOT NULL,
C_Email VARCHAR2(200) NOT NULL
);
DESC Customer;
SELECT * FROM Customer;
CREATE SEQUENCE C_ID_Seq
```

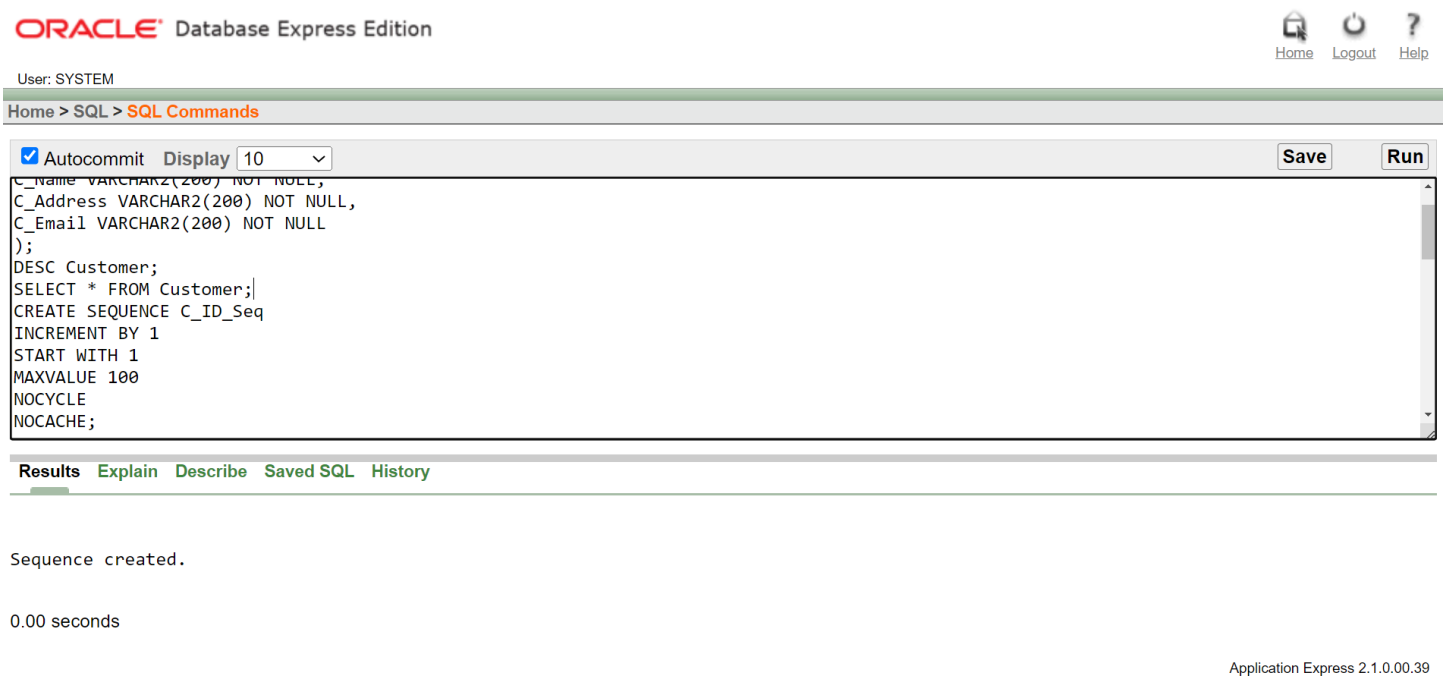
Results Explain Describe Saved SQL History

Object Type **TABLE** Object **CUSTOMER**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER	C_ID	Number	-	-	-	1	-	-	-
	C_NAME	Varchar2	200	-	-	-	-	-	-
	C_ADDRESS	Varchar2	200	-	-	-	-	-	-
	C_EMAIL	Varchar2	200	-	-	-	-	-	-

1 - 4

Application Express 2.1.0.00.39



INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (1, 'Abir', 'Mirpur-1', 'abir@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (2, 'Asif', 'Malibag', 'asif@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (3, 'Aman', 'Dhanmondi', 'aman@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (4, 'Akib', 'Madaripur', 'akib@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (5, 'Abid', 'Cumilla', 'abid@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (6, 'Arif', 'Kuratoli', 'arif@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (7, 'Anim', 'Wari', 'anim@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (8, 'Abdullah', 'Mirpur-10', 'abdullah@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (9, 'Basir', 'Tongi', 'basir@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (10, 'Barik', 'Puran Dhaka', 'barik@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (11, 'Barno', 'Jigatola', 'barno@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (12, 'Toma', 'Azimpur', 'toma@gmail.com');

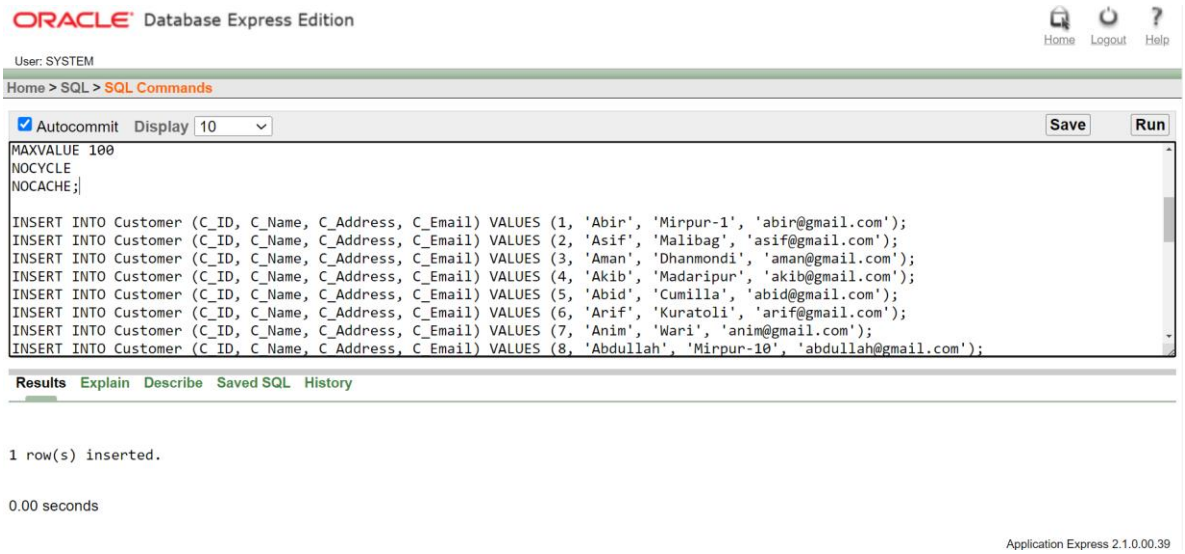
INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (13, 'Riya', 'Sitakunda', 'riya@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (14, 'Mila', 'Shahbag', 'mila@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (15, 'Tanisha', 'Green Road', 'Tanisha@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (16, 'Isha', 'Mohammodpur', 'Isha@gmail.com');

```
INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (17, 'Zarin', 'Kamrangirchor',
'zarin@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (18, 'Ayman', 'Monipur',
'Ayman@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (19, 'Papiya', 'Lalbag', 'Papiya@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (20, 'Tapash', 'Farngate',
'Tapash@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (21, 'Bir', 'Demra', 'bir@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (22, 'Asifa', 'Gulshan', 'asifa@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (23, 'Aman', 'Badda', 'aman@egmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (24, 'Akiba', 'Amirabad', 'akib@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (25, 'Abida', 'Kuril', 'abid@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (26, 'Arifa', 'Bongobazar',
'arifa@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (27, 'Anima', 'Pathapath',
'anima@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (28, 'Ankit', 'Pallabi', 'ankit@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (29, 'Babul', 'Uttara', 'babul@gmail.com');

INSERT INTO Customer (C_ID, C_Name, C_Address, C_Email) VALUES (30, 'Mithun', 'Gazipur',
'mithun@gmail.com');
```



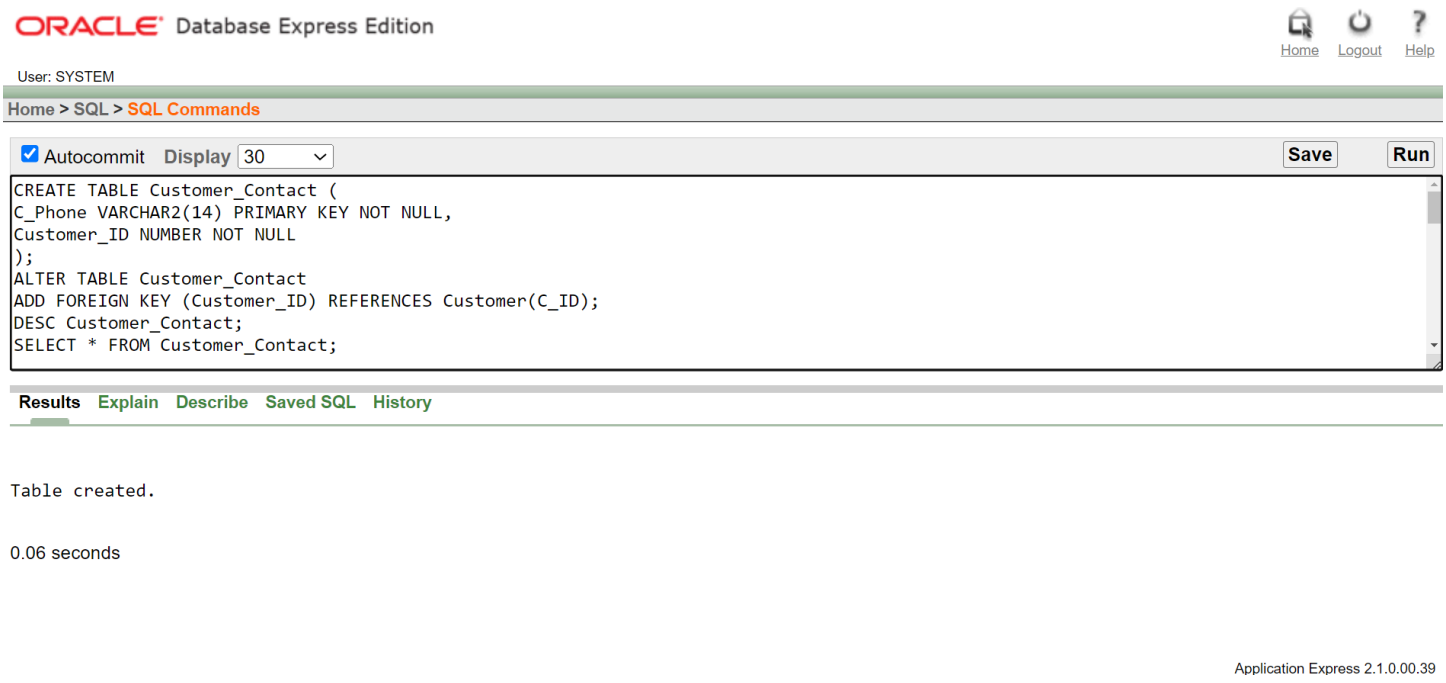
Results Explain Describe Saved SQL History

C_ID	C_NAME	C_ADDRESS	C_EMAIL
1	Abir	Mirpur-1	abir@gmail.com
2	Asif	Malibag	asif@gmail.com
3	Aman	Dhanmondi	aman@gmail.com
4	Akib	Madaripur	akib@gmail.com
5	Abid	Cumilla	abid@gmail.com
6	Arif	Kuratoli	arif@gmail.com
7	Anim	Wari	anim@gmail.com
8	Abdullah	Mirpur-10	abdullah@gmail.com
9	Basir	Tongi	basir@gmail.com
10	Barik	Puran Dhaka	barik@gmail.com
11	Barno	Jigatola	barno@gmail.com
12	Toma	Azimpur	toma@gmail.com
13	Riya	Sitakunda	riya@gmail.com
14	Mila	Shahbag	mila@gmail.com
15	Tanisha	Green Road	Tanisha@gmail.com
16	Isha	Mohammodpur	Isha@gmail.com
17	Zarin	Kamrangirchor	zarin@gmail.com
18	Ayman	Monipur	Ayman@gmail.com
19	Papiya	Lalbag	Papiya@gmail.com
20	Tapash	Farngate	Tapash@gmail.com
21	Bir	Demra	bir@gmail.com
22	Asifa	Gulshan	asifa@gmail.com
23	Aman	Badda	aman@egmail.com
24	Akiba	Amirabad	akib@gmail.com
25	Abida	Kuril	abid@gmail.com
26	Arifa	Bongobazar	arifa@gmail.com
27	Anima	Pathapath	anima@gmail.com
28	Ankit	Pallabi	ankit@gmail.com
29	Babul	Uttara	babul@gmail.com
30	Mithun	Gazipur	mithun@gmail.com

30 rows returned in 0.00 seconds CSV Export

Table Name: Customer_Contact

```
CREATE TABLE Customer_Contact (
C_Phone VARCHAR2(14) PRIMARY KEY NOT NULL,
Customer_ID NUMBER NOT NULL
);
ALTER TABLE Customer_Contact
ADD FOREIGN KEY (Customer_ID) REFERENCES Customer(C_ID);
DESC Customer_Contact;
SELECT * FROM Customer_Contact;
```



ORACLE Database Express Edition

Home Logout Help

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 30 Save Run

```
CREATE TABLE Customer_Contact (
C_Phone VARCHAR2(14) PRIMARY KEY NOT NULL,
Customer_ID NUMBER NOT NULL
);
ALTER TABLE Customer_Contact
ADD FOREIGN KEY (Customer_ID) REFERENCES Customer(C_ID);
DESC Customer_Contact;
SELECT * FROM Customer_Contact;
```

Results Explain Describe Saved SQL History

```
Table altered.
```

0.03 seconds

ORACLE Database Express Edition

Home Logout Help

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 30 Save Run

```
CREATE TABLE Customer_Contact (
C_Phone VARCHAR2(14) PRIMARY KEY NOT NULL,
Customer_ID NUMBER NOT NULL
);
ALTER TABLE Customer_Contact
ADD FOREIGN KEY (Customer_ID) REFERENCES Customer(C_ID);
DESC Customer_Contact;
SELECT * FROM Customer_Contact;
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **CUSTOMER_CONTACT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMER_CONTACT	C_PHONE	Varchar2	14	-	-	1	-	-	-
	CUSTOMER_ID	Number	-	-	-	-	-	-	-

1 - 2

INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0172139675', 1);

INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0158557567', 2);

INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0168557567', 3);

INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0178557567', 4);

INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0148557567', 5);

INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0123456789', 6);

INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0123456743', 7);

INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0193456743', 8);

INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0123456780', 9);

INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0198557597', 10);

```
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0168557870', 11);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0168567567', 12);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0148985767', 13);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0123454780', 14);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0198167597', 15);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0168557880', 16);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0116856757', 17);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0148985769', 18);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0126656810', 19);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0198885597', 20);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0168457870', 21);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0168567467', 22);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0148987877', 23);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0123454310',24);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0197935456', 25);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0134543456', 26);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0124547910', 27);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0123123910', 28);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0129867910', 29);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0129876510', 30);
```

ORACLE Database Express Edition

Home Logout Help

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 30 Save Run

```
CREATE TABLE Customer_Contact (
C_Phone VARCHAR2(14) PRIMARY KEY NOT NULL,
Customer_ID NUMBER NOT NULL
);
ALTER TABLE Customer_Contact
ADD FOREIGN KEY (Customer_ID) REFERENCES Customer(C_ID);
DESC Customer_Contact;
SELECT * FROM Customer_Contact;

INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0198557567', 1);
INSERT INTO Customer_Contact (C_Phone, Customer_ID) VALUES ('0158557567', 2);
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.00 seconds

Application Express 2.1.0.00.39

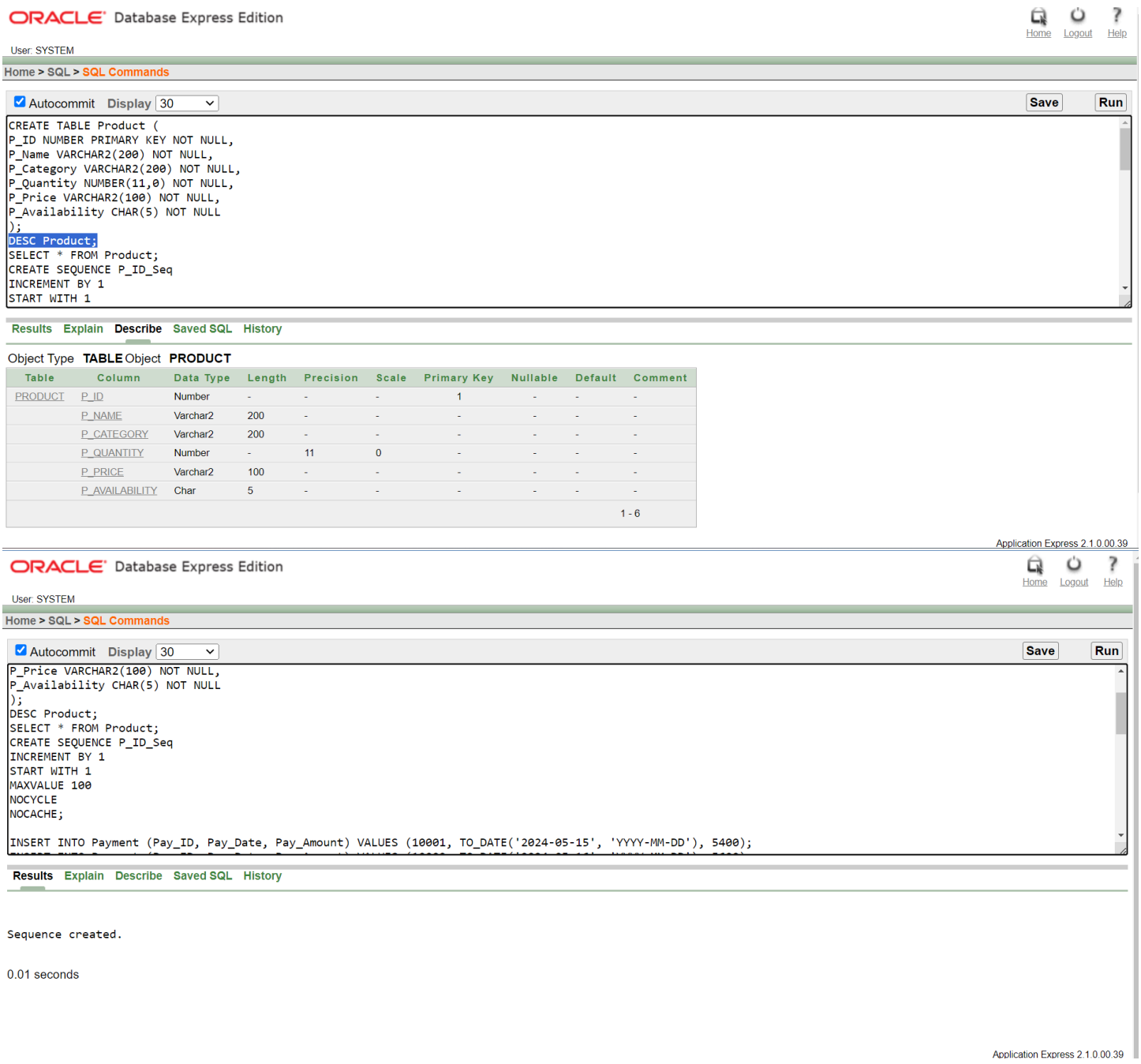
Results Explain Describe Saved SQL History

C_PHONE	CUSTOMER_ID
0198557567	1
0158557567	2
0168557567	3
0178557567	4
0148557567	5
0123456789	6
0123456743	7
0193456743	8
0123456780	9
0198557597	10
0168557870	11
0168567567	12
0148985767	13
0123454780	14
0198167597	15
0168557880	16
0116856757	17
0148985769	18
0126656810	19
0198885597	20
0168457870	21
0168567467	22
0148987877	23
0123454310	24
0134543456	26
0124547910	27
0123123910	28
0129867910	29
0129876510	30
0197935456	25

30 rows returned in 0.00 seconds CSV Export

Table Name: Product

```
CREATE TABLE Product (
P_ID NUMBER PRIMARY KEY NOT NULL,
P_Name VARCHAR2(200) NOT NULL,
P_Category VARCHAR2(200) NOT NULL,
P_Quantity NUMBER(11,0) NOT NULL,
P_Price VARCHAR2(100) NOT NULL,
P_Availability CHAR(5) NOT NULL
);
DESC Product;
SELECT * FROM Product;
CREATE SEQUENCE P_ID_Seq
INCREMENT BY 1
START WITH 1
MAXVALUE 100
NOCYCLE
NOCACHE;
```



INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1001, 'POTATO', 'VEGETABLE', 500, '60 PER KG', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1002, 'TOMATO', 'VEGETABLE', 0, '70 PER KG', 'N');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1003, 'CAKES', 'FOOD', 300, '75 PER KG', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1004, 'SUGAR', 'FOOD', 500, '160 PER KG', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1005, 'FLOUR', 'FOOD', 300, '65 PER KG', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1006, 'BEAN', 'VEGETABLE', 800, '90 PER KG', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1007, 'ONION', 'VEGETABLE', 700, '60 PER KG', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1008, 'GARLIC', 'VEGETABLE', 500, '50 PER KG', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1009, 'RED PEPPER', 'VEGETABLE', 900, '80 PER KG', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1010, 'RADISH', 'VEGETABLE', 400, '70 PER KG', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1011, 'BANANA', 'FRUIT', 500, '120 PER DOZEN', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1012, 'APPLE', 'FRUIT', 100, '260 PER KG', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1013, 'SHAMPOO', 'COSMETICS', 200, '320', 'N');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1014, 'CHANACHUR', 'SNACKS', 60, '1 PER PECKET', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1015, 'PASTA', 'FOOD', 200, '100 PER PACKET', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1016, 'NOODLES', 'FOOD', 0, '50 PER PACKET', 'N');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1017, 'PRINGLES', 'SNACKS', 100, '330 PER PACKET', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1018, 'POTATO CHIPS', 'SNACKS', 200, '20 PER PACKET', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1019, 'KURKURE', 'SNACKS', 100, '20 PER PACKET', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1020, 'MANGO', 'FRUITS', 0, '300 PER KG', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1021, 'MILK', 'DAIRY', 200, '50 PER PACKET', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1022, 'CHOCOLATE MILK', 'DAIRY', 0, '30 PER PACKET', 'N');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1023, 'JUICE', 'DRINK', 100, '40 PER PACKET', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1024, 'DISHWASH', 'CLEANING SUPPLIES', 100, '65 PER PACKET', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1025, 'SOUP', 'FOOD', 200, '50 PER SACHET', 'Y');

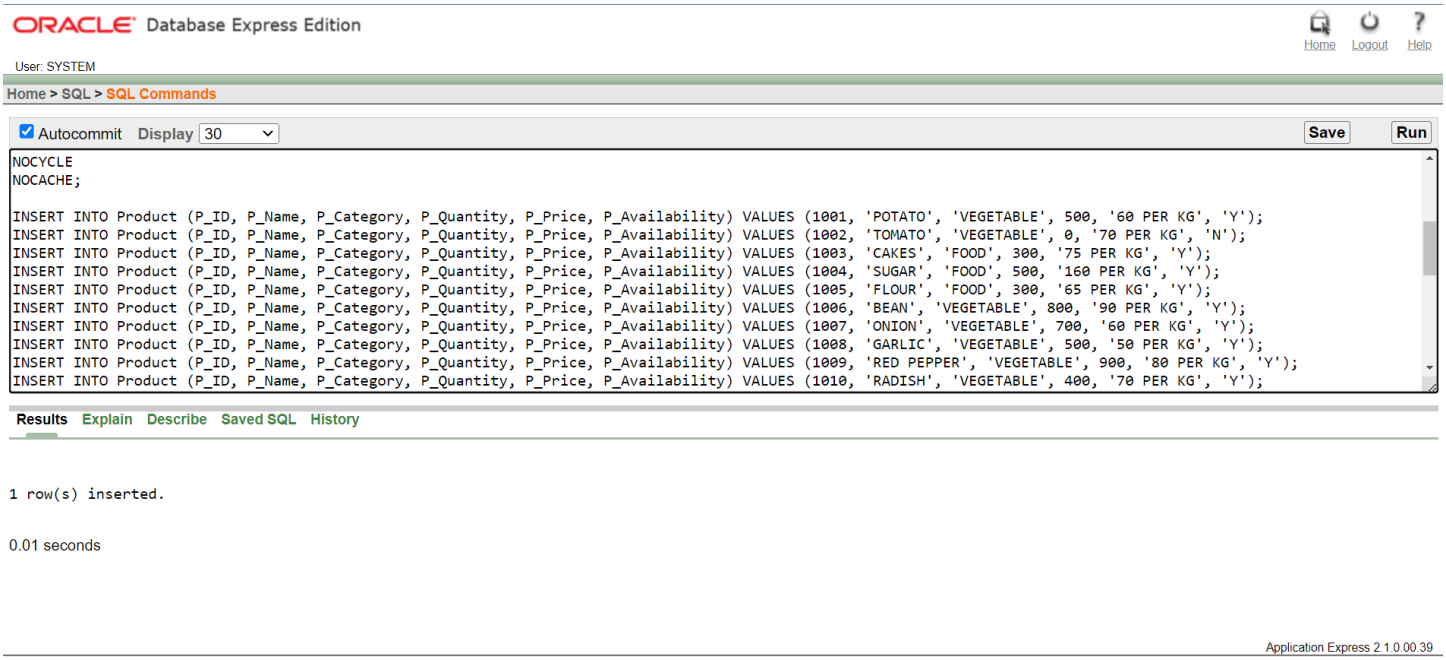
INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1026, 'EGG', 'FOOD', 200, '150 PER DOZEN', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1027, 'RICE', 'FOOD', 200, '67 PER KG', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1028, 'LENTILS', 'FOOD', 300, '50 PER KG', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1029, 'RED SAUCE', 'FOOD', 200, '80 PER BOTTLE', 'Y');

INSERT INTO Product (P_ID, P_Name, P_Category, P_Quantity, P_Price, P_Availability) VALUES (1030, 'RC COLA', 'DRINK', 240, '20 PER BOTTLE', 'Y');



Results Explain Describe Saved SQL History

P_ID	P_NAME	P_CATEGORY	P_QUANTITY	P_PRICE	P_AVAILABILITY
1001	POTATO	VEGETABLE	500	60 PER KG	Y
1002	TOMATO	VEGETABLE	0	70 PER KG	N
1003	CAKES	FOOD	300	75 PER KG	Y
1004	SUGAR	FOOD	500	160 PER KG	Y
1005	FLOUR	FOOD	300	65 PER KG	Y
1006	BEAN	VEGETABLE	800	90 PER KG	Y
1007	ONION	VEGETABLE	700	60 PER KG	Y
1008	GARLIC	VEGETABLE	500	50 PER KG	Y
1009	RED PEPPER	VEGETABLE	900	80 PER KG	Y
1010	RADISH	VEGETABLE	400	70 PER KG	Y
1011	BANANA	FRUIT	500	120 PER DOZEN	Y
1012	APPLE	FRUIT	100	260 PER KG	Y
1013	SHAMPOO	COSMETICS	200	320	N
1014	CHANACHUR	SNACKS	60	1 PER PECKET	Y
1015	PASTA	FOOD	200	100 PER PACKET	Y
1016	NOODLES	FOOD	0	50 PER PACKET	N
1017	PRINGLES	SNACKS	100	330 PER PACKET	Y
1018	POTATO CHIPS	SNACKS	200	20 PER PACKET	Y
1019	KURKURE	SNACKS	100	20 PER PACKET	Y
1020	MANGO	FRUITS	0	300 PER KG	Y
1021	MILK	DAIRY	200	50 PER PACKET	Y
1022	CHOCOLATE MILK	DAIRY	0	30 PER PACKET	N
1023	JUICE	DRINK	100	40 PER PACKET	Y
1024	DISHWASH	CLEANING SUPPLIES	100	65 PER PACKET	Y
1025	SOUP	FOOD	200	50 PER SACHET	Y
1026	EGG	FOOD	200	150 PER DOZEN	Y
1027	RICE	FOOD	200	67 PER KG	Y
1028	LENTILS	FOOD	300	50 PER KG	Y
1029	RED SAUCE	FOOD	200	80 PER BOTTLE	Y
1030	RC COLA	DRINK	240	20 PER BOTTLE	Y

30 rows returned in 0.00 seconds CSV Export

Table Name: Payment

CREATE TABLE Payment (

Pay_ID NUMBER PRIMARY KEY NOT NULL,

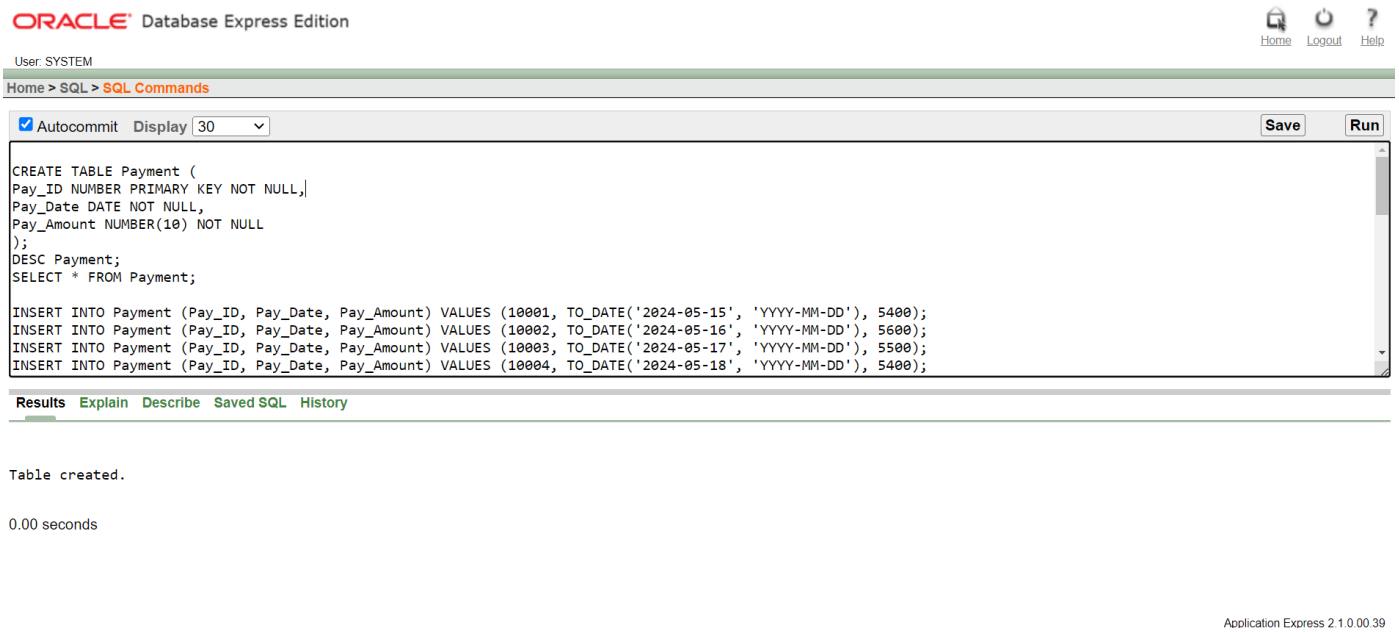
Pay_Date DATE NOT NULL,

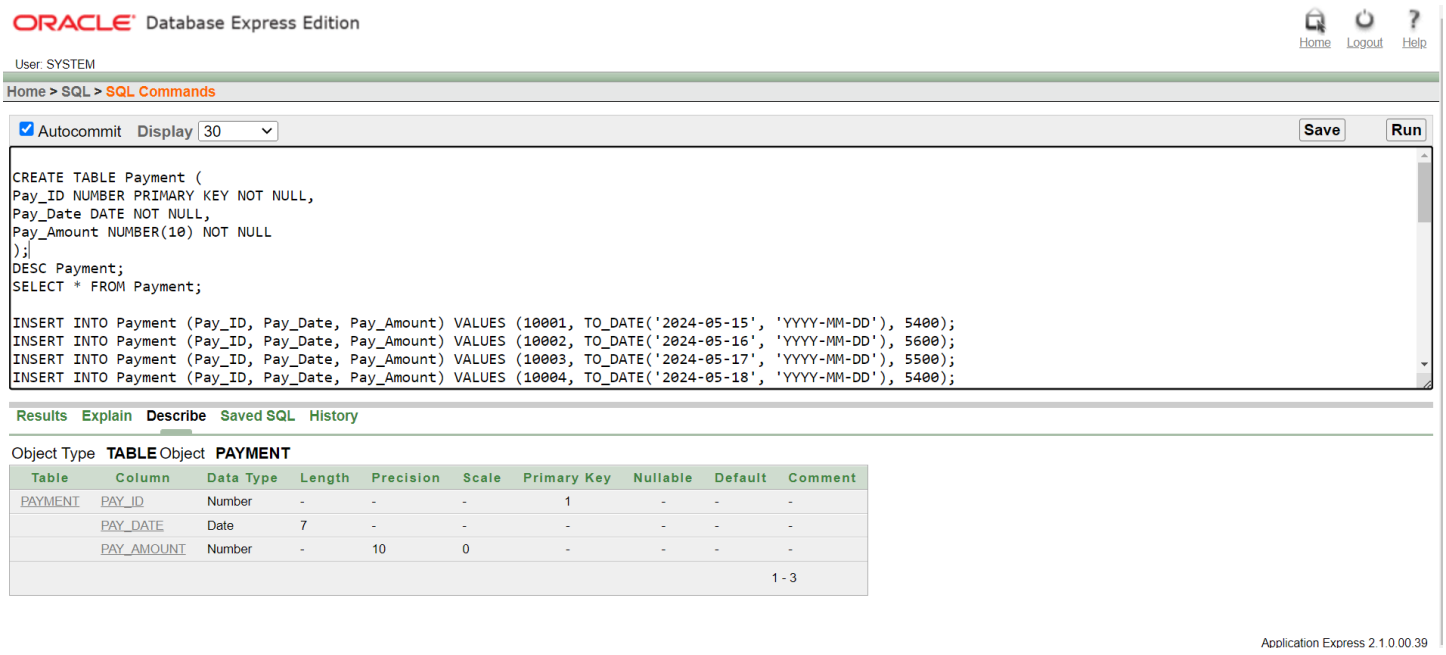
Pay_Amount NUMBER(10) NOT NULL

);

DESC Payment;

SELECT * FROM Payment;





INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10001, TO_DATE('2024-05-15', 'YYYY-MM-DD'), 5400);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10002, TO_DATE('2024-05-16', 'YYYY-MM-DD'), 5600);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10003, TO_DATE('2024-05-17', 'YYYY-MM-DD'), 5500);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10004, TO_DATE('2024-05-18', 'YYYY-MM-DD'), 5400);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10005, TO_DATE('2024-05-19', 'YYYY-MM-DD'), 6400);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10006, TO_DATE('2024-05-11', 'YYYY-MM-DD'), 6400);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10007, TO_DATE('2024-05-21', 'YYYY-MM-DD'), 4000);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10008, TO_DATE('2024-05-22', 'YYYY-MM-DD'), 7400);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10009, TO_DATE('2024-05-23', 'YYYY-MM-DD'), 5500);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10010, TO_DATE('2024-05-24', 'YYYY-MM-DD'), 5490);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10011, TO_DATE('2024-05-25', 'YYYY-MM-DD'), 5400);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10012, TO_DATE('2024-05-26', 'YYYY-MM-DD'), 5400);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10013, TO_DATE('2024-05-27', 'YYYY-MM-DD'), 5500);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10014, TO_DATE('2024-05-28', 'YYYY-MM-DD'), 5400);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10015, TO_DATE('2024-05-29', 'YYYY-MM-DD'), 6400);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10016, TO_DATE('2024-04-01', 'YYYY-MM-DD'), 4400);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10017, TO_DATE('2024-04-02', 'YYYY-MM-DD'), 4000);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10018, TO_DATE('2024-04-03', 'YYYY-MM-DD'), 7400);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10019, TO_DATE('2024-04-04', 'YYYY-MM-DD'), 5500);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10020, TO_DATE('2024-04-24', 'YYYY-MM-DD'), 5490);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10021, TO_DATE('2024-04-25', 'YYYY-MM-DD'), 6400);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10022, TO_DATE('2024-04-26', 'YYYY-MM-DD'), 4000);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10023, TO_DATE('2024-04-27', 'YYYY-MM-DD'), 4000);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10024, TO_DATE('2024-04-28', 'YYYY-MM-DD'), 7400);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10025, TO_DATE('2024-03-01', 'YYYY-MM-DD'), 5000);

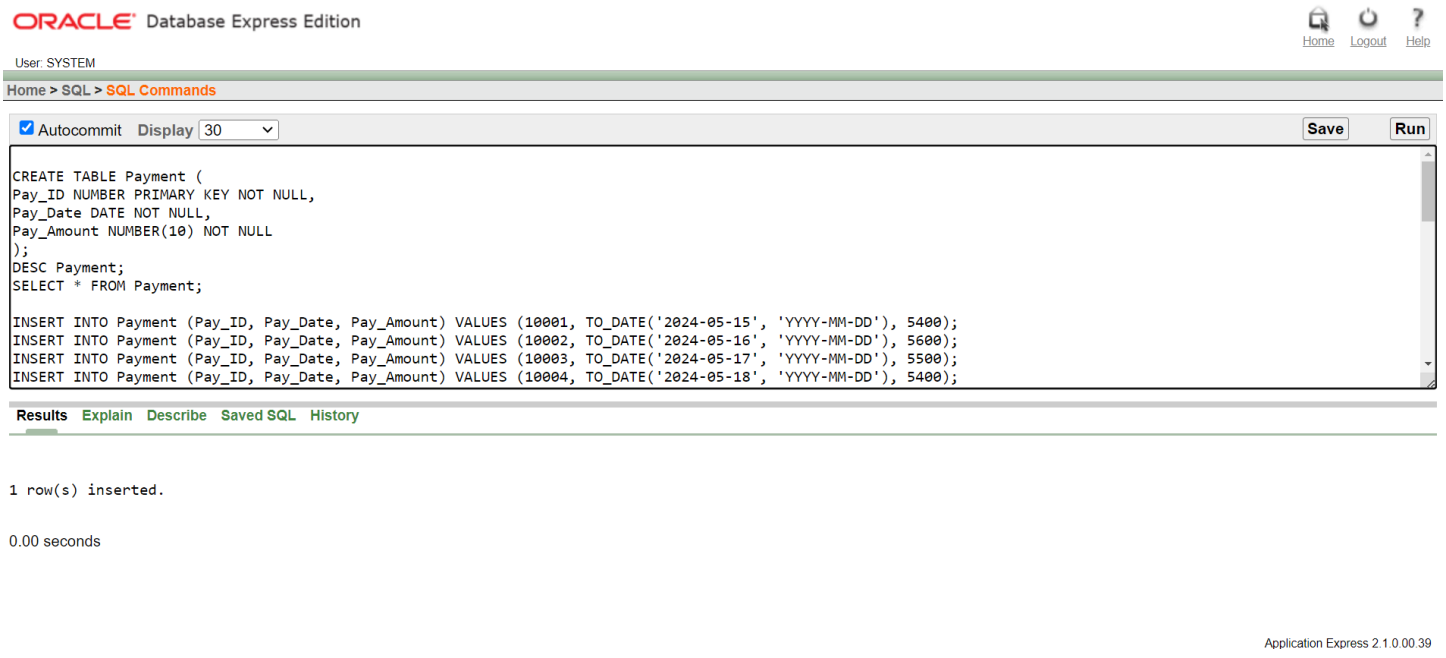
INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10026, TO_DATE('2024-03-02', 'YYYY-MM-DD'), 5090);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10027, TO_DATE('2024-03-03', 'YYYY-MM-DD'), 4000);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10028, TO_DATE('2024-03-04', 'YYYY-MM-DD'), 7400);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10029, TO_DATE('2024-03-05', 'YYYY-MM-DD'), 5000);

INSERT INTO Payment (Pay_ID, Pay_Date, Pay_Amount) VALUES (10030, TO_DATE('2024-03-06', 'YYYY-MM-DD'), 5090);

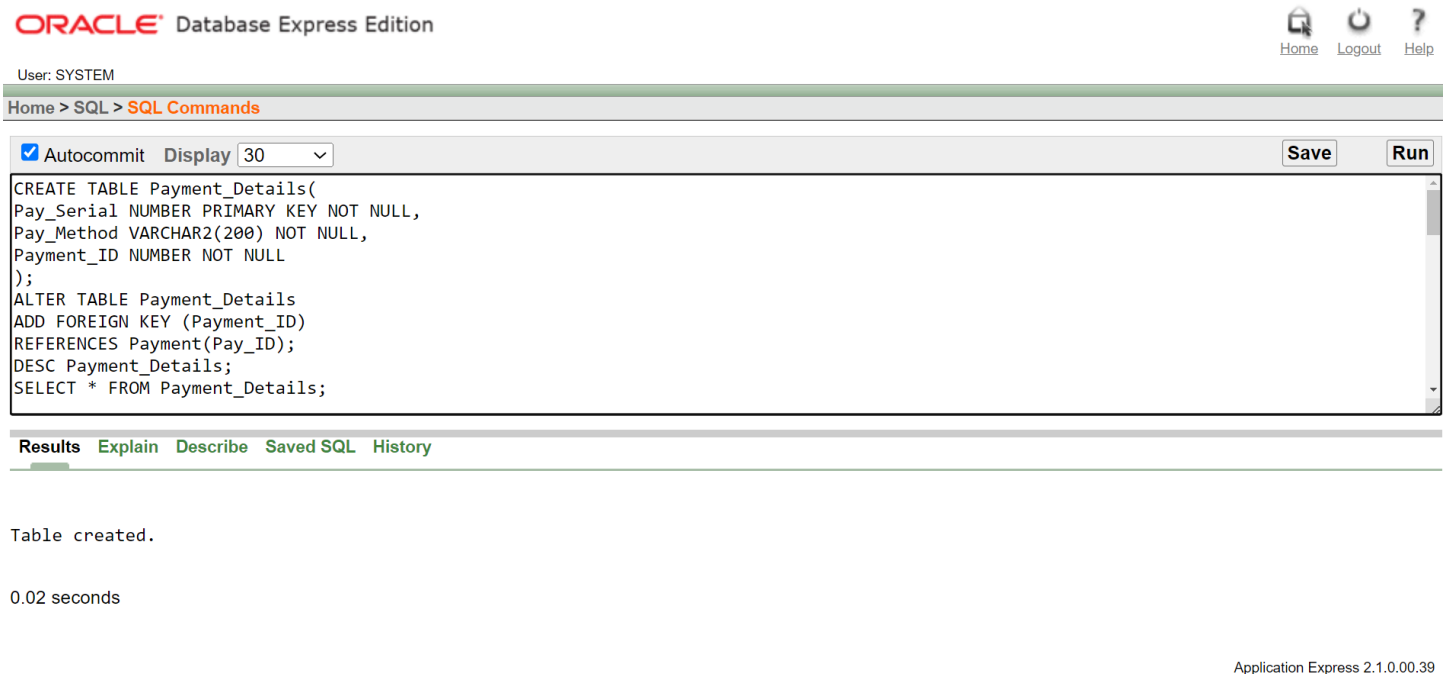


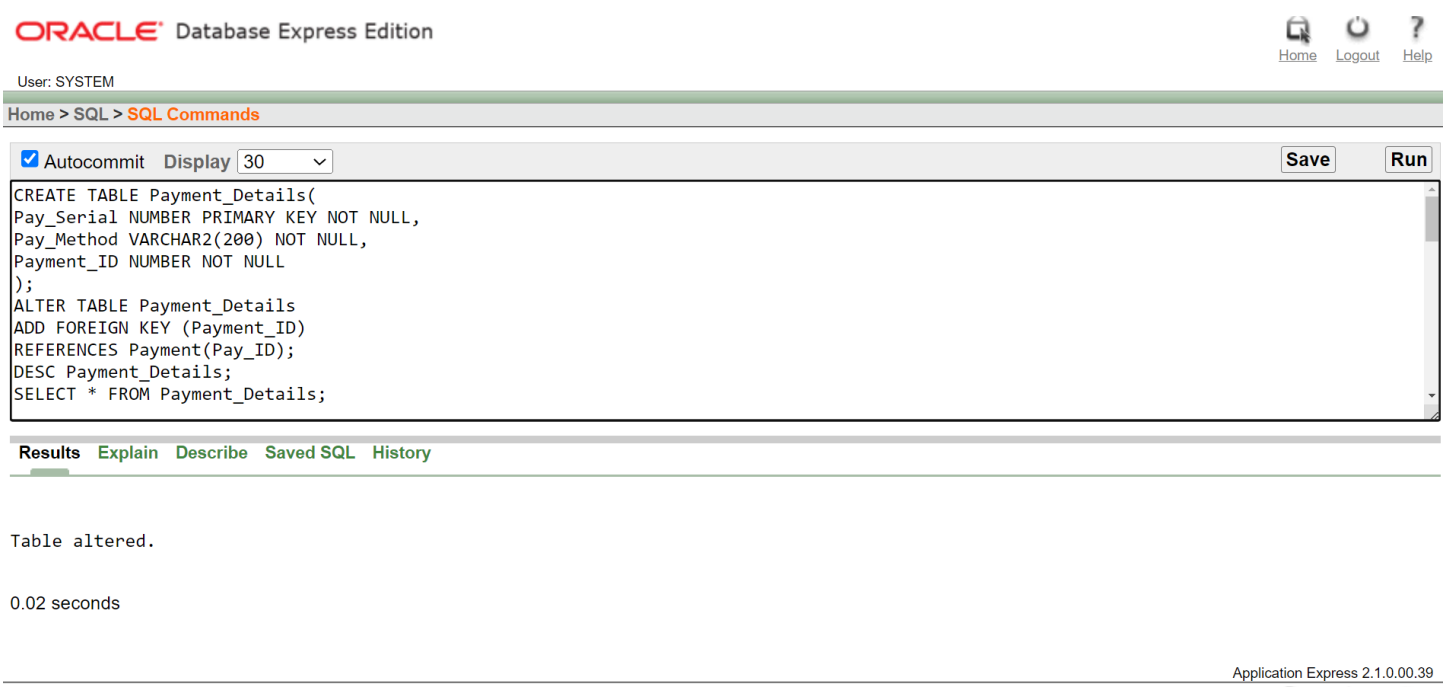
PAY_ID	PAY_DATE	PAY_AMOUNT
10001	15-MAY-24	5400
10002	16-MAY-24	5600
10003	17-MAY-24	5500
10004	18-MAY-24	5400
10005	19-MAY-24	6400
10006	11-MAY-24	6400
10007	21-MAY-24	4000
10008	22-MAY-24	7400
10009	23-MAY-24	5500
10010	24-MAY-24	5490
10011	25-MAY-24	5400
10012	26-MAY-24	5400
10013	27-MAY-24	5500
10014	28-MAY-24	5400
10015	29-MAY-24	6400
10016	01-APR-24	4400
10017	02-APR-24	4000
10018	03-APR-24	7400
10019	04-APR-24	5500
10020	24-APR-24	5490
10021	25-APR-24	6400
10022	26-APR-24	4000
10023	27-APR-24	4000
10024	28-APR-24	7400
10025	01-MAR-24	5000
10026	02-MAR-24	5090
10027	03-MAR-24	4000
10028	04-MAR-24	7400
10029	05-MAR-24	5000
10030	06-MAR-24	5090

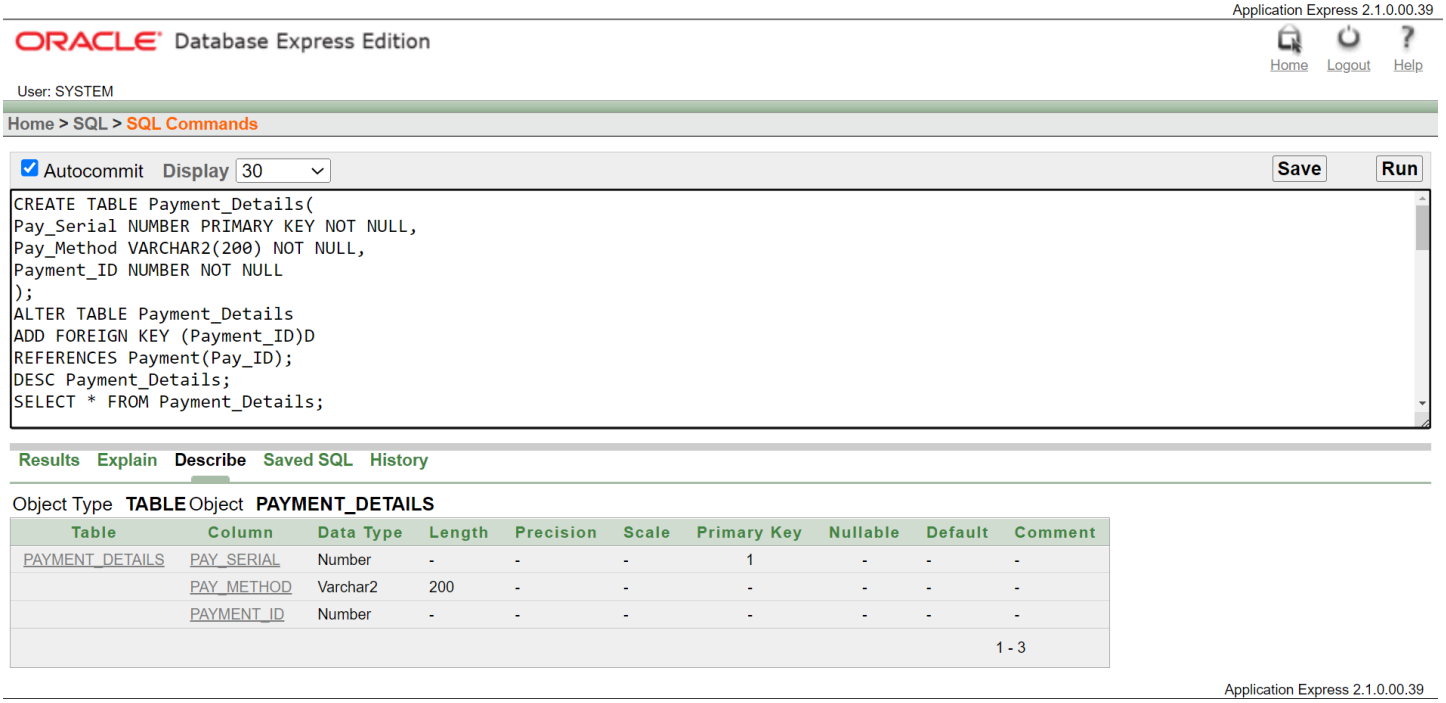
30 rows returned in 0.00 seconds CSV Export

Table Name: Payment_Details

```
CREATE TABLE Payment_Details(
Pay_Serial NUMBER PRIMARY KEY NOT NULL,
Pay_Method VARCHAR2(200) NOT NULL,
Payment_ID NUMBER NOT NULL
);
ALTER TABLE Payment_Details
ADD FOREIGN KEY (Payment_ID)D
REFERENCES Payment(Pay_ID);
DESC Payment_Details;
SELECT * FROM Payment_Details;
```







INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (1, 'CASH', 10001);

INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (2, 'CARD', 10002);

INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (3, 'CASH', 10003);

INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (4, 'CARD', 10004);

INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (5, 'CASH', 10005);

INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (6, 'BKASH', 10006);

INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (7, 'CASH', 10007);

```
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (8, 'BKASH', 10008);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (9, 'CASH', 10009);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (10, 'CARD', 10010);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (11, 'CASH', 10011);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (12, 'CARD', 10012);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (13, 'CASH', 10013);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (14, 'CARD', 10014);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (15, 'CASH', 10015);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (16, 'BKASH', 10016);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (17, 'CASH', 10017);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (18, 'BKASH', 10018);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (19, 'CASH', 10019);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (20, 'CARD', 10020);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (21, 'CASH', 10021);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (22, 'CARD', 10022);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (23, 'CASH', 10023);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (24, 'CARD', 10024);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (25, 'CASH', 10025);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (26, 'BKASH', 10026);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (27, 'CASH', 10027);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (28, 'BKASH', 10028);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (29, 'CASH', 10029);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (30, 'CARD', 10030);
```

ORACLE Database Express Edition

Home Logout Help

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 30 Save Run

```
DESC Payment_Details;
SELECT * FROM Payment_Details;

INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (1, 'CASH', 10001);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (2, 'CARD', 10002);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (3, 'CASH', 10003);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (4, 'CARD', 10004);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (5, 'CASH', 10005);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (6, 'BKASH', 10006);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (7, 'CASH', 10007);
INSERT INTO Payment_Details (Pay_Serial, Pay_Method, Payment_ID) VALUES (8, 'BKASH', 10008);
```

Results Explain Describe Saved SQL History

```
1 row(s) inserted.

0.01 seconds
```

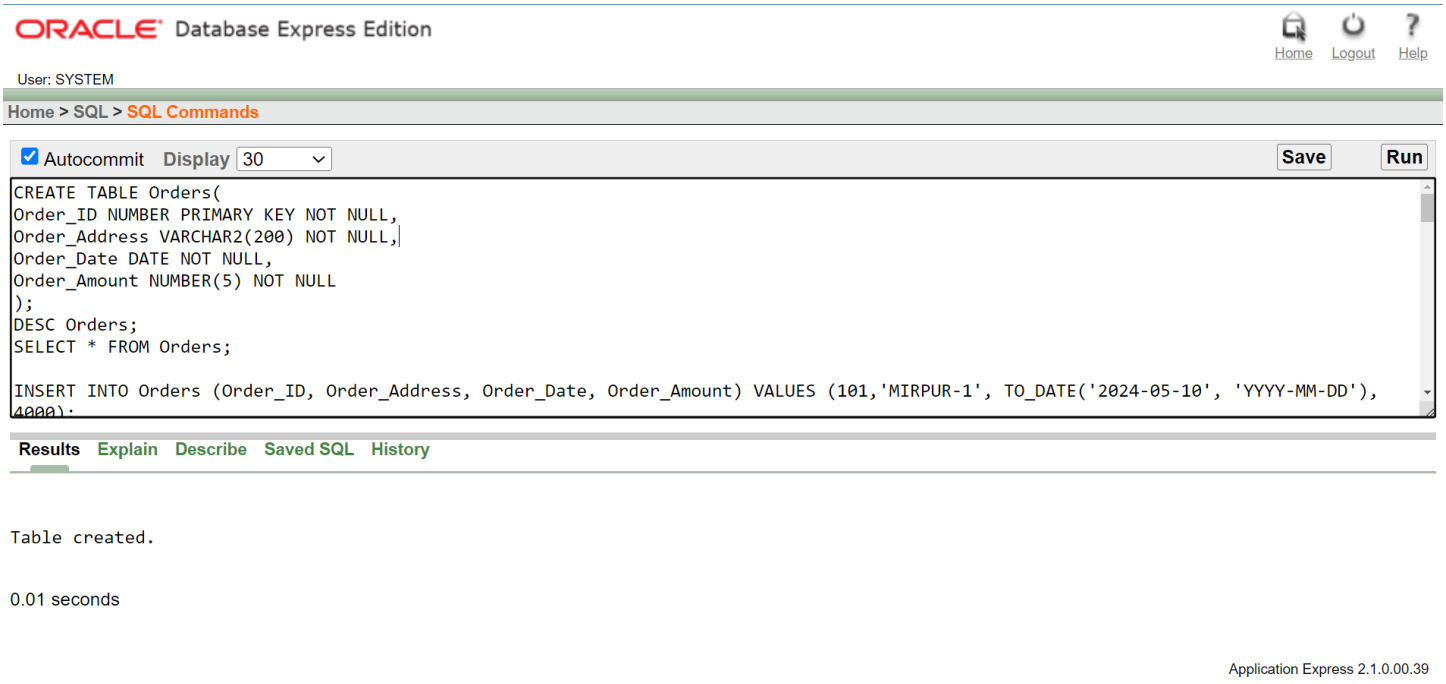
Application Express 2.1.0.00.39

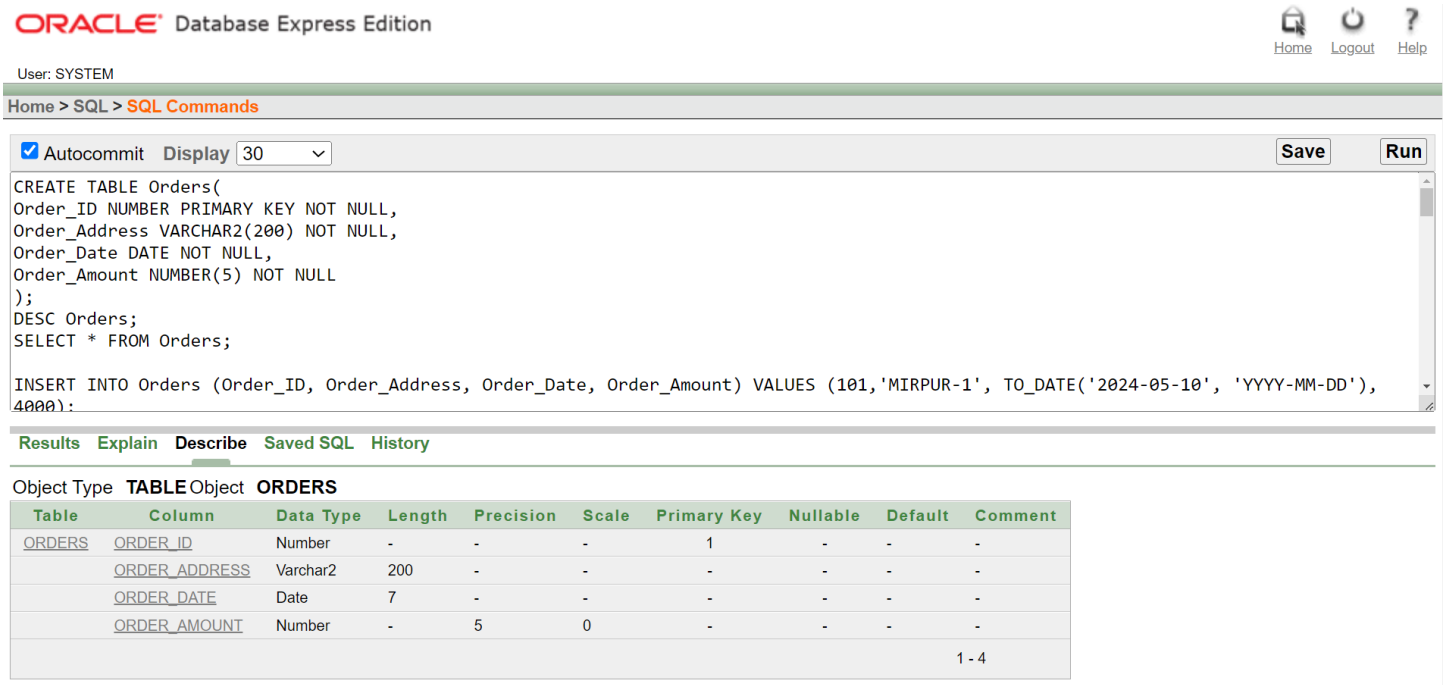
PAY_SERIAL	PAY_METHOD	PAYMENT_ID
1	CASH	10001
2	CARD	10002
3	CASH	10003
4	CARD	10004
5	CASH	10005
6	BKASH	10006
7	CASH	10007
8	BKASH	10008
9	CASH	10009
10	CARD	10010
11	CASH	10011
12	CARD	10012
13	CASH	10013
14	CARD	10014
15	CASH	10015
16	BKASH	10016
17	CASH	10017
18	BKASH	10018
19	CASH	10019
20	CARD	10020
21	CASH	10021
22	CARD	10022
23	CASH	10023
24	CARD	10024
25	CASH	10025
26	BKASH	10026
27	CASH	10027
28	BKASH	10028
29	CASH	10029
30	CARD	10030

30 rows returned in 0.00 seconds CSV Export

Table Name: Order

```
CREATE TABLE Orders(
Order_ID NUMBER PRIMARY KEY NOT NULL,
Order_Address VARCHAR2(200) NOT NULL,
Order_Date DATE NOT NULL,
Order_Amount NUMBER(5) NOT NULL
);
DESC Orders;
SELECT * FROM Orders;
```





INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (101,'MIRPUR-1', TO_DATE('2024-05-10', 'YYYY-MM-DD'), 4000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (102,'DHAKA', TO_DATE('2024-05-10', 'YYYY-MM-DD'), 3000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (103,'DHANMONDI', TO_DATE('2024-05-11', 'YYYY-MM-DD'), 13000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (104,'MADARIPUR', TO_DATE('2024-05-11', 'YYYY-MM-DD'), 4000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (105,'CUMILLA', TO_DATE('2024-05-11', 'YYYY-MM-DD'), 6000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (106,'KURIL', TO_DATE('2024-05-19', 'YYYY-MM-DD'), 3000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (107,'PURANBAZAR', TO_DATE('2024-05-12', 'YYYY-MM-DD'), 7000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (108,'MIRPUR', TO_DATE('2024-05-12', 'YYYY-MM-DD'), 5000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (109,'BASHUNDHARA', TO_DATE('2024-05-13', 'YYYY-MM-DD'), 1000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (110,'UTTARA', TO_DATE('2024-05-13', 'YYYY-MM-DD'), 2000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (111,'PURANBAZAR', TO_DATE('2024-05-13', 'YYYY-MM-DD'), 6000);

```
INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (112,'BADAMTOLA',
TO_DATE('2024-05-13', 'YYYY-MM-DD'), 3000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (113,'KHULNA',
TO_DATE('2024-05-13', 'YYYY-MM-DD'), 5000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (114,'MIRPUR15',
TO_DATE('2024-05-13', 'YYYY-MM-DD'), 1000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (115,'KHULNA',
TO_DATE('2024-05-14', 'YYYY-MM-DD'), 3000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (116,'MIRPUR15',
TO_DATE('2024-05-14', 'YYYY-MM-DD'), 1000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (117,'BARISHAL',
TO_DATE('2024-05-14', 'YYYY-MM-DD'), 3000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (118,'MILONPUR',
TO_DATE('2024-05-15', 'YYYY-MM-DD'), 3000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (119,'LANGLABAD',
TO_DATE('2024-05-15', 'YYYY-MM-DD'), 5000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (120,'KULPODDI',
TO_DATE('2024-05-16', 'YYYY-MM-DD'), 8000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (121,'DHAMAL',
TO_DATE('2024-05-16', 'YYYY-MM-DD'), 2000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (122,'GULSHAN',
TO_DATE('2024-05-17', 'YYYY-MM-DD'), 3000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (123,'BADDA',
TO_DATE('2024-05-17', 'YYYY-MM-DD'), 5000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (124,'AMIRABAD',
TO_DATE('2024-05-17', 'YYYY-MM-DD'), 2500);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (125,'COXSBAZAR',
TO_DATE('2024-05-18', 'YYYY-MM-DD'), 1500);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (126,'PURANROAD',
TO_DATE('2024-05-18', 'YYYY-MM-DD'), 2000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (127,'PURANSOR',
TO_DATE('2024-05-19', 'YYYY-MM-DD'), 1000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (128,'MIRPUR2',
TO_DATE('2024-05-19', 'YYYY-MM-DD'), 4000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (129,'BAGURA',
TO_DATE('2024-05-19', 'YYYY-MM-DD'), 3000);

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (130,'BADDA',
TO_DATE('2024-05-23', 'YYYY-MM-DD'), 4500);
```

ORACLE Database Express Edition

Home Logout Help

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 30 Save Run

```
Order_Amount NUMBER(5) NOT NULL
);
DESC Orders;
SELECT * FROM Orders;

INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (101,'MIRPUR-1', TO_DATE('2024-05-10', 'YYYY-MM-DD'),
4000);
INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (102,'DHAKA', TO_DATE('2024-05-10', 'YYYY-MM-DD'),
3000);
INSERT INTO Orders (Order_ID, Order_Address, Order_Date, Order_Amount) VALUES (103,'DHANMONDI', TO_DATE('2024-05-11', 'YYYY-MM-DD'),
13000);
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.00 seconds

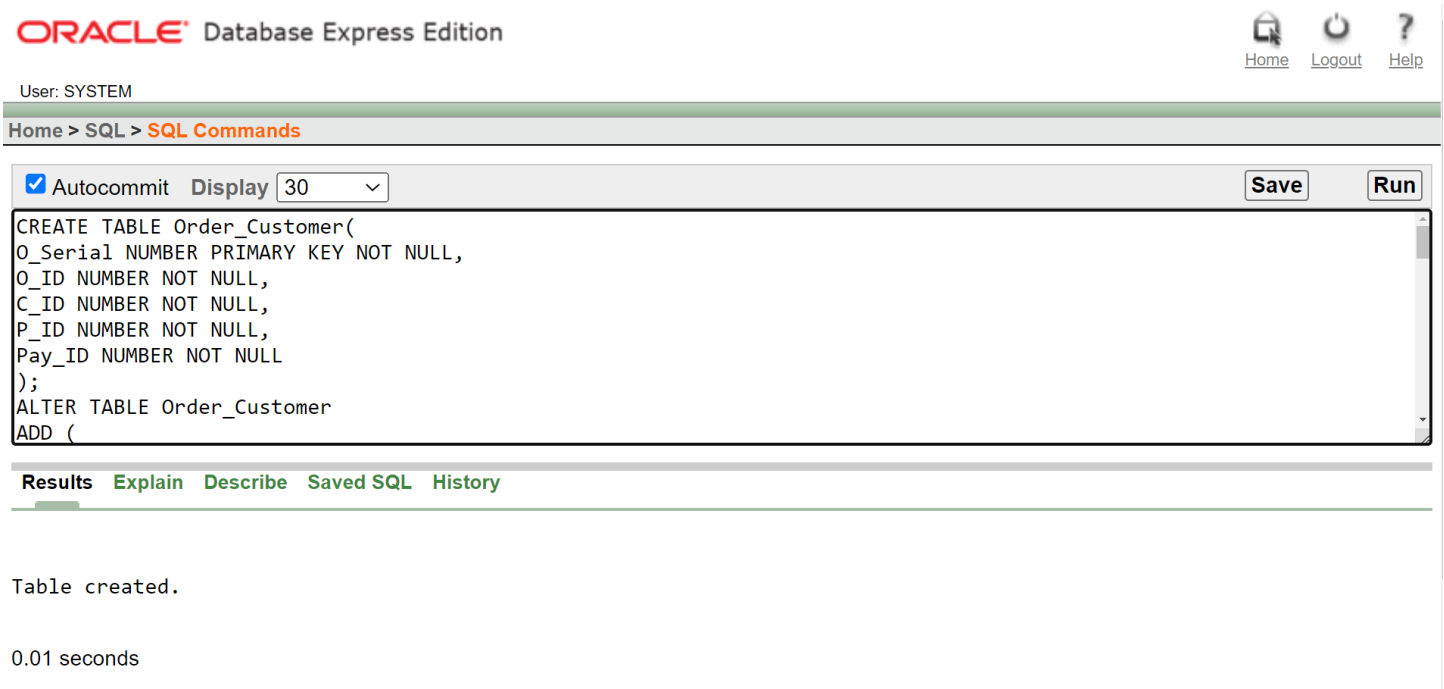
Application Express 2.1.0.00.39

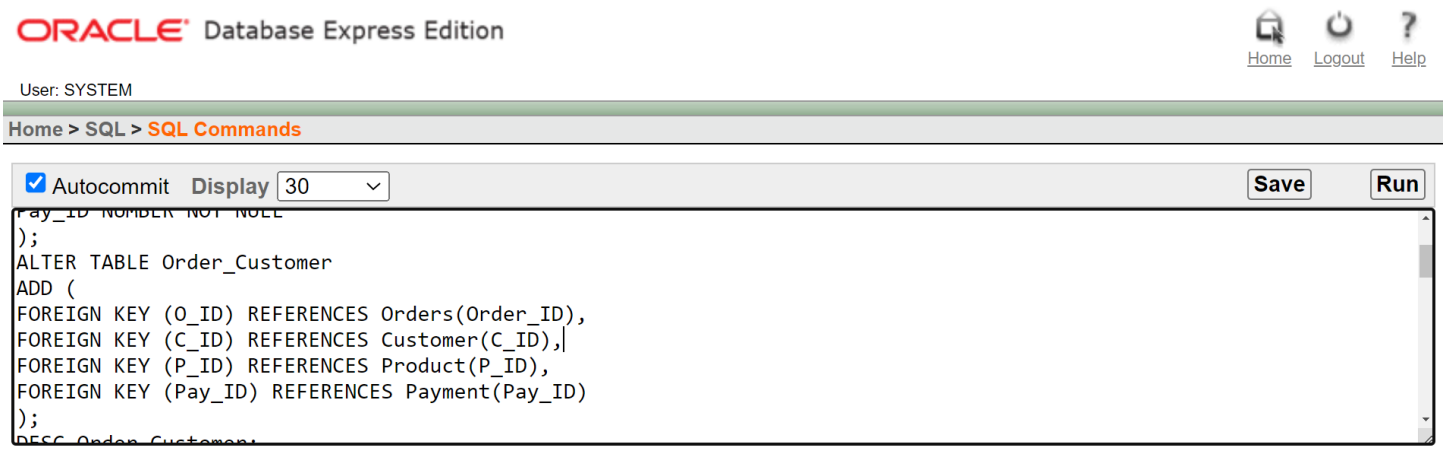
ORDER_ID	ORDER_ADDRESS	ORDER_DATE	ORDER_AMOUNT
101	MIRPUR-1	10-MAY-24	4000
102	DHAKA	10-MAY-24	3000
103	DHANMONDI	11-MAY-24	13000
104	MADARIPUR	11-MAY-24	4000
105	CUMILLA	11-MAY-24	6000
106	KURIL	19-MAY-24	3000
107	PURANBAZAR	12-MAY-24	7000
108	MIRPUR	12-MAY-24	5000
109	BASHUNDHARA	13-MAY-24	1000
110	UTTARA	13-MAY-24	2000
111	PURANBAZAR	13-MAY-24	6000
112	BADAMTOLA	13-MAY-24	3000
113	KHULNA	13-MAY-24	5000
114	MIRPUR15	13-MAY-24	1000
115	KHULNA	14-MAY-24	3000
116	MIRPUR15	14-MAY-24	1000
118	MILONPUR	15-MAY-24	3000
119	LANGLABAD	15-MAY-24	5000
120	KULPODDI	16-MAY-24	8000
121	DHAMAL	16-MAY-24	2000
122	GULSHAN	17-MAY-24	3000
123	BADDA	17-MAY-24	5000
124	AMIRABAD	17-MAY-24	2500
125	COXSBAZAR	18-MAY-24	1500
126	PURANROAD	18-MAY-24	2000
127	PURANSOR	19-MAY-24	1000
128	MIRPUR2	19-MAY-24	4000
129	BAGURA	19-MAY-24	3000
130	BADDA	23-MAY-24	4500
117	BARISHAL	14-MAY-24	3000

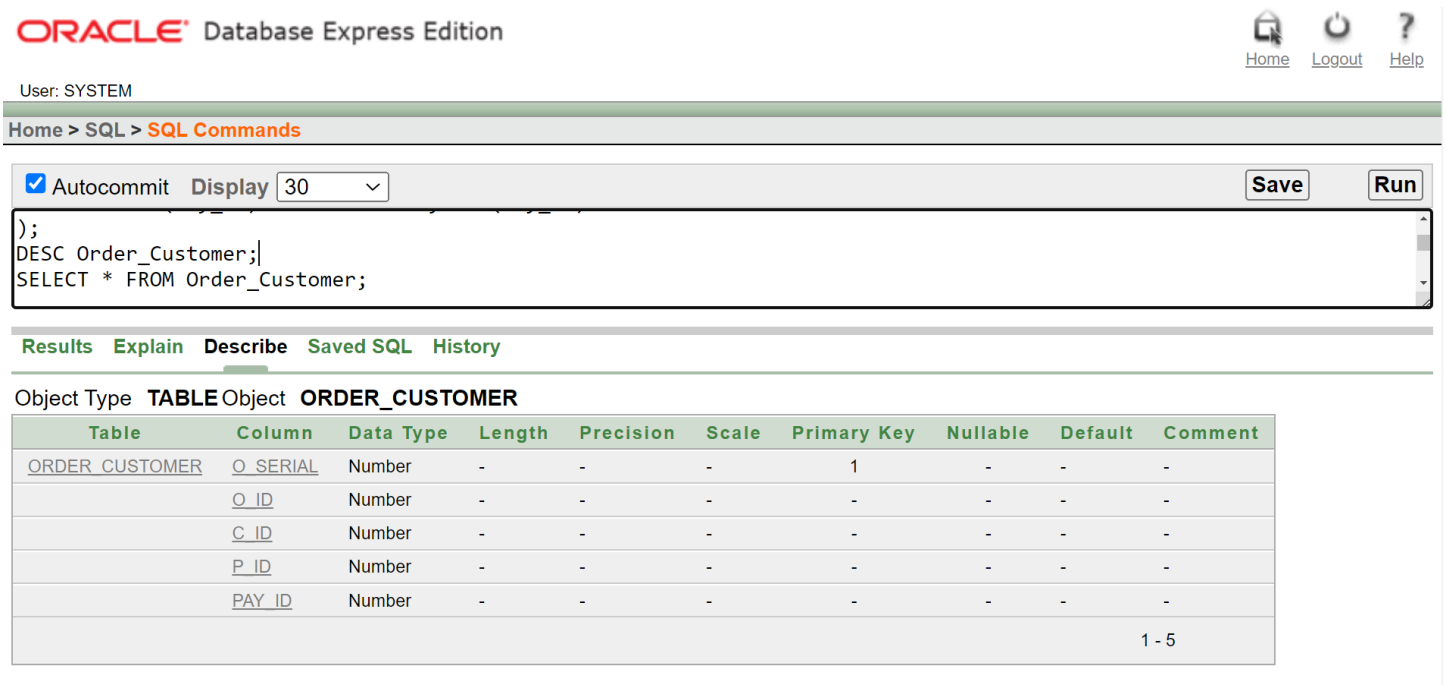
30 rows returned in 0.00 seconds CSV Export

Table Name: Order_Customer

```
CREATE TABLE Order_Customer(
O_Serial NUMBER PRIMARY KEY NOT NULL,
O_ID NUMBER NOT NULL,
C_ID NUMBER NOT NULL,
P_ID NUMBER NOT NULL,
Pay_ID NUMBER NOT NULL
);
ALTER TABLE Order_Customer
ADD (
FOREIGN KEY (O_ID) REFERENCES Orders(Order_ID),
FOREIGN KEY (C_ID) REFERENCES Customer(C_ID),
FOREIGN KEY (P_ID) REFERENCES Product(P_ID),
FOREIGN KEY (Pay_ID) REFERENCES Payment(Pay_ID)
);
DESC Order_Customer;
SELECT * FROM Order_Customer;
```







INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (1, 101, 10, 1005, 10001);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (2, 105, 22, 1005, 10008);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (3, 104, 11, 1006, 10002);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (4, 104, 11, 1005, 10002);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (5, 102, 20, 1007, 10004);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (6, 102, 20, 1008, 10004);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (7, 110, 9, 1002, 10009);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (8, 110, 9, 1010, 10009);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (9, 110, 9, 1009, 10009);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (10, 111, 11, 1011, 10011);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (11, 111, 11, 1012, 10011);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (12, 112, 12, 1012, 10012);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (13, 112, 12, 1013, 10012);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (14, 112, 12, 1014, 10012);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (15, 115, 15, 1015, 10015);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (16, 116, 16, 1016, 10016);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (17, 114, 17, 1017, 10017);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (18, 118, 18, 1018, 10018);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (19, 119, 19, 1019, 10019);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (20, 120, 20, 1020, 10020);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (21, 121, 21, 1021, 10021);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (22, 122, 22, 1022, 10022);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (23, 123, 23, 1023, 10023);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (24, 124, 24, 1024, 10024);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (25, 125, 25, 1025, 10025);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (26, 126, 26, 1026, 10026);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (27, 127, 27, 1027, 10027);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (28, 128, 28, 1028, 10028);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (29, 129, 29, 1029, 10029);

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (30, 130, 30, 1030, 10030);

ORACLE Database Express Edition

Home Logout Help

User: SYSTEM

Home > SQL > SQL Commands

Autocommit Display 30 Save Run

```
ADD (
FOREIGN KEY (O_ID) REFERENCES Orders(Order_ID),
FOREIGN KEY (C_ID) REFERENCES Customer(C_ID),
FOREIGN KEY (P_ID) REFERENCES Product(P_ID),
FOREIGN KEY (Pay_ID) REFERENCES Payment(Pay_ID)
);
DESC Order_Customer;
SELECT * FROM Order_Customer;

INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (1, 101, 10, 1005, 10001);
INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (2, 105, 22, 1005, 10008);
INSERT INTO Order_Customer (O_Serial, O_ID, C_ID, P_ID, Pay_ID) VALUES (3, 104, 11, 1006, 10002);
```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.00 seconds

Results Explain Describe Saved SQL History

O_SERIAL	O_ID	C_ID	P_ID	PAY_ID
1	101	10	1005	10001
2	105	22	1005	10008
3	104	11	1006	10002
4	104	11	1005	10002
5	102	20	1007	10004
6	102	20	1008	10004
7	110	9	1002	10009
8	110	9	1010	10009
9	110	9	1009	10009
10	111	11	1011	10011
11	111	11	1012	10011
12	112	12	1012	10012
13	112	12	1013	10012
14	112	12	1014	10012
15	115	15	1015	10015
16	116	16	1016	10016
17	114	17	1017	10017
18	118	18	1018	10018
19	119	19	1019	10019
20	120	20	1020	10020
21	121	21	1021	10021
22	122	22	1022	10022
23	123	23	1023	10023
24	124	24	1024	10024
25	125	25	1025	10025
26	126	26	1026	10026
27	127	27	1027	10027
28	128	28	1028	10028
29	129	29	1029	10029
30	130	30	1030	10030

30 rows returned in 0.00 seconds CSV Export

Single-Row Functions:

Q. Create a query that retrieves customer IDs and their corresponding names in uppercase from the 'Customer' table.

SQL: SELECT C_ID, UPPER(C_NAME) AS upper_name from Customer;

C_ID	UPPER_NAME
1	ABIR
2	ASIF
3	AMAN
4	AKIB
5	ABID
6	ARIF
7	ANIM
8	ABDULLAH
9	BASIR
10	BARIK
11	BARNO
12	TOMA
13	RIYA
14	MILA
15	TANISHA
16	ISHA
17	ZARIN
18	AYMAN
19	PAPIYA
20	TAPASH
21	BIR
22	ASIFA
23	AMAN
24	AKIBA
25	ABIDA
26	ARIFA
27	ANIMA
28	ANKIT
29	BABUL
30	MITHUN

Group Functions:

Q. Write a Query that calculates the total order amount for each order date in the 'orders' table.

Relational Algebra: γOrder_Date,SUM(Order_Amount) as Total_Amount(Orders)

SQL: Select Order_Date, SUM (order_amount) as total_amount from Orders group by Order_Date;

ORDER_DATE	TOTAL_AMOUNT
10-MAY-24	700.6
23-MAY-24	450
19-MAY-24	1450.33
13-MAY-24	3502.19
14-MAY-24	1000.55
12-MAY-24	1200.76
15-MAY-24	1400
17-MAY-24	1051.51
11-MAY-24	1132.07
18-MAY-24	550
16-MAY-24	1000.35

Q. Write a query to count the total number of orders placed on a specific date, such as May 11th, 2024, from the 'orders' table?"

Relational Algebra: γOrder_Date,COUNT(*) as Total_Orders(σOrder_Date='11−MAY−24'(Orders))

SQL: SELECT Order_Date, COUNT(*) AS total_orders FROM Orders WHERE Order_Date = '11-MAY-24' GROUP BY Order_Date;

ORDER_DATE	TOTAL_ORDERS
11-MAY-24	3

1 rows returned in 0.01 seconds C

View Table:

Q. Create a view table to find Order_Customer details.

Relational Algebra:

ρOrder_Customer_View(O_Serial,O_ID,C_ID,P_ID,Pay_ID,C_Name)(σOrder_Customer.C_ID=Customer.C_ID(Order_Customer⋈Customer))

SQL: CREATE VIEW Order_Customer_View AS SELECT OC.O_Serial, OC.O_ID, OC.C_ID, OC.P_ID, OC.Pay_ID, C.C_Name FROM Order_Customer OC JOIN Customer C ON OC.C_ID = C.C_ID;

SELECT * FROM Order_Customer_View;

O_SERIAL	O_ID	C_ID	P_ID	PAY_ID	C_NAME
9	110	9	1009	10009	Basir
8	110	9	1010	10009	Basir
7	110	9	1002	10009	Basir
1	101	10	1005	10001	Barik
11	111	11	1012	10011	Barno
10	111	11	1011	10011	Barno
4	104	11	1005	10002	Barno
3	104	11	1006	10002	Barno
14	112	12	1014	10012	Toma
13	112	12	1013	10012	Toma
12	112	12	1012	10012	Toma
15	115	15	1015	10015	Tanisha
16	116	16	1016	10016	Isha
17	114	17	1017	10017	Zarin
18	118	18	1018	10018	Ayman
19	119	19	1019	10019	Papiya
20	120	20	1020	10020	Tapash
6	102	20	1008	10004	Tapash
5	102	20	1007	10004	Tapash
21	121	21	1021	10021	Bir
22	122	22	1022	10022	Asifa
2	105	22	1005	10008	Asifa
23	123	23	1023	10023	Aman
24	124	24	1024	10024	Akiba
25	125	25	1025	10025	Abida
26	126	26	1026	10026	Arifa
27	127	27	1027	10027	Anima
28	128	28	1028	10028	Ankit
29	129	29	1029	10029	Babul
30	130	30	1030	10030	Mithun

Q. Create a view table To find customer full details and contact information

Relational algebra:

Customer_Details_View←ρCustomer_Details_View(C_ID,C_Name,C_Address,C_Email,C_Phone)(σCustomer.C_ID=Customer_Contact.Customer_ID(Customer⋈Customer_Contact))

SQL: CREATE VIEW Customer_Details_View AS SELECT C.C_ID, C.C_Name, C.C_Address, C.C_Email, CC.C_Phone FROM Customer C JOIN Customer_Contact CC ON C.C_ID = CC.Customer_ID;

SELECT * FROM Customer_Details_View;

C_ID	C_NAME	C_ADDRESS	C_EMAIL	C_PHONE
1	Abir	Mirpur-1	abir@gmail.com	0198557567
2	Asif	Malibag	asif@gmail.com	0158557567
3	Aman	Dhanmondi	aman@gmail.com	0168557567
4	Akib	Madaripur	akib@gmail.com	0178557567
5	Abid	Cumilla	abid@gmail.com	0148557567
6	Arif	Kuratoli	arif@gmail.com	01234567890
7	Anim	Wari	anim@gmail.com	01234567430
8	Abdullah	Mirpur-10	abdullah@gmail.com	01934567430
9	Basir	Tongi	basir@gmail.com	01234567810
10	Barik	Puran Dhaka	barik@gmail.com	0198557597
11	Barno	Jigatola	barno@gmail.com	01685578870
12	Toma	Azimpur	toma@gmail.com	0168567567
13	Riya	Sitakunda	riya@gmail.com	01489857567
14	Mila	Shahbag	mila@gmail.com	01234547810
15	Tanisha	Green Road	Tanisha@gmail.com	01985578597
16	Isha	Mohammodpur	Isha@gmail.com	01685578880
17	Zarin	Kamrangirchor	zarin@gmail.com	01168567567
18	Ayman	Monipur	Ayman@gmail.com	01489857569
19	Papiya	Lalbag	Papiya@gmail.com	01266567810
20	Tapash	Farngate	Tapash@gmail.com	01988857597
21	Bir	Demra	bir@gmail.com	01684578870
22	Asifa	Gulshan	asifa@gmail.com	01685617467
23	Aman	Badda	aman@egmail.com	01489857877
24	Akiba	Amirabad	akib@gmail.com	01234543100
25	Abida	Kuril	abid@gmail.com	01234547910
26	Arifa	Bongobazar	arifa@gmail.com	01234543456
27	Anima	Pathapath	anima@gmail.com	01245647910
28	Ankit	Pallabi	ankit@gmail.com	01231237910
29	Babul	Uttara	babul@gmail.com	01298767910
30	Mithun	Gazipur	mithun@gmail.com	01239876510

Joins:

Q. Write a query that retrieves the maximum customer ID, the total count of customers, and the count of unique phone numbers from the 'Customer' and 'Customer_Contact' tables.

Relational algebra: π Max_Customer_ID,Total_Customers,Unique_Phone_Numbers(ρ

Customer_Contact(Customer⋈ Customer.C_ID=Customer_Contact.Customer_ID Customer_Contact))

SQL: SELECT MAX(C_ID) AS Max_Customer_ID, COUNT(*) AS Total_Customers, COUNT(DISTINCT C_Phone) AS Unique_Phone_Numbers FROM Customer JOIN Customer_Contact ON Customer.C_ID = Customer_Contact.Customer_ID;

MAX_CUSTOMER_ID	TOTAL_CUSTOMERS	UNIQUE_PHONE_NUMBERS
30	30	30

1 rows returned in 0.00 seconds CSV Export

Q. Create a query that retrieves payment details including serial number, payment method, payment ID, date, and amount for cash payments from the 'Payment_Details' and 'Payment' tables.

Relational algebra:

ρPayment_Cash_View(Pay_Serial,Pay_Method,Pay_ID,Pay_Date,Pay_Amount)(σPayment_Details.Pay_Method = ’CASH’(Payment_Details⋈Payment))

SQL: SELECT Payment_Details.Pay_Serial, Payment_Details.Pay_Method, Payment.Pay_ID, Payment.Pay_Date, Payment.Pay_Amount FROM Payment_Details INNER JOIN Payment ON Payment_Details.Payment_ID = Payment.Pay_ID WHERE Payment_Details.Pay_Method = 'CASH';

PAY_SERIAL	PAY_METHOD	PAY_ID	PAY_DATE	PAY_AMOUNT
1	CASH	10001	15-MAY-24	5400
3	CASH	10003	17-MAY-24	5500
5	CASH	10005	19-MAY-24	6400
7	CASH	10007	21-MAY-24	4000
9	CASH	10009	23-MAY-24	5500
11	CASH	10011	25-MAY-24	5400
13	CASH	10013	27-MAY-24	5500
15	CASH	10015	29-MAY-24	6400
17	CASH	10017	02-APR-24	4000
19	CASH	10019	04-APR-24	5500
21	CASH	10021	25-APR-24	6400
23	CASH	10023	27-APR-24	4000
25	CASH	10025	01-MAR-24	5000
27	CASH	10027	03-MAR-24	4000
29	CASH	10029	05-MAR-24	5000

Q. Write a query to retrieves information about orders, customers, and products where the product category is 'VEGETABLE".

Relational Algebra:

ρVegetable_Orders_View(O_Serial,O_ID,C_ID,P_ID,Pay_ID,Order_Address,Order_Date,Order_Amount,C_Name,C_Address,C_Email,P_Name,P_Category,P_Quantity,P_Price,P_Availability)(σP.P_Category='VEGETABLE'(OC⋈(O⋈(C⋈P))))

SQL: SELECT OC.O_Serial, OC.O_ID, OC.C_ID, OC.P_ID, OC.Pay_ID, O.Order_Address, O.Order_Date, O.Order_Amount, C.C_Name, C.C_Address, C.C_Email, P.P_Name, P.P_Category, P.P_Quantity, P.P_Price, P.P_Availability FROM Order_Customer OC

LEFT JOIN Orders O ON OC.O_ID = O.Order_ID

LEFT JOIN Customer C ON OC.C_ID = C.C_ID

LEFT JOIN Product P ON OC.P_ID = P.P_ID

WHERE P.P_Category = 'VEGETABLE';

O_SERIAL	O_ID	C_ID	P_ID	PAY_ID	ORDER_ADDRESS	ORDER_DATE	ORDER_AMOUNT	C_NAME	C_ADDRESS	C_EMAIL	P_NAME	P_CATEGORY	P_QUANTITY	P_PRICE	P_AVAILABILITY
7	110	9	1002	10009	UTTARA	13-MAY-24	600	Basir	Tongi	basir@gmail.com	TOMATO	VEGETABLE	0	70 PER KG	N
8	110	9	1010	10009	UTTARA	13-MAY-24	600	Basir	Tongi	basir@gmail.com	RADISH	VEGETABLE	400	70 PER KG	Y
9	110	9	1009	10009	UTTARA	13-MAY-24	600	Basir	Tongi	basir@gmail.com	RED PEPPER	VEGETABLE	900	80 PER KG	Y
3	104	11	1006	10002	MADARIPUR	11-MAY-24	400.75	Barno	Jigatola	barno@gmail.com	BEAN	VEGETABLE	800	90 PER KG	Y
5	102	20	1007	10004	DHAKA	10-MAY-24	300.35	Tapash	Farngate	Tapash@gmail.com	ONION	VEGETABLE	700	60 PER KG	Y
6	102	20	1008	10004	DHAKA	10-MAY-24	300.35	Tapash	Farngate	Tapash@gmail.com	GARLIC	VEGETABLE	500	50 PER KG	Y

6 rows returned in 0.00 seconds CSV Export

Q. Write a query that retrieves a comprehensive list of customer information and contact details, including all available data from both the Customer and Customer_Contact tables.

Relational Algebra: (Customer ⋈ Customer_Contact) ∪ (Customer ⋈ Customer_Contact)

SQL: SELECT * FROM Customer c FULL OUTER JOIN Customer_Contact cc ON c.C_ID = cc.Customer_ID

C_ID	C_NAME	C_ADDRESS	C_EMAIL	C_PHONE	CUSTOMER_ID
1	Abir	Mirpur-1	abir@gmail.com	0198557567	1
2	Asif	Malibag	asif@gmail.com	0158557567	2
3	Aman	Dhanmondi	aman@gmail.com	0168557567	3
4	Akib	Madaripur	akib@gmail.com	0178557567	4
5	Abid	Cumilla	abid@gmail.com	0148557567	5
6	Arif	Kuratoli	arif@gmail.com	01234567890	6
7	Anim	Wari	anim@gmail.com	01234567430	7
8	Abdullah	Mirpur-10	abdullah@gmail.com	01934567430	8
9	Basir	Tongi	basir@gmail.com	01234567810	9
10	Barik	Puran Dhaka	barik@gmail.com	0198557597	10
11	Barno	Jigatola	barno@gmail.com	01685578870	11
12	Toma	Azimpur	toma@gmail.com	0168567567	12
13	Riya	Sitakunda	riya@gmail.com	01489857567	13
14	Mila	Shahbag	mila@gmail.com	01234547810	14
15	Tanisha	Green Road	Tanisha@gmail.com	01985578597	15
16	Isha	Mohammodpur	Isha@gmail.com	01685578880	16
17	Zarin	Kamrangirchor	zarin@gmail.com	01168567567	17
18	Ayman	Monipur	Ayman@gmail.com	01489857569	18
19	Papiya	Lalbag	Papiya@gmail.com	01266567810	19
20	Tapash	Farngate	Tapash@gmail.com	01988857597	20
21	Bir	Demra	bir@gmail.com	01684578870	21
22	Asifa	Gulshan	asifa@gmail.com	01685617467	22
23	Aman	Badda	aman@egmail.com	01489857877	23
25	Abida	Kuril	abid@gmail.com	01234547910	25
26	Arifa	Bongobazar	arifa@gmail.com	01234543456	26
27	Anima	Pathapath	anima@gmail.com	01245647910	27
28	Ankit	Pallabi	ankit@gmail.com	01231237910	28
29	Babul	Uttara	babul@gmail.com	01298767910	29
30	Mithun	Gazipur	mithun@gmail.com	01239876510	30
24	Akiba	Amirabad	akib@gmail.com	01234543100	24

Q. Write a query that shows the details of the payment and order/customer pairs where the payment ID is 10002.

Relational algebra: σ Payment.Pay_ID = 10002 (Payment × Order_Customer)

SQL: SELECT * FROM Payment CROSS JOIN Order_Customer WHERE Payment.Pay_ID = 10002;

PAY_ID	PAY_DATE	PAY_AMOUNT	O_SERIAL	O_ID	C_ID	P_ID	PAY_ID
10002	16-MAY-24	5600	1	101	10	1005	10001
10002	16-MAY-24	5600	2	105	22	1005	10008
10002	16-MAY-24	5600	3	104	11	1006	10002
10002	16-MAY-24	5600	4	104	11	1005	10002
10002	16-MAY-24	5600	5	102	20	1007	10004
10002	16-MAY-24	5600	6	102	20	1008	10004
10002	16-MAY-24	5600	7	110	9	1002	10009
10002	16-MAY-24	5600	8	110	9	1010	10009
10002	16-MAY-24	5600	9	110	9	1009	10009
10002	16-MAY-24	5600	10	111	11	1011	10011
10002	16-MAY-24	5600	11	111	11	1012	10011
10002	16-MAY-24	5600	12	112	12	1012	10012
10002	16-MAY-24	5600	13	112	12	1013	10012
10002	16-MAY-24	5600	14	112	12	1014	10012
10002	16-MAY-24	5600	15	115	15	1015	10015
10002	16-MAY-24	5600	16	116	16	1016	10016
10002	16-MAY-24	5600	17	114	17	1017	10017
10002	16-MAY-24	5600	18	118	18	1018	10018
10002	16-MAY-24	5600	19	119	19	1019	10019
10002	16-MAY-24	5600	20	120	20	1020	10020
10002	16-MAY-24	5600	21	121	21	1021	10021
10002	16-MAY-24	5600	22	122	22	1022	10022
10002	16-MAY-24	5600	23	123	23	1023	10023
10002	16-MAY-24	5600	24	124	24	1024	10024
10002	16-MAY-24	5600	25	125	25	1025	10025
10002	16-MAY-24	5600	26	126	26	1026	10026
10002	16-MAY-24	5600	27	127	27	1027	10027
10002	16-MAY-24	5600	28	128	28	1028	10028
10002	16-MAY-24	5600	29	129	29	1029	10029
10002	16-MAY-24	5600	30	130	30	1030	10030

Subquery:

Q. Create a query to retrieve all columns from the 'Order_Customer' table where the corresponding order IDs have order amounts greater than 800 in the 'Orders' table?

Relational Algebra: σOrder_Customer.O_ID∈(πOrder_ID(σOrder_Amount>800(Orders)))(Order_Customer)

SQL: SELECT * FROM Order_Customer WHERE O_ID IN (SELECT Order_ID FROM Orders WHERE Order_Amount > 800);

O_SERIAL	O_ID	C_ID	P_ID	PAY_ID
17	114	17	1017	10017
18	118	18	1018	10018

2 rows returned in 0.00 seconds CSV Export

Q. Write a query to fetch all the records from the 'Customer_Contact' table for customers whose addresses match 'Azimpur' as per the 'Customer' table.

Relational Algebra:

σCustomer_Contact.Customer_ID∈(πC_ID(σC_Address='Azimpur'(Customer)))(Customer_Contact)

SQL: SELECT * FROM Customer_Contact WHERE Customer_ID IN (SELECT C_ID FROM Customer WHERE C_Address = 'Azimpur');

C_PHONE	CUSTOMER_ID
0168567567	12

1 rows returned in 0.01 seconds

Q. Write a query that retrieves the payment details including the payment ID, date, and amount from the Payment table where the payment ID is 10014.

Relational Algebra:

π Pay_ID, Pay_Date, Pay_Amount (

σ Pay_ID=10014 ∧ ∃(π Payment_ID (ρ p (Payment) ⋈ p.Pay_ID=Payment_ID Payment_Details))

)

SQL: SELECT p.Pay_ID, p.Pay_Date, p.Pay_Amount FROM Payment p WHERE p.Pay_ID = 10014 AND EXISTS (SELECT 1 FROM Payment_Details pd WHERE pd.Payment_ID = p.Pay_ID);

Results Explain Describe Saved SQL History

PAY_ID	PAY_DATE	PAY_AMOUNT
10014	28-MAY-24	5400

1 rows returned in 0.01 seconds CSV Export

Q. Write a query that retrieves all orders from the Order_Customer table where at least one corresponding product has the P_ID 1005.

Relational Algebra: σ EXISTS(π1(σ P_ID = 1005(Product)) ⋈ P_ID = p.P_ID(Order_Customer))

SQL: SELECT * FROM Order_Customer o WHERE EXISTS (SELECT 1 FROM Product p WHERE p.P_ID = o.P_ID AND p.P_ID = 1005);

Results Explain Describe Saved SQL History

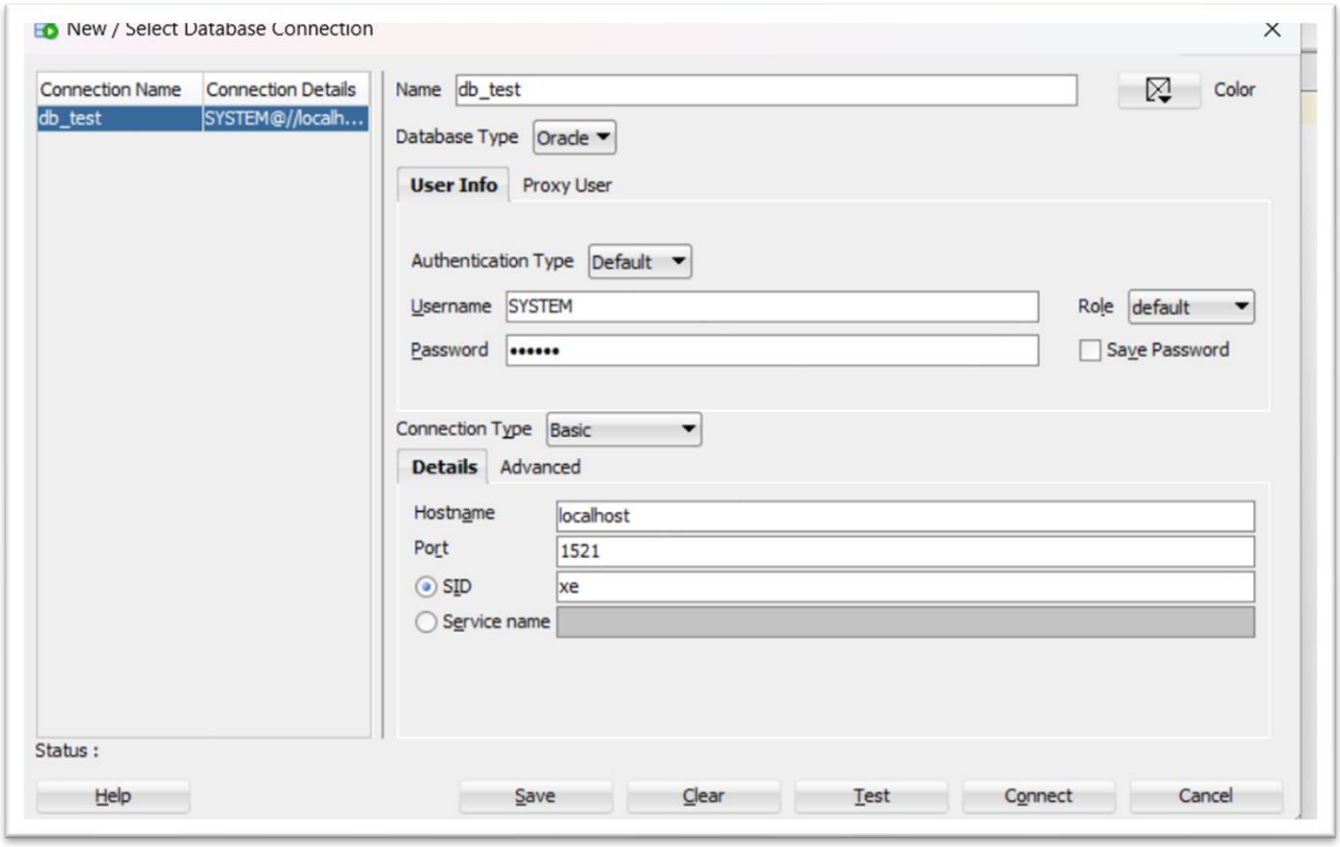
O_SERIAL	O_ID	C_ID	P_ID	PAY_ID
1	101	10	1005	10001
2	105	22	1005	10008
4	104	11	1005	10002

3 rows returned in 0.02 seconds CSV Export

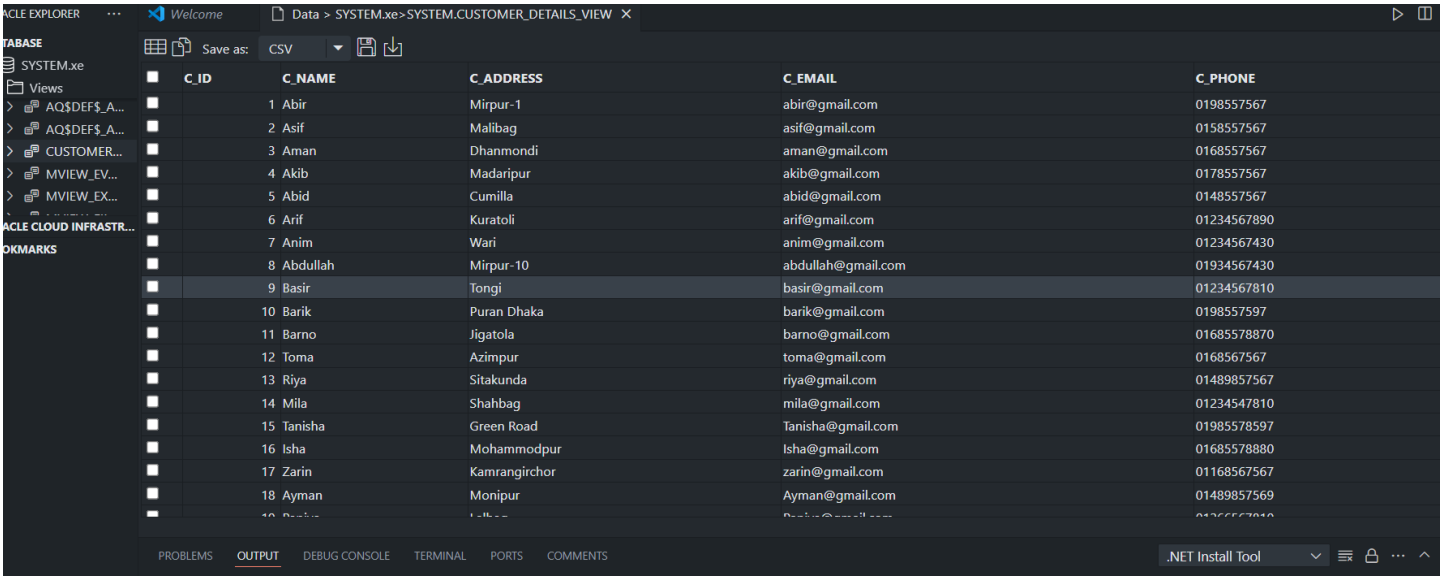
Successful DB Connection

We have used the Oracle SQL Developer to connect Database with VS Code. Oracle SQL Developer for VS Code is a free extension in Visual Studio Code, offering features that simplify database development tasks and enhance productivity. Features include browsing database objects, running SQL statements and scripts, editing and compiling PL/SQL statements, and manipulating and exporting data.

- ORACLE SQL DEVELOPER CONNECTION SETUP:



➢ CONNECTED WITH VS CODE



C_ID	C_NAME	C_ADDRESS	C_EMAIL	C_PHONE
1	Abir	Mirpur-1	abir@gmail.com	0198557567
2	Asif	Malibag	asif@gmail.com	0158557567
3	Aman	Dhanmondi	aman@gmail.com	0168557567
4	Akib	Madaripur	akib@gmail.com	0178557567
5	Abid	Cumilla	abid@gmail.com	0148557567
6	Arif	Kuratoli	arif@gmail.com	01234567890
7	Anim	Wari	anim@gmail.com	01234567430
8	Abdullah	Mirpur-10	abdullah@gmail.com	01934567430
9	Basir	Tongi	basir@gmail.com	01234567810
10	Barik	Puran Dhaka	barik@gmail.com	0198557597
11	Barno	Jigatola	barno@gmail.com	01685578870
12	Toma	Azimpur	toma@gmail.com	0168567567
13	Riya	Sitakunda	riya@gmail.com	01489857567
14	Mila	Shahbag	mila@gmail.com	01234547810
15	Tanisha	Green Road	Tanisha@gmail.com	01985578597
16	Isha	Mohammodpur	Isha@gmail.com	01685578880
17	Zarin	Kamrangirchor	zarin@gmail.com	01168567567
18	Ayman	Monipur	Ayman@gmail.com	01489857569

Target User

1. Consumers:
   - Online shoppers
   - Individuals managing grocery lists
   - Users tracking their orders

2. Grocery Store Staff:
   - Inventory managers
   - Order fulfillment teams
   - Sales tracker

Conclusion

In the grocery management system, customers can order items they want by selecting filters and specifying the quantity. Then, they pay for their order. Once paid, staff members gather the items from the shelves and hand them over to the customer. The goal of this system is to replace manual processes with computerized tools and software, making it easier to manage data and information. The software is designed to be easy for shop owners to use, helping them manage products, customers, suppliers, orders, and payments efficiently. The main aim of this system is to replace manual processes with computerized tools and software, which can store data and information more efficiently in a database. It supports different databases and programming languages, making shop operations smoother, which can lead to more profits and happier customers. Overall, the system aims to make running a grocery store simpler and more effective.