**ONLINE SHOPPING SYSTEM**

DBMS Course Project

**Bachelor of Technology**

In

**Computer Science and Engineering**

**School of Engineering and Sciences**

Submitted by

**SINDHUJA ARNEPALLI – AP22110011444**

**RIDHI GUNTUR – AP22110011467**

**RODA CHINTHAPALLI – AP22110011496**

**A picture containing text

Description automatically generated**

**SRM University–AP**

**Neerukonda, Mangalagiri, Guntur**

**Andhra Pradesh – 522 240**

**[04, 2024]**

|  |
| --- |
| **CONTENT** |
| Abstract 3 |
| Project Background 4 |
| ER Model 5 |
| Description of ER Diagram 5-6 |
| ER Model to Relational Model Mapping 7 |
| Description of Tables 7-8 |
| Creation of data in the tables up to 3rd Normal Form 9 |
| SQL Queries 9-14 |
| Creating Views for the above tables 14-15 |

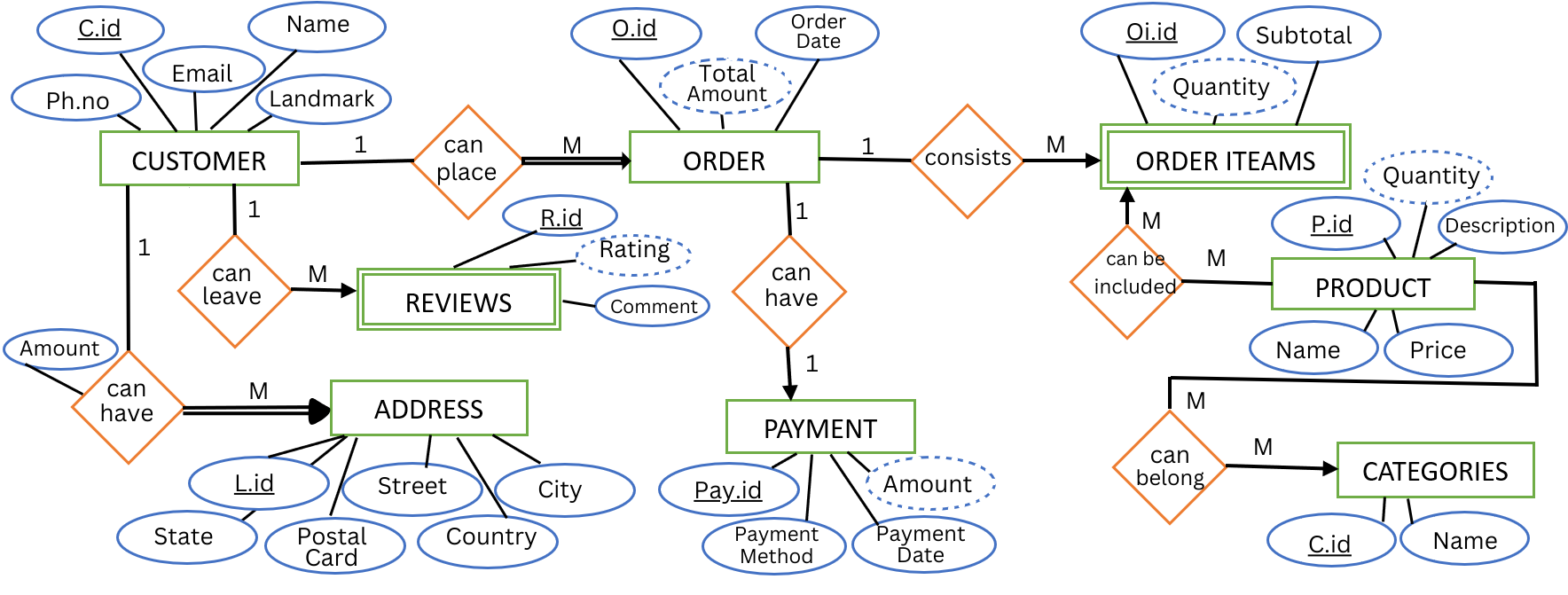
1. **ABSTRACT**

* Customers have a CustomerID (Primary Key), a name, an email, a landmark, and a phone number.
* Products have a ProductID (Primary Key), a name, a description, a price, and a quantity available
* Orders have an OrderID (Primary Key), a CustomerID (Foreign Key), an order date, and a total amount.
* Each order consists of one or more order items.
* OrderItems have an OrderItemID (Primary Key), an OrderID (Foreign Key), a ProductID (Foreign Key), a quantity, and a subtotal.
* Each order item represents a specific product included in an order, along with its quantity and subtotal.
* Payments have a PaymentID (Primary Key), an OrderID (Foreign Key), a payment date, a payment method, and an amount.
* Each order can have one payment associated with it.- Each payment is linked to one order.- Customers can have multiple addresses for shipping and billing.
* Addresses have an AddressID (Primary Key), a CustomerID (Foreign Key), a street address, a city, a state, a postal code, and a country.
* Each customer can have multiple addresses associated with their account.
* Each address is linked to one customer.
* Products can belong to one or more categories.
* Categories have a CategoryID (Primary Key) and a name.
* Each product can be assigned to one or more categories.
* Each category can have multiple products associated with it.
* Customers can leave reviews for products they have purchased.
* Reviews have a ReviewID (Primary Key), a CustomerID (Foreign Key), a ProductID (Foreign Key), a rating, and a comment.
* Each review is linked to one customer and one product.
* Each customer can leave multiple reviews for different products.

1. **PROJECT BACKGROUND**

The project sets out to construct an online shopping platform that mirrors the functionality and versatility of leading e-commerce websites. Here's a detailed overview of what the platform will offer:

1. User Registration and Authentication: Customers will have the option to create an account, providing essential details such as name, email, and phone number. Once registered, users can log in securely, allowing for a personalized shopping experience with features like order history tracking and saved addresses.
2. Product Catalog: The webpage will showcase a diverse range of products, each accompanied by a detailed description, price, and images. Users can easily browse through various categories or utilize search filters to find specific items of interest.
3. Shopping Cart and Checkout: Customers can add desired products to their shopping cart, where they can review their selections, adjust quantities, and remove items if needed. The checkout process will be straightforward, with options for multiple payment methods and secure transaction handling.
4. Order Management: Upon successful completion of a purchase, users will receive confirmation emails containing order details and estimated delivery times. Additionally, customers can track the status of their orders through their account dashboard, ensuring transparency and peace of mind.
5. Address Management: The platform will allow users to manage multiple shipping and billing addresses, enabling flexibility and convenience during the checkout process. Customers can add, edit, or delete addresses as needed, streamlining their shopping experience.
6. Product Reviews and Ratings: Users can leave feedback and ratings for products they have purchased, fostering a sense of community and trust among shoppers. Product reviews will be prominently displayed on product pages, aiding other users in making informed decisions.
7. Category Navigation: With a well-organized category structure, users can easily explore different product offerings and discover new items of interest. Each category will feature a curated selection of products, enhancing the browsing experience.
8. Responsive Design: The webpage will feature a responsive design that adapts seamlessly to various devices and screen sizes, ensuring an optimal viewing experience for users accessing the platform from desktops, laptops, tablets, or smartphones.
9. **ER MODEL**



1. **DESCRIPTION OF ER DIAGRAM**
2. Customer:

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Datatype** | **Description** |
| ***C.id*** | INT | ID of the customer |
| Name | VARCHAR | Name of the customer |
| Ph.no | INT | Phone number of the customer |
| Email | VARCHAR | Email of the customer |
| Landmark | VARCHAR | Landmark of the customer |

1. Order:

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Datatype** | **Description** |
| ***O.id*** | INT | ID number of the product placed |
| Order Date | VARCHAR | Date of the order placed |
| Total Amount | INT | Total amount of the order |

1. Order Items:

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Datatype** | **Description** |
| ***Oi.id*** | INT | ID number of the order placed |
| Quantity | INT | Number of items placed |
| Subtotal | INT | Total amount of all the products |

1. Product:

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Datatype** | **Description** |
| ***P.id*** | INT | ID of the particular product |
| Name | VARCHAR | Name of the particular product |
| Price | INT | Price of the particular product |
| Quantity | INT | Quantity of the particular product |
| Description | VARCHAR | Description of the particular product |

1. Categories:

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Datatype** | **Description** |
| ***Cat.id*** | INT | Product belongs to particular category id |
| Name | VARCHAR | Name of the category |

1. Payment:

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Datatype** | **Description** |
| ***Pay.id*** | INT | Payment ID of the order |
| Payment Method | VARCHAR | Payment method of the order |
| Payment Date | VARCHAR | Payment date of the order |
| Amount | INT | Amount of the order |

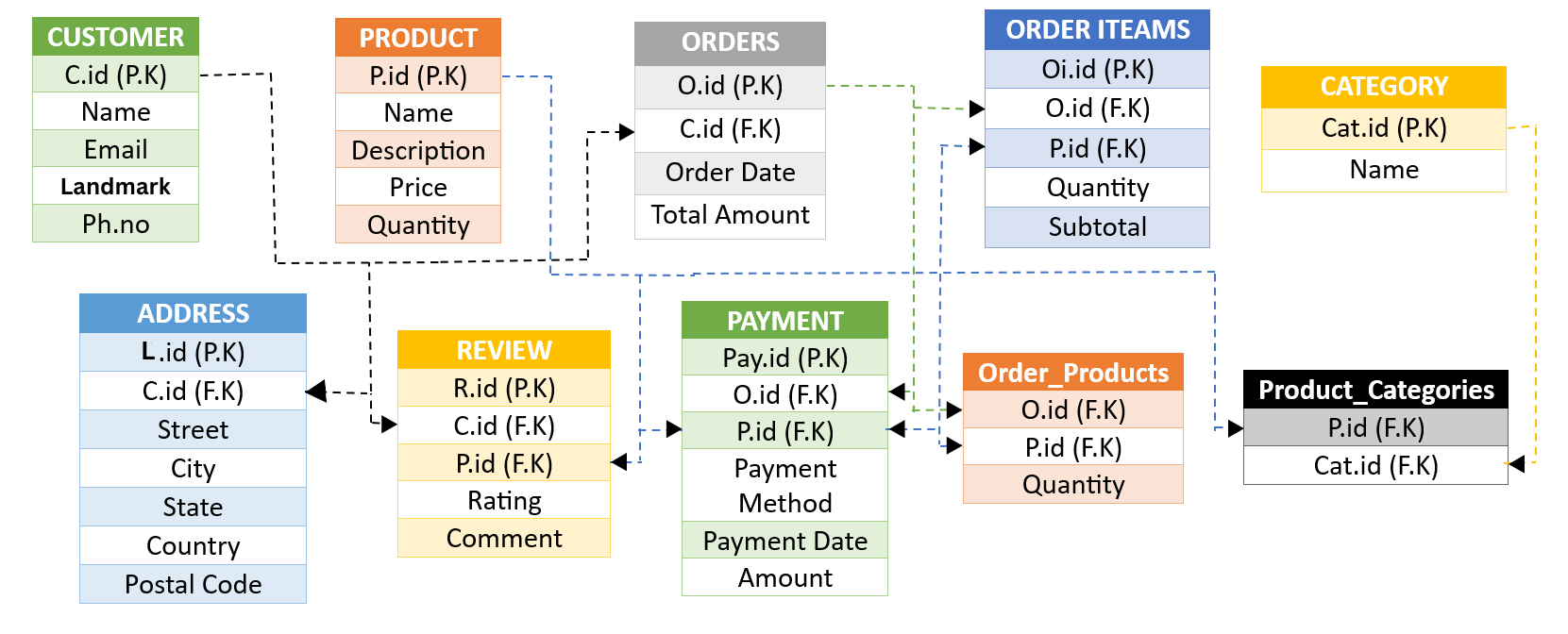
1. Address:

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Datatype** | **Description** |
| ***L.id*** | INT | Landmark id of the customer |
| Street | VARCHAR | Street of the customer |
| City | VARCHAR | City of the customer |
| State | VARCHAR | State of the customer |
| Country | VARCHAR | Country of the customer |
| Postal code | INT | Postal code of the customer |

1. Reviews

|  |  |  |
| --- | --- | --- |
| **Attributes** | **Datatype** | **Description** |
| ***R.id*** | INT | ID of the review |
| Rating | VARCHAR | Ratings given to the review |
| Comment | VARCHAR | Comments that are added to the product |

1. **ER MODEL TO RELATIONAL MODEL MAPPING**

****

1. **DESCRIPTION OF TABLES**
2. Strong Entities

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| C.id | Name | Email | Landmark | Ph.no |

1. Customer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P.id | Name | Description | Price | Quantity |

1. Products
2. Orders

|  |  |  |  |
| --- | --- | --- | --- |
| O.id | C.id (F.k) | Order Date | Total Amount |

1. Payment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Pay.id | O.id (F.K) | Amount | Payment Date | Payment Method |

1. Category

|  |  |
| --- | --- |
| Cat.id | Name |

1. Weak Entities
2. Order Items

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Oi.id | O.id (F.K) | P.id (F.K) | Quantity | Subtotal |

1. Review

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| R.id | C.id (F.K) | P.id (F.K) | Rating | Comment |

1. Mapping Relationships and Cardinalities

|  |  |  |  |
| --- | --- | --- | --- |
| **Relationship** | **Entities** | **Cardinalities** | **Participation** |
| consists | Order-Order Item | 1-M | Partial |
| Can have | Order- Payment | 1-1 | Partial |
| Can have | Customer-Address | 1-M | Full |
| Can belong | Product-Category | M-M | Partial |
| Can leave | Customer-Review | 1-M | Partial |
| Can place | Customer-Order | 1-M | Full |
| Can be included | Product-Order Item | M-M | Partial |

1. Mapping multivalued attributes

No multi values attributes

1. Mapping N-array Entities

No N-array entities

1. **CREATION OF DATA IN THE TABLES UPTO 3RD NORMAL FROM**

**Customers**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **C.id** | **Name** | **Email** | **Landmark** | **Ph.no** |

**Product**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **P.id** | **Name** | **Description** | **Price** | **Quantity** |

**Orders**

|  |  |  |  |
| --- | --- | --- | --- |
| **O.id** | **C.id** | **Order date** | **Total amount** |

**Order items**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Oi.id** | **O.id** | **P.id** | **Quantity** | **Subtotal** |

**Category**

|  |  |
| --- | --- |
| **Cat.id** | **Name** |

**Address**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **A.id** | **C.id** | **Street** | **City** | **State** | **Country** | **Postal code** |

**Review**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **R.id** | **C.id** | **P.id** | **Rating** | **Comment** |

**Payment**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Pay.id** | **O.id** | **P.id** | **Payment method** | **Payment date** | **Amount** |

All the entities are in 1st,2ND and 3rd Normalization Form.

It primarily focuses on ensuring that each attribute within a relation contains only atomic values, no partial functional dependency and no transitive dependency between non-key attributes.

1. **SQL QUERIES**
2. Creating tables

*-- Create CUSTOMER table*

CREATE TABLE CUSTOMER (

C\_id INT PRIMARY KEY,

Name VARCHAR(255),

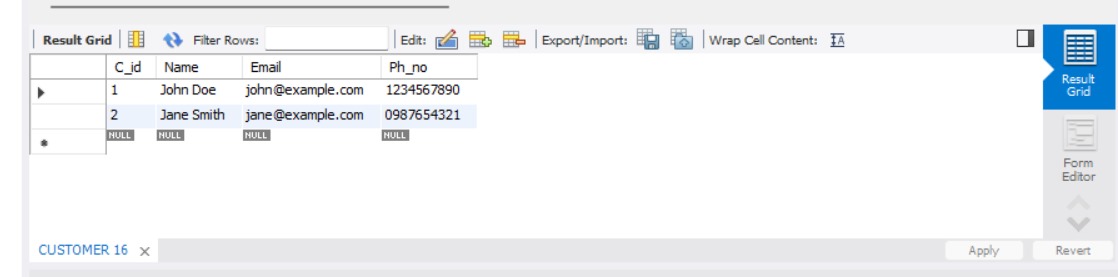
Email VARCHAR(255),

Ph\_no VARCHAR(20)

);

*-- Show contents of CUSTOMER table*

SELECT \* FROM CUSTOMER;



*-- Create PRODUCT table*

CREATE TABLE PRODUCT (

P\_id INT PRIMARY KEY,

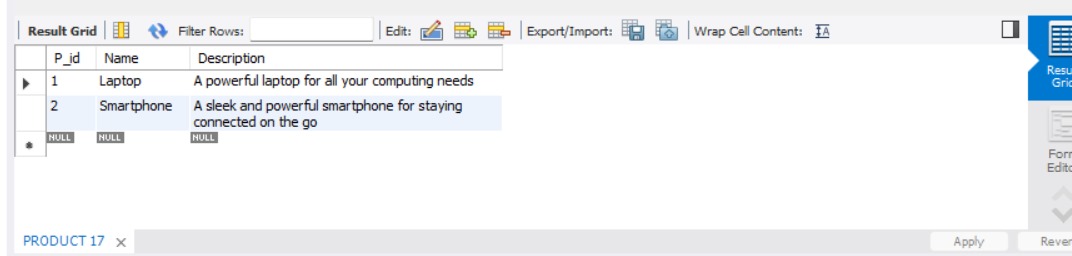
Name VARCHAR(255),

Description TEXT

);

*-- Show contents of PRODUCT table*

SELECT \* FROM PRODUCT;



*-- Create CATEGORY table*

CREATE TABLE CATEGORY (

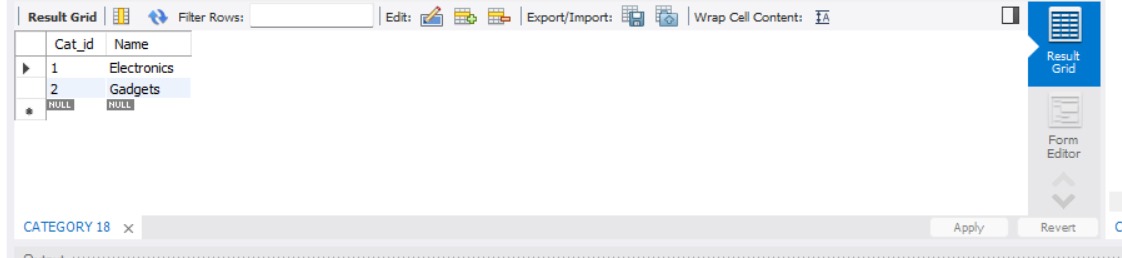
Cat\_id INT PRIMARY KEY,

Name VARCHAR(255)

);

*-- Show contents of CATEGORY table*

SELECT \* FROM CATEGORY;



*-- Create ORDERS table*

CREATE TABLE ORDERS (

O\_id INT PRIMARY KEY,

C\_id INT,

Order\_Date DATE,

Total\_Amount DECIMAL(10,2),

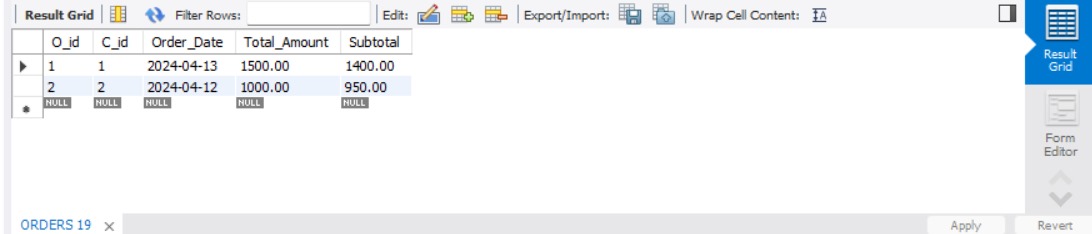
Subtotal DECIMAL(10,2),

FOREIGN KEY (C\_id) REFERENCES CUSTOMER(C\_id)

);

*-- Show contents of ORDERS table*

SELECT \* FROM ORDERS;



*-- Create ORDER\_ITEMS table*

CREATE TABLE ORDER\_ITEMS (

Oi\_id INT PRIMARY KEY,

O\_id INT,

P\_id INT,

Quantity INT,

Price DECIMAL(10,2),

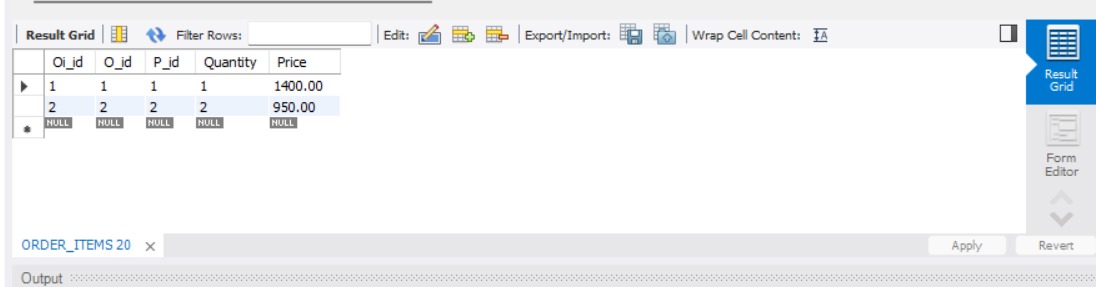
FOREIGN KEY (O\_id) REFERENCES ORDERS(O\_id),

FOREIGN KEY (P\_id) REFERENCES PRODUCT(P\_id)

);

*-- Show contents of ORDER\_ITEMS table*

SELECT \* FROM ORDER\_ITEMS;



*-- Create ADDRESS table*

CREATE TABLE ADDRESS (

A\_id INT PRIMARY KEY,

C\_id INT,

Street VARCHAR(255),

City VARCHAR(100),

State VARCHAR(100),

Country VARCHAR(100),

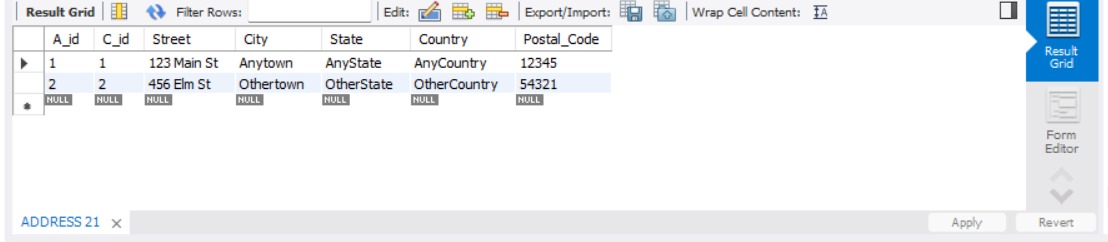
Postal\_Code VARCHAR(20),

FOREIGN KEY (C\_id) REFERENCES CUSTOMER(C\_id)

);

*-- Show contents of ADDRESS table*

SELECT \* FROM ADDRESS;



*-- Create PAYMENT table*

CREATE TABLE PAYMENT (

Pay\_id INT PRIMARY KEY,

O\_id INT,

Amount DECIMAL(10,2),

Method VARCHAR(100),

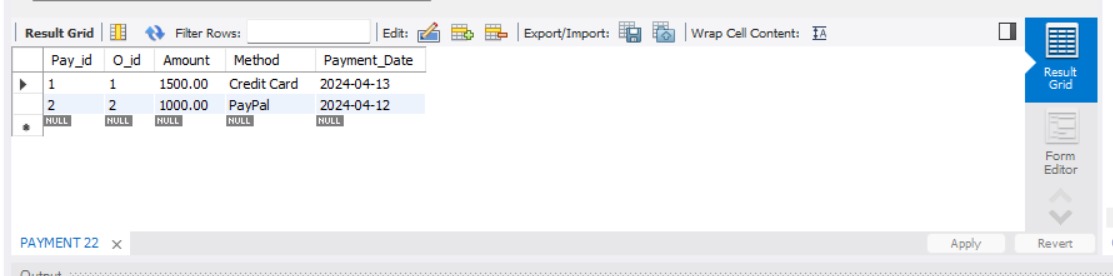
Payment\_Date DATE,

FOREIGN KEY (O\_id) REFERENCES ORDERS(O\_id)

);

*-- Show contents of PAYMENT table*

SELECT \* FROM PAYMENT;



*-- Create REVIEW table*

CREATE TABLE REVIEW (

R\_id INT PRIMARY KEY,

C\_id INT,

O\_id INT,

P\_id INT,

Rating INT,

Comment TEXT,

FOREIGN KEY (C\_id) REFERENCES CUSTOMER(C\_id),

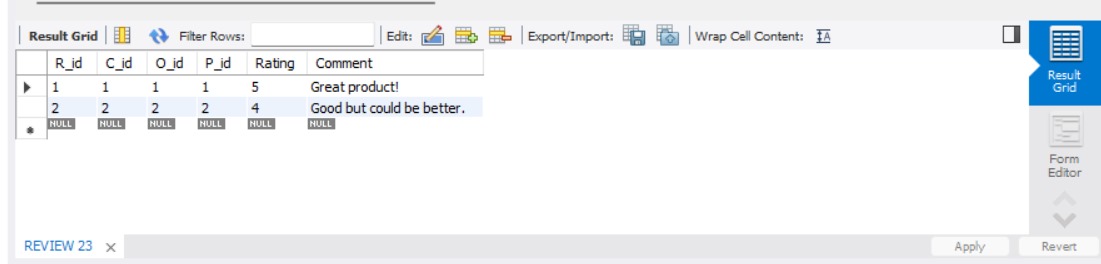
FOREIGN KEY (O\_id) REFERENCES ORDERS(O\_id),

FOREIGN KEY (P\_id) REFERENCES PRODUCT(P\_id)

);

*-- Show contents of REVIEW table*

SELECT \* FROM REVIEW;



*-- Create PRODUCT\_CATEGORIES junction table*

CREATE TABLE PRODUCT\_CATEGORIES (

P\_id INT,

Cat\_id INT,

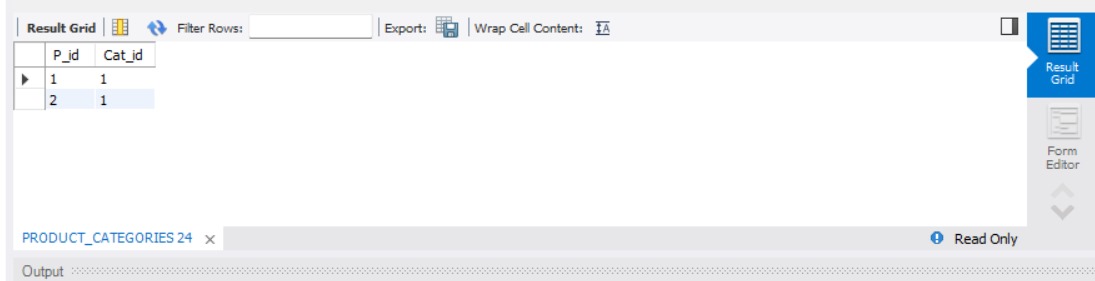
FOREIGN KEY (P\_id) REFERENCES PRODUCT(P\_id),

FOREIGN KEY (Cat\_id) REFERENCES CATEGORY(Cat\_id)

);

*-- Show contents of PRODUCT\_CATEGORIES table*

SELECT \* FROM PRODUCT\_CATEGORIES;



*-- Create ORDER\_PRODUCTS table*

CREATE TABLE ORDER\_PRODUCTS (

OrderProductID INT PRIMARY KEY,

OrderID INT,

ProductID INT,

Quantity INT,

Subtotal DECIMAL(10, 2),

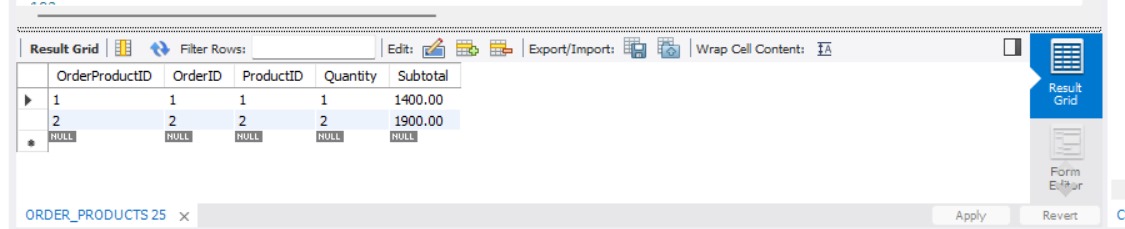
FOREIGN KEY (OrderID) REFERENCES ORDERS(O\_id),

FOREIGN KEY (ProductID) REFERENCES PRODUCT(P\_id)

);

*-- Show contents of ORDER\_PRODUCTS table*

SELECT \* FROM ORDER\_PRODUCTS;

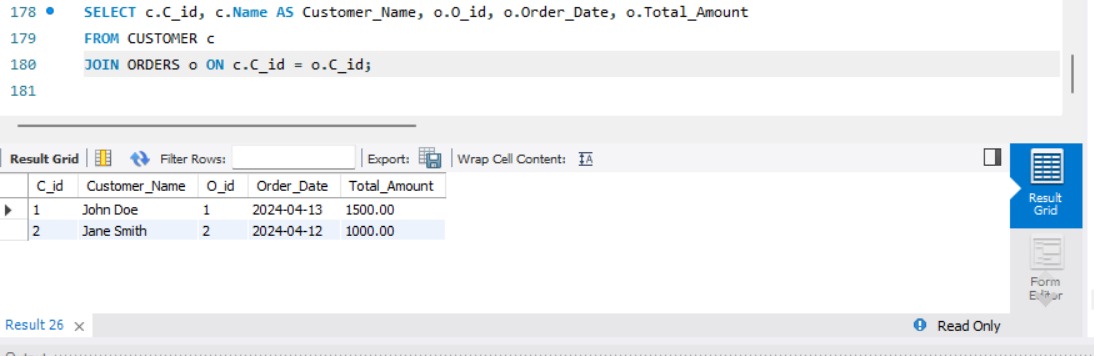


1. APPLYING JOINS

SELECT c.C\_id, c.Name AS Customer\_Name, o.O\_id, o.Order\_Date, o.Total\_Amount

FROM CUSTOMER c

JOIN ORDERS o ON c.C\_id = o.C\_id;



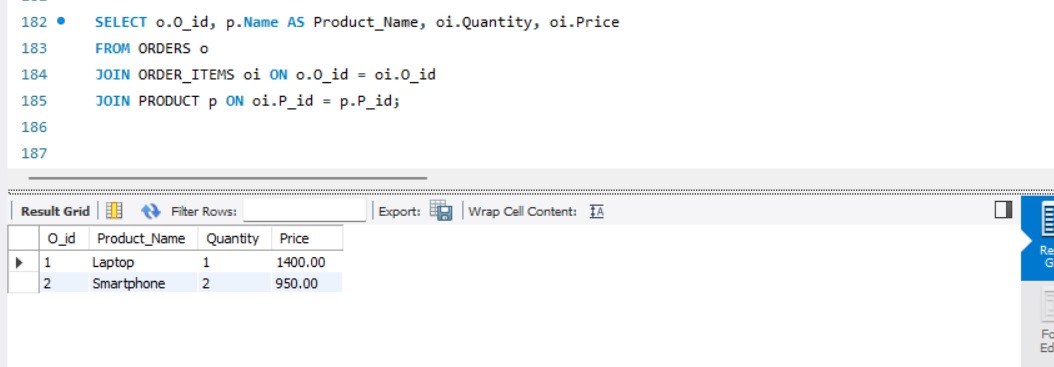
1. APPLYING DOUBLE JOINS

SELECT o.O\_id, p.Name AS Product\_Name, oi.Quantity, oi.Price

FROM ORDERS o

JOIN ORDER\_ITEMS oi ON o.O\_id = oi.O\_id

JOIN PRODUCT p ON oi.P\_id = p.P\_id;



1. **CREATING VIEWS FOR THE ABOVE TABLES**

CREATE VIEW CustOrders AS

SELECT c.C\_id, c.Name AS Customer\_Name, o.O\_id, o.Order\_Date, o.Total\_Amount

FROM CUSTOMER c

JOIN ORDERS o ON c.C\_id = o.C\_id;

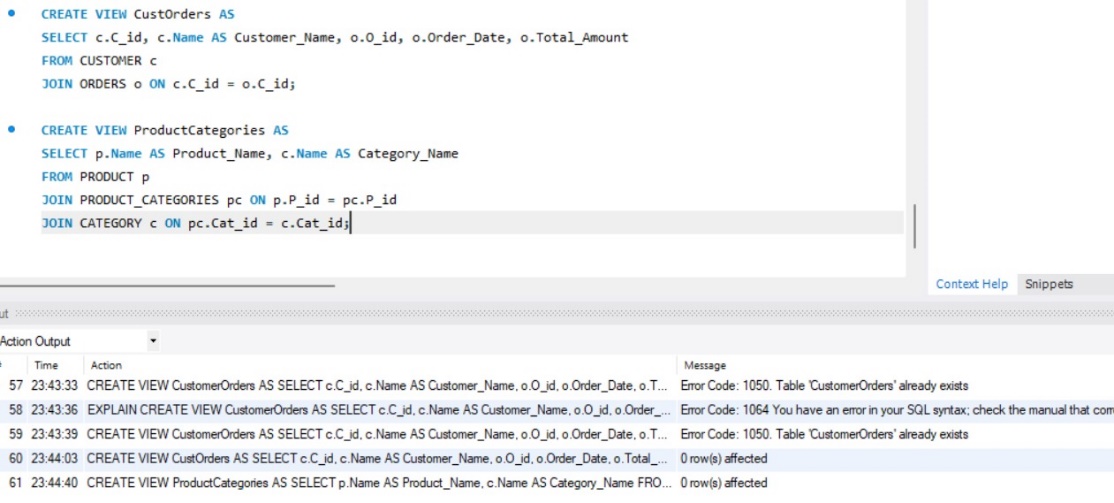
CREATE VIEW ProductCategories AS

SELECT p.Name AS Product\_Name, c.Name AS Category\_Name

FROM PRODUCT p

JOIN PRODUCT\_CATEGORIES pc ON p.P\_id = pc.P\_id

JOIN CATEGORY c ON pc.Cat\_id = c.Cat\_id;



**TOTAL CODE**

CREATE TABLE CUSTOMER (

C\_id INT PRIMARY KEY,

Name VARCHAR(255),

Email VARCHAR(255),

Ph\_no VARCHAR(20)

);

-- Create PRODUCT table

CREATE TABLE PRODUCT (

P\_id INT PRIMARY KEY,

Name VARCHAR(255),

Description TEXT

);

-- Create CATEGORY table

CREATE TABLE CATEGORY (

Cat\_id INT PRIMARY KEY,

Name VARCHAR(255)

);

-- Create ORDERS table

CREATE TABLE ORDERS (

O\_id INT PRIMARY KEY,

C\_id INT,

Order\_Date DATE,

Total\_Amount DECIMAL(10,2),

Subtotal DECIMAL(10,2),

FOREIGN KEY (C\_id) REFERENCES CUSTOMER(C\_id)

);

-- Create ORDER\_ITEMS table

CREATE TABLE ORDER\_ITEMS (

Oi\_id INT PRIMARY KEY,

O\_id INT,

P\_id INT,

Quantity INT,

Price DECIMAL(10,2),

FOREIGN KEY (O\_id) REFERENCES ORDERS(O\_id),

FOREIGN KEY (P\_id) REFERENCES PRODUCT(P\_id)

);

-- Create ADDRESS table

CREATE TABLE ADDRESS (

A\_id INT PRIMARY KEY,

C\_id INT,

Street VARCHAR(255),

City VARCHAR(100),

State VARCHAR(100),

Country VARCHAR(100),

Postal\_Code VARCHAR(20),

FOREIGN KEY (C\_id) REFERENCES CUSTOMER(C\_id)

);

-- Create PAYMENT table

CREATE TABLE PAYMENT (

Pay\_id INT PRIMARY KEY,

O\_id INT,

Amount DECIMAL(10,2),

Method VARCHAR(100),

Payment\_Date DATE,

FOREIGN KEY (O\_id) REFERENCES ORDERS(O\_id)

);

-- Create REVIEW table

CREATE TABLE REVIEW (

R\_id INT PRIMARY KEY,

C\_id INT,

O\_id INT,

P\_id INT,

Rating INT,

Comment TEXT,

FOREIGN KEY (C\_id) REFERENCES CUSTOMER(C\_id),

FOREIGN KEY (O\_id) REFERENCES ORDERS(O\_id),

FOREIGN KEY (P\_id) REFERENCES PRODUCT(P\_id)

);

-- Create PRODUCT\_CATEGORIES junction table

CREATE TABLE PRODUCT\_CATEGORIES (

P\_id INT,

Cat\_id INT,

FOREIGN KEY (P\_id) REFERENCES PRODUCT(P\_id),

FOREIGN KEY (Cat\_id) REFERENCES CATEGORY(Cat\_id)

);

-- Create ORDER\_PRODUCTS table

CREATE TABLE ORDER\_PRODUCTS (

OrderProductID INT PRIMARY KEY,

OrderID INT,

ProductID INT,

Quantity INT,

Subtotal DECIMAL(10, 2),

FOREIGN KEY (OrderID) REFERENCES ORDERS(O\_id),

FOREIGN KEY (ProductID) REFERENCES PRODUCT(P\_id)

);

-- Insert values into CUSTOMER table

INSERT INTO CUSTOMER (C\_id, Name, Email, Ph\_no) VALUES

(1, 'John Doe', 'john@example.com', '1234567890'),

(2, 'Jane Smith', 'jane@example.com', '0987654321');

-- Insert values into PRODUCT table

INSERT INTO PRODUCT (P\_id, Name, Description) VALUES

(1, 'Laptop', 'A powerful laptop for all your computing needs'),

(2, 'Smartphone', 'A sleek and powerful smartphone for staying connected on the go');

-- Insert values into CATEGORY table

INSERT INTO CATEGORY (Cat\_id, Name) VALUES

(1, 'Electronics'),

(2, 'Gadgets');

-- Insert values into ORDERS table

INSERT INTO ORDERS (O\_id, C\_id, Order\_Date, Total\_Amount, Subtotal) VALUES

(1, 1, '2024-04-13', 1500.00, 1400.00),

(2, 2, '2024-04-12', 1000.00, 950.00);

-- Insert values into ORDER\_ITEMS table

INSERT INTO ORDER\_ITEMS (Oi\_id, O\_id, P\_id, Quantity, Price) VALUES

(1, 1, 1, 1, 1400.00),

(2, 2, 2, 2, 950.00);

-- Insert values into ADDRESS table

INSERT INTO ADDRESS (A\_id, C\_id, Street, City, State, Country, Postal\_Code) VALUES

(1, 1, '123 Main St', 'Anytown', 'AnyState', 'AnyCountry', '12345'),

(2, 2, '456 Elm St', 'Othertown', 'OtherState', 'OtherCountry', '54321');

-- Insert values into PAYMENT table

INSERT INTO PAYMENT (Pay\_id, O\_id, Amount, Method, Payment\_Date) VALUES

(1, 1, 1500.00, 'Credit Card', '2024-04-13'),

(2, 2, 1000.00, 'PayPal', '2024-04-12');

-- Insert values into REVIEW table

INSERT INTO REVIEW (R\_id, C\_id, O\_id, P\_id, Rating, Comment) VALUES

(1, 1, 1, 1, 5, 'Great product!'),

(2, 2, 2, 2, 4, 'Good but could be better.');

-- Insert values into PRODUCT\_CATEGORIES table (for junction table)

INSERT INTO PRODUCT\_CATEGORIES (P\_id, Cat\_id) VALUES

(1, 1), -- Laptop belongs to Electronics category

(2, 1); -- Smartphone belongs to Electronics category

-- Insert values into ORDER\_PRODUCTS table

INSERT INTO ORDER\_PRODUCTS (OrderProductID, OrderID, ProductID, Quantity, Subtotal) VALUES

(1, 1, 1, 1, 1400.00),

(2, 2, 2, 2, 1900.00);

-- Show contents of CUSTOMER table

SELECT \* FROM CUSTOMER;

-- Show contents of PRODUCT table

SELECT \* FROM PRODUCT;

-- Show contents of CATEGORY table

SELECT \* FROM CATEGORY;

-- Show contents of ORDERS table

SELECT \* FROM ORDERS;

-- Show contents of ORDER\_ITEMS table

SELECT \* FROM ORDER\_ITEMS;

-- Show contents of ADDRESS table

SELECT \* FROM ADDRESS;

-- Show contents of PAYMENT table

SELECT \* FROM PAYMENT;

-- Show contents of REVIEW table

SELECT \* FROM REVIEW;

-- Show contents of PRODUCT\_CATEGORIES table

SELECT \* FROM PRODUCT\_CATEGORIES;

-- Show contents of ORDER\_PRODUCTS table

SELECT \* FROM ORDER\_PRODUCTS;

SELECT c.C\_id, c.Name AS Customer\_Name, o.O\_id, o.Order\_Date, o.Total\_Amount

FROM CUSTOMER c

JOIN ORDERS o ON c.C\_id = o.C\_id;

SELECT o.O\_id, p.Name AS Product\_Name, oi.Quantity, oi.Price

FROM ORDERS o

JOIN ORDER\_ITEMS oi ON o.O\_id = oi.O\_id

JOIN PRODUCT p ON oi.P\_id = p.P\_id;

CREATE VIEW CustOrders AS

SELECT c.C\_id, c.Name AS Customer\_Name, o.O\_id, o.Order\_Date, o.Total\_Amount

FROM CUSTOMER c

JOIN ORDERS o ON c.C\_id = o.C\_id;

CREATE VIEW ProductCategories AS

SELECT p.Name AS Product\_Name, c.Name AS Category\_Name

FROM PRODUCT p

JOIN PRODUCT\_CATEGORIES pc ON p.P\_id = pc.P\_id

JOIN CATEGORY c ON pc.Cat\_id = c.Cat\_id;