

جامعة أم القرى  
UMM AL-QURA UNIVERSITY

# Database Project

## Pharmacy

Group : 4

supervisor name: Afaf Muslim Salem Al-Mahmadi



Student Name	ID	section number
Taef Abd Ul-Wahab	44512026	4
Raneem khalid alharbi	445002398	4
Lana Nasser Alotaibi	445001397	4
Ghada Shujaa Alotibi	445003024	4
Retaj Khalid Alqurashi	44512037	4

# Business Rules

## Policy Description

The objective of this project is to create a database that manages all operations related to pharmacies, medications, customers, orders, and payments. The database includes several tables containing important data such as pharmacy information, available medications, customers who interact with these pharmacies, orders placed, and customer payments.



# Business Rules

## Procedure Description:

1. A **pharmacy** must have a unique identifier (PharmacyID), name, address, phone number, email, license number, and opening hours.
2. Each **medication** must have a unique identifier (MedicationID), name, brand, price, the pharmacy it belongs to, quantity in stock, and expiry date.
3. A **customer** must have a unique identifier (CustomerID), first name, last name, email, phone number, address, and registration Date.
4. Each **order** must have a unique identifier (OrderID), the customer who placed the order, the pharmacy fulfilling the order, the order date, status (completed, pending, cancelled) , total amount, and whether a prescription is required.
5. Each **payment** must have a unique identifier (PaymentID), the associated order, the customer who made the payment, the amount paid, payment date, method of payment (e.g., credit card, cash), and the payment status (e.g., completed, pending, refunded).
6. A pharmacy can stock multiple medications and Each medication can be stocked by multiple pharmacies .
7. A customer can place multiple orders and Each order is placed by one customer.
8. A pharmacy can receive multiple orders and Each order is associated with one pharmacy.
9. Each order is associated with exactly one payment and Each payment is made for one order.
10. Each customer can have only one payment record associated with them per transaction and Each payment record is uniquely associated with one customer.
11. Each Medication can appear in multiple orders and Each order can contain one medication



# Business Rules

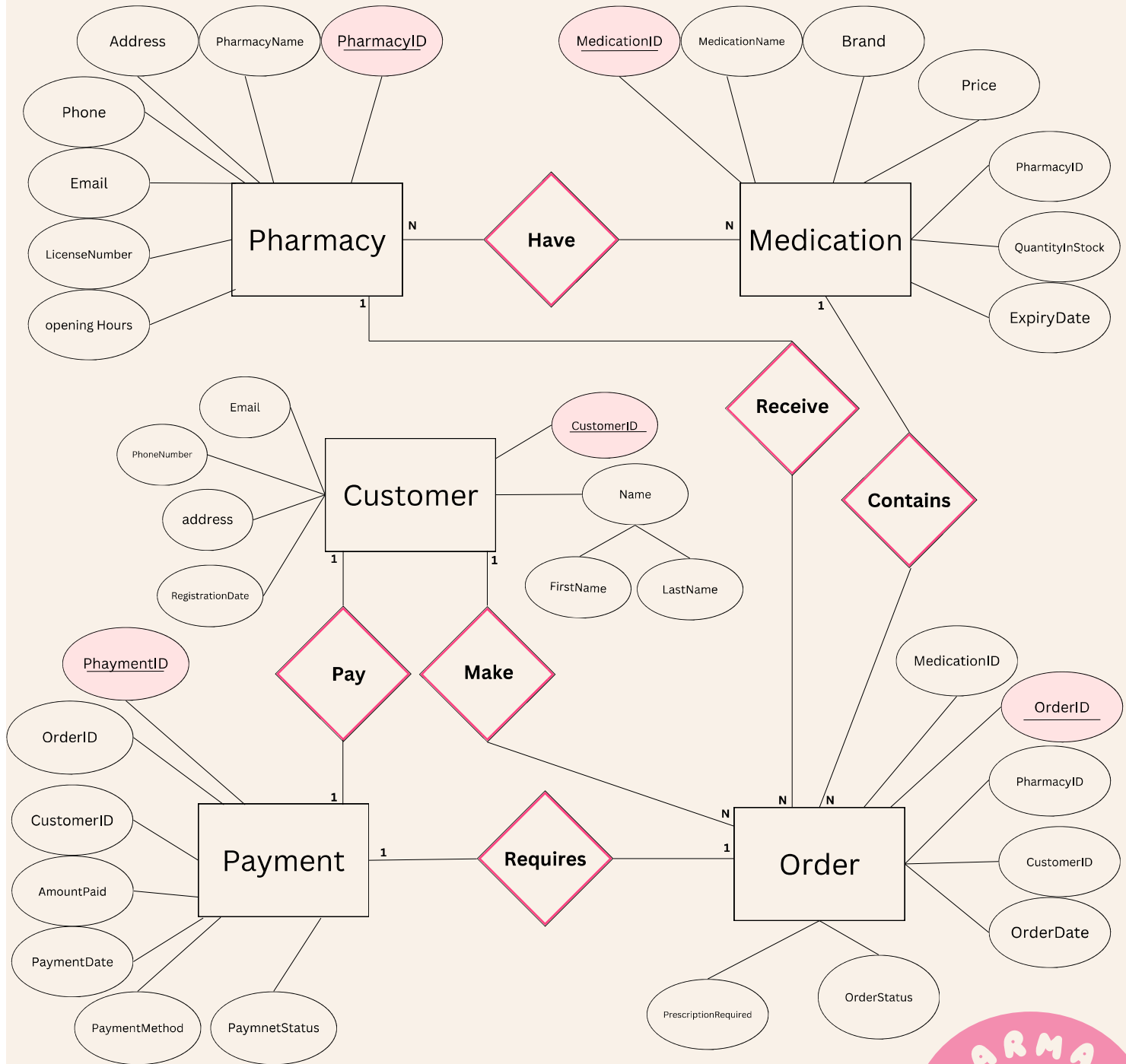
## Principles

- **Data Integrity:** The system maintains accurate data relationships, ensuring that pharmacy information, medication stock, customer orders, and payments are all linked properly. This ensures that customers receive the correct products, and pharmacies maintain adequate stock.
- **Security and Privacy:** Customer data, including email and payment details, is handled securely, ensuring that private information is only accessible by authorized users.
- **Transparency:** All transactions (orders and payments) are clearly recorded, and updates are made to ensure real-time reflection of status (Pending, Completed, Cancelled).
- **Compliance:** All medications are handled in compliance with health regulations, including ensuring accurate expiry dates and removing medications from the system if the stock is insufficient.



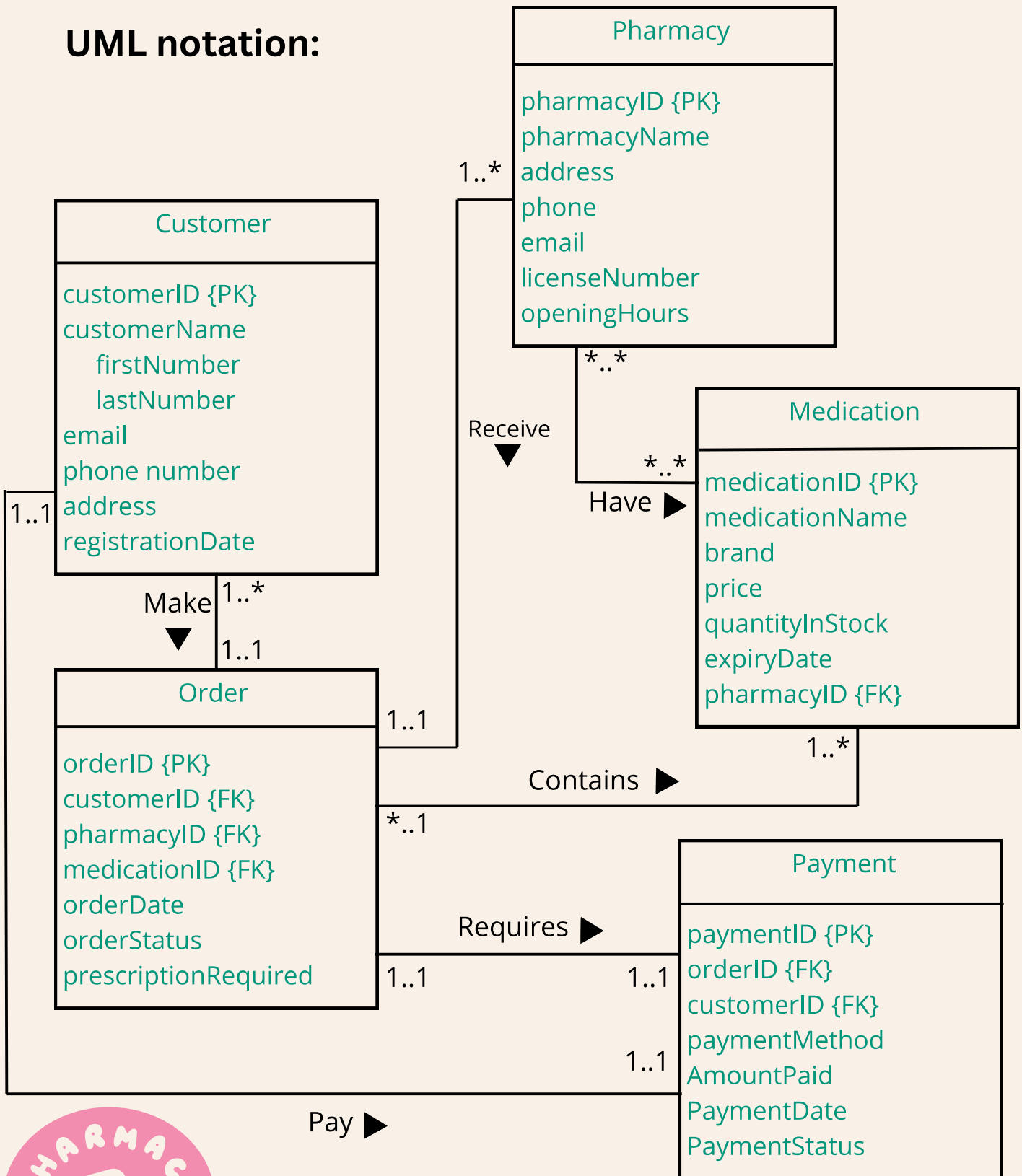
# ER Model

## Chen's Model:



# ER Model

## UML notation:



# Mapping of ER into Relational Schema:

Based on the ER Model:

**Identify entity types:**

- Strong entity types: Pharmacy, Medication , Customer , Orders, payment.
- Weak entity types: None

**Identify relationship types:**

- One-to-One: Pay, Requires
- One-to-Many: Make, Contains, Receive
- Many-to-Many: Have
- N-ary: None

Step 1. Mapping the **Regular** entity type:

Pharmacy

<u>PharmacyID</u>	PharmacyName	Address	Phone	Email	LicenseNumber	OpeningHours
-------------------	--------------	---------	-------	-------	---------------	--------------

Medication

<u>MedicationID</u>	MedicationName	Brand	Price	QuantityInStock	ExpiryDate	PharmcyID
---------------------	----------------	-------	-------	-----------------	------------	-----------

Customer

<u>CustomerID</u>	FirstName	LastName	Email	PhoneNumber	Address	registrationDate
-------------------	-----------	----------	-------	-------------	---------	------------------

Orders

<u>OrderID</u>	CustomerID	PharmacyID	MedcationID	OrderDate	OrderStatus	TotalAmount	prescriptionRequired
----------------	------------	------------	-------------	-----------	-------------	-------------	----------------------

Payment

<u>PaymentID</u>	CustomerID	OrderID	PaymentDate	PaymentStatus	AmountPaid	PaymentMethod
------------------	------------	---------	-------------	---------------	------------	---------------

Step 2. Mapping the **Weak** entity type:

None.



# Mapping of ER into Relational Schema:

Step 3. Mapping of 1:1 relationship type:

Pharmacy

<u>PharmacyID</u>	PharmacyName	Address	Phone	Email	LicenseNumber	OpeningHours
-------------------	--------------	---------	-------	-------	---------------	--------------

Medication

<u>MedicationID</u>	MedicationName	Brand	Price	QuantityInStock	ExpiryDate	PharmacyID
---------------------	----------------	-------	-------	-----------------	------------	------------

Customer

<u>CustomerID</u>	FirstName	LastName	Email	PhoneNumber	Address	registrationDate
-------------------	-----------	----------	-------	-------------	---------	------------------

Orders

<u>OrderID</u>	CustomerID	PharmacyID	MedcationID	OrderDate	OrderStatus	TotalAmount	PrescriptionRequired
----------------	------------	------------	-------------	-----------	-------------	-------------	----------------------

Payment

<u>PaymentID</u>	CustomerID	OrderID	PaymentDate	PaymentStatus	AmountPaid	PaymentMethod
------------------	------------	---------	-------------	---------------	------------	---------------





# Mapping of ER into Relational Schema:

Step 4. Mapping of 1:N relationship type:

Pharmacy

<u>PharmacyID</u>	PharmacyName	Address	Phone	Email	LicenseNumber	OpeningHours
-------------------	--------------	---------	-------	-------	---------------	--------------

Medication

<u>MedicationID</u>	MedicationName	Brand	Price	QuantityInStock	ExpiryDate	PharmacyID
---------------------	----------------	-------	-------	-----------------	------------	------------

Customer

<u>CustomerID</u>	FirstName	LastName	Email	PhoneNumber	Address	registrationDate
-------------------	-----------	----------	-------	-------------	---------	------------------

Orders

<u>OrderID</u>	CustomerID	PharmacyID	MedcationID	OrderDate	OrderStatus	TotalAmount	prescriptionRequired
----------------	------------	------------	-------------	-----------	-------------	-------------	----------------------

Payment

<u>PaymentID</u>	CustomerID	OrderID	PaymentDate	PaymentStatus	AmountPaid	PaymentMethod
------------------	------------	---------	-------------	---------------	------------	---------------



# Mapping of ER into Relational Schema:

Step 5: Mapping of Binary M:N relationship type :

Pharmacy

<u>PharmacyID</u>	PharmacyName	Address	Phone	Email	LicenseNumber	OpeningHours
-------------------	--------------	---------	-------	-------	---------------	--------------

Medication

<u>MedicationID</u>	MedicationName	Brand	Price	QuantityInStock	ExpiryDate	PharmacyID
---------------------	----------------	-------	-------	-----------------	------------	------------

Customer

<u>CustomerID</u>	FirstName	LastName	Email	PhoneNumber	Address	registrationDate
-------------------	-----------	----------	-------	-------------	---------	------------------

Orders

<u>OrderID</u>	CustomerID	PharmacyID	MedcationID	OrderDate	OrderStatus	TotalAmount	prescriptionRequired
----------------	------------	------------	-------------	-----------	-------------	-------------	----------------------

Payment

<u>PaymentID</u>	CustomerID	OrderID	PaymentDate	PaymentStatus	AmountPaid	PaymentMethod
------------------	------------	---------	-------------	---------------	------------	---------------

Have

<u>PharmacyID</u>	<u>MedcationID</u>	StockQuantity
-------------------	--------------------	---------------

Step 6: Mapping of Multivalued

None

Step 7: Mapping of N-ary

None



# The Final Rational Schema

## Pharmacy

<u>PharmacyID</u>	PharmacyName	Address	Phone	Email	LicenseNumber	OpeningHours
-------------------	--------------	---------	-------	-------	---------------	--------------

## Medication

<u>MedicationID</u>	MedicationName	Brand	Price	QuantityInStock	ExpiryDate	PharmacyID
---------------------	----------------	-------	-------	-----------------	------------	------------

## Customer

<u>CustomerID</u>	FirstName	LastName	Email	PhoneNumber	Address	registrationDate
-------------------	-----------	----------	-------	-------------	---------	------------------

## Orders

<u>OrderID</u>	CustomerID	PharmacyID	MedcationID	OrderDate	OrderStatus	TotalAmount	prescriptionRequired
----------------	------------	------------	-------------	-----------	-------------	-------------	----------------------

## Payment

<u>PaymentID</u>	CustomerID	OrderID	PaymentDate	PaymentStatus	AmountPaid	PaymentMethod
------------------	------------	---------	-------------	---------------	------------	---------------

## Have

<u>PharmacyID</u>	<u>MedcationID</u>	<u>StockQuantity</u>
-------------------	--------------------	----------------------



# Screenshots of SQL commands and their results:

## Pharmacy

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
1 SELECT * FROM pharmacy.pharmacy;
```

PharmacyID	PharmacyName	Address	Phone	Email	LicenseNumber	OpeningHours
1	Nahd Pharmacy	123 st, Makkah	111-2245	nahd@gmail.com	abc1122	24/7
2	Nahd Pharmacy	111 st, Jeddah	098-8576	nahd@gmail.com	abc1122	24/7
3	Nahd Pharmacy	432 st, Taif	260-7245	nahd@gmail.com	abc1122	24/7
4	Nahd Pharmacy	565 st, Riyadh	073-6932	nahd@gmail.com	abc1122	24/7
5	Nahd Pharmacy	342 st, Madinah	713-0684	nahd@gmail.com	abc1122	24/7
6	Nahd Pharmacy	355 st, Madinah	791-0428	nahd@gmail.com	abc1122	24/7

pharmacy 1 x

Output

#	Time	Action	Message	Duration / Fetch
1	17:35:48	USE pharmacy	0 row(s) affected	0.000 sec
2	17:35:55	SELECT * FROM pharmacy.pharmacy LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
3	17:36:14	INSERT INTO Pharmacy (PharmacyID, PharmacyName, Address, Phone, Email, LicenseNumber, OpeningHours) V...	1 row(s) affected	0.000 sec
4	17:36:20	SELECT * FROM pharmacy.pharmacy LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
1 SELECT * FROM pharmacy.pharmacy;
```

PharmacyID	PharmacyName	Address	Phone	Email	LicenseNumber	OpeningHours
1	Nahd Pharmacy	123 st, Makkah	111-2245	nahd@gmail.com	abc1122	24/7
2	Nahd Pharmacy	111 st, Jeddah	098-8576	nahd@gmail.com	abc1122	24/7
3	Nahd Pharmacy	432 st, Taif	260-7245	nahd@gmail.com	abc1122	24/7
4	Nahd Pharmacy	565 st, Riyadh	073-6932	nahd@gmail.com	abc1122	24/7
5	Nahd Pharmacy	342 st, Madinah	713-0684	nahd@gmail.com	abc1122	24/7

pharmacy 1 x

Output

#	Time	Action	Message	Duration / Fetch
1	17:35:48	USE pharmacy	0 row(s) affected	0.000 sec
2	17:35:55	SELECT * FROM pharmacy.pharmacy LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
3	17:36:14	INSERT INTO Pharmacy (PharmacyID, PharmacyName, Address, Phone, Email, LicenseNumber, OpeningHours) V...	1 row(s) affected	0.000 sec
4	17:36:20	SELECT * FROM pharmacy.pharmacy LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
5	17:37:05	DELETE FROM pharmacy WHERE PharmacyID=6	1 row(s) affected	0.000 sec
6	17:37:16	SELECT * FROM pharmacy.pharmacy LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
28 WHERE PharmacyID=i;
```

```
29
```

```
30 DELETE FROM pharmacy
```

```
31 WHERE PharmacyID=i;
```

```
32
```

```
33 SELECT * FROM Pharmacy
```

```
34 ORDER BY PharmacyID ASC;
```

```
35
```

```
36 CREATE TABLE IF NOT EXISTS Medications (
```

```
37 MedicationID INT PRIMARY KEY,
```

```
38 MedicationName VARCHAR(225) NOT NULL,
```

```
39 Brand VARCHAR(100),
```

```
40 Price DECIMAL(10,2) NOT NULL,
```

```
41 QuantityInStock INT NOT NULL,
```

```
42 ExpiryDate DATE,
```

```
43
```

```
44
```

PharmacyID	PharmacyName	Address	Phone	Email	LicenseNumber	OpeningHours
1	Nahd Pharmacy	123 st, Makkah	111-2245	nahd@gmail.com	abc1122	24/7
2	Nahd Pharmacy	111 st, Jeddah	098-8576	nahd@gmail.com	abc1122	24/7
3	Nahd Pharmacy	432 st, Taif	260-7245	nahd@gmail.com	abc1122	24/7
4	Nahd Pharmacy	565 st, Riyadh	073-6932	nahd@gmail.com	abc1122	24/7
5	Nahd Pharmacy	342 st, Madinah	713-0684	nahd@gmail.com	abc1122	24/7

Pharmacy 27 x

Output

#	Time	Action	Message	Duration / Fetch
1	13:39:11	SELECT * FROM Pharmacy ORDER BY PharmacyID ASC LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

# Screenshots of SQL commands and their results:

## Medications

projectDB\* pharmacy medications pharmacy medications

Limit to 1000 rows

1 • SELECT \* FROM pharmacy.medications;

Result Grid

MedicationID	MedicationName	Brand	Price	QuantityInStock	ExpiryDate	PharmacyID
1	Paracetamol	Paradol	33.00	100	2026-12-31	1
2	Ibuprofen	Advil	8.75	50	2025-11-30	2
3	Cough Syrup	Benlyn	12.00	25	2025-10-15	3
4	Amoxicillin	Moxatag	18.90	200	2026-01-20	4
5	Antihistamine	Zyrtec	10.00	150	2024-09-15	5
6	prozac	El Lilly	100.00	100	2024-10-11	5

medications 1 x

Apply Revert Context Help Snippets

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	17:39:44	SELECT * FROM pharmacy.medications LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec

projectDB\* pharmacy medications pharmacy medications

Limit to 1000 rows

1 • SELECT \* FROM pharmacy.medications;

Result Grid

MedicationID	MedicationName	Brand	Price	QuantityInStock	ExpiryDate	PharmacyID
1	Paracetamol	Paradol	33.00	100	2026-12-31	1
2	Ibuprofen	Advil	8.75	50	2025-11-30	2
4	Amoxicillin	Moxatag	18.90	200	2026-01-20	4
5	Antihistamine	Zyrtec	10.00	150	2024-09-15	5
6	prozac	El Lilly	100.00	100	2024-10-11	5

medications 1 x

Apply Revert Context Help Snippets

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output

Action Output

#	Time	Action	Message	Duration / Fetch
2	13:39:49	SELECT PharmacyID, COUNT(*) AS MedicationCount FROM Medications GROUP BY PharmacyID LIMIT 0, 1...	4 row(s) returned	0.000 sec / 0.000 sec

projectDB\* pharmacy medications pharmacy medications

Limit to 1000 rows

54 • UPDATE Medications  
55 • SET Price =Price\* 6.00  
56 • WHERE MedicationID = 1;  
57 •  
58 • DELETE FROM Medications  
59 • WHERE QuantityInStock < 30;  
60 •  
61 • SELECT PharmacyID, COUNT(\*) AS MedicationCount  
62 • FROM Medications  
63 • GROUP BY PharmacyID;  
64 •  
65 • SELECT \* FROM Medications;  
66 •  
67 •  
68 •

Result Grid

MedicationID	MedicationName	Brand	Price	QuantityInStock	ExpiryDate	PharmacyID
1	Paracetamol	Paradol	33.00	100	2026-12-31	1
2	Ibuprofen	Advil	8.75	50	2025-11-30	2
4	Amoxicillin	Moxatag	18.90	200	2026-01-20	4
5	Antihistamine	Zyrtec	10.00	150	2024-09-15	5
6	prozac	El Lilly	100.00	100	2024-10-11	5

Medications 39 x

Apply Revert Context Help Snippets

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Output

Action Output

#	Time	Action	Message	Duration / Fetch
2	13:39:49	SELECT PharmacyID, COUNT(*) AS MedicationCount FROM Medications GROUP BY PharmacyID LIMIT 0, 1...	4 row(s) returned	0.000 sec / 0.000 sec

projectDB\* pharmacy medications pharmacy medications

Limit to 1000 rows

54 • UPDATE Medications  
55 • SET Price =Price\* 6.00  
56 • WHERE MedicationID = 1;  
57 •  
58 • DELETE FROM Medications  
59 • WHERE QuantityInStock < 30;  
60 •  
61 • SELECT PharmacyID, COUNT(\*) AS MedicationCount  
62 • FROM Medications  
63 • GROUP BY PharmacyID;  
64 •  
65 • SELECT \* FROM Medications;  
66 •  
67 •  
68 •

Result Grid

PharmacyID	MedicationCount
1	1
2	1
4	1
5	2

Result 38 x

Read Only Context Help Snippets

# Screenshots of SQL commands and their results:

## Customers

projectDB\* medications customers

Limit to 1000 rows

1 \* SELECT \* FROM pharmacy.customers;

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

Filter Rows

1 2 3 4 5

CustomerID	FirstName	LastName	Email	PhoneNumber	Address	RegistrationDate
1	Lana	Alotabi	lana.alotabi@gmail.com	0599880101	Mecca, Saudi Arabia	2025-01-01
2	Ahmed	Ali	ahmed.ali@gmail.com	0595773200	Jeddah, Saudi Arabia	2025-01-02
3	Noura	Salem	noura.salem@gmail.com	0505221055	Dammam, Saudi Arabia	2025-01-03
4	Hassan	Omar	hassan.omar@gmail.com	0571436366	Riyadh, Saudi Arabia	2025-01-04
5	Maha	Khalid	maha.khalid@gmail.com	0504432808	Medina, Saudi Arabia	2025-01-05

customers 1 x

Apply Revert Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	17:39:44	SELECT * FROM pharmacy.medications LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
2	17:40:33	DELETE FROM Medications WHERE QuantityInStock < 30	1 row(s) affected	0.015 sec
3	17:40:45	SELECT * FROM pharmacy.medications LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
4	17:41:23	INSERT INTO Customers (CustomerID, FirstName, LastName, Email, PhoneNumber, Address, RegistrationDate) ...	1 row(s) affected	0.016 sec
5	17:41:36	SELECT * FROM pharmacy.customers LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

projectDB\* customers

Limit to 1000 rows

1 \* SELECT \* FROM pharmacy.customers;

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

Filter Rows

1 2 3 4 5

CustomerID	FirstName	LastName	Email	PhoneNumber	Address	RegistrationDate
1	Lana	Alotabi	lana.alotabi@gmail.com	0599880101	Mecca, Saudi Arabia	2025-01-01
2	Ahmed	Ali	ahmed.ali@gmail.com	0595773200	Jeddah, Saudi Arabia	2025-01-02
3	Noura	Salem	noura.salem@gmail.com	0505221055	Dammam, Saudi Arabia	2025-01-03
5	Maha	Khalid	maha.khalid@gmail.com	0504432808	Medina, Saudi Arabia	2025-01-05

customers 1 x

Apply Revert Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	17:39:44	SELECT * FROM pharmacy.medications LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
2	17:40:33	DELETE FROM Medications WHERE QuantityInStock < 30	1 row(s) affected	0.015 sec
3	17:40:45	SELECT * FROM pharmacy.medications LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
4	17:41:23	INSERT INTO Customers (CustomerID, FirstName, LastName, Email, PhoneNumber, Address, RegistrationDate) ...	1 row(s) affected	0.016 sec
5	17:41:36	SELECT * FROM pharmacy.customers LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
6	17:41:57	DELETE FROM Customers WHERE Email = 'hassan.omar@gmail.com'	1 row(s) affected	0.000 sec
7	17:42:04	SELECT * FROM pharmacy.customers LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

# Screenshots of SQL commands and their results:

## Orders

1 \* SELECT \* FROM pharmacy.customers;

CustomerID	FirstName	LastName	Email	PhoneNumber	Address	RegistrationDate
1	Lana	Alotabi	lana.alotabi@gmail.com	0599880101	Mecca, Saudi Arabia	2025-01-01
2	Ahmed	Ali	ahmed.ali@gmail.com	0595772000	Jeddah, Saudi Arabia	2025-01-02
3	Noura	Salem	noura.salem@gmail.com	0505221055	Dammam, Saudi Arabia	2025-01-03
4	Hassan	Onar	hassan.onar@gmail.com	0571456366	Riyadh, Saudi Arabia	2025-01-04
5	Maha	Khalid	maha.khalid@gmail.com	0504432808	Medina, Saudi Arabia	2025-01-05

customers 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	17:39:44	SELECT * FROM pharmacy.medications LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
2	17:40:33	DELETE FROM Medications WHERE QuantityInStock < 30	1 row(s) affected	0.015 sec
3	17:40:45	SELECT * FROM pharmacy.medications LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
4	17:41:23	INSERT INTO Customers (CustomerID, FirstName, LastName, Email, PhoneNumber, Address, RegistrationDate)	1 row(s) affected	0.016 sec
5	17:41:36	SELECT * FROM pharmacy.customers LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec

projectDB

orders

1 \* SELECT \* FROM pharmacy.orders;

OrderID	CustomerID	PharmacyID	MedicationID	OrderDate	OrderStatus	PrescriptionRequired
1	3	1	4	2025-01-17 12:59:00	Pending	0
2	3	5	2	2024-12-24 17:45:20	Completed	0
3	2	2	1	2025-01-12 03:25:56	Cancelled	0
4	2	2	6	2025-01-13 14:32:00	Completed	1
5	1	1	4	2024-11-30 21:17:47	Pending	0
6	5	3	2	2025-01-08 18:34:00	Completed	0
7	3	4	5	2025-01-15 10:23:42	Pending	0
8	2	1	6	2025-01-02 20:59:34	Pending	1

orders 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
2	21:49:58	INSERT INTO Orders (OrderID, CustomerID, PharmacyID, MedicationID, OrderDate, OrderStatus, TotalAmount)	Error Code: 1054 Unknown column 'TotalAmount' in field list	0.000 sec
3	21:49:59	INSERT INTO Orders (OrderID, CustomerID, PharmacyID, MedicationID, OrderDate, OrderStatus, Prescription...	8 row(s) affected Records: 8 Duplicates: 0 Warnings: 0	0.000 sec
4	21:49:56	SELECT * FROM pharmacy.orders LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec

projectDB

customers

1 \* SELECT \* FROM pharmacy.customers;

CustomerID	FirstName	LastName	Email	PhoneNumber	Address	RegistrationDate
1	Lana	Alotabi	lana.alotabi@gmail.com	0599880101	Mecca, Saudi Arabia	2025-01-01
2	Ahmed	Ali	ahmed.ali@gmail.com	0595772000	Jeddah, Saudi Arabia	2025-01-02
3	Noura	Salem	noura.salem@gmail.com	0505221055	Dammam, Saudi Arabia	2025-01-03
5	Maha	Khalid	maha.khalid@gmail.com	0504432808	Medina, Saudi Arabia	2025-01-05

customers 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	17:39:44	SELECT * FROM pharmacy.medications LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
2	17:40:33	DELETE FROM Medications WHERE QuantityInStock < 30	1 row(s) affected	0.015 sec
3	17:40:45	SELECT * FROM pharmacy.medications LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
4	17:41:23	INSERT INTO Customers (CustomerID, FirstName, LastName, Email, PhoneNumber, Address, RegistrationDate)	1 row(s) affected	0.016 sec
5	17:41:36	SELECT * FROM pharmacy.customers LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
6	17:41:57	DELETE FROM Customers WHERE Email = 'hassan.onar@gmail.com'	1 row(s) affected	0.000 sec
7	17:42:04	SELECT * FROM pharmacy.customers LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

projectDB

orders

130 \* UPDATE Orders

131 SET OrderStatus = 'Completed'

132 WHERE OrderID = 1;

133

134 \* DELETE FROM Orders

135 WHERE OrderID = 5;

136

137 \* SELECT \* FROM Orders

138 WHERE OrderStatus = 'Completed';

139

140 \* SELECT PharmacyID, COUNT(OrderID) AS NumberOfOrders

141 FROM Orders

142 GROUP BY PharmacyID;

OrderID	CustomerID	PharmacyID	MedicationID	OrderDate	OrderStatus	PrescriptionRequired
1	3	1	4	2025-01-17 12:59:00	Completed	0
2	3	5	2	2024-12-24 17:45:20	Completed	0
4	2	2	6	2025-01-13 14:32:00	Completed	1
6	5	3	2	2025-01-08 18:34:00	Completed	0
7	3	4	5	2025-01-15 10:23:42	Completed	0

Orders 1 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
9	21:50:30	DELETE FROM Orders WHERE OrderID = 8	1 row(s) affected	0.016 sec
10	21:50:34	SELECT * FROM pharmacy.orders LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec



# Screenshots of SQL commands and their results:

## Orders

136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148

```
SELECT * FROM Orders
WHERE OrderStatus = 'Completed';

SELECT PharmacyID, COUNT(OrderID) AS NumberOfOrders
FROM Orders
GROUP BY PharmacyID;

SELECT PharmacyID, COUNT(OrderID) AS NumberOfOrders
FROM Orders
GROUP BY PharmacyID
HAVING COUNT(OrderID) > 1;
```

Result Grid

PharmacyID	NumberOfOrders
1	2
2	2

Result 3

Action Output

#	Time	Action	Message	Duration / Fetch
11	21:50:54	SELECT * FROM Orders WHERE OrderStatus = 'Completed' LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
12	21:51:13	SELECT PharmacyID, COUNT(OrderID) AS NumberOfOrders FROM Orders GROUP BY PharmacyID LIMIT 0, ...	5 row(s) returned	0.000 sec / 0.000 sec

136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148

```
SELECT * FROM Orders
WHERE OrderStatus = 'Completed';

SELECT PharmacyID, COUNT(OrderID) AS NumberOfOrders
FROM Orders
GROUP BY PharmacyID;

SELECT PharmacyID, COUNT(OrderID) AS NumberOfOrders
FROM Orders
GROUP BY PharmacyID
HAVING COUNT(OrderID) > 1;
```

Result Grid

PharmacyID	NumberOfOrders
1	2
2	2
3	1
4	1
5	1

Result 2

Action Output

#	Time	Action	Message	Duration / Fetch
10	21:50:34	SELECT * FROM pharmacy orders LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec
11	21:50:54	SELECT * FROM Orders WHERE OrderStatus = 'Completed' LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec
12	21:51:13	SELECT PharmacyID, COUNT(OrderID) AS NumberOfOrders FROM Orders GROUP BY PharmacyID LIMIT 0, ...	5 row(s) returned	0.000 sec / 0.000 sec

147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159

```
HAVING COUNT(OrderID) > 1;

SELECT * FROM Orders
ORDER BY OrderDate DESC;

SELECT Orders.OrderID, Customers.FirstName, Customers.LastName, Pharmacy.PharmacyName
FROM Orders
JOIN Customers ON Orders.CustomerID = Customers.CustomerID
JOIN Pharmacy ON Orders.PharmacyID = Pharmacy.PharmacyID;

CREATE TABLE IF NOT EXISTS Payment (
    PaymentID INT PRIMARY KEY,
    OrderID INT NOT NULL,
```

Result Grid

OrderID	CustomerID	PharmacyID	MedicationID	OrderDate	OrderStatus	PrescriptionRequired
1	3	1	4	2025-01-17 12:59:00	Completed	0
7	3	4	5	2025-01-15 00:23:42	Completed	0
4	2	2	6	2025-01-13 14:32:00	Completed	1
3	2	2	1	2025-01-12 03:25:56	Cancelled	0
6	5	3	2	2025-01-08 18:34:00	Completed	0
2	3	5	2	2024-12-24 17:45:30	Completed	0
5	1	1	4	2024-11-30 21:17:47	Cancelled	0

Orders 4

Action Output

#	Time	Action	Message	Duration / Fetch
12	21:51:13	SELECT PharmacyID, COUNT(OrderID) AS NumberOfOrders FROM Orders GROUP BY PharmacyID LIMIT 0, ...	5 row(s) returned	0.000 sec / 0.000 sec
13	21:51:28	SELECT PharmacyID, COUNT(OrderID) AS NumberOfOrders FROM Orders GROUP BY PharmacyID HAVING...	2 row(s) returned	0.000 sec / 0.000 sec
14	21:51:47	SELECT * FROM Orders ORDER BY OrderDate DESC LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec

147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159

```
HAVING COUNT(OrderID) > 1;

SELECT * FROM Orders
ORDER BY OrderDate DESC;

SELECT Orders.OrderID, Customers.FirstName, Customers.LastName, Pharmacy.PharmacyName
FROM Orders
JOIN Customers ON Orders.CustomerID = Customers.CustomerID
JOIN Pharmacy ON Orders.PharmacyID = Pharmacy.PharmacyID;

CREATE TABLE IF NOT EXISTS Payment (
    PaymentID INT PRIMARY KEY,
    OrderID INT NOT NULL,
```

Result Grid

OrderID	FirstName	LastName	PharmacyName
5	Lina	Alotabi	Nahdi Pharmacy
3	Ahmed	Ali	Nahdi Pharmacy
4	Ahmed	Ali	Nahdi Pharmacy
2	Noura	Salem	Nahdi Pharmacy
7	Noura	Salem	Nahdi Pharmacy
6	Maha	Khalid	Nahdi Pharmacy

Result 5

Action Output

#	Time	Action	Message	Duration / Fetch
13	21:51:28	SELECT PharmacyID, COUNT(OrderID) AS NumberOfOrders FROM Orders GROUP BY PharmacyID HAVING...	2 row(s) returned	0.000 sec / 0.000 sec
14	21:51:47	SELECT * FROM Orders ORDER BY OrderDate DESC LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.000 sec



# Screenshots of SQL commands and their results:

## Payment

1

SELECT \* FROM pharmacy.payment;

Result Grid

PaymentID	OrderID	CustomerID	PaymentMethod	AmountPaid	PaymentDate	PaymentStatus
1	1	3	Credit Card	250.50	2025-01-15	Completed
2	2	3	Cash	120.00	2025-01-16	Pending
3	3	2	Credit Card	300.75	2025-01-17	Completed
4	4	2	Credit Card	75.00	2025-01-17	Refunded
5	5	1	Cash	50.00	2025-01-18	Completed

payment 1

Output

Action Output

#	Time	Action	Message	Duration / Fetch
47	13.35.15	INSERT INTO Payment (PaymentID, OrderID, CustomerID, PaymentMethod, AmountPaid, PaymentDate, Pay...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec

projectDB

pharmacy medications pharmacy customers customers customers orders payment

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

Result Grid

PaymentID	OrderID	CustomerID	PaymentMethod	AmountPaid	PaymentDate	PaymentStatus
1	1	3	Credit Card	250.50	2025-01-15	Completed
2	2	3	Cash	120.00	2025-01-16	Completed
3	3	2	Credit Card	300.75	2025-01-17	Completed
5	5	1	Cash	50.00	2025-01-18	Completed

Payment 31

Output

Action Output

#	Time	Action	Message	Duration / Fetch
50	13.35.42	DELETE FROM Payment WHERE PaymentStatus = 'Refunded'	1 row(s) affected	0.000 sec
51	13.35.47	SELECT * FROM Payment LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

projectDB

pharmacy medications pharmacy customers customers customers orders payment

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

Result Grid

PaymentID	OrderID	CustomerID	PaymentMethod	AmountPaid	PaymentDate	PaymentStatus
1	1	3	Credit Card	250.50	2025-01-15	Completed
2	2	3	Cash	120.00	2025-01-16	Completed
3	3	2	Credit Card	300.75	2025-01-17	Completed
5	5	1	Cash	50.00	2025-01-18	Completed

Payment 32

Output

Action Output

#	Time	Action	Message	Duration / Fetch
51	13.35.47	SELECT * FROM Payment LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
52	13.36.05	SELECT * FROM Payment WHERE PaymentStatus = 'Completed' LIMIT 0, 1000	4 row(s) returned	0.015 sec / 0.000 sec

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

# Screenshots of SQL commands and their results:

## Payment

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
185 SET PaymentStatus = 'Completed';
186 WHERE PaymentID = 2;
187
188 DELETE FROM Payment
189 WHERE PaymentStatus = 'Refunded';
190
191 SELECT * FROM Payment;
192
193 SELECT * FROM Payment WHERE PaymentStatus = 'Completed';
194
195 SELECT * FROM Payment ORDER BY PaymentDate DESC;
196
197 SELECT PaymentMethod, SUM(AmountPaid) AS TotalAmount
198 FROM Payment
199 GROUP BY PaymentMethod;
```

PaymentID	OrderID	CustomerID	PaymentMethod	AmountPaid	PaymentDate	PaymentStatus
5	5	1	Cash	50.00	2025-01-18	Completed
3	3	2	Credit Card	300.75	2025-01-17	Completed
2	2	3	Cash	120.00	2025-01-16	Completed
1	1	3	Credit Card	250.50	2025-01-15	Completed

Payment 23 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
52	13:36:09	SELECT * FROM Payment WHERE PaymentStatus = 'Completed' LIMIT 0, 1000	4 row(s) returned	0.015 sec / 0.000 sec

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
177
178 Execute the selected portion of the script or everything, if there is no selection
179
180 (2, 2, 3, 'Cash', 120.00, '2025-01-16', 'Pending');
181 (3, 3, 2, 'Credit Card', 300.75, '2025-01-17', 'Completed');
182 (4, 4, 2, 'Credit Card', 75.00, '2025-01-17', 'Refunded');
183 (5, 5, 1, 'Cash', 50.00, '2025-01-18', 'Completed');
184
185 UPDATE Payment
186 SET PaymentStatus = 'Completed'
187 WHERE PaymentID = 2;
188
189 DELETE FROM Payment
190 WHERE PaymentStatus = 'Refunded';
191
192 SELECT * FROM Payment;
```

PaymentID	OrderID	CustomerID	PaymentMethod	AmountPaid	PaymentDate	PaymentStatus
1	1	3	Credit Card	250.50	2025-01-15	Completed
2	2	3	Cash	120.00	2025-01-16	Completed
3	3	2	Credit Card	300.75	2025-01-17	Completed
5	5	1	Cash	50.00	2025-01-18	Completed

Payment 31 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
50	13:35:42	DELETE FROM Payment WHERE PaymentStatus = 'Refunded'	1 row(s) affected	0.000 sec
51	13:35:47	SELECT * FROM Payment LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```
191 SELECT * FROM Payment;
192
193 SELECT * FROM Payment WHERE PaymentStatus = 'Completed';
194
195 SELECT * FROM Payment ORDER BY PaymentDate DESC;
196
197 SELECT PaymentMethod, SUM(AmountPaid) AS TotalAmount
198 FROM Payment
199 GROUP BY PaymentMethod;
200
201 SELECT PaymentMethod, COUNT(*) AS PaymentCount
202 FROM Payment
203 GROUP BY PaymentMethod
204 HAVING PaymentCount > 1;
205
```

PaymentMethod	PaymentCount
Credit Card	2
Cash	2

Result 35 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
54	13:36:49	SELECT PaymentMethod, SUM(AmountPaid) AS TotalAmount FROM Payment GROUP BY PaymentMethod	2 row(s) returned	0.000 sec / 0.000 sec
55	13:37:17	SELECT PaymentMethod, COUNT(*) AS PaymentCount FROM Payment GROUP BY PaymentMethod HAVING PaymentCount > 1	2 row(s) returned	0.000 sec / 0.000 sec

# Screenshots of SQL commands and their results:

## Payment

projectDB\* pharmacy pharmacy pharmacy customers

Limit to 1000 rows

193 • SELECT PaymentMethod, SUM(AmountPaid) AS TotalAmount

194 FROM Payment

195 GROUP BY PaymentMethod;

196

197 • SELECT PaymentMethod, COUNT(\*) AS PaymentCount

198 FROM Payment

199 GROUP BY PaymentMethod

200 HAVING PaymentCount > 1;

201

202 • SELECT \*

203 FROM Payment

204 WHERE AmountPaid = (

205 SELECT MAX(AmountPaid)

206 FROM Payment

207 )

208

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

Filter Rows:

Edits

Export/Imports

Wrap Cell Contents

PaymentID	OrderID	CustomerID	PaymentMethod	AmountPaid	PaymentDate	PaymentStatus
3	3	2	Credit Card	300.75	2025-01-17	Completed

Result Grid

Form Editor

Payment 1

Apply

Revert

Context Help

Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
32	13:22:37	DELETE FROM Payment WHERE PaymentStatus = 'Refunded'	1 row(s) affected	0.000 sec
33	13:22:46	SELECT * FROM Payment WHERE AmountPaid = ( SELECT MAX(AmountPaid) FROM Payment ) LIMIT ...	1 row(s) returned	0.000 sec / 0.000 sec

# Screenshots of SQL commands and their results:

## Have

project01 pharmacy pharmacy pharmacy customers

Limit to 1000 rows

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

226 (5, 6, 75);

227

228 SELECT p.PharmacyName, m.MedicationName, h.StockQuantity

229 FROM Have h

230 JOIN Pharmacy p ON h.PharmacyID = p.PharmacyID

231 JOIN Medications m ON h.MedicationID = m.MedicationID;

232

233 SELECT m.MedicationName, h.StockQuantity

234 FROM Have h

235 JOIN Medications m ON h.MedicationID = m.MedicationID

236 WHERE h.PharmacyID = 1;

237

Result Grid

Filter Rows:

Export:

Wrap Cell Contents:

PharmacyName MedicationName StockQuantity

Nahdi Pharmacy Paracetamol 100

Nahdi Pharmacy Ibuprofen 50

Nahdi Pharmacy Amoxicillin 30

Nahdi Pharmacy Amoxicillin 200

Nahdi Pharmacy Antihistamine 150

Nahdi Pharmacy prozac 75

Result 2 x

Read Only Context Help Snippets

Output

Action Output

# Time Action Message Duration / Fetch

1 15:24:56 drop table Have 0 row(s) affected 0.016 sec

2 15:25:04 CREATE TABLE IF NOT EXISTS Have ( PharmacyID INT NOT NULL MedicationID INT NOT NULL, S... 0 row(s) affected 0.015 sec

3 15:25:10 INSERT INTO Have (PharmacyID, MedicationID, StockQuantity) VALUES (1, 1, 100), (1, 2, 50), (2, 4,... 6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0 0.000 sec

4 15:25:14 SELECT p.PharmacyName, m.MedicationName, h.StockQuantity FROM Have h JOIN Pharmacy p ON h.Pharm... 6 row(s) returned 0.000 sec / 0.000 sec

project01 pharmacy pharmacy pharmacy customers

Limit to 1000 rows

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

226 (5, 6, 75);

227

228 SELECT p.PharmacyName, m.MedicationName, h.StockQuantity

229 FROM Have h

230 JOIN Pharmacy p ON h.PharmacyID = p.PharmacyID

231 JOIN Medications m ON h.MedicationID = m.MedicationID;

232

233 SELECT m.MedicationName, h.StockQuantity

234 FROM Have h

235 JOIN Medications m ON h.MedicationID = m.MedicationID

236 WHERE h.PharmacyID = 1;

237

Result Grid

Filter Rows:

Export:

Wrap Cell Contents:

MedicationName StockQuantity

Paracetamol 100

Ibuprofen 50

Result 4 x

Read Only Context Help Snippets

Output

Action Output

# Time Action Message Duration / Fetch

1 15:24:56 drop table Have 0 row(s) affected 0.016 sec

2 15:25:04 CREATE TABLE IF NOT EXISTS Have ( PharmacyID INT NOT NULL MedicationID INT NOT NULL, S... 0 row(s) affected 0.015 sec

3 15:25:10 INSERT INTO Have (PharmacyID, MedicationID, StockQuantity) VALUES (1, 1, 100), (1, 2, 50), (2, 4,... 6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0 0.000 sec

4 15:25:14 SELECT p.PharmacyName, m.MedicationName, h.StockQuantity FROM Have h JOIN Pharmacy p ON h.Pharm... 6 row(s) returned 0.000 sec / 0.000 sec

5 15:25:39 SELECT m.MedicationName, h.StockQuantity FROM Have h JOIN Medications m ON h.MedicationID = m.Med... 2 row(s) returned 0.000 sec / 0.000 sec

6 15:25:17 SELECT m.MedicationName, h.StockQuantity FROM Have h JOIN Medications m ON h.MedicationID = m.Med... 2 row(s) returned 0.000 sec / 0.000 sec

# Screenshots of SQL commands and their results:

## Have

projectIDG - pharmacy pharmacy pharmacy customers

231 JOIN Medications m ON h.MedicationID = m.MedicationID;  
232  
233 \* SELECT m.MedicationName, h.StockQuantity  
234 FROM Have h  
235 JOIN Medications m ON h.MedicationID = m.MedicationID  
236 WHERE h.PharmacyID = 1;  
237  
238 \* SELECT p.PharmacyName, h.StockQuantity  
239 FROM Have h  
240 JOIN Pharmacy p ON h.PharmacyID = p.PharmacyID  
241 WHERE h.MedicationID = 4;  
242

SQLAdditions  
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid  
PharmacyName StockQuantity  
Nahd Pharmacy 30  
Nahd Pharmacy 200

Result 6 x  
Read Only Context Help Snippets

Output  
Action Output  
# Time Action Message Duration / Fetch  
3 15:25:10 INSERT INTO Have (PharmacyID, MedicationID, StockQuantity) VALUES (1, 1, 100), (1, 2, 50), (2, ... 6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0 0.000 sec  
4 15:25:14 SELECT p.PharmacyName, m.MedicationName, h.StockQuantity FROM Have h JOIN Pharmacy p ON h.Pharm... 6 row(s) returned 0.000 sec / 0.000 sec  
5 15:25:39 SELECT m.MedicationName, h.StockQuantity FROM Have h JOIN Medications m ON h.MedicationID = m.Me... 2 row(s) returned 0.000 sec / 0.000 sec  
6 15:26:17 SELECT m.MedicationName, h.StockQuantity FROM Have h JOIN Medications m ON h.MedicationID = m.Me... 2 row(s) returned 0.000 sec / 0.000 sec  
7 15:26:33 SELECT p.PharmacyName, h.StockQuantity FROM Have h JOIN Pharmacy p ON h.PharmacyID = p.Pharmac... 0 row(s) returned 0.000 sec / 0.000 sec  
8 15:26:53 SELECT p.PharmacyName, h.StockQuantity FROM Have h JOIN Pharmacy p ON h.PharmacyID = p.Pharmac... 2 row(s) returned 0.000 sec / 0.000 sec

projectIDG - pharmacy pharmacy pharmacy customers

237  
238 \* SELECT p.PharmacyName, h.StockQuantity  
239 FROM Have h  
240 JOIN Pharmacy p ON h.PharmacyID = p.PharmacyID  
241 WHERE h.MedicationID = 4;  
242  
243 \* SELECT m.MedicationName, SUM(h.StockQuantity) AS TotalStock  
244 FROM Have h  
245 JOIN Medications m ON h.MedicationID = m.MedicationID  
246 WHERE m.MedicationID = 1  
247 GROUP BY m.MedicationName

SQLAdditions  
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid  
MedicationName TotalStock  
Paracetamol 100

Result 7 x  
Read Only Context Help Snippets

Output  
Action Output  
# Time Action Message Duration / Fetch  
4 15:25:14 SELECT p.PharmacyName, m.MedicationName, h.StockQuantity FROM Have h JOIN Pharmacy p ON h.Pharm... 6 row(s) returned 0.000 sec / 0.000 sec  
5 15:25:39 SELECT m.MedicationName, h.StockQuantity FROM Have h JOIN Medications m ON h.MedicationID = m.Me... 2 row(s) returned 0.000 sec / 0.000 sec  
6 15:26:17 SELECT m.MedicationName, h.StockQuantity FROM Have h JOIN Medications m ON h.MedicationID = m.Me... 2 row(s) returned 0.000 sec / 0.000 sec  
7 15:26:33 SELECT p.PharmacyName, h.StockQuantity FROM Have h JOIN Pharmacy p ON h.PharmacyID = p.Pharmac... 0 row(s) returned 0.000 sec / 0.000 sec  
8 15:26:53 SELECT p.PharmacyName, h.StockQuantity FROM Have h JOIN Pharmacy p ON h.PharmacyID = p.Pharmac... 2 row(s) returned 0.000 sec / 0.000 sec  
9 15:27:12 SELECT m.MedicationName, SUM(h.StockQuantity) AS TotalStock FROM Have h JOIN Medications m ON h... 1 row(s) returned 0.000 sec / 0.000 sec



# Screenshots of SQL commands and their results:

## Output

pharmacy medications pharmacy customers customers customers orders payment						SQLAdditions
Output						
Action Output						
#	Time	Action	Message	Duration / Fetch		
1	13:54:45	CREATE SCHEMA IF NOT EXISTS Pharmacy	1 row(s) affected	0.250 sec		
2	13:54:48	USE pharmacy	0 row(s) affected	0.000 sec		
3	13:54:51	CREATE TABLE IF NOT EXISTS Pharmacy ( PharmacyID INT PRIMARY KEY, PharmacyName VARCHAR(50),	0 row(s) affected	0.047 sec		
4	13:54:58	INSERT INTO Pharmacy (PharmacyID, PharmacyName, Address, Phone, Email, LicenseNumber, OpeningHours) ...	1 row(s) affected	0.000 sec		
5	13:54:58	INSERT INTO Pharmacy (PharmacyID, PharmacyName, Address, Phone, Email, LicenseNumber, OpeningHours) ...	1 row(s) affected	0.000 sec		
6	13:54:59	INSERT INTO Pharmacy (PharmacyID, PharmacyName, Address, Phone, Email, LicenseNumber, OpeningHours) ...	1 row(s) affected	0.000 sec		
7	13:54:59	INSERT INTO Pharmacy (PharmacyID, PharmacyName, Address, Phone, Email, LicenseNumber, OpeningHours) ...	1 row(s) affected	0.016 sec		
8	13:54:59	INSERT INTO Pharmacy (PharmacyID, PharmacyName, Address, Phone, Email, LicenseNumber, OpeningHours) ...	1 row(s) affected	0.000 sec		
9	13:54:59	INSERT INTO Pharmacy (PharmacyID, PharmacyName, Address, Phone, Email, LicenseNumber, OpeningHours) ...	1 row(s) affected	0.016 sec		
10	13:55:00	UPDATE Pharmacy SET Phone = '111-2245' WHERE PharmacyID = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec		
11	13:55:12	DELETE FROM pharmacy WHERE PharmacyID = 6	1 row(s) affected	0.000 sec		
12	13:55:15	SELECT * FROM Pharmacy ORDER BY PharmacyID ASC LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec		
13	13:55:28	CREATE TABLE IF NOT EXISTS Medications ( MedicationID INT PRIMARY KEY, MedicationName VA...	0 row(s) affected	0.016 sec		
14	13:55:34	INSERT INTO Medications (MedicationID, MedicationName, Brand, Price, QuantityInStock, ExpiryDate, Phar...	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.016 sec		
15	13:55:41	UPDATE Medications SET Price = Price * 6.00 WHERE MedicationID = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec		
16	13:55:44	DELETE FROM Medications WHERE QuantityInStock < 30	1 row(s) affected	0.000 sec		
17	13:55:52	SELECT PharmacyID, COUNT(*) AS MedicationCount FROM Medications GROUP BY PharmacyID LIMIT 0, 1...	4 row(s) returned	0.000 sec / 0.000 sec		
18	13:56:05	SELECT * FROM Medications LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec		
19	13:56:11	CREATE TABLE IF NOT EXISTS Customers ( CustomerID INT PRIMARY KEY, FirstName VARCHAR(50),	0 row(s) affected	0.047 sec		
20	13:56:17	INSERT INTO Customers (CustomerID, FirstName, LastName, Email, PhoneNumber, Address, RegistrationDat...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.015 sec		
21	13:56:23	UPDATE Customers SET PhoneNumber = '0599880101' WHERE CustomerID = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec		
22	13:56:26	DELETE FROM Customers WHERE Email = 'hassan.omar@gmail.com'	1 row(s) affected	0.000 sec		
23	13:56:33	CREATE TABLE IF NOT EXISTS Orders ( OrderID INT PRIMARY KEY, CustomerID INT, PharmacyID...	0 row(s) affected	0.047 sec		
24	13:56:51	INSERT INTO Orders (OrderID, CustomerID, PharmacyID, OrderDate, OrderStatus, TotalAmount, Prescriptio...	8 row(s) affected Records: 8 Duplicates: 0 Warnings: 0	0.016 sec		
25	13:56:55	UPDATE Orders SET OrderStatus = 'Completed' WHERE OrderID = 7	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec		
26	13:57:00	UPDATE Orders SET OrderStatus = 'Cancelled' WHERE OrderID = 8	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec		
27	13:57:03	UPDATE Orders SET OrderStatus = 'Cancelled' WHERE OrderID = 5	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec		
28	13:57:06	UPDATE Orders SET OrderStatus = 'Cancelled' WHERE OrderID = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec		

pharmacy medications pharmacy customers customers customers orders payment						SQLAdditions
Output						
Action Output						
#	Time	Action	Message	Duration / Fetch		
19	13:56:11	CREATE TABLE IF NOT EXISTS Customers ( CustomerID INT PRIMARY KEY, FirstName VARCHAR(50),	0 row(s) affected	0.047 sec		
20	13:56:17	INSERT INTO Customers (CustomerID, FirstName, LastName, Email, PhoneNumber, Address, RegistrationDat...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.015 sec		
21	13:56:23	UPDATE Customers SET PhoneNumber = '0599880101' WHERE CustomerID = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec		
22	13:56:26	DELETE FROM Customers WHERE Email = 'hassan.omar@gmail.com'	1 row(s) affected	0.000 sec		
23	13:56:33	CREATE TABLE IF NOT EXISTS Orders ( OrderID INT PRIMARY KEY, CustomerID INT, PharmacyID...	0 row(s) affected	0.047 sec		
24	13:56:51	INSERT INTO Orders (OrderID, CustomerID, PharmacyID, OrderDate, OrderStatus, TotalAmount, Prescriptio...	8 row(s) affected Records: 8 Duplicates: 0 Warnings: 0	0.016 sec		
25	13:56:55	UPDATE Orders SET OrderStatus = 'Completed' WHERE OrderID = 7	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec		
26	13:57:00	UPDATE Orders SET OrderStatus = 'Cancelled' WHERE OrderID = 8	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec		
27	13:57:03	UPDATE Orders SET OrderStatus = 'Cancelled' WHERE OrderID = 5	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec		
28	13:57:06	UPDATE Orders SET OrderStatus = 'Cancelled' WHERE OrderID = 1	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec		
29	13:57:11	DELETE FROM Orders WHERE OrderID = 8	1 row(s) affected	0.000 sec		
30	13:57:14	SELECT * FROM Orders WHERE OrderStatus = 'Completed' LIMIT 0, 1000	5 row(s) returned	0.000 sec / 0.000 sec		
31	13:57:19	SELECT PharmacyID, COUNT(OrderID) AS NumberOfOrders FROM Orders GROUP BY PharmacyID LIMIT 0...	5 row(s) returned	0.000 sec / 0.000 sec		
32	13:57:26	SELECT PharmacyID, COUNT(OrderID) AS NumberOfOrders FROM Orders GROUP BY PharmacyID HAVING...	3 row(s) returned	0.000 sec / 0.000 sec		
33	13:57:29	SELECT * FROM Orders ORDER BY OrderDate DESC LIMIT 0, 1000	7 row(s) returned	0.000 sec / 0.016 sec		
34	13:57:37	SELECT * FROM Orders WHERE TotalAmount > ( SELECT AVG(TotalAmount) FROM Orders ) LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec		
35	13:58:01	SELECT Orders OrderID, Customers FirstName, Customers LastName, Pharmacy PharmacyName FROM Order...	7 row(s) returned	0.000 sec / 0.000 sec		
36	13:58:10	CREATE TABLE IF NOT EXISTS Payment ( PaymentID INT PRIMARY KEY, OrderID INT NOT NULL, ...	0 row(s) affected	0.078 sec		
37	13:58:14	INSERT INTO Payment (PaymentID, OrderID, CustomerID, PaymentMethod, AmountPaid, PaymentDate, Pay...	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.016 sec		
38	13:58:17	UPDATE Payment SET PaymentStatus = 'Completed' WHERE PaymentID = 2	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.000 sec		
39	13:58:20	DELETE FROM Payment WHERE PaymentStatus = 'Refunded'	1 row(s) affected	0.000 sec		
40	13:58:26	SELECT * FROM Payment LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec		
41	13:58:30	SELECT * FROM Payment WHERE PaymentStatus = 'Completed' LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec		
42	13:58:34	SELECT * FROM Payment ORDER BY PaymentDate DESC LIMIT 0, 1000	4 row(s) returned	0.016 sec / 0.000 sec		
43	13:58:42	SELECT PaymentMethod, SUM(AmountPaid) AS TotalAmount FROM Payment GROUP BY PaymentMethod L...	2 row(s) returned	0.000 sec / 0.000 sec		
44	13:58:46	SELECT PaymentMethod, COUNT(*) AS PaymentCount FROM Payment GROUP BY PaymentMethod HAVIN...	2 row(s) returned	0.000 sec / 0.000 sec		
45	13:58:53	SELECT p.PaymentID, CONCAT(c.FirstName, ' ', c.LastName) AS CustomerName, p.AmountPaid, p.Paym...	4 row(s) returned	0.000 sec / 0.000 sec		

2	15:25:04	CREATE TABLE IF NOT EXISTS Have ( PharmacyID INT NOT NULL, MedicationID INT NOT NULL, S...	0 row(s) affected	0.015 sec		
3	15:25:10	INSERT INTO Have (PharmacyID, MedicationID, StockQuantity) VALUES (1, 1, 100), (1, 2, 50), (2, 4...	6 row(s) affected Records: 6 Duplicates: 0 Warnings: 0	0.000 sec		
4	15:25:14	SELECT p.PharmacyName, m.MedicationName, h.StockQuantity FROM Have h JOIN Pharmacy p ON h.Pharm...	6 row(s) returned	0.000 sec / 0.000 sec		
5	15:25:39	SELECT m.MedicationName, h.StockQuantity FROM Have h JOIN Medications m ON h.MedicationID = m.Med...	2 row(s) returned	0.000 sec / 0.000 sec		
6	15:26:17	SELECT m.MedicationName, h.StockQuantity FROM Have h JOIN Medications m ON h.MedicationID = m.Med...	2 row(s) returned	0.000 sec / 0.000 sec		
7	15:26:33	SELECT p.PharmacyName, h.StockQuantity FROM Have h JOIN Pharmacy p ON h.PharmacyID = p.PharmacyI...	0 row(s) returned	0.000 sec / 0.000 sec		
8	15:26:53	SELECT p.PharmacyName, h.StockQuantity FROM Have h JOIN Pharmacy p ON h.PharmacyID = p.PharmacyI...	2 row(s) returned	0.000 sec / 0.000 sec		
9	15:27:12	SELECT m.MedicationName, SUM(h.StockQuantity) AS TotalStock FROM Have h JOIN Medications m ON h.M...	1 row(s) returned	0.000 sec / 0.000 sec		

# Task Table

Student Name	ID	Task
Taef Abd Ul-Wahab	44512026	Medications table/Business Rules
Raneem khalid alharbi	445002398	Pharmacy table/ER Model
Lana Nasser Alotaibi	445001397	Customers table /ER Model
Ghada Shujaa Alotibi	445003024	Orders table/Relational Schema
Retaj Khalid Alqurashi	44512037	Payment table/Relational Schema

