

# Creating Database

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- collage
- empinfo
- kirandb
- librarydb
- onlinestore
- prodinfo
- shopping
- stdinfo

Administration Schemas

Information

No object selected

```
1 • create database onlinestore;
2 • use onlinestore;
3
4 • CREATE TABLE Customers (
    customer_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    email VARCHAR(100) UNIQUE,
    phone VARCHAR(20),
    address TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
5
6
7
8
9
10
11
12
13 • CREATE TABLE Products (
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    description TEXT,
    price DECIMAL(10, 2),
    stock INT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
14
15
16
17
18
19
20
21
22 • CREATE TABLE Orders (
    order_id INT PRIMARY KEY AUTO_INCREMENT,
    customer_id INT,
    order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    status VARCHAR(50) DEFAULT 'Pending',
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
);
23
24
25
26
27
28
29
```

Object Info Session

Ready



## Navigator

## SCHEMAS

- Filter objects
- collage
- empinfo
- kirandb
- librarydb
- onlinestore
  - Tables
  - Views
  - Stored Procedures
  - Functions
- prodinfo
- shopping
  - Tables
  - Views
  - Stored Procedures
  - Functions
- stdinfo

Administration Schemas

## Information

No object selected

SQL File 1\* SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* LibraryManagementdb\* Online store database\* x

```
1 •  create database onlinestore;
2 •  use onlinestore;
3
4 •  CREATE TABLE Customers (
5     customer_id INT PRIMARY KEY AUTO_INCREMENT,
6     name VARCHAR(100),
7     email VARCHAR(100) UNIQUE,
8     phone VARCHAR(20),
9     address TEXT,
10    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
11 );
12
13 •  CREATE TABLE Products (
14     product_id INT PRIMARY KEY AUTO_INCREMENT,
15     name VARCHAR(100),
16     description TEXT,
17     price DECIMAL(10, 2),
18     stock INT,
19     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
20 );
21
22 •  CREATE TABLE Orders (
23     order_id INT PRIMARY KEY AUTO_INCREMENT,
24     customer_id INT,
25     order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
26     status VARCHAR(50) DEFAULT 'Pending',
27     FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
28 );
29
--
```

Object Info Session

Ready

# Creating Customers Table

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- collage
- empinfo
- kirandb
- librarydb
- onlinestore
  - Tables
  - Views
  - Stored Procedures
  - Functions
- prodinfo
- shopping
  - Tables
  - Views
  - Stored Procedures
  - Functions
- stdinfo

Administration Schemas

Information

No object selected

Object Info Session

Ready

SQL File 1\* SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* LibraryManagementdb\* Online store database\* ×

```
1 • create database onlinestore;
2 • use onlinestore;
3
4 • CREATE TABLE Customers (
    customer_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    email VARCHAR(100) UNIQUE,
    phone VARCHAR(20),
    address TEXT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
5
6
7
8
9
10
11
12
13 • CREATE TABLE Products (
    product_id INT PRIMARY KEY AUTO_INCREMENT,
    name VARCHAR(100),
    description TEXT,
    price DECIMAL(10, 2),
    stock INT,
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
14
15
16
17
18
19
20
21
22 • CREATE TABLE Orders (
    order_id INT PRIMARY KEY AUTO_INCREMENT,
    customer_id INT,
    order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
    status VARCHAR(50) DEFAULT 'Pending',
    FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
);
23
24
25
26
27
28
29
```

# Creating Products Table

The screenshot shows the MySQL Workbench interface with the 'Online store database\*' tab selected. In the central SQL editor pane, the code for creating the 'Products' table is highlighted. The table structure includes columns for product ID (auto-increment), name (VARCHAR(100)), description (TEXT), price (DECIMAL(10, 2)), stock (INT), and creation timestamp (TIMESTAMP). A foreign key constraint is also defined for the 'customer\_id' column in the 'Orders' table.

System tray icons include a chart showing 532281 +2.00%, a search bar, file explorer, Microsoft Teams, OneDrive, Microsoft 365, Task View, Firefox, WhatsApp, Telegram, Google Chrome, and a cursor icon.

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- collage
- empinfo
- kirandb
- librarydb
- onlinestore
  - Tables
  - Views
  - Stored Procedures
  - Functions
- prodinfo
- shopping
  - Tables
  - Views
  - Stored Procedures
  - Functions
- stdinfo

Administration Schemas

Information

No object selected

Object Info Session

Ready

SQL File 1\* SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* LibraryManagementdb\* Online store database\* ×

```
19     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
20 );
21
22 • CREATE TABLE Orders (
23     order_id INT PRIMARY KEY AUTO_INCREMENT,
24     customer_id INT,
25     order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
26     status VARCHAR(50) DEFAULT 'Pending',
27     FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
28 );
29
30
31 • CREATE TABLE Order_Items (
32     order_item_id INT PRIMARY KEY AUTO_INCREMENT,
33     order_id INT,
34     product_id INT,
35     quantity INT,
36     price DECIMAL(10, 2),
37     FOREIGN KEY (order_id) REFERENCES Orders(order_id),
38     FOREIGN KEY (product_id) REFERENCES Products(product_id)
39 );
40
41 • CREATE TABLE Payments (
42     payment_id INT PRIMARY KEY AUTO_INCREMENT,
43     order_id INT,
44     payment_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
45     amount DECIMAL(10, 2),
46     payment_method VARCHAR(50),
47     status VARCHAR(50),
```

# Creating Orders Table

The screenshot shows the MySQL Workbench interface with the 'Online store database\*' selected in the tabs. The central pane displays SQL code for creating three tables: Orders, Order\_Items, and Payments. The 'Orders' table has columns for order\_id (primary key, auto-increment), customer\_id, order\_date (timestamp, default current\_timestamp), status (varchar(50), default 'Pending'), and a foreign key reference to the 'Customers' table. The 'Order\_Items' table has columns for order\_item\_id (primary key, auto-increment), order\_id, product\_id, quantity, price (decimal(10, 2)), and foreign keys referencing the 'Orders' and 'Products' tables. The 'Payments' table has columns for payment\_id (primary key, auto-increment), order\_id, payment\_date (timestamp, default current\_timestamp), amount (decimal(10, 2)), payment\_method (varchar(50)), and status (varchar(50)). The 'Navigator' pane on the left shows the schema structure of the 'onlinestore' database, including its tables, views, stored procedures, and functions. The bottom taskbar includes icons for various Windows applications like File Explorer, Microsoft Edge, and Task View.

# Creating Order\_Item Table

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- Filter objects
- collage
- empinfo
- kirandb
- librarydb
- onlinestore
- Tables
- Views
- Stored Procedures
- Functions
- prodinfo
- shopping
- Tables
- Views
- Stored Procedures
- Functions
- stdinfo

Administration Schemas

No object selected

```
19     created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP
20 );
21
22 • CREATE TABLE Orders (
23     order_id INT PRIMARY KEY AUTO_INCREMENT,
24     customer_id INT,
25     order_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
26     status VARCHAR(50) DEFAULT 'Pending',
27     FOREIGN KEY (customer_id) REFERENCES Customers(customer_id)
28 );
29
30
31 • CREATE TABLE Order_Items (
32     order_item_id INT PRIMARY KEY AUTO_INCREMENT,
33     order_id INT,
34     product_id INT,
35     quantity INT,
36     price DECIMAL(10, 2),
37     FOREIGN KEY (order_id) REFERENCES Orders(order_id),
38     FOREIGN KEY (product_id) REFERENCES Products(product_id)
39 );
40
41 • CREATE TABLE Payments (
42     payment_id INT PRIMARY KEY AUTO_INCREMENT,
43     order_id INT,
44     payment_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
45     amount DECIMAL(10, 2),
46     payment_method VARCHAR(50),
47     status VARCHAR(50),
```

Object Info Session

Ready

# Creating Payment Table

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- collage
- empinfo
- kirandb
- librarydb
- onlinestore
- Tables
- Views
- Stored Procedures
- Functions

- prodinfo
- shopping
- Tables
- Views
- Stored Procedures
- Functions

- stdinfo

Administration Schemas

Information

No object selected

Object Info Session

Ready

SQL File 1\* SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* LibraryManagementdb\* Online store database\* ×

payment\_method VARCHAR(50),  
status VARCHAR(50),  
FOREIGN KEY (order\_id) REFERENCES Orders(order\_id)  
);

INSERT INTO Customers (name, email, phone, address)  
VALUES  
( 'Kumar', 'kumar@example.com', '2662794174', '57066 Ashley Shoal Apt. 365, West Courtney, NC 21844' ),  
( 'Prakash', 'prakash@example.com', '6830021942', '7423 Matthew Inlet, South Lori, MA 53587' ),  
( 'Pavan', 'pavan@example.com', '2586270068', '45145 Travis Villages Suite 723, South Shannonburgh, VT 78396' ),  
( 'Siva', 'siva@example.com', '3654347450', '51331 Hudson Gardens, Lake Amberstad, NY 54461' ),  
( 'Sai', 'sai@example.com', '7311703214', '10266 Daniels Flats, West Ryan, SC 96938' ),  
( 'Venkatesh', 'venkatesh@example.com', '5314346431', '75879 Sheila Motorway Apt. 138, West Nathan, OR 11966' ),  
( 'Ravi', 'ravi@example.com', '3522343715', '4842 Jacob Mill Apt. 604, East Amy, FL 02256' ),  
( 'Teja', 'teja@example.com', '2347431101', '3748 Martinez Locks, East Michelle, AR 41580' ),  
( 'Manoj', 'manoj@example.com', '3603671823', 'Unit 3550 Box 6011, DPO AA 96873' ),  
( 'Arjun', 'arjun@example.com', '7373709827', '9628 Jacob Port Apt. 076, New Matthew, GA 94091' ),  
( 'Ramesh', 'ramesh@example.com', '3381587407', '7576 Austin Key, South Janefurt, AZ 84744' ),  
( 'Mahesh', 'mahesh@example.com', '1017538743', '75263 Daniel Falls, West Lisa, OH 63058' ),  
( 'Srinu', 'srinu@example.com', '9736487854', '85806 Amanda View Apt. 186, South Melissa, PA 41282' ),  
( 'Naresh', 'naresh@example.com', '5944353873', '3265 Emily Extensions, North Jessica, WY 05327' ),  
( 'Naveen', 'naveen@example.com', '9511853231', '1988 Marcus Mount, West Davidmouth, NH 56061' ),  
( 'Lokesh', 'lokesh@example.com', '6328262812', '2219 Cynthia Drives, Port Jeremy, NM 06398' ),  
( 'Chaitanya', 'chaitanya@example.com', '8201046047', '78778 Davis Fords, West Nathan, ID 35076' ),  
( 'Anil', 'anil@example.com', '4525266525', '078 Michelle Parkway Suite 077, East Nathan, VT 93338' ),  
( 'Rajesh', 'rajesh@example.com', '5807026170', '92543 Ellis Plain, East Andrew, AK 62064' ),  
( 'Vinay', 'vinay@example.com', '2006255191', '8594 Graham Drive, South Samantha, KY 95286' );

73

74 • INSERT INTO Products (name, description, price, stock)

4 532281 +2.00% 01:19 PM 11-04-2025

Search

11:19 AM

ENG IN

11-04-2025

# Inserting Data to Customers

# Inserting Data to Products

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- collage
- empinfo
- kirandb
- librarydb
- onlinestore
  - Tables
  - Views
  - Stored Procedures
  - Functions
- prodinfo
- shopping
  - Tables
  - Views
  - Stored Procedures
  - Functions
- stdinfo

Administration Schemas

Information

No object selected

Object Info Session

Ready

SQL File 1\* SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* LibraryManagementdb\* Online store database\* ×

```
79     ('USB-C Hub', 'Multiport USB-C adapter', 1199.00, 200),
80     ('Webcam', '1080p HD webcam', 1499.00, 60),
81     ('Bluetooth Speaker', 'Portable Bluetooth speaker', 1799.00, 80),
82     ('Smartwatch', 'Fitness smartwatch', 2999.00, 40),
83     ('HD Monitor', '24-inch HD monitor', 8499.00, 30);
84
85
86 • INSERT INTO Orders (customer_id, order_date, status)
87 VALUES
88 (11, '2025-03-25 10:23:00', 'Completed'),
89 (18, '2025-02-14 15:45:00', 'Shipped'),
90 (4, '2025-03-01 09:10:00', 'Completed'),
91 (20, '2025-04-04 11:30:00', 'Pending'),
92 (5, '2025-02-20 08:00:00', 'Pending'),
93 (10, '2025-03-12 13:15:00', 'Completed'),
94 (16, '2025-02-12 16:40:00', 'Completed'),
95 (15, '2025-02-07 17:00:00', 'Shipped'),
96 (9, '2025-02-05 12:05:00', 'Shipped'),
97 (18, '2025-02-15 14:20:00', 'Completed'),
98 (8, '2025-03-11 10:10:00', 'Pending'),
99 (6, '2025-03-22 18:45:00', 'Pending'),
100 (14, '2025-03-06 09:35:00', 'Pending'),
101 (15, '2025-02-14 20:10:00', 'Pending'),
102 (19, '2025-03-19 11:50:00', 'Pending'),
103 (5, '2025-03-11 12:20:00', 'Shipped'),
104 (4, '2025-04-01 07:55:00', 'Pending'),
105 (6, '2025-02-28 10:05:00', 'Shipped'),
106 (15, '2025-03-23 17:25:00', 'Shipped'),
107 (12, '2025-02-26 14:45:00', 'Pending');
```

# Inserting Data to Orders

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- collage
- empinfo
- kirandb
- librarydb
- onlinestore
  - Tables
  - Views
  - Stored Procedures
  - Functions
- prodinfo
- shopping
  - Tables
  - Views
  - Stored Procedures
  - Functions
- stdinfo

Administration Schemas

Information

No object selected

```
109
110 • INSERT INTO Order_Items (order_id, product_id, quantity, price)
111 VALUES
112 (1, 8, 2, 16998.00),
113 (1, 8, 3, 25497.00),
114 (2, 2, 1, 2299.00),
115 (2, 6, 3, 5397.00),
116 (3, 4, 3, 3597.00),
117 (4, 2, 1, 2299.00),
118 (5, 1, 2, 1398.00),
119 (5, 7, 1, 2999.00),
120 (6, 4, 2, 2398.00),
121 (6, 2, 3, 6897.00),
122 (6, 3, 1, 999.00),
123 (7, 6, 3, 5397.00),
124 (7, 5, 1, 1499.00),
125 (7, 1, 2, 1398.00),
126 (8, 5, 2, 2998.00),
127 (9, 3, 3, 2997.00),
128 (9, 7, 1, 2999.00),
129 (10, 1, 1, 699.00),
130 (10, 4, 2, 2398.00),
131 (10, 5, 1, 1499.00);

132
133
134
135 • INSERT INTO Payments (order_id, payment_date, amount, payment_method, status)
136 VALUES
137 (1, '2025-03-27', 600.0, 'Credit Card', 'Completed'),
```

Object Info Session

Ready

# Inserting Data to Order\_Items

# Inserting Data to Payments

MySQL Workbench - Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- collage
- empinfo
- kirandb
- librarydb
- onlinestore
- Tables
- Views
- Stored Procedures
- Functions

- prodinfo
- shopping
- Tables
- Views
- Stored Procedures
- Functions

- stdinfo

Administration Schemas

Information

No object selected

```
127 (9, 3, 3, 2997.00),
128 (9, 7, 1, 2999.00),
129 (10, 1, 1, 699.00),
130 (10, 4, 2, 2398.00),
131 (10, 5, 1, 1499.00);
132
133
134
135 • INSERT INTO Payments (order_id, payment_date, amount, payment_method, status)
VALUES
137 (1, '2025-03-27', 600.0, 'Credit Card', 'Completed'),
138 (3, '2025-03-01', 60.0, 'Net Banking', 'Completed'),
139 (6, '2025-03-13', 232.5, 'Credit Card', 'Completed'),
140 (7, '2025-02-16', 197.97, 'Net Banking', 'Completed'),
141 (10, '2025-02-16', 105.99, 'Credit Card', 'Completed');
142
143
144 • select * from customers;
145 • select * from products;
146 • select * from Orders;
147 • select * from Order_items;
148 • select * from Payments;
149
150 • INSERT INTO Orders (customer_id, order_date, status)
VALUES (5, NOW(), 'Pending');
152
153 • INSERT INTO Order_Items (order_id, product_id, quantity, price)
VALUES
155 (21, 1, 2, 1398.00),
---
```

Object Info Session

Ready

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- collage
- empinfo
- kirandb
- librarydb
- onlinestore
- Tables
- Views
- Stored Procedures
- Functions

- prodinfo
- shopping
- Tables
- Views
- Stored Procedures
- Functions

- stdinfo

Administration Schemas

Information

No object selected

Object Info Session

Ready

SQL File 1\* SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* LibraryManagementdb\* Online store database\* ×

```
160 -- View All Orders with Customer Info
161 • SELECT
162     o.order_id AS order_id,
163     c.name AS customer_name,
164     o.order_date,
165     o.status
166 FROM Orders AS o
167 JOIN Customers AS c ON o.customer_id = c.customer_id
168 ORDER BY o.order_date DESC;

169
170 -- get all orders for a particular customer
171 • SELECT o.order_id, o.order_date, o.status, SUM(oi.quantity * oi.price) AS total
172 FROM Orders o
173 JOIN Order_Items oi ON o.order_id = oi.order_id
174 WHERE o.customer_id = 5
175 GROUP BY o.order_id, o.order_date, o.status;

176
177
178 -- get product stock after orders
179 • SELECT p.name, p.stock - IFNULL(SUM(oi.quantity), 0) AS remaining_stock
180 FROM Products p
181 LEFT JOIN Order_Items oi ON p.product_id = oi.product_id
182 GROUP BY p.product_id;

183
184
185 -- monthly revenue
186 • SELECT
187     DATE_FORMAT(o.order_date, '%Y-%m') AS month,
188     SUM(oi.price) AS total_revenue
189
```

# View All Orders with Customer Info

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- collage
- empinfo
- kirandb
- librarydb
- onlinestore
- Tables
- Views
- Stored Procedures
- Functions

- prodinfo
- shopping
- Tables
- Views
- Stored Procedures
- Functions

- stdinfo

Administration Schemas

Information

No object selected

Object Info Session

Ready

SQL File 1\* SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* LibraryManagementdb\* Online store database\* ×

```
160 -- View All Orders with Customer Info
161 • SELECT
162     o.order_id AS order_id,
163     c.name AS customer_name,
164     o.order_date,
165     o.status
166     FROM Orders AS o
167     JOIN Customers AS c ON o.customer_id = c.customer_id
168     ORDER BY o.order_date DESC;
169
170 -- get all orders for a particular customer
171 • SELECT o.order_id, o.order_date, o.status, SUM(oi.quantity * oi.price) AS total
172     FROM Orders o
173     JOIN Order_Items oi ON o.order_id = oi.order_id
174     WHERE o.customer_id = 5
175     GROUP BY o.order_id, o.order_date, o.status;
176
177
178 -- get product stock after orders
179 • SELECT p.name, p.stock - IFNULL(SUM(oi.quantity), 0) AS remaining_stock
180     FROM Products p
181     LEFT JOIN Order_Items oi ON p.product_id = oi.product_id
182     GROUP BY p.product_id;
183
184
185 -- monthly revenue
186 • SELECT
187     DATE_FORMAT(o.order_date, '%Y-%m') AS month,
188     SUM(oi.price) AS total_revenue
189
```

# Getting Orders for a Particular Customer

532281 +2.00%

Search

11:22 PM 11-04-2025

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- collage
- empinfo
- kirandb
- librarydb
- onlinestore
  - Tables
  - Views
  - Stored Procedures
  - Functions
- prodinfo
- shopping
  - Tables
  - Views
  - Stored Procedures
  - Functions
- stdinfo

Administration Schemas

Information

No object selected

Object Info Session

Ready

SQL File 1\* SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* LibraryManagementdb\* Online store database\* ×

```
166     FROM Orders AS o
167     JOIN Customers AS c ON o.customer_id = c.customer_id
168     ORDER BY o.order_date DESC;
169
170     -- get all orders for a particular customer
171 •   SELECT o.order_id, o.order_date, o.status, SUM(oi.quantity * oi.price) AS total
172     FROM Orders o
173     JOIN Order_Items oi ON o.order_id = oi.order_id
174     WHERE o.customer_id = 5
175     GROUP BY o.order_id, o.order_date, o.status;
176
177
178     -- get product stock after orders
179 •   SELECT p.name, p.stock - IFNULL(SUM(oi.quantity), 0) AS remaining_stock
180     FROM Products p
181     LEFT JOIN Order_Items oi ON p.product_id = oi.product_id
182     GROUP BY p.product_id;
183
184
185     -- monthly revenue
186 •   SELECT
187         DATE_FORMAT(o.order_date, '%Y-%m') AS month,
188         SUM(oi.price) AS total_revenue
189     FROM Orders o
190     JOIN Order_Items oi ON o.order_id = oi.order_id
191     WHERE o.status = 'Completed'
192     GROUP BY month
193     ORDER BY month DESC;
194
```

# Getting Products Stock After Orders

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- collage
- empinfo
- kirandb
- librarydb
- onlinestore
  - Tables
  - Views
  - Stored Procedures
  - Functions
- prodinfo
- shopping
  - Tables
  - Views
  - Stored Procedures
  - Functions
- stdinfo

Administration Schemas

Information

No object selected

Object Info Session

Ready

SQL File 1\* SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* LibraryManagementdb\* Online store database\* ×

```
175     GROUP BY o.order_id, o.order_date, o.status;
```

```
176
```

```
177
```

```
178     -- get product stock after orders
```

```
179 •  SELECT p.name, p.stock - IFNULL(SUM(oi.quantity), 0) AS remaining_stock
```

```
180     FROM Products p
```

```
181     LEFT JOIN Order_Items oi ON p.product_id = oi.product_id
```

```
182     GROUP BY p.product_id;
```

```
183
```

```
184
```

```
185     -- monthly revenue
```

```
186 •  SELECT
```

```
187         DATE_FORMAT(o.order_date, '%Y-%m') AS month,
```

```
188         SUM(oi.price) AS total_revenue
```

```
189     FROM Orders o
```

```
190     JOIN Order_Items oi ON o.order_id = oi.order_id
```

```
191     WHERE o.status = 'Completed'
```

```
192     GROUP BY month
```

```
193     ORDER BY month DESC;
```

```
194
```

```
195
```

```
196     -- top 5 customers by spend
```

```
197 •  SELECT
```

```
198     c.customer_id,
```

```
199     c.name,
```

```
200     c.email,
```

```
201     SUM(oi.price) AS total_spent
```

```
202     FROM Customers c
```

```
203     JOIN Orders o ON c.customer_id = o.customer_id
```

532281 +2.00%

Search

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203

# Monthly Revenue

01:23 PM  
11-04-2025

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- collage
- empinfo
- kirandb
- librarydb
- onlinestore
  - Tables
  - Views
  - Stored Procedures
  - Functions
- prodinfo
- shopping
  - Tables
  - Views
  - Stored Procedures
  - Functions
- stdinfo

Administration Schemas

Information

No object selected

Object Info Session

Ready

SQL File 1\* SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* LibraryManagementdb\* Online store database\* ×

193 ORDER BY month DESC;

194

195

196 -- top 5 customers by spend

197 • SELECT c.customer\_id,  
c.name,  
c.email,  
SUM(oi.price) AS total\_spent

202 FROM Customers c

203 JOIN Orders o ON c.customer\_id = o.customer\_id

204 JOIN Order\_Items oi ON o.order\_id = oi.order\_id

205 WHERE o.status = 'Completed'

206 GROUP BY c.customer\_id

207 ORDER BY total\_spent DESC

208 LIMIT 5;

209

210

211 -- view all pending orders

212 • SELECT o.order\_id,  
c.name AS customer\_name,  
o.order\_date,  
o.status

217 FROM Orders o

218 JOIN Customers c ON o.customer\_id = c.customer\_id

219 WHERE o.status = 'Pending'

220 ORDER BY o.order\_date;

221

4 532281 +2.00%

Search

M365

01:23 PM  
11-04-2025

# Top 5 Customers By Spend

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- collage
- empinfo
- kirandb
- librarydb
- onlinestore
  - Tables
  - Views
  - Stored Procedures
  - Functions
- prodinfo
- shopping
  - Tables
  - Views
  - Stored Procedures
  - Functions
- stdinfo

Administration Schemas

Information

No object selected

```
208     LIMIT 5;
```

```
209
```

```
210
```

```
211     -- view all pending orders
```

```
212 •   SELECT
213     o.order_id,
214     c.name AS customer_name,
215     o.order_date,
216     o.status
217     FROM Orders o
218     JOIN Customers c ON o.customer_id = c.customer_id
219     WHERE o.status = 'Pending'
220     ORDER BY o.order_date;
```

```
221
```

```
222
```

```
223     -- top selling products
```

```
224 •   SELECT
225     p.name,
226     SUM(oi.quantity) AS total_units_sold
227     FROM Order_Items oi
228     JOIN Products p ON oi.product_id = p.product_id
229     GROUP BY p.product_id
230     ORDER BY total_units_sold DESC
231     LIMIT 5;
```

```
232
```

```
233
```

```
234     -- total amount for a specific order
```

```
235 •   SELECT
236     SUM(price) AS total_amount
```

Object Info Session

Ready

# View All Pending Orders

# Top Selling Products

```
208    LIMIT 5;
```

```
209
```

```
210
```

```
211    -- view all pending orders
```

```
212 •   SELECT
```

```
213     o.order_id,
```

```
214     c.name AS customer_name,
```

```
215     o.order_date,
```

```
216     o.status
```

```
217     FROM Orders o
```

```
218     JOIN Customers c ON o.customer_id = c.customer_id
```

```
219     WHERE o.status = 'Pending'
```

```
220     ORDER BY o.order_date;
```

```
221
```

```
222
```

```
223    -- top selling products
```

```
224 •   SELECT
```

```
225     p.name,
```

```
226     SUM(oi.quantity) AS total_units_sold
```

```
227     FROM Order_Items oi
```

```
228     JOIN Products p ON oi.product_id = p.product_id
```

```
229     GROUP BY p.product_id
```

```
230     ORDER BY total_units_sold DESC
```

```
231     LIMIT 5;
```

```
232
```

```
233
```

```
234    -- total amount for a specific order
```

```
235 •   SELECT
```

```
236     SUM(price) AS total_amount
```

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- collage
- empinfo
- kirandb
- librarydb
- onlinestore
- Tables
- Views
- Stored Procedures
- Functions

- prodinfo
- shopping
- Tables
- Views
- Stored Procedures
- Functions

- stdinfo

Administration Schemas

Information

No object selected

Object Info Session

Ready

SQL File 1\* SQL File 3\* SQL File 4\* SQL File 5\* SQL File 6\* SQL File 7\* SQL File 8\* SQL File 9\* LibraryManagementdb\* Online store database\* ×

```
212 • SELECT
213     o.order_id,
214     c.name AS customer_name,
215     o.order_date,
216     o.status
217     FROM Orders o
218     JOIN Customers c ON o.customer_id = c.customer_id
219     WHERE o.status = 'Pending'
220     ORDER BY o.order_date;
221
222
223     -- top selling products
224 • SELECT
225     p.name,
226     SUM(oi.quantity) AS total_units_sold
227     FROM Order_Items oi
228     JOIN Products p ON oi.product_id = p.product_id
229     GROUP BY p.product_id
230     ORDER BY total_units_sold DESC
231     LIMIT 5;
232
233
234     -- total amount for a specific order
235 • SELECT
236     SUM(price) AS total_amount
237     FROM Order_Items
238     WHERE order_id = 21;
239
240
```

# Total Amount For a Specific Order

532281 +2.00%

Search

11-04-2025 01:23 PM ENG IN