

```
create database online_sales;
```

```
show databases;
```

```
use online_sales;
```

```
create table orders(
```

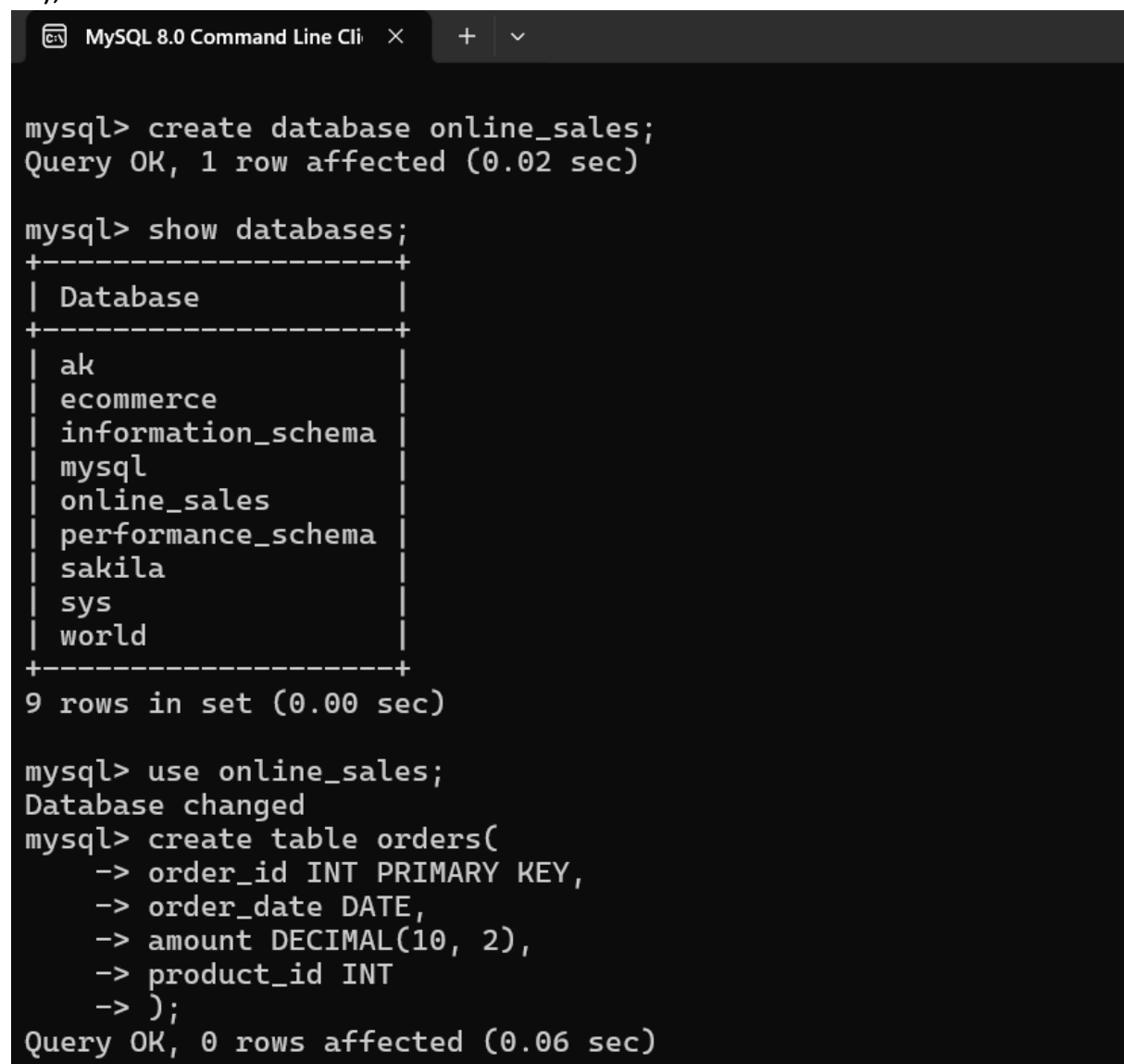
```
    order_id INT PRIMARY KEY,
```

```
    order_date DATE,
```

```
    amount DECIMAL(10, 2),
```

```
    product_id INT
```

```
);
```



```
MySQL 8.0 Command Line Cli  ×  +  v

mysql> create database online_sales;
Query OK, 1 row affected (0.02 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| ak       |
| ecommerce |
| information_schema |
| mysql    |
| online_sales |
| performance_schema |
| sakila   |
| sys      |
| world    |
+-----+
9 rows in set (0.00 sec)

mysql> use online_sales;
Database changed
mysql> create table orders(
    -> order_id INT PRIMARY KEY,
    -> order_date DATE,
    -> amount DECIMAL(10, 2),
    -> product_id INT
    -> );
Query OK, 0 rows affected (0.06 sec)
```

INSERT INTO orders (order_id, order_date, amount, product_id) VALUES

(1, '2023-01-15', 150.00, 101), (2, '2023-01-20', 200.00, 102), (3, '2023-02-10', 300.00, 103), (4, '2023-02-12', 450.00, 101), (5, '2023-03-05', 500.00, 104), (6, '2023-03-18', 100.00, 105), (7, '2024-01-22', 350.00, 106), (8, '2024-02-14', 250.00, 102), (9, '2024-02-25', 400.00, 107), (10, '2024-03-01', 600.00, 108), (11, '2024-03-15', 700.00, 109), (12, '2024-03-28', 300.00, 110);

select * from orders;

```
mysql> INSERT INTO orders (order_id, order_date, amount, product_id) VALUES
-> (1, '2023-01-15', 150.00, 101),
-> (2, '2023-01-20', 200.00, 102),
-> (3, '2023-02-10', 300.00, 103),
-> (4, '2023-02-12', 450.00, 101),
-> (5, '2023-03-05', 500.00, 104),
-> (6, '2023-03-18', 100.00, 105),
-> (7, '2024-01-22', 350.00, 106),
-> (8, '2024-02-14', 250.00, 102),
-> (9, '2024-02-25', 400.00, 107),
-> (10, '2024-03-01', 600.00, 108),
-> (11, '2024-03-15', 700.00, 109),
-> (12, '2024-03-28', 300.00, 110);
```

Query OK, 12 rows affected (0.01 sec)
Records: 12 Duplicates: 0 Warnings: 0

```
mysql> select * from orders;
```

order_id	order_date	amount	product_id
1	2023-01-15	150.00	101
2	2023-01-20	200.00	102
3	2023-02-10	300.00	103
4	2023-02-12	450.00	101
5	2023-03-05	500.00	104
6	2023-03-18	100.00	105
7	2024-01-22	350.00	106
8	2024-02-14	250.00	102
9	2024-02-25	400.00	107
10	2024-03-01	600.00	108
11	2024-03-15	700.00	109
12	2024-03-28	300.00	110

12 rows in set (0.00 sec)

select * from orders limit 10; select order_date,

-> YEAR(order_date) as order_year,

-> MONTH(order_date) as order_month from orders limit 10;

```
mysql> select * from orders limit 10;
```

order_id	order_date	amount	product_id
1	2023-01-15	150.00	101
2	2023-01-20	200.00	102
3	2023-02-10	300.00	103
4	2023-02-12	450.00	101
5	2023-03-05	500.00	104
6	2023-03-18	100.00	105
7	2024-01-22	350.00	106
8	2024-02-14	250.00	102
9	2024-02-25	400.00	107
10	2024-03-01	600.00	108

```
10 rows in set (0.00 sec)
```

```
mysql> select order_date,  
-> YEAR(order_date) as order_year,  
-> MONTH(order_date) as order_month from orders limit 10;
```

order_date	order_year	order_month
2023-01-15	2023	1
2023-01-20	2023	1
2023-02-10	2023	2
2023-02-12	2023	2
2023-03-05	2023	3
2023-03-18	2023	3
2024-01-22	2024	1
2024-02-14	2024	2
2024-02-25	2024	2
2024-03-01	2024	3

```
10 rows in set (0.00 sec)
```

```
select sum(amount) as total_revenue,
```

```
-> count( distinct order_id) as total_order_volume from orders;
```

```
mysql> select sum(amount) as total_revenue,  
-> count( distinct order_id) as total_order_volume from orders;
```

total_revenue	total_order_volume
4300.00	12

```
1 row in set (0.01 sec)
```

select YEAR(order_date) as order_year,

-> MONTH(order_date) as order_month from orders group by

YEAR(order_date), MONTH(order_date);

```
mysql> select order_year, order_month from orders;
ERROR 1054 (42S22): Unknown column 'order_year' in 'field list'
mysql> select YEAR(order_date) as order_year,
-> MONTH(order_date) as order_month from orders group by YEAR(order_date), MONTH(order_date);
+-----+-----+
| order_year | order_month |
+-----+-----+
| 2023      | 1          |
| 2023      | 2          |
| 2023      | 3          |
| 2024      | 1          |
| 2024      | 2          |
| 2024      | 3          |
+-----+-----+
6 rows in set (0.00 sec)
```

select YEAR(order_date) as order_year,

-> MONTH(order_date) as order_month,

-> sum(amount) as total_revenue,

-> count(distinct order_id) as total_order_volume from orders group by
YEAR(order_date), MONTH(order_date);

select YEAR(order_date) as order_year,

-> MONTH(order_date) as order_month,

-> sum(amount) as total_revenue,

-> count(distinct order_id) as total_order_volume from orders group by
YEAR(order_date), MONTH(order_date)

-> order by order_year, order_month;

```
mysql> select YEAR(order_date) as order_year,
-> MONTH(order_date) as order_month,
-> sum(amount) as total_revenue,
-> count( distinct order_id) as total_order_volume from orders group by YEAR(order_date), MONTH(order_date);
```

order_year	order_month	total_revenue	total_order_volume
2023	1	350.00	2
2023	2	750.00	2
2023	3	600.00	2
2024	1	350.00	1
2024	2	650.00	2
2024	3	1600.00	3

6 rows in set (0.00 sec)

```
mysql> select YEAR(order_date) as order_year,
-> MONTH(order_date) as order_month,
-> sum(amount) as total_revenue,
-> count( distinct order_id) as total_order_volume from orders group by YEAR(order_date), MONTH(order_date)
-> order by order_year, order_month;
```

order_year	order_month	total_revenue	total_order_volume
2023	1	350.00	2
2023	2	750.00	2
2023	3	600.00	2
2024	1	350.00	1
2024	2	650.00	2
2024	3	1600.00	3

6 rows in set (0.00 sec)

select YEAR(order_date) as order_year,

-> MONTH(order_date) as order_month,

-> sum(amount) as total_revenue,

-> count(distinct order_id) as total_order_volume from orders where
order_date between '2023-01-01' and '2024-12-31' group by
YEAR(order_date), MONTH(order_date)

-> order by order_year, order_month;

```
mysql> select YEAR(order_date) as order_year,  
-> MONTH(order_date) as order_month,  
-> sum(amount) as total_revenue,  
-> count(distinct order_id) as total_order_volume from orders where order_date between '2023-01-01' and '2024-12-31' group by YEAR(order_da  
te), MONTH(order_date)  
-> order by order_year, order_month;
```

order_year	order_month	total_revenue	total_order_volume
2023	1	350.00	2
2023	2	750.00	2
2023	3	600.00	2
2024	1	350.00	1
2024	2	650.00	1
2024	3	1600.00	3

6 rows in set (0.00 sec)

```
mysql>
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the 'SCHEMAS' tree with 'online_sales' expanded, showing 'orders' as a table. The central query editor contains the SQL query: `SELECT * FROM online_sales.orders;`. Below the query editor, the 'Result Grid' displays 12 rows of data with columns: order_id, order_date, amount, and product_id. The bottom panel shows the 'Output' tab with 'Action Output' selected, displaying a log of three SQL actions and their execution times. A right-hand pane contains a message about disabled automatic context help.

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: orders x orders

Limit to 1000 rows

1 • SELECT * FROM online_sales.orders;

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

Result Grid

order_id	order_date	amount	product_id
1	2023-01-15	150.00	101
2	2023-01-20	200.00	102
3	2023-02-10	300.00	103
4	2023-02-12	450.00	101
5	2023-03-05	500.00	104
6	2023-03-18	100.00	105
7	2024-01-22	350.00	106
8	2024-02-14	250.00	102
9	2024-02-25	400.00	107
10	2024-03-01	600.00	108
11	2024-03-15	700.00	109
12	2024-03-28	300.00	110

orders 1 x

Apply Revert Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	13:20:36	SELECT * FROM online_sales.orders LIMIT 0, 1000	12 row(s) returned	0.015 sec / 0.000 sec
2	13:21:27	SELECT * FROM online_sales.orders LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec
3	13:21:58	SELECT YEAR(order_date) AS order_year, MONTH(order_date) AS o...	6 row(s) returned	0.000 sec / 0.000 sec

Table: orders

Columns:

- order_id int PK
- order_date date
- amount decimal(10,2)
- product_id int

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

- ak
- ecommerce
- online_sales
 - Tables
 - orders
 - Views
 - Stored Procedures
 - Functions
- sakila
- sys
- world

orders orders x

Limit to 1000 rows

```
1 SELECT
2   YEAR(order_date) AS order_year,
3   MONTH(order_date) AS order_month,
4   SUM(amount) AS total_revenue,
5   COUNT(DISTINCT order_id) AS order_volume
6 FROM online_sales.orders
7 WHERE order_date BETWEEN '2023-01-01' AND '2024-12-31'
```

Result Grid

	order_year	order_month	total_revenue	order_volume
▶	2023	1	350.00	2
	2023	2	750.00	2
	2023	3	600.00	2
	2024	1	350.00	1
	2024	2	650.00	2
	2024	3	1600.00	3

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

Result 2 x

Read Only

Context Help Shippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	13:20:36	SELECT * FROM online_sales.orders LIMIT 0, 1000	12 row(s) returned	0.015 sec / 0.000 sec
2	13:21:27	SELECT * FROM online_sales.orders LIMIT 0, 1000	12 row(s) returned	0.000 sec / 0.000 sec
3	13:21:58	SELECT YEAR(order_date) AS order_year, MONTH(order_date) AS o...	6 row(s) returned	0.000 sec / 0.000 sec

Table: orders

Columns:

- order_id int PK
- order_date date
- amount decimal(10,2)
- product_id int