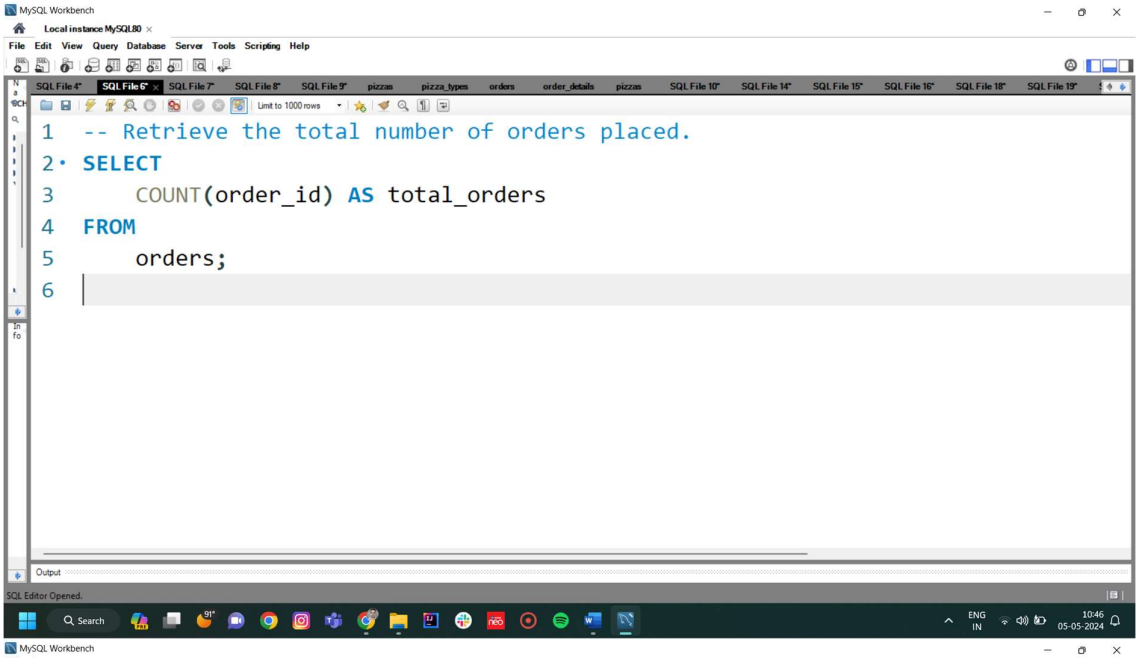
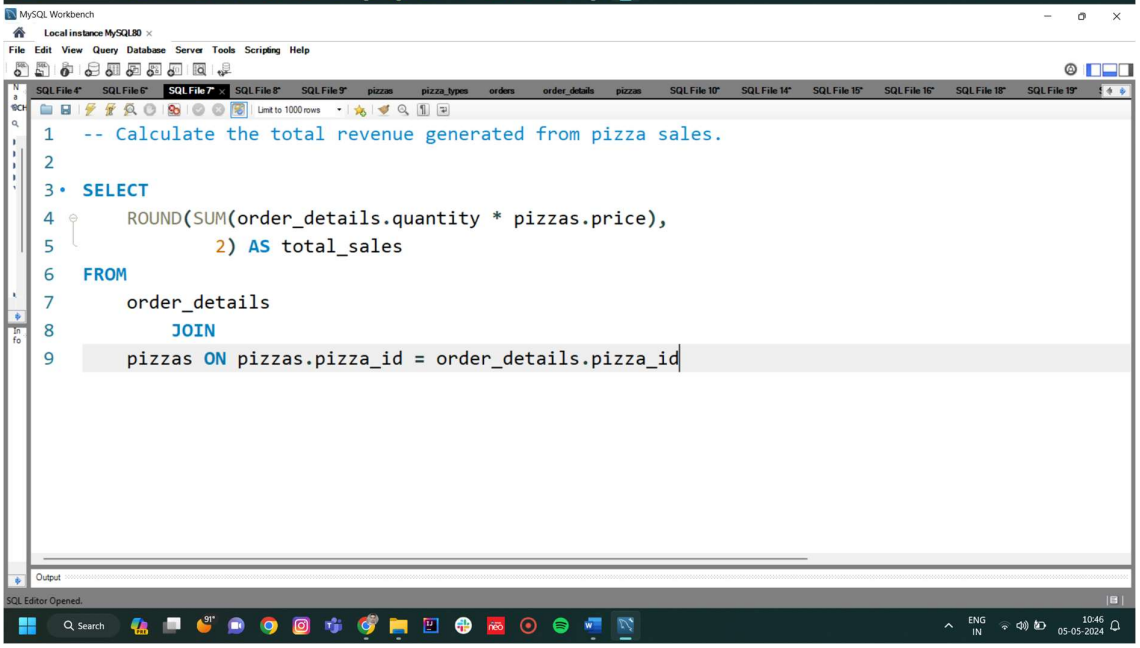
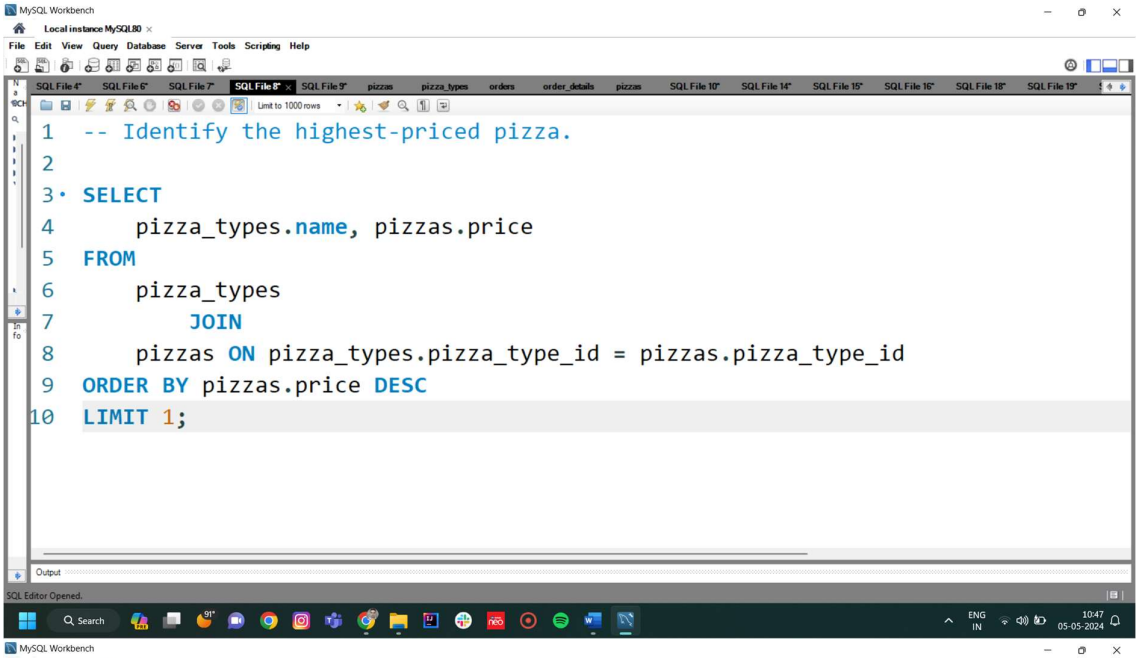
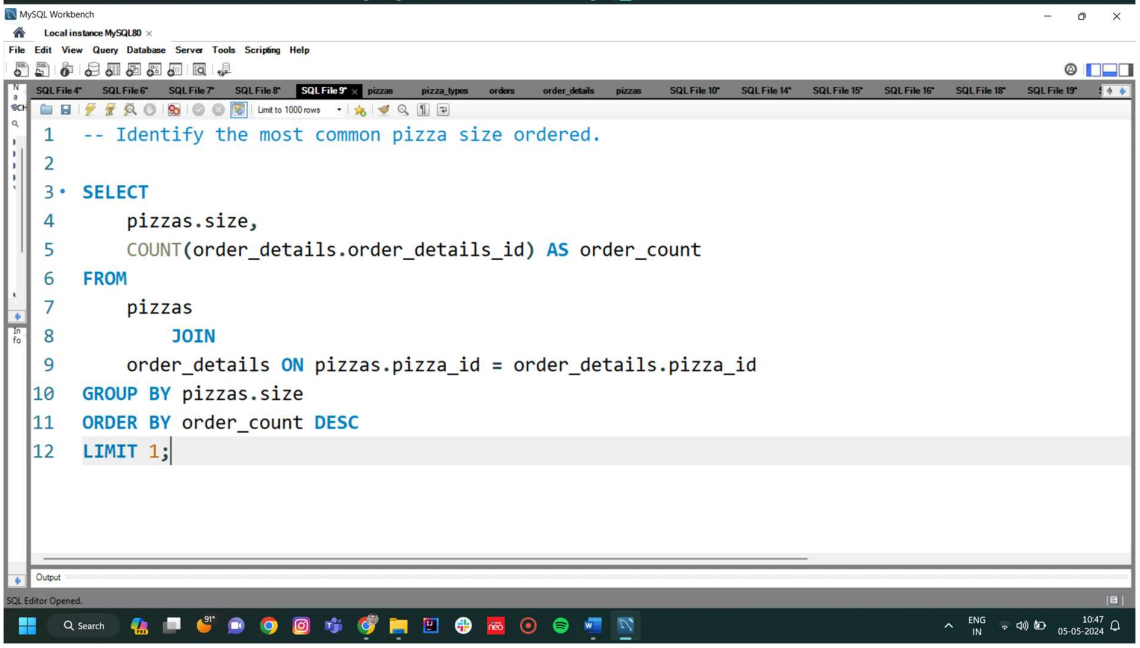
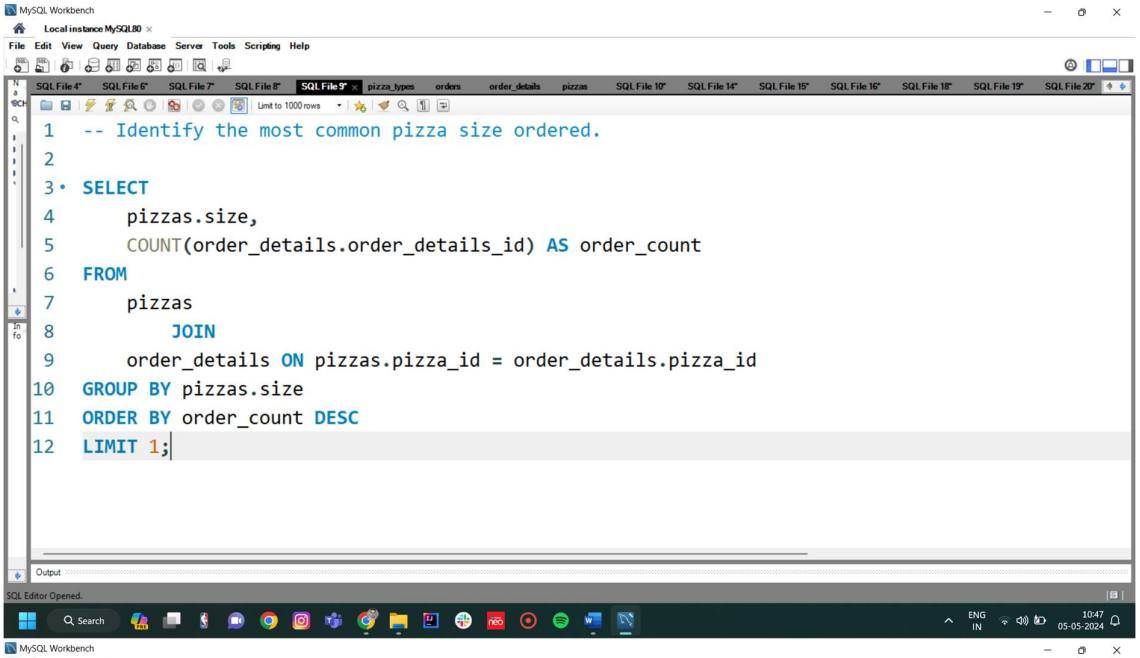
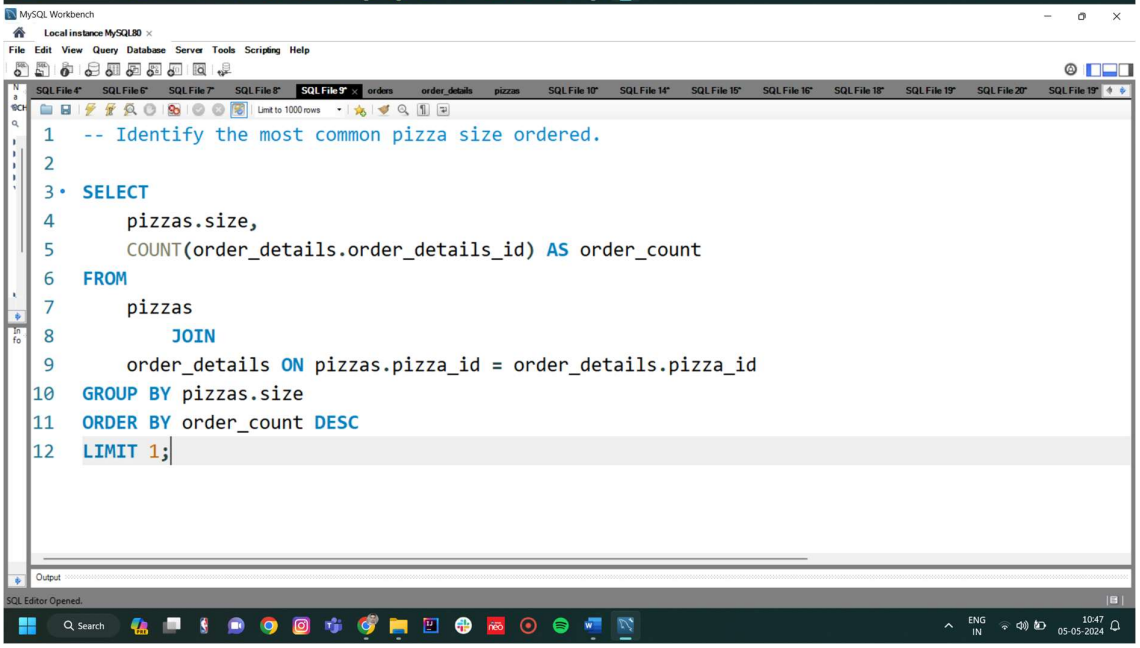

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 6* SQL File 7* SQL File 8* SQL File 9* pizzas pizza_types orders order_details pizzas SQL File 10* SQL File 14* SQL File 15* SQL File 16* SQL File 18* SQL File 19*
Limit to 1000 rows
1 -- Retrieve the total number of orders placed.
2 • SELECT
3 COUNT(order_id) AS total_orders
4 FROM
5 orders;
6
Output
SQL Editor Opened.
Search
ENG IN
10:46 05-05-2024

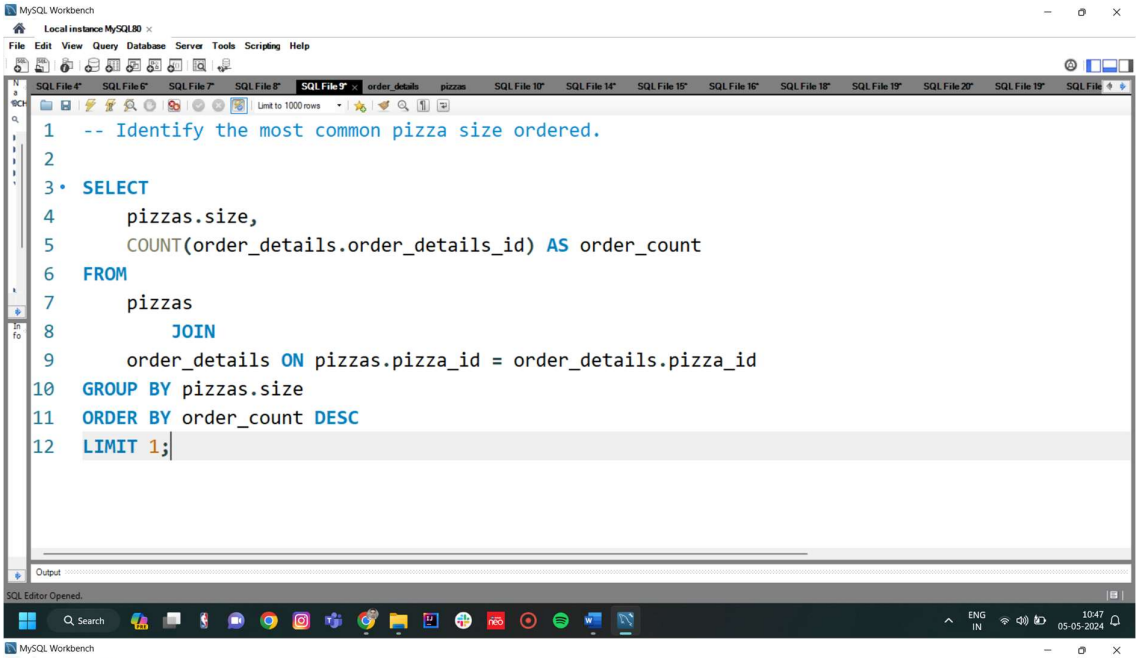

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 6* SQL File 7* SQL File 8* SQL File 9* pizzas pizza_types orders order_details pizzas SQL File 10* SQL File 14* SQL File 15* SQL File 16* SQL File 18* SQL File 19*
Limit to 1000 rows
1 -- Calculate the total revenue generated from pizza sales.
2
3 • SELECT
4 ROUND(SUM(order_details.quantity * pizzas.price),
5 2) AS total_sales
6 FROM
7 order_details
8 JOIN
9 pizzas ON pizzas.pizza_id = order_details.pizza_id
Output
SQL Editor Opened.
Search
ENG IN
10:46 05-05-2024

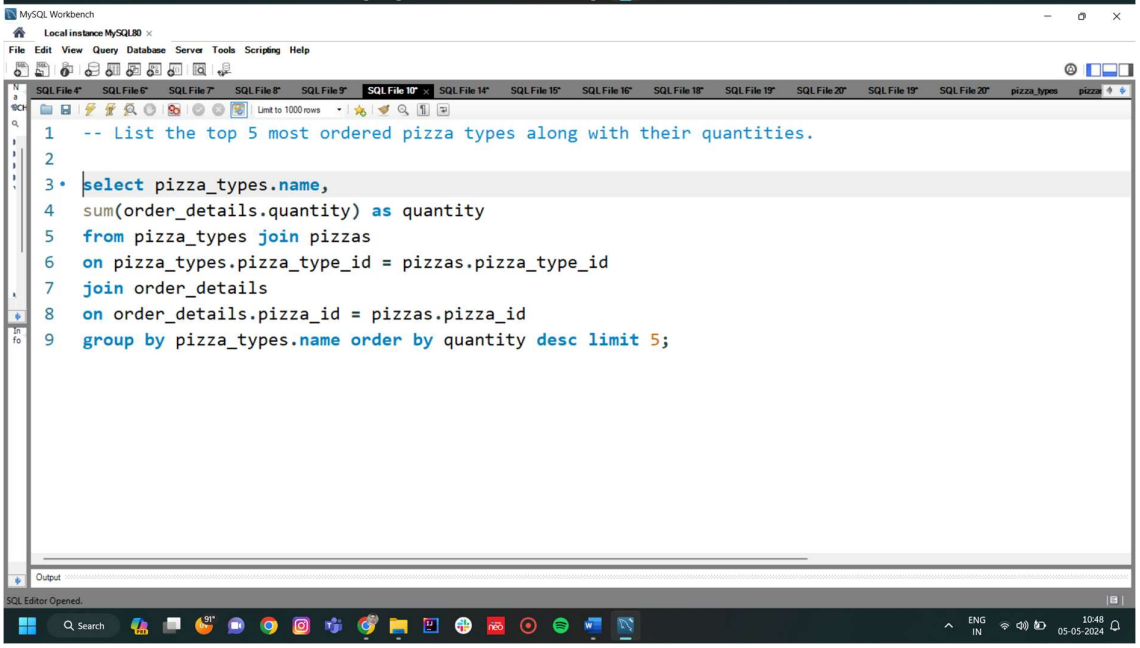

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 6* SQL File 7* SQL File 8* SQL File 9* pizzas pizza_types orders order_details pizzas SQL File 10* SQL File 14* SQL File 15* SQL File 16* SQL File 18* SQL File 19*
Limit to 1000 rows
1 -- Identify the highest-priced pizza.
2
3 SELECT
4 pizza_types.name, pizzas.price
5 FROM
6 pizza_types
7 JOIN
8 pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9 ORDER BY pizzas.price DESC
10 LIMIT 1;
Output
SQL Editor Opened.
Search
ENG IN
10:47 05-05-2024

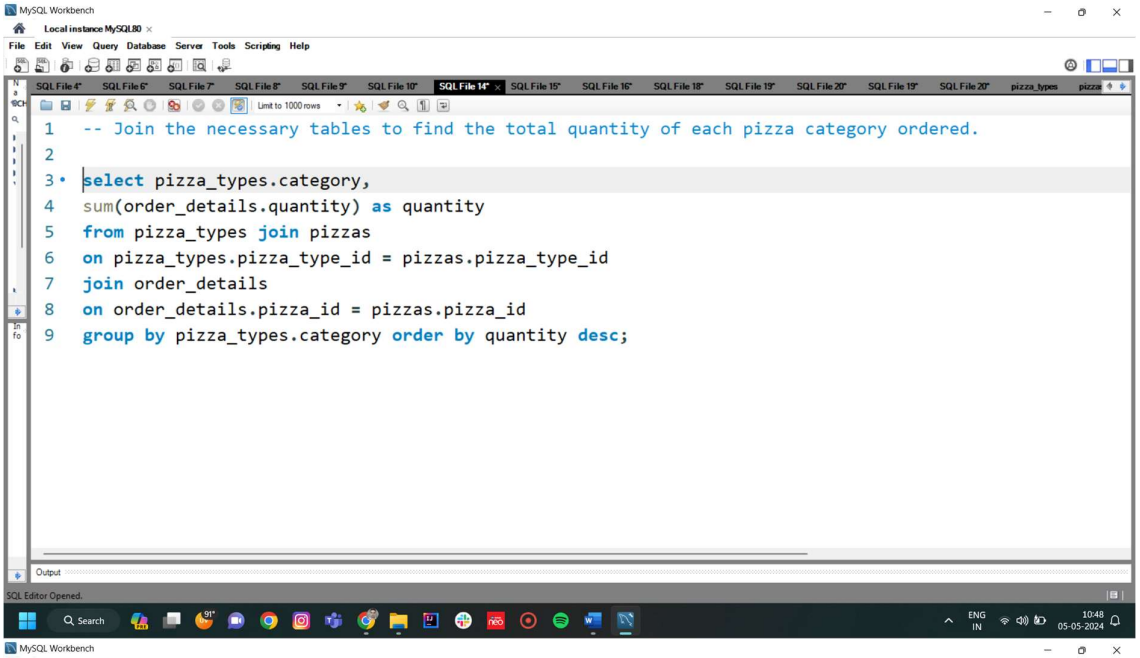

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 6* SQL File 7* SQL File 8* SQL File 9* pizzas pizza_types orders order_details pizzas SQL File 10* SQL File 14* SQL File 15* SQL File 16* SQL File 18* SQL File 19*
Limit to 1000 rows
1 -- Identify the most common pizza size ordered.
2
3 SELECT
4 pizzas.size,
5 COUNT(order_details.order_details_id) AS order_count
6 FROM
7 pizzas
8 JOIN
9 order_details ON pizzas.pizza_id = order_details.pizza_id
10 GROUP BY pizzas.size
11 ORDER BY order_count DESC
12 LIMIT 1;
Output
SQL Editor Opened.
Search
ENG IN
10:47 05-05-2024

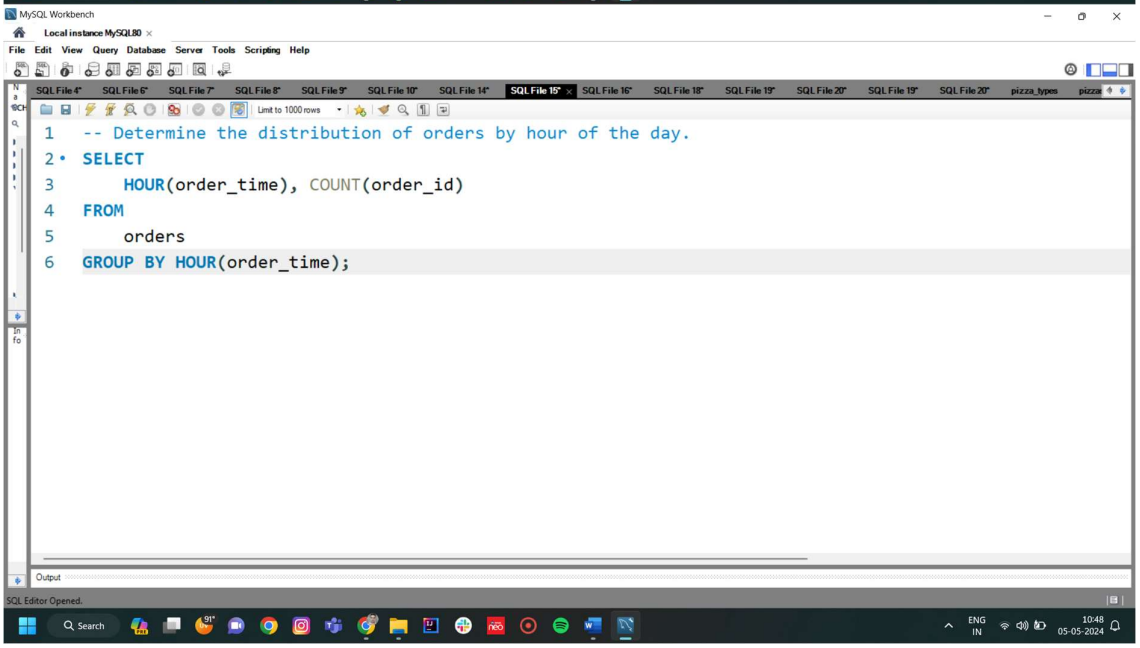

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 6* SQL File 7* SQL File 8* SQL File 9* pizza_types orders order_details pizzas SQL File 10* SQL File 14* SQL File 15* SQL File 16* SQL File 18* SQL File 19* SQL File 20*
Limit to 1000 rows
1 -- Identify the most common pizza size ordered.
2
3 • SELECT
4 pizzas.size,
5 COUNT(order_details.order_details_id) AS order_count
6 FROM
7 pizzas
8 JOIN
9 order_details ON pizzas.pizza_id = order_details.pizza_id
10 GROUP BY pizzas.size
11 ORDER BY order_count DESC
12 LIMIT 1;
Output
SQL Editor Opened.
Search
ENG IN
10:47 05-05-2024

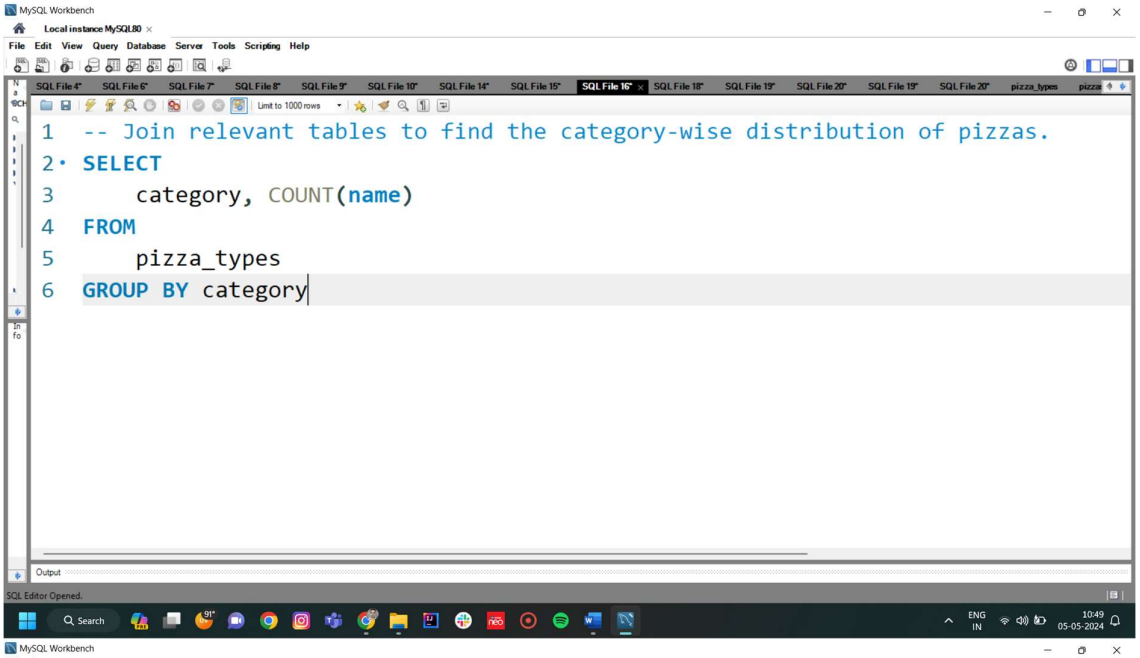

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 6* SQL File 7* SQL File 8* SQL File 9* orders order_details pizzas SQL File 10* SQL File 14* SQL File 15* SQL File 16* SQL File 18* SQL File 19* SQL File 20* SQL File 19*
Limit to 1000 rows
1 -- Identify the most common pizza size ordered.
2
3 • SELECT
4 pizzas.size,
5 COUNT(order_details.order_details_id) AS order_count
6 FROM
7 pizzas
8 JOIN
9 order_details ON pizzas.pizza_id = order_details.pizza_id
10 GROUP BY pizzas.size
11 ORDER BY order_count DESC
12 LIMIT 1;
Output
SQL Editor Opened.
Search
ENG IN
10:47 05-05-2024

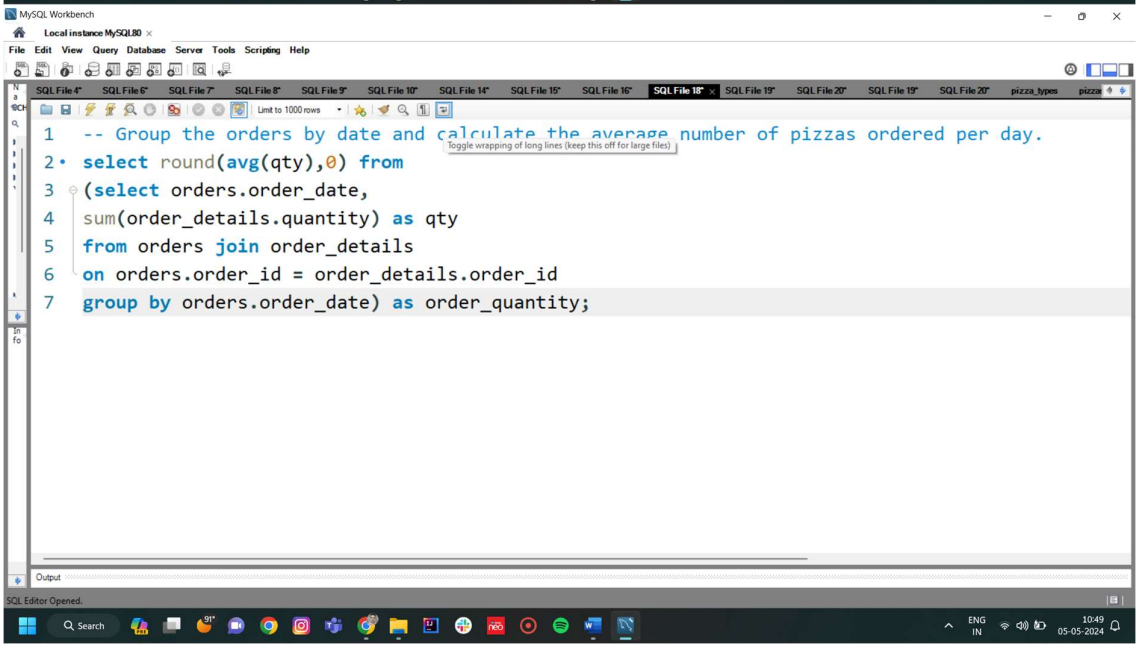

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 6* SQL File 7* SQL File 8* SQL File 9* order_details pizzas SQL File 10* SQL File 14* SQL File 15* SQL File 16* SQL File 18* SQL File 19* SQL File 20* SQL File 19*
Limit to 1000 rows
1 -- Identify the most common pizza size ordered.
2
3 • SELECT
4 pizzas.size,
5 COUNT(order_details.order_details_id) AS order_count
6 FROM
7 pizzas
8 JOIN
9 order_details ON pizzas.pizza_id = order_details.pizza_id
10 GROUP BY pizzas.size
11 ORDER BY order_count DESC
12 LIMIT 1;
Output
SQL Editor Opened.
Search
ENG IN
10:47 05-05-2024

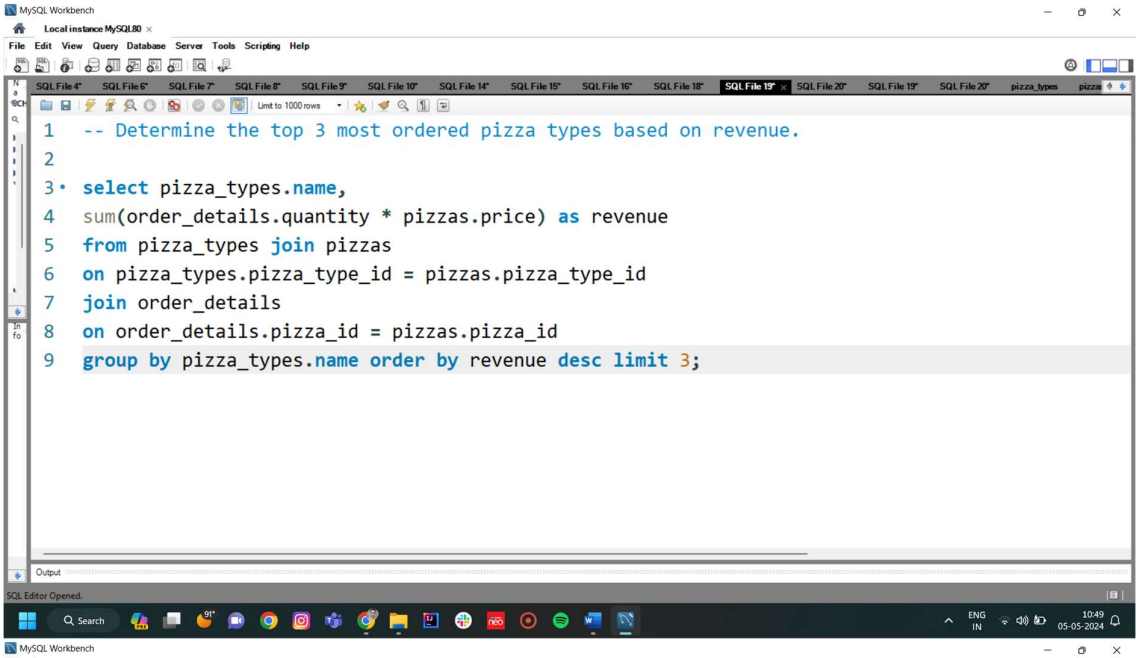

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 14* SQL File 15* SQL File 16* SQL File 18* SQL File 19* SQL File 20* SQL File 19* SQL File 20* pizza_types
Limit to 1000 rows
1 -- List the top 5 most ordered pizza types along with their quantities.
2
3 • select pizza_types.name,
4 sum(order_details.quantity) as quantity
5 from pizza_types join pizzas
6 on pizza_types.pizza_type_id = pizzas.pizza_type_id
7 join order_details
8 on order_details.pizza_id = pizzas.pizza_id
9 group by pizza_types.name order by quantity desc limit 5;
Output
SQL Editor Opened.
Search
ENG IN
10:48 05-05-2024

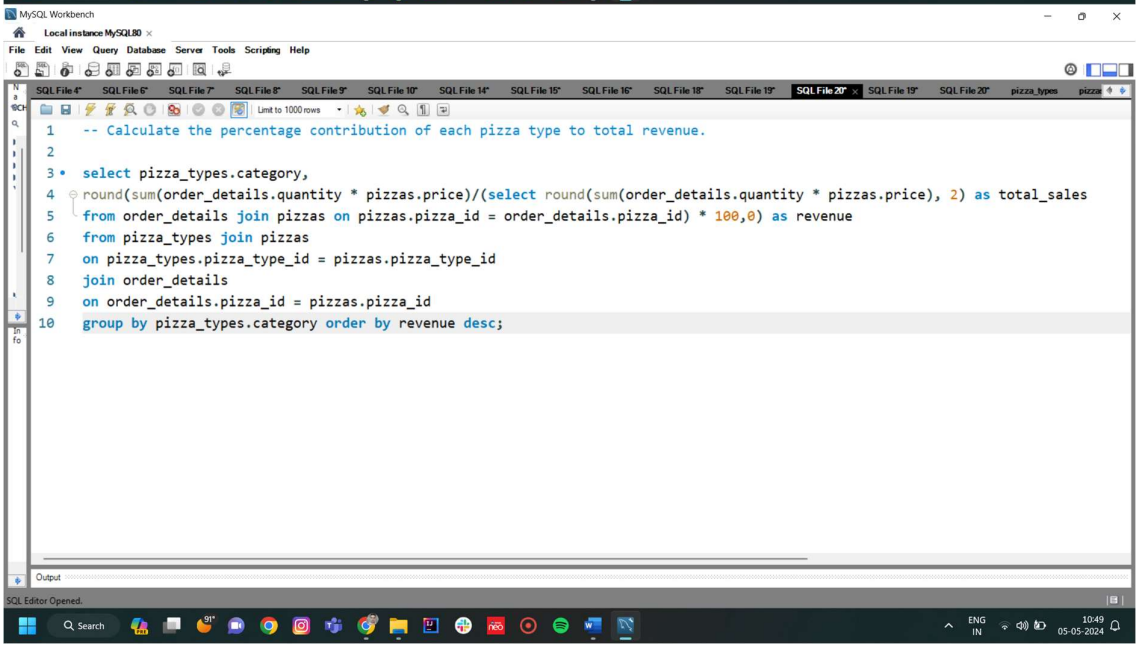

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 14*
Limit to 1000 rows
1 -- Join the necessary tables to find the total quantity of each pizza category ordered.
2
3 • select pizza_types.category,
4 sum(order_details.quantity) as quantity
5 from pizza_types join pizzas
6 on pizza_types.pizza_type_id = pizzas.pizza_type_id
7 join order_details
8 on order_details.pizza_id = pizzas.pizza_id
9 group by pizza_types.category order by quantity desc;
Output
SQL Editor Opened.
Search
ENG IN
10:48 05-05-2024

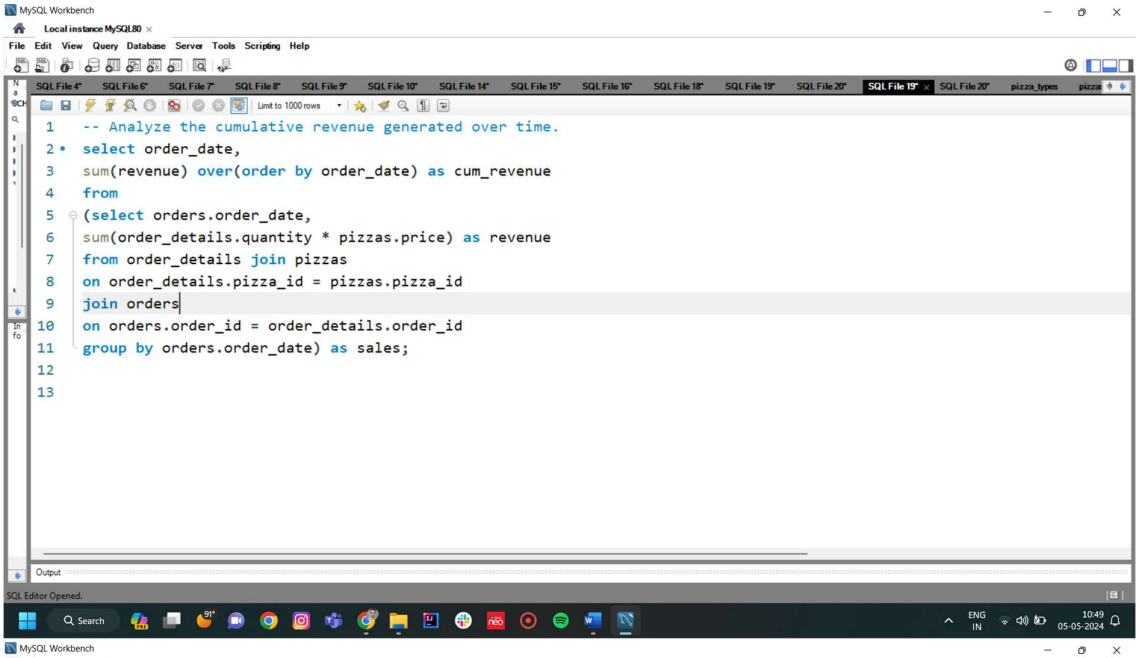

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 15*
Limit to 1000 rows
1 -- Determine the distribution of orders by hour of the day.
2 • SELECT
3 HOUR(order_time), COUNT(order_id)
4 FROM
5 orders
6 GROUP BY HOUR(order_time);
Output
SQL Editor Opened.
Search
ENG IN
10:48 05-05-2024

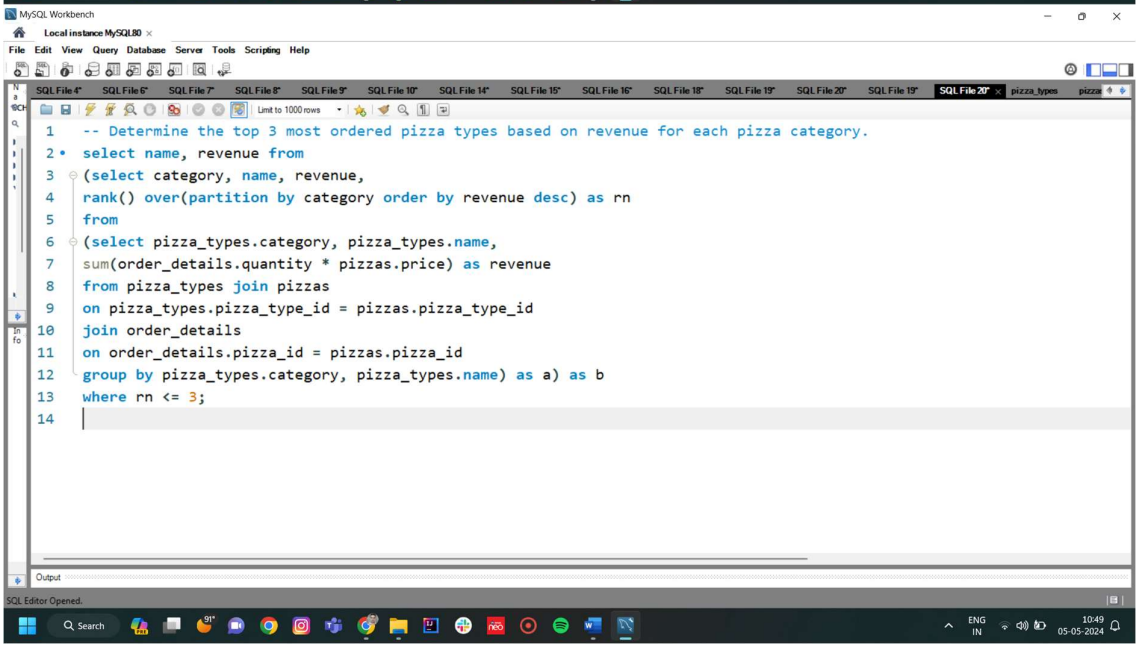

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 14* SQL File 15* SQL File 16* SQL File 18* SQL File 19* SQL File 20* SQL File 19* SQL File 20* pizza_types
Limit to 1000 rows
1 -- Join relevant tables to find the category-wise distribution of pizzas.
2 SELECT
3 category, COUNT(name)
4 FROM
5 pizza_types
6 GROUP BY category
Output
SQL Editor Opened.
Search
ENG IN
10:49 05-05-2024

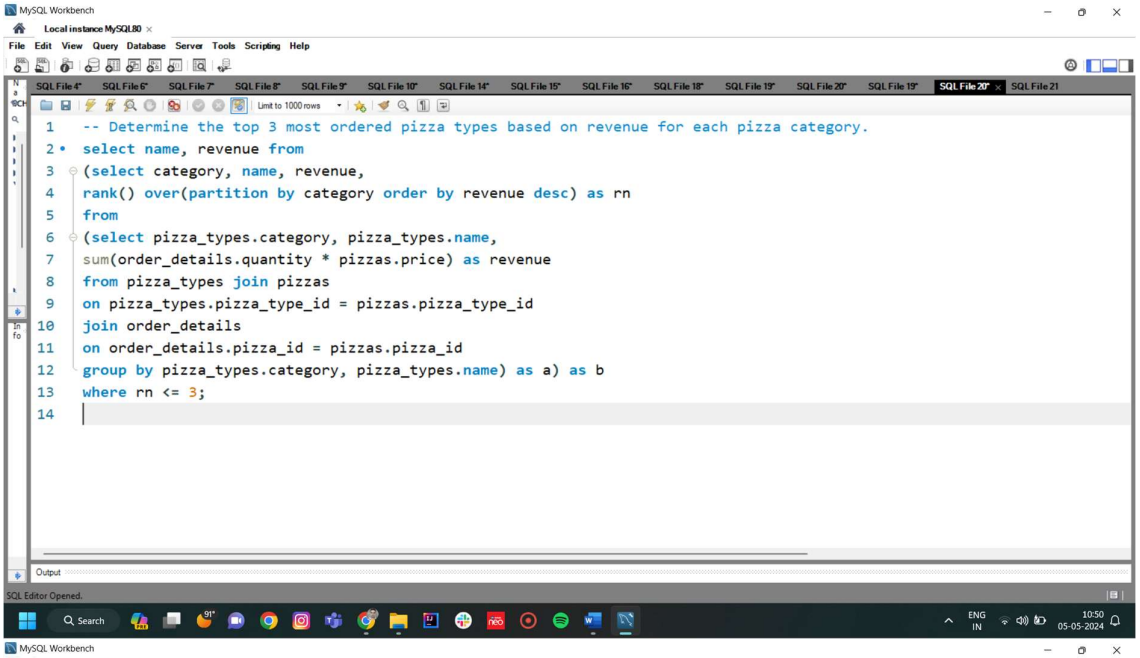

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 14* SQL File 15* SQL File 16* SQL File 18* SQL File 19* SQL File 20* SQL File 19* SQL File 20* pizza_types
Limit to 1000 rows
1 -- Group the orders by date and calculate the average number of pizzas ordered per day.
Toggle wrapping of long lines (keep this off for large files)
2 select round(avg(qty),0) from
3 (select orders.order_date,
4 sum(order_details.quantity) as qty
5 from orders join order_details
6 on orders.order_id = order_details.order_id
7 group by orders.order_date) as order_quantity;
Output
SQL Editor Opened.
Search
ENG IN
10:49 05-05-2024

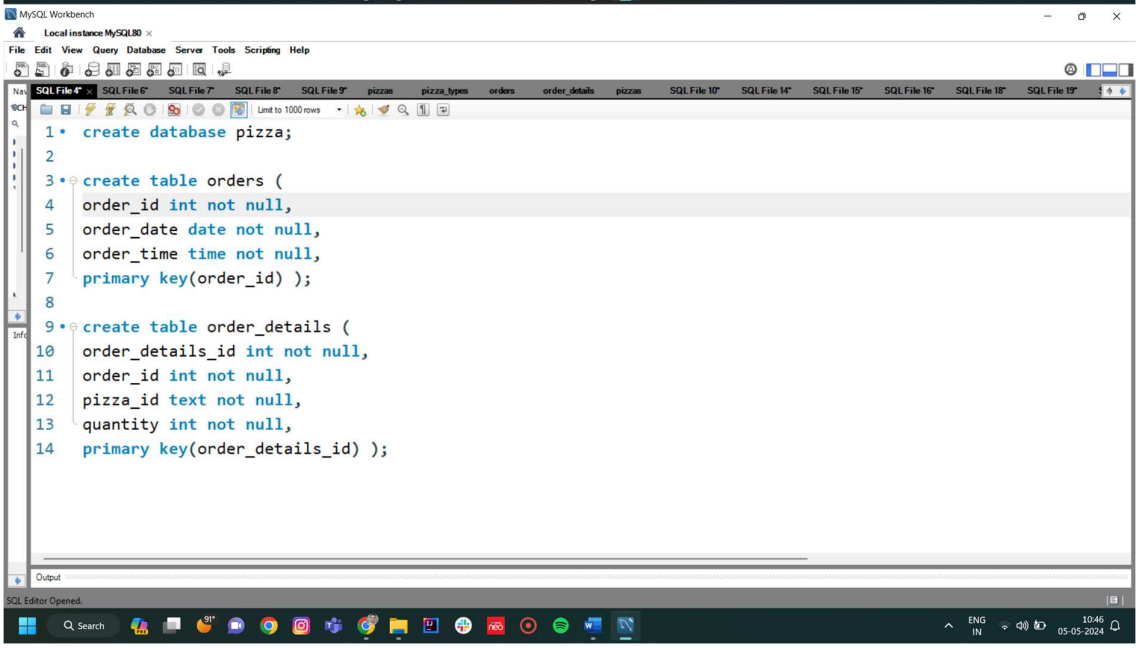

MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 14* SQL File 15* SQL File 16* SQL File 18* SQL File 19* SQL File 20* SQL File 19* SQL File 20* pizza_types
Limit to 1000 rows
1 -- Determine the top 3 most ordered pizza types based on revenue.
2
3 • select pizza_types.name,
4 sum(order_details.quantity * pizzas.price) as revenue
5 from pizza_types join pizzas
6 on pizza_types.pizza_type_id = pizzas.pizza_type_id
7 join order_details
8 on order_details.pizza_id = pizzas.pizza_id
9 group by pizza_types.name order by revenue desc limit 3;
Output
SQL Editor Opened.
Search
ENG IN
10:49 05-05-2024


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 14* SQL File 15* SQL File 16* SQL File 18* SQL File 19* SQL File 20* SQL File 19* SQL File 20* pizza_types
Limit to 1000 rows
1 -- Calculate the percentage contribution of each pizza type to total revenue.
2
3 • select pizza_types.category,
4 round(sum(order_details.quantity * pizzas.price)/(select round(sum(order_details.quantity * pizzas.price), 2) as total_sales
5 from order_details join pizzas on pizzas.pizza_id = order_details.pizza_id) * 100,0) as revenue
6 from pizza_types join pizzas
7 on pizza_types.pizza_type_id = pizzas.pizza_type_id
8 join order_details
9 on order_details.pizza_id = pizzas.pizza_id
10 group by pizza_types.category order by revenue desc;
Output
SQL Editor Opened.
Search
ENG IN
10:49 05-05-2024


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 14* SQL File 15* SQL File 16* SQL File 18* SQL File 19* SQL File 20* SQL File 19* SQL File 20* pizza_types
Limit to 1000 rows
1 -- Analyze the cumulative revenue generated over time.
2 • select order_date,
3 sum(revenue) over(order by order_date) as cum_revenue
4 from
5 (select orders.order_date,
6 sum(order_details.quantity * pizzas.price) as revenue
7 from order_details join pizzas
8 on order_details.pizza_id = pizzas.pizza_id
9 join orders
10 on orders.order_id = order_details.order_id
11 group by orders.order_date) as sales;
12
13
Output
SQL Editor Opened.
Search
ENG IN
10:49 05-05-2024


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 14* SQL File 15* SQL File 16* SQL File 18* SQL File 19* SQL File 20* SQL File 19* SQL File 20* pizza_types
Limit to 1000 rows
1 -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2 • select name, revenue from
3 (select category, name, revenue,
4 rank() over(partition by category order by revenue desc) as rn
5 from
6 (select pizza_types.category, pizza_types.name,
7 sum(order_details.quantity * pizzas.price) as revenue
8 from pizza_types join pizzas
9 on pizza_types.pizza_type_id = pizzas.pizza_type_id
10 join order_details
11 on order_details.pizza_id = pizzas.pizza_id
12 group by pizza_types.category, pizza_types.name) as a) as b
13 where rn <= 3;
14
Output
SQL Editor Opened.
Search
ENG IN
10:49 05-05-2024


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 6* SQL File 7* SQL File 8* SQL File 9* SQL File 10* SQL File 14* SQL File 15* SQL File 16* SQL File 18* SQL File 19* SQL File 20* SQL File 19* SQL File 20* SQL File 21
Limit to 1000 rows
1 -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2 • select name, revenue from
3 (select category, name, revenue,
4 rank() over(partition by category order by revenue desc) as rn
5 from
6 (select pizza_types.category, pizza_types.name,
7 sum(order_details.quantity * pizzas.price) as revenue
8 from pizza_types join pizzas
9 on pizza_types.pizza_type_id = pizzas.pizza_type_id
10 join order_details
11 on order_details.pizza_id = pizzas.pizza_id
12 group by pizza_types.category, pizza_types.name) as a) as b
13 where rn <= 3;
14
Output
SQL Editor Opened.
Search
ENG IN
10:50 05-05-2024


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
SQL File 4* SQL File 6* SQL File 7* SQL File 8* SQL File 9* pizzas pizza_types orders order_details pizzas SQL File 10* SQL File 14* SQL File 15* SQL File 16* SQL File 18* SQL File 19*
Limit to 1000 rows
1 • create database pizza;
2
3 • create table orders (
4 order_id int not null,
5 order_date date not null,
6 order_time time not null,
7 primary key(order_id) );
8
9 • create table order_details (
10 order_details_id int not null,
11 order_id int not null,
12 pizza_id text not null,
13 quantity int not null,
14 primary key(order_details_id) );
Output
SQL Editor Opened.
Search
ENG IN
10:46 05-05-2024