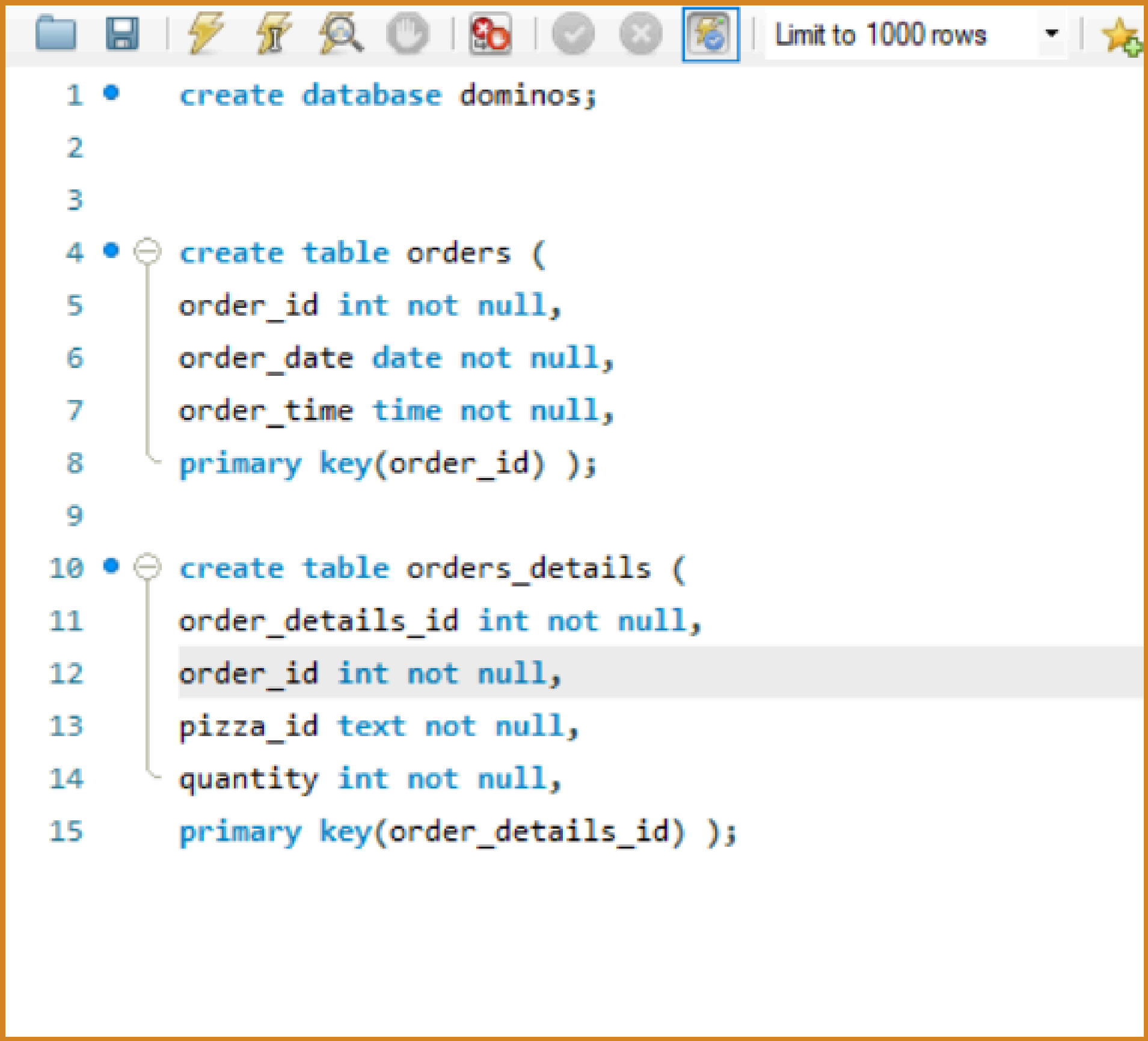
SQL PROJECT

SQL PROJECT ON PIZZA SALES

1.Retrieve the total number of orders placed.
2.Calculate the total revenue generated from pizza sales.
3.Identify the highest-priced pizza.
4.Identify the most common pizza size ordered.
5.List the top 5 most ordered pizza types along with their quantities.
6.Join the necessary tables to find the total quantity of each pizza category ordered.
7.Determine the distribution of orders by hour of the day.
8.Join relevant tables to find the category-wise distribution of pizzas.
9.Group the orders by date and calculate the average number of pizzas ordered per day.
10.Determine the top 3 most ordered pizza types based on revenue.

Pizza time

CREATING TABLES



Pizza

time

```sql
-- Join relevant tables to find the category-wise distribution of pizzas.

SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category
```

Limit to 1000 rows

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	count(name)
Chicken	6
Classic	8
Supreme	9
Veggie	9

Pizza

time

Limit to 1000 rows

```sql
-- Group the orders by date and calculate the average number of pizzas ordered pe
SELECT
    ROUND(AVG(quantity), 0)
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Grid | Filter Rows: | Export: | Wrap Cell Content:

und(avg(quantity),0)
3

Pizza

time

```sql
-- Determine the distribution of orders by hour of the day.

SELECT
    HOUR(order_time), COUNT(order_id) AS order_count
FROM
    orders
GROUP BY HOUR(order_time);
```

Save the script to a file.

Limit to 1000 rows

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

hour(order_time)	order_count
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399

Pizza

time

```sql
-- Group the orders by date and calculate the average number of pizzas ordered
SELECT
    ROUND(AVG(quantity), 0)
FROM
    (SELECT
        orders.order_date, SUM(orders_details.quantity) AS quantity
    FROM
        orders
    JOIN orders_details ON orders.order_id = orders_details.order_id
    GROUP BY orders.order_date) AS order_quantity;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

round(avg(quantity),0)
138

Pizza

time

```
1   -- Join relevant tables to find the category-wise dis
2
3   SELECT
4       category, COUNT(name)
5   FROM
6       pizza_types
7   GROUP BY category
```

Limit to 1000 rows

Result Grid | Filter Rows: | Export: | Wrap Cell Conte

	category	count(name)
▶	Chicken	6
	Classic	8

Pizza

time

```
Limit to 1000 rows
```

```
        pizza_types.category,
        SUM(orders_details.quantity) AS quantity
    FROM
        pizza_types
            JOIN
        pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
            JOIN
        orders_details ON orders_details.pizza_id = pizzas.pizza_id
    GROUP BY pizza_types.category
    ORDER BY quantity DESC;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

Pizza

time

Limit to 1000 rows

```
3       pizza_types.name, SUM(orders_details.quantity) AS quantity
4   FROM
5       pizza_types
6           JOIN
7       pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
8           JOIN
9       orders_details ON orders_details.pizza_id = pizzas.pizza_id
10  GROUP BY pizza_types.name
11  ORDER BY quantity DESC
12  LIMIT 5;
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

Pizza

time

Limit to 1000 rows

```sql
-- Identify the most common pizza size ordered.

SELECT
    pizzas.size,
    COUNT(orders_details.order_details_id) AS order_count
FROM
    pizzas
        JOIN
    orders_details ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY pizzas.size
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

size	order_count
L	18526

Pizza

time

```sql
-- Identify the highest-priced pizza.

SELECT
    pizza_types.name, pizzas.price
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	price
The Greek Pizza	35.95

Pizza

time

```sql
-- Calculate the total revenue generated from pizza sales.

SELECT
    ROUND(SUM(orders_details.quantity * pizzas.price),
            2) AS total_sales
FROM
    orders_details
        JOIN
    pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

total_sales
817860.05

Pizza

time

Limit to 1000 rows

```
1     -- Retrieve the total number of orders placed.
2
3 •   SELECT
4         COUNT(order_id) AS total_orders
5     FROM
6         orders;
7
```

esult Grid | Filter Rows: | Export: | Wrap Cell Content:

total_orders
21350