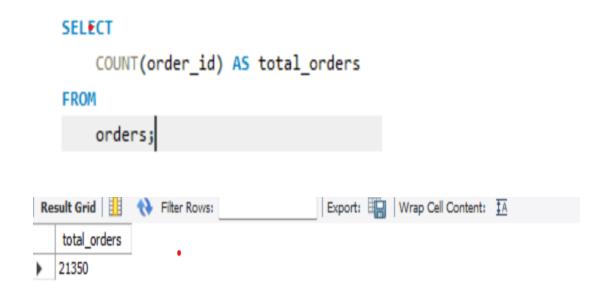
PIZZA SALES ANALYSIS

```
create database pizzahut;
```

create table orders (
 order_id int not null,
 order_date date not null,
 order_time time not null,
 primary key(order_id));

create table orders_details (
 order_details_id int not null,
 order_id int not null,
 pizza_id text not null,
 quantity int not null,
 primary key(order_details_id));

-- Retrieve the total number of order placed.



-- 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
                                     Export: Wrap Cell Content: TA
Result Grid
            Filter Rows:
   total_sales
817860.05
```

-- 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

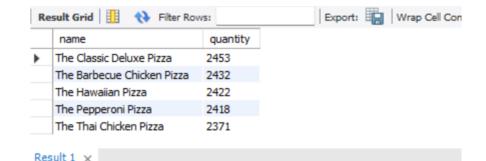
LIMIT 1;

Result Grid  Filter Rows:

Export: Wrap Cell Content: Fetch rows:
```

-- list the top 5 ordered pizza types along with their quantities.

```
SELECT
    pizza_types.name, 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.name
ORDER BY quantity DESC
LIMIT 5;
```



-- join the necessary tables to find the total quantity of each pizza category ordered.

```
SELECT
      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;
                                            Export: Wrap Cell Content: $\overline{A}$
Result Grid
               Filter Rows:
    category
             quantity
             14888
   Classic
            11987
   Supreme
   Chicken
            11050
Result 1 ×
```

-- determine the distribution of order by hour of the day.

```
SELECT

HOUR(time) AS hour, COUNT(order_id) AS order_count

FROM

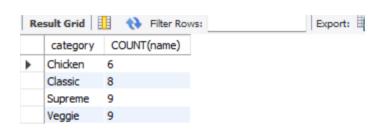
orders

GROUP BY HOUR(time);
```



-- join relevant tables to find the category wise distribution of pizzas.

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



-- group the order by date and calculate the average number of pizza per day.

```
SELECT

ROUND(AVG(quantity), 0)

FROM

(SELECT

orders.date, SUM(orders_details.quantity) AS quantity

FROM

orders

JOIN orders_details ON orders.order_id = orders_details.order_id

GROUP BY orders.date) AS order_quantity;
```



-- determine the top 3 most ordered pizza types based on revenue.

```
SELECT
    pizza types.name,
    SUM(orders_details.quantity * pizzas.price) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id
        JOIN
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
 Export: Wrap Cell Content: A Fetch rows:
                          revenue
   The Thai Chicken Pizza
                          43434.25
    The Barbecue Chicken Pizza
    The California Chicken Pizza 41409.5
```

Result 1 X

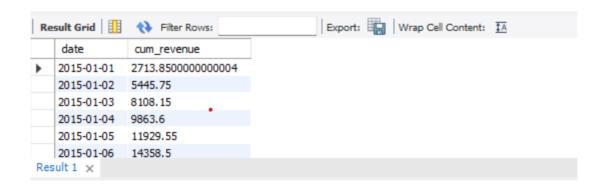
-- calculate the percentage contribution of each pizza type to total revenue.

```
SELECT
    pizza_types.category,
    ROUND(SUM(orders_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(orders_details.quantity * pizzas.price),
                                2) AS total_sales
                FROM
                    orders_details
                    pizzas pizzas ON pizzas.pizza_id = orders_details.pizza_id) * 100,
            2) AS revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue DESC;
                                           Export: Wrap Cell Content: IA
Result Grid Filter Rows:
   category revenue
  Classic
            26.91
           25.46
  Supreme
  Chicken
           23.96
           23.68
```

-- analyze the cumulative revenue generated over time.

```
select date,
sum(revenue) over(order by date) as cum_revenue
from

(select orders.date,
sum(orders_details.quantity * pizzas.price) as revenue
from orders_details join pizzas
on orders_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = orders_details.order_id
group by orders.date)as sales;
```



-- determine the top 3 most ordered pizzas types based on revenue for each pizza category.

```
select name, revenue from
```

```
(select category, name, revenue,
  rank() over(partition by category order by revenue desc)as rn
  from
(select pizza_types.category, pizza_types.name,
  sum((orders_details.quantity) * pizzas.price) as revenue
  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, pizza_types.name) as a) as b
  where rn <= 3;</pre>
```

sult Grid 🚻 💎 Filter Rows:		Export: Wrap Cell Content:
name	revenue	
The Classic Deluxe Pizza	38180.5	
The Hawaiian Pizza	32273.25	
The Pepperoni Pizza	30161.75	
The Spicy Italian Pizza	34831.25	
The Italian Supreme Pizza	33476.75	
The Sicilian Pizza	30940.5	

-- 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

ORDER BY order_count DESC;

Result Grid  Filter Rows:

size order_count

L 18526
```

15385 14137

Result 1 ×