BORCELLE

RESTAURANT

Specialist in Italian Food

BORCELLE
RESTAURANT

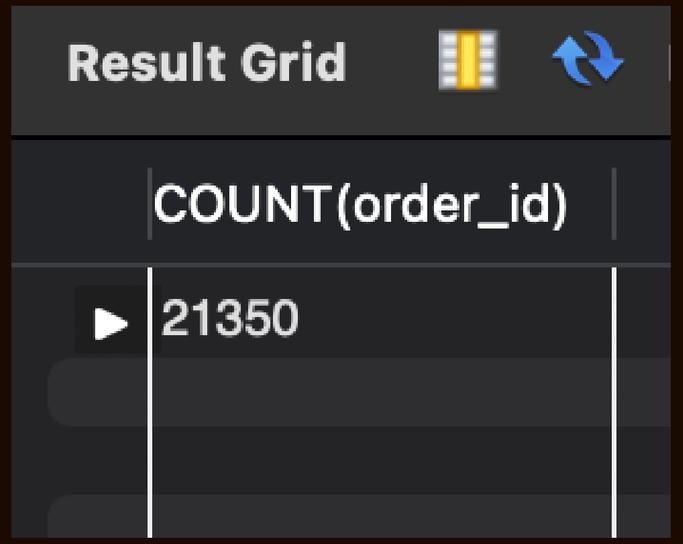
ABOUT

By harnessing the power of SQL queries, this project aims to uncover valuable insights into sales performance, customer trends, and operational efficiencies crucial for optimizing business strategies.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACE

BORCELLE
RESTAURANT

```
SELECT
    COUNT(order_id)
FROM
    orders;
```



CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

BORCELLE
RESTAURANT

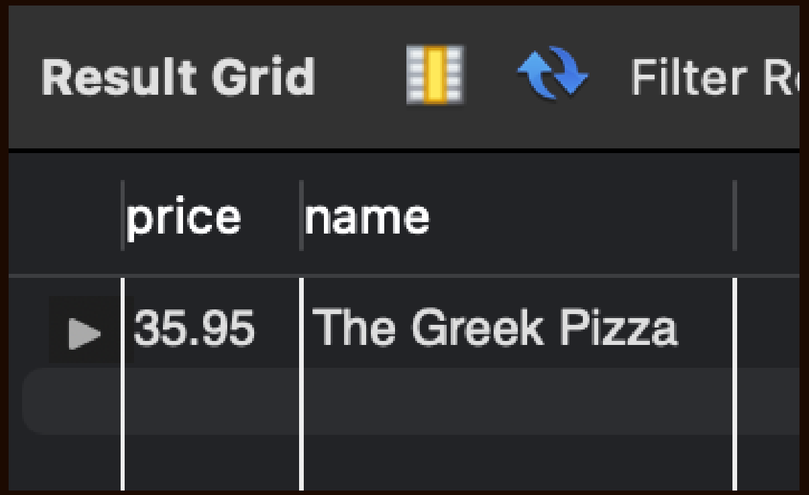
```
SELECT
    SUM(pizzas.price * order_details.quantity)
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id =
order_details.pizza_id
```

SUM(pizzas.price*order_details.quar
817860.049999993

IDENTIFY THE HIGHEST-PRICED PIZZA.

BORCELLE
RESTAURANT

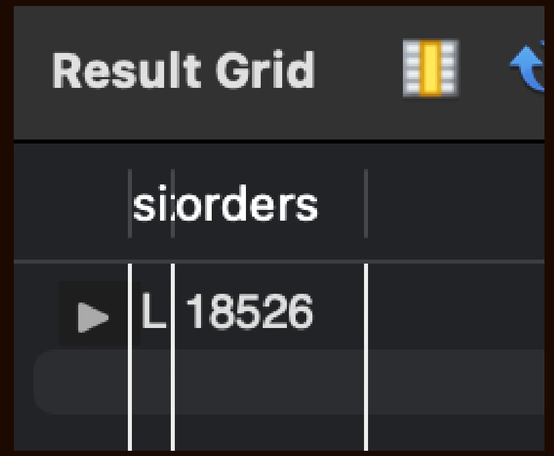
```
SELECT
    pizzas.price, pizza_types.name
FROM
    pizzas
        JOIN
    pizza_types ON pizzas.pizza_type_id =
pizza_types.pizza_type_id
ORDER BY pizzas.price DESC
LIMIT 1
```



IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

BORCELLE
RESTAURANT

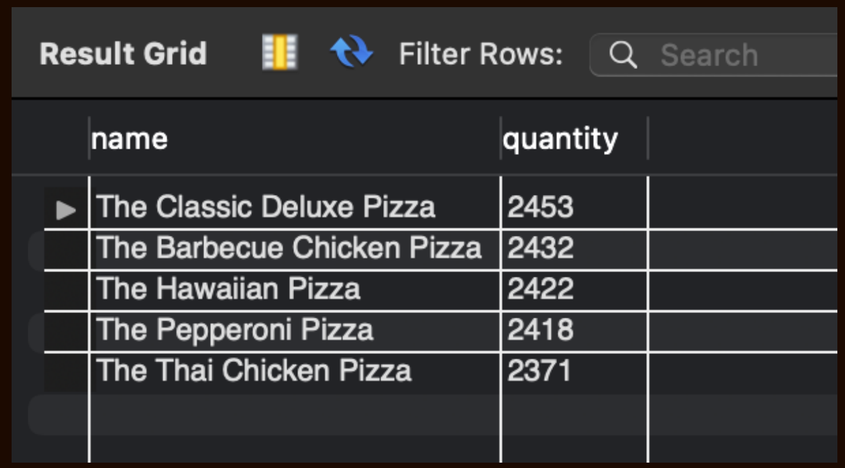
```
SELECT
    pizzas.size, COUNT(order_details.quantity) AS orders
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY orders DESC
LIMIT 1
```



LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

BORCELLE
RESTAURANT

```
SELECT
    pizza_types.name, SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC
LIMIT 5
```



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

JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

BORCELLE
RESTAURANT

```
SELECT
    pizza_types.category,
    SUM(order_details.quantity) AS quantity
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
```

Result Grid Filter Rows: Search

category	quantity
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

BORCELLE
RESTAURANT

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

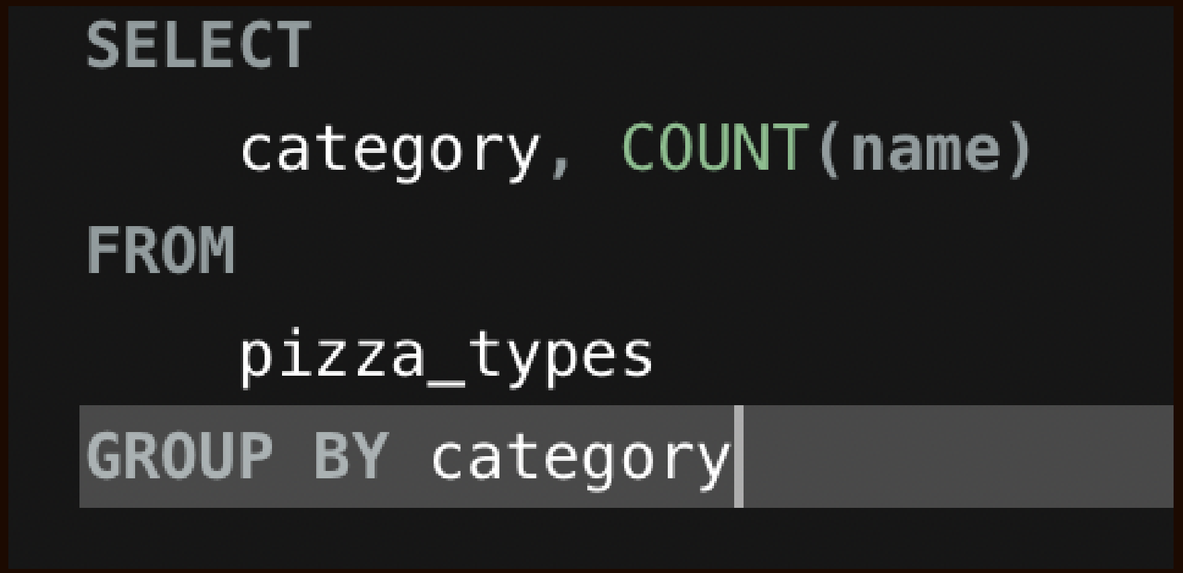
Result Grid | Filter Rows: Search | Exp

HOUR(order_time)	COUNT(order_id)
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920
17	2336
18	2399
19	2009
20	1642
21	1198
22	663
23	28
10	8
9	1

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

BORCELLE
RESTAURANT

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



GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

BORCELLE
RESTAURANT

```
SELECT
    orders.order_date, AVG(order_details.quantity)
FROM
    orders
        JOIN
    order_details ON orders.order_id =
order_details.order_id
GROUP BY orders.order_date
```

Result Grid | Filter Rows: Search | Expor

order_date	AVG(order_details.quanti...
2015-01-01	1.0062
2015-01-02	1.0313
2015-01-03	1.0260
2015-01-04	1.0000
2015-01-05	1.0331
2015-01-06	1.0208
2015-01-07	1.0376
2015-01-08	1.0117
2015-01-09	1.0325
2015-01-10	1.0069
2015-01-11	1.0175
2015-01-12	1.0085
2015-01-13	1.0256
2015-01-14	1.0417
2015-01-15	1.0000
2015-01-16	1.0194
2015-01-17	1.0246
2015-01-18	1.0252
2015-01-19	1.0216
2015-01-20	1.0288
2015-01-21	1.0157
2015-01-22	1.0194
2015-01-23	1.0201
2015-01-24	1.0000

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

BORCELLE
RESTAURANT

```
SELECT
    pizza_types.name,
    ROUND(SUM(order_details.quantity * pizzas.price)) AS
revenue
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id =
pizzas.pizza_type_id
        JOIN
    order_details ON order_details.pizza_id =
pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3
```

Result Grid Filter Rows:

name	revenue
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

BORCELLE
RESTAURANT

```
SELECT
    pizza_types.category,
    ROUND(SUM(pizzas.price * order_details.quantity) / (SELECT
                ROUND(SUM(pizzas.price * order_details.quantity),
                            2)
            FROM
                order_details
                    JOIN
                pizzas ON order_details.pizza_id = pizzas.pizza_id) * 100,
            0)
FROM
    pizza_types
        JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizza_types.category
```

Result Grid | Filter Rows: Search | Expor

order_date	AVG(order_details.quanti...
2015-01-01	1.0062
2015-01-02	1.0313
2015-01-03	1.0260
2015-01-04	1.0000
2015-01-05	1.0331
2015-01-06	1.0208
2015-01-07	1.0376
2015-01-08	1.0117
2015-01-09	1.0325
2015-01-10	1.0069
2015-01-11	1.0175
2015-01-12	1.0085
2015-01-13	1.0256
2015-01-14	1.0417
2015-01-15	1.0000
2015-01-16	1.0194
2015-01-17	1.0246
2015-01-18	1.0252
2015-01-19	1.0216
2015-01-20	1.0288
2015-01-21	1.0157
2015-01-22	1.0194
2015-01-23	1.0201
2015-01-24	1.0000

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

BORCELLE
RESTAURANT

```
SELECT
    orders.order_date,
    SUM(order_details.quantity * pizzas.price) AS revenue
FROM
    orders
        JOIN
    order_details ON orders.order_id = order_details.order_id
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
GROUP BY orders.order_date
```

Result Grid | Filter Rows: Search | Expor

order_date	revenue
2015-01-01	2713.8500000000004
2015-01-02	2731.8999999999996
2015-01-03	2662.3999999999996
2015-01-04	1755.4500000000003
2015-01-05	2065.95
2015-01-06	2428.95
2015-01-07	2202.2000000000003
2015-01-08	2838.3499999999995
2015-01-09	2127.3500000000004
2015-01-10	2463.95
2015-01-11	1872.3000000000002
2015-01-12	1919.0500000000002
2015-01-13	2049.6000000000004
2015-01-14	2527.3999999999996