PIZZA SALES ANALYSIS

Analyzing Sales Data with SQL

Specialist in The Classic Deluxe Pizza

EXPLORING PIZZA SALES STRATEGIES THROUGH SQL

In the data-driven world of business, understanding sales trends is key to success. This analysis harnesses the power of SQL to uncover hidden patterns and insights in pizza sales data. By leveraging SQL's efficiency in managing large datasets, we extract actionable intelligence to inform business decisions and drive growth.

WHY CHOOSE THIS

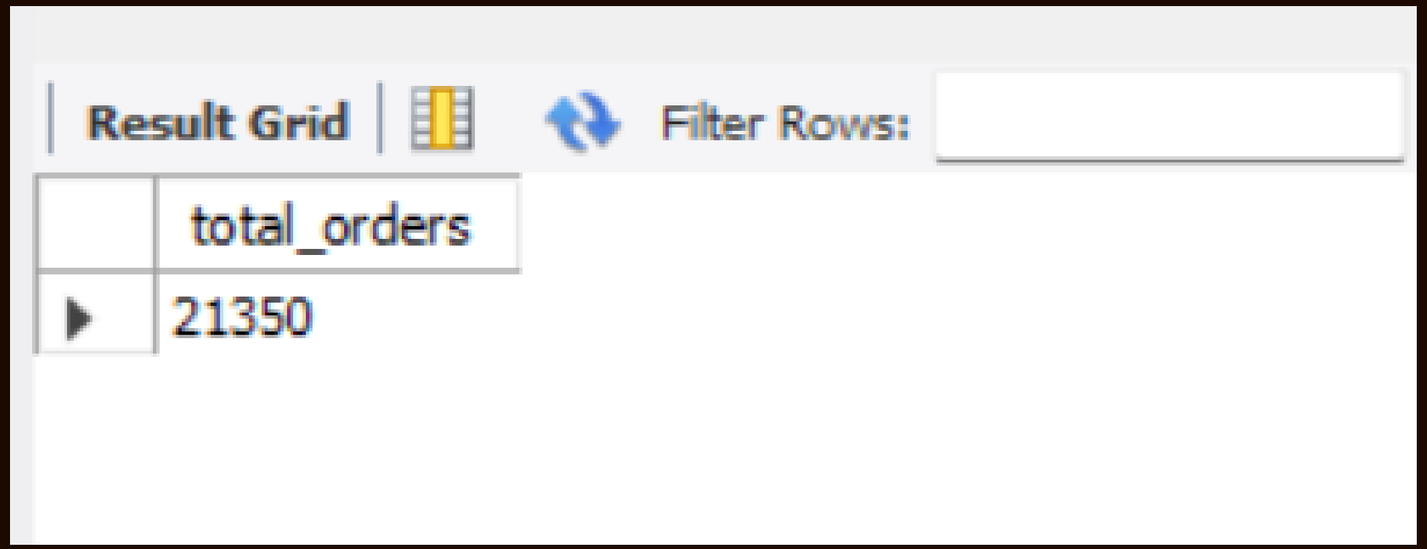
I am undertaking an initiative to leverage SQL for data analysis concerning pizza sales, aiming to impact strategic business decisions. Through analyzing sales trends, customer preferences, and regional performance, the goal is to optimize inventory management, refine marketing strategies, and enhance operational efficiency. This analysis will provide valuable insights to drive profitability, identify growth opportunities, and ensure an enhanced customer experience. Essentially, this project centers on the utilization of data-driven insights to maintain competitiveness in the dynamic pizza industry.

QUERY 1: Retrieve the total number of orders placed.

```
select
count(order_id) as total_orders
from orders;
```

QUERY

RESULT



QUERY 2: Calculate the total revenue generated from pizza sales.

```
select
round(sum(order_details.quantity * pizzas.price),2) as sales_revenue
from order_details
join pizzas
on pizzas.pizza_id = order_details.pizza_id;
```

QUERY

RESULT

Result Grid | Filter Rows:

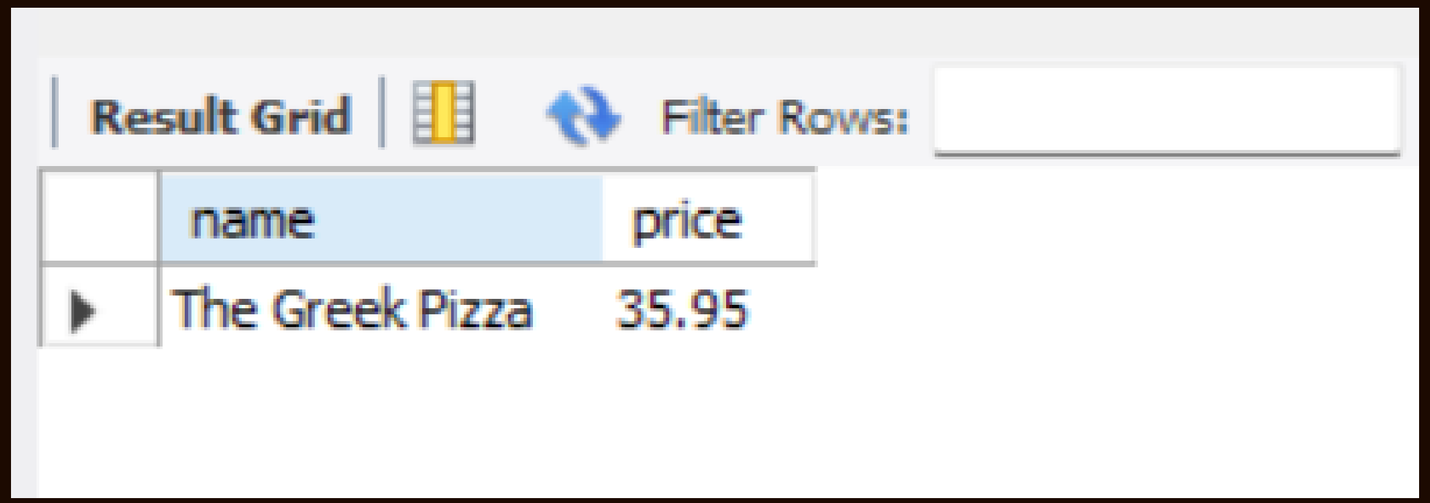
	sales_revenue
▶	817860.05

QUERY 3: 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;
```

QUERY

RESULT

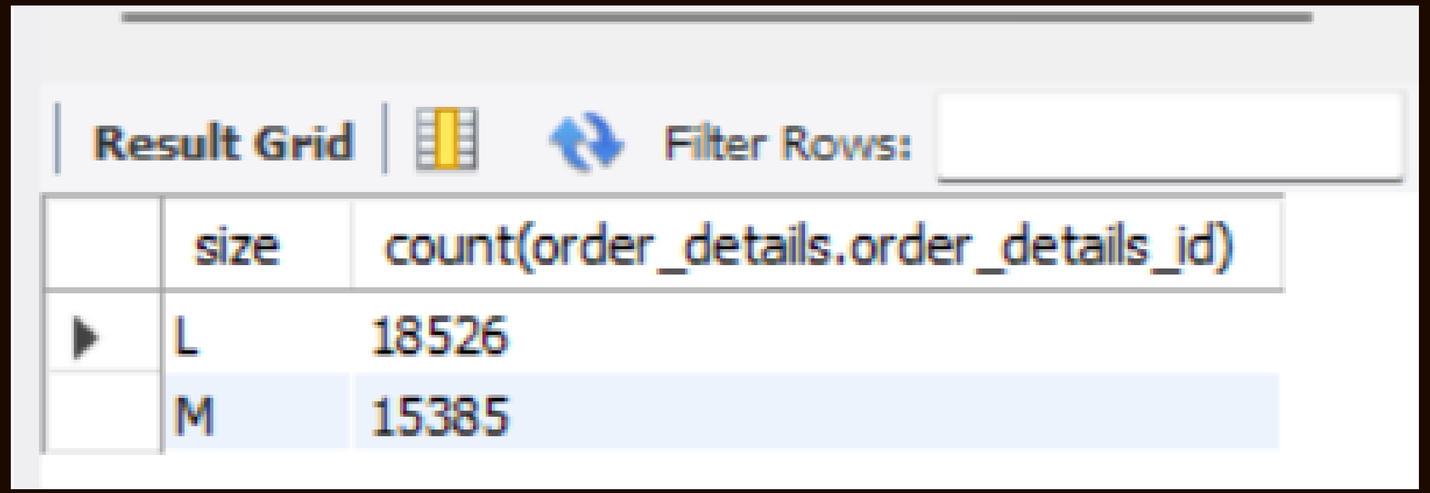


QUERY 4: Identify the most common pizza size ordered.

```
select pizzas.size,
count(order_details.order_details_id)
from pizzas
join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size
order by count(order_details.order_details_id) desc limit 2;
```

QUERY

RESULT



size	count(order_details.order_details_id)
L	18526
M	15385

QUERY 5: Join the necessary tables to find the total quantity of each pizza category ordered.

QUERY

```
select pizza_types.category,
sum(order_details.quantity)
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.category
order by sum(order_details.quantity) desc;
```

RESULT

Result Grid | Filter Rows:

category	sum(order_details.quantity)
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

QUERY 6: Determine the distribution of orders by hour of the day.

```sql
select
hour(order_time),
count(order_id)
from orders
group by hour(order_time)
order by count(order_id);
```

QUERY

RESULT

Result Grid | Filter Rows:

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

QUERY 7: Join relevant tables to find the category-wise distribution of pizzas.

```sql
select distinct category,
count(name)
from pizza_types
group by category
order by count(name) desc;
```

QUERY

RESULT

Result Grid | Filter Rows:

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

QUERY 8: Group the orders by date and calculate the average number of pizzas ordered per day.

```
select round(avg(order_quantity),0)
from (
select sum(order_details.quantity) as order_quantity,
orders.order_date
from order_details
join orders
on order_details.order_id = orders.order_id
group by orders.order_date) as avg_orders;
```

QUERY

RESULT

Result Grid | Filter Rows:

round(avg(order_quantity),0)
138

QUERY 9: Determine the top 3 most ordered pizza types based on revenue.

```sql
select
sum(order_details.quantity * pizzas.price) as sales_revenue,
pizza_types.name
from pizzas
join pizza_types
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 sales_revenue desc
limit 3;
```

QUERY

RESULT

Result Grid | Filter Rows:

sales_revenue	name
43434.25	The Thai Chicken Pizza
42768	The Barbecue Chicken Pizza
41409.5	The California Chicken Pizza

QUERY 10: Calculate the percentage contribution of each pizza type to total revenue.

```sql
select
pizza_types.category,
round((sum(order_details.quantity * pizzas.price) /
(select
sum(order_details.quantity * pizzas.price) as sales_revenue
from order_details
join pizzas
on pizzas.pizza_id = order_details.pizza_id))*100, 2) as sales_revenue
from pizzas
join pizza_types
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
order by sales_revenue desc;
```

QUERY

RESULT

Result Grid | Filter Rows:

category	sales_revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

QUERY 11: Determine the top 3 most ordered pizza types based on revenue.

```sql
select
sum(order_details.quantity * pizzas.price) as sales_revenue,
pizza_types.name
from pizzas
join pizza_types
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 sales_revenue desc
limit 3;
```

QUERY

RESULT

Result Grid | Filter Rows:

sales_revenue	name
43434.25	The Thai Chicken Pizza
42768	The Barbecue Chicken Pizza
41409.5	The California Chicken Pizza

QUERY 12: Calculate the percentage contribution of each pizza type to total revenue.

QUERY

```
select
pizza_types.category,
round((sum(order_details.quantity * pizzas.price) /
(select
sum(order_details.quantity * pizzas.price) as sales_revenue
from order_details
join pizzas
on pizzas.pizza_id = order_details.pizza_id))*100, 2) as sales_revenue
from pizzas
join pizza_types
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
order by sales_revenue desc;
```

RESULT

Result Grid | Filter Rows:

category	sales_revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68

QUERY 13: Analyze the cumulative revenue generated over time.

```sql
select
order_date,
round(sum(sum_revenue)
over(order by order_date),2) as cumulative_revenue
from
(select orders.order_date,
sum(order_details.quantity * pizzas.price) as sum_revenue
from pizzas
join order_details
on pizzas.pizza_id = order_details.pizza_id
join orders
on order_details.order_id = orders.order_id
group by orders.order_date) as sales_revenue;
```

QUERY

Result Grid | Filter Rows:

order_date	cumulative_revenue
2015-03-14	166867.85
2015-03-15	168936.45
2015-03-16	171231.5
2015-03-17	174196.8
2015-03-18	176272.2
2015-03-19	178660.8
2015-03-20	181122.05
2015-03-21	183389.45
2015-03-22	184648.7
2015-03-23	186881.25
2015-03-24	189043.55
2015-03-25	190971.3
2015-03-26	193186.8
2015-03-27	195931.6
2015-03-28	198183.7
2015-03-29	200337.95
2015-03-30	202593.4
2015-03-31	205350
2015-04-01	207526.85
2015-04-02	210074
2015-04-03	212612.2
2015-04-04	215379.75
2015-04-05	217289.6
2015-04-06	219911.95
2015-04-07	222146.2
2015-04-08	224440.15
2015-04-09	226487.45
2015-04-10	228912.4
2015-04-11	231456.15
2015-04-12	233450.45
2015-04-13	235946.65

Result Grid | Filter Rows:

order_date	cumulative_revenue
2015-04-14	238452.35
2015-04-15	241031.2
2015-04-16	243049.15
2015-04-17	245724.8
2015-04-18	248011
2015-04-19	249538.95
2015-04-20	251998.4
2015-04-21	254211.55
2015-04-22	256405
2015-04-23	258831.15
2015-04-24	261810.35
2015-04-25	263899.55
2015-04-26	265666.95
2015-04-27	267847.75
2015-04-28	269590.55
2015-04-29	271419.3
2015-04-30	274086.8
2015-05-01	276658.75
2015-05-02	279058.95
2015-05-03	280891.2
2015-05-04	283180.1
2015-05-05	284893.7
2015-05-06	287203.5
2015-05-07	289432.35
2015-05-08	292484.65
2015-05-09	294853.05
2015-05-10	297141.4
2015-05-11	299529.45
2015-05-12	301829.15
2015-05-13	304090.95
2015-05-14	306785.45

RESULT

QUERY 13: Analyze the cumulative revenue generated over time.

Result Grid | Filter Rows:

order_date	cumulative_revenue
2015-05-15	310171.6
2015-05-16	312452.7
2015-05-17	314281.1
2015-05-18	316490.75
2015-05-19	318477.75
2015-05-20	320850.75
2015-05-21	322913.3
2015-05-22	325548.4
2015-05-23	327992.55
2015-05-24	330189.5
2015-05-25	332293.9
2015-05-26	334170.35
2015-05-27	336267.35
2015-05-28	338283.75
2015-05-29	341284.95
2015-05-30	343771.9
2015-05-31	345489.55
2015-06-01	348557.3
2015-06-02	351007.25
2015-06-03	352914.3
2015-06-04	355197.9
2015-06-05	357898.05
2015-06-06	360179
2015-06-07	362139.75
2015-06-08	364404.7
2015-06-09	366847.25
2015-06-10	368866.65
2015-06-11	371517.15
2015-06-12	373655.75
2015-06-13	376164.65
2015-06-14	378023.65

Result Grid | Filter Rows:

order_date	cumulative_revenue
2015-06-14	378023.65
2015-06-15	380619.25
2015-06-16	382517.55
2015-06-17	384654.65
2015-06-18	386639.15
2015-06-19	389432.6
2015-06-20	391493.2
2015-06-21	393418.4
2015-06-22	395737.7
2015-06-23	397780.45
2015-06-24	400107.95
2015-06-25	402507.1
2015-06-26	405252.6
2015-06-27	408065.5
2015-06-28	409635.2
2015-06-29	411508.8
2015-06-30	413719.75
2015-07-01	415951.25
2015-07-02	418246.05
2015-07-03	421689.05
2015-07-04	425553.25
2015-07-05	427144.7
2015-07-06	429261.6
2015-07-07	431636
2015-07-08	434032.05
2015-07-09	436329.8
2015-07-10	438762.2
2015-07-11	440847.75
2015-07-12	443033.4
2015-07-13	445092.9
2015-07-14	447049.4

Result Grid | Filter Rows:

order_date	cumulative_revenue
2015-07-15	449551.2
2015-07-16	452015.1
2015-07-17	455146.75
2015-07-18	457268.95
2015-07-19	459291.65
2015-07-20	461792.65
2015-07-21	463823.5
2015-07-22	466115.6
2015-07-23	468330.1
2015-07-24	471534.5
2015-07-25	473771.75
2015-07-26	475643.3
2015-07-27	477815.3
2015-07-28	479909.85
2015-07-29	481833.1
2015-07-30	484182.25
2015-07-31	486277.65
2015-08-01	488718.2
2015-08-02	490628.35
2015-08-03	492610.6
2015-08-04	494700.75
2015-08-05	496795.6
2015-08-06	498894.85
2015-08-07	501521.25
2015-08-08	504237.65
2015-08-09	506240.3
2015-08-10	508379.75
2015-08-11	510669.75
2015-08-12	513035.5
2015-08-13	515109.65
2015-08-14	518126.25

Result Grid | Filter Rows:

order_date	cumulative_revenue
2015-08-15	520378.6
2015-08-16	522517.9
2015-08-17	525143.9
2015-08-18	527245.1
2015-08-19	529578.05
2015-08-20	531485.75
2015-08-21	534087.15
2015-08-22	536493.15
2015-08-23	538194.75
2015-08-24	539891.8
2015-08-25	541850.7
2015-08-26	544180.1
2015-08-27	546297.75
2015-08-28	548944.95
2015-08-29	550979.95
2015-08-30	552474.55
2015-08-31	554555.9
2015-09-01	556908.75
2015-09-02	558774.3
2015-09-03	561026.9
2015-09-04	563987.85
2015-09-05	566525.65
2015-09-06	568017.3
2015-09-07	570300.65
2015-09-08	572550.15
2015-09-09	575130.25
2015-09-10	577546.1
2015-09-11	580308
2015-09-12	582896.15
2015-09-13	584734.3
2015-09-14	586899.55

Result Grid | Filter Rows:

order_date	cumulative_revenue
2015-09-15	589449.75
2015-09-16	591635
2015-09-17	593877.05
2015-09-18	596598.6
2015-09-19	598885.15
2015-09-20	600714.15
2015-09-21	602845.6
2015-09-22	605016
2015-09-23	607179
2015-09-26	609425.85
2015-09-27	611740.55
2015-09-28	613775.85
2015-09-29	616537.9
2015-09-30	618735.95
2015-10-01	621938.1
2015-10-02	624012.95
2015-10-03	626413.9
2015-10-04	628556.1
2015-10-06	630772.05
2015-10-07	632864.4
2015-10-08	634840.25
2015-10-09	637352.85
2015-10-10	639663.05
2015-10-11	641579.3
2015-10-13	643905.25
2015-10-14	646051.6
2015-10-15	650371.8
2015-10-16	652926.9
2015-10-17	655266.7
2015-10-18	657062
2015-10-20	659499.15

RESULT

QUERY 13: Analyze the cumulative revenue generated over time.

Result Grid | Filter Rows:

order_date	cumulative_revenue
2015-10-21	661959.65
2015-10-22	664360.55
2015-10-23	666971.2
2015-10-24	669650.7
2015-10-25	671487.75
2015-10-27	673476.4
2015-10-28	675112.35
2015-10-29	677282.1
2015-10-30	680018.7
2015-10-31	682763.55
2015-11-01	684750.2
2015-11-02	687049.3
2015-11-03	688877.85
2015-11-04	690843.95
2015-11-05	693024.3
2015-11-06	696181.8
2015-11-07	698762.75
2015-11-08	700872.9
2015-11-09	703356.65
2015-11-10	705399.25
2015-11-11	707345.45
2015-11-12	709910.25
2015-11-13	712174.8
2015-11-14	714499.55
2015-11-15	716321.2
2015-11-16	718577.85
2015-11-17	720534.4
2015-11-18	722646.1
2015-11-19	725341
2015-11-20	727729.1
2015-11-21	729813.05

Result Grid | Filter Rows:

order_date	cumulative_revenue
2015-11-22	731181.75
2015-11-23	733646.9
2015-11-24	735876.95
2015-11-25	738240.2
2015-11-26	742646.15
2015-11-27	747068.6
2015-11-28	749036.65
2015-11-29	750935.65
2015-11-30	753158.9
2015-12-01	755235.6
2015-12-02	757449.7
2015-12-03	759692.9
2015-12-04	762571.25
2015-12-05	765199.2
2015-12-06	767549.45
2015-12-07	769964.25
2015-12-08	771820.5
2015-12-09	774392.05
2015-12-10	776377.65
2015-12-11	779011.65
2015-12-12	780971.8
2015-12-13	783216.95
2015-12-14	785389.55
2015-12-15	787777
2015-12-16	790011.8
2015-12-17	791892.55
2015-12-18	794778.85
2015-12-19	797083.05
2015-12-20	799187.95
2015-12-21	801288.65
2015-12-22	803171.6

Result Grid | Filter Rows:

order_date	cumulative_revenue
2015-12-23	805415.9
2015-12-24	807553.75
2015-12-26	809196.8
2015-12-27	810615.8
2015-12-28	812253
2015-12-29	813606.25
2015-12-30	814944.05
2015-12-31	817860.05

RESULT

QUERY 14: Determine the top 3 most ordered pizza types based on revenue for each pizza category.

```sql
select
category, name, total_revenue
from (select category, name, total_revenue,
rank() over(partition by category
order by total_revenue desc) as revenue
from(select
pizza_types.name, pizza_types.category,
sum(order_details.quantity * pizzas.price) as total_revenue
from pizzas
join pizza_types
on pizzas.pizza_type_id = pizza_types.pizza_type_id
join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizza_types.category,
pizza_types.name)
as categorywise_revenue) as pizzabased_revenue
where revenue <= 3;
```

QUERY

Result Grid | Filter Rows: | Export:

category	name	total_revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75
Supreme	The Spicy Italian Pizza	34831.25
Supreme	The Italian Supreme Pizza	33476.75
Supreme	The Sicilian Pizza	30940.5
Veggie	The Four Cheese Pizza	32265.70000000065
Veggie	The Mexicana Pizza	26780.75
Veggie	The Five Cheese Pizza	26066.5

RESULT

THANK YOU