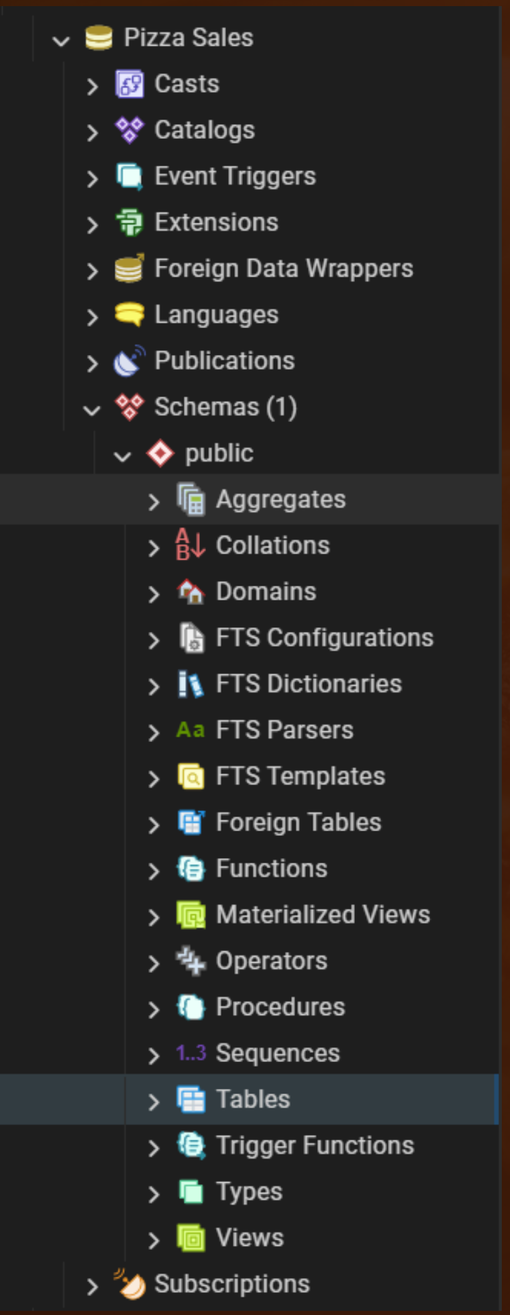
SQL Project

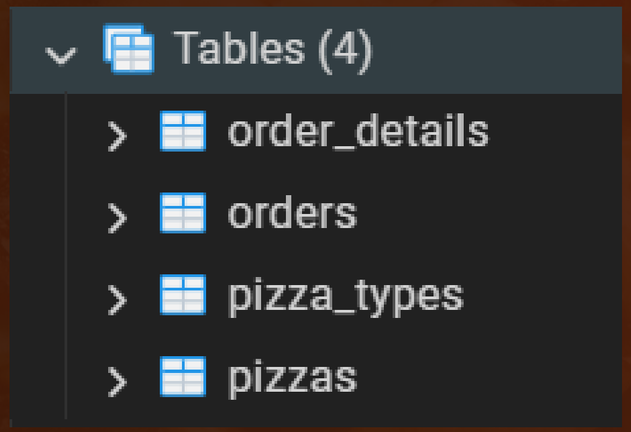
"LIFE IS BETTER WITH PIZZA!"

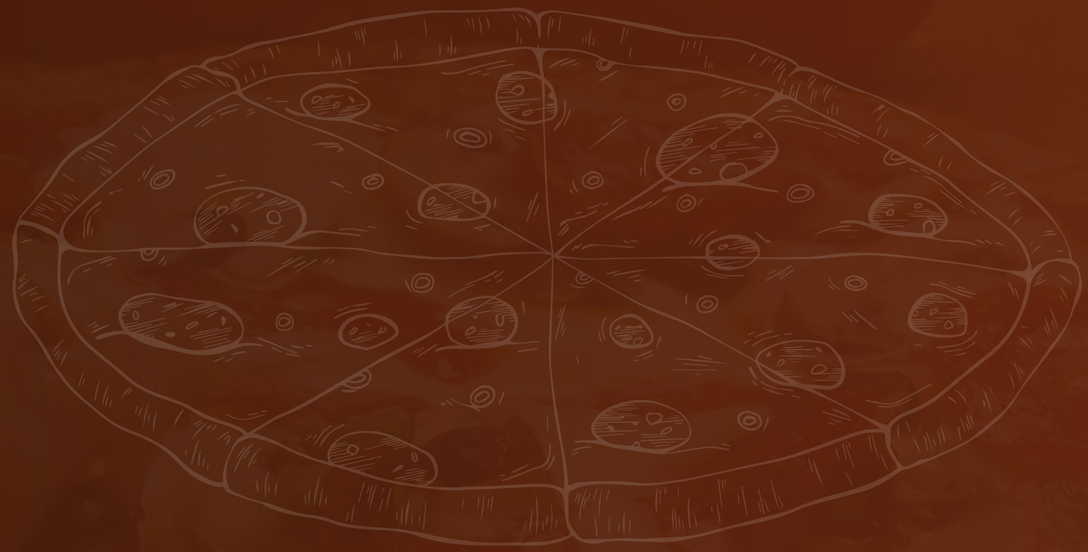
PIZZA SALES ANALYSIS

Intro:

This report provides insights into pizza sales trends, highlighting key factors such as best-selling pizza types, revenue distribution, peak sales periods, and customer preferences.







By analyzing sales data-
we identify growth opportunities,
optimize inventory, and
enhance marketing strategies to boost profitability.

```sql
-- Retrieve the total numbers of orders placed
select count(order_id) as total_orders from orders
```

Data Output | Messages | Notifications

	total_orders bigint
1	21350

```sql
--Calculate the total revenue generated by pizza sales

select round(sum(quantity*price)::numeric, 2) as total_revenue
from order_details as od
join pizzas as p on od.pizza_id=p.pizza_id
```

Data Output | Messages | Notifications | SQL

	total_revenue numeric
1	817860.05

```sql
--Identify the highest priced pizza
select p.pizza_id, p.pizza_type_id, pt.name, pt.category, p.price
from pizzas as p
join pizza_types as pt
on p.pizza_type_id=pt.pizza_type_id
order by p.price desc limit 1
```

Data Output | Messages | Notifications

	pizza_id character varying	pizza_type_id character varying	name character varying	category character varying	price double precision
1	the_greek_xxl	the_greek	The Greek Pizza	Classic	35.95

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

select p.size, count(od.order_id) as total_orders
from order_details as od
join pizzas as p
on od.pizza_id=p.pizza_id
group by size
order by total_orders desc
limit 1
```

Data Output | Messages | Notifications

	size character varying	total_orders bigint
1	L	18526

```sql
--List the top 5 most ordered pizza types along with their quantities

select pt.pizza_type_id, pt.name, sum(od.quantity) as total_quantity
from pizza_types as pt
join pizzas as p on pt.pizza_type_id=p.pizza_type_id
join order_details as od on od.pizza_id=p.pizza_id
group by pt.pizza_type_id, pt.name
order by total_quantity desc
limit 5
```

Data Output | Messages | Notifications

	pizza_type_id character varying	name character varying	total_quantity bigint
1	classic_dlx	The Classic Deluxe Pizza	2453
2	bbq_ckn	The Barbecue Chicken Pizza	2432
3	hawaiian	The Hawaiian Pizza	2422
4	pepperoni	The Pepperoni Pizza	2418
5	thai_ckn	The Thai Chicken Pizza	2371

```sql
-- Joins the necessary tabels to find the total quantity of each pizza category ordered

select pt.category, sum(od.quantity) as total_quantity
from pizza_types as pt
join pizzas as p on pt.pizza_type_id=p.pizza_type_id
join order_details as od on od.pizza_id=p.pizza_id
group by pt.category
order by total_quantity desc
```

Data Output | Messages | Notifications

	category character varying	total_quantity bigint
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

```sql
-- Determine the distribution of orders by hour of the day
select * from orders

select extract(hour from time) as hour , count(order_id) as total_orders
from orders
group by hour
order by total_orders desc
```

Data Output | Messages | Notifications

	hour numeric	total_orders bigint
1	12	2520
2	13	2455
3	18	2399
4	17	2336
5	19	2009
6	16	1920
7	20	1642

Total rows: 15 of 15 Query complete 00:00:04.270 Ln 7, Col 27

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

select category, count(name) as total
from pizza_types
group by category
order by total desc
```

Data Output | Messages | Notifications

	category character varying	total bigint
1	Supreme	9
2	Veggie	9
3	Classic	8
4	Chicken	6

```sql
--Group the orders by date and calculate the average number of pizzas ordered per day.
With date_quantity as
(select o.date, sum(od.quantity) as total_quantity
from orders as o
join order_details as od
on o.order_id=od.order_id
group by date
order by date asc
)
select round(avg(total_quantity),0) as avg_num_of_pizzas_per_day
from date_quantity
```

Data Output | Messages | Notifications

	avg_num_of_pizzas_per_day numeric
1	138

```sql
-- Determine the top 3 most ordered pizza types based on revenue

select pt.name, sum(p.price*od.quantity) as revenue
from pizza_types as pt
join pizzas as p on p.pizza_type_id=pt.pizza_type_id
join order_details as od on od.pizza_id=p.pizza_id
group by pt.name
order by revenue desc limit 3
```

Data Output | Messages | Notifications

	name character varying	revenue double precision
1	The Thai Chicken Pizza	43434.25
2	The Barbecue Chicken Pizza	42768
3	The California Chicken Pizza	41409.5

```sql
--Calculate the percentage contribution of each pizza type to total revenue.


With revenue as
(select pt.category, sum(od.quantity*p.price) as individual_revenue
from pizza_types as pt
join pizzas as p on p.pizza_type_id=pt.pizza_type_id
join order_details as od on od.pizza_id=p.pizza_id
group by pt.category
)
select revenue.category, round(((individual_revenue/total_revenue)*100)::numeric,2) as percentage_contribution
from revenue, (select sum(individual_revenue) as total_revenue from revenue) as total_revenue
order by percentage_contribution desc
```

Data Output | Messages | Notifications

	category character varying	percentage_contribution numeric
1	Classic	26.91
2	Supreme	25.46
3	Chicken	23.96
4	Veggie	23.68

```sql
-- Analyze the cumulative revenue generated over time

With date_revenue as
(select o.date, sum(price*quantity) as revenue
from orders as o
join order_details as od on o.order_id=od.order_id
join pizzas as p on p.pizza_id=od.pizza_id
group by o.date
)
select date_revenue.date, date_revenue.revenue,
sum(date_revenue.revenue) over (order by date_revenue.date) as cumulative_revenue
from date_revenue
```

Data Output | Messages | Notifications

SQL

	date date	revenue double precision	cumulative_revenue double precision
1	2015-01-01	2713.8500000000004	2713.8500000000004
2	2015-01-02	2731.8999999999996	5445.75
3	2015-01-03	2662.3999999999996	8108.15
4	2015-01-04	1755.4500000000003	9863.6
5	2015-01-05	2065.95	11929.55
6	2015-01-06	2428.95	14358.5
7	2015-01-07	2202.2000000000003	16560.7

Total rows: 358 of 358 Query complete 00:00:01.465 Ln 12

```sql
-- Determine thr top 3 most ordered pizza types based on revenue for each pizza category

With rank_table as
(With cat_rev as
(select pt.category, pt.name, sum(p.price*od.quantity) as revenue
from pizza_types as pt
join pizzas as p on p.pizza_type_id=pt.pizza_type_id
join order_details as od on od.pizza_id=p.pizza_id
group by pt.category, pt.name
)
select category, name, revenue, rank() over (partition by category order by revenue desc) as top_ranks
from cat_rev
)
select category, name, revenue, top_ranks
from rank_table
where top_ranks<=3
```

Data Output | Messages | Notifications

	category character varying	name character varying	revenue double precision	top_ranks bigint
1	Chicken	The Thai Chicken Pizza	43434.25	1
2	Chicken	The Barbecue Chicken Pizza	42768	2
3	Chicken	The California Chicken Pizza	41409.5	3
4	Classic	The Classic Deluxe Pizza	38180.5	1
5	Classic	The Hawaiian Pizza	32273.25	2
6	Classic	The Pepperoni Pizza	30161.75	3
7	Supreme	The Spicy Italian Pizza	34831.25	1

Total rows: 12 of 12 Query complete 00:00:02.401 Ln 3, Col 1

Thank You !