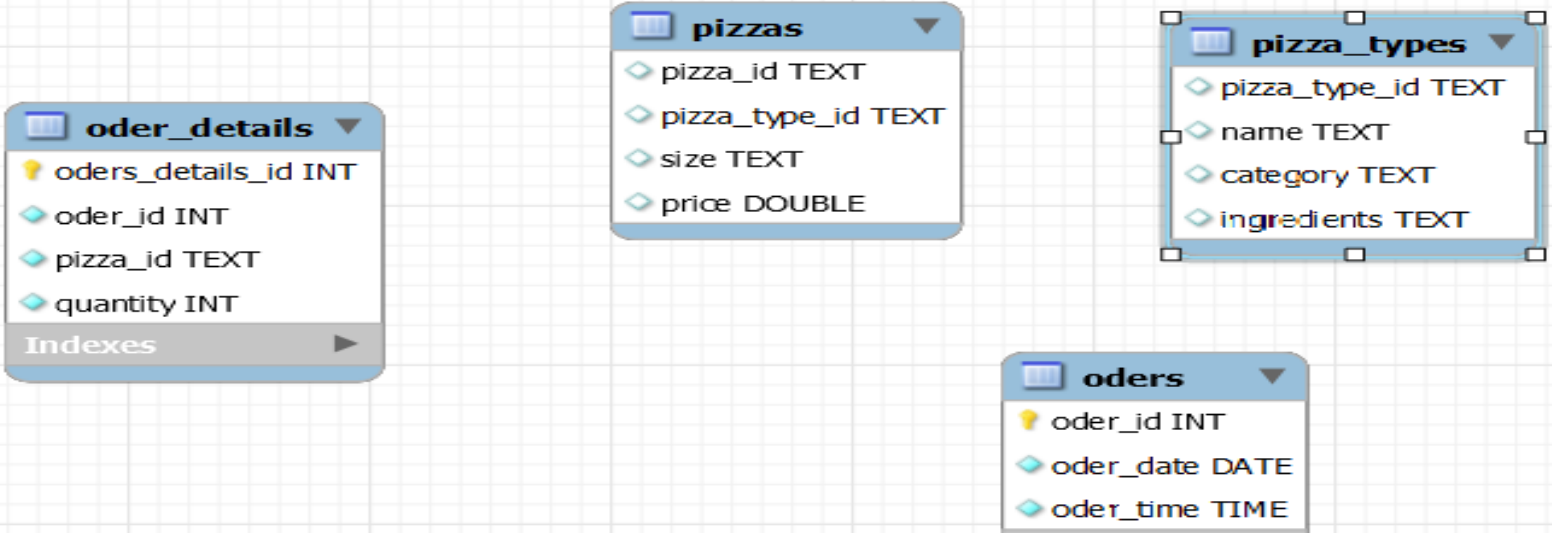
GOOGLE DATA ANALYTICS CAPSTONE PROJECT PIZZA_SALES

In this project I have utilize SQL queries about Pizza_sales

PIZZA_Sales Schema

- Tables



BASIC of MYSQL

oder_details | oders | pizza_types | SQL File 6* | SQL File 7* | SQL File 8* | SQL File 9* | SQL F

Limit to 1000 rows

```
1   /*Retrieve the total number of orders placed.*/
2   use pizzacon;
3   select count(oder_id) as total_oder
4   from oders;
```

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

total_oder

Identify the highest-priced pizza & Identify the most common pizza size ordered.

Identify the highest-priced pizza

```
1   /*Identify the highest-priced pizza.*/
2   use pizzacon;
3   select max(price)
4   from pizzas;
5
6   /*Identify the highest-priced pizza.*/
7   select pizza_types.name, pizzas.price
8   from pizza_types
9   join pizzas
```

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

name	price
The Greek Pizza	35.95

Identify the most common pizza size ordered

```
1   /*Identify the most common pizza size ordered.*/
2   select  pizzas.size, count(oder_details.oders_details_id) as order_count
3   from pizzas join oder_details
4    on pizzas.pizza_id = oder_details.pizza_id
5    group by pizzas.size
6    order by order_count desc ;
```

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

size	order_count
L	18526
M	15385
S	14137

List the top 5 most ordered pizza types along with their quantities. &Calculate the total revenue generated from pizza sales.

List the top 5 most ordered pizza types along with their quantities.

piza* | Administration - Server Status | SQL File 2* | SQL File 3* | SQL File 4* | SQL File 5* | oder_details

Limit to 1000 rows

```
30  /*List the top 5 most ordered pizza types along with their quantities.*/
31
32  select pizza_types.name, sum(oder_details.quantity) as quantity
33    from pizza_types join pizzas
34  on pizza_types.pizza_type_id = pizzas.pizza_type_id
35  join oder_details
36  on oder_details.pizza_id = pizzas.pizza_id
37  group by pizza_types.name
38  order by quantity  desc limit 5;
```

Result 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

Calculate the total revenue generated from pizza sales.

piza* | Administration - Server Status | SQL File 2* | SQL File 3*

Limit to 1000 rows

```
1  /* Calculate the total revenue generated from pizza sales.*/
2  select
3  sum(oder_details.quantity * pizzas.price)
4  from oder_details
5  join pizzas
6  on pizzas.pizza_id = oder_details.pizza_id;
```

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

sum(oder_details.quantity * pizzas.price)
817860.049999993

Intermediate level

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

```sql
/*Join the necessary tables to find the total quantity of each pizza category ordered.*/
select sum(oder_details.quantity) as quantity, pizza_types.category
from oder_details
join pizzas
 on oder_details.pizza_id = pizzas.pizza_id
join pizza_types
on pizza_types.pizza_type_id = pizzas.pizza_type_id
group by pizza_types.category
order by quantity desc;
```

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

quantity	category
14888	Classic
11987	Supreme

Determine the distribution of orders by hour of the day.

```sql
-- Determine the distribution of orders by hour of the day.
 select hour(oder_time) , count(oder_id)
 from oders
group by hour(oder_time);
```

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

hour(oder_time)	count(oder_id)
11	1231
12	2520
13	2455

Intermediate level

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

```
54
55  -- Join relevant tables to find the category-wise distribution of pizzas.
56  select category, count(name)
57  from  pizza_types
58  group by category;
```

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

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

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

Limit to 1000 rows

```
61  -- Group the orders by date and calculate the average
62  -- number of pizzas ordered per day.
63  select round(avg(quantity), 0)
64  from
65      (select oders.oder_date, sum(oder_details.quantity) as quantity
66          from oders
67           join oder_details
68            on oders.oder_id = oder_details.oder_id
69              group by oders.oder_date ) as data;
```

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

round(avg(quantity), 0)

Intermediate level

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

Limit to 1000 rows

```
2  •  select  pizza_types.name,
3     sum(oder_details.quantity * pizzas.price)  as total_revenue
4     from pizza_types
5     join pizzas
6     on pizza_types.pizza_type_id = pizzas.pizza_type_id
7     join oder_details
8     on oder_details.pizza_id =pizzas.pizza_id
9     group by pizza_types.name
10     order by total_revenue desc limit 3 ;
```

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

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

Advanced level

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

```
select pizza_types.category ,
    round( sum(oder_details.quantity * pizzas.price)  /
(select     round(sum(oder_details.quantity * pizzas.price) ,2)
as  total_revenue
from oder_details
      join pizzas
     on pizzas.pizza_id = oder_details.pizza_id) *100 ,2)
      as t_revenue
from pizzas
   join pizza_types  on pizzas.pizza_type_id =pizza_types.pizza_type_id
 join oder_details
        on oder_details.pizza_id = pizzas.pizza_id
         group by  pizza_types.category
          order by t_revenue desc;
```

- Output

Result Grid | Filter R

	category	t_revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Advanced level

- Analyze the cumulative revenue generated over time.

Limit to 1000 rows

```
2  select oder_date, sum(revenue) over(order by oder_date) as cum_rev
3  from
4  (select oders.oder_date, sum(oder_details.quantity * pizzas.price) as revenue
5  from oder_details
6  join pizzas
7  on oder_details.pizza_id = pizzas.pizza_id
8  join oders
9  on oders.oder_id = oder_details.oder_id
10 group by oders.oder_date) as sales;
```

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

oder_date	cum_rev
2015-01-01	2713.8500000000004
2015-01-02	5445.75
2015-01-03	8108.15
2015-01-04	9863.6

Advanced level

- **Determine the top 3 most ordered pizza types based on revenue for each pizza category.**

```
2  select category,name,revenue, rank() over(partition by category order by revenue) as rn
3  from
4  (select pizza_types.category, pizza_types.name, sum(oder_details.quantity * pizzas.price) as revenue
5  from pizza_types
6  join pizzas
7  on pizza_types.pizza_type_id = pizzas.pizza_type_id
8  join oder_details
9  on oder_details.pizza_id = pizzas.pizza_id
10 group by  pizza_types.category, pizza_types.name) as s;
```

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

category	name	revenue	rn
Chicken	The Chicken Pesto Pizza	16701.75	1
Chicken	The Chicken Alfredo Pizza	16900.25	2