DOMINO'S SALES ANALYSIS

ANURAG BISHT

OBJECTIVE

The Pizza Sales Project is a data analysis project aimed at analysing sales data from a pizza restaurant chain. This project utilizes SQL for data extraction and transformation. The goal of this project is to provide insights and actionable information to help the pizza restaurant chain optimize its operations, improve sales, and enhance customer satisfaction.

DATABASE

order_details

- order_details_id INT
- order_id INT
- pizza_id TEXT
- quantity INT

pizza_type

- pizza_type_id VARCHAR(45)
- name VARCHAR(65)
- category VARCHAR(45)
- ingredients VARCHAR(100)

orders

- order_id INT
- order_date DATE
- order_time TIME

Indexes

pizzas

- pizza_id TEXT
- pizza_type_id TEXT
- size TEXT
- price DOUBLE

LEVEL OF QUERIES

BASIC **INTERMEDIATE** **ADVANCED**

Includes

Select, where,
Group, order, joins

Includes

Multi-Joins, Group,
Limit, Sub-queries

Includes

Windows Function,
CTE

BASIC

Retrieve the total number of orders placed.

Solution :

```
 9
10 •    select count(*) AS Total_orders
11      from dominos.orders;
12
```

Result :

Result Grid Filter Rows: Search Export:

Total_orders
21350

Result 1

BASIC

Calculate the total revenue generated from pizza sales.

Solution :

```
select sum(pizzas.price*order_details.quantity) as total_revenue from
dominos.pizzas
join
dominos.order_details on
pizzas.pizza_id=order_details.pizza_id;
```

Result :

Result Grid Filter Rows: Search

total_revenue
817860.049999993

Result 3

BASIC

Identify the highest-priced pizza.

Solution :

```
select pizza_type.name , price
from dominos.pizzas join dominos.pizza_type on
pizzas.pizza_type_id=pizza_type.pizza_type_id
where price = (select max(price) from pizzas);
```

Result :

Result Grid Filter Rows: Search

name	price
The Greek Pizza	35.95

BASIC

Identify the most common pizza size ordered.

Solution :

```
select pizzas.size,count(order_details.quantity) as total
from dominos.pizzas join
dominos.order_details on
pizzas.pizza_id=order_details.pizza_id
group by pizzas.size
order by total desc limit 1;
```

Result :

Result Grid Filter Rows:

size	total
L	18526

BASIC

Retrieve the total number of orders placed.

Solution :

```
select pizza_type.name, sum(order_details.quantity) as times
from order_details
join
pizzas on
order_details.pizza_id=pizzas.pizza_id
join pizza_type on
pizza_type.pizza_type_id=pizzas.pizza_type_id
group by pizza_type.name
order by times desc limit 5;
```

Result :

Result Grid Filter Rows: Search

name	times
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

INTERMEDIATE

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

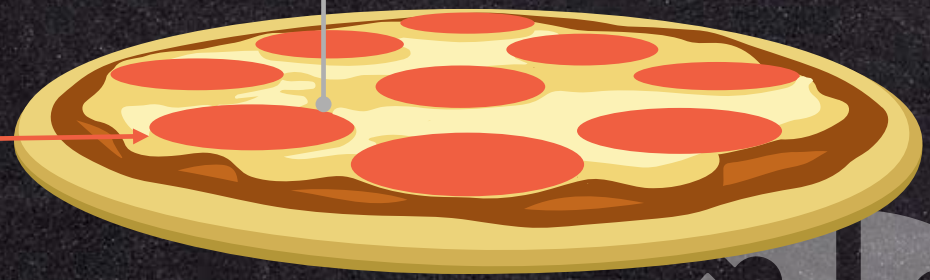
Solution :

```
select pizza_type.category, sum(order_details.quantity) as times
from order_details
join
pizzas on
order_details.pizza_id=pizzas.pizza_id
join pizza_type on
pizza_type.pizza_type_id=pizzas.pizza_type_id
group by pizza_type.category;
```

Result :

Result Grid Filter Rows:

category	times
Classic	14888
Veggie	11649
Supreme	11987
Chicken	11050



INTERMEDIATE

Determine the distribution of orders by hour of the day.

Solution :

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

Result :

Result Grid Filter Rows:

total_order	hour
2520	12
2455	13
2399	18
2336	17
2009	19
1920	16
1642	20

INTERMEDIATE

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

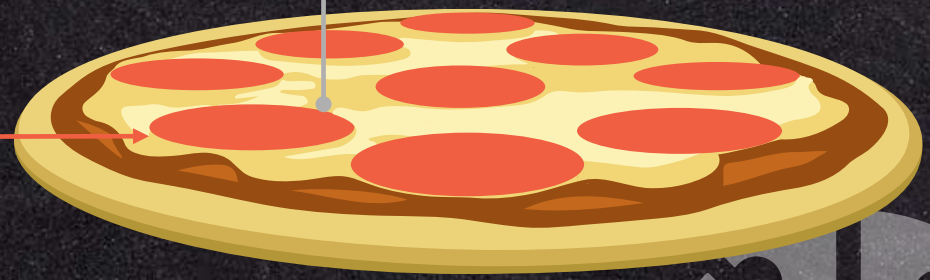
Solution :

```
select category ,count(pizza_type.name) as total_pizzas
from pizza_type
group by category;
```

Result :

Result Grid Filter Rows:

category	total_pizzas
Chicken	6
Classic	8
Supreme	9
Veggie	9



INTERMEDIATE

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

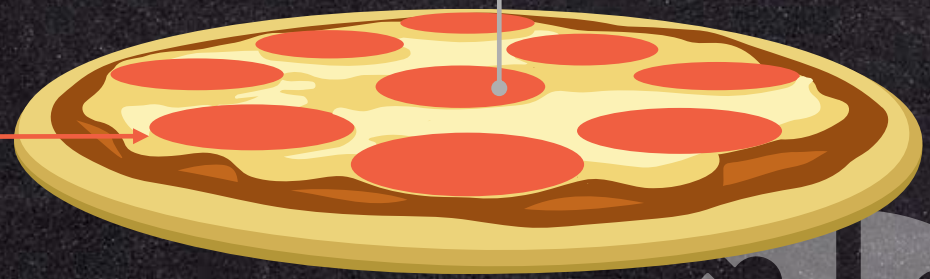
Solution :

```
select round(avg(total_order))from
(select orders.order_date as date, sum(order_details.quantity) as total_order
from orders join order_details on
orders.order_id=order_details.order_id
group by date)as total;
```

Result :

Result Grid | Filter Rows:

round(avg(total_order))
138



INTERMEDIATE

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

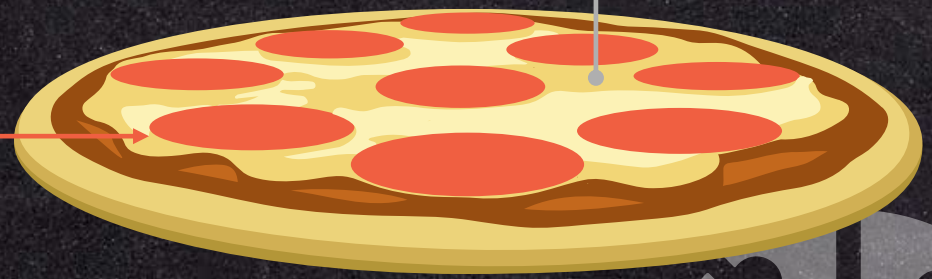
Solution :

```sql
select pizza_type.name, sum(pizzas.price*order_details.quantity) as total_revenue from
pizzas
join
order_details on
pizzas.pizza_id=order_details.pizza_id
join pizza_type on
pizzas.pizza_type_id= pizza_type.pizza_type_id
group by pizza_type.name
order by total_revenue desc limit 3;
```

Result :

Result Grid Filter Rows: Search

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



ADVANCED

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

Solution :

```
select pizza_type.name, sum(pizzas.price*order_details.quantity) as total_revenue,
round(sum(pizzas.price*order_details.quantity)/ (select sum(pizzas.price*order_details.quantity) as total_revenue
from
dominos.pizzas
join
dominos.order_details on
pizzas.pizza_id=order_details.pizza_id) * 100,2) as total_sales
from
pizzas
join
order_details on
pizzas.pizza_id=order_details.pizza_id
join pizza_type on
pizzas.pizza_type_id= pizza_type.pizza_type_id
group by pizza_type.name;
```

Result :

Result Grid Filter Rows: Search Export:

name	total_revenue	total_sales
The Hawaiian Pizza	32273.25	3.95
The Classic Deluxe Pizza	38180.5	4.67
The Five Cheese Pizza	26066.5	3.19
The Italian Supreme Pizza	33476.75	4.09
The Mexicana Pizza	26780.75	3.27
The Thai Chicken Pizza	43434.25	5.31
The Prosciutto and Arugula Pizza	24193.25	2.96
The Barbecue Chicken Pizza	42768	5.23
The Greek Pizza	28454.100000000013	3.48
The Spinach Supreme Pizza	15277.75	1.87
The Green Garden Pizza	13955.75	1.71

ADVANCED

- Analyze the cumulative revenue generated over time.

Solution :

```
with my_cte as (select orders.order_date, round(sum(pizzas.price*order_details.quantity)) as revenue
from pizzas
join
order_details on
pizzas.pizza_id = order_details.pizza_id
join
orders on
orders.order_id=order_details.order_id
group by order_date
order by order_date)
select order_date,revenue,sum(revenue) over(order by order_date rows unbounded preceding) as cumul_revenue
from my_cte;
```

Result :

Result Grid Filter Rows: Search

order_date	revenue	cumul_revenue
2015-01-03	2662	8108
2015-01-04	1755	9863
2015-01-05	2066	11929
2015-01-06	2429	14358
2015-01-07	2202	16560
2015-01-08	2838	19398
2015-01-09	2127	21525
2015-01-10	2464	23989
2015-01-11	1872	25861
2015-01-12	1919	27780
2015-01-13	2050	29830
2015-01-14	2527	32357

ADVANCED

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

Solution :

```sql
with my_cte as (select category,name,revenue, rank()over(Partition by category order by revenue desc) as ranking
from
(select pizza_type.category,pizza_type.name,
sum(pizzas.price*order_details.quantity) as revenue
from pizzas
join
order_details on
pizzas.pizza_id=order_details.pizza_id
join pizza_type on
pizzas.pizza_type_id= pizza_type.pizza_type_id
group by pizza_type.name, pizza_type.category
order by pizza_type.category) as total)
select * from my_cte
where ranking <=3;
```

Result :

Result Grid Filter Rows: Search Export:

category	name	revenue	ranking
Chicken	The Thai Chicken Pizza	43434.25	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41409.5	3
Classic	The Classic Deluxe Pizza	38180.5	1
Classic	The Hawaiian Pizza	32273.25	2
Classic	The Pepperoni Pizza	30161.75	3
Supreme	The Spicy Italian Pizza	34831.25	1
Supreme	The Italian Supreme Pizza	33476.75	2
Supreme	The Sicilian Pizza	30940.5	3
Veggie	The Four Cheese Pizza	32265.70000000065	1
Veggie	The Mexicana Pizza	26780.75	2
Veggie	The Five Cheese Pizza	26066.5	3


THANK YOU
ANURAG BISHT