DOMINOS DATA ANALYTICS

(USING SQL)

LET'S START!

BY : HUZAIFA SHEIKH

DATASET UPLOADED
ON MY GITHUB PAGE

LINK IN BIO :)

TOTAL NUMBER OF ORDERS PLACED.

QUERY :

```sql
SELECT
    COUNT(order_id) AS total_orders
FROM
    orders;
```

RESULT :

	total_orders
▶	21350

HIGHEST PRICED PIZZA.

QUERY :

```sql
SELECT name,price AS highest_price
FROM pizzas JOIN pizza_types
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```
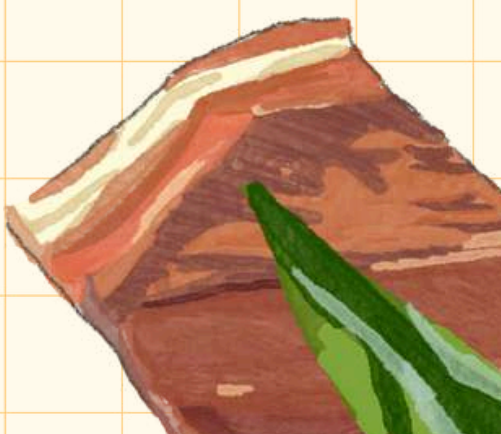
[USINGJOINS]

RESULT :

	name	highest_price
▶	The Greek Pizza	35.95

MOST COMMON PIZZA SIZE ORDERED

QUERY :

```
SELECT size,COUNT(order_details_id) AS ordered

FROM

pizzas JOIN orders_details

ON pizzas.pizza_id = orders_details.pizza_id

GROUP BY size

ORDER BY ordered DESC

LIMIT 1;
```

RESULT :

	size	ordered
▶	L	18526

USING :-
1] JOINS,
2] AGGREGATE FUNCTIONS,
3] ORDER BY

TOP 5 MOST ORDERED PIZZA TYPES WITH QUANTITIES.

QUERY :

```sql
SELECT pizza_types.name, SUM(orders_details.quantity) as quantity
FROM pizzas
JOIN orders_details
ON pizzas.pizza_id = orders_details.pizza_id
JOIN pizza_types
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY name
ORDER BY quantity DESC
LIMIT 5;
```

RESULT :

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

USING :-
1] TRIPLE JOINS,
2] AGGREGATE FUNCTIONS,
3] ORDER BY

TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

QUERY :

```sql
SELECT category, SUM(quantity) as quantity
FROM pizzas
JOIN pizza_types
ON pizzas.pizza_type_id = pizza_types.pizza_type_id
JOIN orders_details
ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY category
ORDER BY quantity DESC;
```

RESULT :

category	quantity
▶ Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

DISTRIBUTION OF ORDERS BY HOUR OF THE DAY

QUERY :
```
SELECT hour(order_time) AS hours, count(order_id) AS order_calls
FROM orders
GROUP BY hours;
```
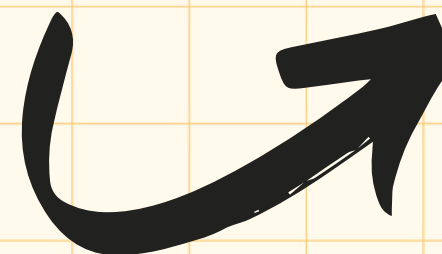
(CALL FROM CUSTOMERS)

RESULT :

hours	order_calls
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

CATEGORY-WISE DISTRIBUTION OF PIZZAS.

QUERY:

```sql
SELECT category,COUNT(pizza_type_id) AS types
FROM pizza_types
GROUP BY category;
```

RESULT:

category	types
Chicken	6
Classic	8
Supreme	9
Veggie	9

Average Number of Pizzas Ordered Per Day.

QUERY :

```sql
SELECT AVG(tot_pizzas) AS averageOrder_day
FROM
(SELECT orders.order_date, sum(orders_details.quantity) as tot_pizzas
FROM orders
JOIN orders_details
ON orders.order_id = orders_details.order_id
GROUP BY orders.order_date) AS order_quantity;
```

RESULT :

averageOrder_day
138.4749

USING :-
1] JOINS,
2] AGGREGATE FUNCTIONS,
3] NESTED QUERIES

TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

QUERY :

```sql
SELECT pizza_types.name, SUM(price*quantity) as T_cost
FROM pizza_types
JOIN pizzas
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN orders_details
ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY T_cost DESC
LIMIT 3;
```

RESULT :

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

PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

QUERY :

```sql
SELECT pizza_types.category, ROUND(SUM(price * quantity)/
(SELECT ROUND(SUM(price * quantity),0) FROM pizzas
JOIN orders_details
ON pizzas.pizza_id = orders_details.pizza_id)*100,1)
AS revenue FROM pizzas JOIN pizza_types
ON pizzas.pizza_type_id = pizza_types.pizza_type_id
JOIN orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY category;
```

RESULT :

category	revenue
Classic	26.9
Veggie	23.7
Supreme	25.5
Chicken	24

CUMULATIVE REVENUE GENERATED OVER TIME.

QUERY :

```sql
SELECT order_date, SUM(amount) OVER (ORDER BY order_date)
AS cumulative_revenue FROM
(SELECT order_date, ROUND(SUM(price * quantity),0) AS amount
FROM orders JOIN orders_details
ON orders.order_id = orders_details.order_id
JOIN pizzas ON pizzas.pizza_id = orders_details.pizza_id
GROUP BY order_date ) AS revenue;
```

RESULT :

order_date	cumulative_revenue
2015-01-01	2714
2015-01-02	5446
2015-01-03	8108
2015-01-04	9863
2015-01-05	11929
2015-01-06	14358
2015-01-07	16560

2015-01-08	19398
2015-01-09	21525
2015-01-10	23989
2015-01-11	25861
2015-01-12	27780
2015-01-13	29830
2015-01-14	32357
2015-01-15	34342
2015-01-16	36936

USING :-

1] WINDOW FUNCTIONS

2] TRIPLE JOINS,

3] AGGREGATE FUNCTIONS

TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY

QUERY :

```sql
SELECT name,revenue,rn
FROM (SELECT category, name, revenue,
RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS rn
FROM (SELECT pizza_types.category, pizza_types.name,
sum((orders_details.quantity) * pizzas.price)
AS revenue FROM pizza_types JOIN pizzas
ON pizza_types.pizza_type_id = pizzas.pizza_type_id
JOIN orders_details ON orders_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category, pizza_types.name) AS a) AS b
WHERE rn <=3;
```

USING :

WINDOW FUNCTIONS

IN NESTED QUERY

RESULT :

name	revenue	rn
The Thai Chicken Pizza	43434.25	1
The Barbecue Chicken Pizza	42768	2
The California Chicken Pizza	41409.5	3
The Classic Deluxe Pizza	38180.5	1
The Hawaiian Pizza	32273.25	2
The Pepperoni Pizza	30161.75	3

The Spicy Italian Pizza	34831.25	1
The Italian Supreme Pizza	33476.75	2
The Sicilian Pizza	30940.5	3
The Four Cheese Pizza	32265.70000000065	1
The Mexicana Pizza	26780.75	2
The Five Cheese Pizza	26066.5	3

THANK YOU

UNVEILING INSIGHTS,
ONE QUERY AT A TIME.