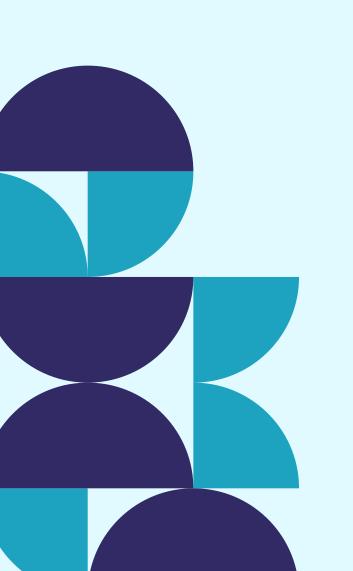


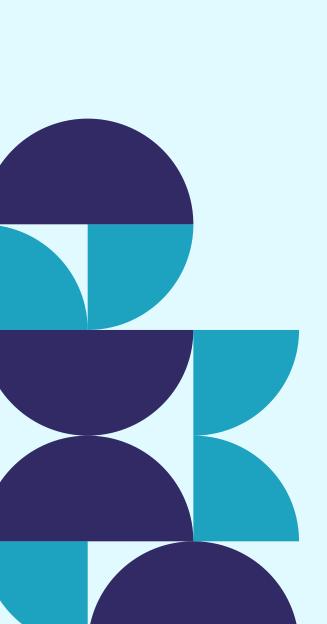
SQL PROJECT

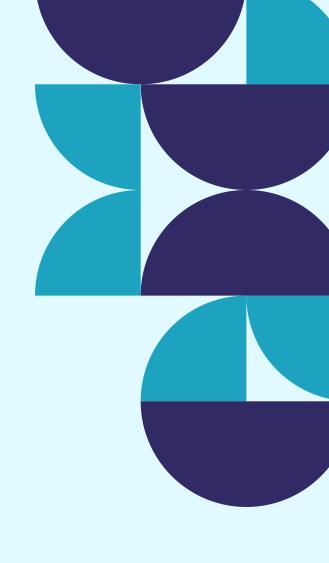
Presented By Shreya Dixit



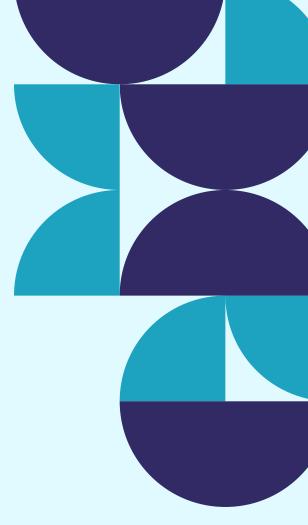
INTRODUCTION

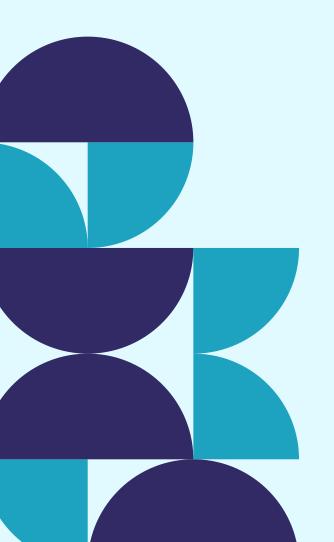
This project consists of some SQL commands which is run on a database to give a practical learning idea.





DATABASE

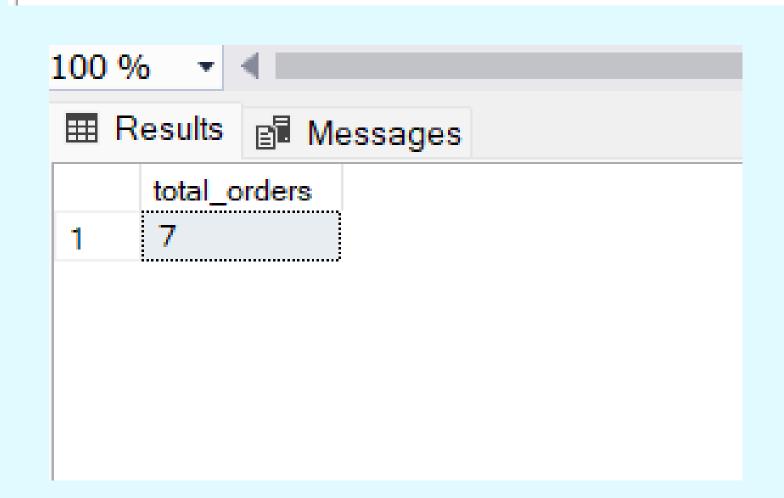


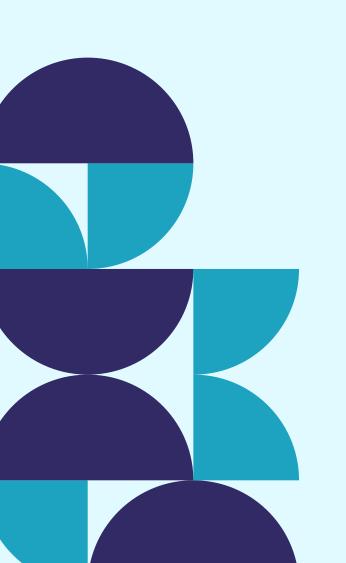


use pizzahut;

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

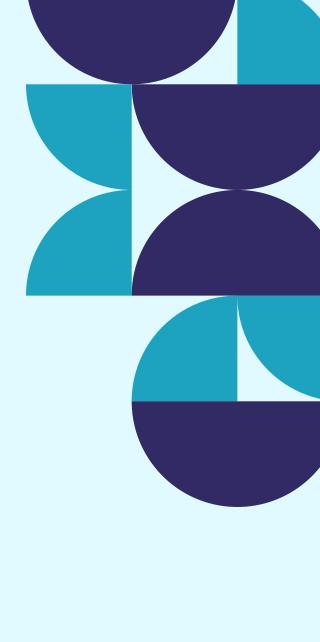
```
select count(order_id) as total_orders from orders;
```





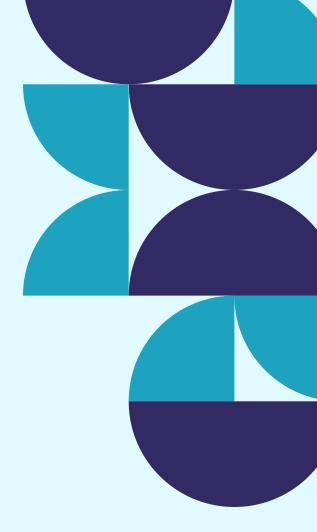
CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
3. SELECT
4 ROUND(SUM(order_details.quantity * pizzas.price),
5 2) AS total_sales
6 FROM
7 order_details
8 JOIN
9 pizzas ON pizzas.pizza_id = order_details.pizza_id
```



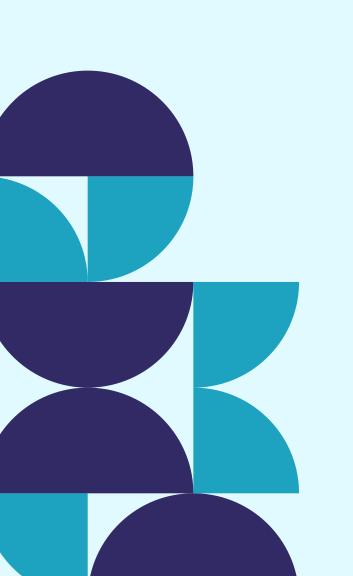
IDENTIFY THE HIGHEST-PRICED PIZZA.

```
3. SELECT
4 pizza_types.name, pizzas.price
5 FROM
6 pizza_types
7 JOIN
8 pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
9 ORDER BY pizzas.price DESC
0 LIMIT 1;
```



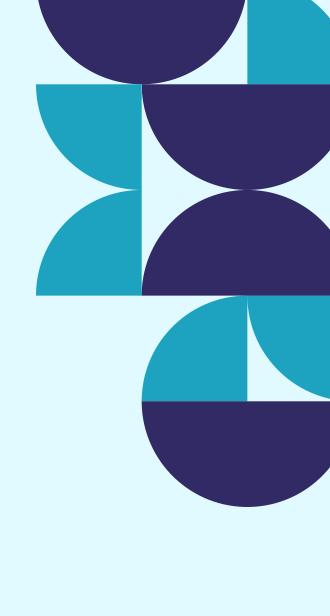
LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
4 · SELECT
         pizza_types.name, SUM(order_details.quantity) AS quantity
    FROM
         pizza_types
              JOIN
         pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
              JOIN
10
         order_details ON order_details.pizza_id = pizzas.pizza_id
    GROUP BY pizza_types.name
    ORDER BY quantity DESC
    LIMIT 5;
                      Export: Wrap Cell Content: IA
  The Barbecue Chicken Pizza 2432
```



JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```
4 * select pizza_types.category,
5 sum(order_details.quantity) as quantity
6 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
10 group by pizza_types.category order by quantity desc;
                  Export: Wrap Cell Content: IA
     11649
     11050
```



DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

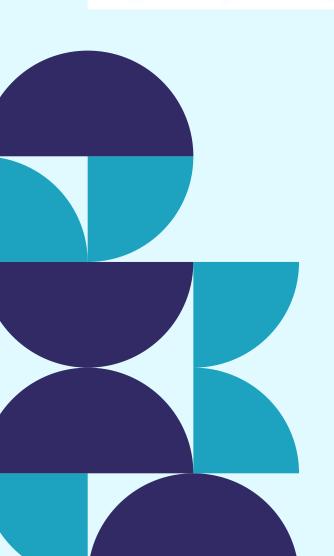
```
SELECT

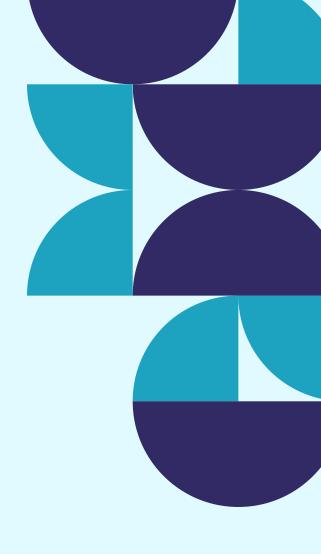
HOUR(order_time) AS hour, COUNT(order_id) AS order_count

FROM

orders

GROUP BY HOUR(order_time);
```





ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select order_date,
sum(revenue) over(order by order_date) as cum_revenue
from
(select orders.order_date,
sum(order_details.quantity * pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.order_date) as sales;
```



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

```
select category, name, revenue,
rank() over(partition by category order by revenue desc) as rn
from
(select pizza_types.category, pizza_types.name,
sum((order_details.quantity) * pizzas.price) as revenue
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, pizza_types.name) as a;
```

