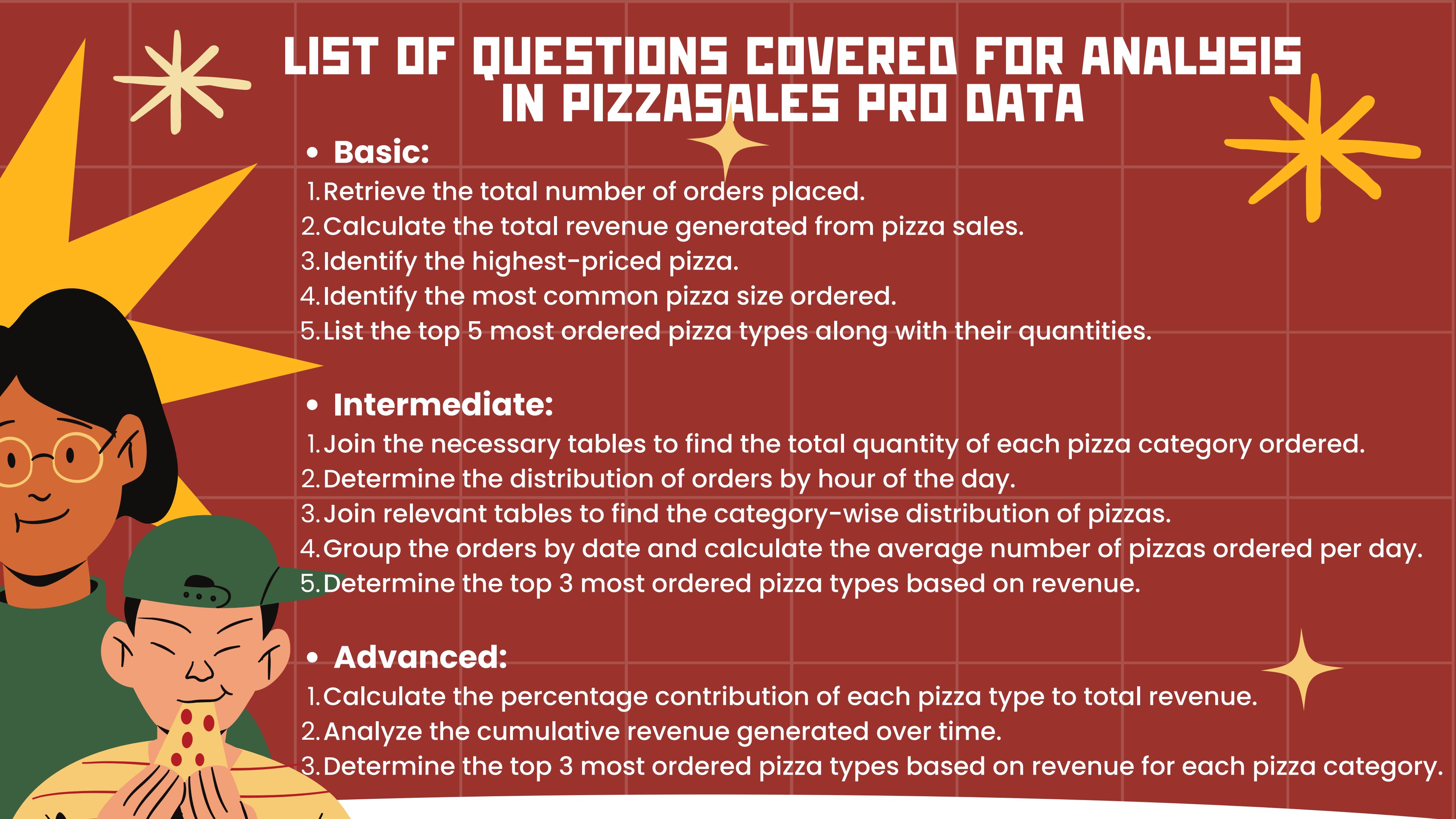


PizzaSales Pro





LIST OF QUESTIONS COVERED FOR ANALYSIS IN PIZZASALES PRO DATA

- **Basic:**

1. Retrieve the total number of orders placed.
2. Calculate the total revenue generated from pizza sales.
3. Identify the highest-priced pizza.
4. Identify the most common pizza size ordered.
5. List the top 5 most ordered pizza types along with their quantities.

- **Intermediate:**

1. Join the necessary tables to find the total quantity of each pizza category ordered.
2. Determine the distribution of orders by hour of the day.
3. Join relevant tables to find the category-wise distribution of pizzas.
4. Group the orders by date and calculate the average number of pizzas ordered per day.
5. Determine the top 3 most ordered pizza types based on revenue.

- **Advanced:**

1. Calculate the percentage contribution of each pizza type to total revenue.
2. Analyze the cumulative revenue generated over time.
3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.

Q : Retrieve the total number of orders placed.

SQL Code :

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

output :

total_orders

21350

Q: Calculate the total revenue generated from pizza sales.

SQL Code :

```
SELECT  
    round(sum(order_details.quantity * pizzas.price),  
        2) AS total_revenue  
  
FROM  
    order_details  
    JOIN  
    pizzas ON pizzas.pizza_id = order_details.pizza_id
```



output :

total_revenue
817860.05

Q : Identify the highest-priced pizza.

SQL Code :

```
SELECT * FROM pizza.pizza_types;  
  
SELECT  
    pizza_types.name, pizzas.price  
FROM  
    pizza_types  
        JOIN  
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
ORDER BY pizzas.price DESC  
LIMIT 1;
```

output :

	name	price
→	The Greek Pizza	35.95

Q : Identify the most common pizza size ordered.

SQL Code :

```
SELECT  
    pizzas.size,  
    COUNT(order_details.order_details_id) AS order_count  
FROM  
    pizzas  
        JOIN  
    order_details ON pizzas.pizza_id = order_details.pizza_id  
GROUP BY pizzas.size  
ORDER BY order_count DESC;
```

Output :

	size	order_count
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28

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

SQL Code :

```
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
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY quantity DESC;
```

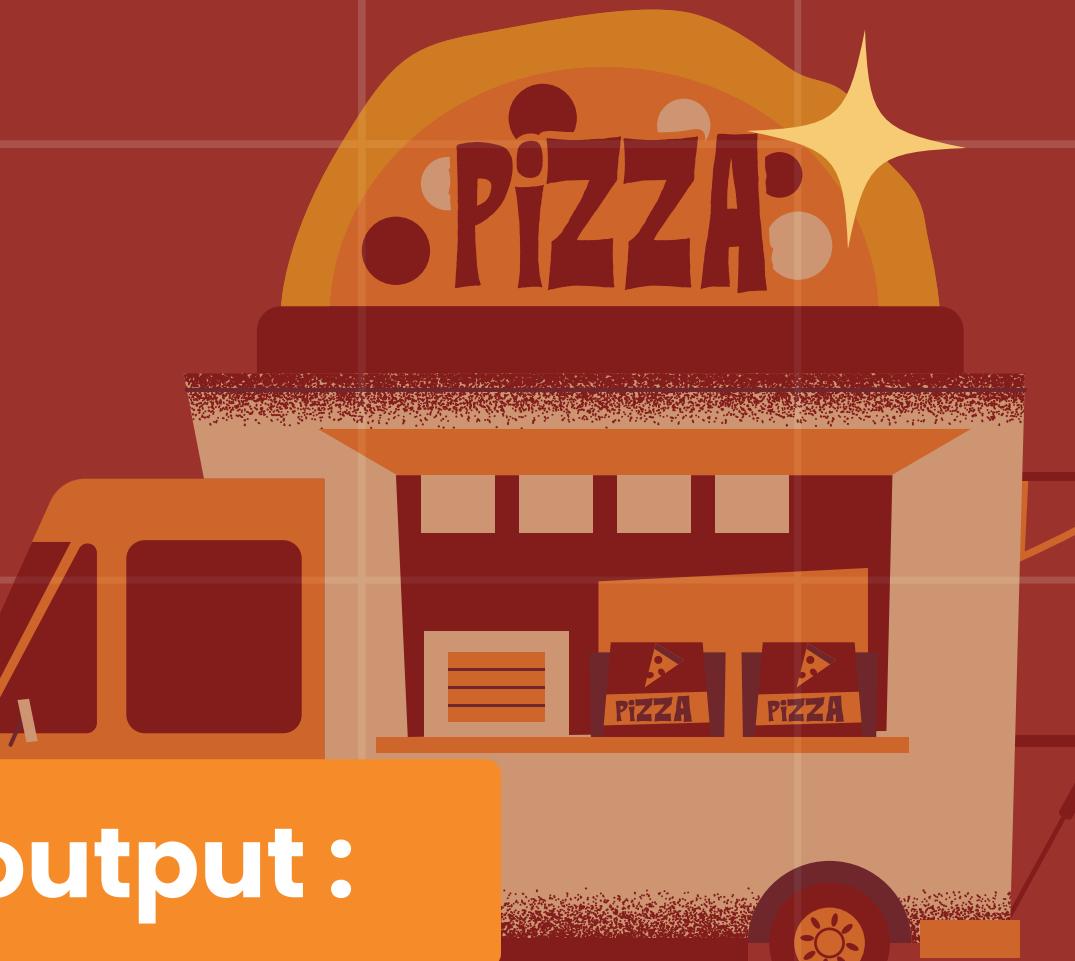
output :

	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
	The California Chicken Pizza	2370
	The Sicilian Pizza	1938
	The Spicy Italian Pizza	1924
	The Southwest Chicken Pizza	1917
	The Big Meat Pizza	1914

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

SQL Code :

```
SELECT  
    pizza_types.category,  
    SUM(order_details.quantity) AS quantity  
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  
ORDER BY quantity DESC;
```



output:

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Q : Determine the distribution of orders by hour of the day.

SQL Code :

```
SELECT  
    HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
FROM  
    orders  
GROUP BY HOUR(order_time);
```

output :

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642

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



SQL Code :

```
select category , count(name) from pizza_types  
group by category
```



output :

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

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



SQL Code :

```
SELECT  
    ROUND(AVG(quantity), 0)  
FROM  
(SELECT  
    orders.order_date, SUM(order_details.quantity) AS quantity  
FROM  
    orders  
JOIN order_details ON orders.order_id = order_details.order_id  
GROUP BY orders.order_date) AS order_quantity;
```

output :

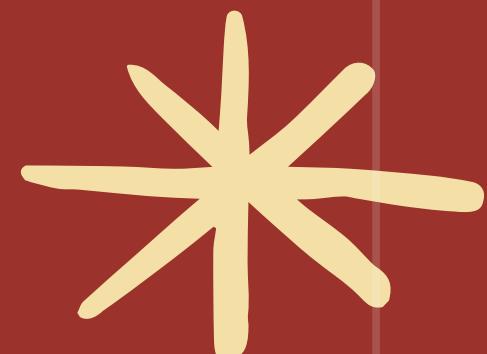
ROUND(AVG(quantity), 0)

138

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

SQL Code :

```
SELECT  
    pizza_types.name,  
    SUM(order_details.quantity * pizzas.price) AS revenue  
FROM  
    pizza_types  
    JOIN  
    pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
    JOIN  
    order_details ON order_details.pizza_id = pizzas.pizza_id  
GROUP BY pizza_types.name  
ORDER BY revenue DESC  
LIMIT 3;
```



output :

	name	revenue
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5



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

SQL Code :

```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
        ROUND(SUM(order_details.quantity * pizzas.price),
        2) AS total_sales
    )
    FROM
        order_details
        JOIN
            pizzas ON pizzas.pizza_id = order_details.pizza_id) * 100,
    2) 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
ORDER BY revenue DESC;
```



output :

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Q : Analyze the cumulative revenue generated over time.

SQL Code :

```
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;
```

output :

	order_date	cum_revenue
▶	2015-01-01	2713.850000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	19399.05
	2015-01-09	21526.4
	2015-01-10	23990.350000000002

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

SQL Code :

```
select name, revenue 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((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) as b  
where rn <= 3;
```

output :

	name	revenue
►	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5
	The Classic Deluxe Pizza	38180.5
	The Hawaiian Pizza	32273.25
	The Pepperoni Pizza	30161.75
	The Spicy Italian Pizza	34831.25



A festive illustration set against a red background with a white grid. In the center, the words "THANK YOU" are written in large, bold, white capital letters. To the left, a person with dark curly hair and a green sweater holds a slice of pizza. To the right, another person with glasses and a green sweater holds a small wrapped gift. Above them, a reindeer with a yellow and orange patterned collar and a bell hangs from its neck. The reindeer has large, expressive eyes and a small smile. The background is decorated with yellow stars and a large yellow starburst at the bottom.

THANK YOU