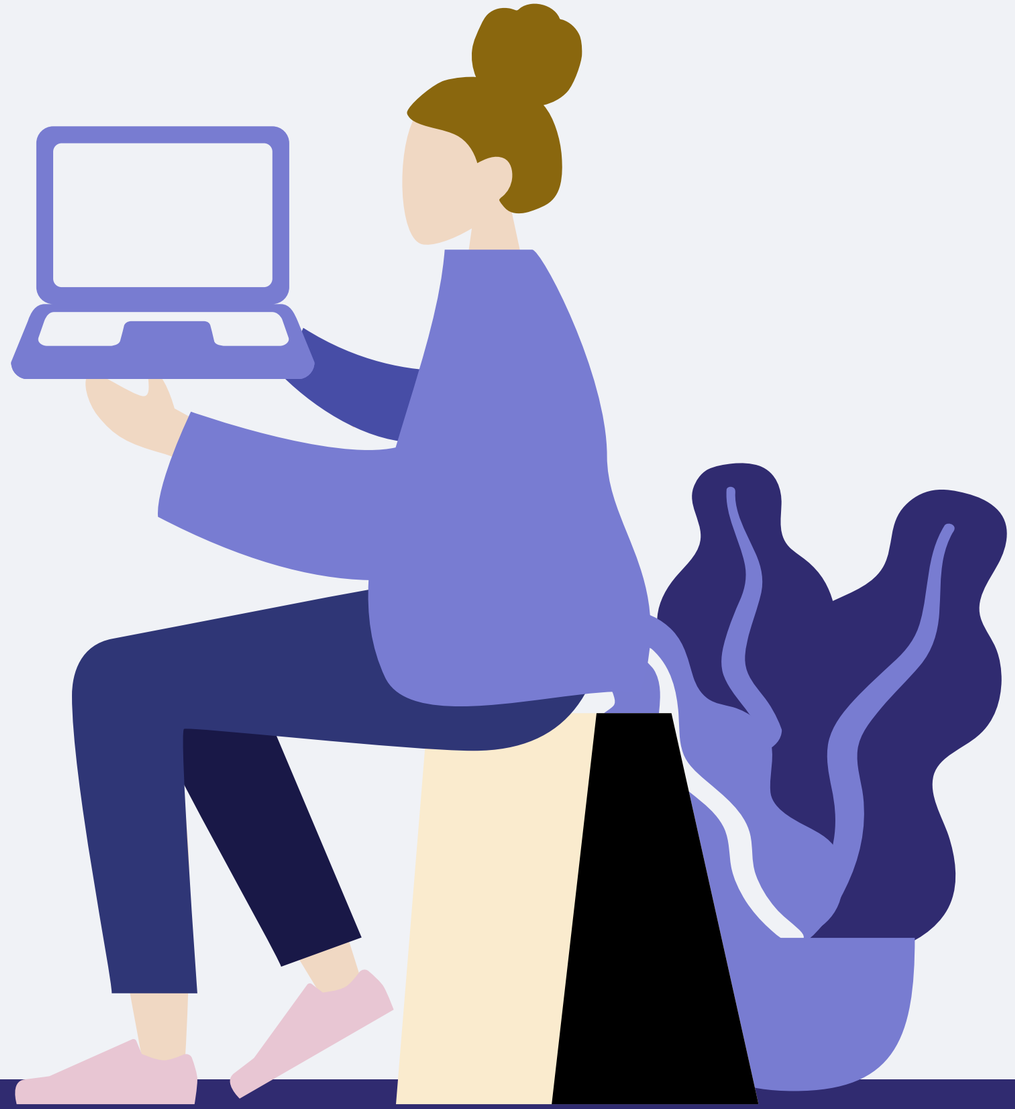
Pizza sales analysis

This presentation displays a thorough analysis of pizza sales in a company



Next, there are some queries.

I utilize SQL to answer those queries. Please look forward on them.

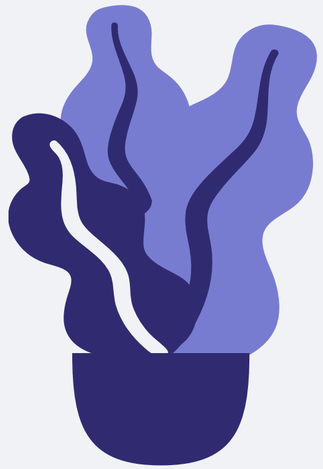
01

Retrieve the total number of orders placed.

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

Solution

	total_orders
▶	21350



02

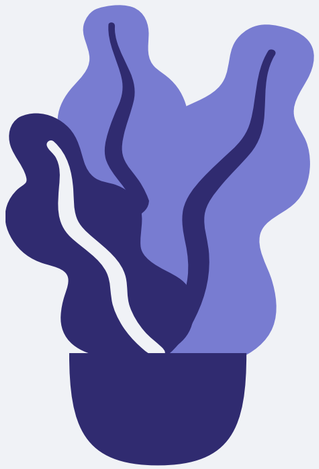
Calculate the total revenue generated from pizza sales.

```
.ECT
 SUM(orders_details.quantity * pizzas.price) AS total_revenu
)M
 orders_details
     JOIN
 pizzas ON pizzas.pizza_id = orders_details.pizza_id
```

Solution

Result Grid | Filt

	total_revenue
▶	817860.04999993



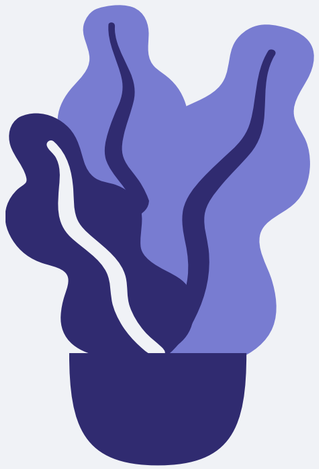
03

Identify the highest-priced pizza.

```
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;
```

Solution

	name	price
▶	The Greek Pizza	35.95



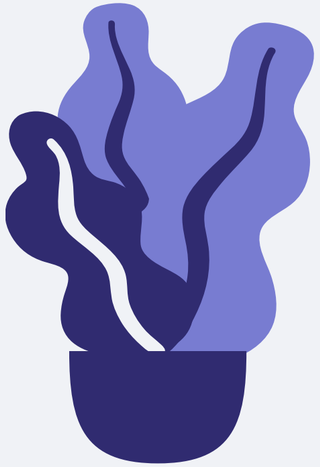
04

Identify the most common pizza size ordered.

Solution

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

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



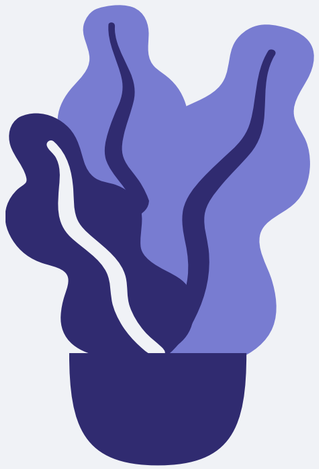
05

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

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

Solution

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



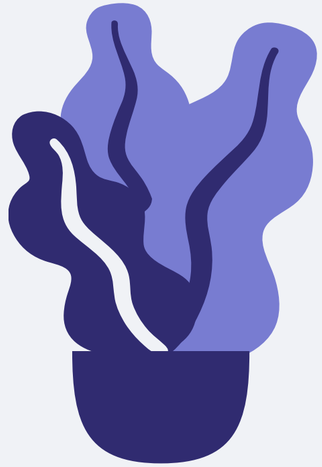
06

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

```sql
SELECT
    pizza_types.name, SUM(orders_details.quantity) AS quantity
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 quantity DESC
LIMIT 5;
```

Solution

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



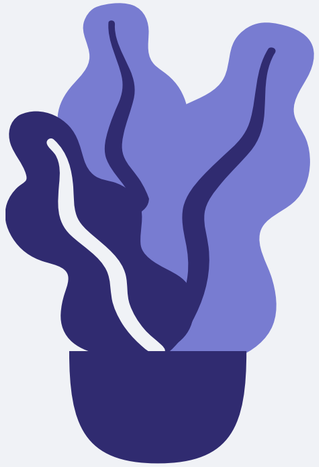
07

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

Solution

```sql
SELECT
    pizza_types.category,
    SUM(orders_details.quantity) AS quantity_sold
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
ORDER BY quantity_sold DESC;
```

	category	quantity_sold
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050



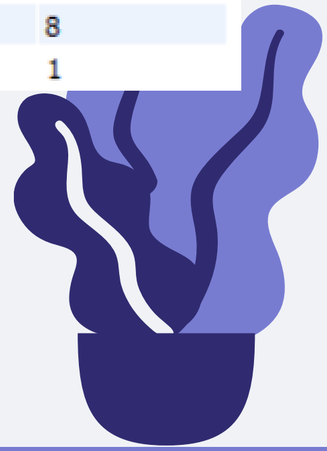
08

Determine the distribution of orders by hour of the day.

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

Solution

	HOUR(order_time)	order_count
▶	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



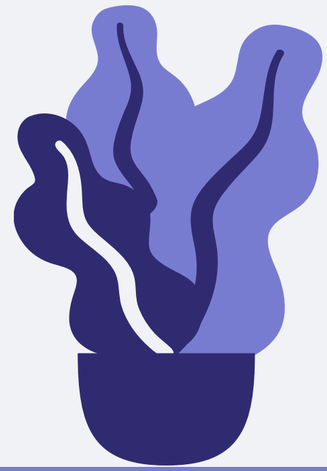
09

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

```
SELECT
    category, COUNT(name)
FROM
    pizza_types
GROUP BY category;
```

Solution

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



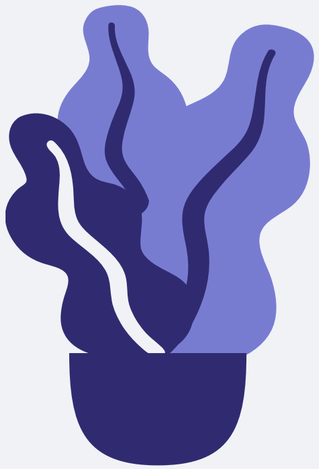
10

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

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

Solution

	ROUND(AVG(quantity), 0)
▶	138



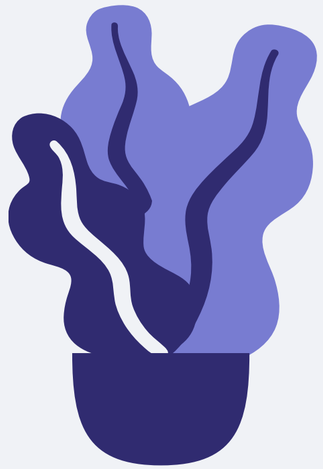
11

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

Solution

```sql
SELECT
    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.name
ORDER BY revenue DESC
LIMIT 3
```

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



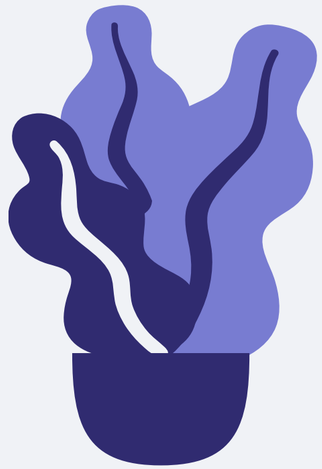
12

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

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

Solution

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68



13

Analyze the cumulative revenue generated over time.

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

Solution

order_date	cuml_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
2015-01-11	25862.65
2015-01-12	27781.7

14

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

```sql
select category 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((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;
```

Solution

name	revenue
Chicken	43434.25
Chicken	42768
Chicken	41409.5
Classic	38180.5
Classic	32273.25
Classic	30161.75
Supreme	34831.25
Supreme	33476.75
Supreme	30940.5
Veggie	32265.70000000065
Veggie	26780.75
Veggie	26066.5

Thank you

Sakshi rana
fresher

Contact Us

sakshirana35283@gmail.com

https://www.linkedin.com/in/sakshi-rana-6893b7219/t