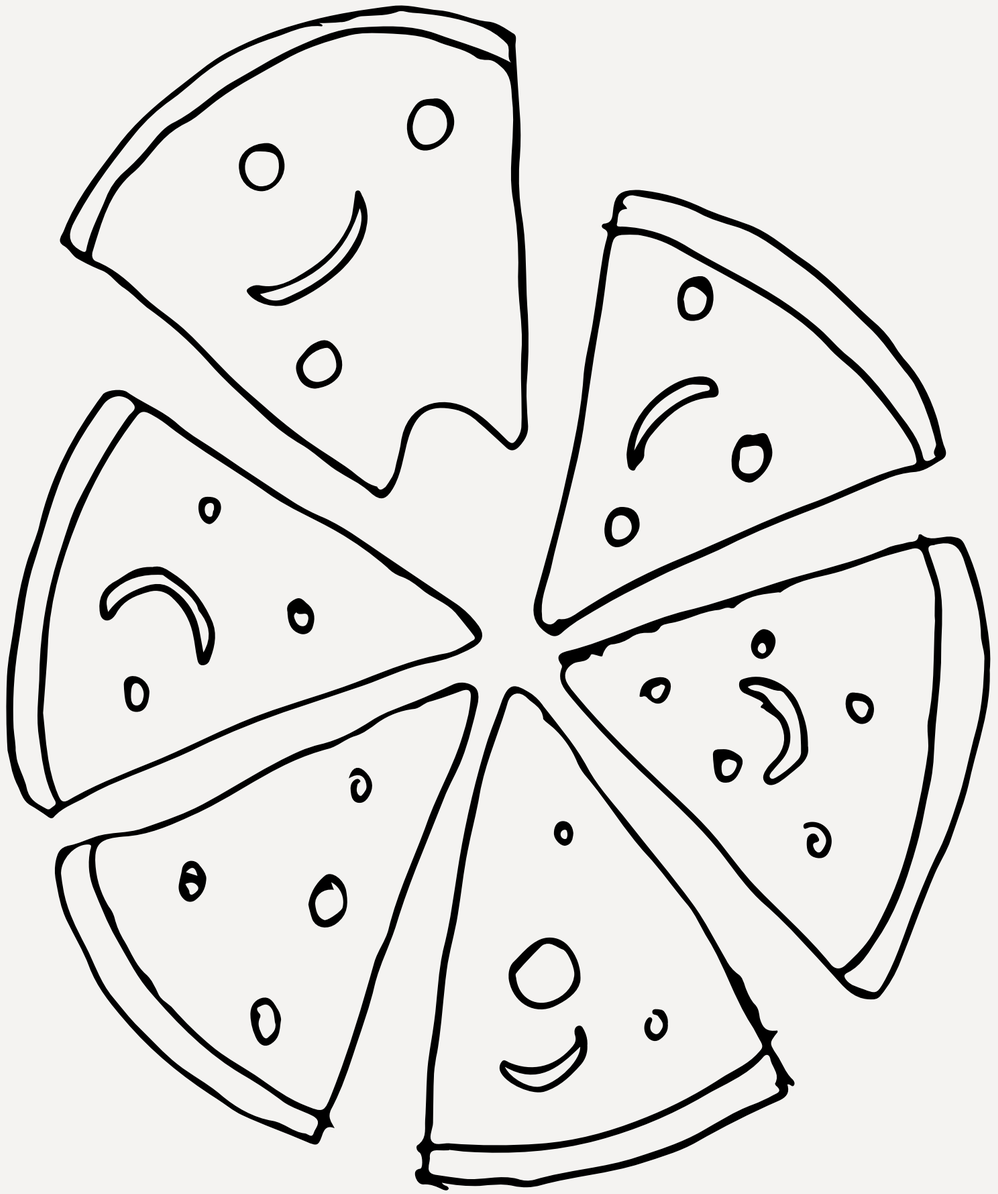
SQL Project

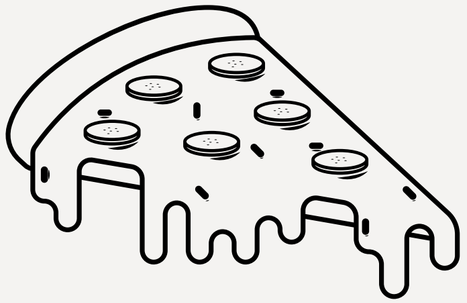
Pizza Republic

sales report



CREATION OF DATABASE

```sql
CREATE DATABASE pizza_republic ;
SELECT
    *
FROM
    pizzas;
SELECT
    *
FROM
    pizza_types;
```



```sql
CREATE TABLE orders (
    order_id INT NOT NULL,
    `date` DATE NOT NULL,
    `time` TIME NOT NULL,
    PRIMARY KEY (order_id)
);
```

```sql
CREATE TABLE order_details (
    order_details_id INT NOT NULL,
    order_id INT NOT NULL,
    pizza_id TEXT NOT NULL,
    quantity INT NOT NULL,
    PRIMARY KEY (order_details_id)
);
```

-- BASIC: QUERY 1

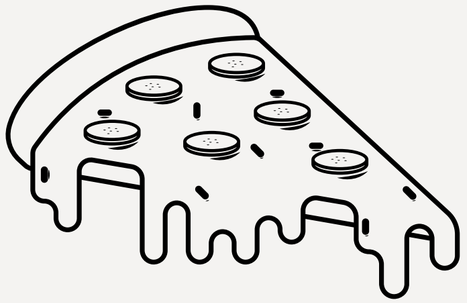
Retrieve the total number of orders placed.

```sql
SELECT * FROM orders ;
```

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

Result Grid

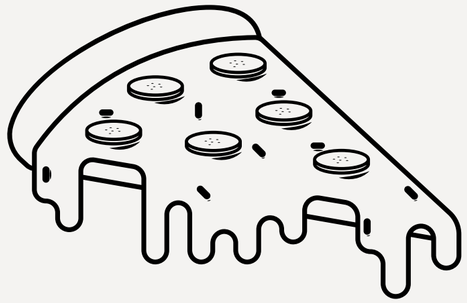
	total_orders
▶	21350



QUERY 2

Calculate the total revenue generated from pizza sales.

```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
            0) AS total_revenue
FROM
    order_details
        JOIN
    pizzas ON pizzas.pizza_id = order_details.pizza_id;
```



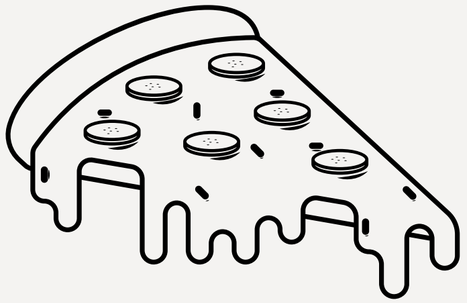
Result Grid

	total_revenue
▶	817860

QUERY 3

Identify the highest-priced pizza.

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



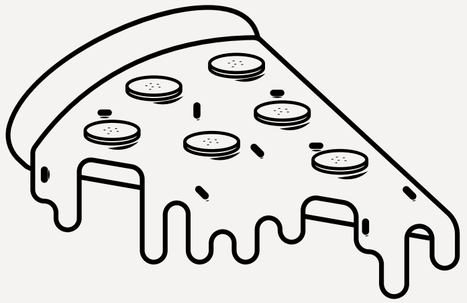
Result Grid | Filter Rows:

name	highest_priced
The Greek Pizza	35.95

QUERY 4

Identify the most common pizza size ordered.

```sql
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
LIMIT 1;
```



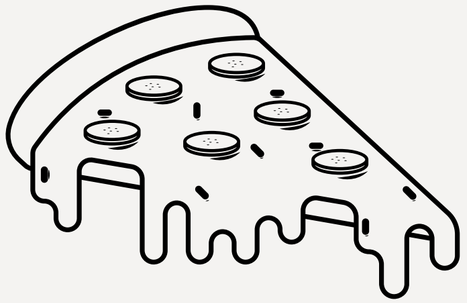
Result Grid | Filter

size	order_count
L	18526

QUERY 5

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

```sql
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
LIMIT 5;
```



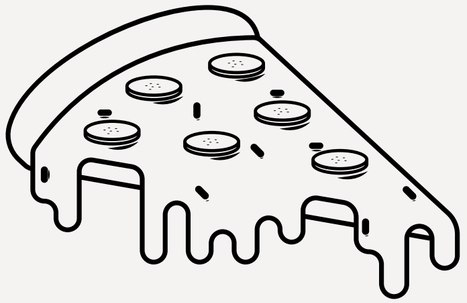
Result Grid | Filter Rows:

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

--INTERMEDIATE: QUERY 6

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

```
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;
```



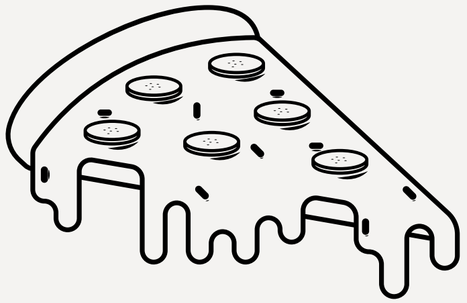
Result Grid | Filter

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

QUERY 7

Determine the distribution of orders by hour of the day

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



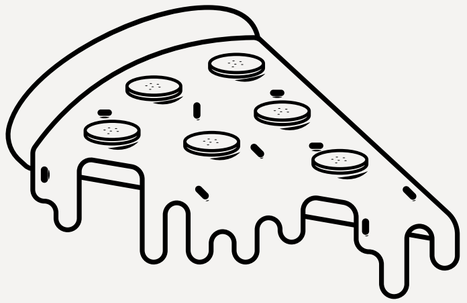
Result Grid | Filt

hour	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

QUERY 8

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

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



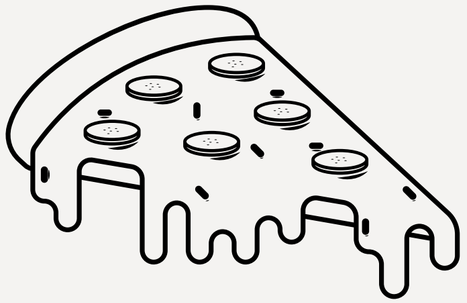
Result Grid

	category	count
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9

QUERY 9

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

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



Result Grid | Filter Rows:

avg_pizza_ordered_per_day
138

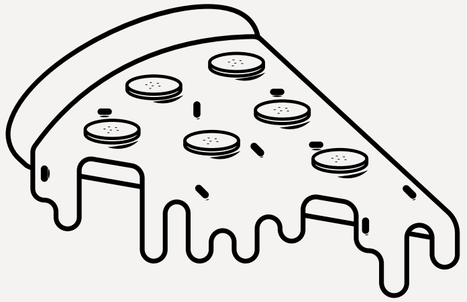
QUERY 10

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

```sql
SELECT
    pizza_types.name,
    ROUND(SUM(order_details.quantity * pizzas.price),
            0) 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;
```

Result Grid | Filter Rows:

name	revenue
The Thai Chicken Pizza	43434
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41410



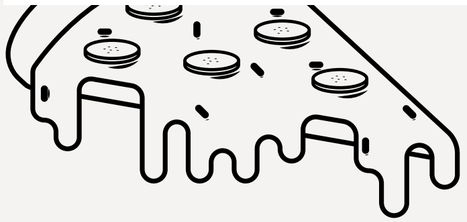
-- ADVANCED: QUERY 11

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

```sql
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price) / (SELECT
                    ROUND(SUM(order_details.quantity * pizzas.price),
                                0) AS total_revenue
                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;
```

Result Grid | Filte

category	revenue
Classic	26.91
Supreme	25.46
Chicken	23.96
Veggie	23.68



QUERY 12

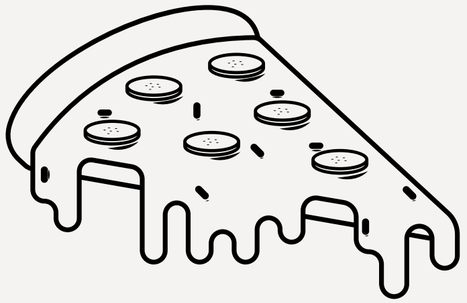
Analyze the cumulative revenue generated over time.

```
SELECT `date`,
round(sum(revenue) OVER(ORDER BY `date`),1) AS cum_revenue
FROM
(SELECT orders.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.date) AS sales;
```

Result Grid | Filter Rows:

date	cum_revenue
2015-01-01	2713.9
2015-01-02	5445.8
2015-01-03	8108.2
2015-01-04	9863.6
2015-01-05	11929.6
2015-01-06	14358.5
2015-01-07	16560.7
2015-01-08	19399
2015-01-09	21526.4
2015-01-10	23990.4
2015-01-11	25862.6
2015-01-12	27781.7
2015-01-13	29831.3
2015-01-14	32358.7
2015-01-15	34343.5
2015-01-16	36937.7

Result 54



QUERY 13

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 `rank`
FROM
(SELECT
        category,
        name,
        revenue,
    RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS `rank`
FROM
(SELECT
    pizza_types.category,
    pizza_types.name,
            ROUND(SUM(order_details.quantity * pizzas.price),0) 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 `rank` <= 3
GROUP BY category, name, revenue;
```

Result Grid | Filter Rows: | Export:

category	name	revenue	rank
Chicken	The Thai Chicken Pizza	43434	1
Chicken	The Barbecue Chicken Pizza	42768	2
Chicken	The California Chicken Pizza	41410	3
Classic	The Classic Deluxe Pizza	38180	1
Classic	The Hawaiian Pizza	32273	2
Classic	The Pepperoni Pizza	30162	3
Supreme	The Spicy Italian Pizza	34831	1
Supreme	The Italian Supreme Pizza	33477	2
Supreme	The Sicilian Pizza	30940	3
Veggie	The Four Cheese Pizza	32266	1
Veggie	The Mexicana Pizza	26781	2
Veggie	The Five Cheese Pizza	26066	3