

PIZZA SALES

SQL

ANALYSIS

```
-- Q1. Retrieve the total number of orders placed.  
SELECT COUNT(order_id) AS 'total_orders' FROM orders;
```

Result Grid	
	total_orders
▶	21350

```
-- Q2. Calculate the total revenue generated from pizza sales.  
SELECT ROUND(SUM(b.quantity*a.price), 2) AS 'total_revenue'  
FROM pizzas a  
JOIN order_details b  
ON a.pizza_id = b.pizza_id;
```

Result Grid	
	total_revenue
▶	817860.05



```
-- Q3. Identify the highest-priced pizza.  
SELECT b.name, a.price  
FROM pizzas a  
JOIN pizza_types b ON a.pizza_type_id = b.pizza_type_id  
WHERE price = (SELECT MAX(price) FROM pizzas);
```

Result Grid		
	name	price
▶	The Greek Pizza	35.95

```
-- Q4. Identify the most common pizza size ordered.  
SELECT c.size, COUNT(a.order_id)  
FROM orders a  
JOIN order_details b ON a.order_id = b.order_id  
JOIN pizzas c ON b.pizza_id = c.pizza_id  
GROUP BY size  
ORDER BY COUNT(a.order_id) DESC LIMIT 1;
```

Result Grid			Filter Rows:
	size	COUNT(a.order_id)	
▶	L	18526	

```
-- Q5. List the top 5 most ordered pizza types along with their quantities.  
SELECT c.name, SUM(a.quantity)  
FROM order_details a  
JOIN pizzas b ON a.pizza_id = b.pizza_id  
JOIN pizza_types c ON b.pizza_type_id = c.pizza_type_id  
GROUP BY c.name  
ORDER BY SUM(a.quantity) DESC LIMIT 5;
```

Result Grid   Filter Rows: <input type="text"/>		
	name	SUM(a.quantity)
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

```
-- Q1. Join the necessary tables to find the total quantity of each pizza category ordered.
SELECT c.category, SUM(a.quantity)
FROM order_details a
JOIN pizzas b on a.pizza_id = b.pizza_id
JOIN pizza_types c ON b.pizza_type_id = c.pizza_type_id
GROUP BY c.category
ORDER BY SUM(a.quantity) DESC;
```

Result Grid			Filter Rows:
	category	SUM(a.quantity)	
▶	Classic	14888	
	Supreme	11987	
	Veggie	11649	
	Chicken	11050	

```
-- Q2. 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)
ORDER BY COUNT(order_id) DESC;
```

Result Grid			Filter
	Hour	Order Count	
▶	12	2520	
	13	2455	
	18	2399	
	17	2336	
	19	2009	
	16	1920	
	20	1642	
	14	1472	
	15	1468	
	11	1231	
	21	1198	
	22	663	
	23	28	
	10	8	
	9	1	

```
-- Q3. Join relevant tables to find the category-wise distribution of pizzas.  
SELECT category, COUNT(name) AS 'no. of pizzas' FROM pizza_types  
GROUP BY category;
```

Result Grid			Filter Rows
	category	no. of pizzas	
▶	Chicken	6	
	Classic	8	
	Supreme	9	
	Veggie	9	

```
-- Q4. Group the orders by date and calculate the average number of pizzas ordered per day.  
SELECT ROUND(AVG(quantity),0)  
FROM (SELECT a.order_date, SUM(b.quantity) AS 'quantity'  
      FROM orders a  
      JOIN order_details b ON a.order_id = b.order_id  
      GROUP BY a.order_date) AS quantity;
```

Result Grid		Filter Rows
	ROUND(AVG(quantity),0)	
▶	138	

```
-- Q5. Determine the top 3 most ordered pizza types based on revenue.
SELECT d.name, SUM(c.price*b.quantity) AS 'Revenue'
FROM orders a
JOIN order_details b ON a.order_id = b.order_id
JOIN pizzas c ON b.pizza_id = c.pizza_id
JOIN pizza_types d ON c.pizza_type_id = d.pizza_type_id
GROUP BY d.name
ORDER BY Revenue DESC LIMIT 3;
```

Result Grid			Filter Rows:
	name	Revenue	
▶	The Thai Chicken Pizza	43434.25	
	The Barbecue Chicken Pizza	42768	
	The California Chicken Pizza	41409.5	


```
-- Q1. Calculate the percentage contribution of each pizza type to total revenue.
SELECT c.category, ROUND((SUM(a.quantity* b.price) / (SELECT SUM(a.quantity* b.price)
FROM order_details a
JOIN pizzas b ON a.pizza_id = b.pizza_id)) * 100, 2) AS '% of each category'
FROM order_details a
JOIN pizzas b ON a.pizza_id = b.pizza_id
JOIN pizza_types c ON b.pizza_type_id = c.pizza_type_id
GROUP BY c.category
ORDER BY '% of each category' DESC;
```

Result Grid			Filter Rows:
	category	% of each category	
▶	Classic	26.91	
	Veggie	23.68	
	Supreme	25.46	
	Chicken	23.96	

-- Q2. Analyze the cumulative revenue generated over time.

```
SELECT order_date, SUM(revenue) OVER(ORDER BY order_date) AS 'Cumulative Sum'
FROM(
  SELECT a.order_date, ROUND(SUM(b.quantity*c.price),2) AS 'revenue'
  FROM orders a
  JOIN order_details b ON a.order_id = b.order_id
  JOIN pizzas c ON b.pizza_id = c.pizza_id
  GROUP BY a.order_date) AS t;
```

Result Grid			Filter Rows:
	order_date	Cumulative Sum	
▶	2015-01-01	2713.85	
	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	21526.399999999998	
	2015-01-10	23990.35	
	2015-01-11	25862.649999999998	
	2015-01-12	27781.699999999997	
	2015-01-13	29831.299999999996	
	2015-01-14	32358.699999999997	
	2015-01-15	34343.5	
	2015-01-16	36937.65	

```
-- Q3. Determine the top 3 most ordered pizza types based on revenue for each pizza category.
SELECT category, name, revenue
FROM
(SELECT category, name, revenue,
RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS 'rn'
FROM
(SELECT c.name, c.category, SUM(a.quantity*b.price) AS 'revenue'
FROM order_details a
JOIN pizzas b ON a.pizza_id = b.pizza_id
JOIN pizza_types c ON b.pizza_type_id = c.pizza_type_id
GROUP BY c.category, c.name) AS t1) AS t2
WHERE rn<=3;
```

Result Grid				Filter Rows:	Export:
	category	name	revenue		
►	Chicken	The Thai Chicken Pizza	43434.25		
	Chicken	The Barbecue Chicken Pizza	42768		
	Chicken	The California Chicken Pizza	41409.5		
	Classic	The Classic Deluxe Pizza	38180.5		
	Classic	The Hawaiian Pizza	32273.25		
	Classic	The Pepperoni Pizza	30161.75		
	Supreme	The Spicy Italian Pizza	34831.25		
	Supreme	The Italian Supreme Pizza	33476.75		
	Supreme	The Sicilian Pizza	30940.5		
	Veggie	The Four Cheese Pizza	32265.70000000065		
	Veggie	The Mexicana Pizza	26780.75		
	Veggie	The Five Cheese Pizza	26066.5		

THANK
YOU

