

The background features a light beige grid pattern. Various food items are illustrated around the edges: a bowl of ramen with chopsticks in the top left, two halves of a soft-boiled egg in the top right, a slice of orange with a star-shaped garnish on the left, two dumplings on the right, two breadsticks with tomato and onion toppings in the bottom left, and a bowl of dark rice with green onions and yellow noodles in the bottom right.

PIZZA SALES ANALYSIS

USING SQL

QUESTION

```
1  -- Retrieve the total number of orders placed.  
2  
3  ● SELECT count(order_id) as Total_orders  
4  FROM pizzahut.orders;
```

ANSWER

Result Grid			
	Total_orders		
▶	21350		

QUESTION

-- Calculate the total revenue generated from pizza sales.

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

ANSWER

Result Grid

	total_revenue
▶	592355.9

QUESTION

-- Identify the highest-priced pizza.

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

ANSWER

Result Grid			Filter
	name	price	
*	The Greek Pizza	35.95	

QUESTION

-- Identify the most common pizza size ordered.

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;

ANSWER

size	order_count
L	13415
M	11157
S	10225
XL	406
XXL	19

QUESTION

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

```
select pizza_types.name, sum(order_details.quantity) as most_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 most_quantity desc
limit 5;
```

ANSWER

name	most_quantity
The Barbecue Chicken Pizza	1790
The Pepperoni Pizza	1754
The Classic Deluxe Pizza	1753
The Hawaiian Pizza	1719
The California Chicken Pizza	1712

QUESTION

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

SELECT

SUM(order_details.quantity) AS total_quantity,
pizza_types.category

FROM

order_details

JOIN

pizzas ON pizzas.pizza_id = order_details.pizza_id

JOIN

pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id

GROUP BY pizza_types.category

ORDER BY total_quantity DESC;

ANSWER

total_quantity	category
10750	Classic
8662	Supreme
8514	Veggie
7971	Chicken

QUESTION

-- Determine the distribution of orders by hour of the day.

SELECT

HOUR(order_time) AS dayhour, **COUNT**(order_id) AS orders

FROM

orders

GROUP BY dayhour

ORDER BY dayhour;

ANSWER

dayhour	orders
9	1
10	8
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

QUESTION

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

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

ANSWER

category	pizza_category
Chicken	6
Classic	8
Supreme	9
Veggie	9

QUESTION

```
-- Group the orders by date and
-- calculate the average number of pizzas ordered per day.
SELECT
    AVG(quantity)
FROM
    (SELECT
        orders.order_date AS date,
        COUNT(order_details.quantity),
        SUM(order_details.quantity) AS quantity
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY date) as orders
```

ANSWER

AVG(quantity)

138.0654

QUESTION

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

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

ANSWER


name	revenue
The Barbecue Chicken Pizza	31430.5
The Thai Chicken Pizza	30862
The California Chicken Pizza	29900

QUESTION

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

```
SELECT
    pizza_types.category,
    (SUM(pizzas.price * order_details.quantity) / (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)) * 100 AS percentage_contribution
FROM
    pizzas
    JOIN
        order_details ON pizzas.pizza_id = order_details.pizza_id
    JOIN
        pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category
ORDER BY Percentage_contribution;
```

ANSWER



category	percentage_contribution
Chicken	23.824570667735394
Veggie	23.896275870637744
Supreme	25.436540768818027
Classic	26.842612692808494



QUESTION

-- Analyze the cumulative revenue generated over time.

```
select date,  
sum(revenue) over(order by date) as cum_revenue
```

```
from
```

```
(select orders.order_date as date, round(sum(pizzas.price*order_details.quantity),2) as revenue  
from pizzas join order_details  
on pizzas.pizza_id = order_details.pizza_id  
join orders  
on order_details.order_id = orders.order_id  
group by date  
order by date) as sales;
```


QUESTION

-- 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 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;
```