

# SQL PROJECT ON PIZZA SALES ANALYSIS



NEXT >

# TABLE SCHEMA



## Table: order\_details

### Columns:

<u>order_details_id</u>	int
order_id	int
pizza_id	text
quantity	int

## Table: orders

### Columns:

<u>order_id</u>	int PK
order_date	date
order_time	time

## Table: pizza\_types

### Columns:

pizza_type_id	text
name	text
category	text
ingredients	text

## Table: pizzas

### Columns:

pizza_id	text
pizza_type_id	text
size	text
price	double

< BACK

NEXT >

# RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.



-- Retrieve the total number of orders placed.

**SELECT**

COUNT(order\_id) **AS** total\_order

**FROM**

orders;

Result Grid	
	total_order
▶	21350

< BACK

NEXT >

# CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.



```
1  -- Calculate the total revenue generated from pizza sales.  
2  
3 • SELECT  
4   ROUND(SUM(order_details.quantity * pizzas.price),  
5         2) AS total_sales  
6   FROM  
7     order_details  
8   JOIN  
9     pizzas ON pizzas.pizza_id = order_details.pizza_id
```

< [Result Grid](#) | [Filter Rows:](#)  | Export: [CSV](#) | Wrap Cell Content: [OFF](#)

total_sales
817860.05

< BACK

NEXT >



# IDENTIFY THE HIGHEST-PRICED PIZZA.

```
1  -- Identify the highest-priced pizza.  
2  
3 • SELECT  
4      pizza_types.name, pizzas.price  
5  FROM  
6      pizza_types  
7      JOIN  
8      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
9  ORDER BY pizzas.price DESC  
10 LIMIT 1;
```

< | Result Grid | | Filter Rows:  | Export: | Wrap Cell Content: | Fetch rows:

	name	price
▶	The Greek Pizza	35.95

< BACK

NEXT >

# IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.



```
1  -- Identify the most common pizza size ordered.  
2  
3 • SELECT  
4      pizzas.size,  
5      COUNT(order_details.order_details_id) AS order_count  
6  FROM  
7      pizzas  
8      JOIN  
9      order_details ON pizzas.pizza_id = order_details.pizza_id  
10 GROUP BY pizzas.size  
11 ORDER BY order_count DESC;
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

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

< BACK

NEXT >

# LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.



```
1  -- List the top 5 most ordered pizza types along with their quantities.  
2  
3 • SELECT  
4      pizza_types.name, SUM(order_details.quantity) AS quantity  
5  FROM  
6      pizza_types  
7      JOIN  
8      pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
9      JOIN  
10     order_details ON order_details.pizza_id = pizzas.pizza_id  
11    GROUP BY pizza_types.name  
12    ORDER BY quantity DESC  
13    LIMIT 5;
```

< [ ] >

Result Grid | Filter Rows:  Export: Wrap Cell Content: Fetch 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

< BACK

NEXT >

# JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.



```
1 -- Join the necessary tables to find the total quantity of each pizza category ordered.  
2  
3 • SELECT  
4     pizza_types.category,  
5         SUM(order_details.quantity) AS quantity  
6 FROM  
7     pizza_types  
8         JOIN  
9     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id  
10        JOIN  
11    order_details ON order_details.pizza_id = pizzas.pizza_id  
12 GROUP BY pizza_types.category  
13 ORDER BY quantity DESC;  
14
```

<

Result Grid | Filter Rows:  Export: Wrap Cell Content:

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

< BACK

NEXT >

# DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.



```
1      -- Determine the distribution of orders by hour of the day.  
2  
3 •  SELECT  
4      HOUR(order_time) AS hour, COUNT(order_id) AS order_count  
5  FROM  
6      orders  
7  GROUP BY HOUR(order_time);
```

Result Grid | Filter

	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

< BACK

NEXT >

# JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.



```
1   -- Join relevant tables to find the category-wise distribution of pizzas.  
2  
3 • SELECT  
4     category, COUNT(name)  
5   FROM  
6     pizza_types  
7   GROUP BY category;
```

result Grid | Filter Rows:  Export: Wrap Cell Content:

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

< BACK

NEXT >

# GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```
1      -- Group the orders by date and calculate the average number of pizzas ordered per day.  
2  
3 •  SELECT  
4      ROUND(AVG(quantity), 0)  
5  FROM  
6    (SELECT  
7      orders.order_date, SUM(order_details.quantity) AS quantity  
8  FROM  
9      orders  
10     JOIN order_details ON orders.order_id = order_details.order_id  
11     GROUP BY orders.order_date) AS order_quantity;
```

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

< BACK

NEXT >

# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.



```
1  -- Determine the top 3 most ordered pizza types based on revenue.  
2  
3 • SELECT  
4      pizza_types.name,  
5      SUM(order_details.quantity * pizzas.price) AS revenue  
6  FROM  
7      pizza_types  
8      JOIN  
9      pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
10     JOIN  
11     order_details ON order_details.pizza_id = pizzas.pizza_id  
12   group by pizza_types.name order by revenue desc limit 3;
```

Result Grid | Filter Rows:  Export: Wrap Cell Content: Fetch rows:

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

< BACK

NEXT >

# CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.



```
1  -- Calculate the percentage contribution of each pizza type to total revenue.  
2  
3 • SELECT  
4      pizza_types.category,  
5      (SUM(order_details.quantity * pizzas.price) / (SELECT  
6          ROUND(SUM(order_details.quantity * pizzas.price),  
7          2) AS total_sales  
8      FROM  
9          order_details  
10         JOIN  
11             pizzas ON pizzas.pizza_id = order_details.pizza_id)) * 100 AS revenue  
12  FROM  
13      pizza_types  
14         JOIN  
15             pizzas ON pizzas.pizza_type_id = pizza_types.pizza_type_id  
16         JOIN  
17             order_details ON order_details.pizza_id = pizzas.pizza_id  
18  GROUP BY pizza_types.category  
19  ORDER BY revenue DESC;
```

Result Grid | Filter Rows:

	category	revenue
▶	Classic	26.90596025566967
	Supreme	25.45631126009862
	Chicken	23.955137556847287
	Veggie	23.682590927384577

< BACK

NEXT >

# ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.



```
1 -- Analyze the cumulative revenue generated over time.  
2  
3 • select order_date,  
4     sum(revenue) over (order by order_date)as cum_revenue  
5   from  
6     (select orders.order_date, sum(order_details.quantity* pizzas.price) as revenue  
7       from order_details join pizzas  
8         on order_details.pizza_id= pizzas.pizza_id  
9       join orders  
10      on orders.order_id= order_details.order_id  
11     group by orders.order_date )as sales;
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

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	23990.35000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.30000000003
2015-01-14	32358.70000000004

Result Grid | Form Editor | Field Types | Query Stats | General

< BACK

NEXT >

# DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```
1  -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.  
2  
3 • select name, revenue from  
4  (select category, name, revenue,  
5   rank() over(partition by category order by revenue desc)as rn  
6  from (select pizza_types.category, pizza_types.name,  
7    sum((order_details.quantity)* pizzas.price)as revenue  
8    from pizza_types join pizzas  
9    on pizza_types.pizza_type_id=pizzas.pizza_type_id  
10   join order_details  
11   on order_details.pizza_id = pizzas.pizza_id  
12   group by pizza_types.category,pizza_types.name)as a) as b
```

Result Grid | Filter Rows:  Export: Wrap Cell Content:

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
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.70000000065
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5

< BACK

NEXT >