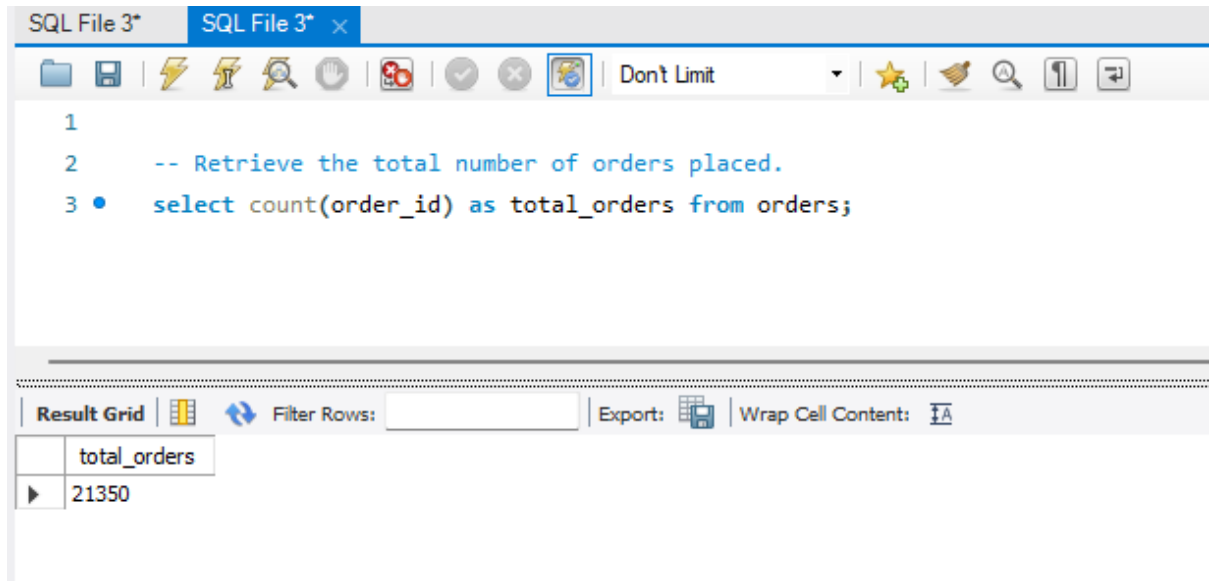


PIZZA HUT OVERALL SALES ANALYSIS

Question 1



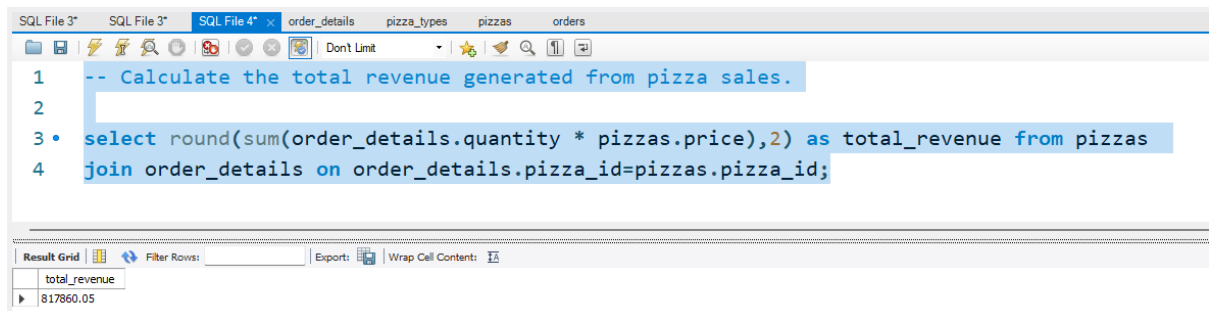
The screenshot shows a SQL IDE window with two tabs: 'SQL File 3*' and 'SQL File 3* x'. The active tab contains the following SQL code:

```
1
2  -- Retrieve the total number of orders placed.
3  • select count(order_id) as total_orders from orders;
```

Below the code editor, the 'Result Grid' is displayed with the following data:

total_orders
21350

Question 2



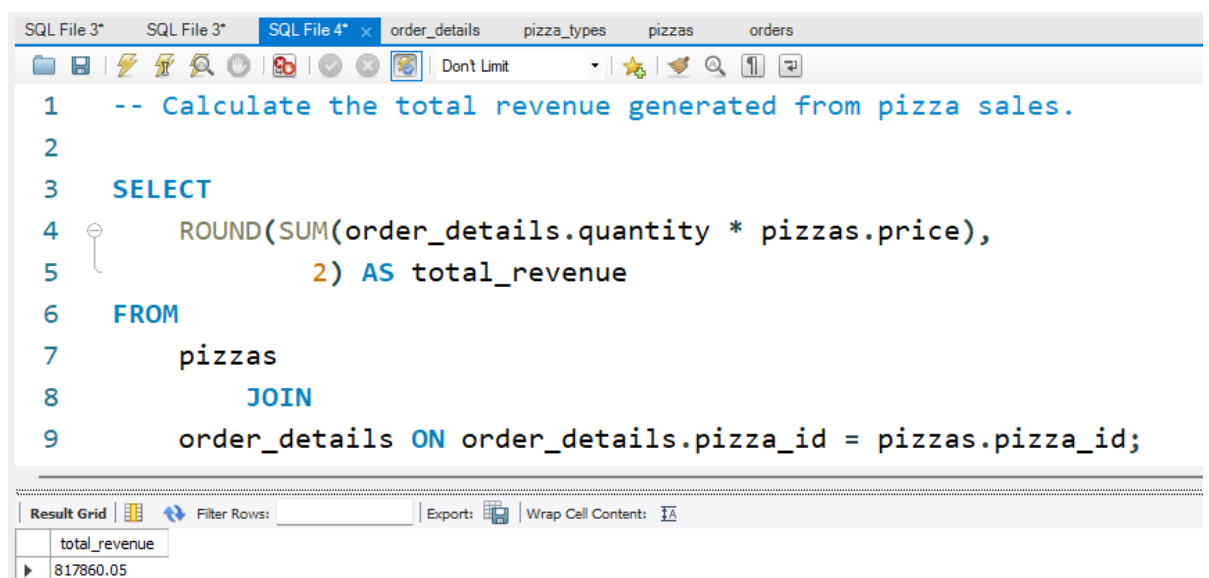
The screenshot shows a SQL IDE window with multiple tabs: 'SQL File 3*', 'SQL File 3*', 'SQL File 4*', 'order_details', 'pizza_types', 'pizzas', and 'orders'. The active tab 'SQL File 4*' contains the following SQL code:

```
1  -- Calculate the total revenue generated from pizza sales.
2
3  • select round(sum(order_details.quantity * pizzas.price),2) as total_revenue from pizzas
4  join order_details on order_details.pizza_id=pizzas.pizza_id;
```

Below the code editor, the 'Result Grid' is displayed with the following data:

total_revenue
817860.05

Question 3



The screenshot shows a SQL IDE window with multiple tabs: 'SQL File 3*', 'SQL File 3*', 'SQL File 4*', 'order_details', 'pizza_types', 'pizzas', and 'orders'. The active tab 'SQL File 4*' contains the following SQL code:

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

Below the code editor, the 'Result Grid' is displayed with the following data:

total_revenue
817860.05

PIZZA HUT OVERALL SALES ANALYSIS

Question 4

The screenshot shows a SQL IDE interface with a query editor and a result grid. The query editor contains the following SQL code:

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

The result grid shows the following data:

name	price
The Greek Pizza	35.95

Question 5

```
1  -- Identify the most common pizza size ordered.
2
3  •  SELECT
4      pizzas.size,
5      COUNT(order_details.order_details_id) AS order_counts
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_counts DESC;
12
```

The screenshot shows a SQL IDE interface with a result grid. The result grid shows the following data:

size	order_counts
L	18526
M	15385
S	14137
XL	544
XXL	28

PIZZA HUT OVERALL SALES ANALYSIS

Question 6

```
1  -- List the top 5 most ordered pizza types along with their quantities.
2
3  •  SELECT
4      pizza_types.name,
5      SUM(order_details.quantity) AS order_quantity
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
13 ORDER BY order_quantity DESC
14 LIMIT 5;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
name	order_quantity		
▶ The Classic Deluxe Pizza	2453		
The Barbecue Chicken Pizza	2432		
The Hawaiian Pizza	2422		
The Pepperoni Pizza	2418		
The Thai Chicken Pizza	2371		

Question 7

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

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
category	total_quantity		
▶ Classic	14888		
Supreme	11987		
Veggie	11649		
Chicken	11050		

PIZZA HUT OVERALL SALES ANALYSIS

Question 8

```
1  -- Group the orders by date and calculate the average number of pizzas ordered per day.
2  SELECT
3      ROUND(AVG(quantity), 0) AS avg_pizza_per_day
4  FROM
5      (SELECT
6          DATE(order_date) AS order_date,
7          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 order_date) AS order_quantity;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
round(avg(quantity),0)			
138			

Question 9

```
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
13  ORDER BY revenue DESC
14  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			

PIZZA HUT OVERALL SALES ANALYSIS

Question 10

```
1  -- Calculate the percentage contribution of each pizza type to total revenue.
2  • SELECT
3      pizza_types.name,
4      SUM(order_details.quantity * pizzas.price) AS revenue
5  FROM
6      pizza_types
7      JOIN
8      pizzas ON pizzas.pizza_type_id = pizza_types.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 revenue DESC;
```

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 Spicy Italian Pizza	34831.25			
	The Southwest Chicken Pizza	34705.75			
	The Italian Supreme Pizza	33476.75			
	The Hawaiian Pizza	32273.25			
	The Four Cheese Pizza	32265.700000000065			
	The Sicilian Pizza	30940.5			
	The Pepperoni Pizza	30161.75			
	The Greek Pizza	28454.100000000013			
	The Mexicana Pizza	26780.75			
	The Five Cheese Pizza	26066.5			
	The Pepper Salami Pizza	25529			
	The Italian Capocollo Pizza	25094			
	The Vegetables + Vegetable...	24374.75			
	The Prosciutto and Arugula ...	24193.25			
	The Napolitana Pizza	24087			

PIZZA HUT OVERALL SALES ANALYSIS

Question 11

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

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

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

Question 12

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

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

	category	revenue
▶	Classic	220053.1000000001
	Supreme	208196.99999999822
	Chicken	195919.5
	Veggie	193690.45000000298