

PIZZA SALES PROJECT SQL

Sales analysis





RAW DATA IN EXCEL FORMATE

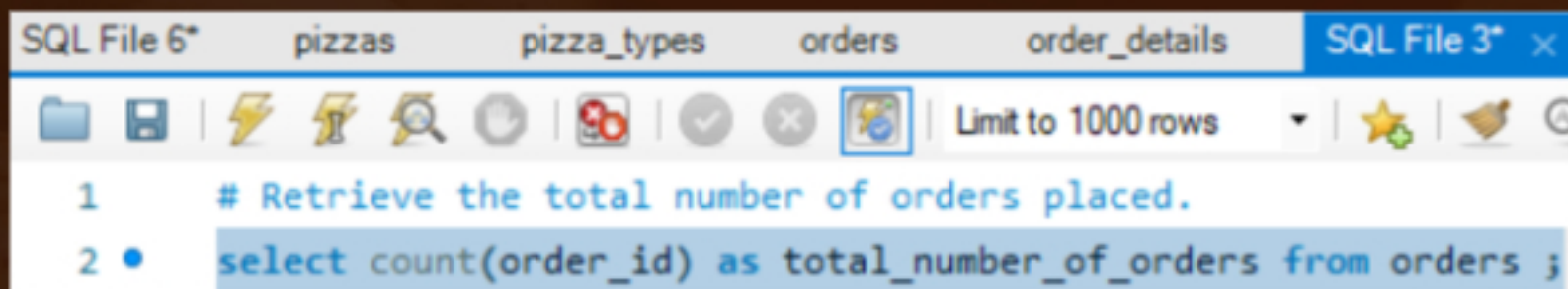
pizza_id	pizza_type	size	price
bbq_ckn	bbq_ckn	S	12.75
bbq_ckn	bbq_ckn	M	16.75
bbq_ckn	bbq_ckn	L	20.75
cali_ckn_s	cali_ckn	S	12.75
cali_ckn_m	cali_ckn	M	16.75
cali_ckn_l	cali_ckn	L	20.75
ckn_alfredo	ckn_alfredo	S	12.75
ckn_alfredo	ckn_alfredo	M	16.75
ckn_alfredo	ckn_alfredo	L	20.75

pizza_type_id	name	category
bbq_ckn	The Barbecue Chicken Pizza	Chicken
cali_ckn	The California Chicken Pizza	Chicken
ckn_alfredo	The Chicken Alfredo Pizza	Chicken
ckn_pesto	The Chicken Pesto Pizza	Chicken
southw_ckn	The Southwest Chicken Pizza	Chicken
thai_ckn	The Thai Chicken Pizza	Chicken
big_meat	The Big Meat Pizza	Classic
classic_dlx	The Classic Deluxe Pizza	Classic
hawaiian	The Hawaiian Pizza	Classic
ital_cpello	The Italian Capocollo Pizza	Classic
napolitana	The Napolitana Pizza	Classic

order_details_id	order_id	pizza_id	quantity
1	1	hawaiian_m	1
2	2	classic_dlx_m	1
3	2	five_cheese_l	1
4	2	ital_supr_l	1
5	2	mexicana_m	1
6	2	thai_ckn_l	1
7	3	ital_supr_m	1
8	3	prsc_argla_l	1
9	4	ital_supr_m	1
10	5	ital_supr_m	1
11	6	bbq_ckn_s	1
12	6	the_greek_s	1

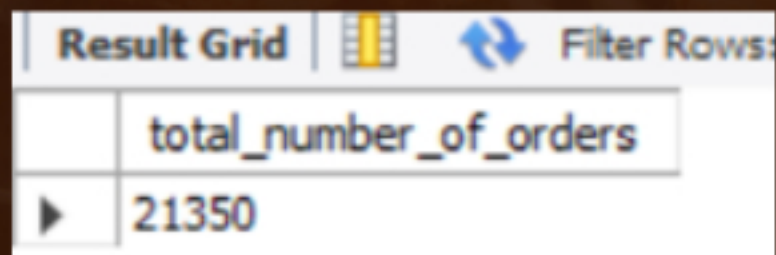
order_id	date	time
1	01-01-2015	11:38:36
2	01-01-2015	11:57:40
3	01-01-2015	12:12:28
4	01-01-2015	12:16:31
5	01-01-2015	12:21:30
6	01-01-2015	12:29:36
7	01-01-2015	12:50:37
8	01-01-2015	12:51:37
9	01-01-2015	12:52:01
10	01-01-2015	13:00:15
11	01-01-2015	13:02:59
12	01-01-2015	13:04:41

Q-1.. Retrieve the total number of orders placed.



The screenshot shows a SQL IDE interface with several tabs: 'SQL File 6*', 'pizzas', 'pizza_types', 'orders', 'order_details', and 'SQL File 3*'. The 'orders' tab is active. Below the tabs is a toolbar with various icons, including a 'Limit to 1000 rows' dropdown. The SQL editor contains the following code:

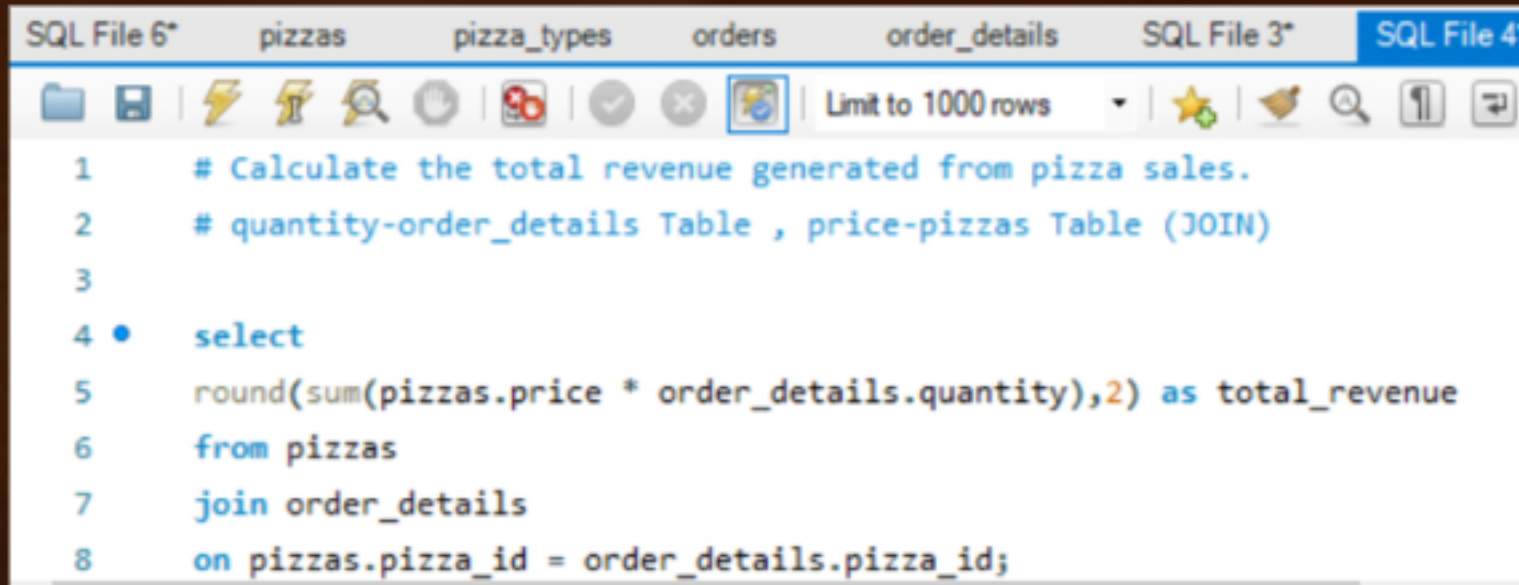
```
1 # Retrieve the total number of orders placed.  
2 • select count(order_id) as total_number_of_orders from orders ;
```



The screenshot shows the 'Result Grid' of the SQL IDE. It has a 'Filter Rows' button. The grid contains one row with the column name 'total_number_of_orders' and the value '21350'.

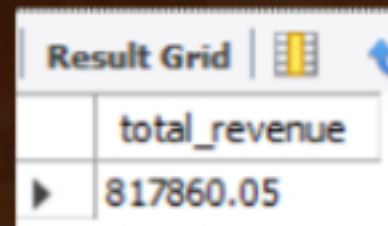
	total_number_of_orders
▶	21350

Q-2.. Calculate the total revenue generated from pizza sales.



The screenshot shows a SQL IDE with several tabs: 'SQL File 6*', 'pizzas', 'pizza_types', 'orders', 'order_details', 'SQL File 3*', and 'SQL File 4*'. The 'SQL File 4*' tab is active, displaying the following SQL query:

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



The screenshot shows the 'Result Grid' tab in the SQL IDE. It displays a single row of data representing the total revenue.

	total_revenue
▶	817860.05

Q-3.. Identify the highest-priced pizza.



```
SQL File 6*  pizzas  pizza_types  orders  order_details  S
Limit to 1000 rows
1  # Identify the highest-priced pizza.
2  •  select pizza_types.name , pizzas.price
3     from pizza_types
4     join pizzas
5     on pizza_types.pizza_type_id = pizzas.pizza_type_id
6     order by pizzas.price desc
7     limit 1;
```

Result Grid			Filter Rows
	name	price	
▶	The Greek Pizza	35.95	

Q-4.. Identify the most common pizza size ordered.

```
SQL File 6*  pizzas  pizza_types  orders  order_details  SQL File
[Icons] | Limit to 1000 rows
1  # Identify the most common pizza size ordered.
2  • select quantity, count(order_details_id)
3    from order_details
4    group by quantity;
5
6  • select pizzas.size, count(order_details.order_details_id)
7    from pizzas join order_details
8    on pizzas.pizza_id=order_details.pizza_id
9    group by pizzas.size
10   order by count(order_details.order_details_id) desc ;
```

Result Grid			Filter Rows:
	size	count(order_details.order_details_id)	
▶	L	18526	
	M	15385	
	S	14137	
	XL	544	
	XXL	28	

Q-5.. 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  # join 3 Table - pizzas,pizza_types,order_details
3  • select
4  pizza_types.name, sum(order_details.quantity)
5  from pizzas
6  join pizza_types
7  on pizzas.pizza_type_id=pizza_types.pizza_type_id
8  join order_details
9  on order_details.pizza_id=pizzas.pizza_id
10 group by pizza_types.name
11 order by sum(order_details.quantity) desc
12 limit 5;
```

Result Grid			Filter Rows:	Exp
	name	sum(order_details.quantity)		
▶	The Classic Deluxe Pizza	2453		
	The Barbecue Chicken Pizza	2432		
	The Hawaiian Pizza	2422		
	The Pepperoni Pizza	2418		
	The Thai Chicken Pizza	2371		

Q-6.. 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  •  select
3      pizza_types.category, sum(order_details.quantity)
4  from pizzas
5  join pizza_types
6  on pizzas.pizza_type_id=pizza_types.pizza_type_id
7  join order_details
8  on order_details.pizza_id=pizzas.pizza_id
9  group by pizza_types.category;
```

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

Q-7.. Determine the distribution of orders by hour of the day.



SQL File 6* pizzas pizza_types orders order_details SQL File 1

Limit to 1000 rows

```
1 # Determine the distribution of orders by hour of the day.
2 • select hour(time),count(order_id) from orders
3 group by hour(time);
```

Result Grid			Filter Rows:
	hour(time)	count(order_id)	
▶	11	1231	
	12	2520	
	13	2455	
	14	1472	
	15	1468	
	16	1920	
	17	2336	
	18	2399	



Q-8.. Join relevant tables to find the category-wise distribution of pizzas.



```
6* pizzas pizza_types orders order_details SQL File 3* SQL File 4* SQL F
Limit to 1000 rows
1 # Join relevant tables to find the category-wise distribution of pizzas.
2 • select category, count(name) from pizza_types
3 group by category;
```

Result Grid		Filter R
	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



Q-9.. Group the orders by date and calculate the average number of pizzas ordered per day.



```
types  orders  order_details  SQL File 3*  SQL File 4*  SQL File 5*  SQL File 6*  SQL File 7*  S
Limit to 1000 rows
1  # Group the orders by date and calculate the average number of pizzas ordered per day
2
3  •  select avg(sum_of_quantity) from
4  (select orders.date, sum(order_details.quantity) as sum_of_quantity
5   from orders join order_details
6   on orders.order_id=order_details.order_id
7   group by orders.date) as ordered_quantity;
```



Result Grid		Filter
	avg(sum_of_quantity)	
▶	138.4749	



Q-10.. Determine the top 3 most ordered pizza types based on revenue.



```
1  # Determine the top 3 most ordered pizza types based on revenue.
2  • select pizza_types.name, sum(order_details.quantity*pizzas.price) as revenue
3  from pizzas join order_details
4  on pizzas.pizza_id=order_details.pizza_id
5  join pizza_types
6  on pizza_types.pizza_type_id=pizzas.pizza_type_id
7  group by pizza_types.name
8  order by revenue desc
9  limit 3;
```

Result Grid				 Filter Rows:	<input type="text"/>
	name	revenue			
▶	The Thai Chicken Pizza	43434.25			
	The Barbecue Chicken Pizza	42768			
	The California Chicken Pizza	41409.5			



Q-11.. Calculate the percentage contribution of each pizza type to total revenue.

```
1  # Calculate the percentage contribution of each pizza type to total revenue.
2  •  select pizza_types.category,
3     round(sum(order_details.quantity*pizzas.price) / (select
4     round(sum(order_details.quantity*pizzas.price),2) as total_sale
5     from order_details join pizzas
6     on pizzas.pizza_id=order_details.pizza_id)*100,2)as revenue
7     from pizza_types join pizzas
8     on pizza_types.pizza_type_id=pizzas.pizza_type_id
9     join order_details
10    on order_details.pizza_id=pizzas.pizza_id
11    group by pizza_types.category;
```

Result Grid			Filter
	category	revenue	
▶	Classic	26.91	
	Veggie	23.68	
	Supreme	25.46	
	Chicken	23.96	

Q-12.. Analyze the cumulative revenue generated over time.

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

Result Grid		Filter Rows:
	date	cum_revenue
▶	2015-01-01	2713.8500000000004
	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

Q-13.. 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  •  select name , revenue from
3  ⊖  (select category,name,revenue,
4     rank() over(partition by category order by revenue desc)as ranks
5     from
6     ⊖  (select pizza_types.category,pizza_types.name, sum(order_details.quantity*pizzas.price) as revenue
7        from pizza_types join pizzas
8        on pizza_types.pizza_type_id=pizzas.pizza_type_id
9        join order_details
10       on order_details.pizza_id=pizzas.pizza_id
11       group by pizza_types.category,pizza_types.name) as a) as b
12  where ranks<=3;
```

Result Grid			Filter Rows:
	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	

