WWW.REALLYGREATSITE.COM

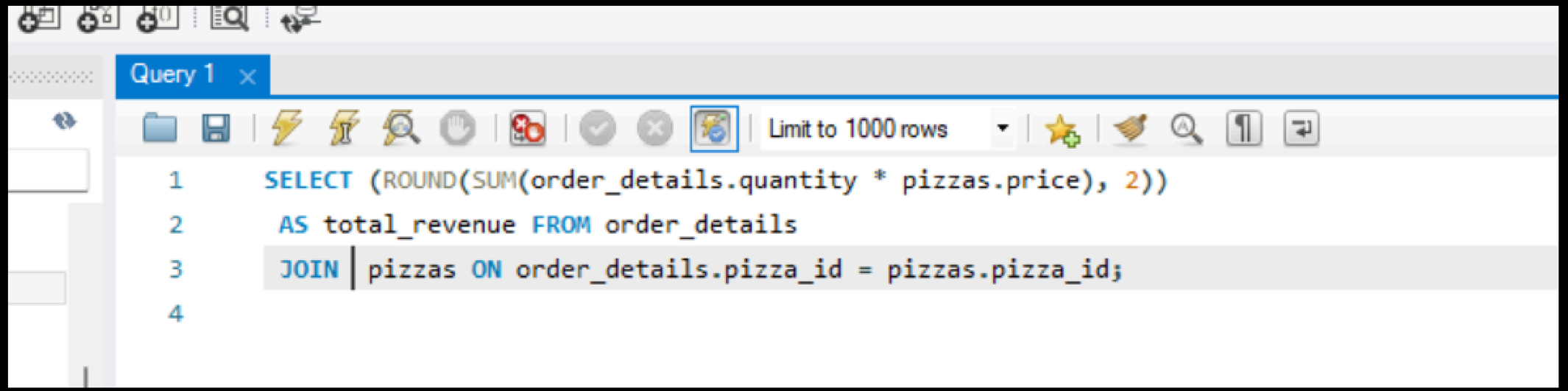
PIZZA HUT

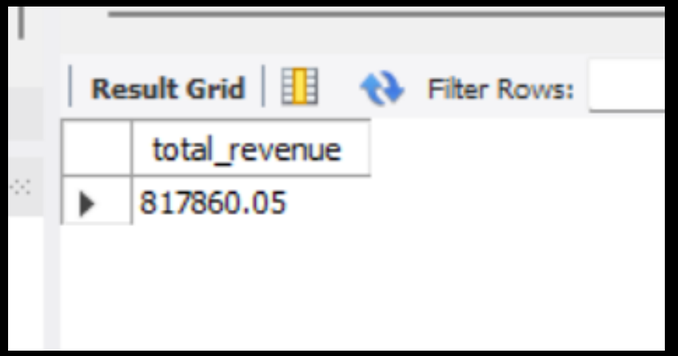
1.RETREIVE THE TOTAL NUMBER OF ORDER PLACED

```
ery 1 ×
Limit to 1000 rows
1 •  select count(order_id) as total_number from orders;
```

total_number
21350

2.Calculate the total revenue generated from pizza sales.





3. IDENTIFY THE HIGHEST-PRICED PIZZA.

Query 1

Limit to 1000 rows

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

Result Grid | Filter Rows:

	name	price
▶	The Greek Pizza	35.95

4.IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
select pizzas.size,count(order_details.order_details_id)
from pizzas join order_details on
pizzas.pizza_id=order_details.pizza_id group by pizzas.size;
```

Result Grid | Filter Rows: | Export:

size	count(order_details.order_details_id)
M	15385
L	18526
S	14137
XL	544
XXL	28

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

Query 1

Limit to 1000 rows

```
1  select pizza_types.name,sum(order_details.quantity) as quantity
2  from pizza_types join pizzas on
3  pizza_types.pizza_type_id=pizzas.pizza_type_id
4  join order_details on order_details.pizza_id=pizzas.pizza_id
5  group by pizza_types.name order by quantity desc limit 5;
6
```

```
5  group by pizza_types.name order by quant
6
```

Result Grid | Filter Rows: | Export:

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

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

Query 1

Limit to 1000 rows

```
1  select pizza_types.category,sum(order_details.quantity)as quantity
2   from pizza_types join pizzas on
3   pizza_types.pizza_type_id=pizzas.pizza_type_id join
4   order_details on order_details.pizza_id=pizzas.pizza_id group  by category
5   order by quantity desc;
6
```

Result Grid | Filter Rows:

category	quantity
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

1.ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
Limit to 1000 rows

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

Result Grid | Filter Rows: | Export:

date	cum_revenue
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

Result 8