



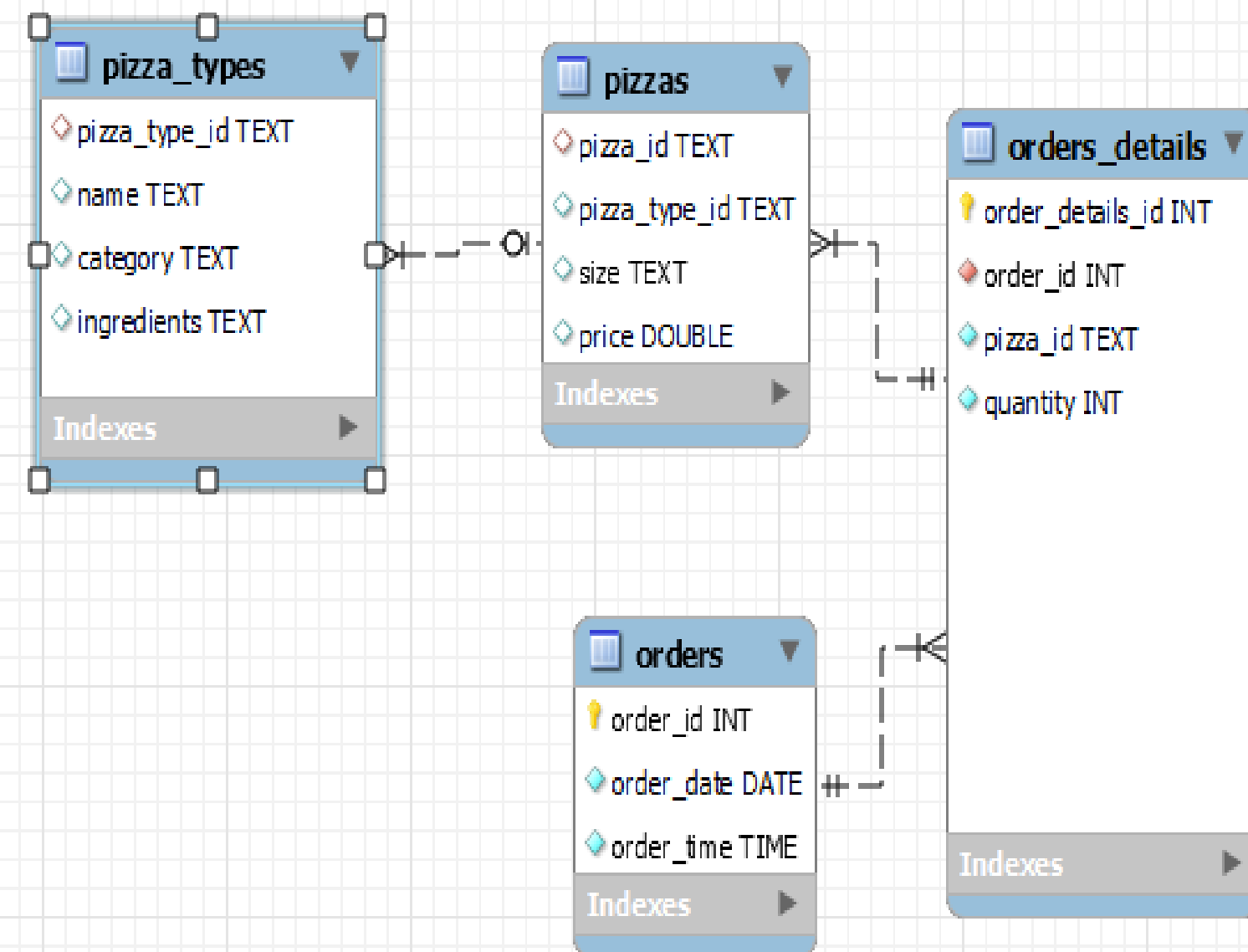
# Pizza Store data analysis

SQL Project



# Project Diagram

This Pizza hut sales Data has 4 tables.



# Top 3 Pizza type by revenue in each category

```
1  -- top 3 pizza type by revenue in each 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
7   (select pizza_types.category, pizza_types.name,
8    sum(orders_details.quantity* pizzas.price) as revenue
9    from pizza_types join pizzas
10   on pizza_types.pizza_type_id = pizzas.pizza_type_id
11   join orders_details
12   on orders_details.pizza_id = pizzas.pizza_id
13   group by pizza_types.category, pizza_types.name) as a) as b
14  where rn<=3 ;
```

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

Result 1 x

Output

Action Output

# Analyze the cumulative revenue generated over time

```
1  -- analyze the cumulative revenue generated over time.alter
2
3  • select order_date,
4      sum(revenue) over(order by order_date) as cum_revenue
5  from
6  (select orders.order_date, sum(orders_details.quantity * pizzas.price) as revenue
7   from orders_details join pizzas
8   on orders_details.pizza_id = pizzas.pizza_id
9   join orders
10  on orders.order_id = orders_details.order_id
11  group by orders.order_date) as sales ;
12
```

&lt;

Result Grid



Filter Rows:

Export:



Wrap Cell Content:

	order_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

Result 1 ×

Output

# Calculate the percentage contribution of each pizza category by revenue.

```
1  -- calculate the % contribution of each pizza category by revenue.
2
3
4  • select pizza_types.category ,
5      sum(orders_details.quantity * pizzas.price)/
6      (select sum(orders_details.quantity * pizzas.price)
7       from orders_details join pizzas
8       on pizzas.pizza_id = orders_details.pizza_id)*100 as rev
9  from pizza_types join pizzas
10     on pizzas.pizza_type_id = pizza_types.pizza_type_id
11     join orders_details
12     on orders_details.pizza_id= pizzas.pizza_id
13     group by pizza_types.category order by rev desc ;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	category	rev
▶	Classic	26.905960255669903
	Supreme	25.45631126009884
	Chicken	23.955137556847493
	Veggie	23.682590927384783

# 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 pizza_types.name, sum(orders_details.quantity * pizzas.price) as rev
4    from orders_details join pizzas
5    on pizzas.pizza_id = orders_details.pizza_id
6    join pizza_types
7    on pizzas.pizza_type_id = pizza_types.pizza_type_id
8    group by pizza_types.name
9    order by rev desc limit 3 ;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows:
	name	rev		
▶	The Thai Chicken Pizza	43434.25		
	The Barbecue Chicken Pizza	42768		
	The California Chicken Pizza	41409.5		

# 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 round(avg (quantity),0) from
4  ( select orders.order_date , sum(orders_details.quantity) as quantity
5    from orders join orders_details
6    on orders.order_id = orders_details.order_id
7    group by orders.order_date)
8    as order_quantity ;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	round(avg (quantity),0)
▶	138

# Category wise Distribution of pizzas

```
1  -- category wise dist of pizzas
2
3  •  select category ,count(name)
4     from pizza_types
5     group by category;
```



Result Grid



Filter Rows:

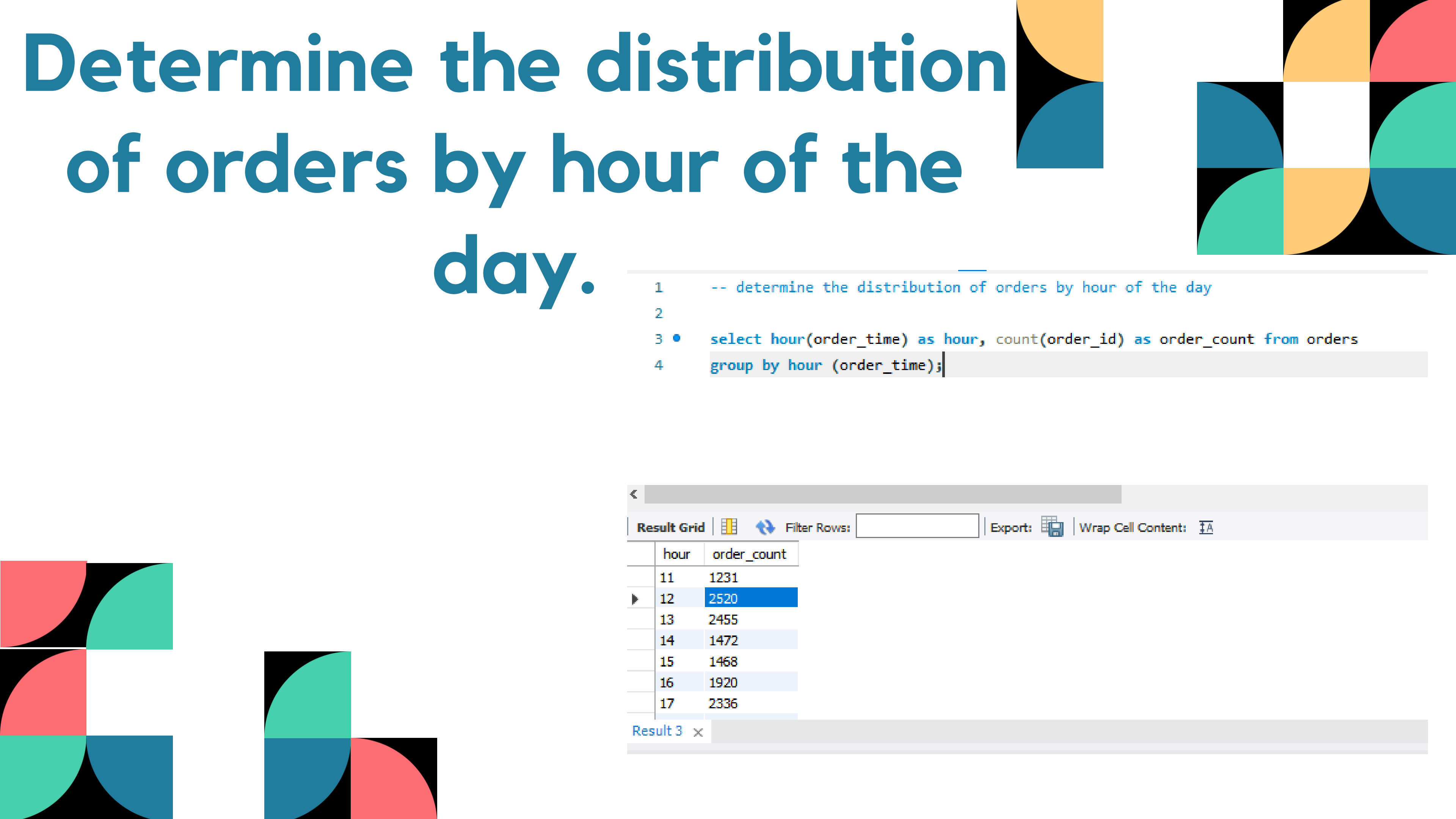
Export:



W

	category	count(name)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9





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

```
1  -- determine the distribution of orders by hour of the day
2
3  • select hour(order_time) as hour, count(order_id) as order_count from orders
4  group by hour (order_time);
```

<

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	hour	order_count
	11	1231
▶	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336

Result 3

×

# Determine the distribution of orders by hour of the day

```
1  -- determine the distribution of orders by hour of the day
2
3  • select hour(order_time) as hour, count(order_id) as order_count from orders
4    group by hour (order_time);
```

<

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	hour	order_count
	11	1231
▶	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336

# Find the total quantity of each pizza category ordered

```
1  -- find the total quantity of each pizza category ordered.
2
3  • select pizza_types.category, sum(orders_details.quantity) as quantity
4    from pizza_types join pizzas
5    on pizza_types.pizza_type_id = pizzas.pizza_type_id
6    join orders_details
7    on orders_details.pizza_id = pizzas.pizza_id
8    group by pizza_types.category order by quantity desc;
```

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

	category	quantity
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

# List 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 pizza_types.name,
4     sum(orders_details.quantity) as quantity
5  from pizza_types join pizzas
6  on pizza_types.pizza_type_id = pizzas.pizza_type_id
7  join orders_details
8  on orders_details.pizza_id = pizzas.pizza_id
9  group by pizza_types.name order by quantity desc 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




# Identify the most common pizza size ordered

```
1  -- identify the most common pizza size ordered
2
3  • select pizzas.size, count(orders_details.order_details_id) as order_count
4     from pizzas join orders_details
5     on pizzas.pizza_id = orders_details.pizza_id
6     group by pizzas.size
7     order by order_count desc ;
8
9
```

Result Grid			Filter Rows:	Export:	Wrap Cell Content:
	size	order_count			
▶	L	18526			
	M	15385			
	S	14137			
	XL	544			
	XXL	28			

# Identify the highest price pizza

```
1  -- identify the highest prices 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 limit 1;
```

Result Grid    Filter Rows: <input type="text" value=""/>   Export:    Wrap Cell Content:    Fetch rows:		
	name	price
▶	The Greek Pizza	35.95

# Retrieve the total no of orders placed

```
1      -- retrieve the total no of orders placed
```

```
2
```

```
3 •  select count(order_id) as total_orders from orders;
```



Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	total_orders
▶	21350

# Calculate the total revenue generated from pizza sales.

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

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	total_revenue
▶	817860.05





**THANK YOU**