Pizza Sales Query

```sql
-- 1. Retrieve the total number of orders placed.
select count(order_id) as total_orders from orders
```

total_orders
21350

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

total_sales
817860.05

```sql
-- 3. Identify the highest-priced pizza.
select pizza_types.name, pizzas.price
from pizzas join pizza_types
on pizzas.pizza_type_id = pizza_types.pizza_type_id
order by pizzas.price desc limit 1;
```

name	price
The Greek Pizza	35.95

```sql
-- 4. Identify the most common pizza size ordered.
select pizzas.size, count(order_details.order_details_id) as order_count
from pizzas join order_details
on pizzas.pizza_id = order_details.pizza_id
group by pizzas.size order by order_count desc;
```

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

```sql
-- 5. List the top 5 most ordered pizza types along with their quantities.
select pizza_types.name, sum(order_details.quantity) as total_quantity
from pizza_types join pizzas on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name
order by total_quantity desc limit 5 ;
```

name	total_quantity
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371

```sql
-- 6. Join the necessary tables to find the total quantity of each pizza category ordered.
select pizza_types.category, sum(order_details.quantity) as total_quantity
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category order by total_quantity;
```

	category	total_quantity
▶	Chicken	11050
	Veggie	11649
	Supreme	11987
	Classic	14888

```sql
-- 7. Determine the distribution of orders by hour of the day.
select hour(time) as hourse, count(order_id) as order_count
from orders
group by hour(time) ;
```

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

```sql
-- 8. Join relevant tables to find the category-wise distribution of pizzas.
select category , count(name) from pizza_types
group by category;
```

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

```sql
-- 9. Group the orders by date and calculate the average number of pizzas ordered per day.
select avg(total_quantity) from
( select orders.date, sum(order_details.quantity) as total_quantity
from orders join order_details
on orders.order_id = order_details.order_id
group by orders.date) as order_quantity;
```

	avg(total_quantity)
▶	138.4749

```sql
-- 10. Determine the top 3 most ordered pizza types based on revenue.
select sum(pizzas.price * order_details.quantity) as revenue, pizza_types.name
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.name order by revenue desc limit 3;
```

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

```sql
-- 11. Analyze the cumulative revenue generated over time.
select date,
sum(revenue) over (order by date) as cum_revenue
from
(select orders.date,
sum(order_details.quantity * pizzas.price) as revenue
from order_details join orders
on order_details.order_id = orders.order_id
join pizzas
on pizzas.pizza_id = order_details.pizza_id
group by orders.date order by revenue) as sales;
```

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
2015-01-09	21526.4
2015-01-10	23990.350000000002
2015-01-11	25862.65
2015-01-12	27781.7
2015-01-13	29831.300000000003

```sql
-- 12.Determine the top 3 most ordered pizza types based on revenue for each pizza category.
select name , revenue
from
(select revenue, category, name,
rank() over(partition by category order by revenue) as rn
from
(select pizza_types.category, pizza_types.name,
sum((order_details.quantity )* pizzas.price) as revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
join order_details
on order_details.pizza_id = pizzas.pizza_id
group by pizza_types.category, pizza_types.name )as a) as b
where rn <= 3;
```

name	revenue
The Chicken Pesto Pizza	16701.75
The Chicken Alfredo Pizza	16900.25
The Southwest Chicken Pizza	34705.75
The Pepperoni, Mushroom, and Peppers Pizza	18834.5
The Big Meat Pizza	22968
The Napolitana Pizza	24087
The Brie Carre Pizza	11588.4999999999
The Spinach Supreme Pizza	15277.75
The Calabrese Pizza	15934.25
The Green Garden Pizza	13955.75
The Mediterranean Pizza	15360.5
The Spinach Pesto Pizza	15596

```sql
-- 13. Calculate the percentage contribution of each pizza type to total revenue.
select pizza_types.category,
round(sum(pizzas.price * order_details.quantity) /
(select sum(order_details.quantity * pizzas.price)
from order_details
join pizzas on pizzas.pizza_id = order_details.pizza_id) * 100,2) as revenue_per
from pizzas
join pizza_types on pizzas.pizza_type_id = pizza_types.pizza_type_id
join order_details on pizzas.pizza_id = order_details.pizza_id
group by pizza_types.category
order by revenue_per desc;
```

	category	revenue_per
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68