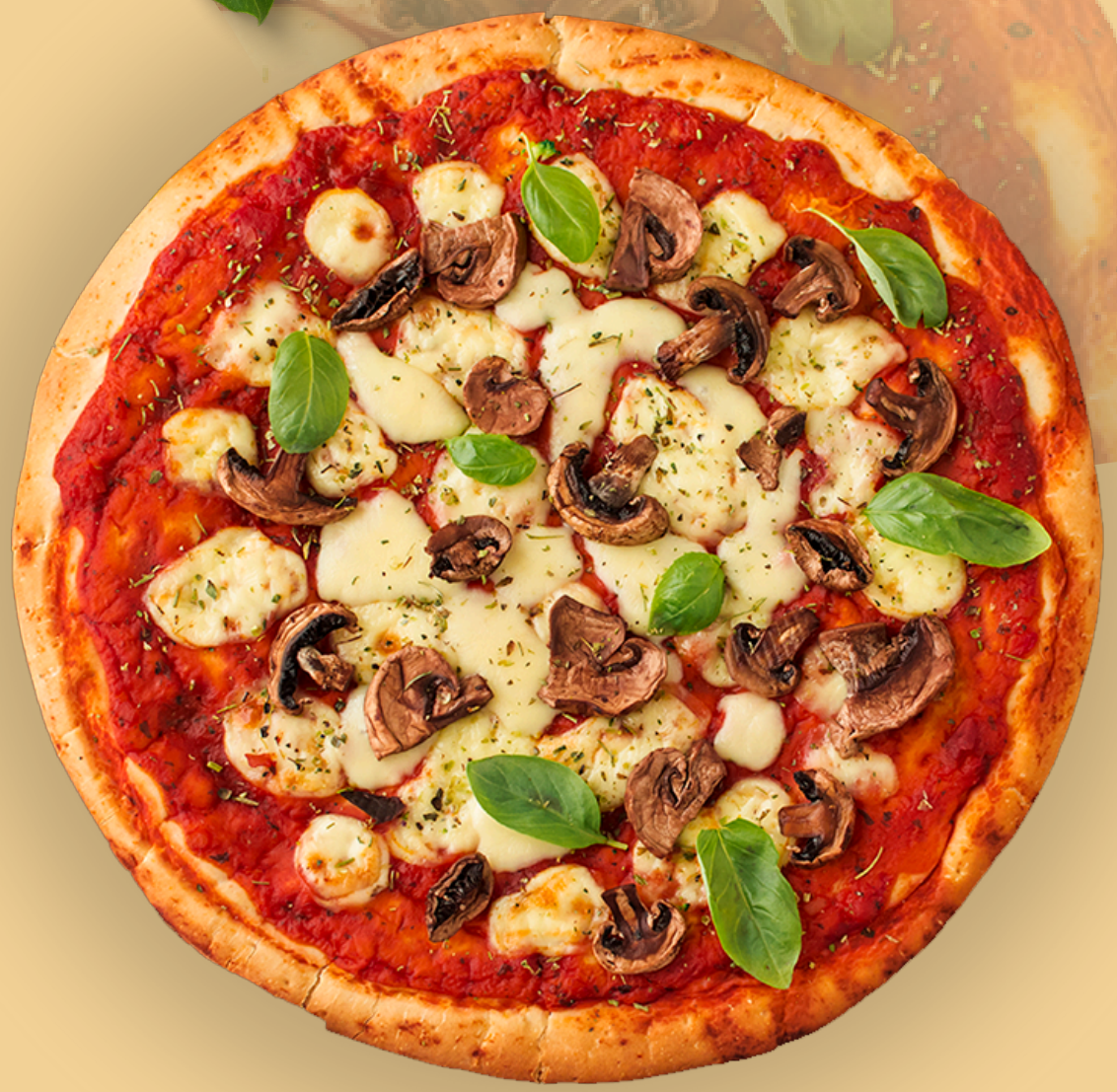
Analyzing Pizza Sales Through Sql



Schema

```
create database pizzajunction;


create table orders (
order_id integer not null,
order_date date not null,
order_time time not null,
constraint pk_id primary key(order_id) );


create table order_details (
order_details_id integer not null,
order_id integer not null,
pizza_id text not null,
quantity integer not null,
constraint pk_details primary key(order_details_id) );
```

Retrieve the total number of orders placed.

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

Result Grid

total_orders
2726

Calculate the total revenue generated from pizza sales.

```sql
-- Calculate the total revenue generated from pizza sales.

select
sum(order_details.quantity*pizzas.price) as total_revenue
from order_details join pizzas
on pizzas.pizza_id = order_details.pizza_id
```

Result Grid

	total_revenue
▶	11778.3

Identify the highest-priced pizza along with its price.

```sql
select pizza_types.name,pizzas.price
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
order by pizzas.price desc limit 1;
```

Result Grid | Filter Rows

	name	price
▶	The Greek Pizza	35.95

Identify the lowest-priced pizza along with its price.

```sql
select pizza_types.name,pizzas.price
from pizza_types join pizzas
on pizza_types.pizza_type_id = pizzas.pizza_type_id
WHERE price = (SELECT MIN(price) FROM pizzas);
```

Result Grid | Filter Rows:

	name	price
▶	The Pepperoni Pizza	9.75

Identify the most common pizza size ordered.

```sql
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 limit 1;
```

Result Grid | Filte

	size	order_count
▶	L	287

List the top 5 most ordered pizza types along with their quantities.

```sql
select pizza_types.name,sum(order_details.quantity) as 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 quantity desc limit 5;
```

Result Grid | Filter Rows:

name	quantity
The Barbecue Chicken Pizza	40
The Italian Supreme Pizza	38
The Thai Chicken Pizza	36
The Pepperoni Pizza	36
The California Chicken Pizza	34

Identify the least ordered pizza types along with its quantity.

```sql
select pizza_types.name,sum(order_details.quantity) as 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 quantity asc limit 1;
```

Result Grid | Filter Rows:

name	quantity
The Calabrese Pizza	4

Join the necessary tables to find the total quantity of each pizza category ordered.

```sql
select pizza_types.category,
sum(order_details.quantity) as 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 quantity desc;
```

Result Grid

category	quantity
Classic	207
Supreme	170
Chicken	167
Veggie	162

Determine the distribution of orders by hour of the day.

```sql
select hour(order_time) as hours,count(order_id) as orders
from orders
group by hour(order_time);
```

Result Grid

hours	orders
11	159
12	304
13	291
14	238
15	187
16	234
17	302
18	301
19	250
20	223
21	152
22	84
23	1

Query Output

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

```
select category, count(name) as pizza_count from pizza_types
group by category;
```

Result Grid | Filter R

category	pizza_count
Chicken	6
Classic	8
Supreme	9
Veggie	9

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

```sql
select round(avg(quantity)) from
(select orders.order_date,sum(order_details.quantity) as quantity
from orders join order_details
on orders.order_id = order_details.order_id
group by orders.order_date) as order_quantity;
```

Result Grid | Filter

	round(avg(quantity))
▶	141

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

```sql
select pizza_types.name,
sum(order_details.quantity*pizzas.price) as revenue
from pizza_types join pizzas
on pizzas.pizza_type_id = pizza_types.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;
```

Result Grid | Filter Rows:

name	revenue
The Barbecue Chicken Pizza	726
The Italian Supreme Pizza	679
The Thai Chicken Pizza	667

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

```sql
select pizza_types.category,
round(sum(order_details.quantity*pizzas.price) / (select
round(sum(order_details.quantity*pizzas.price))
as total_sales from order_details join pizzas
on pizzas.pizza_id = order_details.pizza_id)*100,2) 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 order by revenue desc;
```

Result Grid | Filt

category	revenue
Classic	25.77
Chicken	25.55
Supreme	25.32
Veggie	23.36

Analyze the cumulative revenue generated over time.

```sql
select order_date,round(sum(revenue) over(order by order_date),1)
as cum_revenue
from (select orders.order_date,
sum(order_details.quantity*pizzas.price) as revenue
from order_details join pizzas
on order_details.pizza_id = pizzas.pizza_id
join orders
on orders.order_id = order_details.order_id
group by orders.order_date) as sales;
```

Result Grid | Filter Rows

order_date	cum_revenue
2015-01-01	2713.9
2015-01-02	5445.8
2015-01-03	8108.2
2015-01-04	9863.6
2015-01-05	11778.3