



LARANA PIZZA

PIZZA SALES ANALYSIS





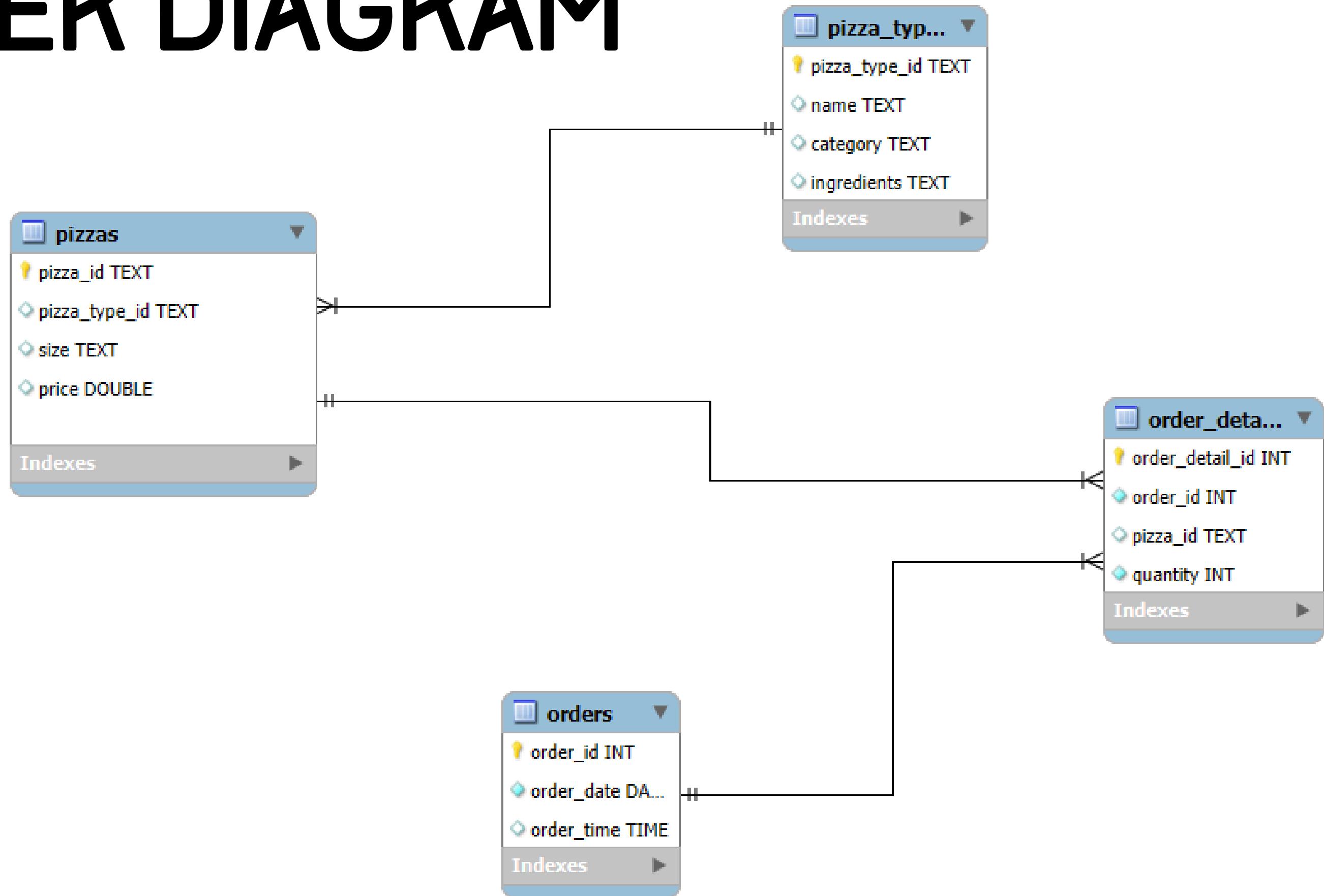
LARANA PIZZA

WELCOME TO PIZZAHUT

I WORKED ON A PROJECT CALLED PIZZA SALES ANALYSIS, WHERE I CREATED A RELATIONAL DATABASE SYSTEM TO ANALYZE THE SALES AND PERFORMANCE OF DIFFERENT PIZZAS. THE GOAL WAS TO GAIN INSIGHTS INTO SALES TRENDS, POPULAR PIZZAS, AND REVENUE GENERATION USING SQL QUERIES.



ER DIAGRAM



1. IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

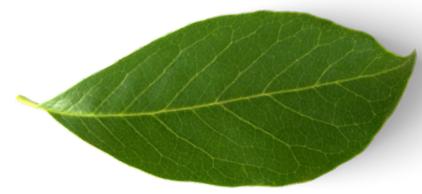
-- 1. Identify the most common pizza size ordered.

```
SELECT  
    SUM(quantity) AS total_orders, size  
FROM  
    pizzas p  
        INNER JOIN  
    order_details o ON p.pizza_id = o.pizza_id  
GROUP BY p.size  
ORDER BY total_orders desc;
```

Result Grid | Filter Rows:

	total_orders	size
▶	248	L
	175	M
	151	S
	8	XL

2. RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED



-- 2. Retrieve the total number of orders placed.

```
SELECT  
    COUNT(*) AS total_order  
FROM  
    orders;
```

Result Grid	
	total_order
▶	21350



3. CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES

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

```
SELECT  
    SUM(p.price * o.quantity) AS total_revenue  
FROM  
    pizzas p  
    INNER JOIN  
    order_details o ON p.pizza_id = o.pizza_id;
```

Result Grid	
	total_revenue
▶	9735.85

4. IDENTIFY THE HIGHEST-PRICED PIZZA.

```
-- 4. Identify the highest-priced pizza.  
SELECT  
    pt.name, p.price  
FROM  
    pizzas p  
        INNER JOIN  
    pizza_types pt ON p.pizza_type_id = pt.pizza_type_id  
ORDER BY p.price DESC  
LIMIT 1;
```

Result Grid

	name	price
▶	The Greek Pizza	35.95

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

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

	name	sum(quantity)
▶	The Barbecue Chicken Pizza	35
	The Pepperoni Pizza	34
	The Italian Supreme Pizza	33
	The Thai Chicken Pizza	31
	The Classic Deluxe Pizza	29

6. FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED

```
-- find the total quantity of each pizza category ordered.  
SELECT  
    pt.category, SUM(o.quantity) AS total_quantity  
FROM  
    pizza_types pt  
        INNER JOIN  
    pizzas p ON pt.pizza_type_id = p.pizza_type_id  
        INNER JOIN  
    order_details o ON o.pizza_id = p.pizza_id  
GROUP BY pt.category;
```

Result Grid | Filter Row

	category	total_quantity
▶	Classic	182
	Veggie	132
	Supreme	133
	Chicken	135



7. LIST ALL AVAILABLE PIZZAS ALONG WITH THEIR SIZE AND PRICES

```
-- list all available pizzas along with their size and prices;
```

SELECT

```
    pt.name, p.size, p.price
```

FROM

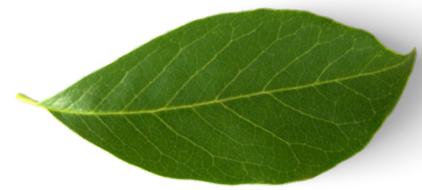
```
pizzas p
```

```
    INNER JOIN
```

```
pizza_types pt ON p.pizza_type_id = pt.pizza_type_id;
```

	name	size	price
▶	The Barbecue Chicken Pizza	S	12.75
	The Barbecue Chicken Pizza	M	16.75
	The Barbecue Chicken Pizza	L	20.75
	The California Chicken Pizza	S	12.75
	The California Chicken Pizza	M	16.75
	The California Chicken Pizza	L	20.75
	The Chicken Alfredo Pizza	S	12.75
	The Chicken Alfredo Pizza	M	16.75
	The Chicken Alfredo Pizza	L	20.75
	The Chicken Pesto Pizza	S	12.75
	The Chicken Pesto Pizza	M	16.75
	The Chicken Pesto Pizza	L	20.75

8.DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY



```
-- Determine the distribution of orders by hour of the day.
```

```
SELECT
```

```
    HOUR(order_time), COUNT(order_id)
```

```
FROM
```

```
    orders
```

```
GROUP BY HOUR(order_time);
```

	hour(order_time)	count(order_id)
▶	11	1231
	12	2520
	13	2455
	14	1472
	15	1468
	16	1920
	17	2336
	18	2399
	19	2009
	20	1642
	21	1198
	22	667

Result 2 ×





S. FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.



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

SELECT

```
category, COUNT(name) AS name  
FROM  
pizza_types  
GROUP BY category;
```

Result Grid | Filter |

	category	name
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



10.GROUP THE ORDERS BY DATE AND CALCULATE THE AVG NUMBER OF PIZZAS ORDERED PER DAY

```
1    -- Group the orders by date and calculate the avg number of pizzas ordered per day.  
2 • ⏷ select avg(total_quantity)as avg_pizza_ordered_per_day from (  
3     select o.order_date,count(od.order_id) as total_quantity  
4     from orders o  
5     inner join  
6     order_details od  
7     on o.order_id=od.order_id  
8     group by o.order_date) as daily_goals ;
```

The screenshot shows a database query results window. At the top, there are navigation icons (back, forward, search, refresh, etc.) and tabs for 'Result Grid' and 'Filter Rows'. The 'Result Grid' tab is selected. Below the tabs, there is a header row with the column name 'avg_pizza_ordered_per_day'. Underneath is a single data row containing the value '128.0000'. The entire grid has a light gray background.

	avg_pizza_ordered_per_day
▶	128.0000

11. TOTAL REVENUE GENERATED BY EACH PIZZA TYPE

```
-- total revenue generated by each pizza type
SELECT
    pt.category, SUM(p.price * od.quantity) AS total_revenue
FROM
    pizzas p
        INNER JOIN
    order_details od ON od.pizza_id = p.pizza_id
        INNER JOIN
    pizza_types pt ON pt.pizza_type_id = p.pizza_type_id
GROUP BY pt.category;
```

Result Grid | Filter Rows: []

	category	total_revenue
>	Classic	2715.25
	Veggie	2249.2
	Supreme	2346.15
	Chicken	2425.25

12.MOST POPULAR PIZZA

```
-- most popular pizza
select pt.name, count(od.quantity) as quantity
from order_details od
inner join pizzas p
on od.pizza_id=p.pizza_id
inner join pizza_types pt
on pt.pizza_type_id=p.pizza_type_id
group by name
order by quantity desc limit 5 ;
```

Result Grid | Filter Rows:

	name	quantity
▶	The Pepperoni Pizza	29
	The Italian Supreme Pizza	28
	The Barbecue Chicken Pizza	28
	The Thai Chicken Pizza	27
	The California Chicken Pizza	24



THANK YOU!

