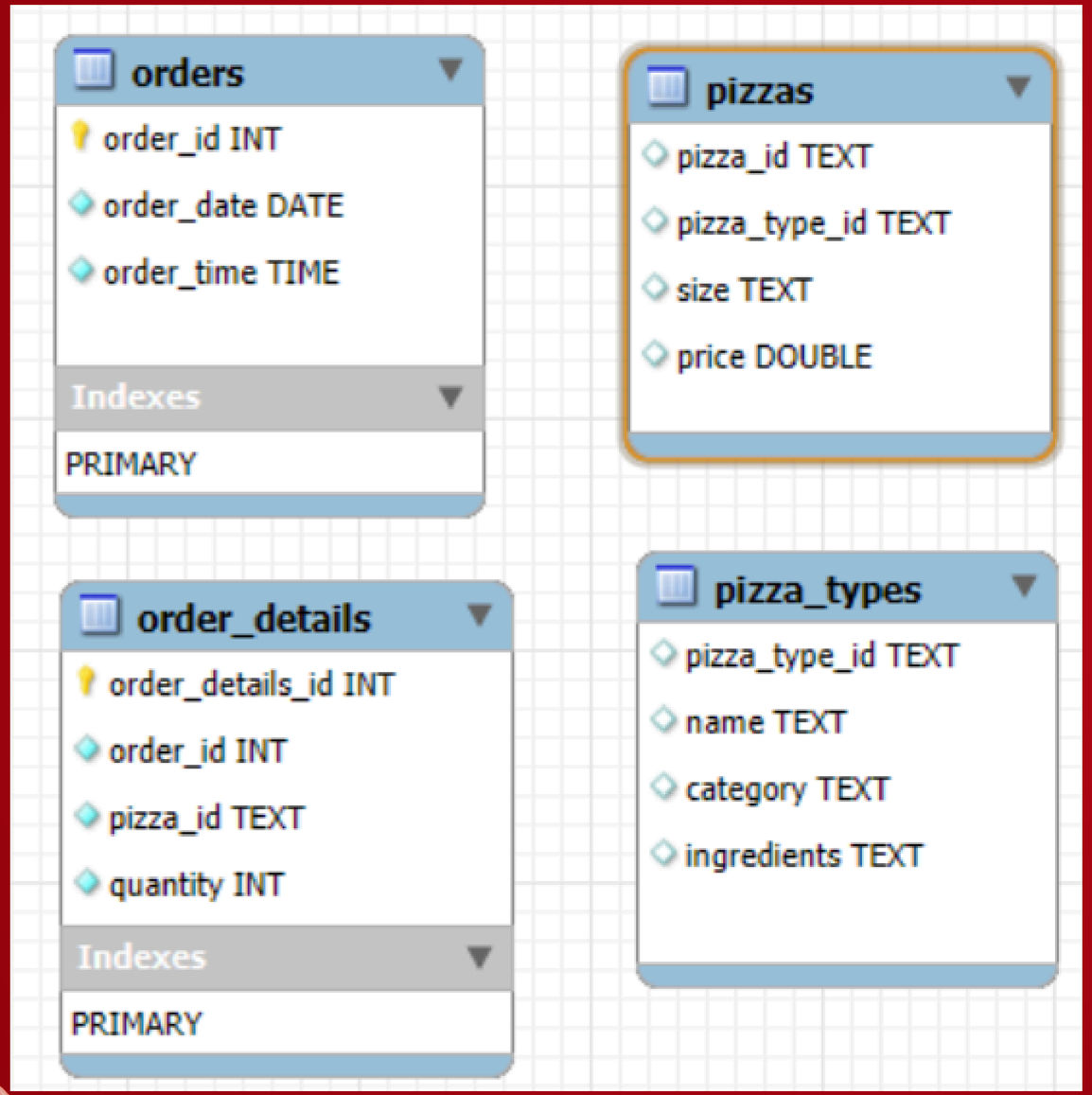
PIZZA SALES ANALYSIS WITH SQL

WELCOME TO SCHEMA



OBJECTIVES

BASIC:

- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.
- CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.
- IDENTIFY THE HIGHEST-PRICED PIZZA.
- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.
- LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

INTERMEDIATE:

- JOIN THE NECESSARY TABLES TO FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.
- DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.
- JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.
- GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.
- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

ADVANCED:

- CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.
- ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.
- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

```sql
SELECT
    COUNT(order_details_id) AS total
FROM
    order_details;
```

Result Grid

	total
▶	48620

CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```sql
SELECT
    ROUND(SUM(o.quantity * p.price), 2) AS totalRevenue
FROM
    order_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id;
```

totalRevenue
817860.05

IDENTIFY THE HIGHEST-PRICED PIZZA.

```
SELECT
    p.pizza_id, p.price, t.name
FROM
    pizzas p
        JOIN
    pizza_types t ON p.pizza_type_id = t.pizza_type_id
ORDER BY price DESC
LIMIT 1;
```

	pizza_id	price	name
▶	the_greek_xxl	35.95	The Greek Pizza

IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT
    (p.size), (SUM(o.quantity))
FROM
    order_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id
GROUP BY p.size
ORDER BY SUM(o.quantity) DESC
LIMIT 1
;
```

	size	(SUM(o.quantity))
▶	L	18956

18956

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

```
SELECT
    (t.name), (SUM(o.quantity))
FROM
    order_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id
        JOIN
    pizza_types t ON p.pizza_type_id = t.pizza_type_id
GROUP BY t.name
ORDER BY SUM(o.quantity) DESC
LIMIT 5
;
```

	name	(SUM(o.quantity))
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

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

```
SELECT
    (t.category), (SUM(o.quantity))
FROM
    order_details o
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id
        JOIN
    pizza_types t ON p.pizza_type_id = t.pizza_type_id
GROUP BY t.category
ORDER BY SUM(o.quantity) DESC

;
```

	category	(SUM(o.quantity))
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT
    hour(o.order_time) AS hod, SUM(d.quantity) AS quantity
FROM
    order_details d
        JOIN
    orders o ON d.order_id = o.order_id
GROUP BY hod
ORDER BY quantity DESC
;
```

	hod	quantity
▶	12	6776
	13	6413
	18	5417
	17	5211
	19	4406

JOIN RELEVANT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```sql
SELECT
    category, COUNT(name) AS total_count
FROM
    pizza_types
GROUP BY category
ORDER BY total_count DESC;
```

	category	total_count
▶	Supreme	9
	Veggie	9
	Classic	8
	Chicken	6

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

```sql
SELECT
    ROUND(AVG(quantity), 0) AS average_per_day
FROM
    (SELECT
        (o.order_date) AS day, SUM(d.quantity) AS quantity
    FROM
        order_details d
    JOIN orders o ON d.order_id = o.order_id
    GROUP BY day) AS data
;
```

	average_per_day
▶	138

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```sql
SELECT
    t.name AS name, ROUND(SUM(p.price * o.Quantity), 0) AS rev
FROM
    pizza_types t
        JOIN
    pizzas p ON t.pizza_type_id = p.pizza_type_id
        JOIN
    order_details o ON o.pizza_id = p.pizza_id
GROUP BY name
ORDER BY rev DESC
LIMIT 3;
```

	name	rev
▶	The Thai Chicken Pizza	43434
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41410

CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```sql
SELECT
    t.category,
    ROUND((SUM(p.price * o.Quantity) / (SELECT
                    ROUND(SUM(o.quantity * p.price), 2) AS totalRevenue
                FROM
                    order_details o
                        JOIN
                    pizzas p ON o.pizza_id = p.pizza_id) * 100),
            0) AS rev
FROM
    pizza_types t
        JOIN
    pizzas p ON t.pizza_type_id = p.pizza_type_id
        JOIN
    order_details o ON o.pizza_id = p.pizza_id
GROUP BY t.category
ORDER BY rev DESC;
```

	category	rev
▶	Classic	27
	Supreme	25
	Veggie	24
	Chicken	24

ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
select date,
sum(rev) over (order by date) as cumulative_rev
from

(SELECT
    t.order_date AS date, ROUND(SUM(p.price * o.Quantity), 0) AS rev
FROM
    orders t
        JOIN
    order_details o ON t.order_id = o.order_id
        JOIN
    pizzas p ON o.pizza_id = p.pizza_id
GROUP BY date) as sales;
```

	date	cumulative_rev
▶	2015-01-01	2714
	2015-01-02	5446
	2015-01-03	8108
	2015-01-04	9863
	2015-01-05	11929

DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```sql
select category,name from
(select category, name, rev,
(rank () over(partition by category order by rev desc)) as rnk
from
(select t.category,t.name , sum(p.price * o.quantity) as rev from order_details o
join pizzas p
on p.pizza_id = o.pizza_id
join pizza_types t
on t.pizza_type_id = p.pizza_type_id
group by t.category,t.name) as a) as b
where  rnk <=3 ;
```

	category	name
▶	Chicken	The Thai Chicken Pizza
	Chicken	The Barbecue Chicken Pizza
	Chicken	The California Chicken Pizza
	Classic	The Classic Deluxe Pizza
	Classic	The Hawaiian Pizza

BEST SELLER

★★★★★

THE CLASSIC DELUXE PIZZA

LARANA PIZZA

THANK YOU!