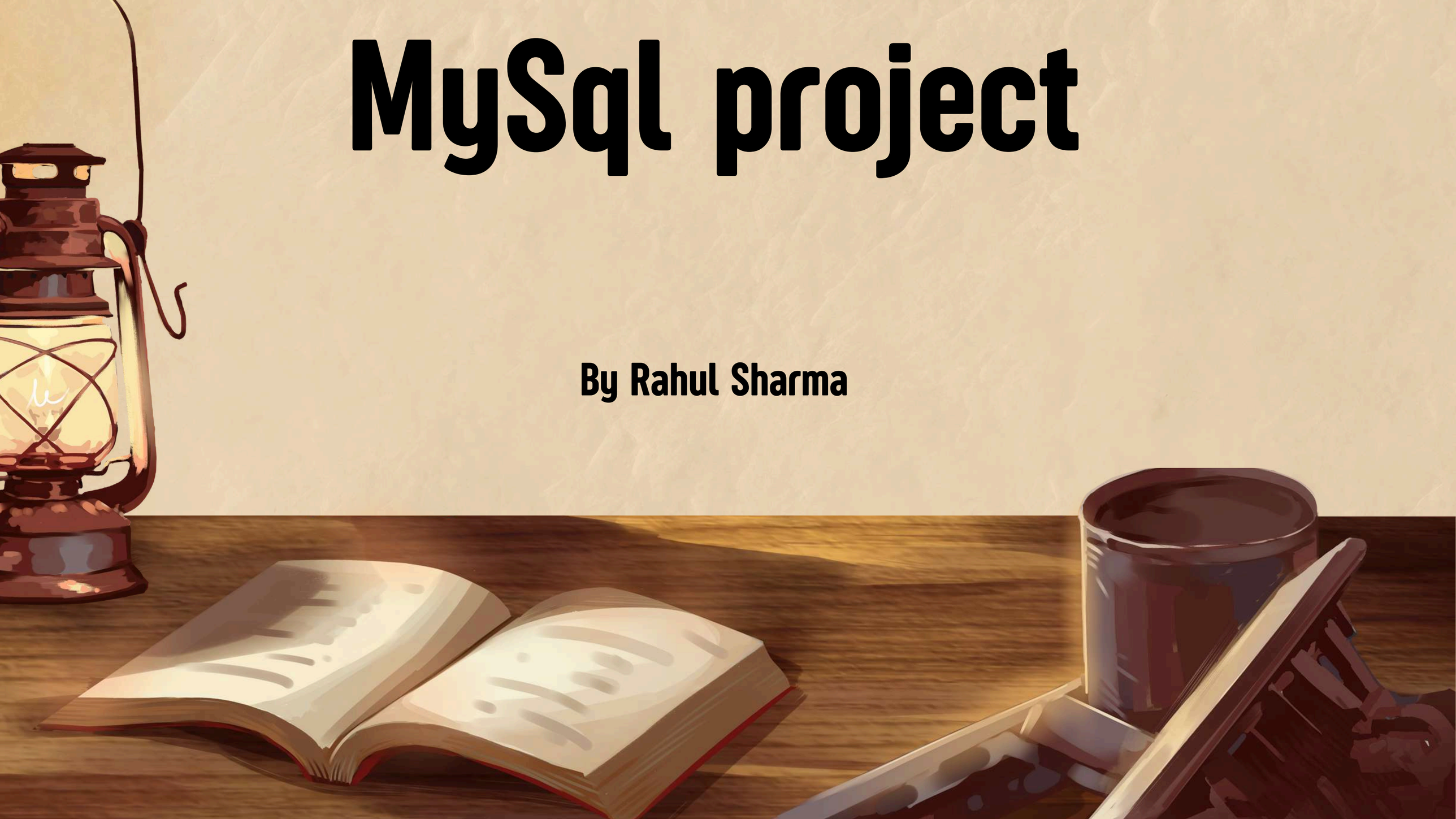
MySql project

By Rahul Sharma

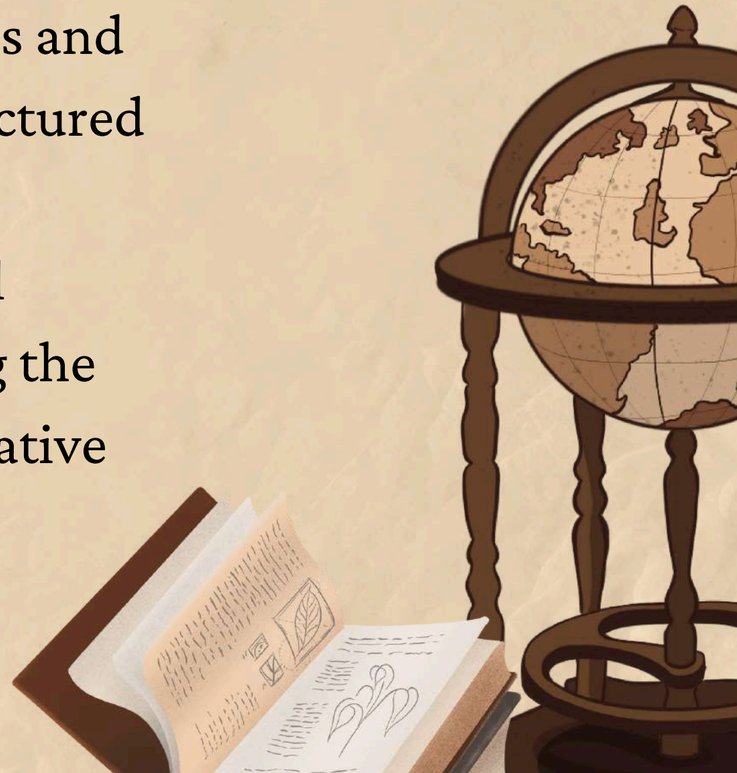


About project

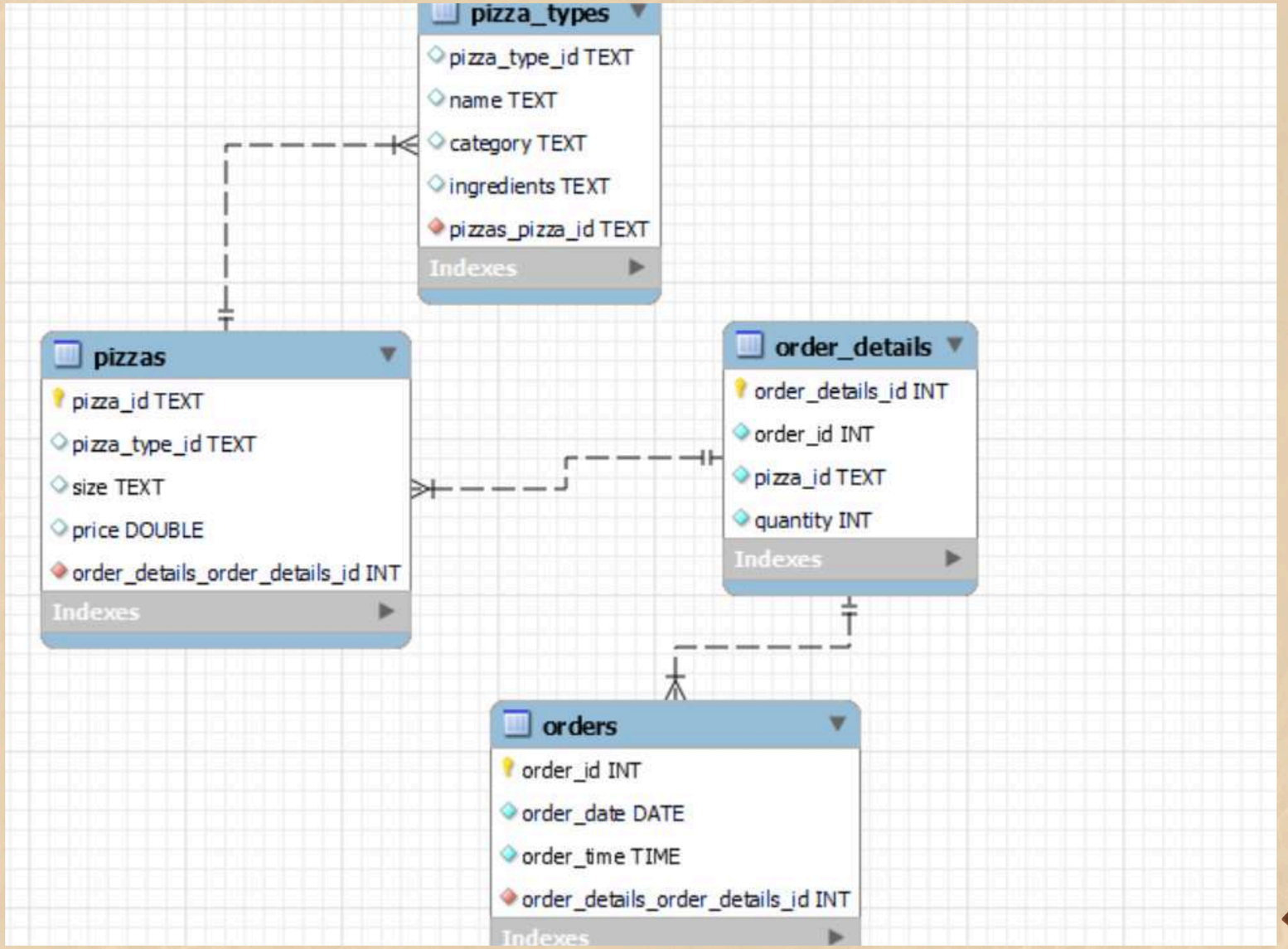
For my MySQL project, I analyze and wrote diffrent diffrent queries for pizza sales data effectively. The project encompasses several key features designed to provide valuable insights into sales performance and trends.

The database includes tables for orders,pizzas, pizza_types and order_details each interconnected through a carefully structured Entity-Relationship (ER) diagram.

I executed various SQL queries to extract meaningful information, such as calculating total revenue, analyzing the distribution of different pizza categories, tracking cumulative sales over time, and identifying best-selling items.



Database Schema



Calculate the total revenue generated from pizza sales.

```
SELECT
ROUND(SUM(p.price * od.quantity), 2) total_sales
FROM
 pizzas p
  JOIN
order_details od USING (pizza_id)
```

Identify the most common pizza size ordered.y

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

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

```
SELECT
pt.name, SUM(od.quantity) quantity
FROM
pizzas p
JOIN
order_details od USING (pizza_id)
JOIN
pizza_types pt USING (pizza_type_id)
GROUP BY pt.name
ORDER BY quantity DESC
LIMIT 5;
```

Determine the distribution of orders by hour of the day.

```
SELECT
order_date,
HOUR(order_time) hour,
COUNT(order_id) total_orders
FROM
orders
GROUP BY order_date , HOUR(order_time)
```

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

```
with x as(select pt.category, pt.name,
sum(p.price*od.quantity)revenue, dense_rank()
over(partition by pt.category order by
sum(p.price*od.quantity) desc) ranks from pizzas p
join order_details od using(pizza_id)
join pizza_types pt using(pizza_type_id)
group by pt.category,pt.name)

select x.category,x.name,x.revenue from x
where ranks < 4
```

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

```
with x as(select pt.category, sum(p.price*od.quantity)
total_sales from pizzas p
join order_details od using(pizza_id)
join pizza_types pt using(pizza_type_id)
group by pt.category)

select x.category, round(x.total_sales/(select
sum(x.total_sales) from x)*100,2) as
percentage_contribution from x
order by percentage_contribution desc
```

Analyze the cumulative revenue generated over time.

```
with x as(select o.order_date as date,
round(sum(p.price*od.quantity),2) as revenue from
pizzas p
join order_details od using(pizza_id)
join orders o using(order_id)
group by o.order_date)

select x.date, sum(revenue) over(order by x.date
asc) as revenue from x
```

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

```
SELECT
ROUND(AVG(quantity), 2) AS avg_pizza_per_day
FROM
(SELECT
o.order_date, SUM(od.quantity) quantity
FROM
orders o
JOIN order_details od USING (order_id)
GROUP BY o.order_date) AS order_quantity
```

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

```
SELECT
pt.name, SUM(od.quantity * p.price) total_sales
FROM
pizzas p
JOIN
order_details od USING (pizza_id)
JOIN
pizza_types pt USING (pizza_type_id)
GROUP BY pt.name
ORDER BY total_sales DESC
LIMIT 3;
```


Thank You