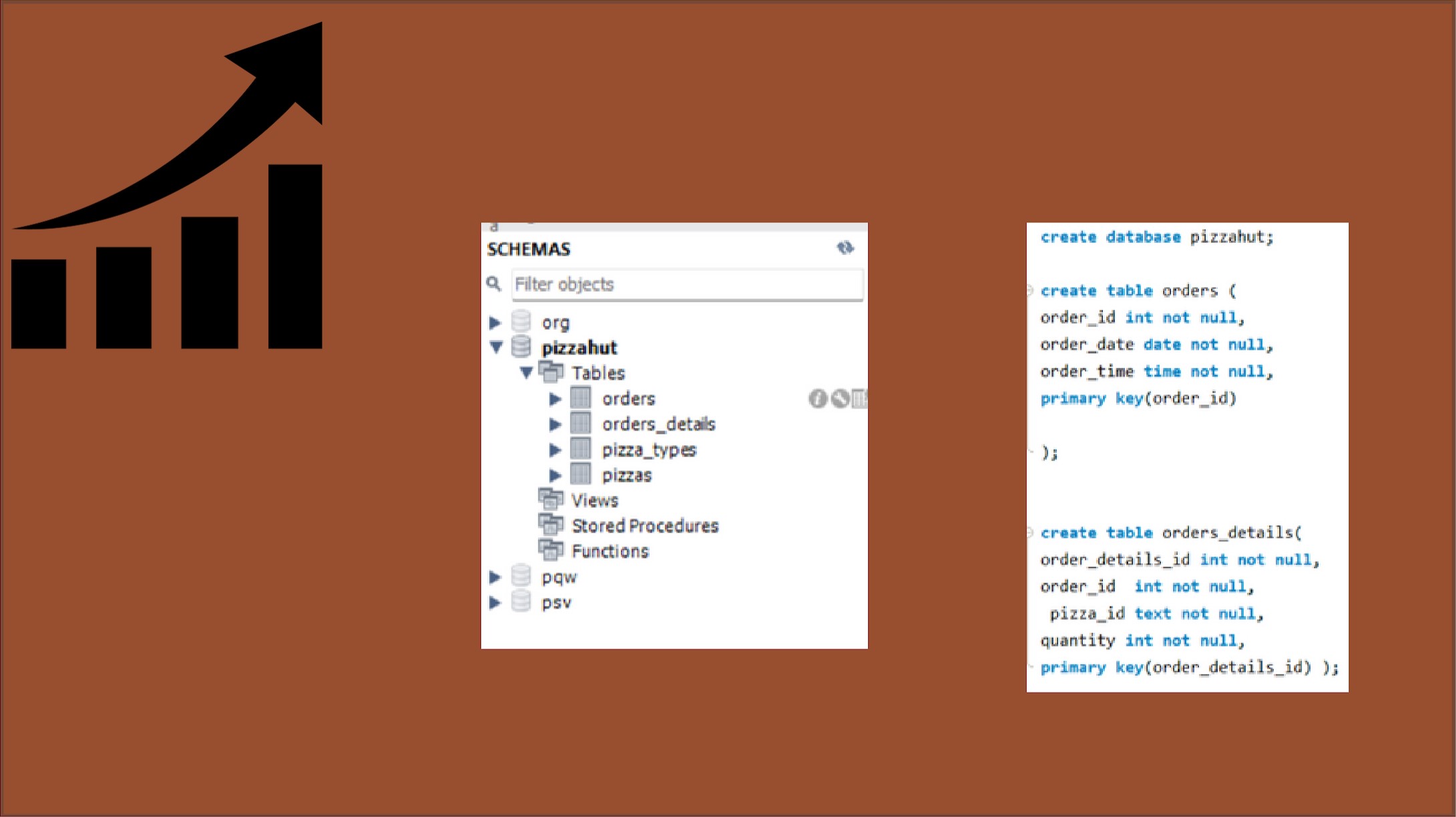
|  |
| --- |
| SQL project  On Pizza sales  -------by prashant patel |



etrieve the total number of orders placed.

SELECT

COUNT (order\_id) AS total\_orders

FROM orders •

Result Grid

|  |  |
| --- | --- |
|  | total orders |
|  | 21350 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Calculate the total revenue generated from pizza sales.  SELECT  \* pizzas-price),  2) AS total\_revenue  FROM orders details JOIN pizzas ON pizzas-pizza\_id — orders\_details.pizza {d;  Result Grid | | | | | | | |
|  | | total\_revenue | | | | | |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Identify the highest-priced pizza,  SELECT pizza\_types. name, pizzas-price  FROM pizza.types  JOIN pizzas ON pizza\_types. pizzas.  ORDER BY pizzas. price DESC  LIMIT 1 •  Result Grid   |  |  |  | | --- | --- | --- | |  | name | price | |  | The Greek Pizza | 35095 | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Identify the most common pizza size ordered,  SELECT pizzas. size,   id) AS orders\_count  FRCPA pizzas JOIN orders\_details  ON pizzas.pizza\_id = orders details. pizza id  GROUP BY pizzas.size  Result Grid Fiher   |  |  |  | | --- | --- | --- | |  | size | orders\_count | |  | M  s XL  XXL | 18526  15385 14137 544  28 |   ORDER BY orders count DESC; |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| List the top 5 most ordered pizza types along with their quantities,  SELECTpi zza\_types name, ,quantity) Result Grid I Filter Rcvd\*   |  |  |  | | --- | --- | --- | |  | name | quantity | |  | 'The Classic Deluxe Pizza  The Barbecue Chicken Pizza  'The Hawaiian Pizza  The Pepperoni Pizza  The Thai chicken Pizza | 2453  2432 2422  2418  2371 |   AS quantity  FROM pi zza\_types  JOIN pizzas ON pizza\_types.pizza\_type\_id = pizzas, pizza\_type\_id  JOIN orders\_detaiis ON  pizzas.pizza id  GROUP BY pizza\_types.name  ORDER BY quantity DESC  LIMIT 5 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Join the necessary tables to find the total quantity of each pizza category ordered.   |  |  |  | | --- | --- | --- | |  | category | quantty | |  | Classic  Supreme  Veggie  Chicken | 14688  11987  11649  11050 |   select pizza\_types. category, sum(orders details. quantity) AS quantity from Result Grid pizza\_types join pizzas on pizza\_types. pizzas. pizza\_type\_id join orders\_details on orders\_details . pizza\_id pizzas. pizza id group by pizza\_types e category order by quantity desc; |

Determine the distribution of orders by hour of the day.

Result Grid I F

|  |  |  |
| --- | --- | --- |
|  | hot-f | order count |
|  | 1.1  12  13  14  16  17  18  19  20  21  22  23  10  9 | 1231  2520  2455  1472  1920  23%  2399  2C09  1642  1198  663  23 |

HOUR(order\_time) AS hour,

COUNT (order\_id) AS order count

FROM orders

GROUP BY HOUR(order\_time)

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

select category, count (name) from pizza.types group by category;

Result Grid I d} FiherRe

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

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

0) as avg\_order\_per\_day

FROM

(SELECT orders. order date, Result Grid

AS quantity

|  |  |  |
| --- | --- | --- |
|  | | avg\_order\_per\_day |
|  |  | 138 |

orders

JOIN orders details ON orders. order id = orders details.order id

GROUP BY orders. order\_date) AS order\_quantity;

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

select pizza\_types. name, sum (orders\_detaiis.quantity \* pizzas . price ) as revenue from pizza\_types join pizzas on pizzas. pizza\_types. Result Grid Fifter Rov6{

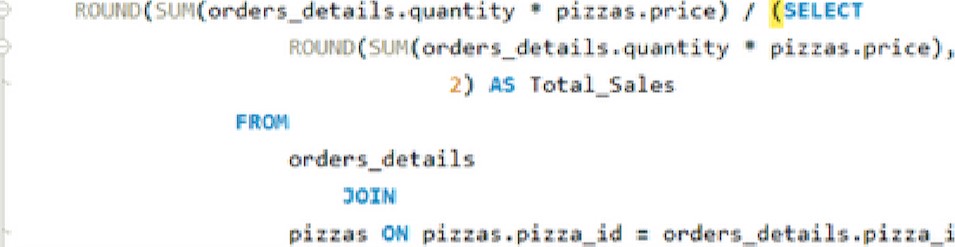


|  |  |  |
| --- | --- | --- |
|  | name | revenue |
|  | The Thai Chicken Pizza  The Barbecue Chicken Pizza  'The California Chicken Pizza | 4343425  42768  41409, 5 |

pizza\_type\_id join orders\_details on orders\_details. pizza id = pizzas. pizza\_id group by pizza\_types.name order by revenue desc limit 3

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

pizza\_types category,

price) / (SELECT v quantity • u price) Result Grid

|  |  |  |
| --- | --- | --- |
|  | category | REvenue |
|  | Classic  Supreme  Chicken  Veggie | 26.91  25.46  23.96  23.68 |

a ) AS REvenue

FROM pizzaztypes

JOIN

z pizzas-pizza\_id

GROUP EY



orders

ORDER BY Revenue OESC;

Analyze the cumulative revenue generated over time.

select order date, sum(revenue) over(order by order\_date) as Result Grid 1 cum revenue from

(select orders.order\_date, quantity \* pizzas. price) as revenue from orders details join pizzas on orders details.pizza\_id = pizzas. pizza\_id join orders on orders. order id = orders details.order id group by orders.order date) as sales;

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

select name, revenue from Result Grid I 

|  |  |  |
| --- | --- | --- |
|  |  | revenue |
|  | The Thö Chidcen Pizza  'The Barbecue eocken Pizza Chidcen Pnzza  The Classic Deluxe Pizza  'The Ha•wæan Pizza  The Pepperon Pizza  'The Spicy Italian Pozze  Sl-ven•te Pizza  'The Pizza  'The Fou Cheese Ptzza  'The Mexjcarue Pizza  'The five Cheese Pizza | 4343425 42768  41409, 5 38180.5  32273125  30161.75  34831,25 33476, 75 30940,5 |
| 32265,  2678075 260665 |

(select category, name, revenue, rank() over(partition by category order by revenue desc)

as nn from

(select pizza.types. category, pizza\_types .name,  \* pizzas. price)

as revenue from pizza\_types join pizzas on pizza\_types  : pizzas join orders\_details

on  : pizzas.pizza\_id group by pizza\_types. category, pizza\_types.name) as a) as b where rn 3

Group the orders by date and cal. the avg. no. of pizzas ordered per day.