

FAST FOOD DATA ANALYSIS SQL PROJECT

By Samay Kukreja

TOPICS COVERED

01

Aggregate Functions

02

Joins

03

CTE

04

Window Function

PROBLEM STATEMENT : -- RETRIEVE THE TOTAL NUMBER OF ORDERS PLACED.

SELECT COUNT(*) FROM ORDERS

	count bigint	🔒
1	21350	

PROBLEM STATEMENT : -- CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES.

```
SELECT SUM(P.PRICE*OP.QUANTITY) FROM PIZZAS P  
JOIN ORDER_DETAILS OP ON P.PIZZA_ID = OP.PIZZA_ID
```

	sum numeric 
1	817860.05

PROBLEM STATEMENT : -- IDENTIFY THE HIGHEST-PRICED PIZZA.

```
SELECT PIZZA_ID, PRICE FROM PIZZAS  
ORDER BY PRICE DESC  
LIMIT 1
```

	pizza_id character varying 	price numeric (10,2) 
1	the_greek_xx	35.95

PROBLEM STATEMENT : -- LIST THE TOP 5 MOST ORDERED PIZZA TYPES ALONG WITH THEIR QUANTITIES.

```
SELECT PT.NAME, SUM(OD.QUANTITY) AS SUM_TOTAL FROM PIZZA_TYPES PT
    JOIN PIZZAS P ON P.PIZZA_TYPE_ID = PT.PIZZA_TYPE_ID
    JOIN ORDER_DETAILS OD ON P.PIZZA_ID = OD.PIZZA_ID
        GROUP BY PT.NAME
        ORDER BY SUM_TOTAL DESC
        LIMIT 5
```

	name character varying	sum_total bigint
1	The Classic Deluxe Pizza	2453
2	The Barbecue Chicken Pizza	2432
3	The Hawaiian Pizza	2422
4	The Pepperoni Pizza	2418
5	The Thai Chicken Pizza	2371

PROBLEM STATEMENT : -- IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED.

```
SELECT COUNT(SIZE), SIZE FROM PIZZAS P  
JOIN ORDER_DETAILS OD ON P.PIZZA_ID = OD.PIZZA_ID  
GROUP BY SIZE  
ORDER BY COUNT(SIZE) DESC
```

	count bigint	size character varying
1	18526	L
2	15385	M
3	14137	S
4	544	XL
5	28	XXL

PROBLEM STATEMENT : -- FIND THE TOTAL QUANTITY OF EACH PIZZA CATEGORY ORDERED.

```
SELECT PT.CATEGORY AS CATEGORY, SUM(QUANTITY) FROM ORDER_DETAILS OD
    JOIN PIZZAS P ON P.PIZZA_ID = OD.PIZZA_ID
    JOIN PIZZA_TYPES PT ON PT.PIZZA_TYPE_ID = P.PIZZA_TYPE_ID
    GROUP BY CATEGORY
    ORDER BY SUM(QUANTITY) DESC
```

	category character varying 	sum bigint 
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

PROBLEM STATEMENT : -- DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.

```
SELECT COUNT(ORDER_ID),  
EXTRACT(HOUR FROM ORDER_TIME) AS  
HOURS FROM ORDERS  
GROUP BY HOURS  
ORDER BY HOURS DESC
```

	count bigint	hours numeric
1	28	23
2	663	22
3	1198	21
4	1642	20
5	2009	19
6	2399	18
7	2336	17
8	1920	16
9	1468	15
10	1472	14

PROBLEM STATEMENT : -- FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.

```
SELECT CATEGORY, COUNT(CATEGORY) FROM PIZZA_TYPES  
GROUP BY CATEGORY  
ORDER BY COUNT(CATEGORY) DESC
```

	category character varying 	count bigint 
1	Supreme	9
2	Veggie	9
3	Classic	8
4	Chicken	6

PROBLEM STATEMENT : -- GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERED PER DAY.

```
SELECT AVG(QUANT) FROM(
SELECT SUM(OD.QUANTITY) AS QUANT, EXTRACT(DAY FROM O.ORDER_DATE) AS DAY, EXTRACT(MONTH
FROM O.ORDER_DATE) AS MONTH FROM ORDER_DETAILS OD
JOIN ORDERS O ON OD.ORDER_ID = O.ORDER_ID
GROUP BY DAY, MONTH
ORDER BY MONTH, DAY
) AS AVERAGE_QUANTITY
```

	avg numeric	🔒
1	138.4748603351955307	

PROBLEM STATEMENT : -- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE.

```
SELECT P.PIZZA_TYPE_ID, SUM(PRICE*QUANTITY) AS REVENUE FROM PIZZAS P
JOIN ORDER_DETAILS OD ON P.PIZZA_ID = OD.PIZZA_ID
GROUP BY P.PIZZA_TYPE_ID
ORDER BY REVENUE DESC
LIMIT 3
```

	pizza_type_id character varying 	revenue numeric 
1	thai_ckn	43434.25
2	bbq_ckn	42768.00
3	cali_ckn	41409.50

PROBLEM STATEMENT : -- CALCULATE THE PERCENTAGE CONTRIBUTION OF EACH PIZZA TYPE TO TOTAL REVENUE.

```
SELECT PT.CATEGORY, (SUM(OD.QUANTITY * P.PRICE)/(SELECT SUM(OD.QUANTITY * P.PRICE) FROM ORDER_DETAILS OD JOIN PIZZAS P ON OD.PIZZA_ID = P.PIZZA_ID))*100 AS REVENUE  
FROM PIZZA_TYPES PT  
JOIN PIZZAS P ON PT.PIZZA_TYPE_ID = P.PIZZA_TYPE_ID  
JOIN ORDER_DETAILS OD ON OD.PIZZA_ID = P.PIZZA_ID  
GROUP BY PT.CATEGORY  
ORDER BY REVENUE DESC
```

	category character varying	revenue numeric
1	Classic	26.90596025566965888600
2	Supreme	25.45631126009883964900
3	Chicken	23.95513755684728701400
4	Veggie	23.68259092738421445100

PROBLEM STATEMENT :-- ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME.

```
SELECT ORDER_DATE, SUM(REVENUE) OVER(ORDER BY  
ORDER_DATE) AS CUMULATIVE_REVENUE FROM  
(  
SELECT O.ORDER_DATE, SUM(P.PRICE * OD.QUANTITY)  
AS REVENUE FROM ORDERS O  
JOIN ORDER_DETAILS OD ON OD.ORDER_ID =  
O.ORDER_ID  
JOIN PIZZAS P ON P.PIZZA_ID = OD.PIZZA_ID  
GROUP BY O.ORDER_DATE) AS CUMULATIVE_DATA
```

	order_date date	cumulative_revenue numeric
1	2015-01-01	2713.85
2	2015-01-02	5445.75
3	2015-01-03	8108.15
4	2015-01-04	9863.60
5	2015-01-05	11929.55
6	2015-01-06	14358.50
7	2015-01-07	16560.70
8	2015-01-08	19399.05
9	2015-01-09	21526.40
10	2015-01-10	23990.35

PROBLEM STATEMENT : -- DETERMINE THE TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY.

```

SELECT CATEGORY, NAME, REVENUE FROM (
SELECT CATEGORY, NAME, REVENUE, DENSE_RANK()
OVER(PARTITION BY CATEGORY ORDER BY REVENUE
     DESC) AS ABC FROM(
SELECT PT.CATEGORY, PT.NAME, SUM(P.PRICE *
OD.QUANTITY) AS REVENUE
     FROM PIZZA_TYPES PT
JOIN PIZZAS P ON PT.PIZZA_TYPE_ID = P.PIZZA_TYPE_ID
JOIN ORDER_DETAILS OD ON P.PIZZA_ID = OD.PIZZA_ID
     GROUP BY PT.CATEGORY, PT.NAME
) AS DEF) AS XYZ
WHERE ABC <= 3

```

	category character varying 	name character varying 	revenue numeric 
1	Chicken	The Thai Chicken Pizza	43434.25
2	Chicken	The Barbecue Chicken Pizza	42768.00
3	Chicken	The California Chicken Pizza	41409.50
4	Classic	The Classic Deluxe Pizza	38180.50
5	Classic	The Hawaiian Pizza	32273.25
6	Classic	The Pepperoni Pizza	30161.75
7	Supreme	The Spicy Italian Pizza	34831.25
8	Supreme	The Italian Supreme Pizza	33476.75
9	Supreme	The Sicilian Pizza	30940.50
10	Veggie	The Four Cheese Pizza	32265.70