

# *Pizza hut* **SALES ANALYSIS**

SQL + ANALYSIS —







**Q. Retrieve the total number of orders placed.**

```
SELECT  
    COUNT(order_id) AS total_number_of_orders  
FROM  
    orders;
```



total_number_of_orders
21350

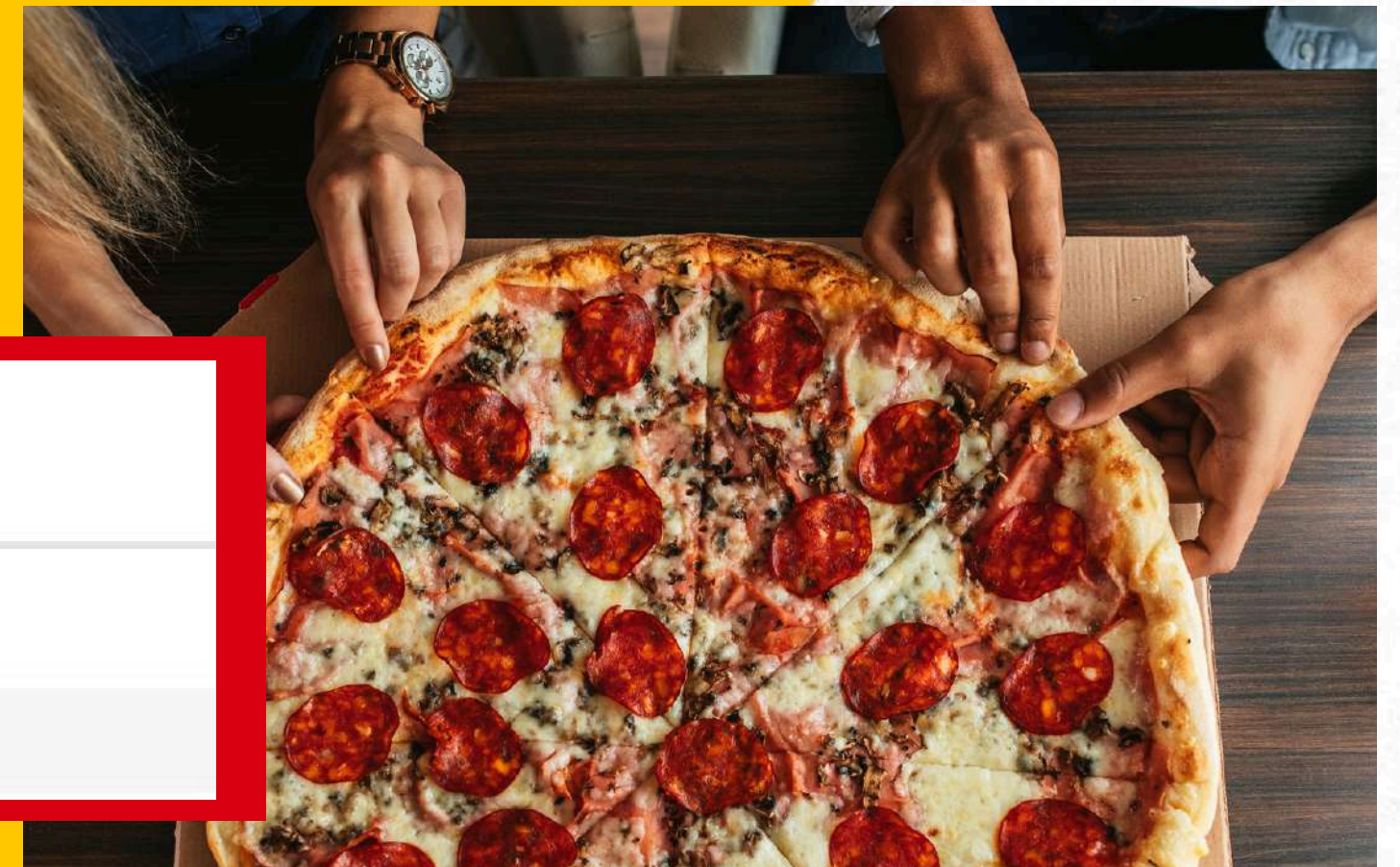


**Q. Calculate the total revenue generated from pizza sales.**

```
SELECT
    ROUND(SUM(order_details.quantity * pizzas.price),
          2) AS total_revenue
FROM
    order_details
    JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
```

total_revenue
---------------

817860.05
-----------





**Q. Identify the highest-priced pizza.**

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

name	price
The Greek Pizza	35.95







**Q. Identify the most common pizza size ordered.**

```
SELECT
  pizzas.size,
  COUNT(order_details.order_details_id) as pizza_quantity
FROM
  pizzas
  JOIN
  order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY pizza_quantity DESC
LIMIT 2;
```



size	pizza_quanti...
L	18526
M	15385



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

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



name	total_order
The Classic Deluxe Pizza	2453
The Barbecue Chicken Pizza	2432
The Hawaiian Pizza	2422
The Pepperoni Pizza	2418
The Thai Chicken Pizza	2371



**Q. Join the necessary tables to find the total quantity of each pizza category ordered**

```
SELECT
  pizza_types.category,
  SUM(order_details.quantity) AS total_quantity
FROM
  order_details
  JOIN
  pizzas ON order_details.pizza_id = pizzas.pizza_id
  JOIN
  pizza_types ON pizza_types.pizza_type_id = pizzas.pizza_type_id
GROUP BY pizza_types.category
ORDER BY total_quantity DESC;
```

category	total_quant...
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050







**Q. Determine the distribution of orders by hour of the day.**

```
SELECT
    HOUR(order_time) AS hours, COUNT(order_id) AS total_order
FROM
    orders
GROUP BY hours
ORDER BY total_order;
```

hours	total_order
9	1
10	8
23	28
22	663
21	1198
11	1231
15	1468
14	1472
20	1642
16	1920
19	2009
17	2336
18	2399
13	2455
12	2520





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

```
select pizza_types.category, count(pizza_types.name)
from pizza_types
group by category
order by count(name);
```



category	count(pizza_types.name)
Chicken	6
Classic	8
Supreme	9
Veggie	9



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

```
SELECT
    ROUND(AVG(total_order), 0)
FROM
    (SELECT
        orders.order_date,
        SUM(order_details.quantity) AS total_order
    FROM
        orders
    JOIN order_details ON orders.order_id = order_details.order_id
    GROUP BY orders.order_date) AS order_by_date;
```

ROUND(AVG(total_order),...
----------------------------

138
-----

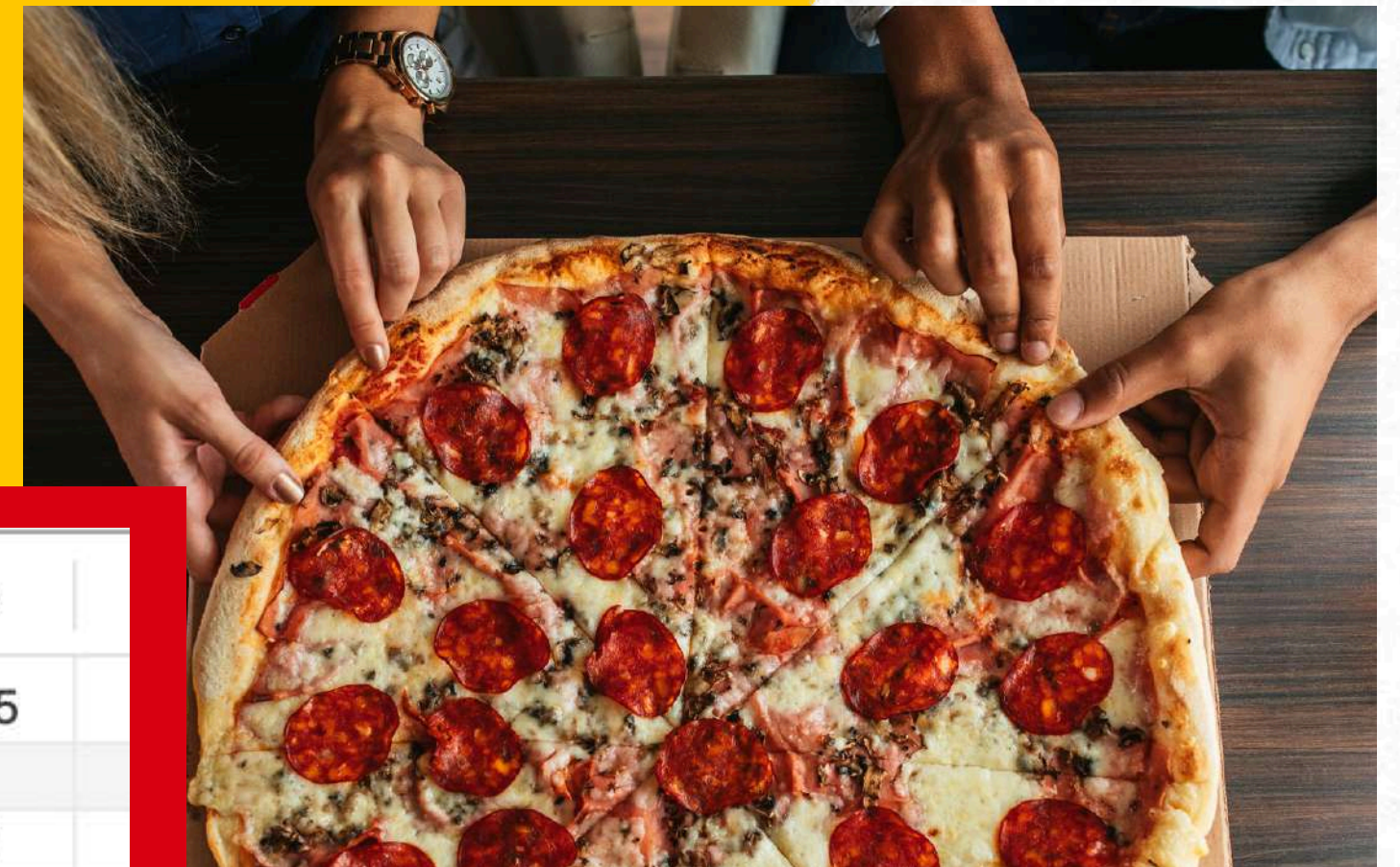




**Q. Determine the top 3 most ordered pizza types based on revenue.**

```
SELECT
    pizza_types.name,
    SUM(pizzas.price * order_details.quantity) AS revenue
FROM
    pizza_types
    JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
    JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.name
ORDER BY revenue DESC
LIMIT 3;
```

name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5





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

```
SELECT
  pizza_types.category,
  ROUND(SUM(pizzas.price * order_details.quantity) * 100 / (SELECT
    SUM(order_details.quantity * pizzas.price)
    FROM
      order_details
      JOIN
        pizzas ON order_details.pizza_id = pizzas.pizza_id),
    2) AS revenue_percentage
FROM
  pizza_types
  JOIN
    pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
  JOIN
    order_details ON order_details.pizza_id = pizzas.pizza_id
GROUP BY pizza_types.category
ORDER BY revenue_percentage
LIMIT 5;
```

category	revenue_percenta...
Veggie	23.68
Chicken	23.96
Supreme	25.46
Classic	26.91



Q. Analyze the cumulative revenue generated over time.

```
SELECT
  order_date,
  SUM(revenue) OVER (ORDER BY order_date) AS cumulative_revenue
FROM (
  SELECT
    orders.order_date,
    ROUND(SUM(pizzas.price * order_details.quantity), 2) AS revenue
  FROM orders
  JOIN order_details ON orders.order_id = order_details.order_id
  JOIN pizzas ON pizzas.pizza_id = order_details.pizza_id
  GROUP BY orders.order_date
) AS daily_revenue;
```

order_date	cumulative_revenue
2001-01-15	2713.85
2001-02-15	5903.049999999999
2001-03-15	7501.599999999999
2001-04-15	9678.449999999999
2001-05-15	12250.399999999998
2001-06-15	15318.149999999998
2001-07-15	17549.649999999998
2001-08-15	19990.199999999997
2001-09-15	22343.049999999996
2001-10-15	25545.199999999997
2001-11-15	27531.85
2001-12-15	29608.55
2002-01-15	32340.45
2002-02-15	34669.05
2002-03-15	37048.100000000006
2002-04-15	39595.250000000001
2002-05-15	41995.450000000004
2002-06-15	44445.4
2002-07-15	46740.200000000004
2002-08-15	48650.350000000006
2002-09-15	50515.900000000001
2002-10-15	52590.750000000001
2002-11-15	54889.850000000006
2002-12-15	57103.950000000004
2003-01-15	59766.350000000006
2003-02-15	62381.150000000001
2003-03-15	64669.050000000001
2003-04-15	67207.250000000001
2003-05-15	69039.500000000001
2003-06-15	70946.550000000002
2003-07-15	74389.550000000002
2003-08-15	76371.800000000002
2003-09-15	78624.400000000002
2003-10-15	81025.350000000002
2003-11-15	82853.900000000002
2003-12-15	85097.100000000002
2004-01-15	86852.550000000002
2004-02-15	89086.450000000001
2004-03-15	91530.750000000001
2004-04-15	94298.300000000002
2004-05-15	96587.200000000001
2004-06-15	98870.800000000002
2004-07-15	102735.000000000001
2004-08-15	104825.150000000001

order_date	cumulative_revenue
2004-09-15	107786.1
2004-10-15	109928.3
2004-11-15	111894.400000000001
2004-12-15	114772.750000000001
2005-01-15	116838.700000000001
2005-02-15	119054.500000000001
2005-03-15	121405.150000000001
2005-04-15	123315.000000000001
2005-05-15	125028.600000000002
2005-06-15	127728.750000000001
2005-07-15	129320.200000000001
2005-08-15	131415.050000000002
2005-09-15	133952.85
2005-11-15	136133.2
2005-12-15	138761.150000000002
2006-01-15	141190.100000000003
2006-02-15	143700.050000000005
2006-03-15	146214.000000000006
2006-04-15	148836.350000000006
2006-05-15	151146.150000000005
2006-06-15	153427.100000000006
2006-07-15	155544.000000000006
2006-08-15	157643.250000000006
2006-09-15	159134.900000000005
2006-10-15	161350.850000000006
2006-11-15	164508.350000000006
2006-12-15	166858.600000000006
2007-01-15	169060.800000000008
2007-02-15	171298.450000000007
2007-03-15	173698.900000000008
2007-04-15	175933.150000000008
2007-05-15	178162.000000000001
2007-06-15	180122.750000000001
2007-07-15	182497.150000000008
2007-08-15	185123.550000000008
2007-09-15	187406.900000000008
2007-10-15	189499.250000000001
2007-11-15	192080.200000000001
2007-12-15	194495.000000000001
2008-01-15	197333.350000000001
2008-02-15	199368.350000000001
2008-03-15	201556.500000000001
2008-04-15	203850.450000000001
2008-05-15	206902.750000000001

order_date	cumulative_revenue
2027-09-15	721869.400000000005
2027-10-15	723858.050000000005
2027-11-15	728280.500000000005
2027-12-15	729699.500000000005
2028-01-15	731715.500000000005
2028-02-15	734255.100000000004
2028-03-15	736507.200000000004
2028-04-15	738250.000000000005
2028-05-15	740266.400000000005
2028-06-15	741836.100000000004
2028-07-15	743930.650000000005
2028-08-15	746577.850000000004
2028-09-15	748613.150000000005
2028-10-15	750249.100000000004
2028-11-15	752217.150000000005
2028-12-15	753854.350000000004
2029-01-15	755899.650000000005
2029-03-15	758053.900000000005
2029-04-15	759882.650000000005
2029-05-15	762883.850000000004
2029-06-15	764757.450000000004
2029-07-15	766680.700000000004
2029-08-15	768715.700000000004
2029-09-15	771477.750000000005
2029-10-15	773647.500000000005
2029-11-15	775546.500000000005
2029-12-15	776899.750000000005
2030-01-15	779170.050000000005
2030-03-15	781425.500000000005
2030-04-15	784093.000000000005
2030-05-15	786579.950000000004
2030-06-15	788790.900000000004
2030-07-15	791140.050000000004
2030-08-15	792634.650000000004
2030-09-15	794832.700000000004
2030-10-15	797569.300000000004
2030-11-15	799792.550000000004
2030-12-15	801130.350000000004
2031-01-15	803548.200000000004
2031-03-15	806304.800000000004
2031-05-15	808022.450000000004
2031-07-15	810117.850000000004
2031-08-15	812199.200000000004
2031-10-15	814944.050000000004
2031-12-15	817860.050000000004





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

```
select name, revenue
from
(select category, name, revenue,
rank() over(partition by category order by revenue desc ) as revenue_category_name
from
(select pizza_types.category, pizza_types.name,
ROUND(SUM(pizzas.price * order_details.quantity), 2) AS revenue
from pizza_types join pizzas
on pizza_types.pizza_type_id=pizzas.pizza_type_id
join order_details
on order_details.pizza_id=pizzas.pizza_id
group by pizza_types.category, pizza_types.name) as A) as B
where revenue_category_name <= 3;
```



name	revenue
The Thai Chicken Pizza	43434.25
The Barbecue Chicken Pizza	42768
The California Chicken Pizza	41409.5
The Classic Deluxe Pizza	38180.5
The Hawaiian Pizza	32273.25
The Pepperoni Pizza	30161.75
The Spicy Italian Pizza	34831.25
The Italian Supreme Pizza	33476.75
The Sicilian Pizza	30940.5
The Four Cheese Pizza	32265.7
The Mexicana Pizza	26780.75
The Five Cheese Pizza	26066.5



*Pizza hut*  
**THANK**  
**YOU**

