

Where Every Slice is a Taste of Perfection

WELCOME TO PIZZA RESTO

ORDER
NOW

Start Your Slide





ABOUT OUR PIZZA SALES

Our Progress of Pizza Sales

Here we will show our pizza sales and total orders , which pizza is delivered more revenue and how many orders per day average. let us discuss.....

TOTAL ORDERS

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

Result Grid	
name	price
The Greek Pizza	35.95

HIGH COST PIZZA

total_orders
21350

```
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 ;
```



TOTAL REVENUE

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

The screenshot shows a MySQL Workbench result grid. At the top, there are buttons for 'Result Grid' (selected), 'Grid View' (represented by a grid icon), 'Filter' (represented by a funnel icon), and 'Reset'. Below the grid, the column header 'total_sales' is displayed in a bold, multi-colored font. In the data row, the value '356723.85' is shown in a large, multi-colored font. A small navigation arrow is visible on the left side of the grid.

	total_sales
▶	356723.85



AVERAGE NUMBER OF PIZZAS ORDERED PER DAY



Result Grid	
	Filter Rows:
	avg_pizza_ordered_per_day
▶	61

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

CUMULATIVE REVENUE GENERATED OVER TIME

```

select order_date,
       sum(revenue) over(order by
                           order_date) as cum_revenue
  from
    (select orders.order_date,
           sum(order_details.quantity *
                pizzas.price) as revenue
      from order_details join pizzas
        on
          order_details.pizza_id=pizzas.pizza_i
         d
        join orders
          on orders.order_id =
             order_details.order_id
       group by orders.order_date) as
      sales;
  
```

order_date	cum_revenue	order_date	cum_revenue	order_date	cum_revenue	order_date	cum_revenue
2015-01-01	1136.3500000000001	2015-02-23	53740.1	2015-05-24	144220.85	2015-12-05	333884.00000000047
2015-01-02	2245.8500000000004	2015-02-24	54758.9	2015-05-25	145142.55000000002	2015-12-06	334799.6500000005
2015-01-03	3374.8500000000004	2015-02-25	55783.35	2015-05-26	145941.15000000002	2015-12-07	335745.3500000005
2015-01-04	4260.1500000000001	2015-02-26	56656.25	2015-05-27	146816.85000000003	2015-12-08	336586.6000000005
2015-01-05	5159.1	2015-02-27	57841.35	2015-05-28	147725.00000000003	2015-12-09	337637.6500000005
2015-01-06	6232.3	2015-02-28	58846.75	2015-05-29	148852.75000000003	2015-12-10	338674.1000000005
2015-01-07	7195.25	2015-03-01	59636.05	2015-05-30	149878.55000000002	2015-12-11	339883.5000000005
2015-01-08	8352.4	2015-03-02	60584.9	2015-05-31	150643.00000000003	2015-12-12	340801.20000000054
2015-01-09	9386.85	2015-03-03	61466.70000000004	2015-06-01	151831.70000000004	2015-12-13	341731.0000000005
2015-01-10	10501.15	2015-03-04	62467.4	2015-06-02	152815.80000000005	2015-12-14	342698.0000000005
2015-01-11	11341.949999999999	2015-03-05	63546.65	2015-06-03	153694.45000000004	2015-12-15	343678.2500000005
2015-01-12	12223.749999999998	2015-03-06	64701.65	2015-06-04	154607.45000000004	2015-12-16	344839.6000000005
2015-01-13	13031.649999999998	2015-03-07	65706.8	2015-06-05	155768.90000000005	2015-12-17	345818.9000000005
2015-01-14	14148.649999999998	2015-03-08	66708.15000000001	2015-06-06	156814.65000000005	2015-12-18	347029.75000000047
2015-01-15	15126.699999999997	2015-03-09	67642.15000000001	2015-06-07	157714.90000000005	2015-12-19	347993.25000000047
2015-01-16	16257.899999999998	2015-03-10	68646.65000000001	2015-06-08	158685.05000000005	2015-12-20	348896.00000000047
2015-01-17	17215.449999999997	2015-03-11	69673.35	2015-06-09	159791.15000000005	2015-12-21	349795.05000000045
2015-01-18	18064.399999999998	2015-03-12	70570.8	2015-06-10	160718.30000000005	2015-12-22	350761.60000000044
2015-01-19	18961.749999999996	2015-03-13	71702.15000000001	2015-06-11	161871.20000000004	2015-12-23	351710.50000000047
2015-01-20	20071.749999999996	2015-03-14	72659.1	2015-06-12	162940.30000000005	2015-12-24	352606.7000000005
2015-01-21	20859.849999999995	2015-03-15	73464.6	2015-06-13	163971.75000000006	2015-12-26	353262.9500000005
2015-01-22	21960.599999999995	2015-03-16	74559.85	2015-06-14	164799.55000000005	2015-12-27	353807.30000000045
2015-01-23	23103.749999999996	2015-03-17	75721.70000000001	2015-06-15	166087.40000000005	2015-12-28	354447.60000000044
2015-01-24	24114.549999999996	2015-03-18	76765.65000000001	2015-06-16	166984.20000000004	2015-12-29	354929.90000000043
2015-01-25	24786.049999999996	2015-03-19	77741.45000000001	2015-06-17	167909.15000000005	2015-12-30	355480.15000000043
2015-01-26	25728.999999999996	2015-03-20	78816.80000000002	2015-06-18	168861.30000000005	2015-12-31	356723.85000000044

TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE

Result Grid | Filter Rows:

	name	revenue
▶	The Barbecue Chicken Pizza	41230.75
	The California Chicken Pizza	30102.75
	The Classic Deluxe Pizza	23548

```
select pizza_types.name,  
sum(order_details.quantity *  
pizzas.price) as revenue  
from pizza_types join pizzas  
on pizzas.pizza_type_id =  
pizza_types.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;
```

REVENUE FOR EACH PIZZA CATEGORY.

```
select name, revenue from
(select category, name, revenue,rank()
over(partition by category order by revenue desc)
as rn
from
(select pizza_types.category, pizza_types.name,
sum((order_details.quantity) * pizzas.price) 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 rn <= 3;
```

name	revenue
The Barbecue Chicken Pizza	41230.75
The California Chicken Pizza	
The Chicken Alfredo Pizza	11606
The Classic Deluxe Pizza	23548
The Big Meat Pizza	20340
The Hawaiian Pizza	15546.5
The Italian Supreme Pizza	14359.75
The Calabrese Pizza	12463.5
The Brie Carre Pizza	9057.94999999937
The Four Cheese Pizza	17526.550000000232
The Five Cheese Pizza	15299.5
The Mexicana Pizza	9428.5

PIZZA TYPE TO TOTAL REVENUE

```
select pizza_types.category,  
round(sum(order_details.quantity*pizzas.pr  
ice) /(SELECT  
    ROUND(SUM(order_details.quantity *  
pizzas.price),  
    2) AS total_sales  
  
FROM  
    order_details  
        JOIN  
            pizzas ON pizzas.pizza_id =  
order_details.pizza_id) *100,2) 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 order by  
revenue desc;
```

Result Grid | Filter Rows:

	category	revenue
▶	Chicken	30.8
	Classic	28.06
	Supreme	20.63
	Veggie	20.51

Pizza Resto Presentation

THANK YOU
FOR ATTENTION

See You Next

www.reallygreatsite.com