



*Today's special*



# SQL PIZZA SALES PROJECT



# TOTAL REVENUE GENERATED

```
use pizza;

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
```

	Result Grid		
	total_revenue		
▶		817860.05	

# MAX 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;
```

Result Grid | Filter Row

	name	price
▶	The Greek Pizza	35.95

# MOST COMMON SIZED PIZZA ORDERED

```
SELECT
    pizzas.size, COUNT(order_details.order_details_id) as order_count
FROM
    pizzas
        JOIN
    order_details ON pizzas.pizza_id = order_details.pizza_id
GROUP BY pizzas.size
ORDER BY order_count DESC
LIMIT 1;
```

Result Grid				
	size	order_count		
▶	L	18526		

# REVENUE PER PIZZA TYPE

```
SELECT
    pizza_types.category,
    ROUND(SUM(order_details.quantity * pizzas.price),
        0) 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

	category	revenue
▶	Classic	220053
	Supreme	208197
	Chicken	195920
	Veggie	193690

# TOP 3 BEST SELLING PIZZAS WITH THEIR QUANTITIES

```
SELECT
    pizza_types.name,
    SUM(order_details.quantity) AS quantity,
    ROUND(SUM(order_details.quantity * pizzas.price),
        0) 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	quantity	revenue
▶	The Thai Chicken Pizza	2371	43434
	The Barbecue Chicken Pizza	2432	42768
	The California Chicken Pizza	2370	41410

# ORDERS WITH THE HIGHEST QUANTITY OF SINGLE PIZZA

SELECT

```
    order_details.order_id,  
    pizza_types.name,  
    MAX(order_details.quantity) AS quantity  
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 order_details.order_id , pizza_types.name  
ORDER BY quantity DESC limit 5;
```

	order_id	name	quantity
▶	15674	The California Chicken Pizza	4
	8286	The Big Meat Pizza	4
	5264	The Big Meat Pizza	4
	7404	The Big Meat Pizza	3
	8040	The Thai Chicken Pizza	3

# EXTRACT BUSIEST HOUR OF THE DAY

**SELECT**

```
HOUR(order_time), COUNT(order_id) as total_orders
FROM
    orders
GROUP BY HOUR(order_time)
order by total_orders desc;
```

	HOUR(order_time)	total_orders
▶	12	2520
	13	2455
	18	2399
	17	2336
	19	2009
	16	1920
	20	1642

# PERCENTAGE CACULATION ON EACH PIZZA TYPE TO TOTAL REVENUE

```
SELECT
    pizza_types.category,
    round(SUM(order_details.quantity * pizzas.price) / (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) * 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 |

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

# EXTRACT 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_id  
join orders  
on orders.order_id = order_details.order_id  
group by orders.order_date) as sales
```

Result Grid		
	order_date	cum_revenue
▶	2015-01-01	2713.8500000000004
	2015-01-02	5445.75
	2015-01-03	8108.15
	2015-01-04	9863.6
	2015-01-05	11929.55

# TOP 3 MOST ORDERED PIZZA TYPES BASED ON REVENUE FOR EACH PIZZA CATEGORY

```
SELECT category, name, revenue,  
RANK() OVER(PARTITION BY category ORDER BY revenue DESC) AS rankings  
FROM  
(SELECT pizza_types.category, pizza_types.name,  
ROUND(SUM(order_details.quantity*pizzas.price),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;
```

	category	name	revenue	rankings
▶	Chicken	The Thai Chicken Pizza	43434.25	1
	Chicken	The Barbecue Chicken Pizza	42768	2
	Chicken	The California Chicken Pizza	41409.5	3
	Chicken	The Southwest Chicken Pizza	34705.75	4
	Chicken	The Chicken Alfredo Pizza	16900.25	5

# FIND MOST COMMON INGREDIENT COMBINATIONS

```
SELECT
    pizza_types.ingredients,
    COUNT(order_details.pizza_id) AS frequency
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    pizza_types ON pizzas.pizza_type_id = pizza_types.pizza_type_id
GROUP BY pizza_types.ingredients
ORDER BY frequency DESC
LIMIT 5;
```

Result Grid		Filter Rows:	Export:
	ingredients	frequency	
▶	Pepperoni, Mushrooms, Red Onions, Red Pepp...	2416	
	Barbecued Chicken, Red Peppers, Green Pepp...	2372	
	Sliced Ham, Pineapple, Mozzarella Cheese	2370	
	Mozzarella Cheese, Pepperoni	2369	
	Chicken, Pineapple, Tomatoes, Red Peppers, T...	2315	

# FIND MOST EXPENSIVE ORDERS AND ON WHICH DAY

```
SELECT
    orders.order_date,
    orders.order_id,
    SUM(order_details.quantity * pizzas.price) AS total_cost
FROM
    order_details
        JOIN
    pizzas ON order_details.pizza_id = pizzas.pizza_id
        JOIN
    orders ON order_details.order_id = orders.order_id
GROUP BY orders.order_date,orders.order_id
ORDER BY total_cost DESC
LIMIT 5;
```

	order_date	order_id	total_cost
▶	2015-11-18	18845	444.2
	2015-06-30	10760	417.15
	2015-01-19	1096	285.15
	2015-04-14	6169	284
	2015-01-13	740	280.95

# FIND DAILY REVENUE TREND WITH QUANTITIES

SELECT

```
    orders.order_date,  
    SUM(order_details.quantity) AS quantity,  
    ROUND(SUM(order_details.quantity * pizzas.price),  
        0) AS total_revenue
```

FROM

```
    order_details  
        JOIN
```

```
    pizzas ON order_details.pizza_id = pizzas.pizza_id  
        JOIN
```

```
    orders ON order_details.order_id = orders.order_id
```

GROUP BY orders.order\_date , quantity

ORDER BY total\_revenue desc;

	order_date	quantity	total_revenue
▶	2015-11-27	255	4275
	2015-11-26	256	4226
	2015-10-15	254	4195
	2015-07-04	232	3840
	2015-05-15	203	3300

# BASKET ANALYSIS: FREQUENTLY ORDERED PIZZA PAIRS

```
SELECT
    od1.pizza_id AS pizza1,
    od2.pizza_id AS pizza2,
    COUNT(*) AS orders
FROM
    order_details od1
        JOIN
    order_details od2 ON od1.order_id = od2.order_id
        AND od1.pizza_id < od2.pizza_id
GROUP BY pizza1 , pizza2
ORDER BY orders DESC
LIMIT 10;
```

	pizza1	pizza2	orders
▶	big_meat_s	thai_dkn_l	144
	big_meat_s	four_cheese_l	126
	big_meat_s	five_cheese_l	125
	big_meat_s	classic_dlx_m	105
	big_meat_s	spicy_ital_l	104

# FIND THE LEAST POPULAR PIZZAS

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

	pizza_id	name	total_sold
▶	the_greek_xxl	The Greek Pizza	28
	green_garden_l	The Green Garden Pizza	95
	chn_alfredo_s	The Chicken Alfredo Pizza	96
	calabrese_s	The Calabrese Pizza	99
	mexicana_s	The Mexicana Pizza	162



# GITHUB:

