



```
1  -- Identify the most common pizza size ordered.
2  •  SELECT PIZZAS.SIZE, COUNT(ORDER_DETAILS.ORDER_DETAILS_ID) AS ORDER_COUNT
3     FROM PIZZAS JOIN ORDER_DETAILS
4     ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID
5     GROUP BY PIZZAS.SIZE ORDER BY ORDER_COUNT DESC;
6
7
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	SIZE	ORDER_COUNT
▶	L	18526
	M	15385
	S	14137
	XL	544
	XXL	28



```
1  -- List the top 5 most ordered pizza types along with their quantities.
2  • SELECT PIZZA_TYPES.NAME,
3      SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY
4  FROM PIZZA_TYPES JOIN PIZZAS
5  ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
6  JOIN ORDER_DETAILS
7  ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
8  GROUP BY PIZZA_TYPES.NAME ORDER BY QUANTITY DESC LIMIT 5;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



Fetch rows:

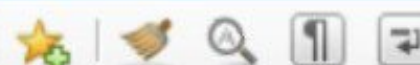


	NAME	QUANTITY
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418
	The Thai Chicken Pizza	2371

Result 6 x



Limit to 1000 rows



```
1  -- Join the necessary tables to find the total quantity of each pizza category ordered.
2  • SELECT PIZZA_TYPES.CATEGORY,
3      SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY
4  FROM PIZZA_TYPES JOIN PIZZAS
5  ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
6  JOIN ORDER_DETAILS
7  ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
8  GROUP BY PIZZA_TYPES.CATEGORY ORDER BY QUANTITY DESC;
```

Result Grid



Filter Rows:

Export:

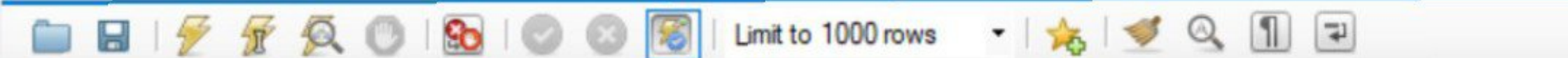


Wrap Cell Content:



	CATEGORY	QUANTITY
▶	Classic	14888
	Supreme	11987
	Veggie	11649
	Chicken	11050

Result 2 x



```

1  -- Determine the distribution of orders by hour of the day.
2
3  • SELECT HOUR(ORDER_TIME) AS HOUR, COUNT(ORDER_ID) AS ORDER_COUNT FROM ORDERS
4  GROUP BY HOUR(ORDER_TIME)
    
```

Result Grid Filter Rows: Export: Wrap Cell Content:

	HOUR	ORDER_COUNT
	18	2399
	19	2009
	20	1642
	21	1198
▶	22	663
	23	28
	10	8
	9	1



Limit to 1000 rows

```
1  -- Join relevant tables to find the category-wise distribution of pizzas.  
2  •  SELECT CATEGORY, COUNT(NAME) FROM PIZZA_TYPES  
3  GROUP BY CATEGORY
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	CATEGORY	COUNT(NAME)
▶	Chicken	6
	Classic	8
	Supreme	9
	Veggie	9



```
1  -- Group the orders by date and calculate the average number of pizzas ordered per da
2
3  •  SELECT ROUND(AVG(QUANTITY),0) AS AVG_PIZZA_ORDERED_PER_DAY
4  FROM
5  (SELECT ORDERS.ORDER_DATE, SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY
6   FROM ORDERS JOIN ORDER_DETAILS
7   ON ORDERS.ORDER_ID = ORDER_DETAILS.ORDER_ID
8   GROUP BY ORDERS.ORDER_DATE) AS ORDER_QUANTITY;
9
```




```
1  -- Determine the top 3 most ordered pizza types based on revenue.
2  •  SELECT PIZZA_TYPES.NAME,
3     SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE) AS REVENUE
4  FROM PIZZA_TYPES JOIN PIZZAS
5     ON PIZZAS.PIZZA_TYPE_ID = PIZZA_TYPES.PIZZA_TYPE_ID
6     JOIN ORDER_DETAILS
7     ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
8     GROUP BY PIZZA_TYPES.NAME ORDER BY REVENUE DESC LIMIT 3;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



Fetch rows:



	NAME	REVENUE
▶	The Thai Chicken Pizza	43434.25
	The Barbecue Chicken Pizza	42768
	The California Chicken Pizza	41409.5



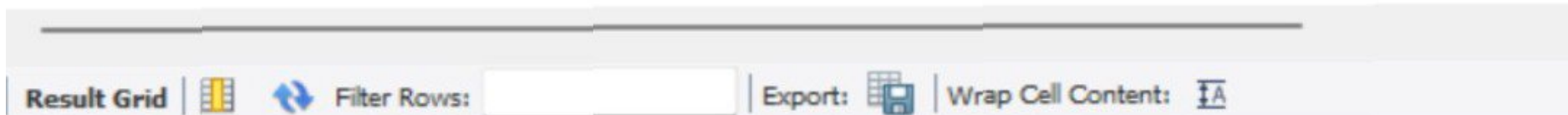
Limit to 1000 rows



```
1  -- Calculate the percentage contribution of each pizza type to total revenue.
2
3  •  SELECT PIZZA_TYPES.CATEGORY,
4  ROUND((SUM(ORDER_DETAILS.QUANTITY*PIZZAS.PRICE) / (SELECT
5  ROUND(SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE),
6  2) AS TOTAL_SALES
7  FROM
8  ORDER_DETAILS
9  JOIN
10  PIZZAS ON PIZZAS.PIZZA_ID = ORDER_DETAILS.PIZZA_ID)) * 100,2) AS REVENUE
11  FROM PIZZA_TYPES JOIN PIZZAS
12  ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
13  JOIN ORDER_DETAILS
14  ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
15  GROUP BY PIZZA_TYPES.CATEGORY ORDER BY REVENUE DESC;
```




```
1  -- Analyze the cumulative revenue generated over time.
2  •  SELECT ORDER_DATE,
3     SUM(REVENUE) OVER(ORDER BY ORDER_DATE) AS CUM_REVENUE
4  FROM
5  (SELECT ORDERS.ORDER_DATE,
6     SUM(ORDER_DETAILS.QUANTITY * PIZZAS.PRICE) AS REVENUE
7   FROM ORDER_DETAILS JOIN PIZZAS
8   ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
9   JOIN ORDERS
10  ON ORDERS.ORDER_ID = ORDER_DETAILS.ORDER_ID
11  GROUP BY ORDERS.ORDER_DATE) AS SALES;
12
```



	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
	2015-01-06	14358.5
	2015-01-07	16560.7
	2015-01-08	18770.05



```
1  -- Determine the top 3 most ordered pizza types based on revenue for each pizza category.
2  ●  SELECT NAME, REVENUE FROM
3  (SELECT CATEGORY, NAME, REVENUE,
4  RANK() OVER(PARTITION BY CATEGORY ORDER BY REVENUE DESC) AS RN
5  FROM
6  (SELECT PIZZA_TYPES.CATEGORY, PIZZA_TYPES.NAME,
7  SUM((ORDER_DETAILS.QUANTITY) * PIZZAS.PRICE) AS REVENUE
8  FROM PIZZA_TYPES JOIN PIZZAS
9  ON PIZZA_TYPES.PIZZA_TYPE_ID = PIZZAS.PIZZA_TYPE_ID
10 JOIN ORDER_DETAILS
11 ON ORDER_DETAILS.PIZZA_ID = PIZZAS.PIZZA_ID
12 GROUP BY PIZZA_TYPES.CATEGORY, PIZZA_TYPES.NAME) AS A) AS B
13 WHERE RN<=3;
```

Result Grid



Filter Rows:

Export:



Wrap Cell Content:



	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	32476.75

Result 3