

ps910243 Add files via upload

37d5bf0 · now History

Code

Blame





19 lines (16 loc) · 857 Bytes



Raw



```
1  Basic:
2  Retrieve the total number of orders placed.
3  Calculate the total revenue generated from pizza sales.
4  Identify the highest-priced pizza.
5  Identify the most common pizza size ordered.
6  List the top 5 most ordered pizza types along with their quantities.
7
8
9  Intermediate:
10 Join the necessary tables to find the total quantity of each pizza category ordered.
11 Determine the distribution of orders by hour of the day.
12 Join relevant tables to find the category-wise distribution of pizzas.
13 Group the orders by date and calculate the average number of pizzas ordered per day.
14 Determine the top 3 most ordered pizza types based on revenue.
15
16 Advanced:
17 Calculate the percentage contribution of each pizza type to total revenue.
18 Analyze the cumulative revenue generated over time.
19 Determine the top 3 most ordered pizza types based on revenue for each pizza category.
```

Name	Date modified	Type	Size
 order_details	29-08-2025 18:31	Microsoft Excel Co...	1,278 KB
 orders	29-08-2025 18:31	Microsoft Excel Co...	553 KB
 pizza_types	29-08-2025 18:31	Microsoft Excel Co...	4 KB
 pizzas	29-08-2025 18:31	Microsoft Excel Co...	4 KB



```
1  -- Retrieve the total number of orders placed.
2 • SELECT count(order_id) FROM pizzahut.orders;
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid | Filter Rows: | Export: Wrap Cell Content:

	count(order_id)
▶	21350



Result 1 x

Read Only Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓ 51	22:05:43	SELECT * FROM pizzahut.orders	21350 row(s) returned	0.000 sec / 0.000 sec
✓ 52	22:05:45	SELECT * FROM pizzahut.pizza_types	32 row(s) returned	0.000 sec / 0.000 sec
✓ 53	22:05:47	SELECT * FROM pizzahut.pizzas	96 row(s) returned	0.000 sec / 0.000 sec
✓ 54	22:08:51	SELECT count(order_id) FROM pizzahut.orders	1 row(s) returned	0.000 sec / 0.000 sec
✓ 55	22:09:01	SELECT count(order_id) FROM pizzahut.orders	1 row(s) returned	0.000 sec / 0.000 sec
✓ 56	22:10:25	SELECT count(order_id) FROM pizzahut.orders	1 row(s) returned	0.000 sec / 0.000 sec

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

1  -- Calculate the total revenue generated from pizza sales.
2  • select
3  round(sum(order_details.quantity * pizzas.price),2) as total_sales
4  from order_details join pizzas
5  on pizzas.pizza_id = order_details.pizza_id
6

```

Result Grid   Filter Rows:   Export:   Wrap Cell Content:

total_sales
817860.05



Result 4 x   Read Only   Context Help   Snippets

Output

#	Time	Action	Message	Duration / Fetch
61	22:27:27	select (order_details.quantity * pizzas.price) from order_details add pizza_id on pizzas.pizza_id = order_details.p...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
62	22:27:53	select (order_details.quantity * pizzas.price) from order_details join pizza_id on pizzas.pizza_id = order_details.pi...	Error Code: 1146. Table 'pizzahut.pizza_id' doesn't exist	0.000 sec
63	22:28:21	select (order_details.quantity * pizzas.price) from order_details join pizzas on pizzas.pizza_id = order_details.pizz...	48620 row(s) returned	0.047 sec / 0.046 sec
64	22:29:07	select (order_details.quantity * pizzas.price) as total_sales from order_details join pizzas on pizzas.pizza_id = ord...	48620 row(s) returned	0.000 sec / 0.078 sec
65	22:29:25	select sum(order_details.quantity * pizzas.price) as total_sales from order_details join pizzas on pizzas.pizza_id = ...	1 row(s) returned	0.047 sec / 0.000 sec
66	22:30:15	select round(sum(order_details.quantity * pizzas.price),2) as total_sales from order_details join pizzas on pizzas....	1 row(s) returned	0.062 sec / 0.000 sec

```
1  -- Retrieve the total number of orders placed.  
2  
3 • select count(order_id) as total_orders from orders;
```

I

---

Result Grid		 Filter Rows: _____	Export: 	Wrap Cell Content: 
	total_orders			
▶	21350			



```
round(sum(order_details.quantity * pizzas.price), 2) as total_sales  
from order_details join pizzas  
on pizzas.pizza_id = order_details.pizza_id
```



1 -- Identify the highest-priced pizza.

2

3 • **select** pizza\_types.name, pizzas.price

4 **from** pizza\_types **join** pizzas

5 **on** pizza\_types.pizza\_type\_id = pizzas.pizza\_type\_id

6 **order by** pizzas.price **desc limit** 1;

7



```
1  -- Identify the most common pizza size ordered.  
2  
3  select pizzas.size, count(order_details.order_de  
4  from pizzas join order_details  
5  on pizzas.pizza_id = order_details.pizza_id  
6  group by pizzas.size order by order_count desc ;  
7
```

	size	order_count
▶	L	18526





Don't Limit

```
1  -- Identify the most common pizza size ordered.
2
3  select pizzas.size, count(order_details.order_d
4  from pizzas join order_details
5  on pizzas.pizza_id = order_details.pizza_id
6  group by pizzas.size order by order_count desc
7
```

```

4 • SELECT
5     pizza_types.name, SUM(order_details.quantity) AS quantity
6 FROM
7     pizza_types
8     JOIN
9     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10    JOIN
11    order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.name
13 ORDER BY quantity DESC
14 LIMIT 5;

```

Result Grid |  |  Filter Rows: \_\_\_\_\_ | Export:  | Wrap Cell Content: 

	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418

2 -- total quantity of each pizza category ordered.

3

4 • **SELECT**

5 pizza\_types.category,

6 SUM(order\_details.quantity) **AS** quantity

7 **FROM**

8 pizza\_types

9 **JOIN**

10 pizzas **ON** pizza\_types.pizza\_type\_id = pizzas.pizza\_type\_id

11 **JOIN**

12 order\_details **ON** order\_details.pizza\_id = pizzas.pizza\_id

13 **GROUP BY** pizza\_types.category

14 **ORDER BY** quantity **DESC**; I

15



3 • **SELECT**

4       **HOUR**(order\_time) **AS** hour, COUNT(order\_id) **AS** order\_count

5 **FROM**

6       orders

7 **GROUP BY HOUR**(order\_time);

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472





```
1  -- Join relevant tables to find the
2  -- category-wise distribution of pizzas.
3
4  select category , count(name) from pizza_types
5  group by category;
```

---

Result Grid   Filter Rows: \_\_\_\_\_ | Export:  | Wrap Cell Content: 

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





ps910243 Add files via upload

37d5bf0 · now

History

Code

Blame

19 lines (16 loc) · 857 Bytes







Raw



```
1 Basic:
2 Retrieve the total number of orders placed.
3 Calculate the total revenue generated from pizza sales.
4 Identify the highest-priced pizza.
5 Identify the most common pizza size ordered.
6 List the top 5 most ordered pizza types along with their quantities.
7
8
9 Intermediate:
10 Join the necessary tables to find the total quantity of each pizza category ordered.
11 Determine the distribution of orders by hour of the day.
12 Join relevant tables to find the category-wise distribution of pizzas.
13 Group the orders by date and calculate the average number of pizzas ordered per day.
14 Determine the top 3 most ordered pizza types based on revenue.
15
16 Advanced:
17 Calculate the percentage contribution of each pizza type to total revenue.
18 Analyze the cumulative revenue generated over time.
19 Determine the top 3 most ordered pizza types based on revenue for each pizza category.
```

ENG  
IN10:44  
31-08-2025

Name	Date modified	Type	Size
 order_details	29-08-2025 18:31	Microsoft Excel Co...	1,278 KB
 orders	29-08-2025 18:31	Microsoft Excel Co...	553 KB
 pizza_types	29-08-2025 18:31	Microsoft Excel Co...	4 KB
 pizzas	29-08-2025 18:31	Microsoft Excel Co...	4 KB

```
1  -- Retrieve the total number of orders placed.
2 • SELECT count(order_id) FROM pizzahut.orders;
```

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid   Filter Rows: \_\_\_\_\_ | Export:  | Wrap Cell Content: 

count(order_id)
21350

Read Only Context Help Snippets

**Output** :::::

Action Output

#	Time	Action	Message	Duration / Fetch
51	22:05:43	SELECT * FROM pizzahut.orders	21350 row(s) returned	0.000 sec / 0.000 sec
52	22:05:45	SELECT * FROM pizzahut.pizza_types	32 row(s) returned	0.000 sec / 0.000 sec
53	22:05:47	SELECT * FROM pizzahut.pizzas	96 row(s) returned	0.000 sec / 0.000 sec
54	22:08:51	SELECT count(order_id) FROM pizzahut.orders	1 row(s) returned	0.000 sec / 0.000 sec
55	22:09:01	SELECT count(order_id) FROM pizzahut.orders	1 row(s) returned	0.000 sec / 0.000 sec
56	22:10:25	SELECT count(order_id) FROM pizzahut.orders	1 row(s) returned	0.000 sec / 0.000 sec

MySQL Workbench

Project x

File Edit View Query Database Server Tools Scripting Help

SQL SQL

Navigator

Query 1 order\_details orders pizza\_types pizzas SQL File 4\* SQL File 5\*

SCHEMAS

Filter objects

pizzahut

Tables

order\_details

orders

pizza\_types

pizzas

Views

Stored Procedures

Functions

sakila

sys

world

xyz

xyz\_company

1 -- Calculate the total revenue generated from pizza sales.

2 • select

3 round(sum(order\_details.quantity \* pizzas.price),2) as total\_sales

4 from order\_details join pizzas

5 on pizzas.pizza\_id = order\_details.pizza\_id

6

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Result Grid

Filter Rows: Export: Wrap Cell Content: I A

total_sales
817860.05

Administration Schemas

Information

Table: order\_details

Columns:

order_details_id	int
order_id	PK
pizza_id	int
quantity	text

Object Info Session

Result 4 x

Read Only Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
61	22:27:27	select (order_details.quantity * pizzas.price) from order_details add pizza_id on pizzas.pizza_id = order_details.p...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL se...	0.000 sec
62	22:27:53	select (order_details.quantity * pizzas.price) from order_details join pizza_id on pizzas.pizza_id = order_details.pi...	Error Code: 1146. Table 'pizzahut.pizza_id' doesn't exist	0.000 sec
63	22:28:21	select (order_details.quantity * pizzas.price) from order_details join pizzas on pizzas.pizza_id = order_details.pizz...	48620 row(s) returned	0.047 sec / 0.046 sec
64	22:29:07	select (order_details.quantity * pizzas.price) as total_sales from order_details join pizzas on pizzas.pizza_id = ord...	48620 row(s) returned	0.000 sec / 0.078 sec
65	22:29:25	select sum(order_details.quantity * pizzas.price) as total_sales from order_details join pizzas on pizzas.pizza_id =...	1 row(s) returned	0.047 sec / 0.000 sec
66	22:30:15	select round(sum(order_details.quantity * pizzas.price),2) as total_sales from order_details join pizzas on pizzas....	1 row(s) returned	0.062 sec / 0.000 sec

Windows Taskbar

22:30 2025-08-08





1 -- Retrieve the total number of orders placed.

2

3 • **select** count(**order\_id**) **as** total\_orders **from** orders;

I

Result Grid Filter Rows:  Export: Wrap Cell Content:

	total_orders
▶	21350



```
1  -- Calculate the total revenue generated from pizza sales.
```

```
2
```

```
3  select
```

```
4  round(sum(order_details.quantity * pizzas.price),2) as total_sales
```

```
5  from order_details join pizzas
```

```
6  on pizzas.pizza_id = order_details.pizza_id
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	total_sales
▶	817860.05





Navigator

SCHEMAS

Filter objects

- ▶ classicmodels
- ▶ demo
- ▼ pizzahut
  - ▶ Tables
    - ▶ order\_details
    - ▶ orders
    - ▶ pizza\_types
    - ▶ pizzas
  - ▶ Views
  - ▶ Stored Procedures
  - ▶ Functions
- ▶ sakila
- ▶ students
- ▶ sys
- ▶ world

Query 1

SQL File 2\*

SQL File 3\*

SQL File 4\* x



Don't Limit

```
1  -- Identify the highest-priced pizza.
2
3  select pizza_types.name, pizzas.price
4  from pizza_types join pizzas
5  on pizza_types.pizza_type_id = pizzas.pizza_type_id
6  order by pizzas.price desc limit 1;
7
```

Administration Schemas

Information

Result Grid Filter Rows:  Export: Wrap Cell Content: 

	name	price
▶	The Greek Pizza	35.95
	The Greek Pizza	

Table: order\_details



```
1  -- Identify the most common pizza size ordered.
2
3  select pizzas.size, count(order_details.order_de
4  from pizzas join order_details
5  on pizzas.pizza_id = order_details.pizza_id
6  group by pizzas.size order by order_count desc ;
7
```

	size	order_count
▶	L	18526



```
1  -- Identify the most common pizza size ordered.
2
3  select pizzas.size, count(order_details.order_id) as order_count
4  from pizzas join order_details
5  on pizzas.pizza_id = order_details.pizza_id
6  group by pizzas.size order by order_count desc
7
```

	size	order_count
▶	L	18526



```
4 • SELECT
5     pizza_types.name, SUM(order_details.quantity) AS quantity
6 FROM
7     pizza_types
8     JOIN
9     pizzas ON pizza_types.pizza_type_id = pizzas.pizza_type_id
10    JOIN
11    order_details ON order_details.pizza_id = pizzas.pizza_id
12 GROUP BY pizza_types.name
13 ORDER BY quantity DESC
14 LIMIT 5;
```

---

Result Grid |   Filter Rows: \_\_\_\_\_ | Export:  | Wrap Cell Content: 

	name	quantity
▶	The Classic Deluxe Pizza	2453
	The Barbecue Chicken Pizza	2432
	The Hawaiian Pizza	2422
	The Pepperoni Pizza	2418



2 -- total quantity of each pizza category ordered.

3

4 • **SELECT**

5 pizza\_types.category,

6 SUM(order\_details.quantity) **AS** quantity

7 **FROM**

8 pizza\_types

9 **JOIN**

10 pizzas **ON** pizza\_types.pizza\_type\_id = pizzas.pizza\_type\_id

11 **JOIN**

12 order\_details **ON** order\_details.pizza\_id = pizzas.pizza\_id

13 **GROUP BY** pizza\_types.category

14 **ORDER BY** quantity **DESC**; I

15



3 • **SELECT**

4       **HOUR**(order\_time) **AS** hour, COUNT(order\_id) **AS** order\_count

5 **FROM**

6       orders

7 **GROUP BY HOUR**(order\_time);|

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

	hour	order_count
▶	11	1231
	12	2520
	13	2455
	14	1472



```
1  -- Join relevant tables to find the
2  -- category-wise distribution of pizzas.
3
4  select category , count(name) from pizza_types
5  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
2  -- number of pizzas ordered per day.
3
4• select avg(quantity) from
5  (select orders.order_date, sum(order_details.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 ;
```

	avg(quantity)
▶	138.4749



```
1  -- Determine the top 3 most ordered pizza types based on  
2  
3  select pizza_types.name,  
4  sum(order_details.quantity * pizzas.price) as revenue  
5  from pizza_types join pizzas  
6  on pizzas.pizza_type_id = pizza_types.pizza_type_id  
7  join order_details  
8  on order_details.pizza_id = pizzas.pizza_id  
9  group by pizza_types.name order by revenue desc limit 3;  
10
```

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



```
3 • select order_date,
4   sum(revenue) over(order by order_date) as cum_revenue
5 from
6 (select orders.order_date,
7   sum(order_details.quantity * pizzas.price) as revenue
8 from order_details join pizzas
9 on order_details.pizza_id = pizzas.pizza_id
10 join orders
11 on orders.order_id = order_details.order_id
12 group by orders.order_date) as sales;
```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	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	14752.5



Don't Limit



```
3
4 • select pizza_types.category,
5 round(sum(order_details.quantity*pizzas.price) / (SELECT
6 ROUND(SUM(order_details.quantity * pizzas.price),
7 2) AS total_sales
8 FROM
9 order_details
10 JOIN
11 pizzas ON pizzas.pizza_id = order_details.pizza_id) *100,2) as revenue
12 from pizza_types join pizzas
13 on pizza_types.pizza_type_id = pizzas.pizza_type_id
14 join order_details
15 on order_details.pizza_id = pizzas.pizza_id
16 group by pizza_types.category order by revenue desc;
```

Result Grid Filter Rows: Export: Wrap Cell Content:

	category	revenue
▶	Classic	26.91
	Supreme	25.46
	Chicken	23.96
	Veggie	23.68

```
1  -- Determine the top 3 most ordered pizza types based on  
2  
3  select pizza_types.name,  
4  sum(order_details.quantity * pizzas.price) as revenue  
5  from pizza_types join pizzas  
6  on pizzas.pizza_type_id = pizza_types.pizza_type_id  
7  join order_details  
8  on order_details.pizza_id = pizzas.pizza_id  
9  group by pizza_types.name order by revenue desc limit 3;  
10
```

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



```
3 • select order_date,
4   sum(revenue) over(order by order_date) as cum_revenue
5 from
6 (select orders.order_date,
7   sum(order_details.quantity * pizzas.price) as revenue
8 from order_details join pizzas
9 on order_details.pizza_id = pizzas.pizza_id
10 join orders
11 on orders.order_id = order_details.order_id
12 group by orders.order_date) as sales;
```

Result Grid   Filter Rows:  Export:  Wrap Cell Content: 

	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	14752.5



```
select pizza_types.category,  
round(sum(order_details.quantity*pizzas.price) / (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;
```