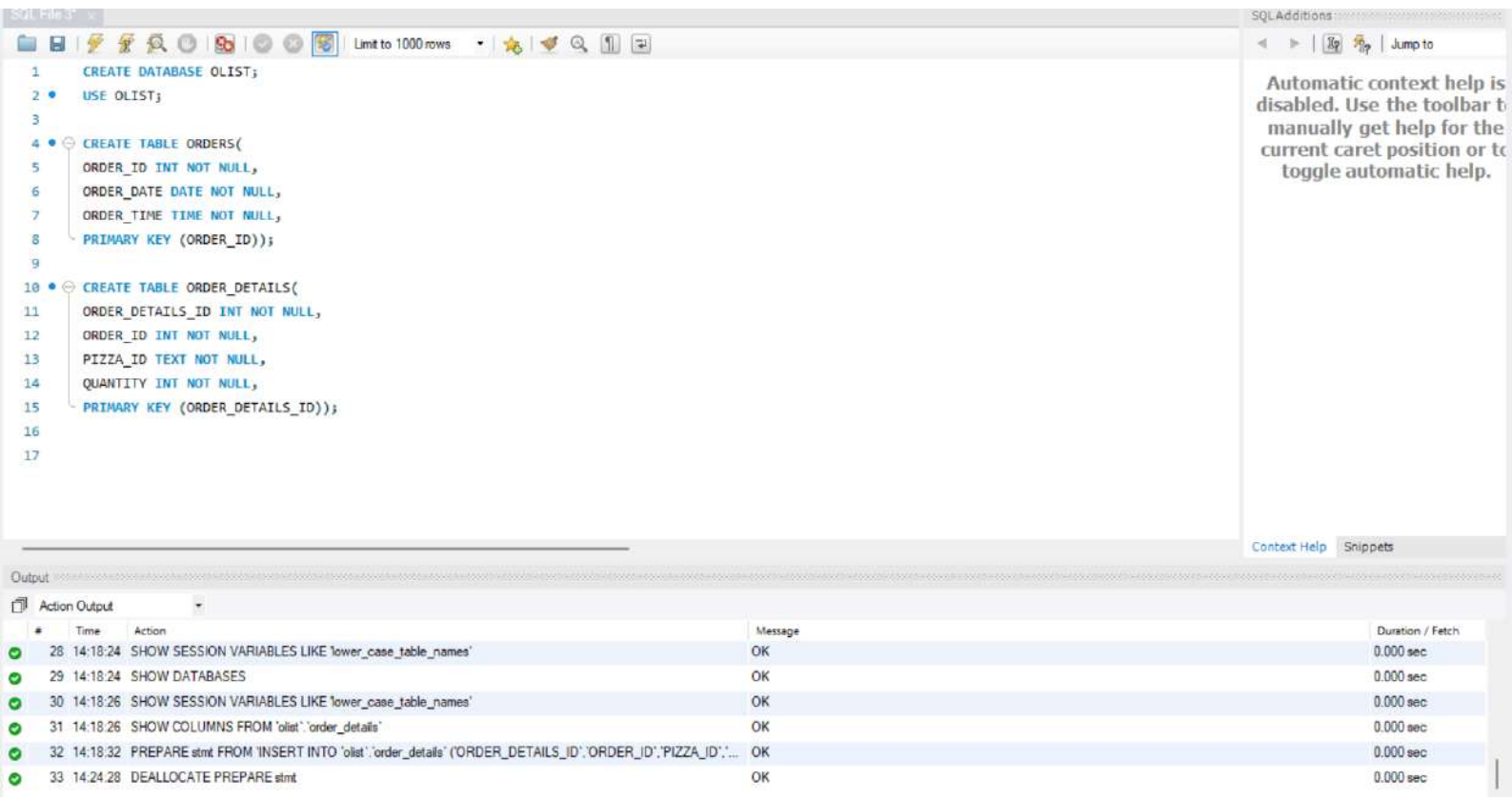
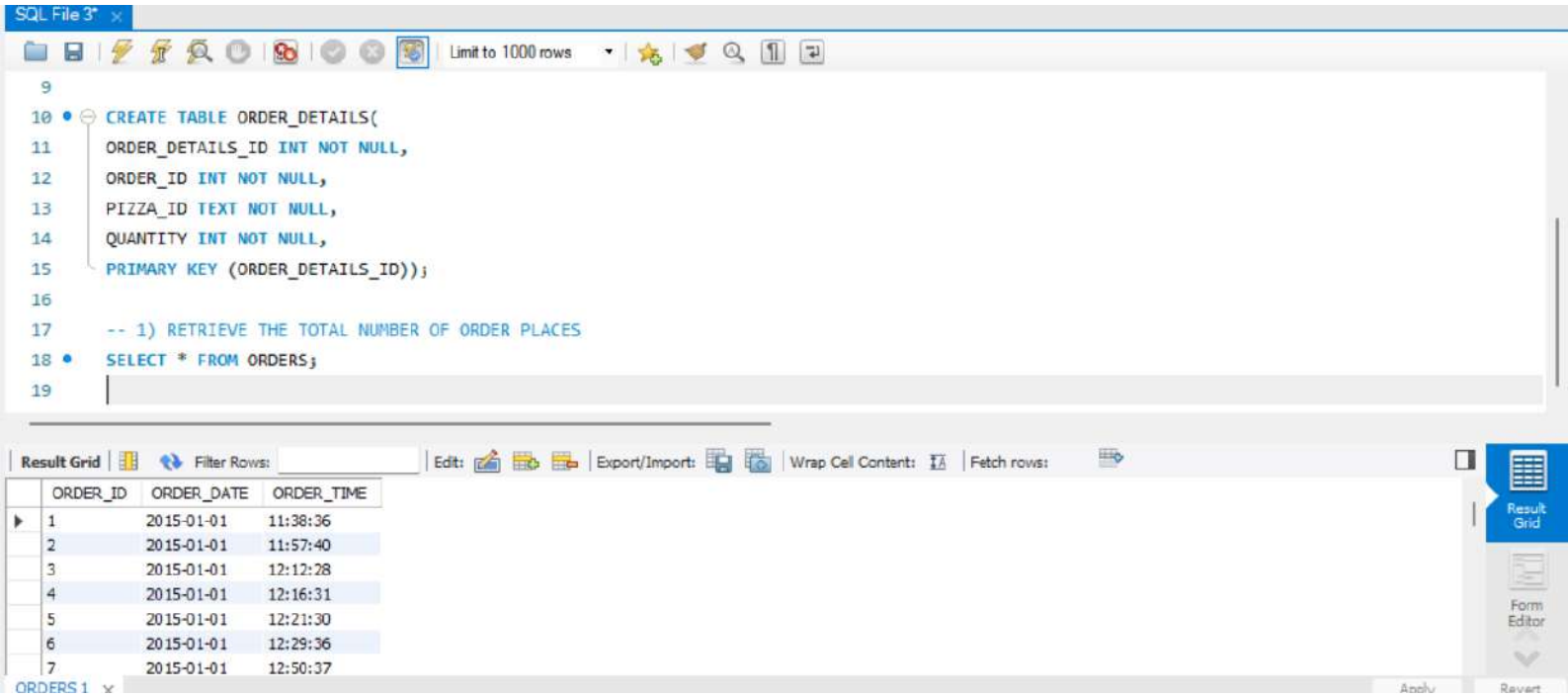


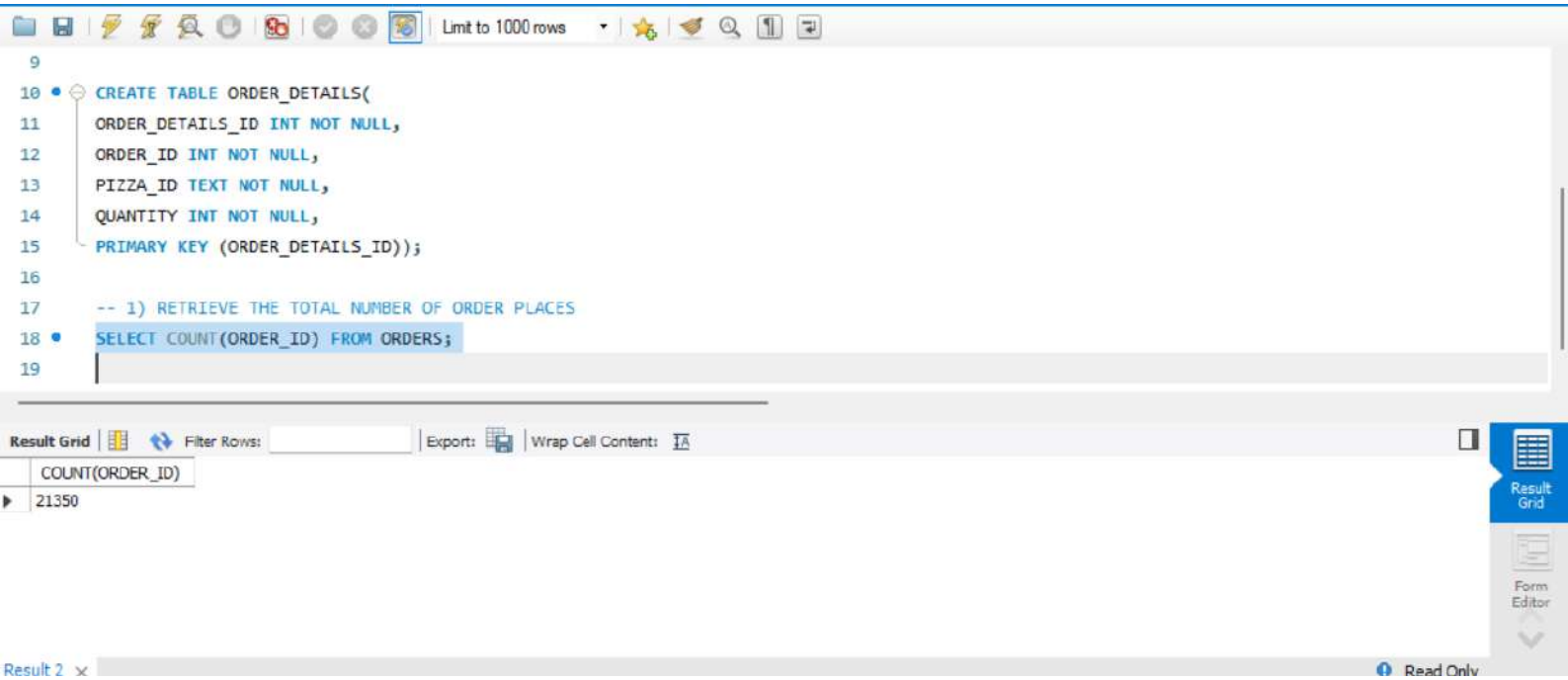

```
1   CREATE DATABASE OLIST;
2   USE OLIST;
3
4   CREATE TABLE ORDERS(
5   ORDER_ID INT NOT NULL,
6   ORDER_DATE DATE NOT NULL,
7   ORDER_TIME TIME NOT NULL,
8   PRIMARY KEY (ORDER_ID));
9
10  CREATE TABLE ORDER_DETAILS(
11  ORDER_DETAILS_ID INT NOT NULL,
12  ORDER_ID INT NOT NULL,
13  PIZZA_ID TEXT NOT NULL,
14  QUANTITY INT NOT NULL,
15  PRIMARY KEY (ORDER_DETAILS_ID));
16
17
```

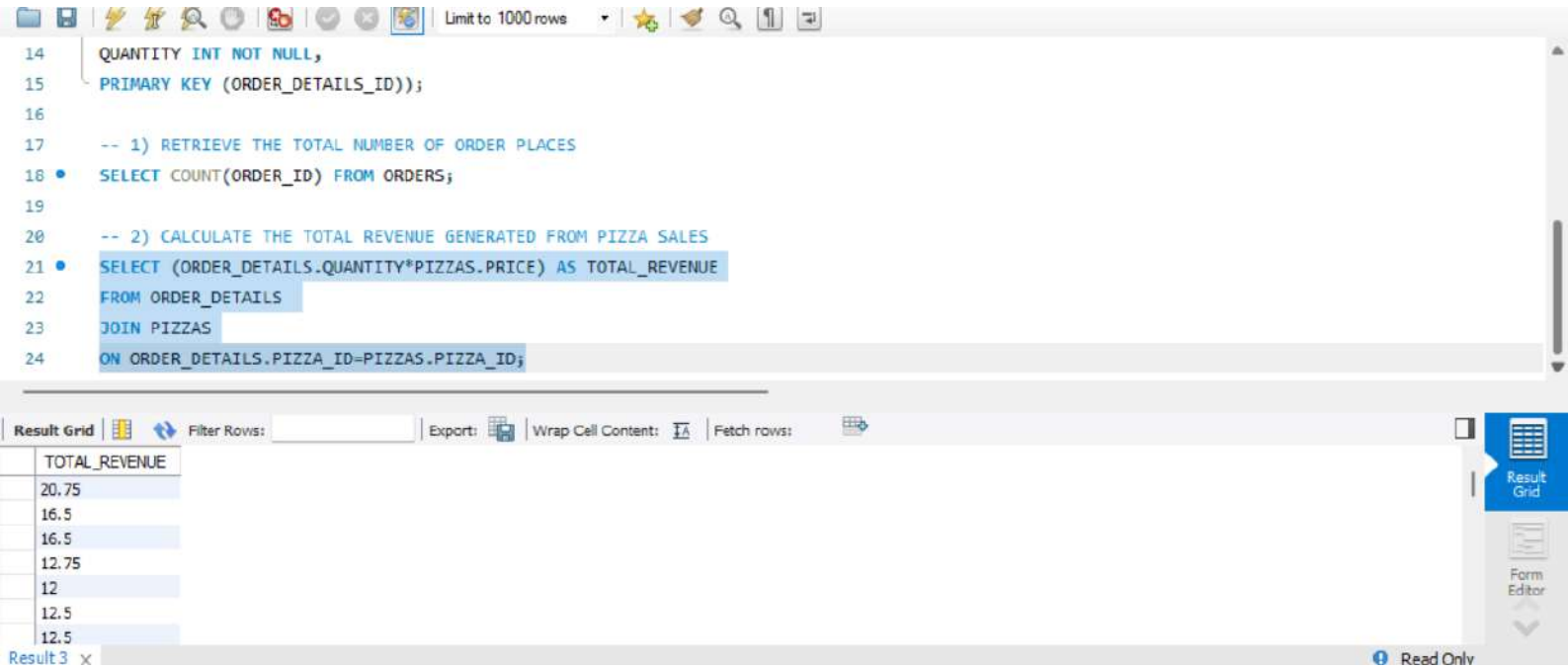
Output

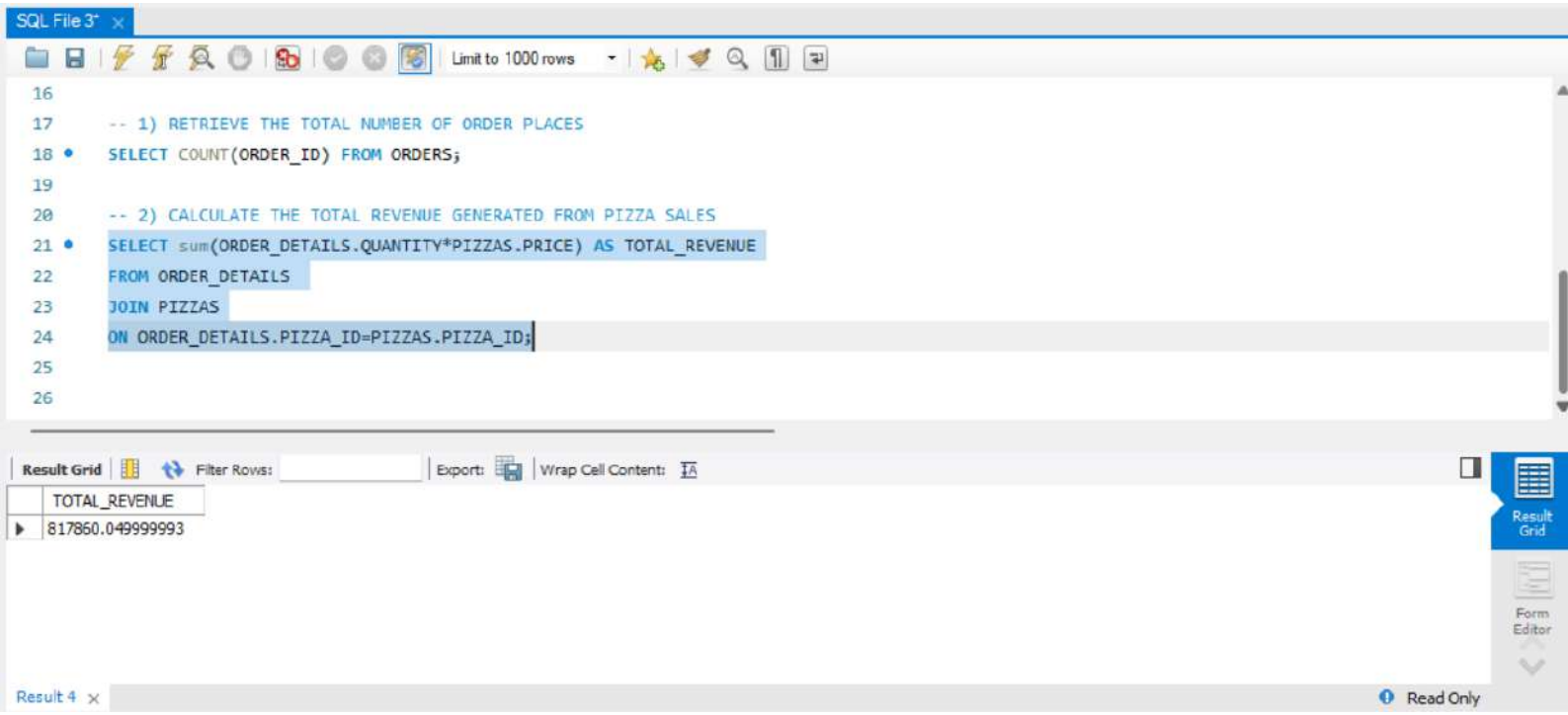
Action Output

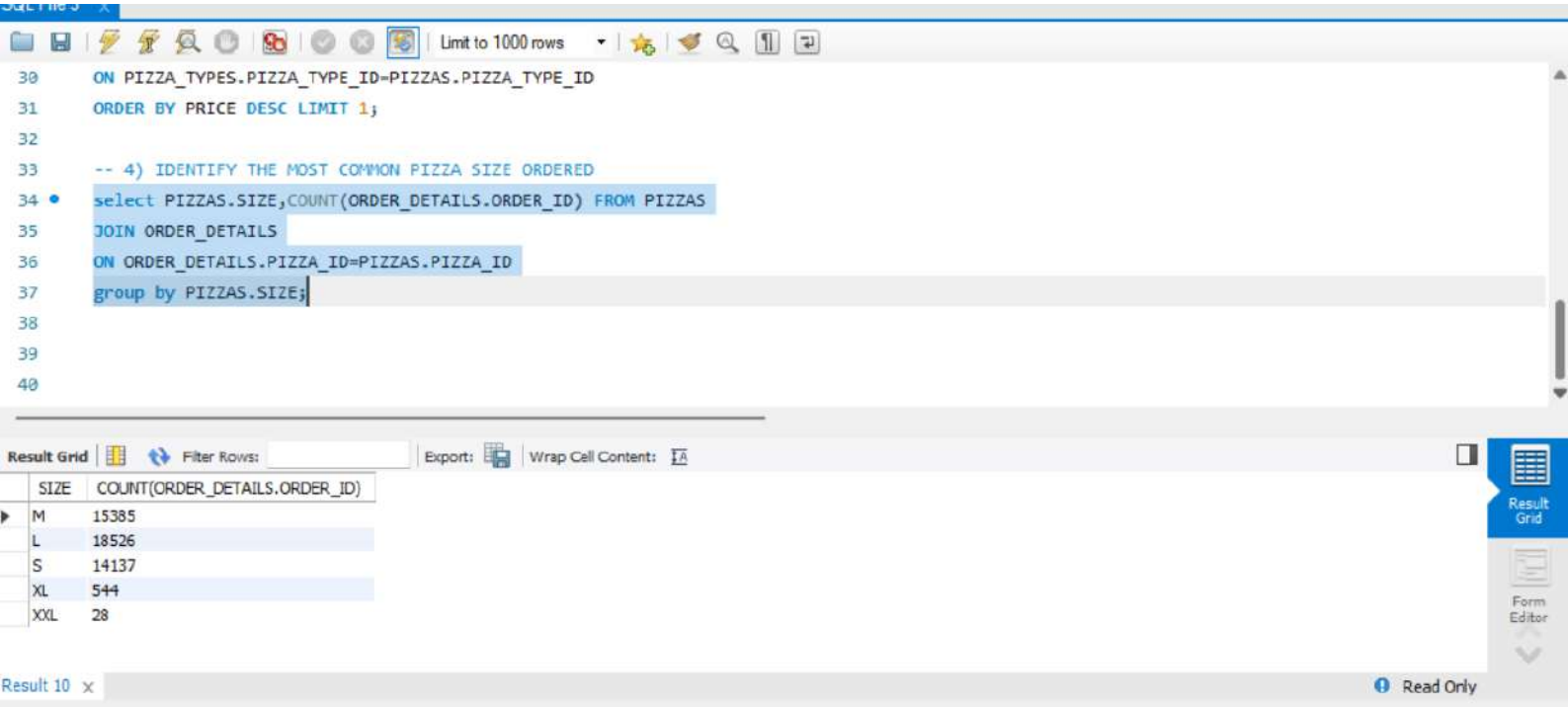
#	Time	Action	Message	Duration / Fetch
28	14:18:24	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0.000 sec
29	14:18:24	SHOW DATABASES	OK	0.000 sec
30	14:18:26	SHOW SESSION VARIABLES LIKE 'lower_case_table_names'	OK	0.000 sec
31	14:18:26	SHOW COLUMNS FROM 'olist'.'order_details'	OK	0.000 sec
32	14:18:32	PREPARE stmt FROM 'INSERT INTO 'olist'.'order_details' ('ORDER_DETAILS_ID','ORDER_ID','PIZZA_ID','...	OK	0.000 sec
33	14:24:28	DEALLOCATE PREPARE stmt	OK	0.000 sec

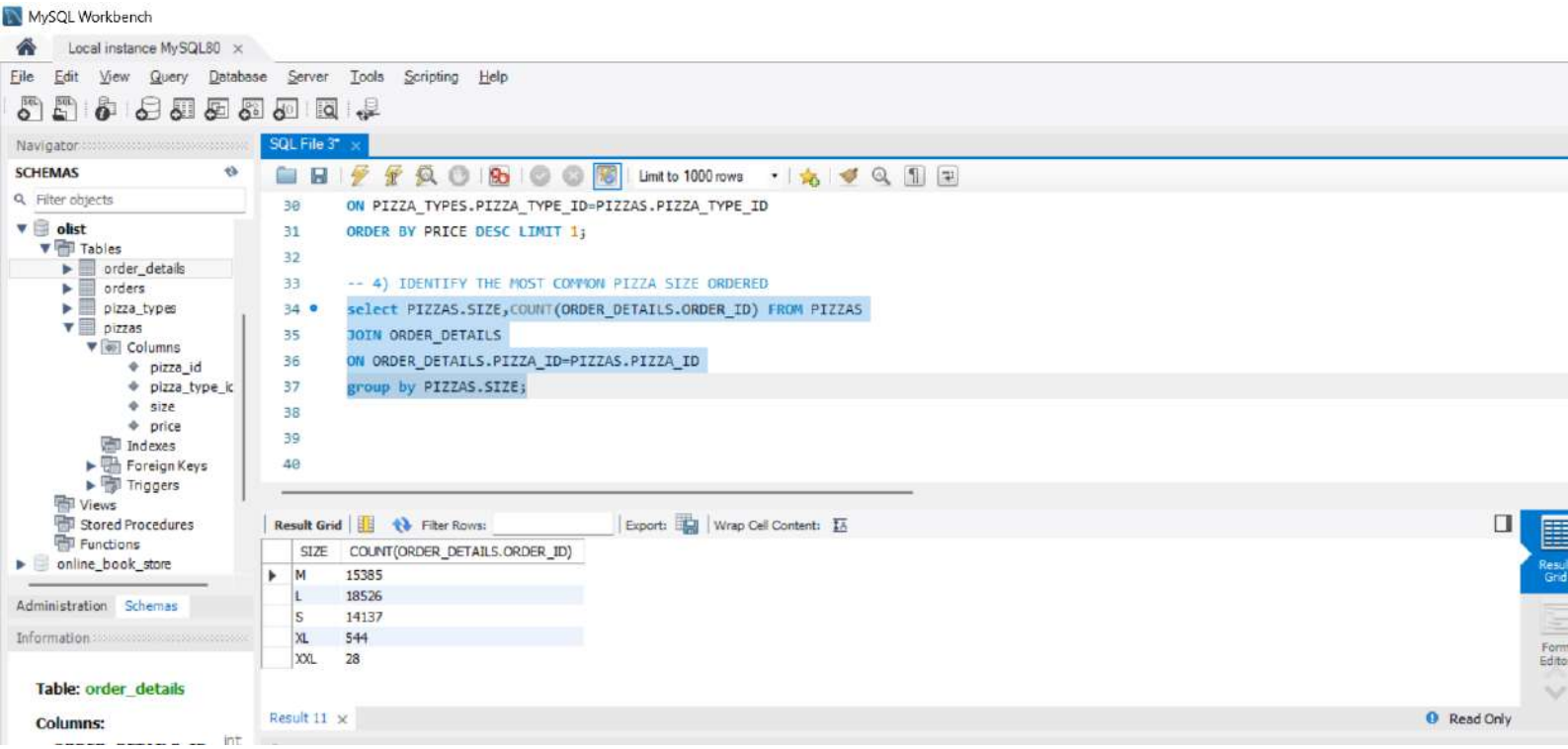

SQL File 3*
Limit to 1000 rows
9
10 CREATE TABLE ORDER_DETAILS(
11 ORDER_DETAILS_ID INT NOT NULL,
12 ORDER_ID INT NOT NULL,
13 PIZZA_ID TEXT NOT NULL,
14 QUANTITY INT NOT NULL,
15 PRIMARY KEY (ORDER_DETAILS_ID));
16
17 -- 1) RETRIEVE THE TOTAL NUMBER OF ORDER PLACES
18 SELECT * FROM ORDERS;
19
Result Grid
Filter Rows:
Edit:
Export/Import:
Wrap Cell Content:
Fetch rows:
ORDER_ID ORDER_DATE ORDER_TIME
1 2015-01-01 11:38:36
2 2015-01-01 11:57:40
3 2015-01-01 12:12:28
4 2015-01-01 12:16:31
5 2015-01-01 12:21:30
6 2015-01-01 12:29:36
7 2015-01-01 12:50:37
ORDERS 1
Result Grid
Form Editor


Limit to 1000 rows
CREATE TABLE ORDER_DETAILS(
ORDER_DETAILS_ID INT NOT NULL,
ORDER_ID INT NOT NULL,
PIZZA_ID TEXT NOT NULL,
QUANTITY INT NOT NULL,
PRIMARY KEY (ORDER_DETAILS_ID));
-- 1) RETRIEVE THE TOTAL NUMBER OF ORDER PLACES
SELECT COUNT(ORDER_ID) FROM ORDERS;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
COUNT(ORDER_ID)
21350
Result Grid
Form Editor
Result 2
Read Only


Limit to 1000 rows
QUANTITY INT NOT NULL,
PRIMARY KEY (ORDER_DETAILS_ID));
-- 1) RETRIEVE THE TOTAL NUMBER OF ORDER PLACES
SELECT COUNT(ORDER_ID) FROM ORDERS;
-- 2) CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES
SELECT (ORDER_DETAILS.QUANTITY*PIZZAS.PRICE) AS TOTAL_REVENUE
FROM ORDER_DETAILS
JOIN PIZZAS
ON ORDER_DETAILS.PIZZA_ID=PIZZAS.PIZZA_ID;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Fetch rows:
TOTAL_REVENUE
20.75
16.5
16.5
12.75
12
12.5
12.5
Result 3
Read Only
Result Grid
Form Editor


SQL File 3*
Limit to 1000 rows
16
17 -- 1) RETRIEVE THE TOTAL NUMBER OF ORDER PLACES
18 SELECT COUNT(ORDER_ID) FROM ORDERS;
19
20 -- 2) CALCULATE THE TOTAL REVENUE GENERATED FROM PIZZA SALES
21 SELECT sum(ORDER_DETAILS.QUANTITY*PIZZAS.PRICE) AS TOTAL_REVENUE
22 FROM ORDER_DETAILS
23 JOIN PIZZAS
24 ON ORDER_DETAILS.PIZZA_ID=PIZZAS.PIZZA_ID;
25
26
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
TOTAL_REVENUE
817860.049999993
Result Grid
Form Editor
Result 4
Read Only


Limit to 1000 rows
30 ON PIZZA_TYPES.PIZZA_TYPE_ID=PIZZAS.PIZZA_TYPE_ID
31 ORDER BY PRICE DESC LIMIT 1;
32
33 -- 4) IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED
34 select PIZZAS.SIZE,COUNT(ORDER_DETAILS.ORDER_ID) FROM PIZZAS
35 JOIN ORDER_DETAILS
36 ON ORDER_DETAILS.PIZZA_ID=PIZZAS.PIZZA_ID
37 group by PIZZAS.SIZE;
38
39
40
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
SIZE
COUNT(ORDER_DETAILS.ORDER_ID)
M 15385
L 18526
S 14137
XL 544
XXL 28
Result Grid
Form Editor
Result 10
Read Only


MySQL Workbench
Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
olist
Tables
order_details
orders
pizza_types
pizzas
Columns
pizza_id
pizza_type_id
size
price
Indexes
Foreign Keys
Triggers
Views
Stored Procedures
Functions
online_book_store
Administration Schemas
Information
Table: order_details
Columns:
SQL File 3*
Limit to 1000 rows
ON PIZZA_TYPES.PIZZA_TYPE_ID=PIZZAS.PIZZA_TYPE_ID
ORDER BY PRICE DESC LIMIT 1;
-- 4) IDENTIFY THE MOST COMMON PIZZA SIZE ORDERED
select PIZZAS.SIZE,COUNT(ORDER_DETAILS.ORDER_ID) FROM PIZZAS
JOIN ORDER_DETAILS
ON ORDER_DETAILS.PIZZA_ID=PIZZAS.PIZZA_ID
group by PIZZAS.SIZE;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
SIZE
COUNT(ORDER_DETAILS.ORDER_ID)
M 15385
L 18526
S 14137
XL 544
XXL 28
Result 11
Read Only

SQL File 3*

```
37    group by PIZZAS.SIZE;
38
39    -- 5) List the top 5 most orderd pizza types along with their quantities.
40    SELECT PIZZA_TYPES.name, SUM(ORDER_DETAILS.quantity) AS total_quantity
41    FROM PIZZA_TYPES
42    JOIN PIZZAS ON PIZZA_TYPES.pizza_type_id = PIZZAS.pizza_type_id
43    JOIN ORDER_DETAILS ON ORDER_DETAILS.pizza_id = PIZZAS.pizza_id
44    GROUP BY PIZZA_TYPES.name
45    ORDER BY total_quantity desc
46    LIMIT 5;
47
```

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

name	total_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 Grid | Form Editor

Result 13 | Read Only

SQL File 3*

```
45    ORDER BY total_quantity desc
46    LIMIT 5;
47
48    -- 6) JOIN THE NECESSARY TABLE TO FIND THE TOTAL QUANTITY OF EACH PIZZA ORDERD.
49    SELECT PIZZA_TYPES.CATEGORY,SUM(ORDER_DETAILS.QUANTITY) AS QUANTITY
50    FROM PIZZA_TYPES JOIN PIZZAS
51    ON PIZZA_TYPES.PIZZA_TYPE_ID=PIZZAS.PIZZA_TYPE_ID
52    JOIN ORDER_DETAILS
53    ON  ORDER_DETAILS.PIZZA_ID=PIZZAS.PIZZA_ID
54    GROUP BY PIZZA_TYPES.CATEGORY ORDER BY QUANTITY DESC;
55
```

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

CATEGORY	QUANTITY
Classic	14888
Supreme	11987
Veggie	11649
Chicken	11050

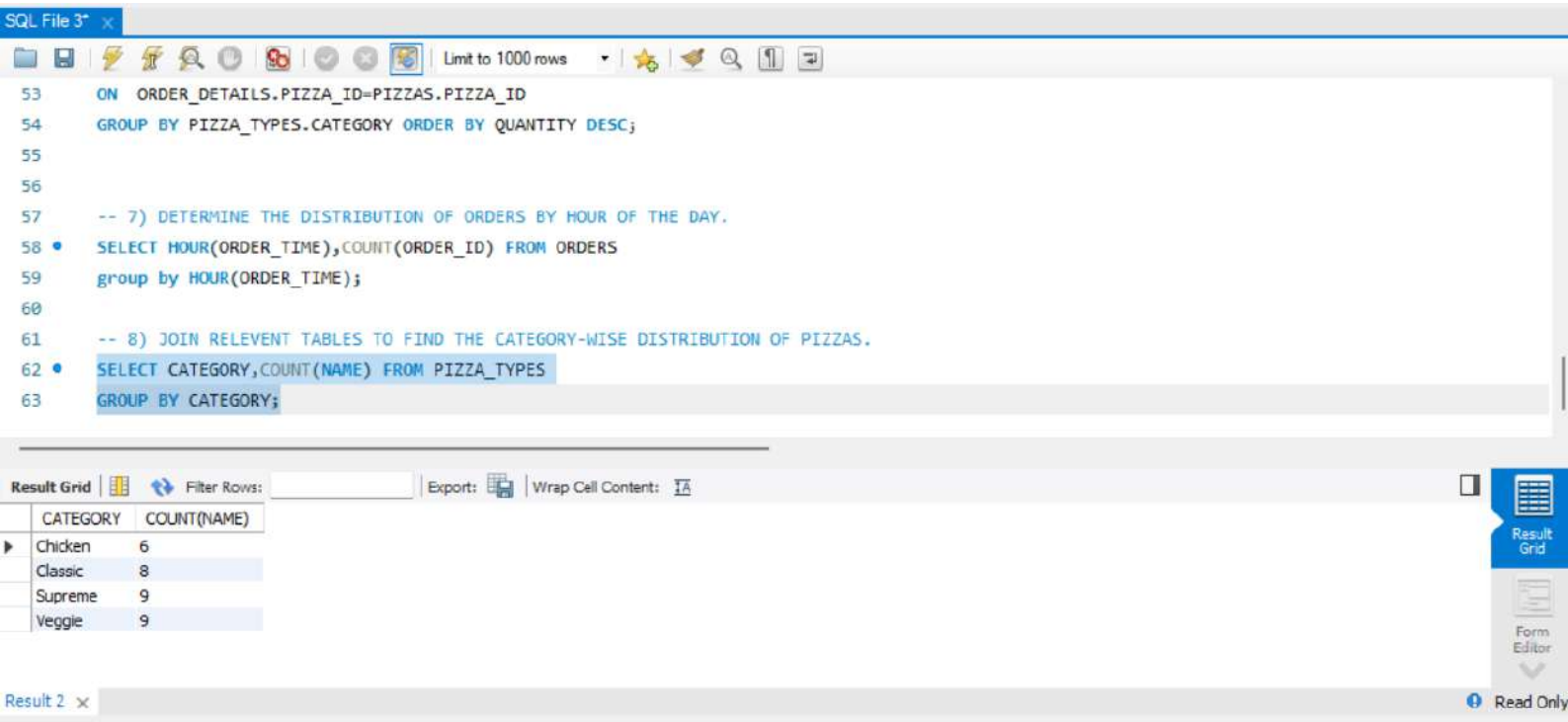
SQL File 3*

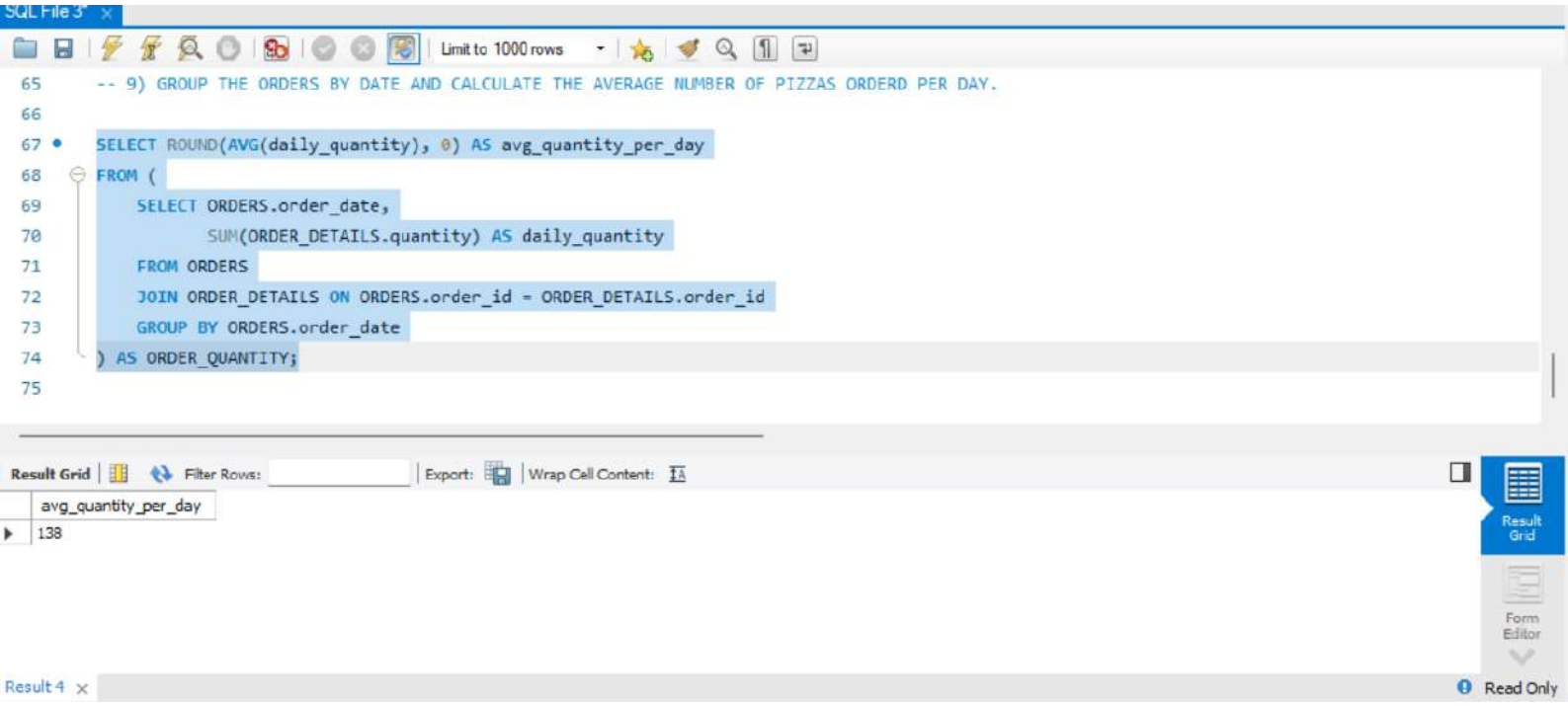
```
50    FROM PIZZA_TYPES JOIN PIZZAS
51    ON PIZZA_TYPES.PIZZA_TYPE_ID=PIZZAS.PIZZA_TYPE_ID
52    JOIN ORDER_DETAILS
53    ON  ORDER_DETAILS.PIZZA_ID=PIZZAS.PIZZA_ID
54    GROUP BY PIZZA_TYPES.CATEGORY ORDER BY QUANTITY DESC;
55
56
57    -- 7) DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.
58    SELECT HOUR(ORDER_TIME),COUNT(ORDER_ID) FROM ORDERS
59    group by HOUR(ORDER_TIME);
60
```

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

HOUR(ORDER_TIME)	COUNT(ORDER_ID)
11	1231
12	2520
13	2455
14	1472
15	1468
16	1920

Result 1 | Read Only


SQL File 3*
Limit to 1000 rows
53 ON ORDER_DETAILS.PIZZA_ID=PIZZAS.PIZZA_ID
54 GROUP BY PIZZA_TYPES.CATEGORY ORDER BY QUANTITY DESC;
55
56
57 -- 7) DETERMINE THE DISTRIBUTION OF ORDERS BY HOUR OF THE DAY.
58 SELECT HOUR(ORDER_TIME),COUNT(ORDER_ID) FROM ORDERS
59 group by HOUR(ORDER_TIME);
60
61 -- 8) JOIN RELEVENT TABLES TO FIND THE CATEGORY-WISE DISTRIBUTION OF PIZZAS.
62 SELECT CATEGORY,COUNT(NAME) FROM PIZZA_TYPES
63 GROUP BY CATEGORY;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
CATEGORY COUNT(NAME)
Chicken 6
Classic 8
Supreme 9
Veggie 9
Result Grid
Form Editor
Result 2
Read Only


SQL File 3*
Limit to 1000 rows
-- 9) GROUP THE ORDERS BY DATE AND CALCULATE THE AVERAGE NUMBER OF PIZZAS ORDERD PER DAY.
SELECT ROUND(AVG(daily_quantity), 0) AS avg_quantity_per_day
FROM (
SELECT ORDERS.order_date,
SUM(ORDER_DETAILS.quantity) AS daily_quantity
FROM ORDERS
JOIN ORDER_DETAILS ON ORDERS.order_id = ORDER_DETAILS.order_id
GROUP BY ORDERS.order_date
) AS ORDER_QUANTITY;
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
avg_quantity_per_day
138
Result Grid
Form Editor
Result 4
Read Only

SQL File 3*

Limit to 1000 rows

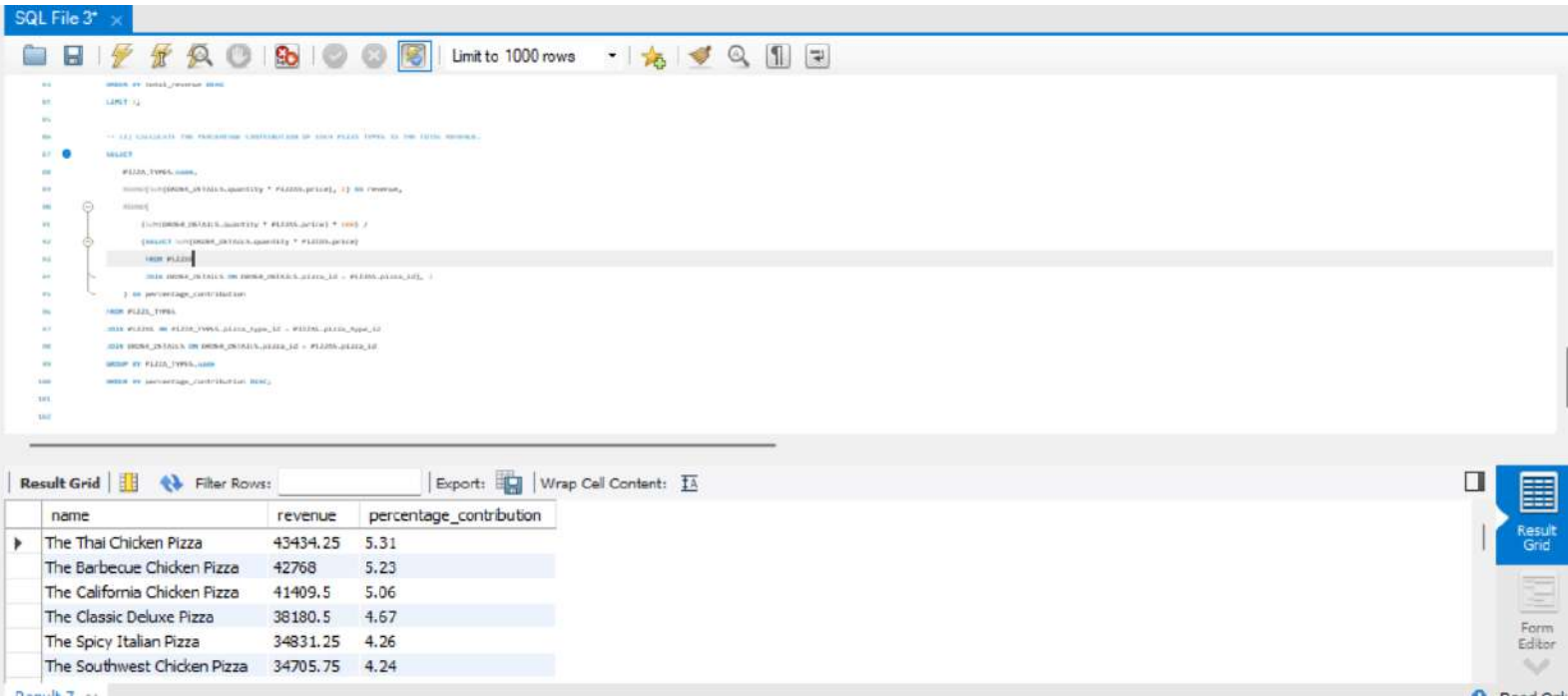
```
76  -- 10) DETERMINE THE TOP 3 MOST ORDERD PIZZA TYPES BASED ON REVENUE.
77  SELECT PIZZA_TYPES.name,
78         SUM(ORDER_DETAILS.quantity * PIZZAS.price) AS total_revenue
79  FROM PIZZA_TYPES
80  JOIN PIZZAS ON PIZZA_TYPES.pizza_type_id = PIZZAS.pizza_type_id
81  JOIN ORDER_DETAILS ON ORDER_DETAILS.pizza_id = PIZZAS.pizza_id
82  GROUP BY PIZZA_TYPES.name
83  ORDER BY total_revenue DESC
84  LIMIT 3;
85
86
```

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

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

Result Grid

Form Editor



name	revenue	percentage_contribution
The Thai Chicken Pizza	43434.25	5.31
The Barbecue Chicken Pizza	42768	5.23
The California Chicken Pizza	41409.5	5.06
The Classic Deluxe Pizza	38180.5	4.67
The Spicy Italian Pizza	34831.25	4.26
The Southwest Chicken Pizza	34705.75	4.24

SQL File 3*

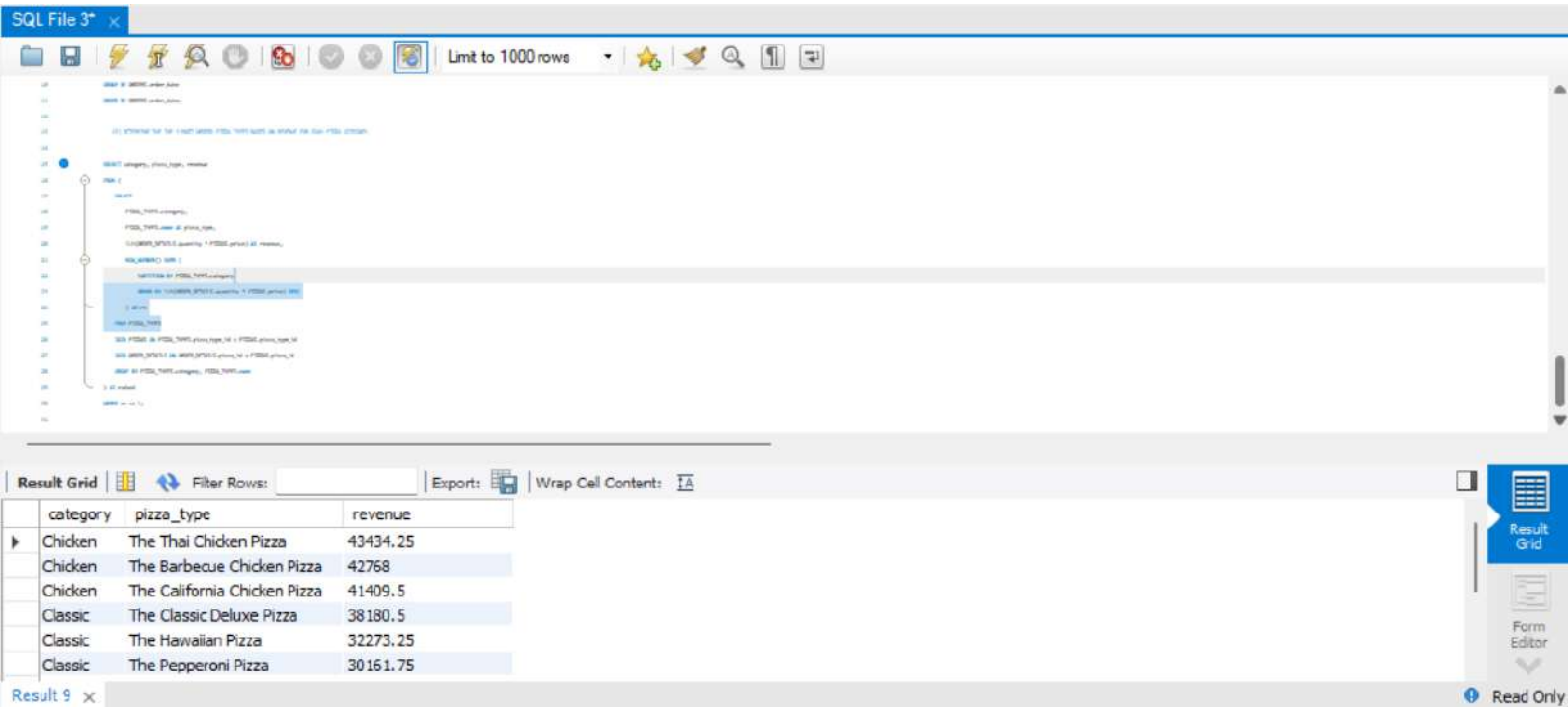
```sql
-- 12) ANALYZE THE CUMULATIVE REVENUE GENERATED OVER TIME
SELECT
    ORDERS.order_date,
    ROUND(SUM(ORDER_DETAILS.quantity * PIZZAS.price), 2) AS daily_revenue,
    ROUND(SUM(SUM(ORDER_DETAILS.quantity * PIZZAS.price)) OVER (ORDER BY ORDERS.order_date), 2) AS cumulative_revenue
FROM ORDERS
JOIN ORDER_DETAILS ON ORDERS.order_id = ORDER_DETAILS.order_id
JOIN PIZZAS ON ORDER_DETAILS.pizza_id = PIZZAS.pizza_id
GROUP BY ORDERS.order_date
ORDER BY ORDERS.order_date;
```

Result Grid

order_date	daily_revenue	cumulative_revenue
2015-01-01	2713.85	2713.85
2015-01-02	2731.9	5445.75
2015-01-03	2662.4	8108.15
2015-01-04	1755.45	9863.6
2015-01-05	2065.95	11929.55
2015-01-06	2428.95	14358.5

Result 8

Read Only



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

category	pizza_type	revenue
Chicken	The Thai Chicken Pizza	43434.25
Chicken	The Barbecue Chicken Pizza	42768
Chicken	The California Chicken Pizza	41409.5
Classic	The Classic Deluxe Pizza	38180.5
Classic	The Hawaiian Pizza	32273.25
Classic	The Pepperoni Pizza	30161.75

Result 9 | Read Only

SQL File 3*

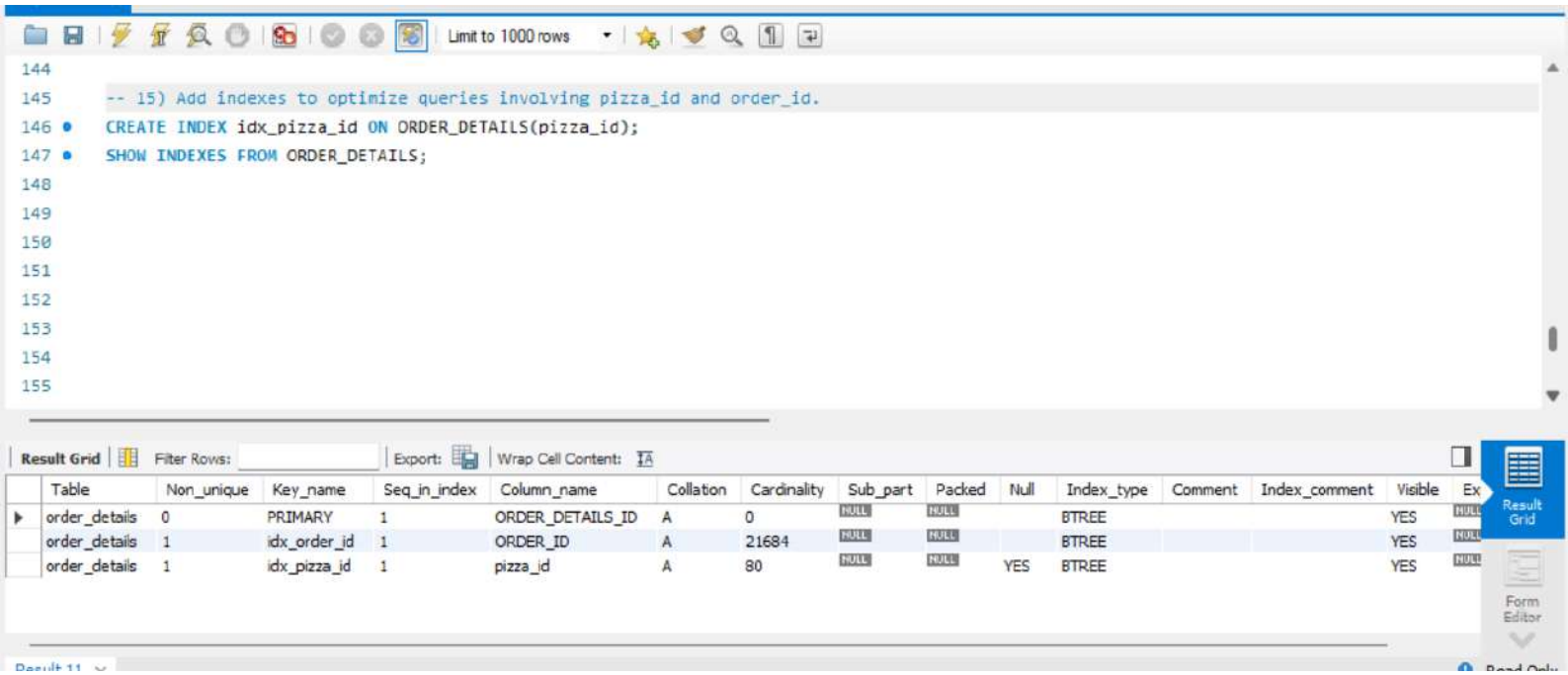
```sql
-- 14) Create a view that shows each pizza type with total quantity and total revenue.
CREATE OR REPLACE VIEW pizza_summary_view AS
SELECT
    PIZZA_TYPES.name AS pizza_name,
    SUM(ORDER_DETAILS.quantity) AS total_quantity,
    ROUND(SUM(ORDER_DETAILS.quantity * PIZZAS.price), 2) AS total_revenue
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;

SELECT * FROM pizza_summary_view;
```

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

pizza_name	total_quantity	total_revenue
The Hawaiian Pizza	2422	32273.25
The Classic Deluxe Pizza	2453	38180.5
The Five Cheese Pizza	1409	26066.5
The Italian Supreme Pizza	1884	33476.75
The Mexicana Pizza	1484	26780.75
The Thai Chicken Pizza	2371	43434.25

pizza_summary_view 10

Result Grid | Form Editor | Read Only


Limit to 1000 rows
144
145 -- 15) Add indexes to optimize queries involving pizza_id and order_id.
146 CREATE INDEX idx_pizza_id ON ORDER_DETAILS(pizza_id);
147 SHOW INDEXES FROM ORDER_DETAILS;
148
149
150
151
152
153
154
155
Result Grid
Filter Rows:
Export:
Wrap Cell Content:
Table | Non_unique | Key_name | Seq_in_index | Column_name | Collation | Cardinality | Sub_part | Packed | Null | Index_type | Comment | Index_comment | Visible | Ex
order_details | 0 | PRIMARY | 1 | ORDER_DETAILS_ID | A | 0 | NULL | NULL | | BTREE | | | YES | NULL
order_details | 1 | idx_order_id | 1 | ORDER_ID | A | 21684 | NULL | NULL | | BTREE | | | YES | NULL
order_details | 1 | idx_pizza_id | 1 | pizza_id | A | 80 | NULL | NULL | YES | BTREE | | | YES | NULL
Result Grid
Form Editor