

◊ Project Title

Customer, Order, and Product Data Analysis Using SQL

◊ Project Objective

The objective of this project is to **design, populate, and analyze a customer database** using SQL. The project demonstrates the use of:

- Database creation
- Table relationships using foreign keys
- Data insertion
- Data retrieval using SELECT
- Filtering with WHERE
- Aggregation using COUNT, AVG, MAX, MIN
- Grouping and filtering using GROUP BY and HAVING
- Joins (INNER, LEFT, RIGHT)
- Stored procedures and triggers

◊ Database Overview

- Database Name
- Customer_data

◊ Tables Description

① customer_data

Stores customer personal and category information.

Column Name	Description
customer_ID	Unique customer identifier (Primary Key)
category_name	Product category
customer_city	City of customer
customer_count	Customer country
customer_Fname	Customer first name

[2] customer_segment

Stores segmentation and market details.

Column Name	Description
customer_segment	Customer type (Consumer/Home Office)
customer_state	State code
customer_zipcode	Zip code
market	Market region
customer_ID	Foreign key referencing <i>customer_data</i>

[3] customer_order

Stores order transaction details.

Column Name	Description
order_id	Order identifier
order_customerID	Customer ID
order_date	Date of order
order_Region	Order region
order_item_total	Total order value
order_quantity	Quantity ordered

[4] product

Stores product pricing and profit information.

Column Name	Description
product_price	Product price
profit_price	Profit percentage
profit_perorder	Profit per order
sales	Sales value

- **Data Filtering**

```
WHERE customer_id > 568  
AND category_name = 'Hiking';
```

- **Aggregation & Grouping**

```
SELECT category_name, COUNT(*)  
FROM customer_data  
GROUP BY category_name  
HAVING COUNT(*) > 1;
```

- **Aggregation & Grouping**
COUNT()

AVG()

MAX()

MIN()

Used to analyze:

Sales

Profit

Order quantity

- **Joins**

INNER JOIN
LEFT JOIN
RIGHT JOIN

Used to combine **customer details with market segments**.

- **Stored Procedure**

GETcustomer_ID

Purpose:

Fetch customer details using customer ID as input parameter.

- **Trigger**

BEFORE UPDATE

Demonstrates automated execution of SQL logic before data modification.

- Customers are mostly concentrated in **Hiking** and **Camping** categories
- Majority belong to **Home Office** segment
- Sales values vary significantly across products
- Some products show **negative profit margins**
- Multiple customers belong to the **same market region**

• Tools & Technologies

- **Database:** MySQL
- **Language:** SQL

- Concepts Used:
 - DDL (CREATE, ALTER)
 - DML (INSERT, SELECT)
 - Constraints
 - Joins
 - Procedures & Triggers

◊ Conclusion

- This project successfully demonstrates **end-to-end SQL database handling**, from creation to advanced querying. It reflects real-world business data analysis.

MySQL Workbench

Local instance MySQL80 ×

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

- amazon
- amazon1
- amazon2
- customer
- customer_data**
- Views
- Stored Procedures
- Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

e 10* SQL File 11* SQL File 12* SQL File 16* appointments billing doctors SQL File 13* Employee Department DATA* SQL File 19* SQL File 20* SQL File 22* SQL File 22* ×

1 • CREATE DATABASE customer_data;

2

3 • CREATE TABLE customer_data (

4 customer_ID INT PRIMARY KEY,

5 category_name VARCHAR(255),

6 customer_city VARCHAR(200),

7 customer_count VARCHAR(250),

8 customer_Fname VARCHAR(200)

9);

10 • INSERT INTO customer_data

11 VALUES (568,'Hiking','Caguas','Puerto Rico','Sean'),

12 (3341,'Hiking','Caguas','Puerto Rico','Carol'),

13 (7459,'Camping ','Mayaguez','Puerto Rico','Mary'),

14 (7454,'Hiking','Mayaguez','Puerto Rico','Mary'),

15 (10740,'Shooting','Caguas','Puerto Rico','Mary'),

16 (5193,'Camping ','Mayaguez','Puerto Rico','Amanda'),

17 (2092,'Camping ','Mayaguez','Puerto Rico','Mary'),

18 (9645,'Hiking','Mayaguez','Puerto Rico','Amanda'),

19 (997,'Hiking','Mayaguez','Puerto Rico','Mary'),

20 (3339,'Hiking ','Mayaguez','Puerto Rico','Mary'),

21 (4899,'Camping ','Mayaguez','Puerto Rico','Mary'),

22 (1232,'Hiking','Mayaguez','Puerto Rico','Mary').

Output

#	Time	Action	Message	Duration / Fetch
86	12:17:15	SELECT * FROM product ORDER BY profit_price LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.016 sec
87	12:17:20	SELECT * FROM product ORDER BY sales DESC LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
88	12:17:27	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
89	12:18:12	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

Navigator :: e 10* SQL File 11* SQL File 12* SQL File 16* appointments billing doctors SQL File 13* Employee Department DATA* SQL

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information ::

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

```

25   (5421,'Shop By Sport','Caguas','Puerto Rico','Elizabeth'),
26   (9923,'Womens Apparel','Caguas','Puerto Rico','Katherine');
27
28 • CREATE TABLE customer_segment (
29   customer_segment VARCHAR (200),
30   customer_state VARCHAR(250),
31   customer_zipcode VARCHAR(200),
32   market VARCHAR(250),
33   customer_ID INT ,
34   FOREIGN KEY (customer_ID) REFERENCES customer_data(customer_id)
35 );
36 • INSERT INTO customer_segment
37   VALUES ('consumer','PR','725','LATAM',568),
38   ('consumer','PR','725','LATAM',3341),
39   ('consumer','PR','680','LATAM',7459),
40   ('consumer','PR','680','LATAM',7454),
41   ('Home Office','PR','725','Pacific Asia',10740),
42   ('Home Office','PR','725','Pacific Asia',5193),
43   ('Home Office','PR','725','USCA',2092),
44   ('Home Office','PR','725','USCA',9645),
45   ('Home Office','PR','725','USCA',997),
46   ('Home Office','PR','725','Africa',32291);

```

Output

#	Time	Action	Message
86	12:17:15	SELECT * FROM product ORDER BY profit_price LIMIT 0, 1000	32 row(s) returned
87	12:17:20	SELECT * FROM product ORDER BY sales DESC LIMIT 0, 1000	32 row(s) returned
88	12:17:27	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned
89	12:18:12	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned

Navigator SQL File 11* SQL File 12* SQL File 16* appointments billing doctors SQL File 13* Employee Department DATA* SQL File 19* SQL File 20* SQL File 22* SQL File 22*

SCHEMAS

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
- Stored Procedures
- Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

```

52   ('Home Office','PR','725','Europe',9923);
53
54 • CREATE TABLE customer_order (
55   order_id VARCHAR(300),
56   order_customerID VARCHAR(200),
57   order_date DATE,
58   order_Region VARCHAR(250),
59   order_item_total VARCHAR(200),
60   order_quantity VARCHAR(200)
61 );
62 • INSERT INTO customer_order
63   VALUES (61558,568,'2017-06-17','South America','269.980011',1),
64   (57472,3341,'2017-04-18','South America','269.980011',1),
65   (59898,7459,'2017-05-24','Central America','263.980011',1),
66   (59898,7459,'2017-05-24','Central America','263.980011',1),
67   (58644,10740,'2017-06-05','Central America','130.490005',1),
68   (56882,2092,'2017-10-04','Caribbean','254.979995',1),
69   (51638,9645,'2017-01-23','Central America','254.979995',1),
70   (28744,9083,'2017-02-24','South Asia','115.180003',2),
71   (45461,4741,'2017-10-25','West Asia','79.1800031',2),
72   (34773,11169,'2017-05-22','US Center','95.98000336',2),
73   (46414,1535,'2016-08-11','Central Africa','113.0700012',3),

```

Output

#	Time	Action	Message	Duration / Fetch
86	12:17:15	SELECT * FROM product ORDER BY profit_price LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.016 sec
87	12:17:20	SELECT * FROM product ORDER BY sales DESC LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
88	12:17:27	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
89	12:18:12	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

Navigator e 10* SQL File 11* SQL File 12* SQL File 16* appointments billing doctors SQL File 13* Employee Department DATA* SQL File 19* SQL File 20* SQL File 22* SQL File 22* ×

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

```

76   (63936,11329,'2017-07-22','Western Europe','139.5',3),
77   (67892,3182,'2017-09-18','Western Europe','112.5',3),
78   (56359,1025,'2017-02-04','Central America','178.1699982',3);
79
80 • CREATE TABLE product (
81   product_price VARCHAR(250),
82   profit_price VARCHAR(200),
83   profit_perorder VARCHAR(200),
84   sales VARCHAR(250)
85 );
86 • INSERT INTO product
87   VALUES (299.980011,'26.10%',78.29000092,800),
88   (299.980011,'5.40%',-16.20000076,800),
89   (149.9900055,'-63.80%',-95.65000153,149.9900055),
90   (299.980011,'23.80%',71.40000153,800),
91   (59.99000168,'-25.60%',-30.75,119.9800034),
92   (50,'33.60%',33.59999847,100),
93   (39.99000168,'30.90%',24.69000053,79.98000336),
94   (54.97000122,'23.00%',25.23999977,109.9400024),
95   (59.99000168,'38.40%',46.0699997,119.9800034),
96   (99.98999786,'44.10%',132.2899933,800),
97   (39.99000168,'10.20%',12.19999981,119.97000121);

```

Output

#	Time	Action	Message	Duration / Fetch
86	12:17:15	SELECT * FROM product ORDER BY profit_price LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.016 sec
87	12:17:20	SELECT * FROM product ORDER BY sales DESC LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
88	12:17:27	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
89	12:18:12	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

Navigator SQL File 11* SQL File 12* SQL File 16* appointments billing doctors SQL File 13* Employee Department DATA* SQL File 19* SQL File 20* SQL File 22* SQL File 22* X

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

```

103
104 •   SELECT *
105     FROM customer_data
106     WHERE customer_id > 568;
107
108 •   SELECT *
109     FROM customer_data
110     WHERE customer_id > 568 AND category_name = 'Hiking';
111
112 •   SELECT *
113     FROM customer_data
114     WHERE customer_id > 568 AND category_name = 'Camping ';
115
116 •   SELECT category_name, COUNT(*) AS Total_Categoryname
117     FROM customer_data
118     GROUP BY category_name
119     HAVING COUNT(*) > 1;
120
121 •   SELECT sales, AVG(product_price) AS Total_productprice
122     FROM product
123     GROUP BY sales
124     HAVING AVG(product_price) > 50;

```

Output

#	Time	Action	Message	Duration / Fetch
86	12:17:15	SELECT * FROM product ORDER BY profit_price LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.016 sec
87	12:17:20	SELECT * FROM product ORDER BY sales DESC LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
88	12:17:27	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
89	12:18:12	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

Navigator SQL File 11* SQL File 12* SQL File 16* appointments billing doctors SQL File 13* Employee Department DATA* SQL File 19* SQL File 20* SQL File 22* SQL File 22* X

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

```

121 •   SELECT sales, AVG(product_price) AS Total_productprice
122     FROM product
123     GROUP BY sales
124     HAVING AVG(product_price) > $0;
125
126 •   SELECT profit_price, AVG(sales) AS Total_sales
127     FROM product
128     GROUP BY profit_price
129     HAVING AVG(sales) > 100;
130
131 •   SELECT profit_price, sales
132     FROM product;
133
134 •   SELECT customer_ID, market
135     FROM customer_segment;
136
137 •   SELECT order_item_total, order_quantity
138     FROM customer_order;
139
140 •   SELECT category_name, customer_Fname
141     FROM customer_data;
142

```

Output

Action Output	#	Time	Action	Message	Duration / Fetch
	86	12:17:15	SELECT * FROM product ORDER BY profit_price LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.016 sec
	87	12:17:20	SELECT * FROM product ORDER BY sales DESC LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
	88	12:17:27	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
	89	12:18:12	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

The screenshot shows the MySQL Workbench application. The left sidebar displays the 'SCHEMAS' tree, which includes the 'customer_data' schema expanded to show 'Tables' (customer_data, customer_order, customer_segment, product) and 'Views'. The main area is a SQL editor window with the following code:

```
e 10*   SQL File 11*   SQL File 12*   SQL File 16*   appointments   billing   doctors   SQL File 13*   Employee Department DATA*   SQL File 19*   SQL File 20*   SQL File 22*   SQL File 22* x

142
143 •    SELECT COUNT(*)category_name FROM customer_data;
144 •    SELECT COUNT(*)order_quantity FROM customer_order;
145 •    SELECT COUNT(*)customer_state FROM customer_segment;
146 •    SELECT COUNT(*)profit_perorder FROM product;
147 •    SELECT COUNT(*) AS sales FROM customer_data;
148 •    SELECT COUNT(*) AS profit_price FROM customer_data;
149 •    SELECT DISTINCT sales FROM product;
150 •    SELECT DISTINCT profit_price FROM products;
151
152 DELIMITER //
153
154 •    CREATE TRIGGER customer_data
155 before update on customer_data
156 for each row
157 INSERT INTO customer_data
158 values (12323,'Hiking','Caguas','Puerto Rico','Sean');
159 END;//
160
161
162 DELIMITER //
163 •    CREATE PROCEDURE GETCustomer_ID (IN customer_ID INT)
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
389
390
391
392
393
394
395
396
397
398
399
399
400
401
402
403
404
405
406
407
408
409
409
410
411
412
413
414
415
416
417
418
419
419
420
421
422
423
424
425
426
427
428
429
429
430
431
432
433
434
435
436
437
438
439
439
440
441
442
443
444
445
446
447
448
449
449
450
451
452
453
454
455
456
457
458
459
459
460
461
462
463
464
465
466
467
468
469
469
470
471
472
473
474
475
476
477
478
479
479
480
481
482
483
484
485
486
487
488
489
489
490
491
492
493
494
495
496
497
498
499
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
588
589
589
590
591
592
593
594
595
596
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
618
619
619
620
621
622
623
624
625
626
627
628
629
629
630
631
632
633
634
635
636
637
638
639
639
640
641
642
643
644
645
646
647
648
649
649
650
651
652
653
654
655
656
657
658
659
659
660
661
662
663
664
665
666
667
668
669
669
670
671
672
673
674
675
676
677
678
679
679
680
681
682
683
684
685
686
687
688
689
689
690
691
692
693
694
695
696
697
698
698
699
699
700
701
702
703
704
705
706
707
708
709
709
710
711
712
713
714
715
716
717
718
719
719
720
721
722
723
724
725
726
727
728
729
729
730
731
732
733
734
735
736
737
738
739
739
740
741
742
743
744
745
746
747
748
749
749
750
751
752
753
754
755
756
757
758
759
759
760
761
762
763
764
765
766
767
768
769
769
770
771
772
773
774
775
776
777
778
779
779
780
781
782
783
784
785
786
787
788
789
789
790
791
792
793
794
795
796
797
797
798
799
799
800
801
802
803
804
805
806
807
808
809
809
810
811
812
813
814
815
816
817
818
819
819
820
821
822
823
824
825
826
827
828
829
829
830
831
832
833
834
835
836
837
838
839
839
840
841
842
843
844
845
846
847
848
849
849
850
851
852
853
854
855
856
857
858
859
859
860
861
862
863
864
865
866
867
868
869
869
870
871
872
873
874
875
876
877
878
879
879
880
881
882
883
884
885
886
887
888
889
889
890
891
892
893
894
895
896
897
897
898
899
899
900
901
902
903
904
905
906
907
908
909
909
910
911
912
913
914
915
916
917
918
919
919
920
921
922
923
924
925
926
927
928
929
929
930
931
932
933
934
935
936
937
938
939
939
940
941
942
943
944
945
946
947
948
949
949
950
951
952
953
954
955
956
957
958
959
959
960
961
962
963
964
965
966
967
968
969
969
970
971
972
973
974
975
976
977
978
979
979
980
981
982
983
984
985
986
987
988
989
989
990
991
992
993
994
995
996
997
997
998
999
999
1000
1000
```

Navigator

SCHEMAS

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
- Stored Procedures
- Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

SQL Editor

```

e 10* SQL File 11* SQL File 12* SQL File 16* appointments billing doctors SQL File 13* Employee Department DATA* SQL File 19* SQL File 20* SQL File 22* SQL File 22* ×
160  SELECT * FROM customer_data;
161
162  DELIMITER //
163  • CREATE PROCEDURE GETcustomer_ID (IN customer_ID INT)
164  BEGIN
165      SELECT * FROM customer_data WHERE id = customer_ID;
166  END; //
167
168
169  • SELECT
170      cd.customer_Fname,
171      cd.customer_City,
172      cs.customer_Segment,
173      cs.market
174  FROM
175      customer_data cd
176  INNER JOIN
177      customer_segment cs ON cd.customer_ID = cs.customer_ID;
178
179  SELECT
180      cd.customer_Fname,
181      cd.customer_City.

```

Output

#	Time	Action	Message	Duration / Fetch
86	12:17:15	SELECT * FROM product ORDER BY profit_price LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.016 sec
87	12:17:20	SELECT * FROM product ORDER BY sales DESC LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
88	12:17:27	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
89	12:18:12	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

Schemas

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department
- Administration
- Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Result 87 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
123	12:40:16	SELECT MAX(sales) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
124	12:40:29	SELECT MIN(sales) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
125	12:40:41	SELECT AVG(sales) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
126	12:40:52	SELECT MAX(profit_price) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
127	12:41:03	SELECT MIN(profit_price) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Result Grid | Filter Rows: _____ | Export: | Wrap Cell Content: | **MIN(profit_price)** | -158.40% | Result Grid | Form Editor | Field Types | Read Only

```

199  SELECT MAX(order_quantity) FROM customer_order;
200  SELECT MIN(order_quantity) FROM customer_order;
201  SELECT MAX(sales) FROM product;
202  SELECT MIN(sales) FROM product;
203  SELECT AVG(sales) FROM product;
204  SELECT MAX(profit_price) FROM product;
205  SELECT MIN(profit_price) FROM product;
206  SELECT AVG(profit_price) FROM product;
207
208  SELECT *

```

SCHEMAS

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Result 88 x

Output

Action Output	#	Time	Action	Message	Duration / Fetch
124 12:40:29 SELECT MIN(sales) FROM product LIMIT 0, 1000	124	12:40:29	SELECT MIN(sales) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
125 12:40:41 SELECT AVG(sales) FROM product LIMIT 0, 1000	125	12:40:41	SELECT AVG(sales) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
126 12:40:52 SELECT MAX(profit_price) FROM product LIMIT 0, 1000	126	12:40:52	SELECT MAX(profit_price) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
127 12:41:03 SELECT MIN(profit_price) FROM product LIMIT 0, 1000	127	12:41:03	SELECT MIN(profit_price) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
128 12:41:16 SELECT AVG(profit_price) FROM product LIMIT 0, 1000	128	12:41:16	SELECT AVG(profit_price) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Schemas

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
- Stored Procedures
- Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Code Editor

```

205  SELECT MIN(profit_price) FROM product;
206  SELECT AVG(profit_price) FROM product;
207
208  SELECT *
209  FROM product
210  ORDER BY sales DESC;
211
212  SELECT *
213  FROM product
214  ORDER BY profit price;

```

Result Grid

product_price	profit_price	profit_perorder	sales
299.980011	26.10%	78.29000092	800
299.980011	5.40%	-16.20000076	800
299.980011	23.80%	71.40000153	800
99.98999786	44.10%	132.2899933	800
299.980011	-8.30%	-24.89999962	800
299.980011	-8.30%	-24.89999962	800
299.980011	26.10%	78.29000092	800
299.980011	5.40%	-16.20000076	800
299.980011	23.80%	71.40000153	800
99.98999786	44.10%	132.2899933	800
...
product 89	X	24.89999967	800

Output

#	Time	Action	Message	Duration / Fetch
125	12:40:41	SELECT AVG(sales) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
126	12:40:52	SELECT MAX(profit_price) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
127	12:41:03	SELECT MIN(profit_price) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
128	12:41:16	SELECT AVG(profit_price) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
129	12:41:29	SELECT * FROM product ORDER BY sales DESC LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Result Grid Form Editor Field Types Read Only

Navigator e 10* SQL File 11* SQL File 12* SQL File 16* appointments billing doctors SQL File 13* Employee Department DATA* SQL File 19* SQL File 20* SQL File 22* SQL File 22* X

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	order_customerID	order_date	order_region	order_item_total	order_quantity
varchar(250)	varchar(200)	date	varchar(250)	varchar(200)	varchar(200)

```

178
179   SELECT
180     cd.customer_Fname,
181     cd.customer_city,
182     cs.customer_segment,
183     cs.market
184   FROM
185     customer_data cd
186   LEFT JOIN
187     customer_segment cs ON cd.customer_ID = cs.customer_ID;
188
189   SELECT |
190     cd.customer_Fname,
191     cd.customer_city,
192     cs.customer_segment,
193     cs.market
194   FROM
195     customer_data cd
196   RIGHT JOIN
197     customer_segment cs ON cd.customer_ID = cs.customer_ID;
198
199   SELECT MAX(order_quantity) FROM customer_order;

```

Output

#	Time	Action	Message	Duration / Fetch
86	12:17:15	SELECT * FROM product ORDER BY profit_price LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.016 sec
87	12:17:20	SELECT * FROM product ORDER BY sales DESC LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
88	12:17:27	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
89	12:18:12	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

Navigator SQL File 11* SQL File 12* SQL File 16* appointments billing doctors SQL File 13* Employee Department DATA* SQL File 19* SQL File 20* SQL File 22* SQL File 22*

SCHEMAS

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

```

206  SELECT AVG(profit_price) FROM product;
207
208  SELECT *
209    FROM product
210   ORDER BY sales DESC;
211
212  SELECT *
213    FROM product
214   ORDER BY profit_price;
215
216  SELECT * FROM customer_data;
217  SELECT * FROM customer_order;
218  SELECT * FROM product;
219
220
221
222
223
224
225
226
???

```

Output

#	Time	Action	Message	Duration / Fetch
86	12:17:15	SELECT * FROM product ORDER BY profit_price LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.016 sec
87	12:17:20	SELECT * FROM product ORDER BY sales DESC LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
88	12:17:27	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
89	12:18:12	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Code Editor

```

1 • CREATE DATABASE customer_data;
2
3 • CREATE TABLE customer_data (
4     customer_ID INT PRIMARY KEY,
5     category_name VARCHAR(255),
6     customer_city VARCHAR(200),
7     customer_count VARCHAR(250),
8     customer_Fname VARCHAR(200)
9 );
10 • INSERT INTO customer_data

```

Result Grid

customer_ID	category_name	customer_city	customer_count	customer_Fname
568	Hiking	Caguas	Puerto Rico	Sean
714	Camping	Mayaguez	Puerto Rico	Mark
997	Hiking	Mayaguez	Puerto Rico	Mary
2092	Camping	Mayaguez	Puerto Rico	Mary
3339	Hiking	Mayaguez	Puerto Rico	Mary
3341	Hiking	Caguas	Puerto Rico	Carol
4899	Camping	Mayaguez	Puerto Rico	Mary
5193	Camping	Mayaguez	Puerto Rico	Amanda
5421	Shop By Sport	Caguas	Puerto Rico	Elizabeth
7454	Hiking	Mayaguez	Puerto Rico	Mary
7455	Camping	Mayaguez	Puerto Rico	Mary

Output

#	Time	Action	Message	Duration / Fetch
91	12:30:15	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
92	12:30:28	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
93	12:30:54	SELECT * FROM customer_segment LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
94	12:30:58	SELECT * FROM product	32 row(s) returned	0.000 sec / 0.000 sec
95	12:31:02	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Code Editor

```

28 • CREATE TABLE customer_segment (
29   customer_segment VARCHAR (200),
30   customer_state VARCHAR(250),
31   customer_zipcode VARCHAR(200),
32   market VARCHAR(250),
33   customer_ID INT ,
34   FOREIGN KEY (customer_ID) REFERENCES customer_data(customer_id)
35   );
36 • INSERT INTO customer_segment
37   VALUES ('consumer','PR','725','LATAM',568),

```

Result Grid

customer_segment	customer_state	customer_zipcode	market	customer_ID
consumer	PR	725	LATAM	568
consumer	PR	725	LATAM	3341
consumer	PR	680	LATAM	7459
consumer	PR	680	LATAM	7454
Home Office	PR	725	Pacific Asia	10740
Home Office	PR	725	Pacific Asia	5193
Home Office	PR	725	USCA	2092
Home Office	PR	725	USCA	9645
Home Office	PR	725	USCA	997
Home Office	PR	725	Africa	3339
Home Office	PR	725	Africa	1800

customer_segment 57 ×

Output

#	Time	Action	Message	Duration / Fetch
93	12:30:54	SELECT * FROM customer_segment LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
94	12:30:58	SELECT * FROM product LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
95	12:31:02	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
96	12:31:22	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
97	12:32:01	SELECT * FROM customer_segment LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	order_customerID	order_date	order_region	order_item_total	order_quantity
61558	568	2017-06-17	South America	269,980011	1
57472	3341	2017-04-18	South America	269,980011	1
59898	7459	2017-05-24	Central America	263,980011	1
59898	7459	2017-05-24	Central America	263,980011	1
58644	10740	2017-06-05	Central America	130,4900055	1
56882	2092	2017-10-04	Caribbean	254,9799957	1
51638	9645	2017-01-23	Central America	254,9799957	1
28744	9083	2017-02-24	South Asia	115,1800003	2
45461	4741	2017-10-25	West Asia	79,18000031	2
34773	11169	2017-05-22	US Center	95,98000336	2
45461	1098	2016-08-11	Central Africa	112,0700017	?

customer_order 58 ×

Result Grid

Filter Rows: Export: Wrap Cell Content:

order_id	order_customerID	order_date	order_region	order_item_total	order_quantity
61558	568	2017-06-17	South America	269,980011	1
57472	3341	2017-04-18	South America	269,980011	1
59898	7459	2017-05-24	Central America	263,980011	1
59898	7459	2017-05-24	Central America	263,980011	1
58644	10740	2017-06-05	Central America	130,4900055	1
56882	2092	2017-10-04	Caribbean	254,9799957	1
51638	9645	2017-01-23	Central America	254,9799957	1
28744	9083	2017-02-24	South Asia	115,1800003	2
45461	4741	2017-10-25	West Asia	79,18000031	2
34773	11169	2017-05-22	US Center	95,98000336	2
45461	1098	2016-08-11	Central Africa	112,0700017	?

Output

Action Output

#	Time	Action	Message	Duration / Fetch
94	12:30:58	SELECT * FROM product LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
95	12:31:02	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
96	12:31:22	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
97	12:32:01	SELECT * FROM customer_segment LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
98	12:32:21	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Navigator

SCHEMAS

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

e 10* **SQL File 11*** **SQL File 12*** **SQL File 16*** **appointments** **billing** **doctors** **SQL File 13*** **Employee Department DATA*** **SQL File 19*** **SQL File 20*** **SQL File 21*** **SQL File 22*** **SQL File 23***

79

```

80 • CREATE TABLE product (
81   product_price VARCHAR(250),
82   profit_price VARCHAR(200),
83   profit_perorder VARCHAR(200),
84   sales VARCHAR(250)
85 );
86 • INSERT INTO product
87   VALUES (299.980011,'26.10%',78.29000092,800),
88   (299.980011,'5.40%',-16.20000076,800),

```

Result Grid

product_price	profit_price	profit_perorder	sales
299.980011	26.10%	78.29000092	800
299.980011	5.40%	-16.20000076	800
149.9900055	-63.80%	-95.65000153	149.9900055
299.980011	23.80%	71.40000153	800
59.99000168	-25.60%	-30.75	119.9800034
50	33.60%	33.59999847	100
39.99000168	30.90%	24.69000053	79.98000336
54.97000122	23.00%	25.23999977	109.9400024
59.99000168	38.40%	46.0699997	119.9800034
99.98999786	44.10%	132.2899933	800
20.00000000	10.00%	11.00000001	110.0700012

product 59 X

Output

#	Time	Action	Message	Duration / Fetch
95	12:31:02	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
96	12:31:22	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
97	12:32:01	SELECT * FROM customer_segment LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
98	12:32:21	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
99	12:32:47	SELECT * FROM product LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

customer_data 60 X

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

customer_ID	category_name	customer_city	customer_count	customer_Fname
714	Camping	Mayaguez	Puerto Rico	Mark
997	Hiking	Mayaguez	Puerto Rico	Mary
2092	Camping	Mayaguez	Puerto Rico	Mary
3339	Hiking	Mayaguez	Puerto Rico	Mary
3341	Hiking	Caguas	Puerto Rico	Carol
4899	Camping	Mayaguez	Puerto Rico	Mary
5193	Camping	Mayaguez	Puerto Rico	Amanda
5421	Shop By Sport	Caguas	Puerto Rico	Elizabeth
7454	Hiking	Mayaguez	Puerto Rico	Mary
7459	Camping	Mayaguez	Puerto Rico	Mary
8224	Camping	Mayaguez	Puerto Rico	Mary

Output:

#	Time	Action	Message	Duration / Fetch
96	12:31:22	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
97	12:32:01	SELECT * FROM customer_segment LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
98	12:32:21	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
99	12:32:47	SELECT * FROM product LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
100	12:33:07	SELECT * FROM customer_data WHERE customer_id > 568 LIMIT 0, 1000	15 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

customer_data

```

103
104 •   SELECT *
105     FROM customer_data
106     WHERE customer_id > 568;
107
108 •   SELECT *
109     FROM customer_data
110     WHERE customer_id > 568 AND category_name = 'Hiking';
111
112 •   SELECT *

```

Result Grid

customer_ID	category_name	customer_city	customer_count	customer_Fname
997	Hiking	Mayaguez	Puerto Rico	Mary
3339	Hiking	Mayaguez	Puerto Rico	Mary
3341	Hiking	Caguas	Puerto Rico	Carol
7454	Hiking	Mayaguez	Puerto Rico	Mary
9645	Hiking	Mayaguez	Puerto Rico	Amanda
12322	Hiking	Mayaguez	Puerto Rico	Mary
HULL	HULL	HULL	HULL	HULL

Output

Action Output	#	Time	Action	Message	Duration / Fetch
97 12:32:01	SELECT * FROM customer_segment LIMIT 0, 1000			16 row(s) returned	0.000 sec / 0.000 sec
98 12:32:21	SELECT * FROM customer_order LIMIT 0, 1000			16 row(s) returned	0.000 sec / 0.000 sec
99 12:32:47	SELECT * FROM product LIMIT 0, 1000			32 row(s) returned	0.000 sec / 0.000 sec
100 12:33:07	SELECT * FROM customer_data WHERE customer_id > 568 LIMIT 0, 1000			15 row(s) returned	0.000 sec / 0.000 sec
101 12:33:24	SELECT * FROM customer_data WHERE customer_id > 568 AND category_name = 'Hiking' LIMIT 0, 1000			6 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SCHEMAS

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

customer_data 62

Output

#	Time	Action	Message	Duration / Fetch
98	12:32:21	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s)	0.000 sec / 0.000 sec
99	12:32:47	SELECT * FROM product LIMIT 0, 1000	32 row(s)	0.000 sec / 0.000 sec
100	12:33:07	SELECT * FROM customer_data WHERE customer_id > 568 LIMIT 0, 1000	15 row(s)	0.000 sec / 0.000 sec
101	12:33:24	SELECT * FROM customer_data WHERE customer_id > 568 AND category_name = 'Hiking' LIMIT 0, 1000	6 row(s)	0.000 sec / 0.000 sec

MySQL Workbench

SCHEMAS

- amazon
- amazon1
- amazon2
- customer
- customers
- data
- department**
 - departments
 - employees
 - salaries
 - salary_history**
 - Columns
 - emp_id
 - old_salary
 - new_salary

Administration Schemas

Information

Schema: department

```

31 •   SELECT e.emp_name, d.dept_name
32   FROM employees e
33   JOIN departments d ON e.dept_id = d.dept_id;
34
35
36 •   SELECT e.emp_name, s.salary
37   FROM employees e
38   JOIN salaries s ON e.emp_id = s.emp_id
39   WHERE s.salary > 50000;
40
41 •   SELECT d.dept_name, AVG(s.salary) AS avg_salary
42   FROM employees e
43   JOIN departments d ON e.dept_id = d.dept_id
44   JOIN salaries s ON e.emp_id = s.emp_id
45   GROUP BY d.dept_name;
46
47 •   SELECT emp_name
48   FROM employees
49   WHERE emp_id = (
50       SELECT emp_id
51       FROM salaries
52       WHERE salary = (SELECT MAX(salary) FROM salaries)
53

```

Output

#	Time	Action	Message	Duration / Fetch
71	12:55:22	SELECT d.dept_name, COUNT(e.emp_id) AS total_employees FROM departments d LEFT JOIN employees e ON d.dept_id = e.dept_id;	3 row(s) returned	0.016 sec / 0.000 sec
72	12:55:28	UPDATE salaries SET salary = 65000 WHERE emp_id = 101	0 row(s) affected Rows matched: 1 Changed: 0 Warnings: 0	0.016 sec
73	12:55:47	SELECT * FROM department.departments LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
74	12:56:23	SELECT * FROM department.employees LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
75	12:57:12	SELECT * FROM department.salaries LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SCHEMAS

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information ::::::::::::::::::::

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Code Editor

```

119 HAVING COUNT(*) > 1;
120
121 • SELECT sales, AVG(product_price) AS Total_productprice
122   FROM product
123   GROUP BY sales
124   HAVING AVG(product_price) > 50;
125
126 • SELECT profit_price, AVG(sales) AS Total_sales
127   FROM product
128   GROUP BY profit price

```

Result Grid

sales	Total_productprice
800	266.648342143334
149.9900055	149.9900055
119.9800034	59.99000168
109.9400024	54.97000122
179.9700012	59.99000168

Action Output

#	Time	Action	Message	Duration / Fetch
100	12:33:07	SELECT * FROM customer_data WHERE customer_id > 568 LIMIT 0, 1000	15 row(s) returned	0.000 sec / 0.000 sec
101	12:33:24	SELECT * FROM customer_data WHERE customer_id > 568 AND category_name = 'Hiking' LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
102	12:33:39	SELECT * FROM customer_data WHERE customer_id > 568 AND category_name = 'Camping' LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
103	12:34:39	SELECT category_name, COUNT(*) AS Total_Categoryname FROM customer_data GROUP BY category_name	2 row(s) returned	0.000 sec / 0.000 sec
104	12:34:52	SELECT sales, AVG(product_price) AS Total_productprice FROM product GROUP BY sales HAVING AVG(sales) > 1	5 row(s) returned	0.015 sec / 0.000 sec

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Code Editor

```

122 FROM product
123 GROUP BY sales
124 HAVING AVG(product_price) > 50;
125
126 • SELECT profit_price, AVG(sales) AS Total_sales
127 FROM product
128 GROUP BY profit_price
129 HAVING AVG(sales) > 100;
130
131 • SELECT profit_price, sales
  
```

Result Grid

profit_price	Total_sales
26.10%	800
5.40%	800
-63.80%	149.9900055
23.80%	800
-25.60%	119.9800034
23.00%	109.9400024
38.40%	119.9800034
44.10%	800
10.20%	119.9700012
-158.40%	179.9700012
??	110.0200012

Result 65 x

Output

#	Time	Action	Message	Duration / Fetch
101	12:33:24	SELECT * FROM customer_data WHERE customer_id > 568 AND category_name = 'Hiking' LIMIT 0, 1000	6 row(s) returned	0.000 sec / 0.000 sec
102	12:33:39	SELECT * FROM customer_data WHERE customer_id > 568 AND category_name = 'Camping' LIMIT 0, ...	4 row(s) returned	0.000 sec / 0.000 sec
103	12:34:39	SELECT category_name, COUNT(*) AS Total_Categories FROM customer_data GROUP BY category...	2 row(s) returned	0.000 sec / 0.000 sec
104	12:34:52	SELECT sales, AVG(product_price) AS Total_productprice FROM product GROUP BY sales HAVING AV...	5 row(s) returned	0.015 sec / 0.000 sec
105	12:35:17	SELECT profit_price, AVG(sales) AS Total_sales FROM product GROUP BY profit_price HAVING AVG(sa...	13 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SCHEMAS

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
- Stored Procedures
- Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

product 66 X

Result Grid

profit_price	sales
26.10%	800
5.40%	800
-63.80%	149.9900055
23.80%	800
-25.60%	119.9800034
33.60%	100
30.90%	79.98000336
23.00%	109.9400024
38.40%	119.9800034
44.10%	800
in more	110.0200012

product 66 X

Action Output

#	Time	Action	Message	Duration / Fetch
102	12:33:39	SELECT * FROM customer_data WHERE customer_id > 568 AND category_name = 'Camping' LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec
103	12:34:39	SELECT category_name, COUNT(*) AS Total_Categorynamne FROM customer_data GROUP BY category_name	2 row(s) returned	0.000 sec / 0.000 sec
104	12:34:52	SELECT sales, AVG(product_price) AS Total_productprice FROM product GROUP BY sales HAVING AVG(sales) > 100	5 row(s) returned	0.015 sec / 0.000 sec
105	12:35:17	SELECT profit_price, AVG(sales) AS Total_sales FROM product GROUP BY profit_price HAVING AVG(sales) > 100	13 row(s) returned	0.000 sec / 0.000 sec
106	12:35:29	SELECT profit_price, sales FROM product LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
- Stored Procedures
- Functions

customers

data

department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Code Editor

```

131 •   SELECT profit_price, sales
132     FROM product;
133
134 •   SELECT customer_ID, market
135     FROM customer_segment;
136
137 •   SELECT order_item_total, order_quantity
138     FROM customer_order;
139
140 •   SELECT category_name, customer_Fname
  
```

Result Grid

customer_ID	market
568	LATAM
3341	LATAM
7459	LATAM
7454	LATAM
10740	Pacific Asia
5193	Pacific Asia
2092	USCA
9645	USCA
997	USCA
3339	Africa
1000	Africa
customer_segment	67

Output

Action Output	#	Time	Action	Message	Duration / Fetch
103 12:34:39	SELECT category_name, COUNT(*) AS Total_Categoryname FROM customer_data GROUP BY category...	2 row(s) returned	0.000 sec / 0.000 sec		
104 12:34:52	SELECT sales, AVG(product_price) AS Total_productprice FROM product GROUP BY sales HAVING AVG(sa...	5 row(s) returned	0.015 sec / 0.000 sec		
105 12:35:17	SELECT profit_price, AVG(sales) AS Total_sales FROM product GROUP BY profit_price HAVING AVG(sa...	13 row(s) returned	0.000 sec / 0.000 sec		
106 12:35:29	SELECT profit_price, sales FROM product LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec		
107 12:35:46	SELECT customer_ID, market FROM customer_segment LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec		

Object Info Session

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Object Info Session

```

134 •   SELECT customer_ID, market
135     FROM customer_segment;
136
137 •   SELECT order_item_total, order_quantity
138     FROM customer_order;
139
140 •   SELECT category_name, customer_Fname
141     FROM customer_data;
142
143 •   SELECT COUNT(*)category name FROM customer data;

```

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

order_item_total	order_quantity
269.980011	1
269.980011	1
263.980011	1
263.980011	1
130.4900055	1
254.9799957	1
254.9799957	1
115.1800003	2
79.18000031	2
95.98000336	2
112.0700012	?
customer_order 68	x

Output

#	Time	Action	Message	Duration / Fetch
104	12:34:52	SELECT sales, AVG(product_price) AS Total_productprice FROM product GROUP BY sales HAVING AVG(sales) > 100	5 row(s) returned	0.015 sec / 0.000 sec
105	12:35:17	SELECT profit_price, AVG(sales) AS Total_sales FROM product GROUP BY profit_price HAVING AVG(sales) > 100	13 row(s) returned	0.000 sec / 0.000 sec
106	12:35:29	SELECT profit_price, sales FROM product LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
107	12:35:46	SELECT customer_ID, market FROM customer_segment LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
108	12:35:59	SELECT order_item_total, order_quantity FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

SCHEMAS

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

customer_data 69 ×

Code Editor

```

137 •   SELECT order_item_total, order_quantity
138   FROM customer_order;
139
140 •   SELECT category_name, customer_Fname
141   FROM customer_data;
142
143 •   SELECT COUNT(*)category_name FROM customer_data;
144 •   SELECT COUNT(*)order_quantity FROM customer_order;
145 •   SELECT COUNT(*)customer_state FROM customer_segment;
146 •   SELECT COUNT(*)profit perorder FROM product;

```

Result Grid

category_name	customer_Fname
Hiking	Sean
Camping	Mark
Hiking	Mary
Camping	Mary
Hiking	Mary
Hiking	Carol
Camping	Mary
Camping	Amanda
Shop By Sport	Elizabeth
Hiking	Mary

Action Output

#	Time	Action	Message	Duration / Fetch
105	12:35:17	SELECT profit_price, AVG(sales) AS Total_sales FROM product GROUP BY profit_price HAVING AVG(sa... 13 row(s) returned	13 row(s) returned	0.000 sec / 0.000 sec
106	12:35:29	SELECT profit_price, sales FROM product LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
107	12:35:46	SELECT customer_ID, market FROM customer_segment LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
108	12:35:59	SELECT order_item_total, order_quantity FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
109	12:36:11	SELECT category_name, customer_Fname FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

	order_id	order_customerID	order_date	order_Region	order_item_total	order_quantity
	varchar(250)	varchar(200)	date	varchar(250)	varchar(200)	varchar(200)

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor | Field Types | Read Only

```

141   FROM customer_data;
142
143 •   SELECT COUNT(*)category_name FROM customer_data;
144 •   SELECT COUNT(*)order_quantity FROM customer_order;
145 •   SELECT COUNT(*)customer_state FROM customer_segment;
146 •   SELECT COUNT(*)profit_perorder FROM product;
147 •   SELECT COUNT(*) AS sales FROM customer_data;
148 •   SELECT COUNT(*) AS profit_price FROM customer_data;
149 •   SELECT DISTINCT sales FROM product;
150 •   SELECT DISTINCT profit price FROM product;

```

category_name

16

Result 70 x

Output

#	Time	Action	Message	Duration / Fetch
106	12:35:29	SELECT profit_price, sales FROM product LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
107	12:35:46	SELECT customer_ID, market FROM customer_segment LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
108	12:35:59	SELECT order_item_total, order_quantity FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
109	12:36:11	SELECT category_name, customer_Fname FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
110	12:36:41	SELECT COUNT(*)category_name FROM customer_data LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec

Object Info Session

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor | Field Types | Read Only

```

141   FROM customer_data;
142
143 •   SELECT COUNT(*)category_name FROM customer_data;
144 •   SELECT COUNT(*)order_quantity FROM customer_order;
145 •   SELECT COUNT(*)customer_state FROM customer_segment;
146 •   SELECT COUNT(*)profit_perorder FROM product;
147 •   SELECT COUNT(*) AS sales FROM customer_data;
148 •   SELECT COUNT(*) AS profit_price FROM customer_data;
149 •   SELECT DISTINCT sales FROM product;
150 •   SELECT DISTINCT profit price FROM product;
  
```

order_quantity
16

Result 71 x

Action Output

#	Time	Action	Message	Duration / Fetch
107	12:35:46	SELECT customer_ID, market FROM customer_segment LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
108	12:35:59	SELECT order_item_total, order_quantity FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
109	12:36:11	SELECT category_name, customer_Fname FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
110	12:36:41	SELECT COUNT(*)category_name FROM customer_data LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
111	12:36:55	SELECT COUNT(*)order_quantity FROM customer_order LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Result 72 x

Output Action Output

#	Time	Action	Message	Duration / Fetch
108	12:35:59	SELECT order_item_total, order_quantity FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
109	12:36:11	SELECT category_name, customer_Fname FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
110	12:36:41	SELECT COUNT(*)category_name FROM customer_data LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
111	12:36:55	SELECT COUNT(*)order_quantity FROM customer_order LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
112	12:37:07	SELECT COUNT(*)customer_state FROM customer_segment LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec

Object Info Session

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

	order_id	order_customerID	order_date	order_Region	order_item_total	order_quantity
	varchar(250)	varchar(200)	date	varchar(250)	varchar(200)	varchar(200)

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

```

141   FROM customer_data;
142
143 •   SELECT COUNT(*)category_name FROM customer_data;
144 •   SELECT COUNT(*)order_quantity FROM customer_order;
145 •   SELECT COUNT(*)customer_state FROM customer_segment;
146 •   SELECT COUNT(*)profit_perorder FROM product;
147 •   SELECT COUNT(*) AS sales FROM customer_data;
148 •   SELECT COUNT(*) AS profit_price FROM customer_data;
149 •   SELECT DISTINCT sales FROM product;
150 •   SELECT DISTINCT profit price FROM product;

```

profit_perorder

32

Result 73 x

Action Output

#	Time	Action	Message	Duration / Fetch
109	12:36:11	SELECT category_name, customer_Fname FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
110	12:36:41	SELECT COUNT(*)category_name FROM customer_data LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
111	12:36:55	SELECT COUNT(*)order_quantity FROM customer_order LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
112	12:37:07	SELECT COUNT(*)customer_state FROM customer_segment LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
113	12:37:17	SELECT COUNT(*)profit_perorder FROM product LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec

Object Info Session Read Only

Schemas

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
- Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
- Views
- Stored Procedures
- Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor | Field Types | Read Only

```

141   FROM customer_data;
142
143 •   SELECT COUNT(*)category_name FROM customer_data;
144 •   SELECT COUNT(*)order_quantity FROM customer_order;
145 •   SELECT COUNT(*)customer_state FROM customer_segments;
146 •   SELECT COUNT(*)profit_perorder FROM product;
147 •   SELECT COUNT(*) AS sales FROM customer_data;
148 •   SELECT COUNT(*) AS profit_price FROM customer_data;
149 •   SELECT DISTINCT sales FROM product;
150 •   SELECT DISTINCT profit price FROM product;

```

Result 74 x

Action Output

#	Time	Action	Message	Duration / Fetch
110	12:36:41	SELECT COUNT(*)category_name FROM customer_data LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
111	12:36:55	SELECT COUNT(*)order_quantity FROM customer_order LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
112	12:37:07	SELECT COUNT(*)customer_state FROM customer_segments LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
113	12:37:17	SELECT COUNT(*)profit_perorder FROM product LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
114	12:37:43	SELECT COUNT(*) AS sales FROM customer_data LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Result 75 x Read Only

Output

Action Output	#	Time	Action	Message	Duration / Fetch
111 12:36:55	SELECT COUNT("order_quantity") FROM customer_order LIMIT 0, 1000			1 row(s) returned	0.000 sec / 0.000 sec
112 12:37:07	SELECT COUNT("customer_state") FROM customer_segment LIMIT 0, 1000			1 row(s) returned	0.015 sec / 0.000 sec
113 12:37:17	SELECT COUNT("profit_perorder") FROM product LIMIT 0, 1000			1 row(s) returned	0.015 sec / 0.000 sec
114 12:37:43	SELECT COUNT("AS sales") FROM customer_data LIMIT 0, 1000			1 row(s) returned	0.000 sec / 0.000 sec
115 12:38:00	SELECT COUNT("AS profit_price") FROM customer_data LIMIT 0, 1000			1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Result 75 x Read Only

Action Output

#	Time	Action	Message	Duration / Fetch
111	12:36:55	SELECT COUNT("order_quantity") FROM customer_order LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
112	12:37:07	SELECT COUNT("customer_state") FROM customer_segment LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
113	12:37:17	SELECT COUNT("profit_perorder") FROM product LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
114	12:37:43	SELECT COUNT("sales") AS sales FROM customer_data LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
115	12:38:00	SELECT COUNT("profit_price") AS profit_price FROM customer_data LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
- Stored Procedures
- Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Code Editor

```

144 •   SELECT COUNT(*) AS order_quantity FROM customer_order;
145 •   SELECT COUNT(*) AS customer_state FROM customer_segment;
146 •   SELECT COUNT(*) AS profit_perorder FROM product;
147 •   SELECT COUNT(*) AS sales FROM customer_data;
148 •   SELECT COUNT(*) AS profit_price FROM customer_data;
149 •   SELECT DISTINCT sales FROM product;
150 •   SELECT DISTINCT profit_price FROM product;
151
152 DELIMITER //
```

Result Grid

sales
800
149.9900055
119.9800034
100
79.98000336
109.9400024
119.9700012
179.9700012

product 76 x

Output

Action Output	#	Time	Action	Message	Duration / Fetch
112 12:37:07 SELECT COUNT(*) AS customer_state FROM customer_segment LIMIT 0, 1000	112	12:37:07	SELECT COUNT(*) AS customer_state FROM customer_segment LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
113 12:37:17 SELECT COUNT(*) AS profit_perorder FROM product LIMIT 0, 1000	113	12:37:17	SELECT COUNT(*) AS profit_perorder FROM product LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
114 12:37:43 SELECT COUNT(*) AS sales FROM customer_data LIMIT 0, 1000	114	12:37:43	SELECT COUNT(*) AS sales FROM customer_data LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
115 12:38:00 SELECT COUNT(*) AS profit_price FROM customer_data LIMIT 0, 1000	115	12:38:00	SELECT COUNT(*) AS profit_price FROM customer_data LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
116 12:38:17 SELECT DISTINCT sales FROM product LIMIT 0, 1000	116	12:38:17	SELECT DISTINCT sales FROM product LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Object Info Session

```

144 •   SELECT COUNT(*) AS profit_perorder FROM product;
145 •   SELECT COUNT(*) AS sales FROM customer_data;
146 •   SELECT COUNT(*) AS profit_price FROM customer_data;
147 •   SELECT COUNT(*) AS profit_price FROM product;
148 •   SELECT COUNT(*) AS profit_price FROM customer_data;
149 •   SELECT DISTINCT sales FROM product;
150 •   SELECT DISTINCT profit_price FROM product;
151
152 DELIMITER //
153

```

Result Grid

Filter Rows: Export: Wrap Cell Content:

profit_price
26.10%
5.40%
-63.80%
23.80%
-25.60%
33.60%
30.90%
23.00%
38.40%
44.10%
10.20%

product 77 ×

Output

Action Output

#	Time	Action	Message	Duration / Fetch
113	12:37:17	SELECT COUNT(*) AS profit_perorder FROM product LIMIT 0, 1000	1 row(s) returned	0.015 sec / 0.000 sec
114	12:37:43	SELECT COUNT(*) AS sales FROM customer_data LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
115	12:38:00	SELECT COUNT(*) AS profit_price FROM customer_data LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
116	12:38:17	SELECT DISTINCT sales FROM product LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
117	12:38:29	SELECT DISTINCT profit_price FROM product LIMIT 0, 1000	15 row(s) returned	0.015 sec / 0.000 sec

Result Grid Form Editor Field Types Read Only

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
- Stored Procedures
- Functions

customers

data

department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Code Editor

```

169 • SELECT
170   cd.customer_Fname,
171   cd.customer_city,
172   cs.customer_segment,
173   cs.market
174   FROM
175     customer_data cd
176   INNER JOIN
177     customer_segment cs ON cd.customer_ID = cs.customer_ID;
178

```

Result Grid

customer_Fname	customer_city	customer_segment	market
Sean	Caguas	consumer	LATAM
Mark	Mayaguez	consumer	LATAM
Mary	Mayaguez	Home Office	USCA
Mary	Mayaguez	Home Office	USCA
Mary	Mayaguez	Home Office	Africa
Carol	Caguas	consumer	LATAM
Mary	Mayaguez	Home Office	Africa
Amanda	Mayaguez	Home Office	Pacific Asia
Elizabeth	Caguas	Home Office	Europe
Mary	Mayaguez	consumer	LATAM
Mary	Mayaguez	consumer	LATAM

Result 78 x

Output

#	Time	Action	Message	Duration / Fetch
114	12:37:43	SELECT COUNT(*) AS sales FROM customer_data LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
115	12:38:00	SELECT COUNT(*) AS profit_price FROM customer_data LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
116	12:38:17	SELECT DISTINCT sales FROM product LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
117	12:38:29	SELECT DISTINCT profit_price FROM product LIMIT 0, 1000	15 row(s) returned	0.015 sec / 0.000 sec
118	12:38:46	SELECT cd.customer_Fname, cd.customer_city, cs.customer_segment, cs.market FROM ...	16 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Navigator e 10* SQL File 11* SQL File 12* SQL File 16* appointments billing doctors SQL File 13* Employee Department DATA* SQL File 19* SQL File 20* SQL File 22* SQL File 22* ×

SCHEMAS

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

```

190     cd.customer_fname,
191     cd.customer_city,
192     cs.customer_segment,
193     cs.market
194   FROM
195     customer_data cd
196   RIGHT JOIN
197     customer_segment cs ON cd.customer_ID = cs.customer_ID;
198
199   SELECT MAX(order quantity) FROM customer order;

```

Result Grid

customer_Fname	customer_cty	customer_segment	market
Sean	Caguas	consumer	LATAM
Carol	Caguas	consumer	LATAM
Mary	Mayaguez	consumer	LATAM
Mary	Mayaguez	consumer	LATAM
Mary	Caguas	Home Office	Pacific Asia
Ananda	Mayaguez	Home Office	Pacific Asia
Mary	Mayaguez	Home Office	USCA
Ananda	Mayaguez	Home Office	USCA
Mary	Mayaguez	Home Office	USCA
Mary	Mayaguez	Home Office	Africa
Mario	Mayaguez	Home Office	Africa

Result 80 x

Output

#	Time	Action	Message	Duration / Fetch
116	12:38:17	SELECT DISTINCT sales FROM product LIMIT 0, 1000	8 row(s) returned	0.000 sec / 0.000 sec
117	12:38:29	SELECT DISTINCT profit_price FROM product LIMIT 0, 1000	15 row(s) returned	0.015 sec / 0.000 sec
118	12:38:46	SELECT cd.customer_fname, cd.customer_city, cs.customer_segment, cs.market FROM ...	16 row(s) returned	0.000 sec / 0.000 sec
119	12:39:12	SELECT cd.customer_fname, cd.customer_city, cs.customer_segment, cs.market FROM ...	16 row(s) returned	0.000 sec / 0.000 sec
120	12:39:33	SELECT cd.customer_fname, cd.customer_city, cs.customer_segment, cs.market FROM ...	16 row(s) returned	0.000 sec / 0.000 sec

Result Grid Form Editor Field Types Read Only

Schemas

```

193     cs.market
194     FROM
195     customer_data cd
196     RIGHT JOIN
197     customer_segment cs ON cd.customer_ID = cs.customer_ID;
198
199     SELECT MAX(order_quantity) FROM customer_order;
200     SELECT MIN(order_quantity) FROM customer_order;
201     SELECT MAX(sales) FROM product;
202     SELECT MIN(sales) FROM product;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Result Grid | Form Editor | Field Types | Read Only

	MAX(order_quantity)
▶ 3	3

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Action Output

#	Time	Action	Message	Duration / Fetch
117	12:38:29	SELECT DISTINCT profit_price FROM product LIMIT 0, 1000	15 row(s) returned	0.015 sec / 0.000 sec
118	12:38:46	SELECT cd.customer_fname, cd.customer_city, cs.customer_segment, cs.market FROM ...	16 row(s) returned	0.000 sec / 0.000 sec
119	12:39:12	SELECT cd.customer_fname, cd.customer_city, cs.customer_segment, cs.market FROM ...	16 row(s) returned	0.000 sec / 0.000 sec
120	12:39:33	SELECT cd.customer_fname, cd.customer_city, cs.customer_segment, cs.market FROM ...	16 row(s) returned	0.000 sec / 0.000 sec
121	12:39:47	SELECT MAX(order_quantity) FROM customer_order LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

The screenshot shows a SQL Server Management Studio (SSMS) interface. The left sidebar displays the database schema, including tables like customer_data, customer_order, and product. The main area shows a query window with the following code:

```
e 10* SQL File 11* SQL File 12* SQL File 16* appointments billing doctors SQL File 13* Employee Department DATA* SQL File 19* SQL File 20* SQL File 22* SQL File 22* x +\n\n193     cs.market\n194     FROM\n195         customer_data cd\n196     RIGHT JOIN\n197         customer_segment cs ON cd.customer_ID = cs.customer_ID;\n198\n199     SELECT MAX(order_quantity) FROM customer_order;\n200     SELECT MIN(order_quantity) FROM customer_order;\n201     SELECT MAX(sales) FROM product;\n202     SELECT MIN(sales) FROM product;
```

The results grid shows the output of the first query:

MIN(order_quantity)
1

The bottom pane shows the execution history:

Action Output
Time Action
118 12:38:46 SELECT cd.customer_Fname, cd.customer_Lname, cd.customer_email, cd.customer_city, cd.customer_segment, cs.market
119 12:39:12 SELECT cd.customer_Fname, cd.customer_Lname, cd.customer_email, cd.customer_city, cd.customer_segment, cs.market
120 12:39:33 SELECT cd.customer_Fname, cd.customer_Lname, cd.customer_email, cd.customer_city, cd.customer_segment, cs.market
121 12:39:47 SELECT MAX(order_quantity) FROM customer_order
122 12:40:02 SELECT MIN(order_quantity) FROM customer_order

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department
- Administration Schemas

Information

Table: customer_order

Columns:

	order_id	varchar(250)
order_customerID	varchar(200)	
order_date	date	
order_Region	varchar(250)	
order_item_total	varchar(200)	
order_quantity	varchar(200)	

Result 83 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
119	12:39:12	SELECT cd.customer_Fname, cd.customer_city, cs.customer_segment, cs.market FROM ...	16 row(s) returned	0.000 sec / 0.000 sec
120	12:39:33	SELECT cd.customer_Fname, cd.customer_city, cs.customer_segment, cs.market FROM ...	16 row(s) returned	0.000 sec / 0.000 sec
121	12:39:47	SELECT MAX(order_quantity) FROM customer_order LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
122	12:40:02	SELECT MIN(order_quantity) FROM customer_order LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
123	12:40:16	SELECT MAX(sales) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Schemas

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

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

```

199  SELECT MAX(order_quantity) FROM customer_order;
200  SELECT MIN(order_quantity) FROM customer_order;
201  SELECT MAX(sales) FROM product;
202  SELECT MIN(sales) FROM product;
203  SELECT AVG(sales) FROM product;
204  SELECT MAX(profit_price) FROM product;
205  SELECT MIN(profit_price) FROM product;
206  SELECT AVG(profit_price) FROM product;
207
208  SELECT *
  
```

Result 84 x

Output

#	Time	Action	Message	Duration / Fetch
120	12:39:33	SELECT cd.customer_fname, cd.customer_city, cs.customer_segment, cs.market FROM ...	16 row(s) returned	0.000 sec / 0.000 sec
121	12:39:47	SELECT MAX(order_quantity) FROM customer_order LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
122	12:40:02	SELECT MIN(order_quantity) FROM customer_order LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
123	12:40:16	SELECT MAX(sales) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
124	12:40:39	SELECT MIN(sales) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

SCHEMAS

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Result Grid

```

199  SELECT MAX(order_quantity) FROM customer_order;
200  SELECT MIN(order_quantity) FROM customer_order;
201  SELECT MAX(sales) FROM product;
202  SELECT MIN(sales) FROM product;
203  SELECT AVG(sales) FROM product;
204  SELECT MAX(profit_price) FROM product;
205  SELECT MIN(profit_price) FROM product;
206  SELECT AVG(profit_price) FROM product;
207
208  SELECT *

```

Filter Rows: Export: Wrap Cell Content:

AVG(sales)
379.98437642874995

Result 85 x

Output

Action Output	#	Time	Action	Message	Duration / Fetch
121 12:39:47 SELECT MAX(order_quantity) FROM customer_order LIMIT 0, 1000	121	12:39:47	SELECT MAX(order_quantity) FROM customer_order LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
122 12:40:02 SELECT MIN(order_quantity) FROM customer_order LIMIT 0, 1000	122	12:40:02	SELECT MIN(order_quantity) FROM customer_order LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
123 12:40:16 SELECT MAX(sales) FROM product LIMIT 0, 1000	123	12:40:16	SELECT MAX(sales) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
124 12:40:29 SELECT MIN(sales) FROM product LIMIT 0, 1000	124	12:40:29	SELECT MIN(sales) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
125 12:40:41 SELECT AVG(sales) FROM product LIMIT 0, 1000	125	12:40:41	SELECT AVG(sales) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Result 86 x

Action Output

#	Time	Action	Message	Duration / Fetch
122	12:40:02	SELECT MIN(order_quantity) FROM customer_order LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
123	12:40:16	SELECT MAX(sales) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
124	12:40:29	SELECT MIN(sales) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
125	12:40:41	SELECT AVG(sales) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
126	12:40:52	SELECT MAX(profit_price) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

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

MAX(profit_price)

5.40%

Read Only

SCHEMAS

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

```

208 SELECT *
209 FROM product
210 ORDER BY sales DESC;
211
212 SELECT *
213 FROM product
214 ORDER BY profit_price;
215
216 SELECT * FROM customer_data;
217 SELECT * FROM customer_segment;

```

Result Grid

product_price	profit_price	profit_perorder	sales
59.99000168	-158.40%	-285.0700073	179.9700012
59.99000168	-158.40%	-285.0700073	179.9700012
59.99000168	-25.60%	-30.75	119.9800034
59.99000168	-25.60%	-30.75	119.9800034
149.9900055	-63.80%	-95.65000153	149.9900055
149.9900055	-63.80%	-95.65000153	149.9900055
299.980011	-8.30%	-24.89999962	800
299.980011	-8.30%	-24.89999962	800
299.980011	-8.30%	-24.89999962	800
299.980011	-8.30%	-24.89999962	800
30.00000168	10.20%	12.10000081	110.0200012

product 90 X

Action Output

#	Time	Action	Message	Duration / Fetch
126	12:40:52	SELECT MAX(profit_price) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
127	12:41:03	SELECT MIN(profit_price) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
128	12:41:16	SELECT AVG(profit_price) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
129	12:41:29	SELECT * FROM product ORDER BY sales DESC LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
130	12:41:56	SELECT * FROM product ORDER BY profit_price LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

customer_data 91 X

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

customer_ID	category_name	customer_city	customer_count	customer_Fname
568	Hiking	Caguas	Puerto Rico	Sean
714	Camping	Mayaguez	Puerto Rico	Mark
997	Hiking	Mayaguez	Puerto Rico	Mary
2092	Camping	Mayaguez	Puerto Rico	Mary
3339	Hiking	Mayaguez	Puerto Rico	Mary
3341	Hiking	Caguas	Puerto Rico	Carol
4899	Camping	Mayaguez	Puerto Rico	Mary
5193	Camping	Mayaguez	Puerto Rico	Elizabeth
5421	Shop By Sport	Caguas	Puerto Rico	Elizabeth
7454	Hiking	Mayaguez	Puerto Rico	Mary
7460	Camping	Manatí	Puerto Rico	Mary

customer_data 91 X

Output ::::

#	Time	Action	Message	Duration / Fetch
127	12:41:03	SELECT MIN(profit_price) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
128	12:41:16	SELECT AVG(profit_price) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
129	12:41:29	SELECT * FROM product ORDER BY sales DESC LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
130	12:41:56	SELECT * FROM product ORDER BY profit_price LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
131	12:42:10	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
 - Stored Procedures
 - Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

customer_data 91 X

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: |

customer_ID	category_name	customer_city	customer_count	customer_Fname
568	Hiking	Caguas	Puerto Rico	Sean
714	Camping	Mayaguez	Puerto Rico	Mark
997	Hiking	Mayaguez	Puerto Rico	Mary
2092	Camping	Mayaguez	Puerto Rico	Mary
3339	Hiking	Mayaguez	Puerto Rico	Mary
3341	Hiking	Caguas	Puerto Rico	Carol
4899	Camping	Mayaguez	Puerto Rico	Mary
5193	Camping	Mayaguez	Puerto Rico	Elizabeth
5421	Shop By Sport	Caguas	Puerto Rico	Elizabeth
7454	Hiking	Mayaguez	Puerto Rico	Mary
7460	Camping	Manatí	Puerto Rico	Mary

customer_data 91 X

Output ::::

#	Time	Action	Message	Duration / Fetch
127	12:41:03	SELECT MIN(profit_price) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
128	12:41:16	SELECT AVG(profit_price) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
129	12:41:29	SELECT * FROM product ORDER BY sales DESC LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
130	12:41:56	SELECT * FROM product ORDER BY profit_price LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
131	12:42:10	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

SCHEMAS

Filter objects

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
- Stored Procedures
- Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

Code Editor

```

211
212     SELECT *
213     FROM product
214     ORDER BY profit_price;
215
216     SELECT * FROM customer_data;
217     SELECT * FROM customer_segment;
218     SELECT * FROM customer_order;
219     SELECT * FROM product;
220
  
```

Result Grid

customer_segment	customer_state	customer_zipcode	market	customer_ID
consumer	PR	725	LATAM	568
consumer	PR	725	LATAM	3341
consumer	PR	680	LATAM	7459
consumer	PR	680	LATAM	7454
Home Office	PR	725	Pacific Asia	10740
Home Office	PR	725	Pacific Asia	5193
Home Office	PR	725	USCA	2092
Home Office	PR	725	USCA	9645
Home Office	PR	725	USCA	997
Home Office	PR	725	Africa	3339
Home Office	PR	725	Africa	10000

Output

#	Time	Action	Message	Duration / Fetch
128	12:41:16	SELECT AVG(profit_price) FROM product LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
129	12:41:29	SELECT * FROM product ORDER BY sales DESC LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
130	12:41:56	SELECT * FROM product ORDER BY profit_price LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
131	12:42:10	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
132	12:42:22	SELECT * FROM customer_segment LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Result Grid Form Editor Field Types Read Only

SCHEMAS

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
- Stored Procedures
- Functions

customers

data

department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

```

211
212     SELECT *
213     FROM product
214     ORDER BY profit_price;
215
216     SELECT * FROM customer_data;
217     SELECT * FROM customer_segment;
218     SELECT * FROM customer_orders;
219     SELECT * FROM product;
220

```

Result Grid

order_id	order_customerID	order_date	order_Region	order_item_total	order_quantity
61558	568	2017-06-17	South America	269.980011	1
57472	3341	2017-04-18	South America	269.980011	1
59898	7459	2017-05-24	Central America	263.980011	1
59898	7459	2017-05-24	Central America	263.980011	1
58644	10740	2017-06-05	Central America	130.4900055	1
56882	2092	2017-10-04	Caribbean	254.9799957	1
51638	9645	2017-01-23	Central America	254.9799957	1
28744	9083	2017-02-24	South Asia	115.1800003	2
45461	4741	2017-10-25	West Asia	79.18000031	2
34773	11169	2017-05-22	US Center	95.9800036	2
ACAA1	105E	2016-09-11	Central Africa	112.0700017	2

Output

#	Time	Action	Message	Duration / Fetch
129	12:41:29	SELECT * FROM product ORDER BY sales DESC LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
130	12:41:56	SELECT * FROM product ORDER BY profit_price LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
131	12:42:10	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
132	12:42:22	SELECT * FROM customer_segment LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
133	12:42:39	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

Schemas

- amazon
- amazon1
- amazon2
- customer
- customer_data**
 - Tables
 - customer_data
 - customer_order
 - customer_segment
 - product
 - Views
- Stored Procedures
- Functions
- customers
- data
- department

Administration Schemas

Information

Table: customer_order

Columns:

order_id	varchar(250)
order_customerID	varchar(200)
order_date	date
order_Region	varchar(250)
order_item_total	varchar(200)
order_quantity	varchar(200)

```

211
212     SELECT *
213     FROM product
214     ORDER BY profit_price;
215
216     SELECT * FROM customer_data;
217     SELECT * FROM customer_segment;
218     SELECT * FROM customer_order;
219     SELECT * FROM products;
220

```

Result Grid

product_price	profit_price	profit_perorder	sales
299.980011	26.10%	78.29000092	800
299.980011	5.40%	-16.20000076	800
149.9900055	-63.80%	-95.65000153	149.9900055
299.980011	23.80%	71.40000153	800
59.99000168	-25.60%	-30.75	119.9800034
50	33.60%	33.59999847	100
39.99000168	30.90%	24.69000053	79.98000336
54.97000122	23.00%	25.23999977	109.9400024
59.99000168	38.40%	46.0699997	119.9800034
99.98999786	44.10%	132.2899933	800
an nnnnnnnn	10.70%	12.10000081	110.6300012

product 94 X

Output

#	Time	Action	Message	Duration / Fetch
130	12:41:56	SELECT * FROM product ORDER BY profit_price LIMIT 0, 1000	32 row(s) returned	0.000 sec / 0.000 sec
131	12:42:10	SELECT * FROM customer_data LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
132	12:42:22	SELECT * FROM customer_segment LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
133	12:42:39	SELECT * FROM customer_order LIMIT 0, 1000	16 row(s) returned	0.000 sec / 0.000 sec
134	12:42:51	SELECT * FROM products LIMIT 0, 1000	Copilot 32 row(s) returned	0.000 sec / 0.000 sec

Object Info Session