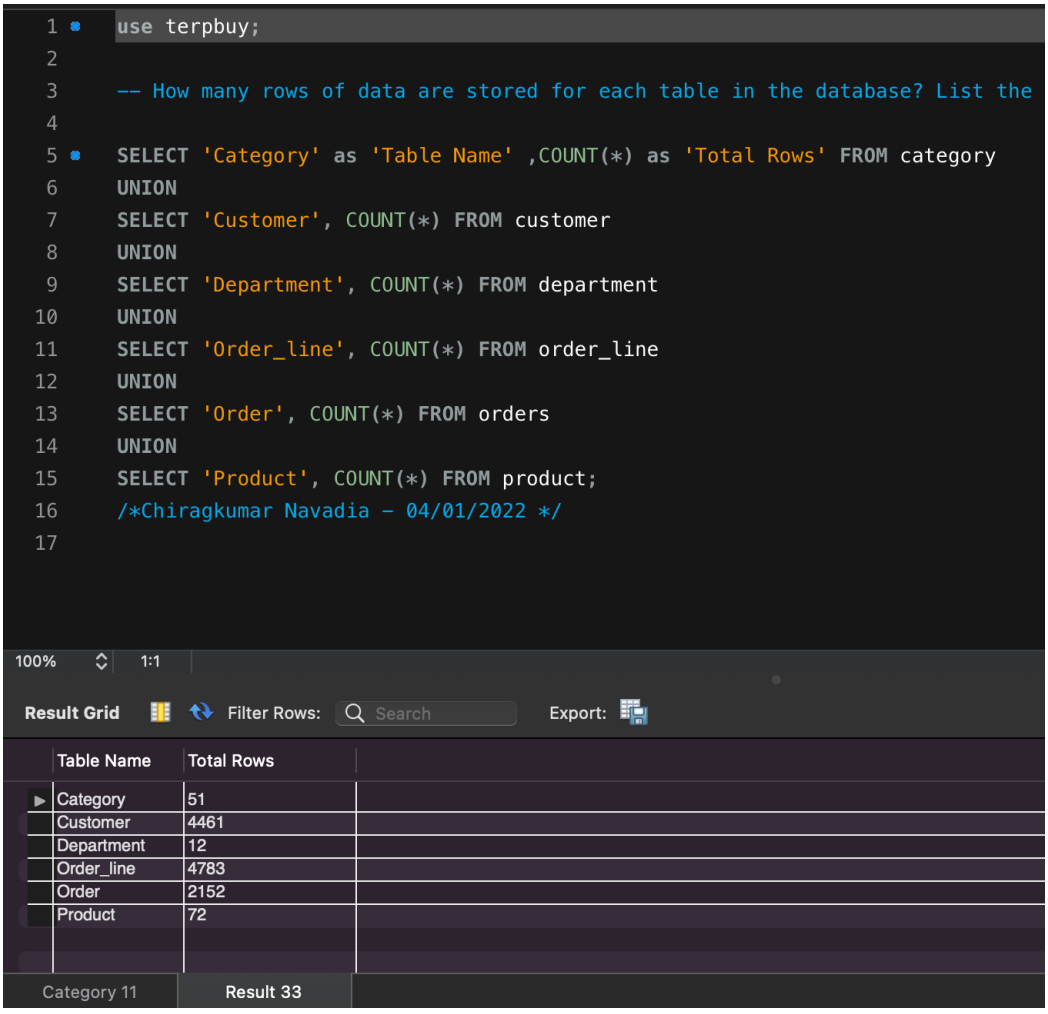
TERPBUY PROJECT ASSIGNMENT

MySQL Queries

By Chiragkumar Navadia

Query 1: How many rows of data are stored for each table in the database? List the name of each table followed by the number of rows it has.

```
SELECT 'Category' as 'Table Name' ,COUNT(*) as 'Total Rows' FROM category
UNION
SELECT 'Customer', COUNT(*) FROM customer
UNION
SELECT 'Department', COUNT(*) FROM department
UNION
SELECT 'Order_line', COUNT(*) FROM order_line
UNION
SELECT 'Order', COUNT(*) FROM orders
UNION
SELECT 'Product', COUNT(*) FROM product;
```



Query # 1 Snap Shot

Query 2: Which products are considered high-priced products? A high-priced product has a price exceeding $100.00. List the names and prices of the high-priced products.

SELECT product_name, product_price
FROM product
WHERE product_price > 100;

```
18    -- Query 2: Which products are considered high-priced products? A high-priced product has a price exceeding $100.00.
19    -- List the names and prices of the high-priced products.
20
21    SELECT product_name, product_price
22    FROM product
23    WHERE product_price > 100;
24
25    /*Chiragkumar Navadia - 04/01/2022 */
26
27
28
29
```

100% 1:27

Result Grid | Filter Rows: Search | Export:

product_name	produ...
Nike Mens CJ Elite 2 TD Football Cleat	129.99
Diamondback Womens Serene Classic Comfort Bi	299.98
Field & Stream Sportsman 16 Gun Fire Safe	399.98
Pelican Sunstream 100 Kayak	199.99
Web Camera	452.04
Childrens heaters	357.10
Dell Laptop	1500.00
Industrial CONSUMER electronics	252.88

Result Grid
Form Editor

Query 3: List all orders placed by customers in the state of Florida. Note: The state abbreviation for Florida is 'FL'. Include the customers' first names, last names, city, and segment, along with the order ID and order date.

```
SELECT c.first_name, c.last_name, c.city, c.segment, o.order_id, o.order_date, c.state
FROM customer c
INNER JOIN orders o ON o.customer_id = c.customer_id
WHERE State = 'FL';
```

```
26
27  /* Query 3: List all orders placed by customers in the state of Florida.
28  Note: The state abbreviation for Florida is 'FL'.
29  Include the customers' first names, last names, city, and segment, along with the order ID and order date.*/
30
31  SELECT c.first_name, c.last_name, c.city, c.segment, o.order_id, o.order_date, c.state
32  FROM customer c
33  INNER JOIN orders o ON o.customer_id = c.customer_id
34  WHERE State = 'FL';
35
36  /*Chiragkumar Navadia - 04/01/2022 */
37
38
```

100% 38:36

Result Grid | Filter Rows: Search | Export:

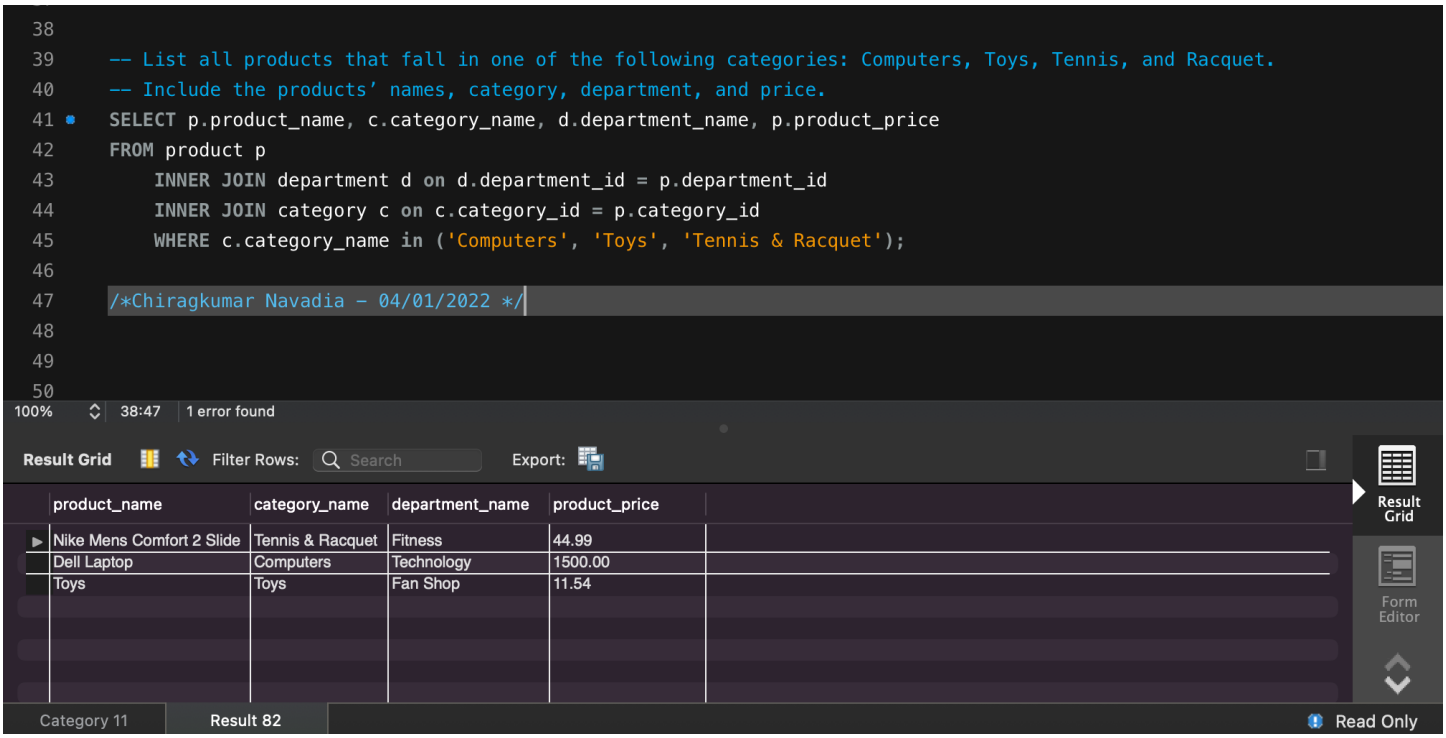
first_name	last_name	city	segment	order_id	order_date	state
Laura	Smith	Winter Park	CORPORATE	20366	2018-10-24	FL
Linda	Murray	Pompano Beach	CORPORATE	20428	2018-10-25	FL
Mary	Smith	Tallahassee	CORPORATE	20492	2018-10-26	FL
Mary	Morrison	Brandon	HOME_OFFICE	20745	2018-10-29	FL
Jose	Smith	Miami	CORPORATE	20877	2018-10-31	FL
Patricia	Smith	Fort Lauderdale	CORPORATE	21239	2018-11-06	FL
Mary	Harris	Miami	CORPORATE	21278	2018-11-06	FL
Mary	Weaver	Miami	CONSUMER	22082	2018-11-18	FL

Result Grid | Form Editor

Category 11 | Result 39 | Read Only

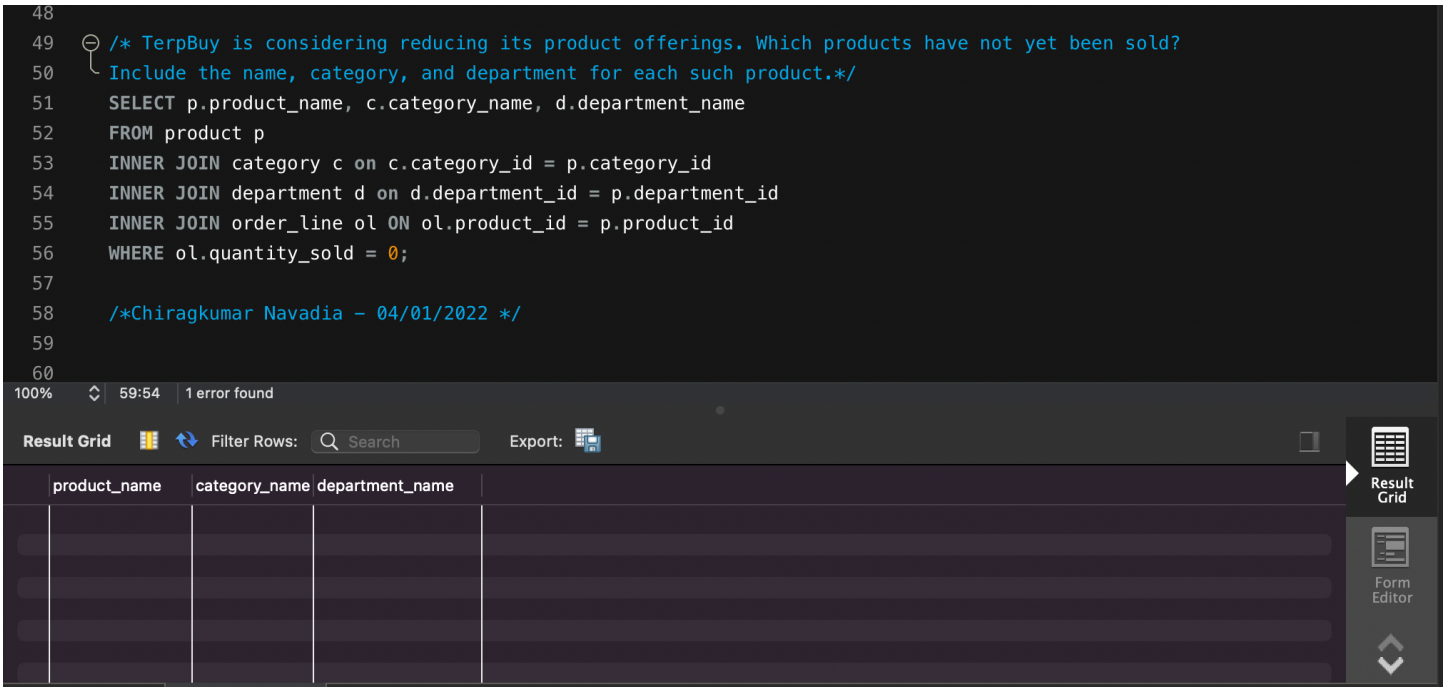
Query 4: List all products that fall in one of the following categories: Computers, Toys, Tennis, and Racquet. Include the products' names, category, department, and price.

```
SELECT p.product_name, c.category_name, d.department_name, p.product_price
FROM product p
        INNER JOIN department d on d.department_id = p.department_id
        INNER JOIN category c on c.category_id = p.category_id
        WHERE c.category_name in ('Computers', 'Toys', 'Tennis & Racquet');
```



Query 5: TerpBuy is considering reducing its product offerings. Which products have not yet been sold? Include the name, category, and department for each such product.

```
SELECT p.product_name, c.category_name, d.department_name
FROM product p
INNER JOIN category c on c.category_id = p.category_id
INNER JOIN department d on d.department_id = p.department_id
INNER JOIN order_line ol ON ol.product_id = p.product_id
WHERE ol.quantity_sold = 0;
```



Query 6: List the names of all cities from where orders are shipped. Also, for such cities, find the number of orders for which shipping was delayed. Sort the list of cities in order from the highest to the least number of shipping orders.

SELECT DISTINCT order_city, count(order_id) as 'Total Delayed Orders'
FROM orders
WHERE actual_shipping_days > scheduled_shipping_days
GROUP BY order_city
ORDER BY COUNT(order_id) DESC;

/* Chiragkumar Navadia, 04/03/2022 */

```
61
62  /* Query 6: List the names of all cities from where orders are shipped.
63     Also, for such cities, find the number of orders for which shipping was delayed.
64     Sort the list of cities in order from the highest to the least number of shipping orders.*/
65
66  SELECT DISTINCT order_city, count(order_id) as 'Total Delayed Orders'
67  FROM orders
68  WHERE actual_shipping_days > scheduled_shipping_days
69  GROUP BY order_city
70  ORDER BY COUNT(order_id) DESC;
71
72  /* Chiragkumar Navadia, 04/03/2022 */
73
```

100% 38:72

Result Grid | Filter Rows: Search | Export:

order_city	Total Delayed Orders
Bangalore	51
Mumbai	45
Pune	41
Delhi	37
Chennai	32
Surat	31
Visakhapatnam	30
Hyderabad	29
Gorakhpur	27
Ajmer	26
Agra	25
Kanpur	24
Jaipur	24
Aurangabad	21

Result Grid | Form Editor | Field Types | Query Stats

Result 1 | Read Only

Query 7: How many customers are there in each segment? Show the most popular segment at the top of the result. Incorporate a column alias in the result.

SELECT Segment, count(customer_id) as Total_Customers
FROM customer
GROUP BY Segment
ORDER BY Total_Customers DESC;

```
76  /* Query 7: How many customers are there in each segment?
77  Show the most popular segment at the top of the result. Incorporate a column alias in the result.*/
78  SELECT Segment, count(customer_id) as Total_Customers
79  FROM customer
80  GROUP BY Segment
81  ORDER BY Total_Customers DESC;
82  /* Chiragkumar Navadia, 04/04/2022 */
83
```

100% 17:80

Result Grid | Filter Rows: Search | Export:

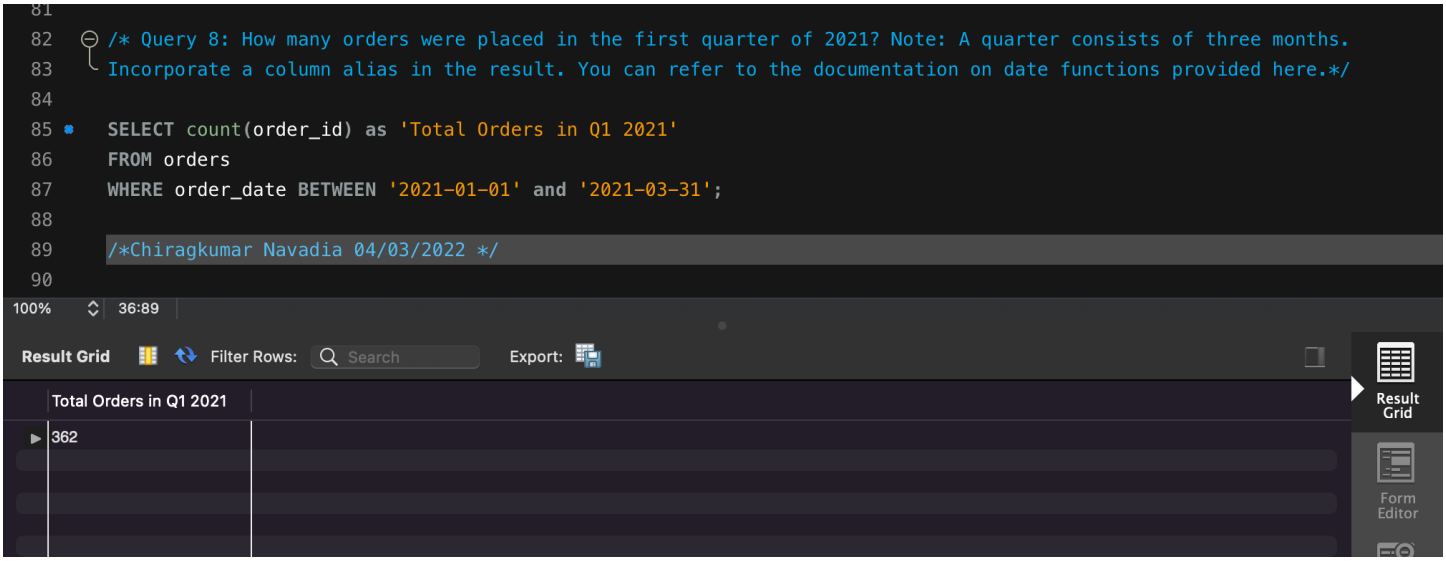
Segment	Total_Custome...
CONSUMER	2312
CORPORATE	1312
HOME_OFFICE	837

Result Grid | Form Editor | Field Types | Query Stats

Result 97 | Read Only

Query 8: How many orders were placed in the first quarter of 2021? Note: A quarter consists of three months. Incorporate a column alias in the result. You can refer to the documentation on date functions provided here.*/

SELECT count(order_id) as 'Total Orders in Q1 2021'
FROM orders
WHERE order_date BETWEEN '2021-01-01' and '2021-03-31';



Query 9: List in alphabetical order all states supporting multiple customer segments.

SELECT state, COUNT(DISTINCT segment) as Number_of_Segments
FROM customer
GROUP BY state
HAVING COUNT (DISTINCT segment) > 1
ORDER BY state;

```
93    /* Query 9: List in alphabetical order all states supporting multiple customer segments. */
94
95 •  SELECT state, COUNT(DISTINCT segment) as Number_of_Segments
96    FROM customer
97    GROUP BY state
98    HAVING COUNT(DISTINCT segment) > 1
99    ORDER BY state;
100
101   /*Chiragkumar Navadia 04/03/2022 */
102
103
```

100% 36:101

Result Grid Filter Rows: Search Export:

state	Number_of_Segme...
AR	2
AZ	3
CA	3
CO	3
CT	3
DC	3
DE	3
FL	3
GA	3
HI	3
IA	3
ID	2
IL	3
IN	3

Result 98 Read Only

Result Grid Form Editor Field Types Query Stats

Query 10: To help the commercial sales department with its marketing, find all customers in the corporate segment who have not placed any orders. Include each customers' first name, last name, street, city, state, and zip code. Sort the results by the last name first and then by the first name.

SELECT DISTINCT c.last_name, c.first_name, c.street, c.city, c.state, c.zipcode, c.segment
FROM customer c
INNER JOIN orders o ON o.customer_id = c.customer_id
WHERE c.segment = 'Corporate' AND o.order_id = null
ORDER BY last_name, first_name;

```
/*Query 10: To help the commercial sales department with its marketing,
find all customers in the corporate segment who have not placed any orders.
Include each customers' first name, last name, street, city, state, and zip code.
Sort the results by the last name first and then by the first name.*/

SELECT DISTINCT c.last_name, c.first_name, c.street, c.city, c.state, c.zipcode, c.segment
FROM customer c
INNER JOIN orders o ON o.customer_id = c.customer_id
WHERE c.segment = 'Corporate' and o.order_id = null
ORDER BY last_name, first_name;
/*Chiragkumar Navadia 04/04/2022*/
```

100% 35:114

Result Grid Filter Rows: Search Export:

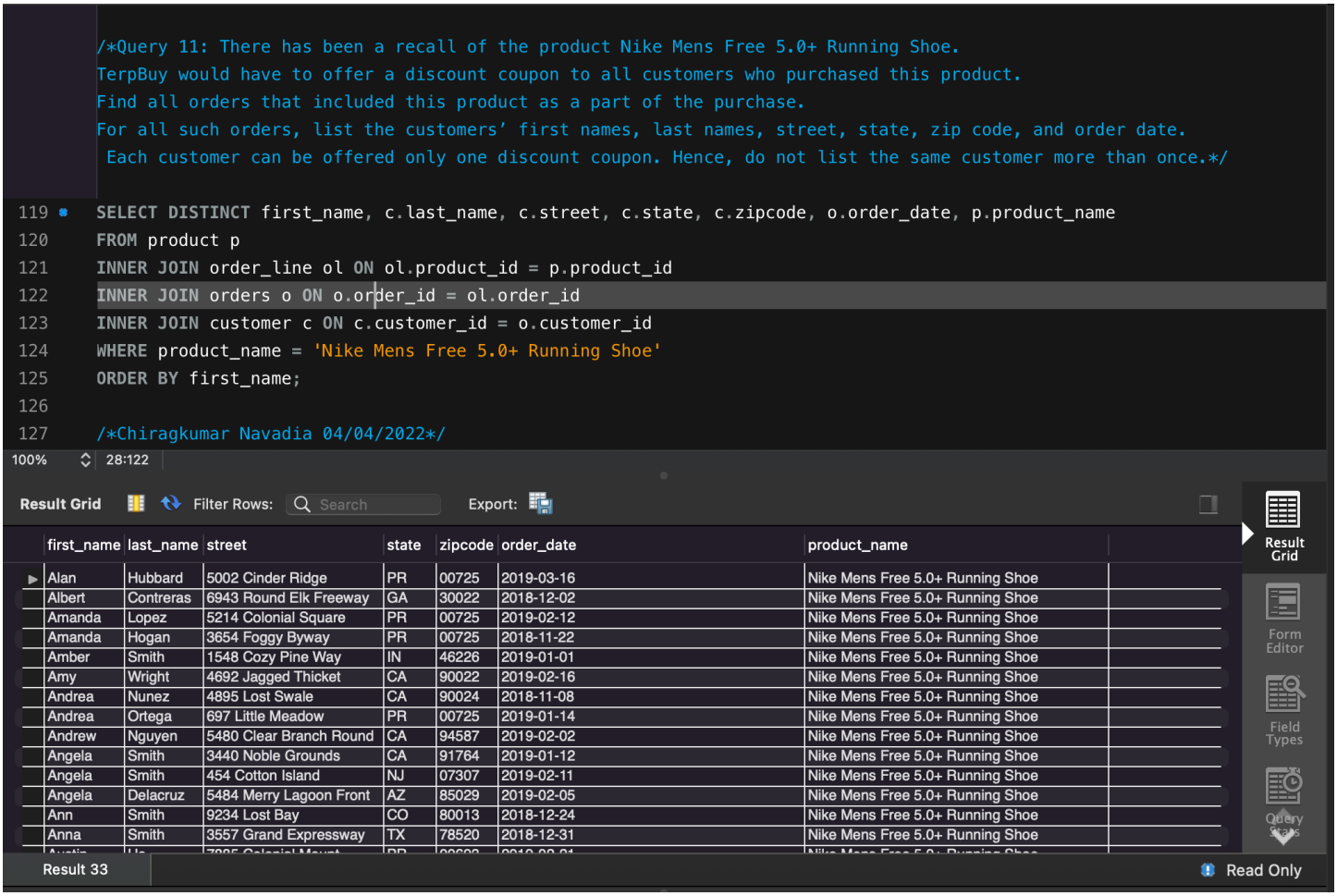
last_name	first_name	street	city	state	zipcode	segment

Result Grid Form Editor Field Types Query Stats

Result 99 Read Only

Query 11: There has been a recall of the product Nike Mens Free 5.0+ Running Shoe. TerpBuy would have to offer a discount coupon to all customers who purchased this product. Find all orders that included this product as a part of the purchase. For all such orders, list the customers' first names, last names, street, state, zip code, and order date. Each customer can be offered only one discount coupon. Hence, do not list the same customer more than once.

SELECT DISTINCT first_name, c.last_name, c.street, c.state, c.zipcode, o.order_date, p.product_name
FROM product p
INNER JOIN order_line ol ON ol.product_id = p.product_id
INNER JOIN orders o ON o.order_id = ol.order_id
INNER JOIN customer c ON c.customer_id = o.customer_id
WHERE product_name = 'Nike Mens Free 5.0+ Running Shoe'
ORDER BY first_name;



```
/*Query 11: There has been a recall of the product Nike Mens Free 5.0+ Running Shoe.
TerpBuy would have to offer a discount coupon to all customers who purchased this product.
Find all orders that included this product as a part of the purchase.
For all such orders, list the customers' first names, last names, street, state, zip code, and order date.
 Each customer can be offered only one discount coupon. Hence, do not list the same customer more than once.*/

SELECT DISTINCT first_name, c.last_name, c.street, c.state, c.zipcode, o.order_date, p.product_name
FROM product p
INNER JOIN order_line ol ON ol.product_id = p.product_id
INNER JOIN orders o ON o.order_id = ol.order_id
INNER JOIN customer c ON c.customer_id = o.customer_id
WHERE product_name = 'Nike Mens Free 5.0+ Running Shoe'
ORDER BY first_name;

/*Chiragkumar Navadia 04/04/2022*/
```

100% 28:122

Result Grid | Filter Rows: Search | Export:

first_name	last_name	street	state	zipcode	order_date	product_name
Alan	Hubbard	5002 Cinder Ridge	PR	00725	2019-03-16	Nike Mens Free 5.0+ Running Shoe
Albert	Contreras	6943 Round Elk Freeway	GA	30022	2018-12-02	Nike Mens Free 5.0+ Running Shoe
Amanda	Lopez	5214 Colonial Square	PR	00725	2019-02-12	Nike Mens Free 5.0+ Running Shoe
Amanda	Hogan	3654 Foggy Byway	PR	00725	2018-11-22	Nike Mens Free 5.0+ Running Shoe
Amber	Smith	1548 Cozy Pine Way	IN	46226	2019-01-01	Nike Mens Free 5.0+ Running Shoe
Amy	Wright	4692 Jagged Thicket	CA	90022	2019-02-16	Nike Mens Free 5.0+ Running Shoe
Andrea	Nunez	4895 Lost Swale	CA	90024	2018-11-08	Nike Mens Free 5.0+ Running Shoe
Andrea	Ortega	697 Little Meadow	PR	00725	2019-01-14	Nike Mens Free 5.0+ Running Shoe
Andrew	Nguyen	5480 Clear Branch Round	CA	94587	2019-02-02	Nike Mens Free 5.0+ Running Shoe
Angela	Smith	3440 Noble Grounds	CA	91764	2019-01-12	Nike Mens Free 5.0+ Running Shoe
Angela	Smith	454 Cotton Island	NJ	07307	2019-02-11	Nike Mens Free 5.0+ Running Shoe
Angela	Delacruz	5484 Merry Lagoon Front	AZ	85029	2019-02-05	Nike Mens Free 5.0+ Running Shoe
Ann	Smith	9234 Lost Bay	CO	80013	2018-12-24	Nike Mens Free 5.0+ Running Shoe
Anna	Smith	3557 Grand Expressway	TX	78520	2018-12-31	Nike Mens Free 5.0+ Running Shoe

Result Grid | Form Editor | Field Types | Query Stats

Result 33 | Read Only

Query 12: Premium customers are those customers who have placed orders with order amounts greater than the average order amount.
For each customer, find the first and last names, and the order amount for all orders that exceeded the average order amount.

SELECT c.first_name, c.last_name, sum(ol.total_price) as Order_Amount
FROM customer c
INNER JOIN orders o ON o.customer_id = c.customer_id
INNER JOIN order_line ol ON ol.order_id = o.order_id
GROUP BY c.first_name, c.last_name
HAVING Order_Amount > (select AVG(total_price) as avg from order_line)
ORDER BY Order_Amount;

```
/*Query 12: Premium customers are those customers who have placed orders with order amounts greater than the average
order amount.
For each customer, find the first and last names, and the order amount for all orders that exceeded the average order
amount.*/

SELECT c.first_name, c.last_name, sum(ol.total_price) as Order_Amount
FROM customer c
INNER JOIN orders o ON o.customer_id = c.customer_id
INNER JOIN order_line ol ON ol.order_id = o.order_id
GROUP BY c.first_name, c.last_name
HAVING Order_Amount > (select AVG(total_price) as avg from order_line)
ORDER BY Order_Amount;
/*Chiragkumar Navadia 04/04/2022*/
```

100% 36:144

Result Grid | Filter Rows: Search | Export: | Fetch rows:

first_name	last_name	Order_Amount
Jeffrey	Whitney	207.96
Stephanie	Morris	209.93
Hannah	Jennings	210.85
Nichole	Holloway	210.85
Erica	Robinson	210.85
Constance	Austin	210.85
Kimberly	Lopez	210.85
Orli	Atkinson	210.85
Tana	Austin	210.85
Gillian	Baker	210.85
Ursula	Andrade	210.85
Quin	Mcleod	210.85
Naida	Sharpe	210.85
Simone	Ayers	210.85

Result Grid | Form Editor | Field Types | Query Stats

Result 100 | Read Only