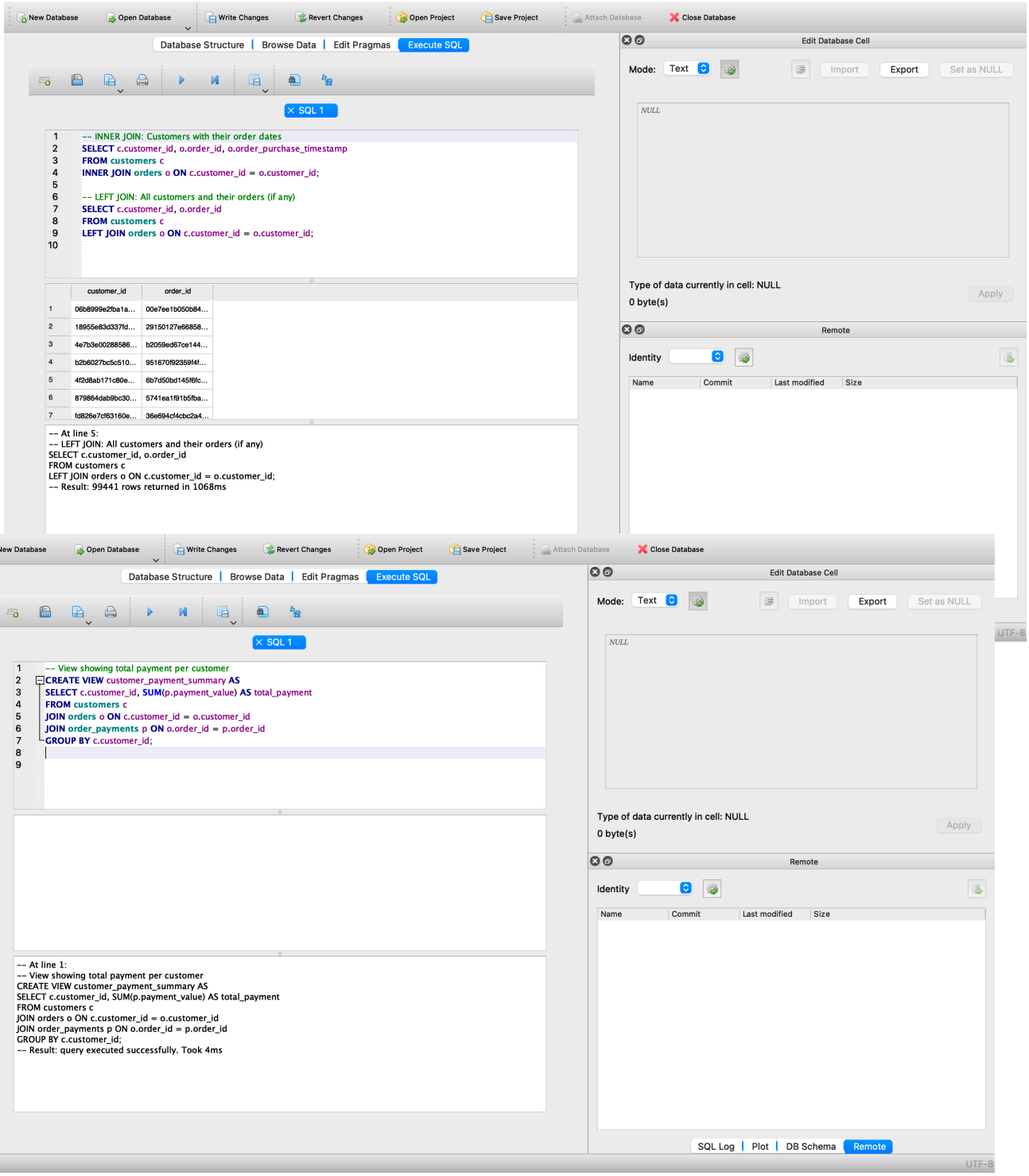
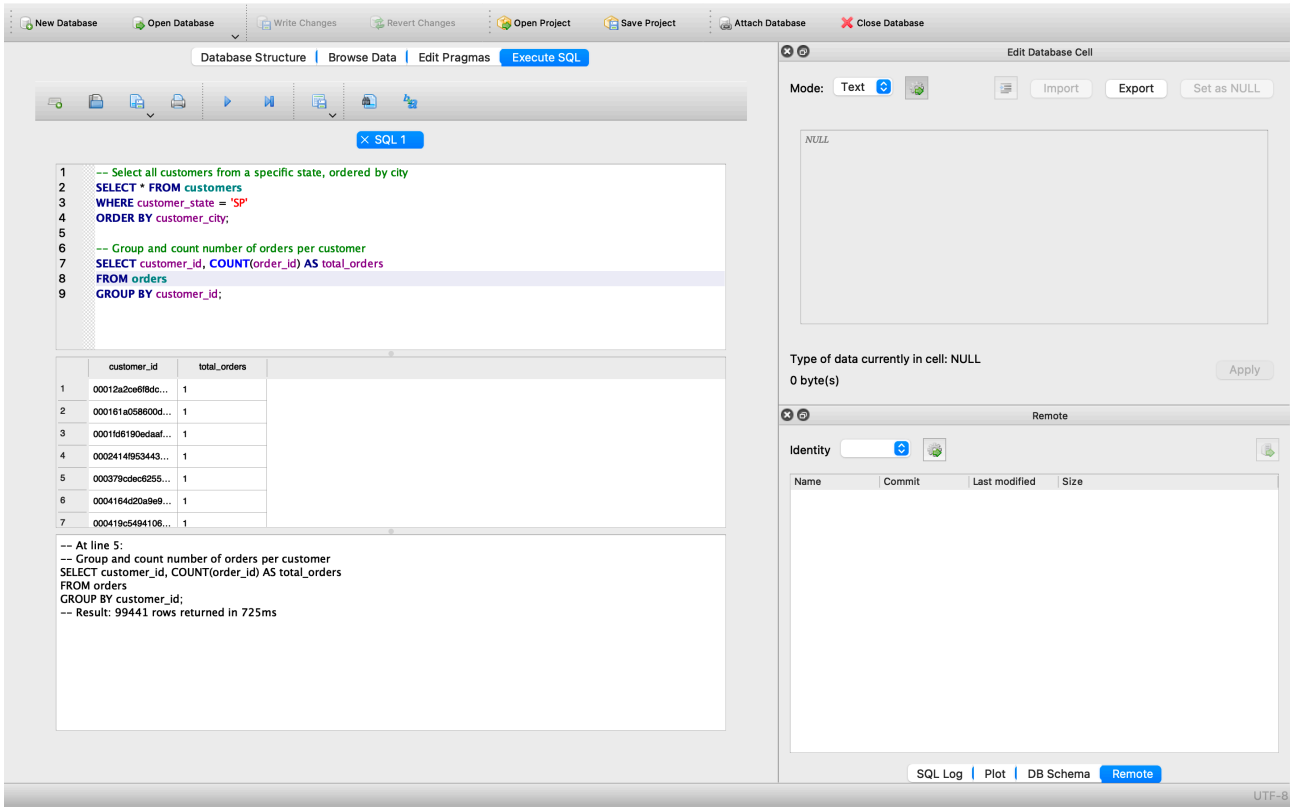
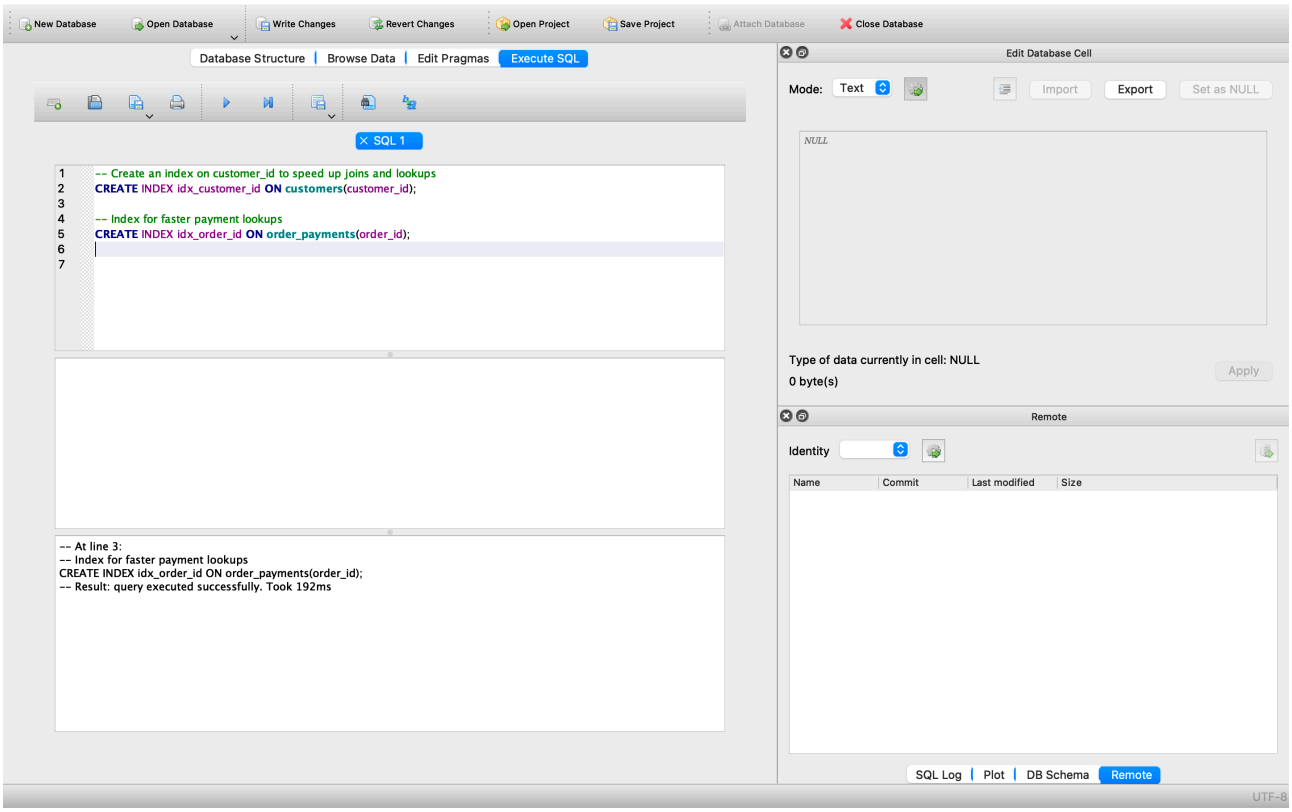
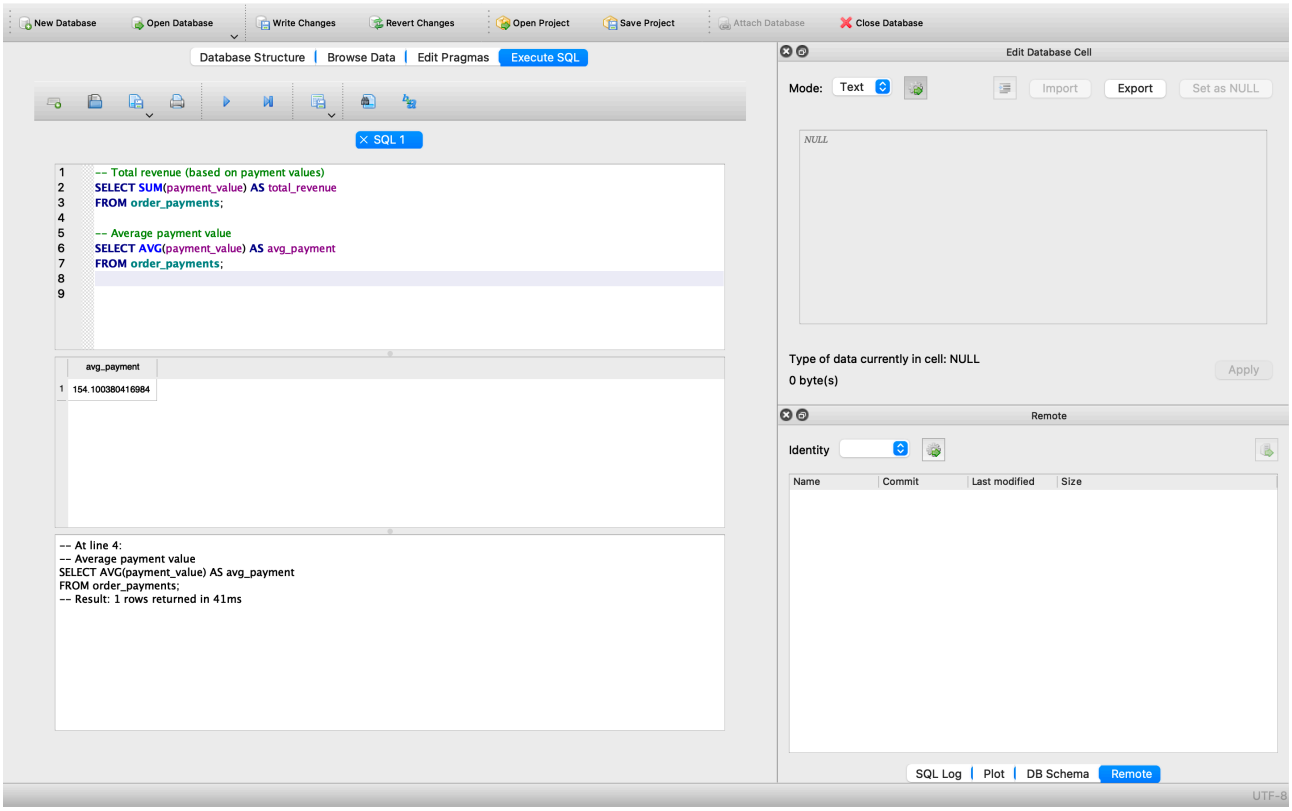
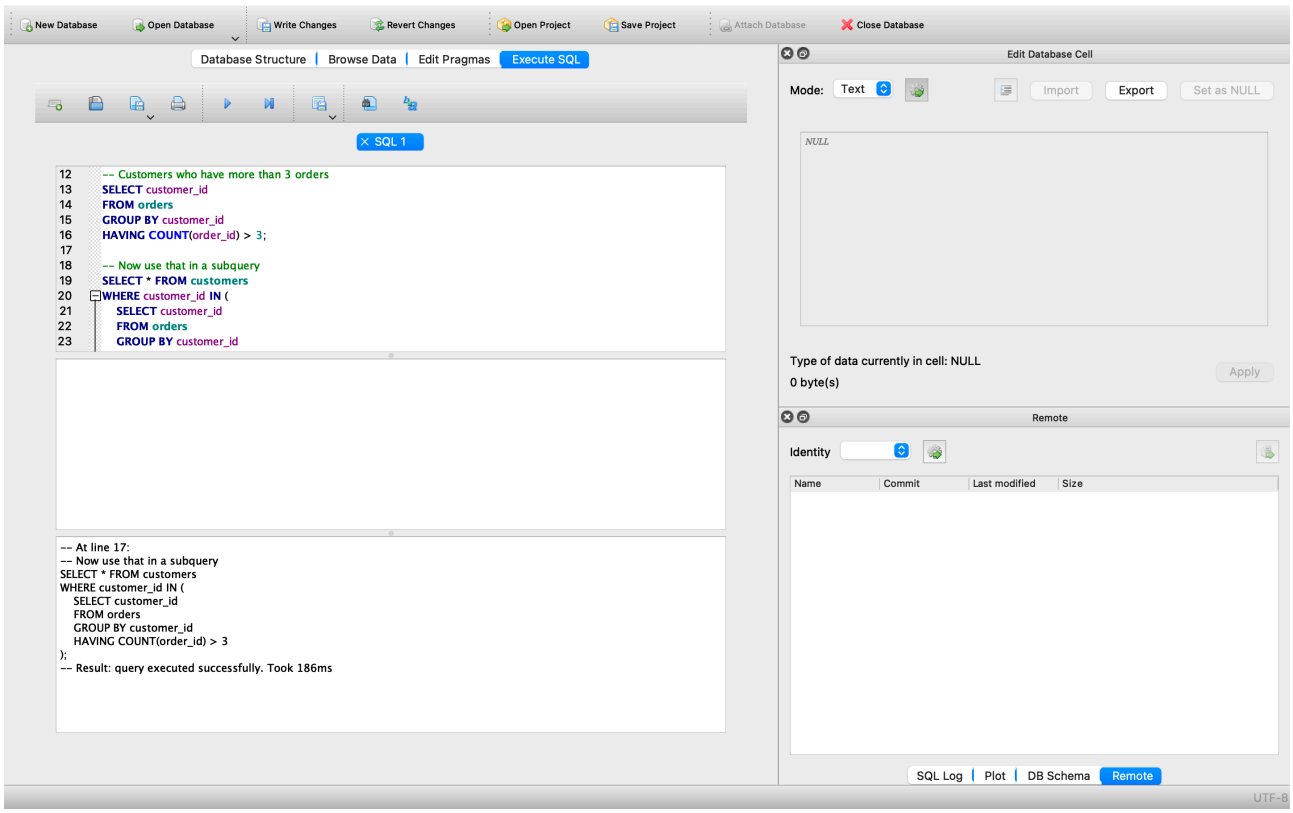

New Database
Open Database
Write Changes
Revert Changes
Open Project
Save Project
Attach Database
Close Database
Database Structure
Browse Data
Edit Pragmas
Execute SQL
SQL 1
1 -- INNER JOIN: Customers with their order dates
2 SELECT c.customer_id, o.order_id, o.order_purchase_timestamp
3 FROM customers c
4 INNER JOIN orders o ON c.customer_id = o.customer_id;
5
6 -- LEFT JOIN: All customers and their orders (if any)
7 SELECT c.customer_id, o.order_id
8 FROM customers c
9 LEFT JOIN orders o ON c.customer_id = o.customer_id;
10
customer_id
order_id
1 06b8999e2fba1a... 00e7ee1b050b84...
2 18955e83d337fd... 29150127e66858...
3 4e7b3e00288586... b2059ed67ce144...
4 b2b6027bc5c510... 951670f92359f4f...
5 4f2d8ab171c80e... 6b7d50bd145f6fc...
6 879864dab9bc30... 5741ea1f91b5fba...
7 fd826e7cf63160e... 36e694cf4cbc2a4...
-- At line 5:
-- LEFT JOIN: All customers and their orders (if any)
SELECT c.customer_id, o.order_id
FROM customers c
LEFT JOIN orders o ON c.customer_id = o.customer_id;
-- Result: 99441 rows returned in 1068ms
Edit Database Cell
Mode: Text
Import
Export
Set as NULL
NULL
Type of data currently in cell: NULL
0 byte(s)
Apply
Remote
Identity
Name
Commit
Last modified
Size
UTF-8

New Database
Open Database
Write Changes
Revert Changes
Open Project
Save Project
Attach Database
Close Database
Database Structure
Browse Data
Edit Pragmas
Execute SQL
SQL 1
1 -- View showing total payment per customer
2 CREATE VIEW customer_payment_summary AS
3 SELECT c.customer_id, SUM(p.payment_value) AS total_payment
4 FROM customers c
5 JOIN orders o ON c.customer_id = o.customer_id
6 JOIN order_payments p ON o.order_id = p.order_id
7 GROUP BY c.customer_id;
8
9
-- At line 1:
-- View showing total payment per customer
CREATE VIEW customer_payment_summary AS
SELECT c.customer_id, SUM(p.payment_value) AS total_payment
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
JOIN order_payments p ON o.order_id = p.order_id
GROUP BY c.customer_id;
-- Result: query executed successfully. Took 4ms
Edit Database Cell
Mode: Text
Import
Export
Set as NULL
NULL
Type of data currently in cell: NULL
0 byte(s)
Apply
Remote
Identity
Name
Commit
Last modified
Size
SQL Log
Plot
DB Schema
Remote
UTF-8


New Database
Open Database
Write Changes
Revert Changes
Open Project
Save Project
Attach Database
Close Database
Database Structure
Browse Data
Edit Pragmas
Execute SQL
SQL 1
-- Select all customers from a specific state, ordered by city
SELECT * FROM customers
WHERE customer_state = 'SP'
ORDER BY customer_city;
-- Group and count number of orders per customer
SELECT customer_id, COUNT(order_id) AS total_orders
FROM orders
GROUP BY customer_id;
customer_id
total_orders
1 00012a2ce6f8dc... 1
2 000161a058600d... 1
3 0001fd6190edaaf... 1
4 0002414f953443... 1
5 000379cdec6255... 1
6 0004164d20a9e9... 1
7 000419c5494106... 1
-- At line 5:
-- Group and count number of orders per customer
SELECT customer_id, COUNT(order_id) AS total_orders
FROM orders
GROUP BY customer_id;
-- Result: 99441 rows returned in 725ms
Edit Database Cell
Mode: Text
Import
Export
Set as NULL
NULL
Type of data currently in cell: NULL
0 byte(s)
Apply
Remote
Identity
Name
Commit
Last modified
Size
SQL Log
Plot
DB Schema
Remote
UTF-8


New Database
Open Database
Write Changes
Revert Changes
Open Project
Save Project
Attach Database
Close Database
Database Structure
Browse Data
Edit Pragmas
Execute SQL
SQL 1
-- Create an index on customer_id to speed up joins and lookups
CREATE INDEX idx_customer_id ON customers(customer_id);
-- Index for faster payment lookups
CREATE INDEX idx_order_id ON order_payments(order_id);
-- At line 3:
-- Index for faster payment lookups
CREATE INDEX idx_order_id ON order_payments(order_id);
-- Result: query executed successfully. Took 192ms
Edit Database Cell
Mode: Text
Import
Export
Set as NULL
NULL
Type of data currently in cell: NULL
0 byte(s)
Apply
Remote
Identity
Name
Commit
Last modified
Size
SQL Log
Plot
DB Schema
Remote
UTF-8


New Database
Open Database
Write Changes
Revert Changes
Open Project
Save Project
Attach Database
Close Database
Database Structure
Browse Data
Edit Pragmas
Execute SQL
SQL 1
1 -- Total revenue (based on payment values)
2 SELECT SUM(payment_value) AS total_revenue
3 FROM order_payments;
4
5 -- Average payment value
6 SELECT AVG(payment_value) AS avg_payment
7 FROM order_payments;
8
9
avg_payment
1 154.100380416984
-- At line 4:
-- Average payment value
SELECT AVG(payment_value) AS avg_payment
FROM order_payments;
-- Result: 1 rows returned in 41ms
Edit Database Cell
Mode: Text
Import
Export
Set as NULL
NULL
Type of data currently in cell: NULL
0 byte(s)
Apply
Remote
Identity
Name
Commit
Last modified
Size
SQL Log
Plot
DB Schema
Remote
UTF-8


New Database
Open Database
Write Changes
Revert Changes
Open Project
Save Project
Attach Database
Close Database
Database Structure
Browse Data
Edit Pragmas
Execute SQL
SQL 1
12 -- Customers who have more than 3 orders
13 SELECT customer_id
14 FROM orders
15 GROUP BY customer_id
16 HAVING COUNT(order_id) > 3;
17
18 -- Now use that in a subquery
19 SELECT * FROM customers
20 WHERE customer_id IN (
21 SELECT customer_id
22 FROM orders
23 GROUP BY customer_id
-- At line 17:
-- Now use that in a subquery
SELECT * FROM customers
WHERE customer_id IN (
SELECT customer_id
FROM orders
GROUP BY customer_id
HAVING COUNT(order_id) > 3
);
-- Result: query executed successfully. Took 186ms
Edit Database Cell
Mode: Text
Import
Export
Set as NULL
NULL
Type of data currently in cell: NULL
0 byte(s)
Apply
Remote
Identity
Name
Commit
Last modified
Size
SQL Log
Plot
DB Schema
Remote
UTF-8