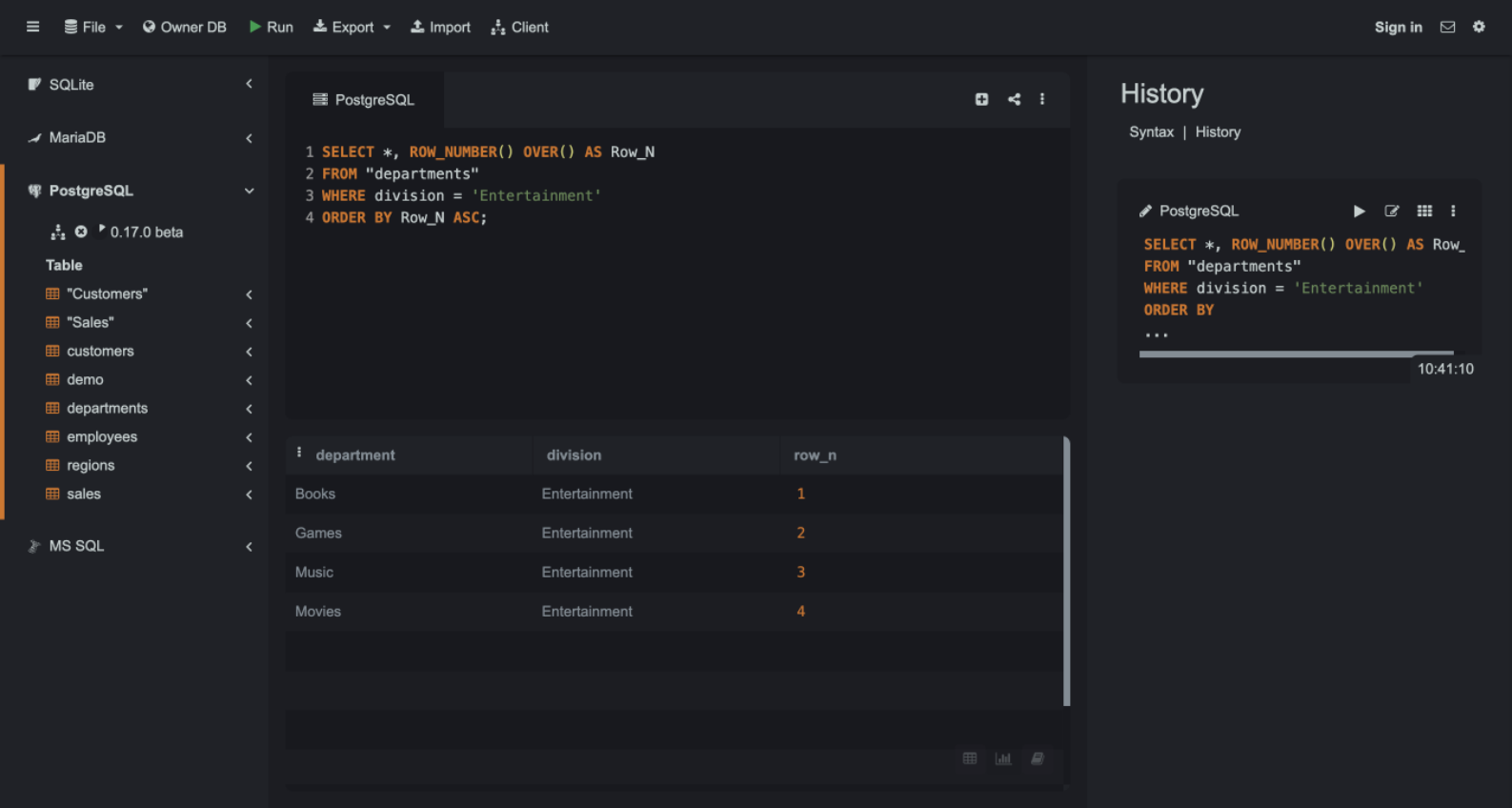
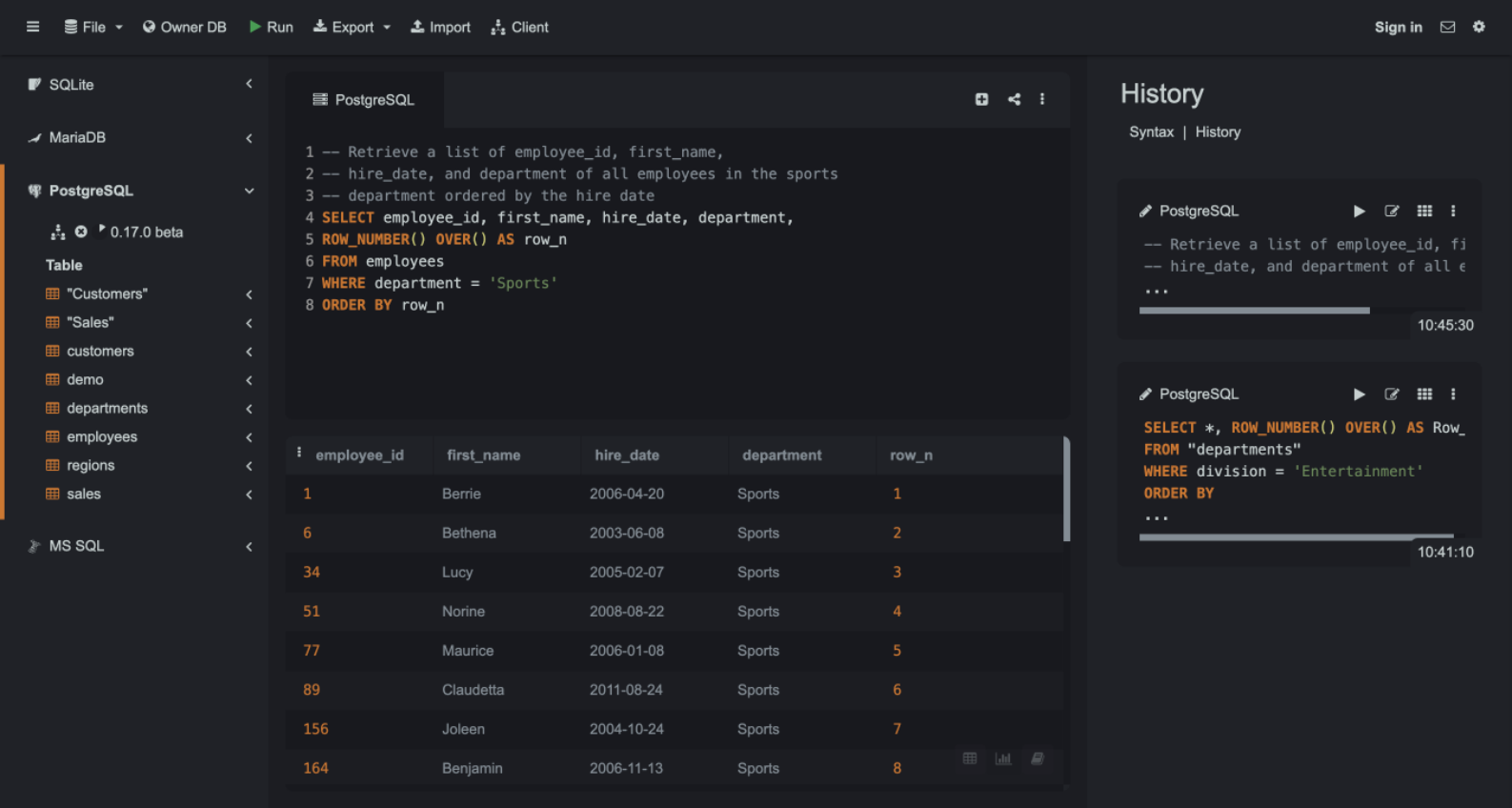
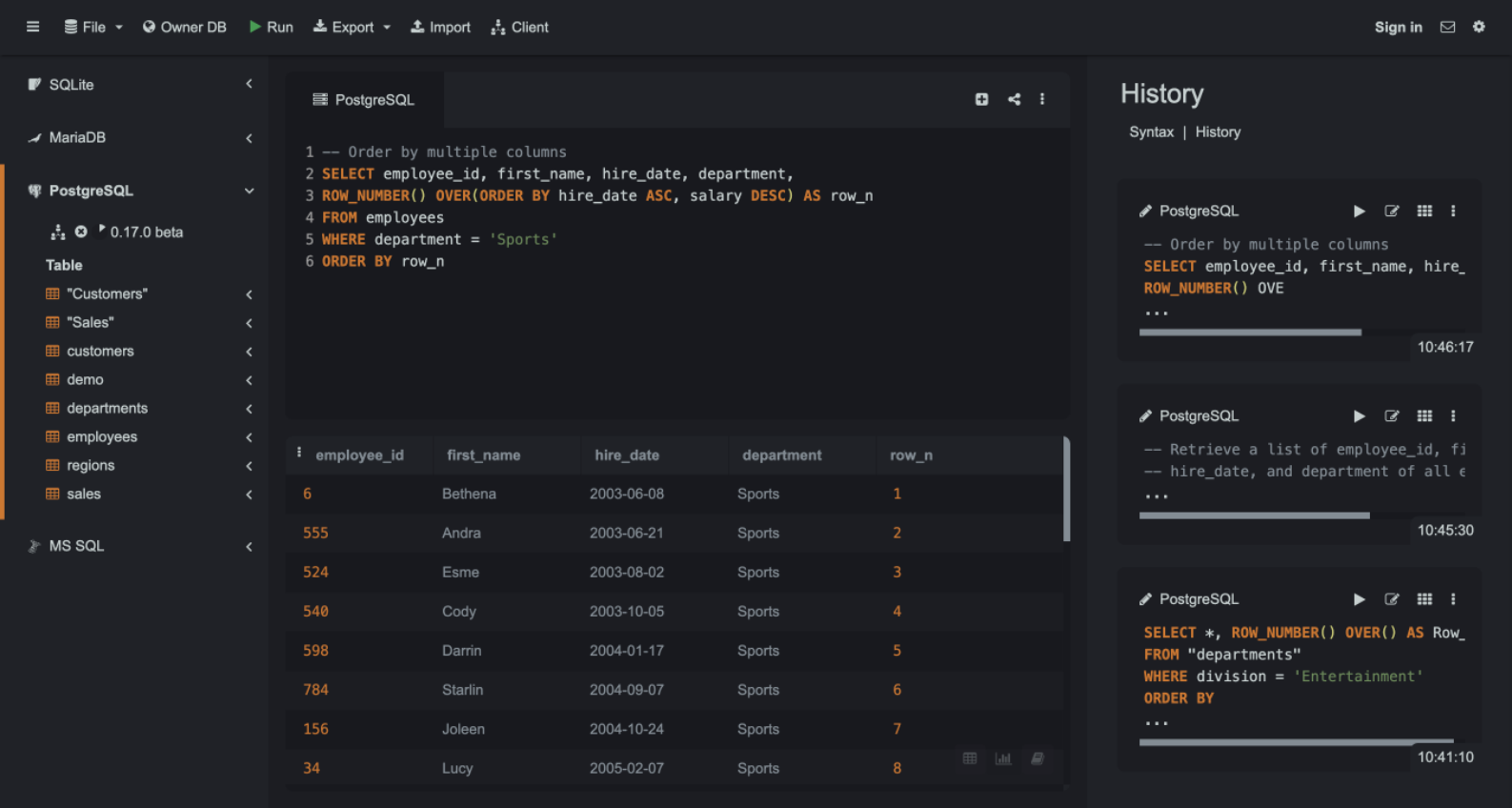
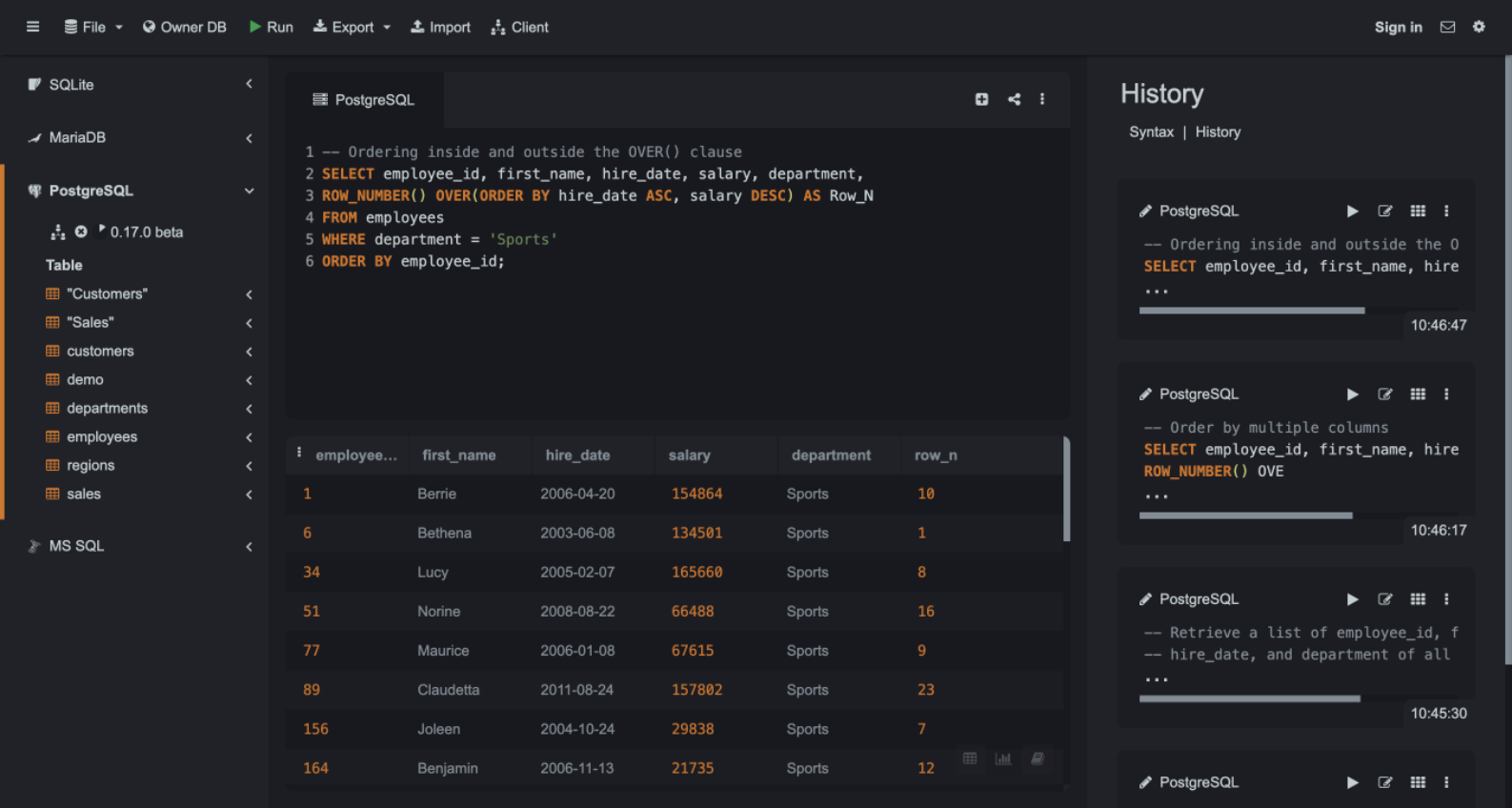
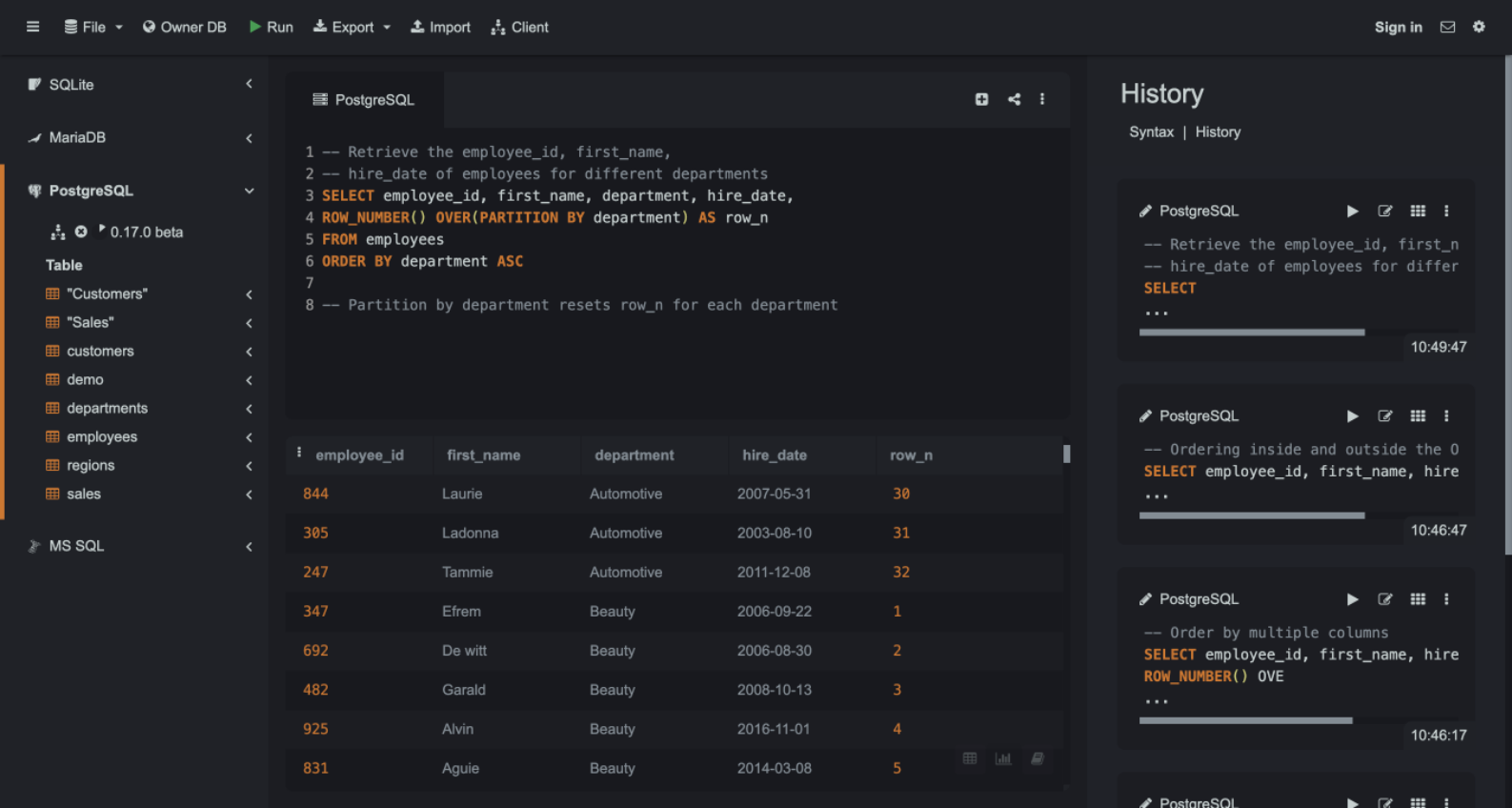
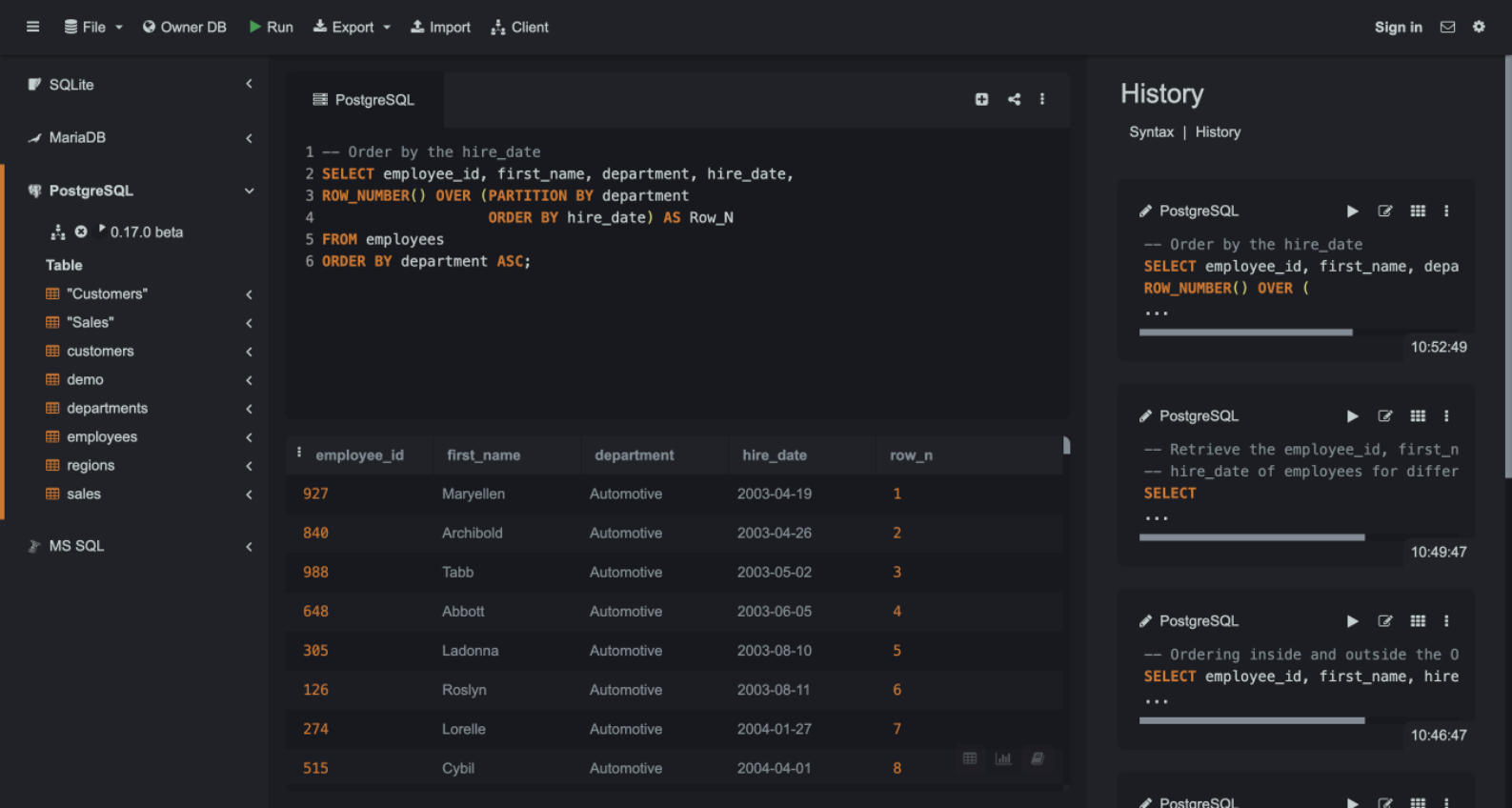

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
regions
sales
MS SQL
PostgreSQL
1 SELECT *, ROW_NUMBER() OVER() AS Row_N
2 FROM "departments"
3 WHERE division = 'Entertainment'
4 ORDER BY Row_N ASC;
department
division
row_n
Books
Entertainment
1
Games
Entertainment
2
Music
Entertainment
3
Movies
Entertainment
4
History
Syntax | History
PostgreSQL
SELECT *, ROW_NUMBER() OVER() AS Row_
FROM "departments"
WHERE division = 'Entertainment'
ORDER BY
...
10:41:10

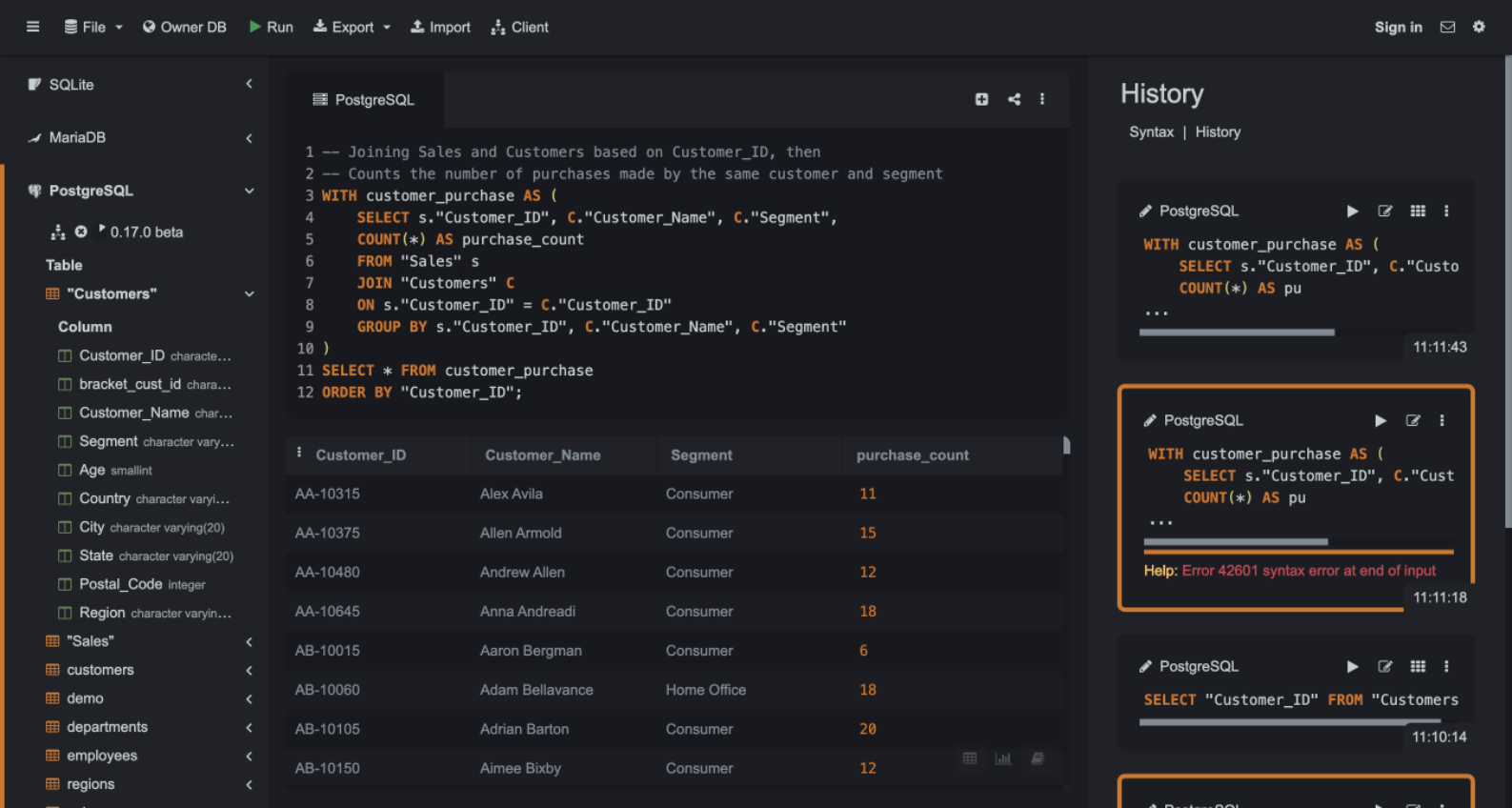

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
regions
sales
MS SQL
PostgreSQL
1 -- Retrieve a list of employee_id, first_name,
2 -- hire_date, and department of all employees in the sports
3 -- department ordered by the hire date
4 SELECT employee_id, first_name, hire_date, department,
5 ROW_NUMBER() OVER() AS row_n
6 FROM employees
7 WHERE department = 'Sports'
8 ORDER BY row_n
employee_id
first_name
hire_date
department
row_n
1 Berrie 2006-04-20 Sports 1
6 Bethena 2003-06-08 Sports 2
34 Lucy 2005-02-07 Sports 3
51 Norine 2008-08-22 Sports 4
77 Maurice 2006-01-08 Sports 5
89 Claudetta 2011-08-24 Sports 6
156 Joleen 2004-10-24 Sports 7
164 Benjamin 2006-11-13 Sports 8
History
Syntax | History
PostgreSQL
-- Retrieve a list of employee_id, fi
-- hire_date, and department of all e
...
10:45:30
PostgreSQL
SELECT *, ROW_NUMBER() OVER() AS Row_
FROM "departments"
WHERE division = 'Entertainment'
ORDER BY
...
10:41:10

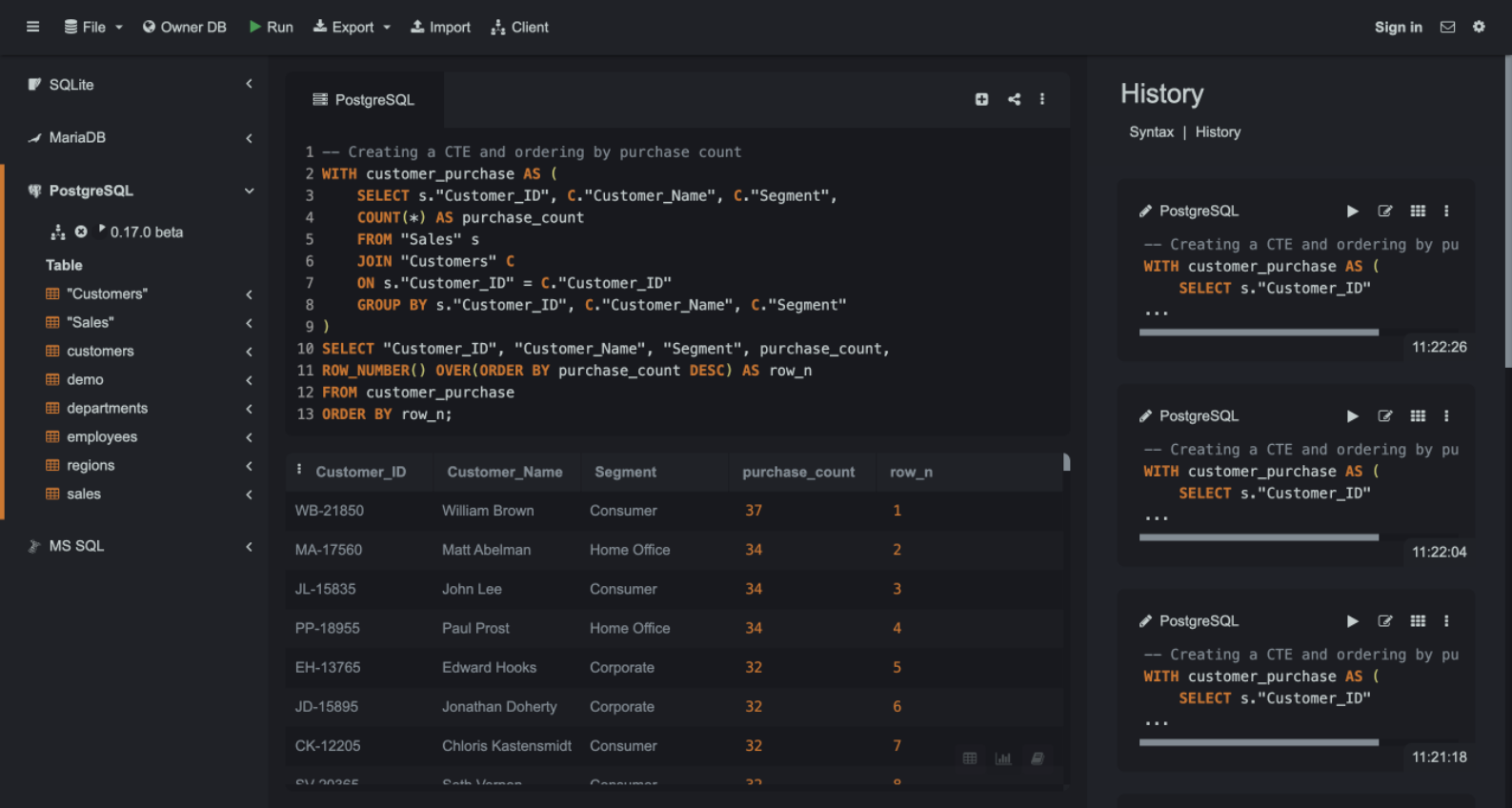

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
regions
sales
MS SQL
PostgreSQL
1 -- Order by multiple columns
2 SELECT employee_id, first_name, hire_date, department,
3 ROW_NUMBER() OVER(ORDER BY hire_date ASC, salary DESC) AS row_n
4 FROM employees
5 WHERE department = 'Sports'
6 ORDER BY row_n
employee_id
first_name
hire_date
department
row_n
6 Bethena 2003-06-08 Sports 1
555 Andra 2003-06-21 Sports 2
524 Esme 2003-08-02 Sports 3
540 Cody 2003-10-05 Sports 4
598 Darrin 2004-01-17 Sports 5
784 Starlin 2004-09-07 Sports 6
156 Joleen 2004-10-24 Sports 7
34 Lucy 2005-02-07 Sports 8
History
Syntax | History
PostgreSQL
-- Order by multiple columns
SELECT employee_id, first_name, hire_
ROW_NUMBER() OVE
...
10:46:17
PostgreSQL
-- Retrieve a list of employee_id, fi
-- hire_date, and department of all e
...
10:45:30
PostgreSQL
SELECT *, ROW_NUMBER() OVER() AS Row_
FROM "departments"
WHERE division = 'Entertainment'
ORDER BY
...
10:41:10

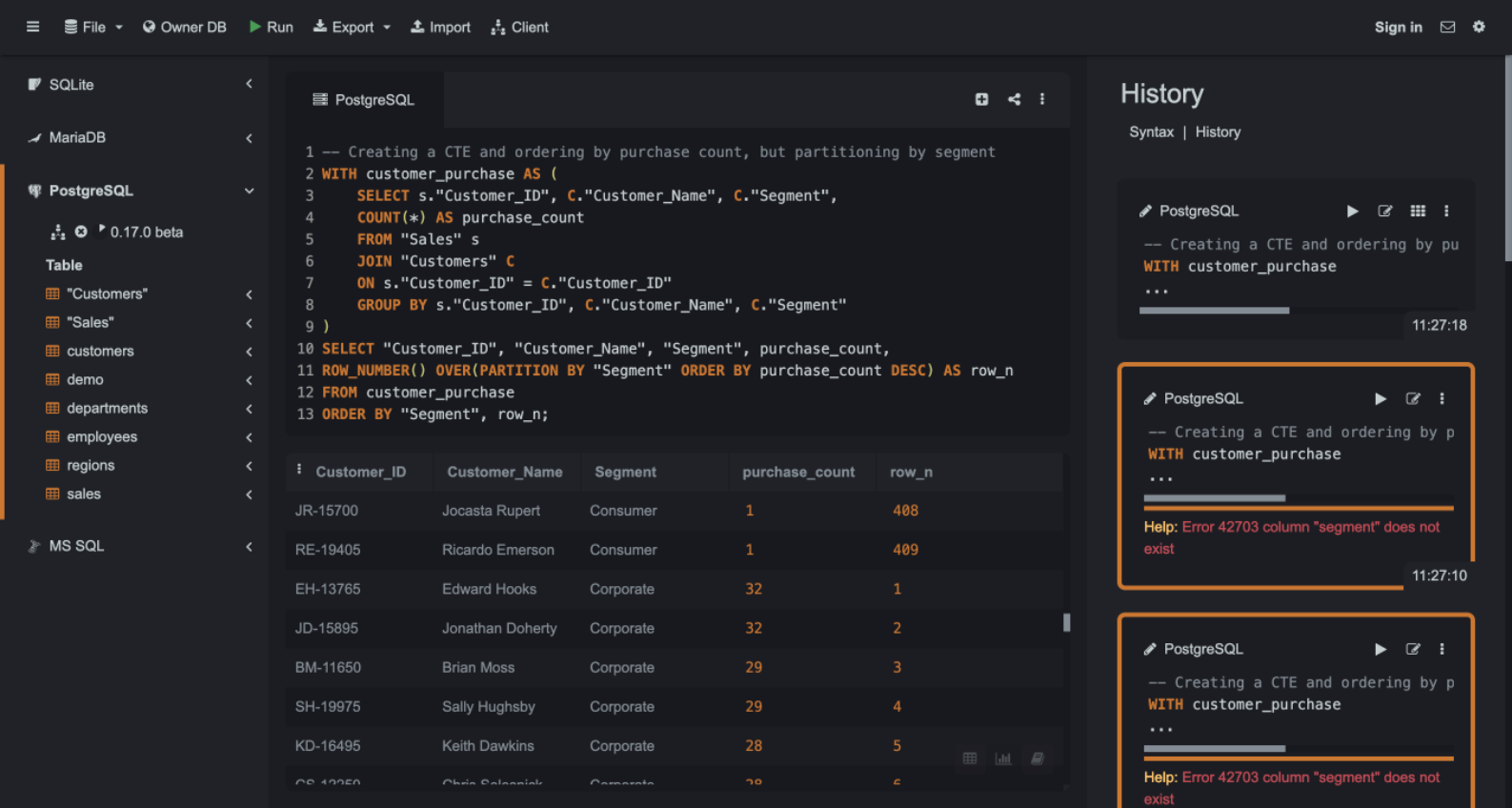

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
regions
sales
MS SQL
PostgreSQL
1 -- Ordering inside and outside the OVER() clause
2 SELECT employee_id, first_name, hire_date, salary, department,
3 ROW_NUMBER() OVER(ORDER BY hire_date ASC, salary DESC) AS Row_N
4 FROM employees
5 WHERE department = 'Sports'
6 ORDER BY employee_id;
employee...
first_name
hire_date
salary
department
row_n
1 Berrie 2006-04-20 154864 Sports 10
6 Bethena 2003-06-08 134501 Sports 1
34 Lucy 2005-02-07 165660 Sports 8
51 Norine 2008-08-22 66488 Sports 16
77 Maurice 2006-01-08 67615 Sports 9
89 Claudetta 2011-08-24 157802 Sports 23
156 Joleen 2004-10-24 29838 Sports 7
164 Benjamin 2006-11-13 21735 Sports 12
History
Syntax | History
PostgreSQL
-- Ordering inside and outside the O
SELECT employee_id, first_name, hire
...
10:46:47
PostgreSQL
-- Order by multiple columns
SELECT employee_id, first_name, hire
ROW_NUMBER() OVE
...
10:46:17
PostgreSQL
-- Retrieve a list of employee_id, f
-- hire_date, and department of all
...
10:45:30
PostgreSQL

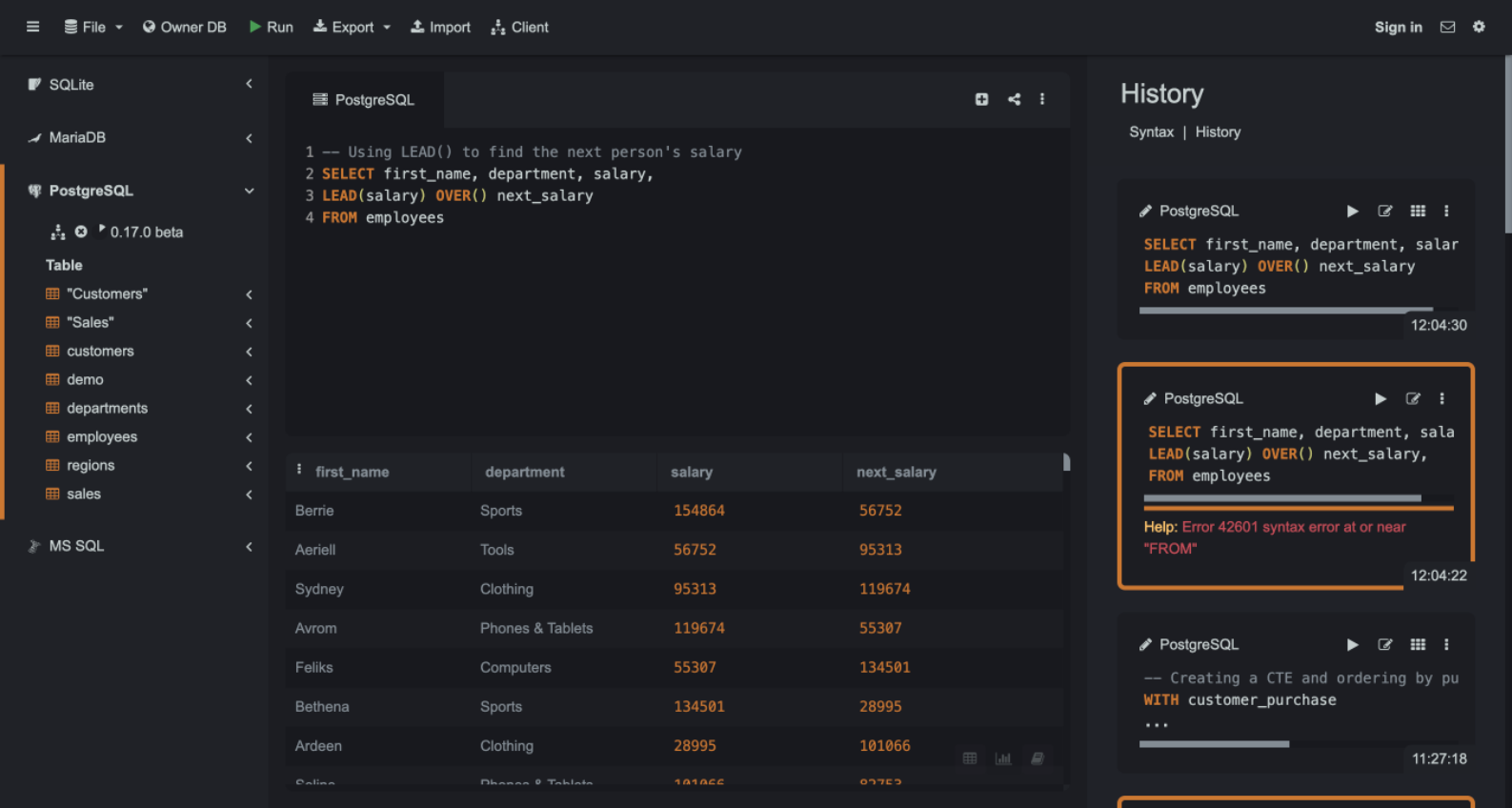

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
regions
sales
MS SQL
PostgreSQL
1 -- Retrieve the employee_id, first_name,
2 -- hire_date of employees for different departments
3 SELECT employee_id, first_name, department, hire_date,
4 ROW_NUMBER() OVER(PARTITION BY department) AS row_n
5 FROM employees
6 ORDER BY department ASC
7
8 -- Partition by department resets row_n for each department
employee_id
first_name
department
hire_date
row_n
844 Laurie Automotive 2007-05-31 30
305 Ladonna Automotive 2003-08-10 31
247 Tammie Automotive 2011-12-08 32
347 Efrem Beauty 2006-09-22 1
692 De witt Beauty 2006-08-30 2
482 Garald Beauty 2008-10-13 3
925 Alvin Beauty 2016-11-01 4
831 Aguie Beauty 2014-03-08 5
History
Syntax | History
PostgreSQL
-- Retrieve the employee_id, first_n
-- hire_date of employees for differ
SELECT
...
10:49:47
PostgreSQL
-- Ordering inside and outside the O
SELECT employee_id, first_name, hire
...
10:46:47
PostgreSQL
-- Order by multiple columns
SELECT employee_id, first_name, hire
ROW_NUMBER() OVE
...
10:46:17

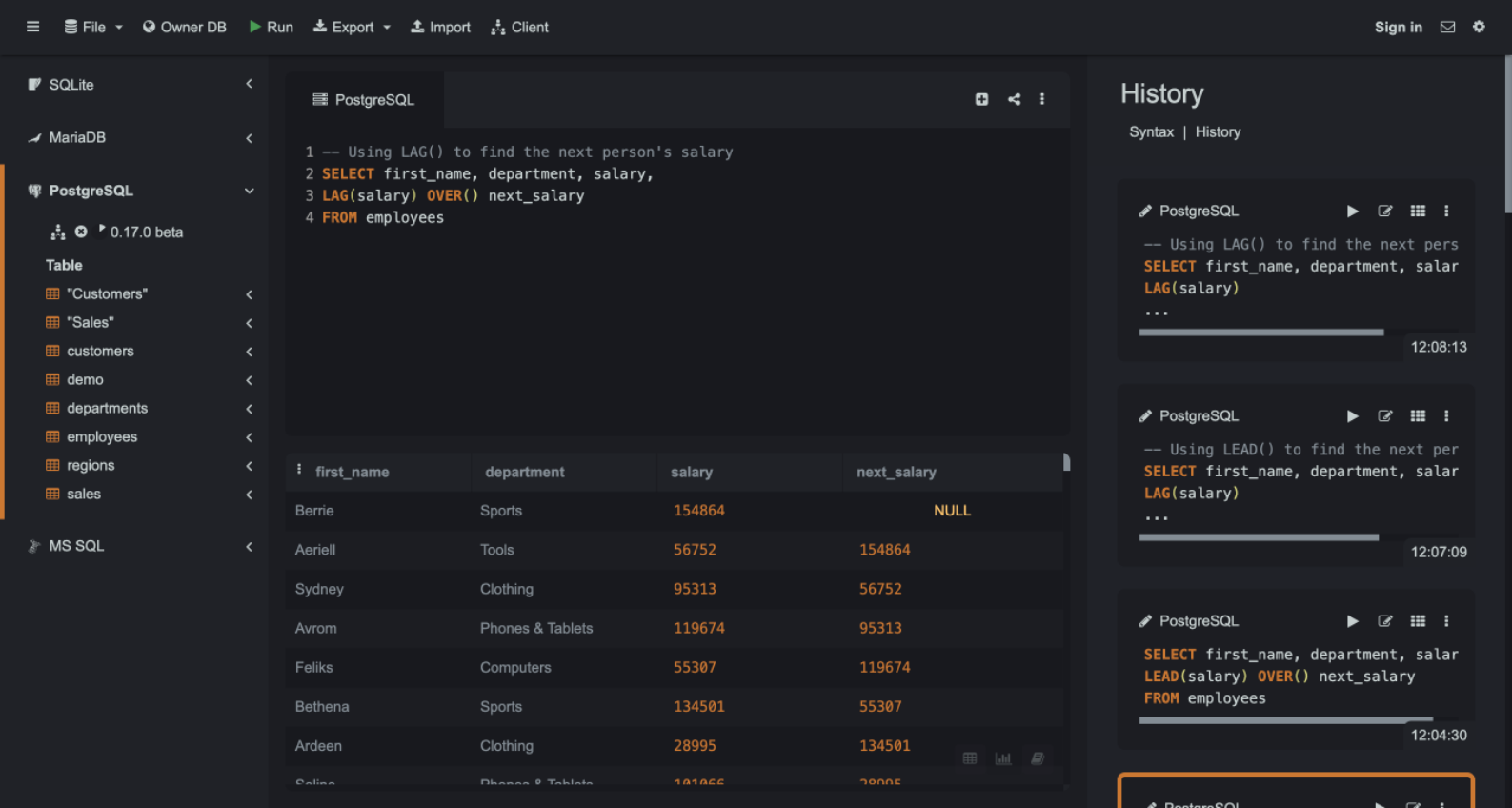

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
regions
sales
MS SQL
PostgreSQL
1 -- Order by the hire_date
2 SELECT employee_id, first_name, department, hire_date,
3 ROW_NUMBER() OVER (PARTITION BY department
4 ORDER BY hire_date) AS Row_N
5 FROM employees
6 ORDER BY department ASC;
employee_id
first_name
department
hire_date
row_n
927 Maryellen Automotive 2003-04-19 1
840 Archibold Automotive 2003-04-26 2
988 Tabb Automotive 2003-05-02 3
648 Abbott Automotive 2003-06-05 4
305 Ladonna Automotive 2003-08-10 5
126 Roslyn Automotive 2003-08-11 6
274 Lorelle Automotive 2004-01-27 7
515 Cybil Automotive 2004-04-01 8
History
Syntax | History
PostgreSQL
-- Order by the hire_date
SELECT employee_id, first_name, depa
ROW_NUMBER() OVER (
...
10:52:49
PostgreSQL
-- Retrieve the employee_id, first_n
-- hire_date of employees for differ
SELECT
...
10:49:47
PostgreSQL
-- Ordering inside and outside the O
SELECT employee_id, first_name, hire
...
10:46:47
PostgreSQL

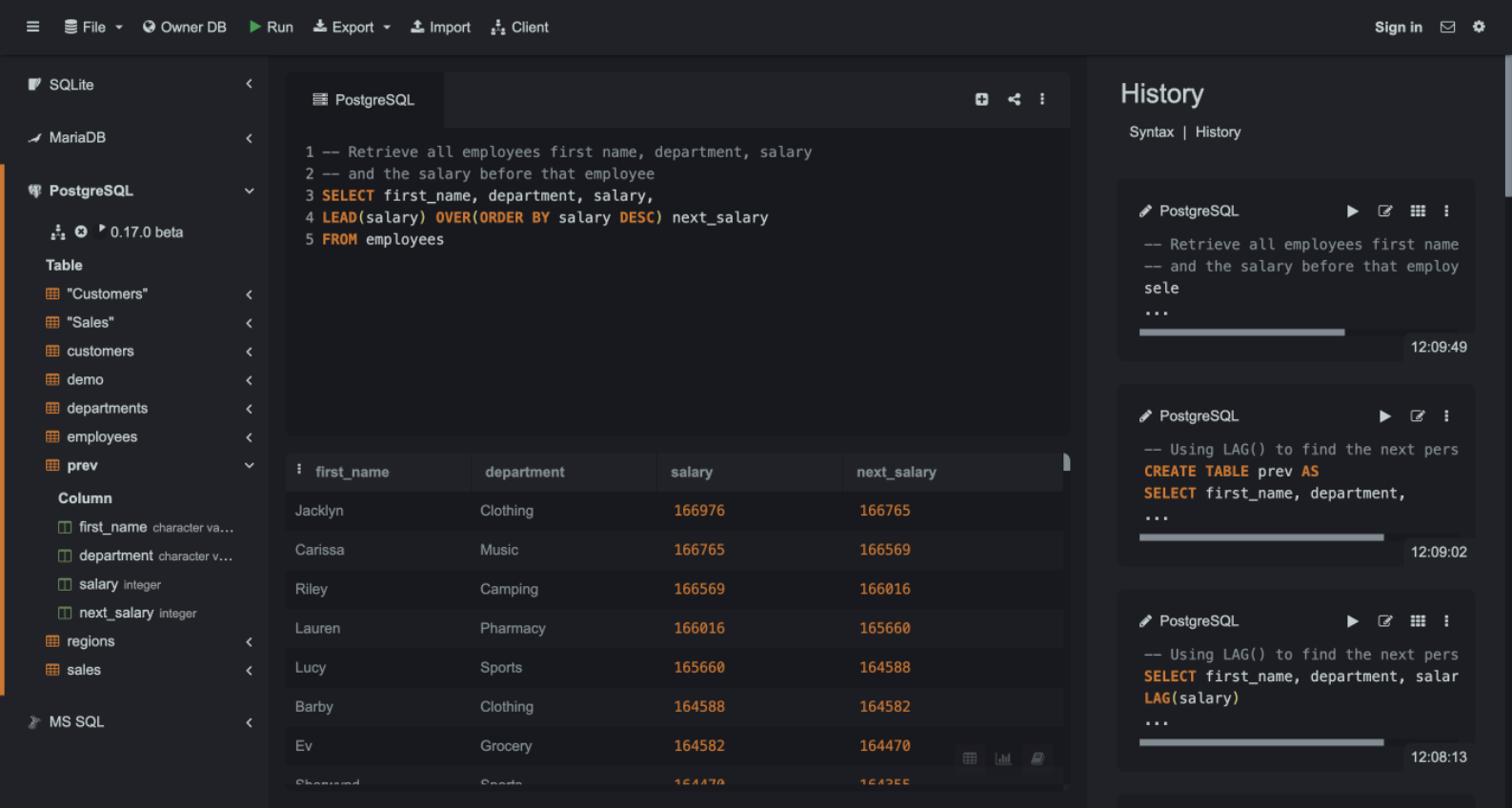

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
Column
Customer_ID characte...
bracket_cust_id chara...
Customer_Name char...
Segment character vary...
Age smallint
Country character varyi...
City character varying(20)
State character varying(20)
Postal_Code integer
Region character varyin...
"Sales"
customers
demo
departments
employees
regions
PostgreSQL
1 -- Joining Sales and Customers based on Customer_ID, then
2 -- Counts the number of purchases made by the same customer and segment
3 WITH customer_purchase AS (
4 SELECT s."Customer_ID", C."Customer_Name", C."Segment",
5 COUNT(*) AS purchase_count
6 FROM "Sales" s
7 JOIN "Customers" C
8 ON s."Customer_ID" = C."Customer_ID"
9 GROUP BY s."Customer_ID", C."Customer_Name", C."Segment"
10)
11 SELECT * FROM customer_purchase
12 ORDER BY "Customer_ID";
Customer_ID
Customer_Name
Segment
purchase_count
AA-10315 Alex Avila Consumer 11
AA-10375 Allen Armold Consumer 15
AA-10480 Andrew Allen Consumer 12
AA-10645 Anna Andreadi Consumer 18
AB-10015 Aaron Bergman Consumer 6
AB-10060 Adam Bellavance Home Office 18
AB-10105 Adrian Barton Consumer 20
AB-10150 Aimee Bixby Consumer 12
History
Syntax | History
PostgreSQL
WITH customer_purchase AS (
SELECT s."Customer_ID", C."Custo
COUNT(*) AS pu
...
11:11:43
PostgreSQL
WITH customer_purchase AS (
SELECT s."Customer_ID", C."Cust
COUNT(*) AS pu
...
Help: Error 42601 syntax error at end of input
11:11:18
PostgreSQL
SELECT "Customer_ID" FROM "Customers
11:10:14

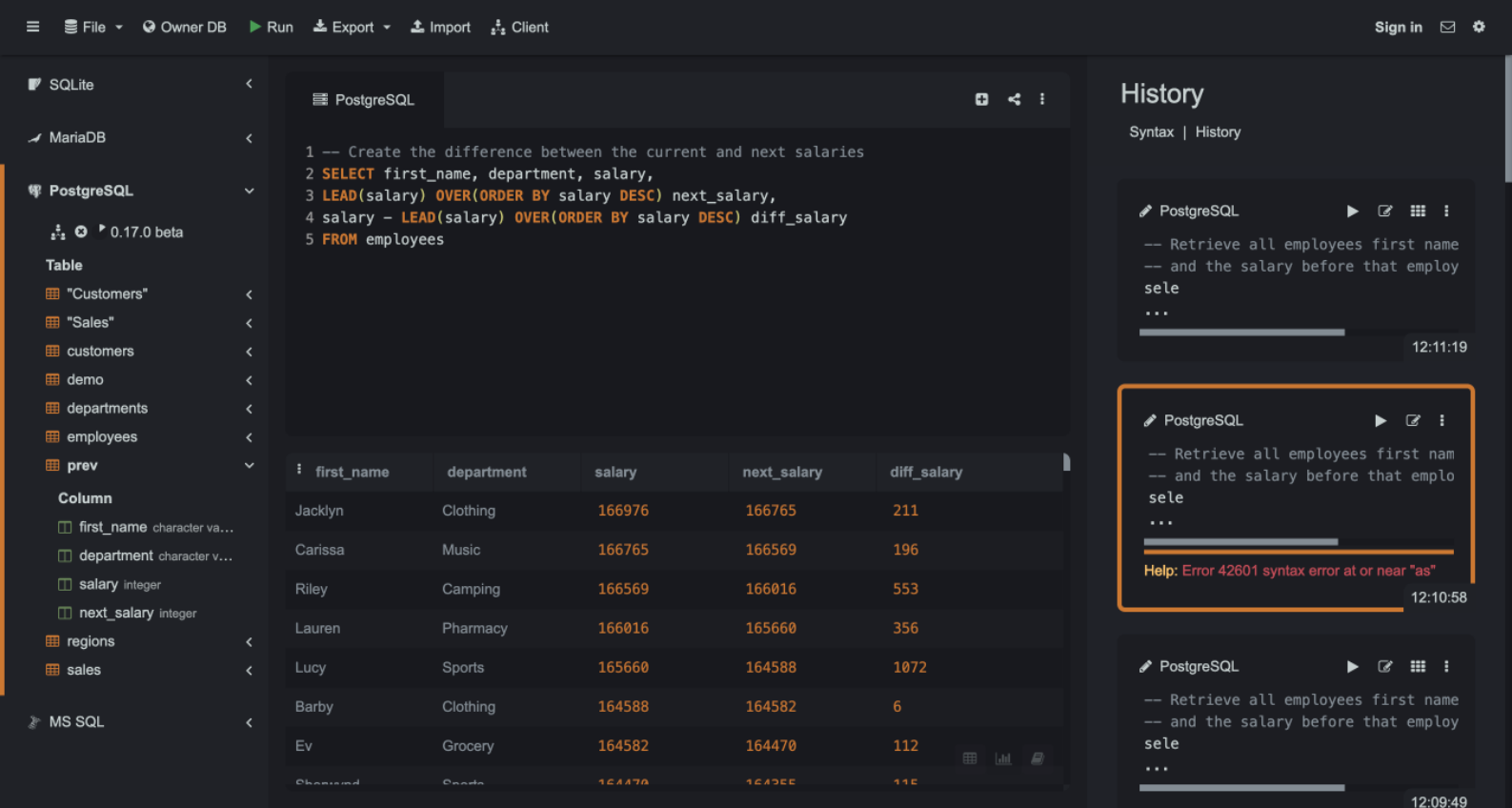

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
regions
sales
MS SQL
PostgreSQL
1 -- Creating a CTE and ordering by purchase count
2 WITH customer_purchase AS (
3 SELECT s."Customer_ID", C."Customer_Name", C."Segment",
4 COUNT(*) AS purchase_count
5 FROM "Sales" s
6 JOIN "Customers" C
7 ON s."Customer_ID" = C."Customer_ID"
8 GROUP BY s."Customer_ID", C."Customer_Name", C."Segment"
9)
10 SELECT "Customer_ID", "Customer_Name", "Segment", purchase_count,
11 ROW_NUMBER() OVER(ORDER BY purchase_count DESC) AS row_n
12 FROM customer_purchase
13 ORDER BY row_n;
Customer_ID
Customer_Name
Segment
purchase_count
row_n
WB-21850
William Brown
Consumer
37
1
MA-17560
Matt Abelman
Home Office
34
2
JL-15835
John Lee
Consumer
34
3
PP-18955
Paul Prost
Home Office
34
4
EH-13765
Edward Hooks
Corporate
32
5
JD-15895
Jonathan Doherty
Corporate
32
6
CK-12205
Chloris Kastensmidt
Consumer
32
7
History
Syntax | History
PostgreSQL
-- Creating a CTE and ordering by pu
WITH customer_purchase AS (
SELECT s."Customer_ID"
...
11:22:26
PostgreSQL
-- Creating a CTE and ordering by pu
WITH customer_purchase AS (
SELECT s."Customer_ID"
...
11:22:04
PostgreSQL
-- Creating a CTE and ordering by pu
WITH customer_purchase AS (
SELECT s."Customer_ID"
...
11:21:18

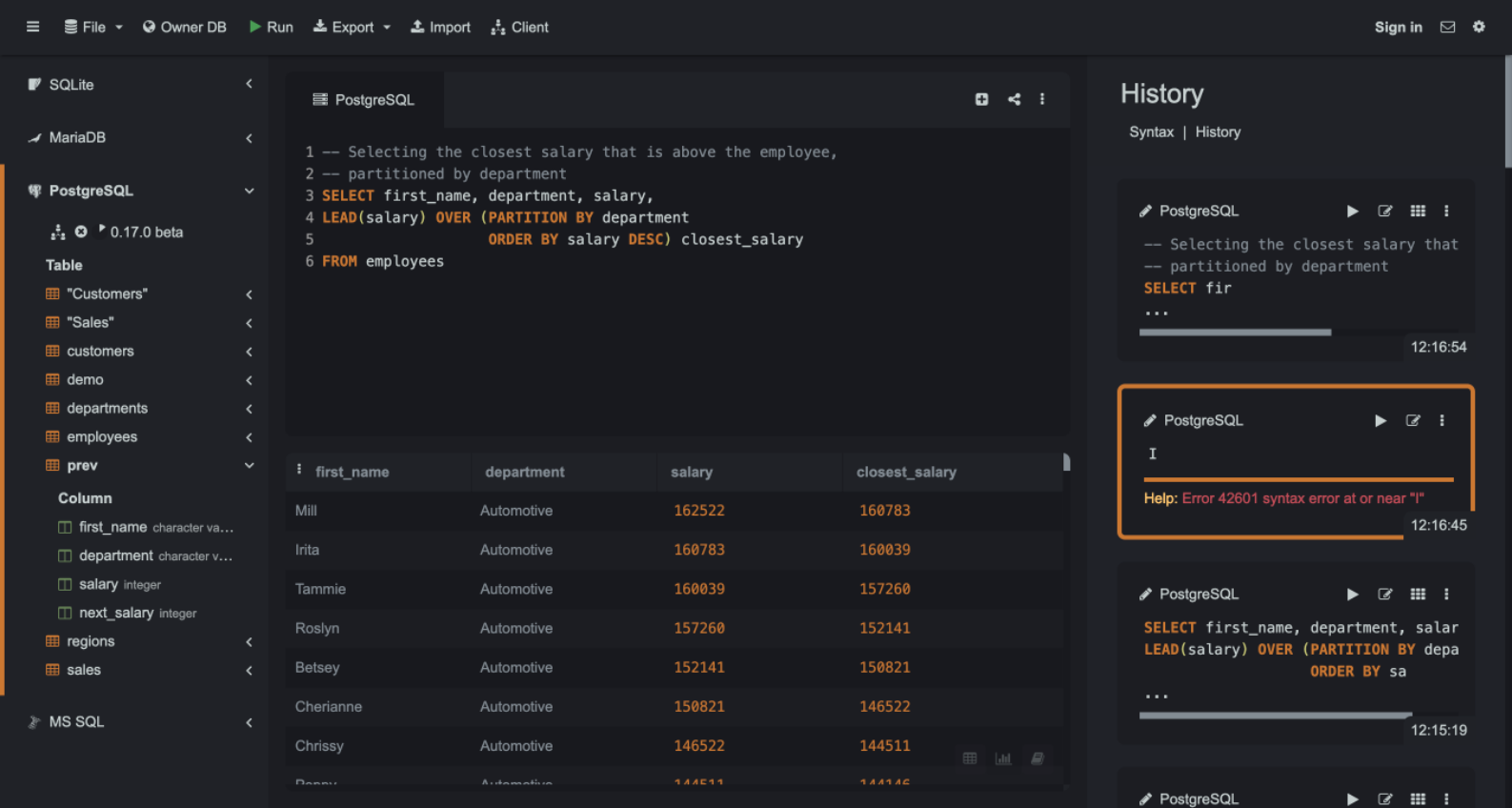

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
regions
sales
MS SQL
PostgreSQL
1 -- Creating a CTE and ordering by purchase count, but partitioning by segment
2 WITH customer_purchase AS (
3 SELECT s."Customer_ID", C."Customer_Name", C."Segment",
4 COUNT(*) AS purchase_count
5 FROM "Sales" s
6 JOIN "Customers" C
7 ON s."Customer_ID" = C."Customer_ID"
8 GROUP BY s."Customer_ID", C."Customer_Name", C."Segment"
9)
10 SELECT "Customer_ID", "Customer_Name", "Segment", purchase_count,
11 ROW_NUMBER() OVER(PARTITION BY "Segment" ORDER BY purchase_count DESC) AS row_n
12 FROM customer_purchase
13 ORDER BY "Segment", row_n;
Customer_ID
Customer_Name
Segment
purchase_count
row_n
JR-15700
Jocasta Rupert
Consumer
1
408
RE-19405
Ricardo Emerson
Consumer
1
409
EH-13765
Edward Hooks
Corporate
32
1
JD-15895
Jonathan Doherty
Corporate
32
2
BM-11650
Brian Moss
Corporate
29
3
SH-19975
Sally Hughsby
Corporate
29
4
KD-16495
Keith Dawkins
Corporate
28
5
History
Syntax | History
PostgreSQL
-- Creating a CTE and ordering by pu
WITH customer_purchase
...
11:27:18
PostgreSQL
-- Creating a CTE and ordering by p
WITH customer_purchase
...
Help: Error 42703 column "segment" does not exist
11:27:10
PostgreSQL
-- Creating a CTE and ordering by p
WITH customer_purchase
...
Help: Error 42703 column "segment" does not exist

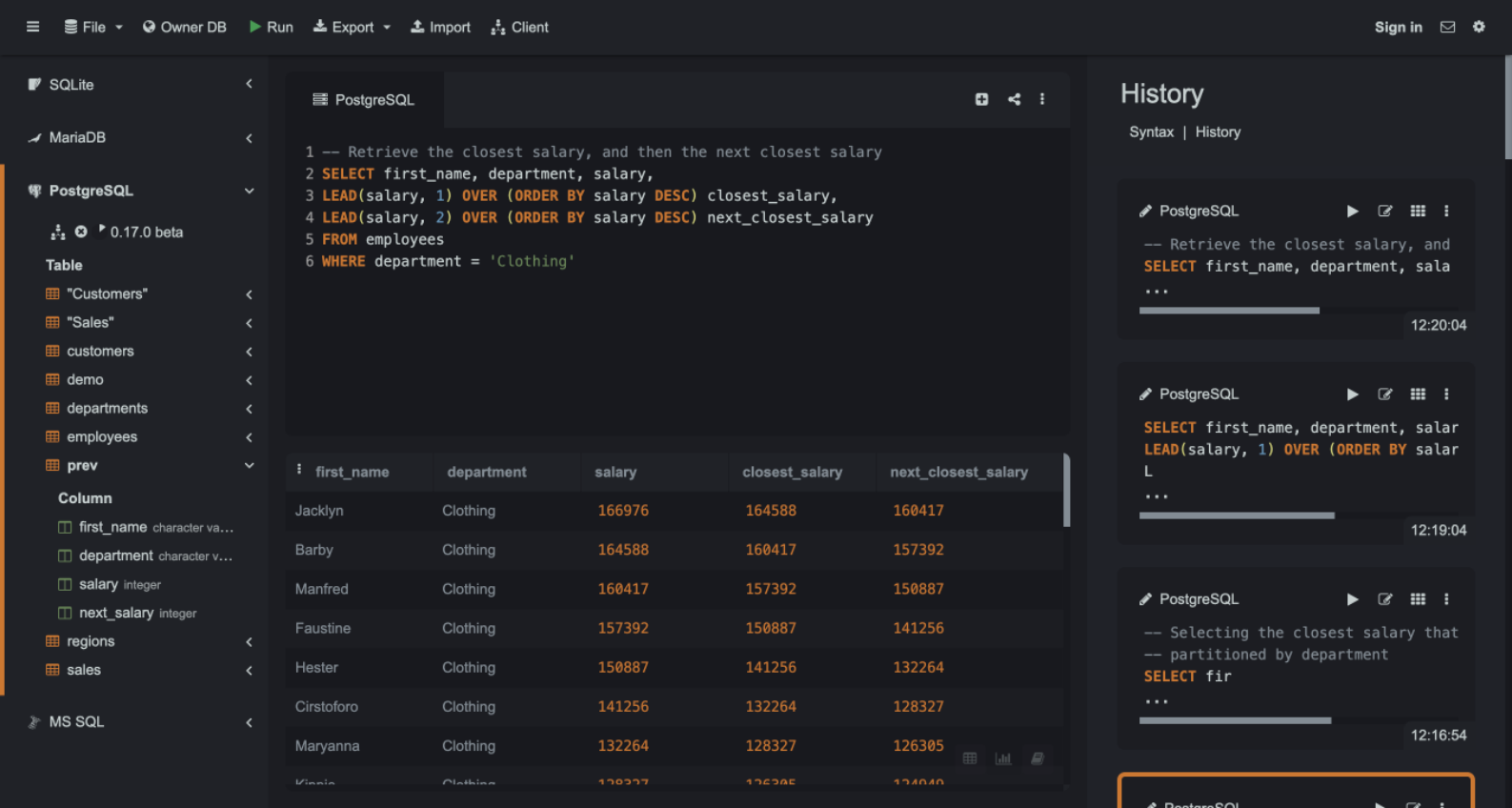

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
regions
sales
MS SQL
PostgreSQL
1 -- Using LEAD() to find the next person's salary
2 SELECT first_name, department, salary,
3 LEAD(salary) OVER() next_salary
4 FROM employees
first_name
department
salary
next_salary
Berrie
Sports
154864
56752
Aeriell
Tools
56752
95313
Sydney
Clothing
95313
119674
Avrom
Phones & Tablets
119674
55307
Feliks
Computers
55307
134501
Bethena
Sports
134501
28995
Ardeen
Clothing
28995
101066
History
Syntax | History
PostgreSQL
SELECT first_name, department, salar
LEAD(salary) OVER() next_salary
FROM employees
12:04:30
PostgreSQL
SELECT first_name, department, sala
LEAD(salary) OVER() next_salary,
FROM employees
Help: Error 42601 syntax error at or near "FROM"
12:04:22
PostgreSQL
-- Creating a CTE and ordering by pu
WITH customer_purchase
...
11:27:18

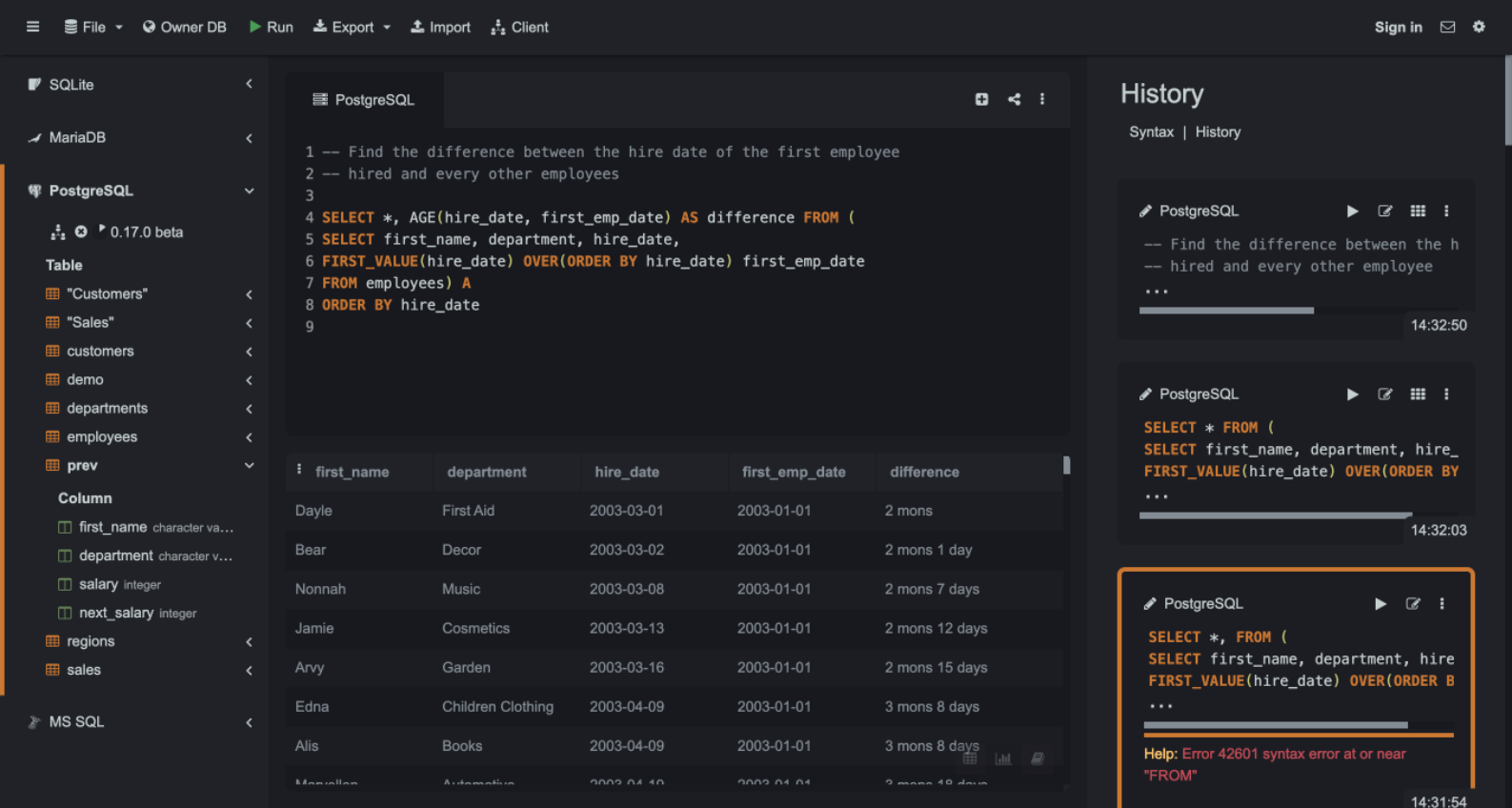

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
regions
sales
MS SQL
PostgreSQL
1 -- Using LAG() to find the next person's salary
2 SELECT first_name, department, salary,
3 LAG(salary) OVER() next_salary
4 FROM employees
first_name
department
salary
next_salary
Berrie
Sports
154864
NULL
Aeriell
Tools
56752
154864
Sydney
Clothing
95313
56752
Avrom
Phones & Tablets
119674
95313
Feliks
Computers
55307
119674
Bethena
Sports
134501
55307
Ardeen
Clothing
28995
134501
History
Syntax | History
PostgreSQL
-- Using LAG() to find the next pers
SELECT first_name, department, salar
LAG(salary)
...
12:08:13
PostgreSQL
-- Using LEAD() to find the next per
SELECT first_name, department, salar
LAG(salary)
...
12:07:09
PostgreSQL
SELECT first_name, department, salar
LEAD(salary) OVER() next_salary
FROM employees
12:04:30

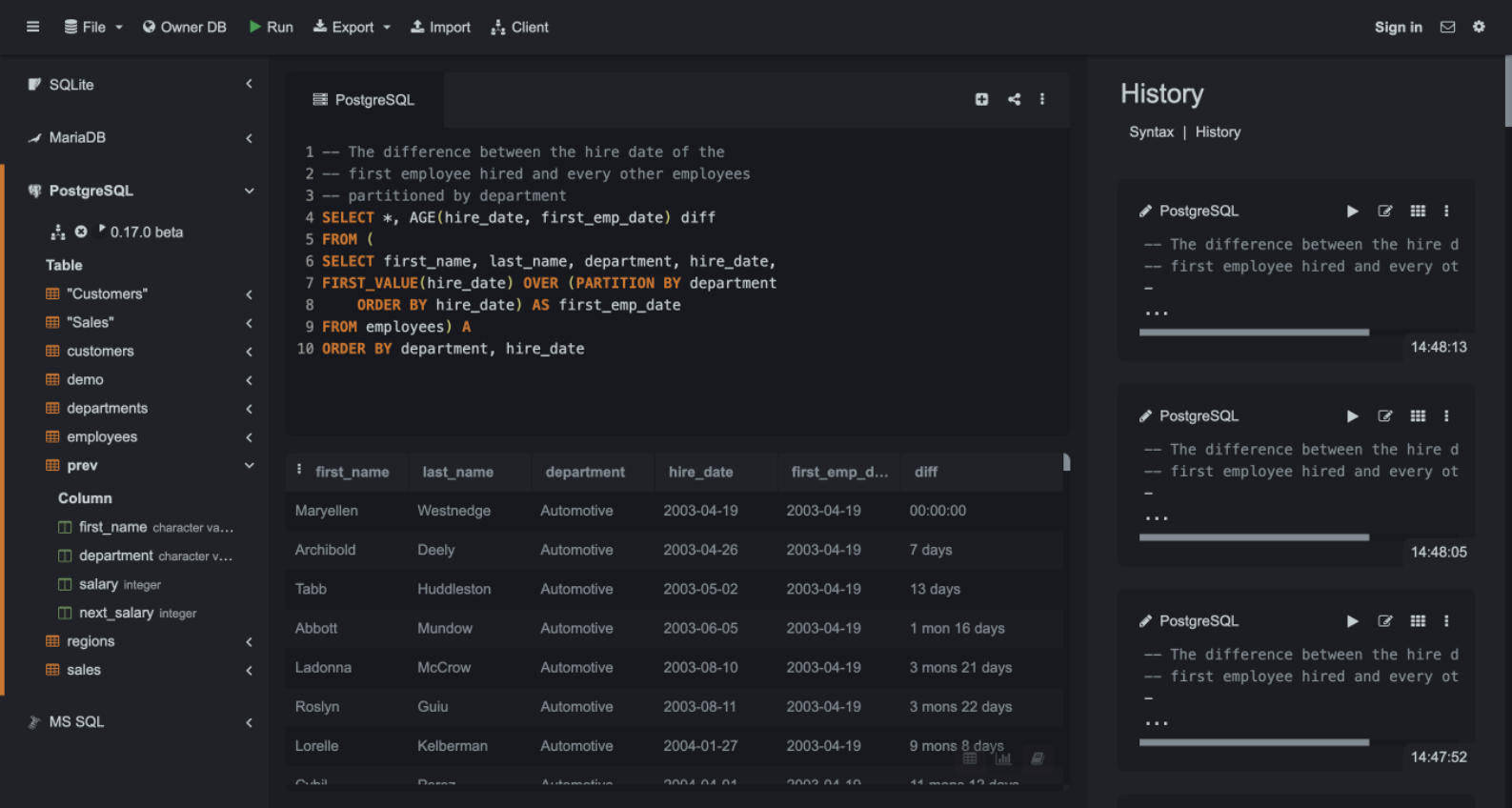

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
Column
first_name character va...
department character v...
salary integer
next_salary integer
regions
sales
MS SQL
PostgreSQL
1 -- Retrieve all employees first name, department, salary
2 -- and the salary before that employee
3 SELECT first_name, department, salary,
4 LEAD(salary) OVER(ORDER BY salary DESC) next_salary
5 FROM employees
first_name
department
salary
next_salary
Jacklyn Clothing 166976 166765
Carissa Music 166765 166569
Riley Camping 166569 166016
Lauren Pharmacy 166016 165660
Lucy Sports 165660 164588
Barby Clothing 164588 164582
Ev Grocery 164582 164470
History
Syntax | History
PostgreSQL
-- Retrieve all employees first name
-- and the salary before that employ
sele
...
12:09:49
PostgreSQL
-- Using LAG() to find the next pers
CREATE TABLE prev AS
SELECT first_name, department,
...
12:09:02
PostgreSQL
-- Using LAG() to find the next pers
SELECT first_name, department, salar
LAG(salary)
...
12:08:13

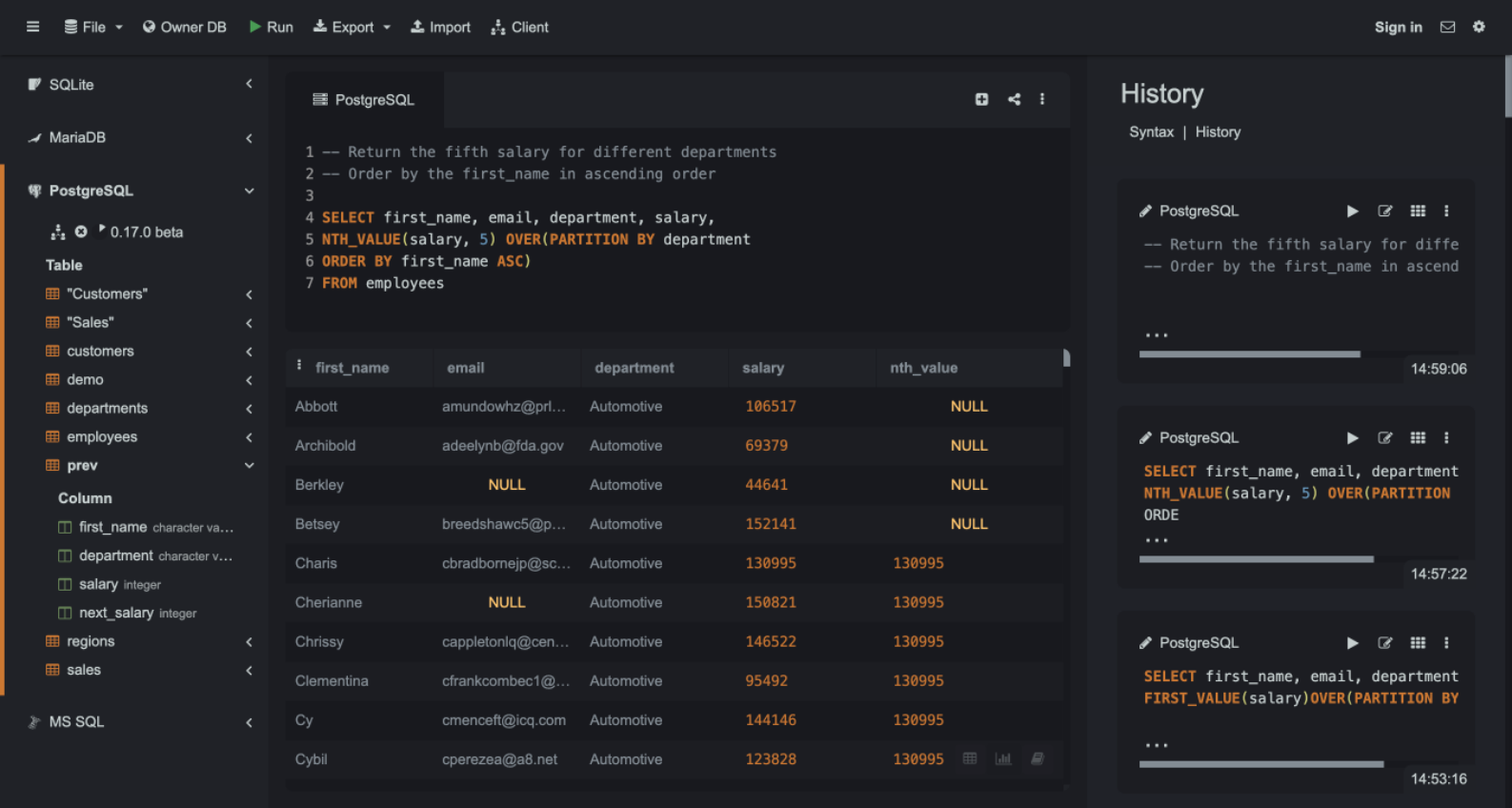

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
Column
first_name character va...
department character v...
salary integer
next_salary integer
regions
sales
MS SQL
PostgreSQL
1 -- Create the difference between the current and next salaries
2 SELECT first_name, department, salary,
3 LEAD(salary) OVER(ORDER BY salary DESC) next_salary,
4 salary - LEAD(salary) OVER(ORDER BY salary DESC) diff_salary
5 FROM employees
first_name
department
salary
next_salary
diff_salary
Jacklyn Clothing 166976 166765 211
Carissa Music 166765 166569 196
Riley Camping 166569 166016 553
Lauren Pharmacy 166016 165660 356
Lucy Sports 165660 164588 1072
Barby Clothing 164588 164582 6
Ev Grocery 164582 164470 112
History
Syntax | History
PostgreSQL
-- Retrieve all employees first name
-- and the salary before that employ
sele
...
12:11:19
PostgreSQL
-- Retrieve all employees first nam
-- and the salary before that emplo
sele
...
Help: Error 42601 syntax error at or near "as"
12:10:58
PostgreSQL
-- Retrieve all employees first name
-- and the salary before that employ
sele
...
12:09:49

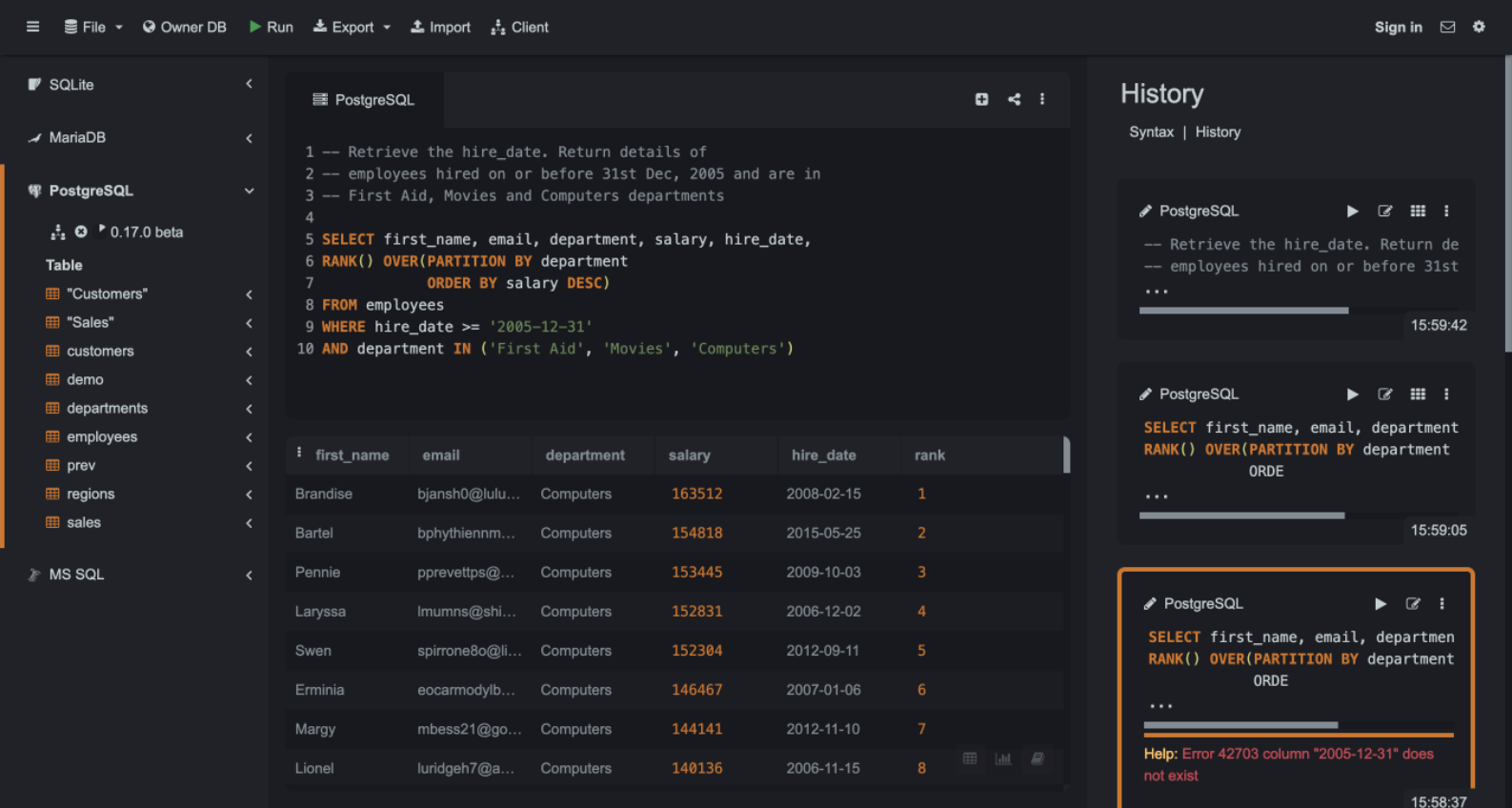

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
Column
first_name character va...
department character v...
salary integer
next_salary integer
regions
sales
MS SQL
PostgreSQL
1 -- Selecting the closest salary that is above the employee,
2 -- partitioned by department
3 SELECT first_name, department, salary,
4 LEAD(salary) OVER (PARTITION BY department
5 ORDER BY salary DESC) closest_salary
6 FROM employees
first_name
department
salary
closest_salary
Mill Automotive 162522 160783
Irita Automotive 160783 160039
Tammie Automotive 160039 157260
Roslyn Automotive 157260 152141
Betsey Automotive 152141 150821
Cherianne Automotive 150821 146522
Chrissy Automotive 146522 144511
History
Syntax | History
PostgreSQL
-- Selecting the closest salary that
-- partitioned by department
SELECT fir
...
12:16:54
PostgreSQL
I
Help: Error 42601 syntax error at or near "I"
12:16:45
PostgreSQL
SELECT first_name, department, salar
LEAD(salary) OVER (PARTITION BY depa
ORDER BY sa
...
12:15:19
PostgreSQL

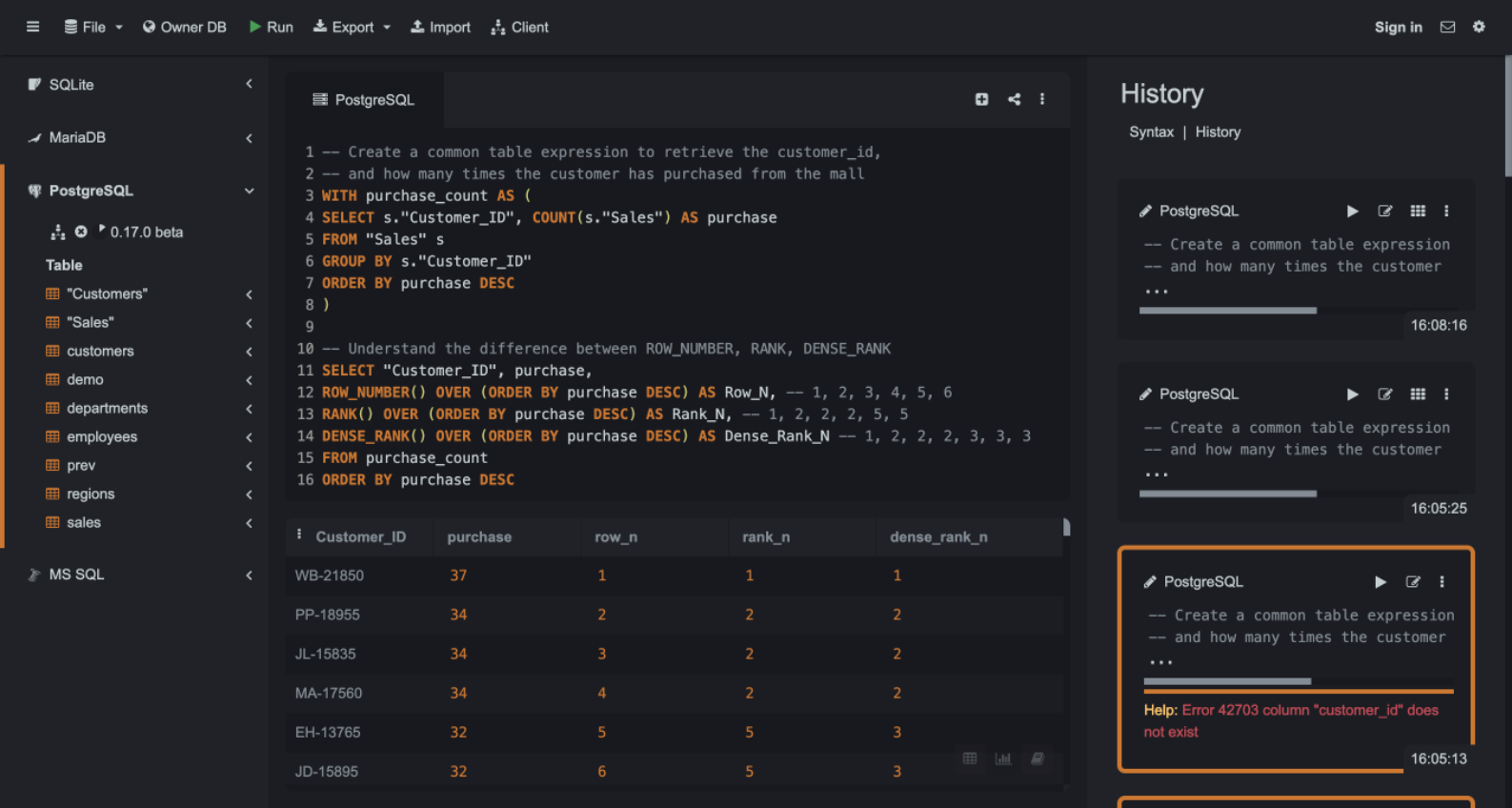

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
Column
first_name character va...
department character v...
salary integer
next_salary integer
regions
sales
MS SQL
PostgreSQL
1 -- Retrieve the closest salary, and then the next closest salary
2 SELECT first_name, department, salary,
3 LEAD(salary, 1) OVER (ORDER BY salary DESC) closest_salary,
4 LEAD(salary, 2) OVER (ORDER BY salary DESC) next_closest_salary
5 FROM employees
6 WHERE department = 'Clothing'
first_name
department
salary
closest_salary
next_closest_salary
Jacklyn Clothing 166976 164588 160417
Barby Clothing 164588 160417 157392
Manfred Clothing 160417 157392 150887
Faustine Clothing 157392 150887 141256
Hester Clothing 150887 141256 132264
Cirstoforo Clothing 141256 132264 128327
Maryanna Clothing 132264 128327 126305
History
Syntax | History
PostgreSQL
-- Retrieve the closest salary, and
SELECT first_name, department, sala
...
12:20:04
PostgreSQL
SELECT first_name, department, salar
LEAD(salary, 1) OVER (ORDER BY salar
L
...
12:19:04
PostgreSQL
-- Selecting the closest salary that
-- partitioned by department
SELECT fir
...
12:16:54

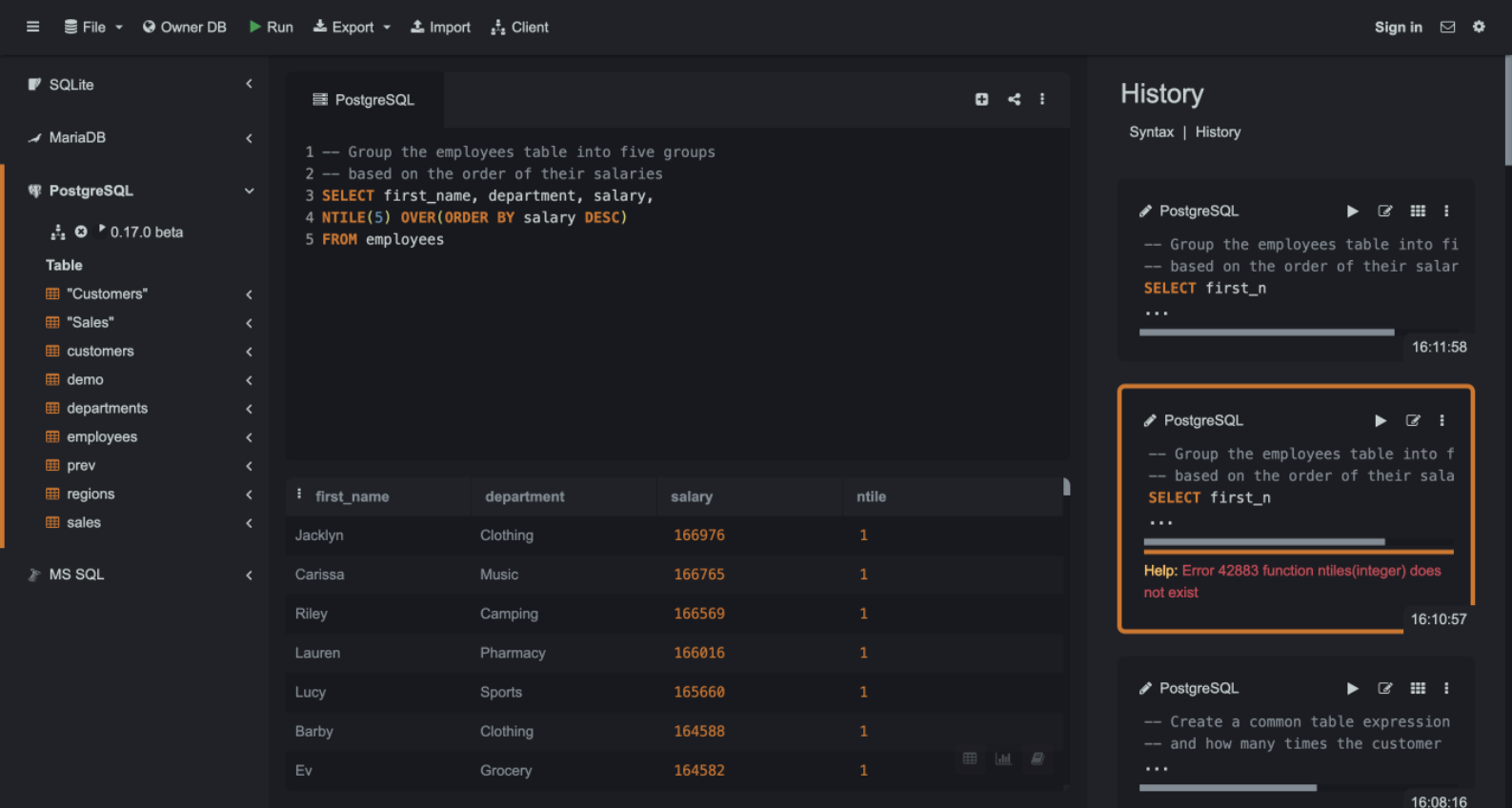

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
Column
first_name character va...
department character v...
salary integer
next_salary integer
regions
sales
MS SQL
PostgreSQL
1 -- Find the difference between the hire date of the first employee
2 -- hired and every other employees
3
4 SELECT *, AGE(hire_date, first_emp_date) AS difference FROM (
5 SELECT first_name, department, hire_date,
6 FIRST_VALUE(hire_date) OVER(ORDER BY hire_date) first_emp_date
7 FROM employees) A
8 ORDER BY hire_date
9
first_name
department
hire_date
first_emp_date
difference
Dayle
First Aid
2003-03-01
2003-01-01
2 mons
Bear
Decor
2003-03-02
2003-01-01
2 mons 1 day
Nonnah
Music
2003-03-08
2003-01-01
2 mons 7 days
Jamie
Cosmetics
2003-03-13
2003-01-01
2 mons 12 days
Arvy
Garden
2003-03-16
2003-01-01
2 mons 15 days
Edna
Children Clothing
2003-04-09
2003-01-01
3 mons 8 days
Alis
Books
2003-04-09
2003-01-01
3 mons 8 days
History
Syntax | History
PostgreSQL
-- Find the difference between the h
-- hired and every other employee
...
14:32:50
PostgreSQL
SELECT * FROM (
SELECT first_name, department, hire_
FIRST_VALUE(hire_date) OVER(ORDER BY
...
14:32:03
PostgreSQL
SELECT *, FROM (
SELECT first_name, department, hire
FIRST_VALUE(hire_date) OVER(ORDER B
...
Help: Error 42601 syntax error at or near "FROM"
14:31:54

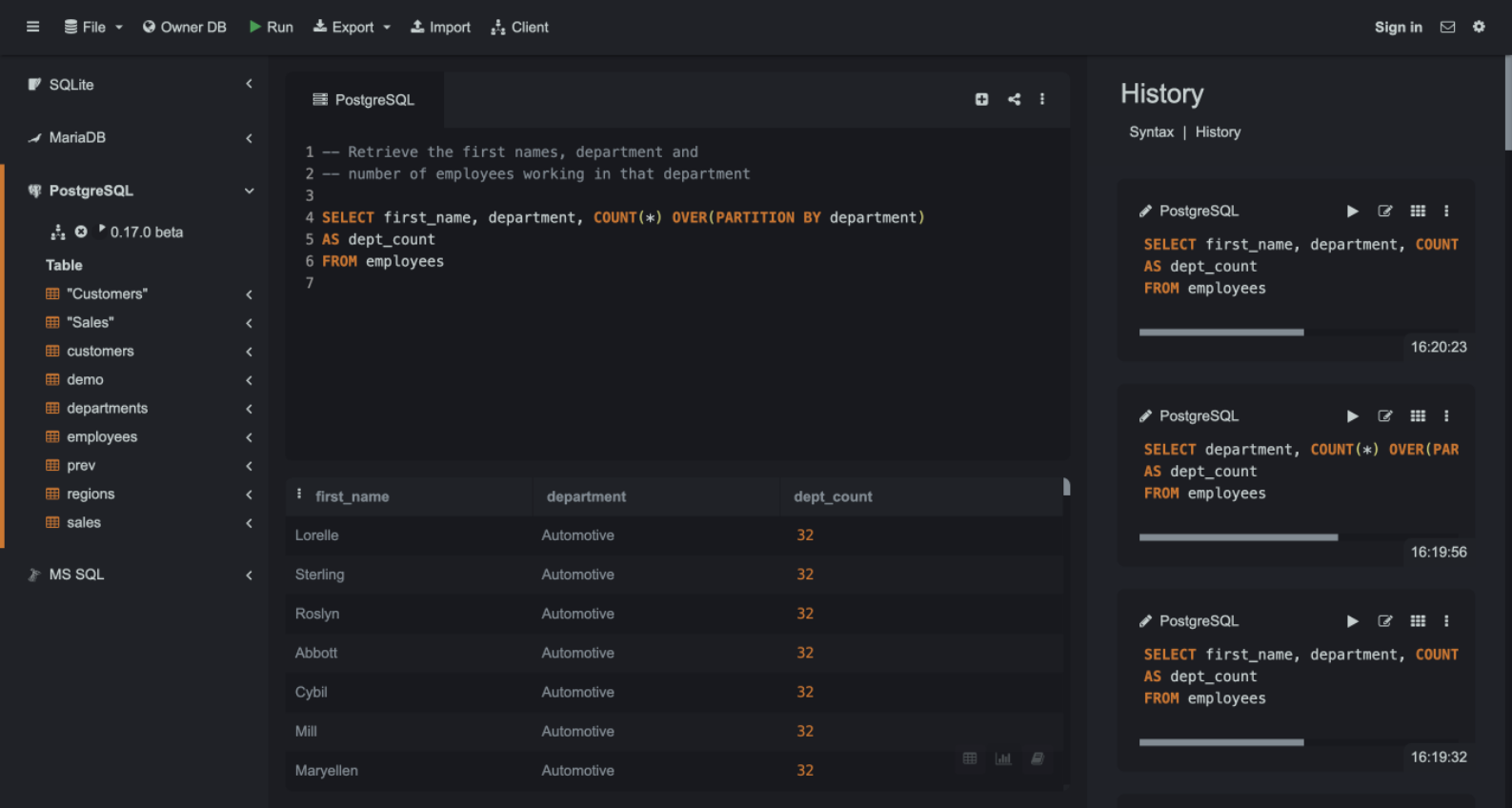

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
Column
first_name character va...
department character v...
salary integer
next_salary integer
regions
sales
MS SQL
PostgreSQL
1 -- The difference between the hire date of the
2 -- first employee hired and every other employees
3 -- partitioned by department
4 SELECT *, AGE(hire_date, first_emp_date) diff
5 FROM (
6 SELECT first_name, last_name, department, hire_date,
7 FIRST_VALUE(hire_date) OVER (PARTITION BY department
8 ORDER BY hire_date) AS first_emp_date
9 FROM employees) A
10 ORDER BY department, hire_date
first_name
last_name
department
hire_date
first_emp_d...
diff
Maryellen Westnedge Automotive 2003-04-19 2003-04-19 00:00:00
Archibold Deely Automotive 2003-04-26 2003-04-19 7 days
Tabb Huddleston Automotive 2003-05-02 2003-04-19 13 days
Abbott Mundow Automotive 2003-06-05 2003-04-19 1 mon 16 days
Ladonna McCrow Automotive 2003-08-10 2003-04-19 3 mons 21 days
Roslyn Guiu Automotive 2003-08-11 2003-04-19 3 mons 22 days
Lorelle Kelberman Automotive 2004-01-27 2003-04-19 9 mons 8 days
History
Syntax | History
PostgreSQL
-- The difference between the hire d
-- first employee hired and every ot
14:48:13
PostgreSQL
-- The difference between the hire d
-- first employee hired and every ot
14:48:05
PostgreSQL
-- The difference between the hire d
-- first employee hired and every ot
14:47:52

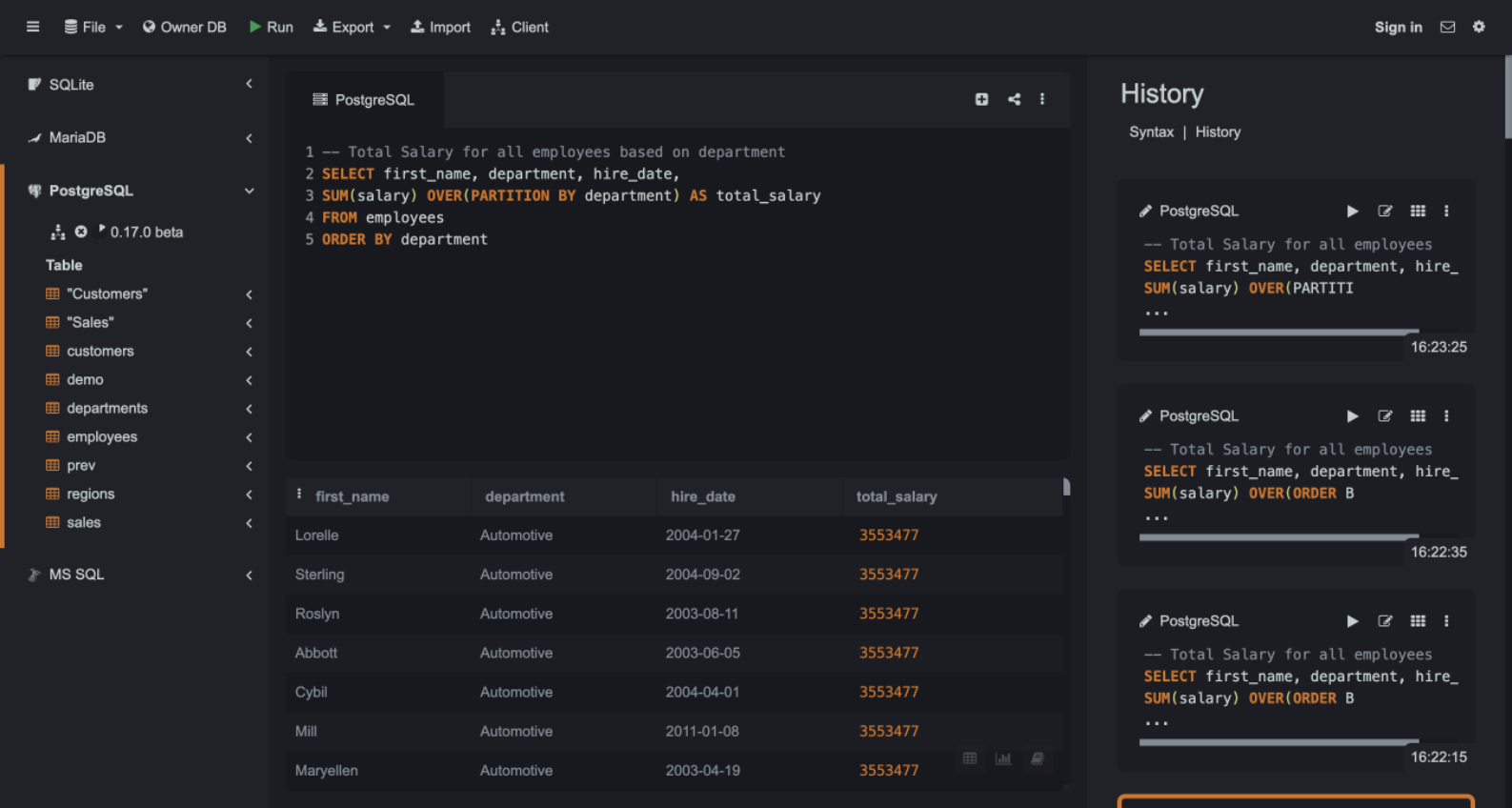

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
Column
first_name character va...
department character v...
salary integer
next_salary integer
regions
sales
MS SQL
PostgreSQL
1 -- Return the fifth salary for different departments
2 -- Order by the first_name in ascending order
3
4 SELECT first_name, email, department, salary,
5 NTH_VALUE(salary, 5) OVER(PARTITION BY department
6 ORDER BY first_name ASC)
7 FROM employees
first_name email department salary nth_value
Abbott amundowhz@prl... Automotive 106517 NULL
Archibold adeelynb@fda.gov Automotive 69379 NULL
Berkley NULL Automotive 44641 NULL
Betsey breedshawc5@p... Automotive 152141 NULL
Charis cbradbornejp@sc... Automotive 130995 130995
Cherianne NULL Automotive 150821 130995
Chrissy cappletonlq@cen... Automotive 146522 130995
Clementina cfrankcombec1@... Automotive 95492 130995
Cy cmenceft@icq.com Automotive 144146 130995
Cybil cperezea@a8.net Automotive 123828 130995
History
Syntax | History
PostgreSQL
-- Return the fifth salary for diffe
-- Order by the first_name in ascend
...
14:59:06
PostgreSQL
SELECT first_name, email, department
NTH_VALUE(salary, 5) OVER(PARTITION
ORDE
...
14:57:22
PostgreSQL
SELECT first_name, email, department
FIRST_VALUE(salary)OVER(PARTITION BY
...
14:53:16

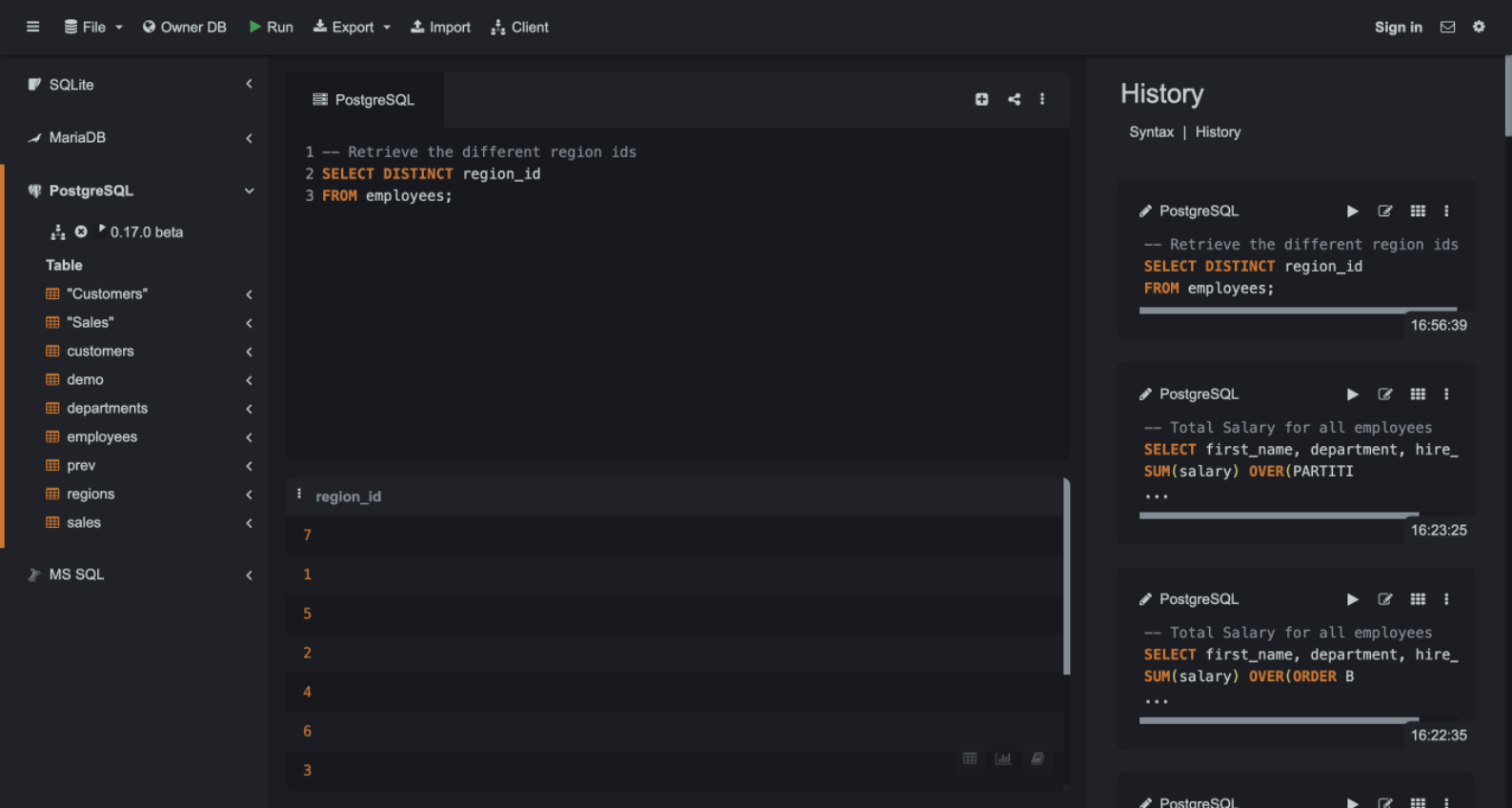

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
regions
sales
MS SQL
PostgreSQL
1 -- Retrieve the hire_date. Return details of
2 -- employees hired on or before 31st Dec, 2005 and are in
3 -- First Aid, Movies and Computers departments
4
5 SELECT first_name, email, department, salary, hire_date,
6 RANK() OVER(PARTITION BY department
7 ORDER BY salary DESC)
8 FROM employees
9 WHERE hire_date >= '2005-12-31'
10 AND department IN ('First Aid', 'Movies', 'Computers')
first_name email department salary hire_date rank
Brandise bjansh0@lulu... Computers 163512 2008-02-15 1
Bartel bphythiennm... Computers 154818 2015-05-25 2
Pennie pprevettps@... Computers 153445 2009-10-03 3
Laryssa lmumns@shi... Computers 152831 2006-12-02 4
Swen spirrone8o@li... Computers 152304 2012-09-11 5
Erminia eocarmodylb... Computers 146467 2007-01-06 6
Margy mbess21@go... Computers 144141 2012-11-10 7
Lionel luridgeh7@a... Computers 140136 2006-11-15 8
History
Syntax | History
PostgreSQL
-- Retrieve the hire_date. Return de
-- employees hired on or before 31st
...
15:59:42
PostgreSQL
SELECT first_name, email, department
RANK() OVER(PARTITION BY department
ORDE
...
15:59:05
PostgreSQL
SELECT first_name, email, departmen
RANK() OVER(PARTITION BY department
ORDE
...
Help: Error 42703 column "2005-12-31" does not exist
15:58:37

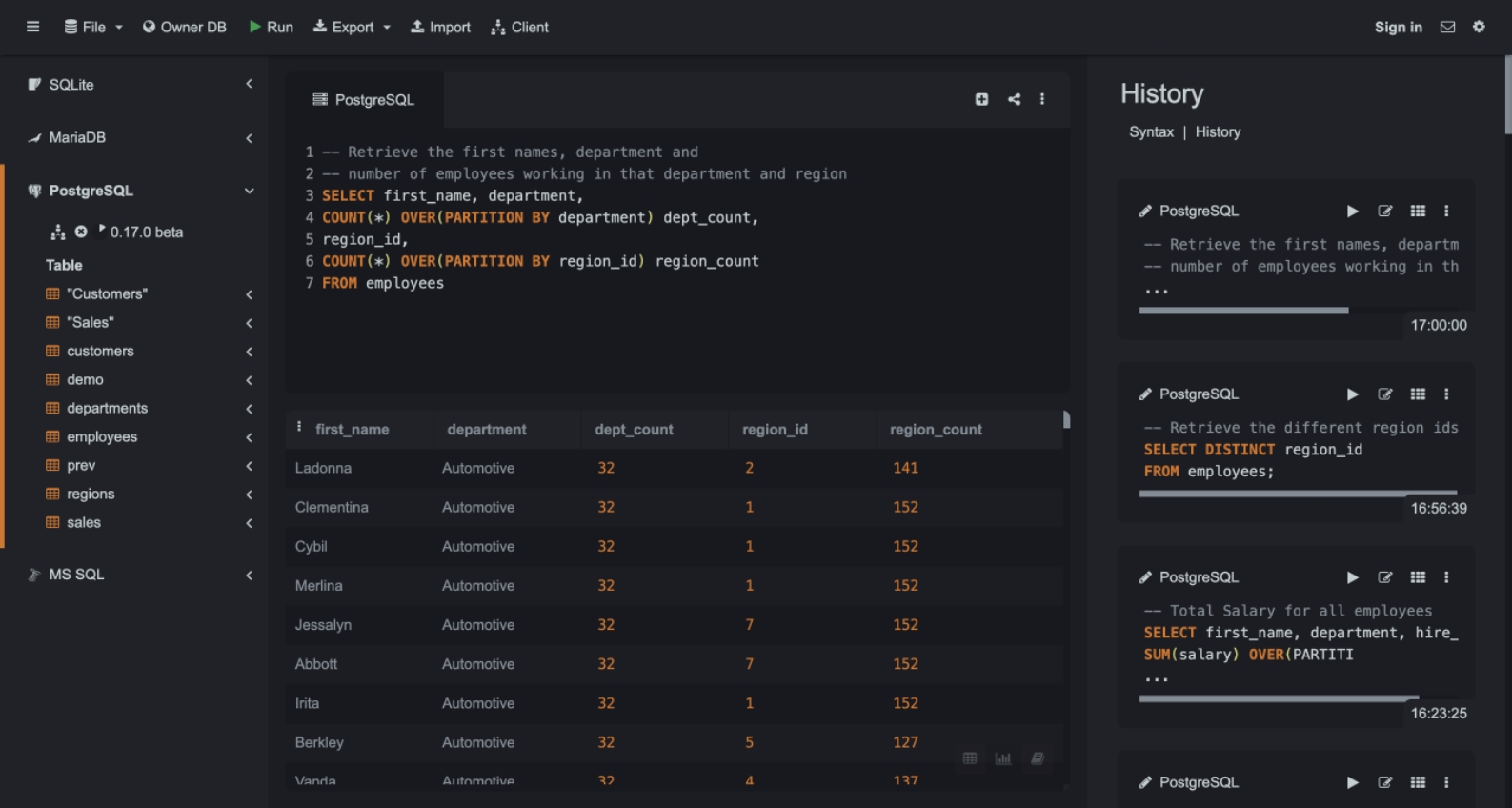

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
regions
sales
MS SQL
PostgreSQL
1 -- Create a common table expression to retrieve the customer_id,
2 -- and how many times the customer has purchased from the mall
3 WITH purchase_count AS (
4 SELECT s."Customer_ID", COUNT(s."Sales") AS purchase
5 FROM "Sales" s
6 GROUP BY s."Customer_ID"
7 ORDER BY purchase DESC
8)
9
10 -- Understand the difference between ROW_NUMBER, RANK, DENSE_RANK
11 SELECT "Customer_ID", purchase,
12 ROW_NUMBER() OVER (ORDER BY purchase DESC) AS Row_N, -- 1, 2, 3, 4, 5, 6
13 RANK() OVER (ORDER BY purchase DESC) AS Rank_N, -- 1, 2, 2, 2, 5, 5
14 DENSE_RANK() OVER (ORDER BY purchase DESC) AS Dense_Rank_N -- 1, 2, 2, 2, 3, 3, 3
15 FROM purchase_count
16 ORDER BY purchase DESC
Customer_ID
purchase
row_n
rank_n
dense_rank_n
WB-21850 37 1 1 1
PP-18955 34 2 2 2
JL-15835 34 3 2 2
MA-17560 34 4 2 2
EH-13765 32 5 5 3
JD-15895 32 6 5 3
History
Syntax | History
PostgreSQL
-- Create a common table expression
-- and how many times the customer
...
16:08:16
PostgreSQL
-- Create a common table expression
-- and how many times the customer
...
16:05:25
PostgreSQL
-- Create a common table expression
-- and how many times the customer
...
Help: Error 42703 column "customer_id" does not exist
16:05:13

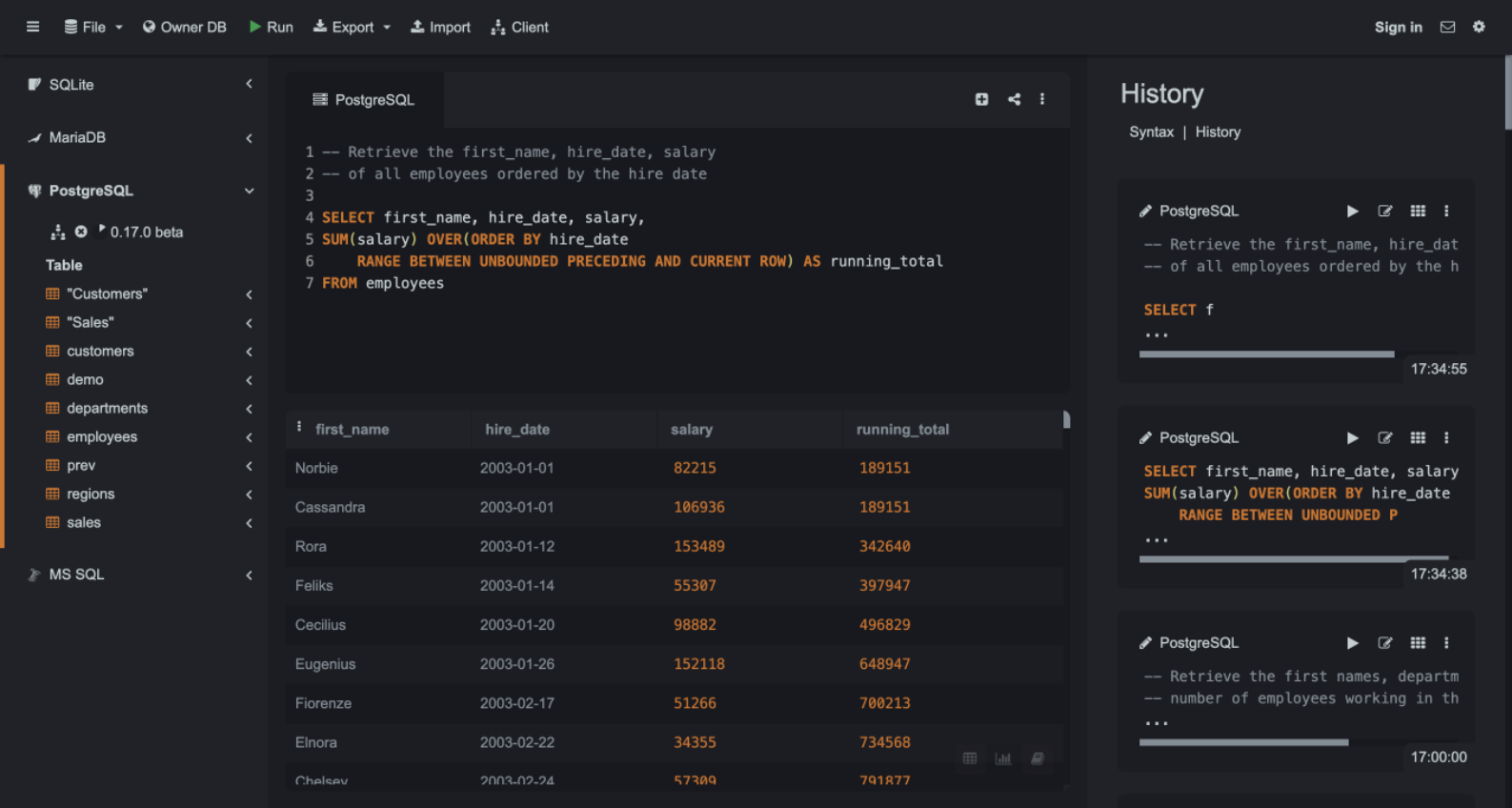

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
regions
sales
MS SQL
PostgreSQL
1 -- Group the employees table into five groups
2 -- based on the order of their salaries
3 SELECT first_name, department, salary,
4 NTILE(5) OVER(ORDER BY salary DESC)
5 FROM employees
first_name
department
salary
ntile
Jacklyn
Clothing
166976
1
Carissa
Music
166765
1
Riley
Camping
166569
1
Lauren
Pharmacy
166016
1
Lucy
Sports
165660
1
Barby
Clothing
164588
1
Ev
Grocery
164582
1
History
Syntax | History
PostgreSQL
-- Group the employees table into fi
-- based on the order of their salar
SELECT first_n
...
16:11:58
PostgreSQL
-- Group the employees table into f
-- based on the order of their sala
SELECT first_n
...
Help: Error 42883 function ntiles(integer) does not exist
16:10:57
PostgreSQL
-- Create a common table expression
-- and how many times the customer
...
16:08:16

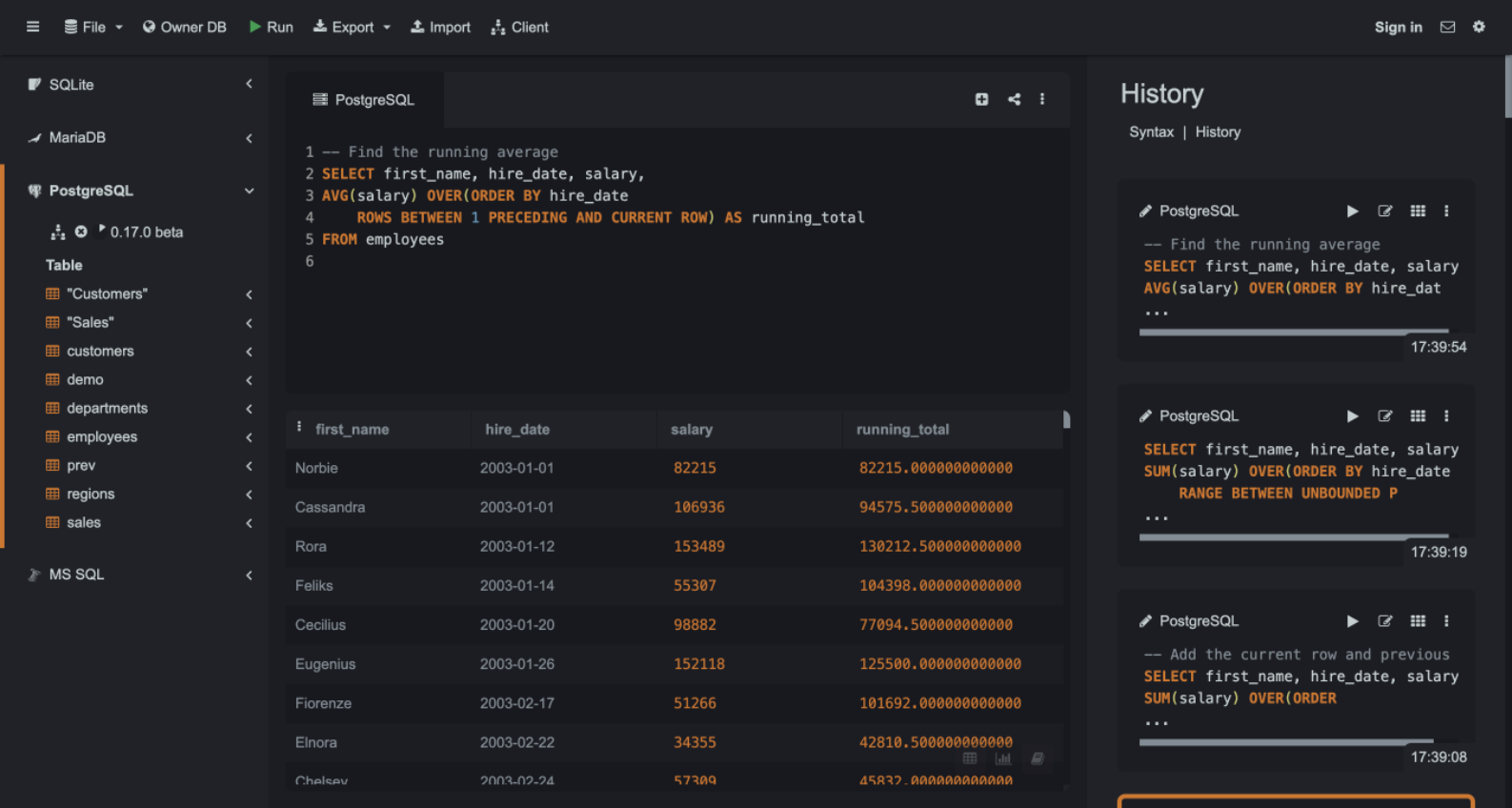

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
regions
sales
MS SQL
PostgreSQL
1 -- Retrieve the first names, department and
2 -- number of employees working in that department
3
4 SELECT first_name, department, COUNT(*) OVER(PARTITION BY department)
5 AS dept_count
6 FROM employees
7
first_name
department
dept_count
Lorelle Automotive 32
Sterling Automotive 32
Roslyn Automotive 32
Abbott Automotive 32
Cybil Automotive 32
Mill Automotive 32
Maryellen Automotive 32
History
Syntax | History
PostgreSQL
SELECT first_name, department, COUNT
AS dept_count
FROM employees
16:20:23
PostgreSQL
SELECT department, COUNT(*) OVER(PAR
AS dept_count
FROM employees
16:19:56
PostgreSQL
SELECT first_name, department, COUNT
AS dept_count
FROM employees
16:19:32

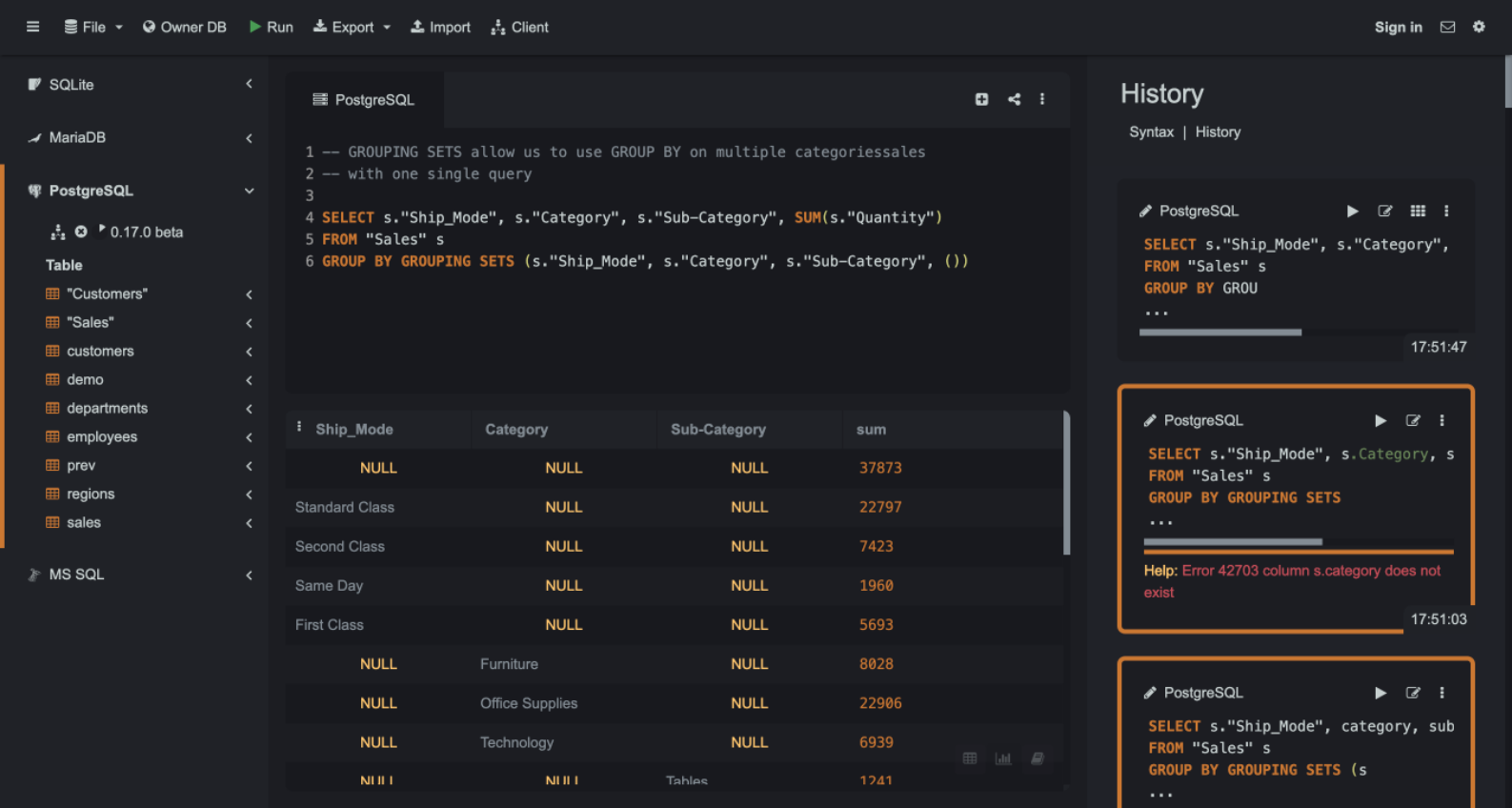

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
regions
sales
MS SQL
PostgreSQL
1 -- Total Salary for all employees based on department
2 SELECT first_name, department, hire_date,
3 SUM(salary) OVER(PARTITION BY department) AS total_salary
4 FROM employees
5 ORDER BY department
first_name
department
hire_date
total_salary
Lorelle
Automotive
2004-01-27
3553477
Sterling
Automotive
2004-09-02
3553477
Roslyn
Automotive
2003-08-11
3553477
Abbott
Automotive
2003-06-05
3553477
Cybil
Automotive
2004-04-01
3553477
Mill
Automotive
2011-01-08
3553477
Maryellen
Automotive
2003-04-19
3553477
History
Syntax | History
PostgreSQL
-- Total Salary for all employees
SELECT first_name, department, hire_
SUM(salary) OVER(PARTITI
...
16:23:25
PostgreSQL
-- Total Salary for all employees
SELECT first_name, department, hire_
SUM(salary) OVER(ORDER B
...
16:22:35
PostgreSQL
-- Total Salary for all employees
SELECT first_name, department, hire_
SUM(salary) OVER(ORDER B
...
16:22:15

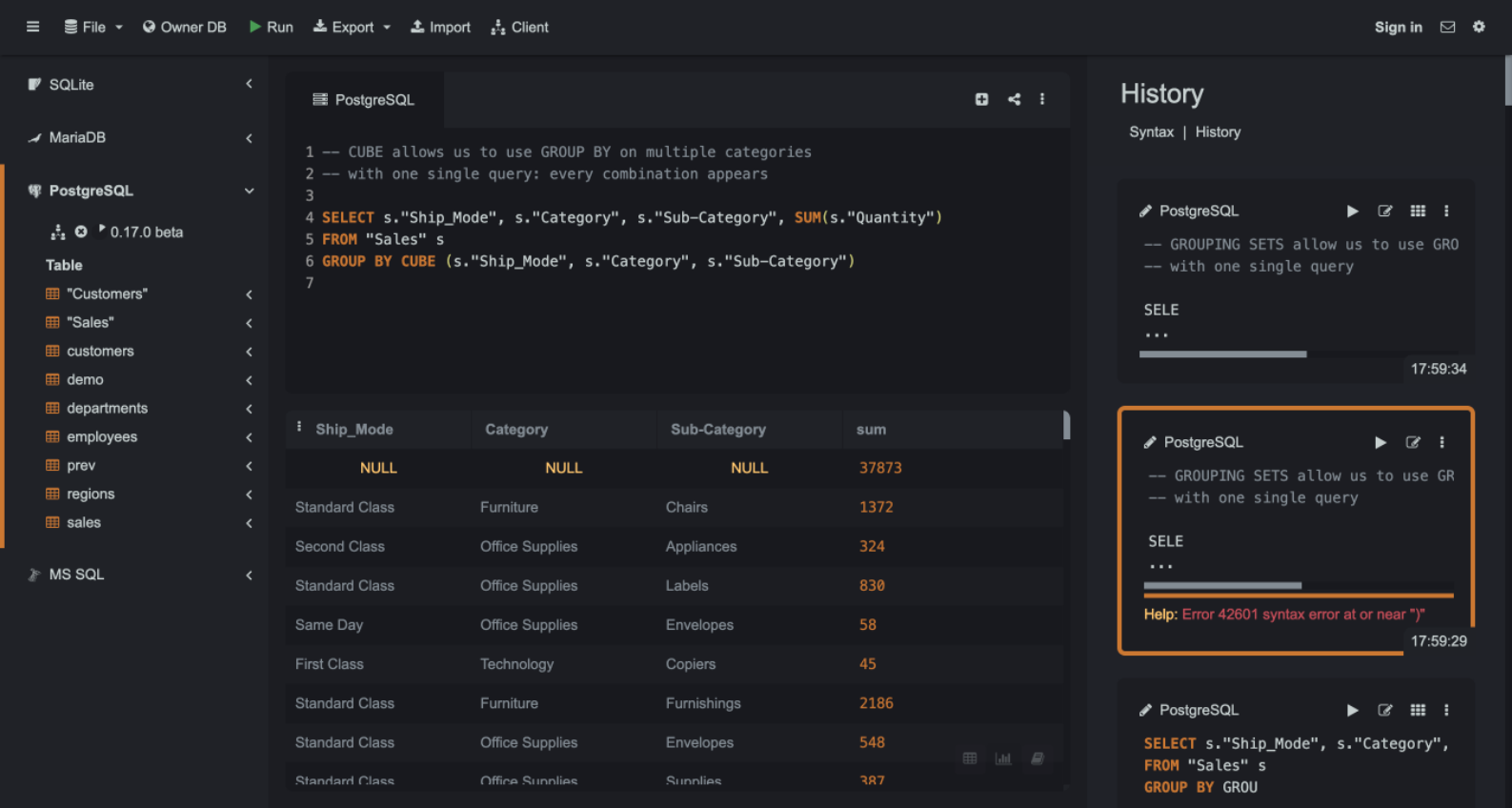

File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
regions
sales
MS SQL
PostgreSQL
1 -- Retrieve the different region ids
2 SELECT DISTINCT region_id
3 FROM employees;
region_id
7
1
5
2
4
6
3
History
Syntax | History
PostgreSQL
-- Retrieve the different region ids
SELECT DISTINCT region_id
FROM employees;
16:56:39
PostgreSQL
-- Total Salary for all employees
SELECT first_name, department, hire_
SUM(salary) OVER(PARTITI
...
16:23:25
PostgreSQL
-- Total Salary for all employees
SELECT first_name, department, hire_
SUM(salary) OVER(ORDER B
...
16:22:35


File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
regions
sales
MS SQL
PostgreSQL
1 -- Retrieve the first names, department and
2 -- number of employees working in that department and region
3 SELECT first_name, department,
4 COUNT(*) OVER(PARTITION BY department) dept_count,
5 region_id,
6 COUNT(*) OVER(PARTITION BY region_id) region_count
7 FROM employees
first_name
department
dept_count
region_id
region_count
Ladonna Automotive 32 2 141
Clementina Automotive 32 1 152
Cybil Automotive 32 1 152
Merlina Automotive 32 1 152
Jessalyn Automotive 32 7 152
Abbott Automotive 32 7 152
Irita Automotive 32 1 152
Berkley Automotive 32 5 127
Vanda Automotive 32 4 137
History
Syntax | History
PostgreSQL
-- Retrieve the first names, departm
-- number of employees working in th
...
17:00:00
PostgreSQL
-- Retrieve the different region ids
SELECT DISTINCT region_id
FROM employees;
16:56:39
PostgreSQL
-- Total Salary for all employees
SELECT first_name, department, hire_
SUM(salary) OVER(PARTITI
...
16:23:25
PostgreSQL


File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
regions
sales
MS SQL
PostgreSQL
1 -- Retrieve the first_name, hire_date, salary
2 -- of all employees ordered by the hire date
3
4 SELECT first_name, hire_date, salary,
5 SUM(salary) OVER(ORDER BY hire_date
6 RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS running_total
7 FROM employees
first_name
hire_date
salary
running_total
Norbie 2003-01-01 82215 189151
Cassandra 2003-01-01 106936 189151
Rora 2003-01-12 153489 342640
Feliks 2003-01-14 55307 397947
Cecilius 2003-01-20 98882 496829
Eugenius 2003-01-26 152118 648947
Fiorenze 2003-02-17 51266 700213
Elnora 2003-02-22 34355 734568
Chelsey 2003-02-24 57309 791877
History
Syntax | History
PostgreSQL
-- Retrieve the first_name, hire_dat
-- of all employees ordered by the h
SELECT f
...
17:34:55
PostgreSQL
SELECT first_name, hire_date, salary
SUM(salary) OVER(ORDER BY hire_date
RANGE BETWEEN UNBOUNDED P
...
17:34:38
PostgreSQL
-- Retrieve the first names, departm
-- number of employees working in th
...
17:00:00


File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
regions
sales
MS SQL
PostgreSQL
1 -- Find the running average
2 SELECT first_name, hire_date, salary,
3 AVG(salary) OVER(ORDER BY hire_date
4 ROWS BETWEEN 1 PRECEDING AND CURRENT ROW) AS running_total
5 FROM employees
6
first_name
hire_date
salary
running_total
Norbie 2003-01-01 82215 82215.000000000000
Cassandra 2003-01-01 106936 94575.500000000000
Rora 2003-01-12 153489 130212.500000000000
Feliks 2003-01-14 55307 104398.000000000000
Cecilius 2003-01-20 98882 77094.500000000000
Eugenius 2003-01-26 152118 125500.000000000000
Fiorenze 2003-02-17 51266 101692.000000000000
Elnora 2003-02-22 34355 42810.500000000000
Chelsey 2003-02-24 57309 45832.000000000000
History
Syntax | History
PostgreSQL
-- Find the running average
SELECT first_name, hire_date, salary
AVG(salary) OVER(ORDER BY hire_dat
...
17:39:54
PostgreSQL
SELECT first_name, hire_date, salary
SUM(salary) OVER(ORDER BY hire_date
RANGE BETWEEN UNBOUNDED P
...
17:39:19
PostgreSQL
-- Add the current row and previous
SELECT first_name, hire_date, salary
SUM(salary) OVER(ORDER
...
17:39:08


File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
regions
sales
MS SQL
PostgreSQL
1 -- GROUPING SETS allow us to use GROUP BY on multiple categoriessales
2 -- with one single query
3
4 SELECT s."Ship_Mode", s."Category", s."Sub-Category", SUM(s."Quantity")
5 FROM "Sales" s
6 GROUP BY GROUPING SETS (s."Ship_Mode", s."Category", s."Sub-Category", ())
Ship_Mode
Category
Sub-Category
sum
NULL NULL NULL 37873
Standard Class NULL NULL 22797
Second Class NULL NULL 7423
Same Day NULL NULL 1960
First Class NULL NULL 5693
NULL Furniture NULL 8028
NULL Office Supplies NULL 22906
NULL Technology NULL 6939
NULL NULL Tables 1241
History
Syntax | History
PostgreSQL
SELECT s."Ship_Mode", s."Category",
FROM "Sales" s
GROUP BY GROU
...
17:51:47
PostgreSQL
SELECT s."Ship_Mode", s.Category, s
FROM "Sales" s
GROUP BY GROUPING SETS
...
Help: Error 42703 column s.category does not exist
17:51:03
PostgreSQL
SELECT s."Ship_Mode", category, sub
FROM "Sales" s
GROUP BY GROUPING SETS (s
...


File
Owner DB
Run
Export
Import
Client
Sign in
SQLite
MariaDB
PostgreSQL
0.17.0 beta
Table
"Customers"
"Sales"
customers
demo
departments
employees
prev
regions
sales
MS SQL
PostgreSQL
1 -- CUBE allows us to use GROUP BY on multiple categories
2 -- with one single query: every combination appears
3
4 SELECT s."Ship_Mode", s."Category", s."Sub-Category", SUM(s."Quantity")
5 FROM "Sales" s
6 GROUP BY CUBE (s."Ship_Mode", s."Category", s."Sub-Category")
7
Ship_Mode
Category
Sub-Category
sum
NULL NULL NULL 37873
Standard Class Furniture Chairs 1372
Second Class Office Supplies Appliances 324
Standard Class Office Supplies Labels 830
Same Day Office Supplies Envelopes 58
First Class Technology Copiers 45
Standard Class Furniture Furnishings 2186
Standard Class Office Supplies Envelopes 548
Standard Class Office Supplies Supplies 387
History
Syntax | History
PostgreSQL
-- GROUPING SETS allow us to use GRO
-- with one single query
SELE
...
17:59:34
PostgreSQL
-- GROUPING SETS allow us to use GR
-- with one single query
SELE
...
Help: Error 42601 syntax error at or near ")"
17:59:29
PostgreSQL
SELECT s."Ship_Mode", s."Category",
FROM "Sales" s
GROUP BY GROU