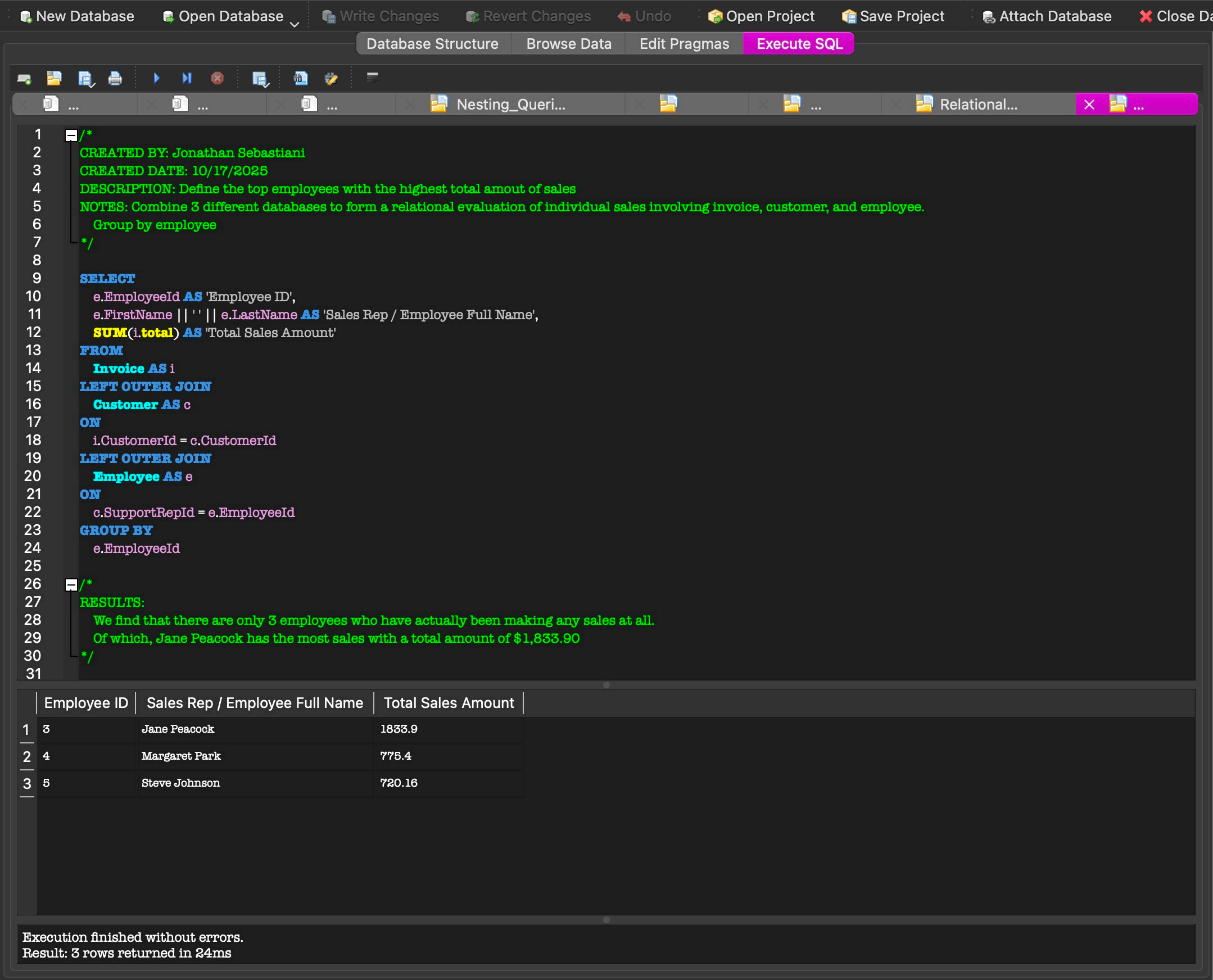
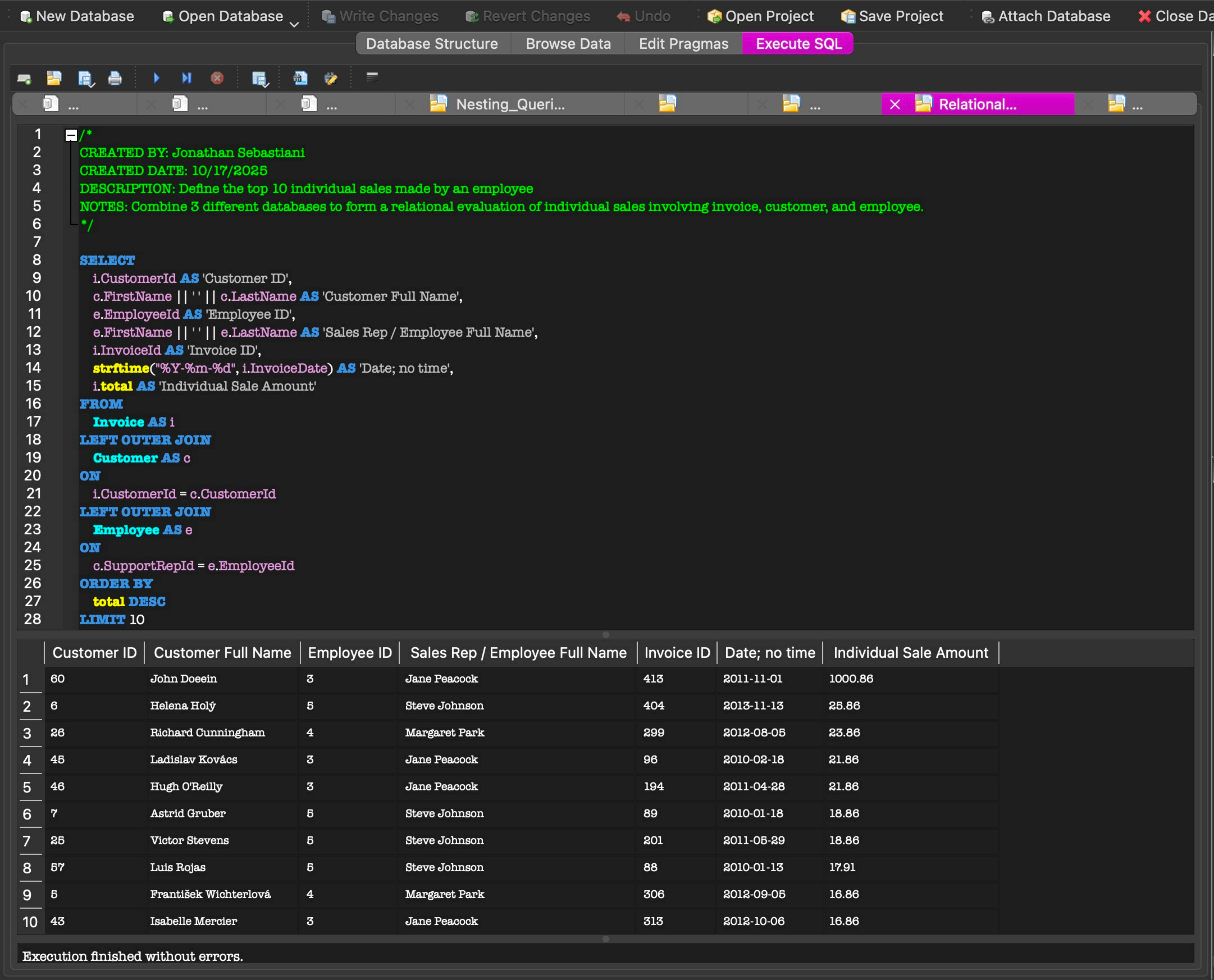


```
/*
CREATED BY: Jonathan Sebastiani
CREATED DATE: 10/17/2025
DESCRIPTION: Define the top employees with the highest total amout of sales
NOTES: Combine 3 different databases to form a relational evaluation of individual sales involving invoice, customer, and employee.
  Group by employee
*/

SELECT
  e.EmployeeId AS 'Employee ID',
  e.FirstName || ' ' || e.LastName AS 'Sales Rep / Employee Full Name',
  SUM(i.total) AS 'Total Sales Amount'
FROM
  Invoice AS i
LEFT OUTER JOIN
  Customer AS c
ON
  i.CustomerId = c.CustomerId
LEFT OUTER JOIN
  Employee AS e
ON
  c.SupportRepId = e.EmployeeId
GROUP BY
  e.EmployeeId

/*
RESULTS:
  We find that there are only 3 employees who have actually been making any sales at all.
  Of which, Jane Peacock has the most sales with a total amount of $1,833.90
*/
```

	Employee ID	Sales Rep / Employee Full Name	Total Sales Amount
1	3	Jane Peacock	1833.9
2	4	Margaret Park	775.4
3	5	Steve Johnson	720.16

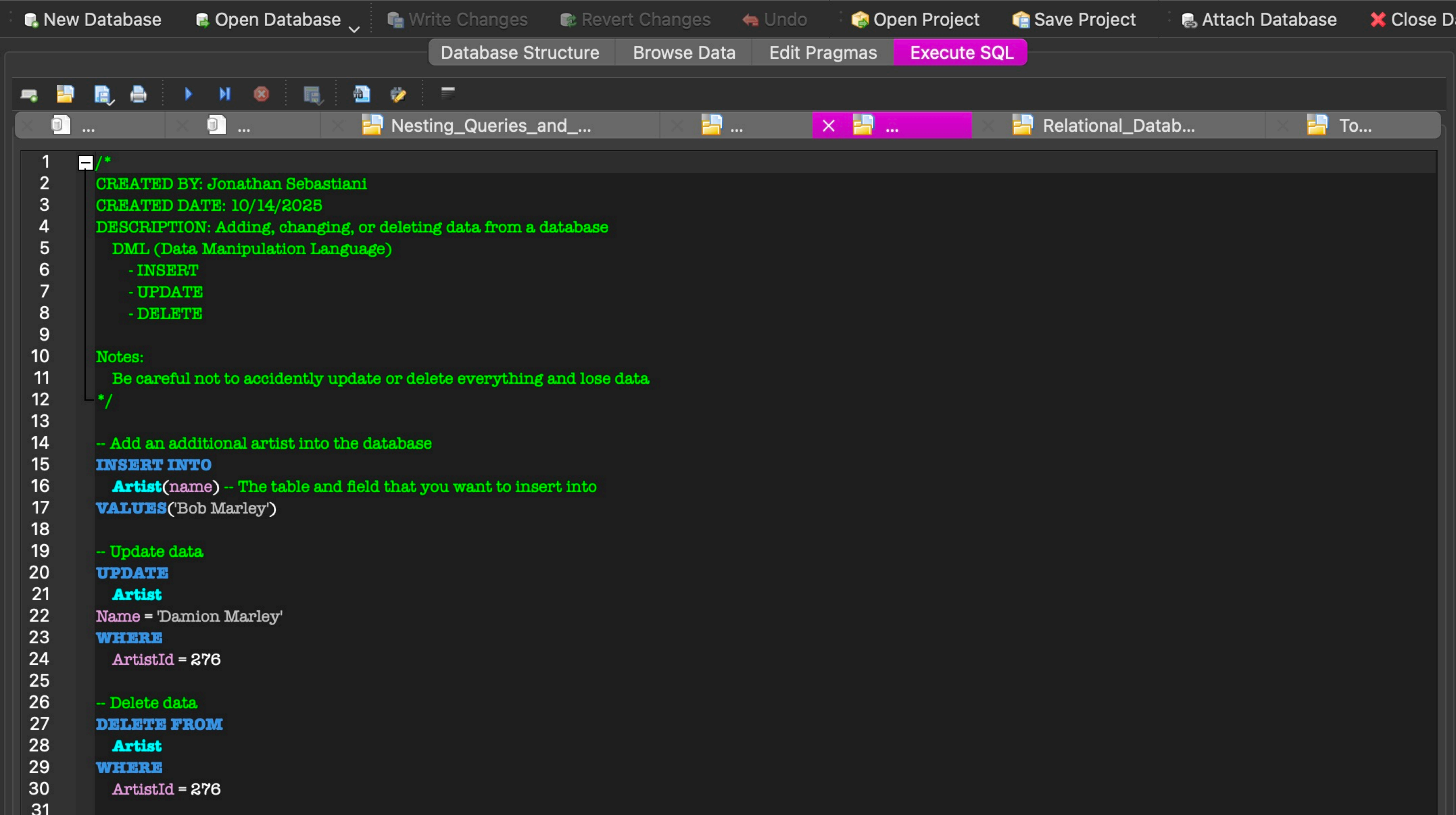
```
Execution finished without errors.
Result: 3 rows returned in 24ms
```



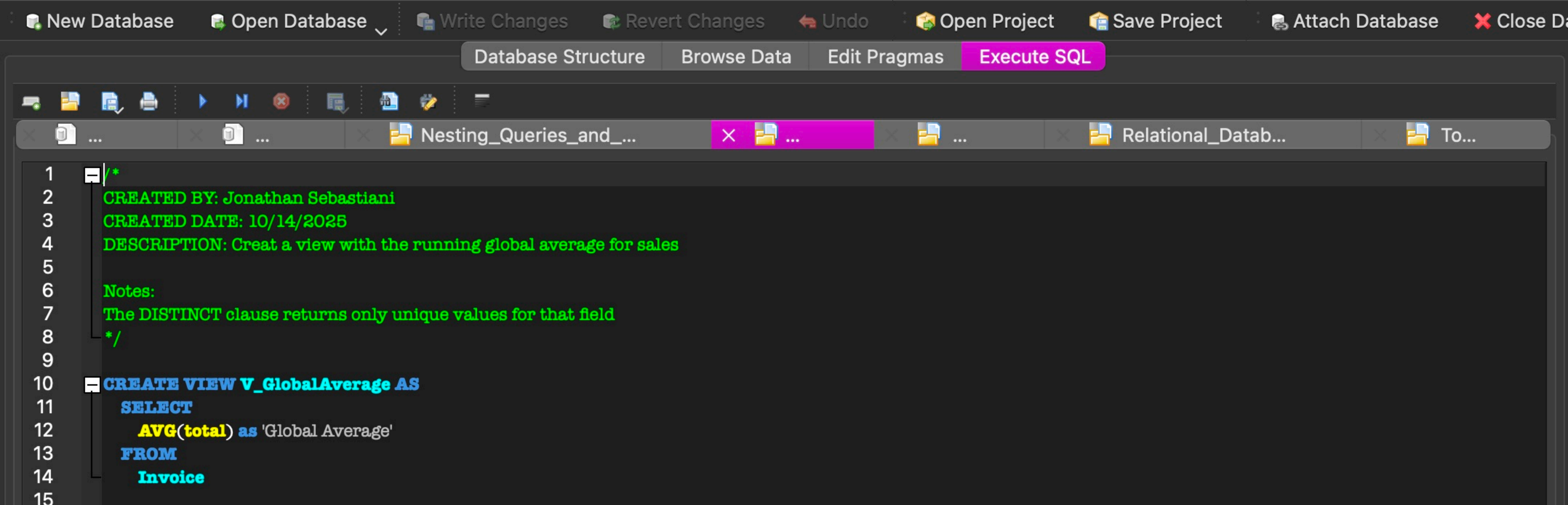
```sql
/*
CREATED BY: Jonathan Sebastiani
CREATED DATE: 10/17/2025
DESCRIPTION: Define the top 10 individual sales made by an employee
NOTES: Combine 3 different databases to form a relational evaluation of individual sales involving invoice, customer, and employee.
*/

SELECT
  i.CustomerId AS 'Customer ID',
  c.FirstName || ' ' || c.LastName AS 'Customer Full Name',
  e.EmployeeId AS 'Employee ID',
  e.FirstName || ' ' || e.LastName AS 'Sales Rep / Employee Full Name',
  i.InvoiceId AS 'Invoice ID',
  strftime("%Y-%m-%d", i.InvoiceDate) AS 'Date; no time',
  i.total AS 'Individual Sale Amount'
FROM
  Invoice AS i
LEFT OUTER JOIN
  Customer AS c
ON
  i.CustomerId = c.CustomerId
LEFT OUTER JOIN
  Employee AS e
ON
  c.SupportRepId = e.EmployeeId
ORDER BY
  total DESC
LIMIT 10
```

	Customer ID	Customer Full Name	Employee ID	Sales Rep / Employee Full Name	Invoice ID	Date; no time	Individual Sale Amount
1	60	John Doeein	3	Jane Peacock	413	2011-11-01	1000.86
2	6	Helena Holý	5	Steve Johnson	404	2013-11-13	25.86
3	26	Richard Cunningham	4	Margaret Park	299	2012-08-05	23.86
4	45	Ladislav Kovács	3	Jane Peacock	96	2010-02-18	21.86
5	46	Hugh O'Reilly	3	Jane Peacock	194	2011-04-28	21.86
6	7	Astrid Gruber	5	Steve Johnson	89	2010-01-18	18.86
7	25	Victor Stevens	5	Steve Johnson	201	2011-05-29	18.86
8	57	Luis Rojas	5	Steve Johnson	88	2010-01-13	17.91
9	5	František Wichterlová	4	Margaret Park	306	2012-09-05	16.86
10	43	Isabelle Mercier	3	Jane Peacock	313	2012-10-06	16.86

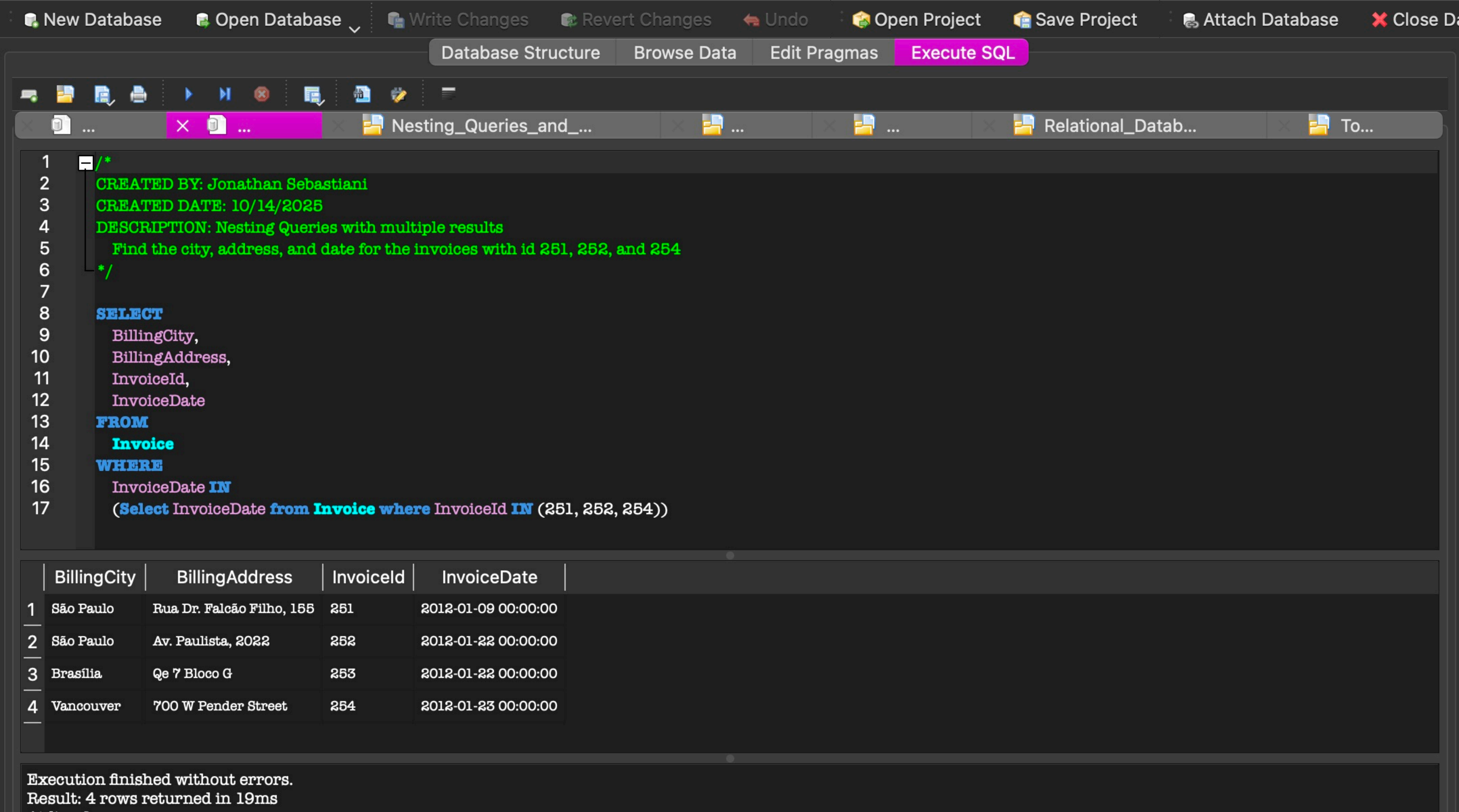
Execution finished without errors.



```sql
/*
CREATED BY: Jonathan Sebastiani
CREATED DATE: 10/14/2025
DESCRIPTION: Adding, changing, or deleting data from a database
  DML (Data Manipulation Language)
    - INSERT
    - UPDATE
    - DELETE

Notes:
  Be careful not to accidently update or delete everything and lose data
*/

-- Add an additional artist into the database
INSERT INTO
  Artist(name) -- The table and field that you want to insert into
VALUES('Bob Marley')

-- Update data
UPDATE
  Artist
Name = 'Damion Marley'
WHERE
  ArtistId = 276

-- Delete data
DELETE FROM
  Artist
WHERE
  ArtistId = 276
```



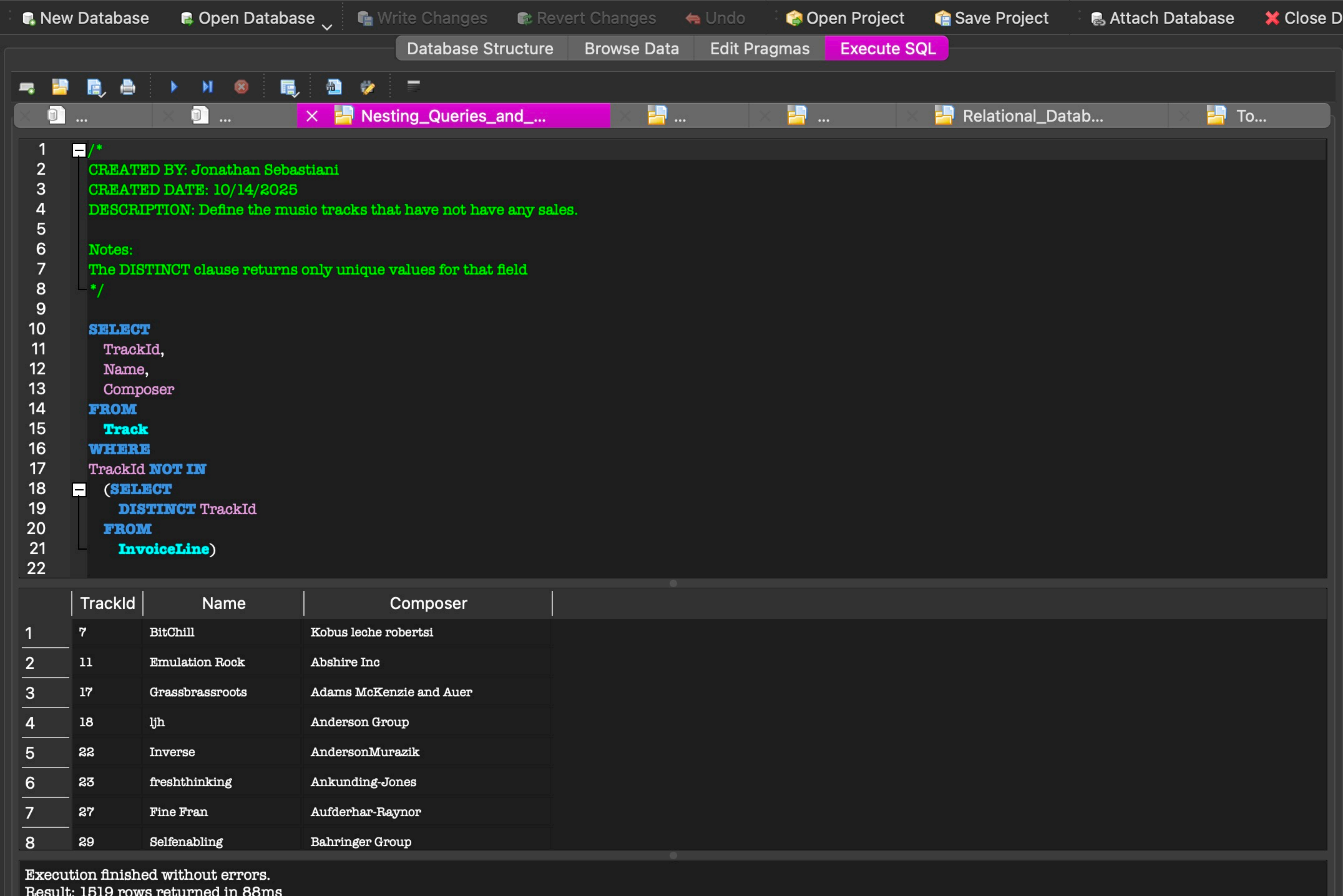
```
/*
CREATED BY: Jonathan Sebastiani
CREATED DATE: 10/14/2025
DESCRIPTION: Creat a view with the running global average for sales

Notes:
The DISTINCT clause returns only unique values for that field
*/

CREATE VIEW V_GlobalAverage AS
  SELECT
    AVG(total) as 'Global Average'
  FROM
    Invoice
```



```
/*
CREATED BY: Jonathan Sebastiani
CREATED DATE: 10/14/2025
DESCRIPTION: Nesting Queries with multiple results
  Find the city, address, and date for the invoices with id 251, 252, and 254
 */

SELECT
  BillingCity,
  BillingAddress,
  InvoiceId,
  InvoiceDate
FROM
  Invoice
WHERE
  InvoiceDate IN
  (Select InvoiceDate from Invoice where InvoiceId IN (251, 252, 254))
```

	BillingCity	BillingAddress	InvoiceId	InvoiceDate
1	São Paulo	Rua Dr. Falcão Filho, 155	251	2012-01-09 00:00:00
2	São Paulo	Av. Paulista, 2022	252	2012-01-22 00:00:00
3	Brasília	Qe 7 Bloco G	253	2012-01-22 00:00:00
4	Vancouver	700 W Pender Street	254	2012-01-23 00:00:00

```
Execution finished without errors.
Result: 4 rows returned in 19ms
```



```
/*
CREATED BY: Jonathan Sebastiani
CREATED DATE: 10/14/2025
DESCRIPTION: Define the music tracks that have not have any sales.

Notes:
The DISTINCT clause returns only unique values for that field
*/

SELECT
  TrackId,
  Name,
  Composer
FROM
  Track
WHERE
TrackId NOT IN
  (SELECT
    DISTINCT TrackId
  FROM
    InvoiceLine)

```

	TrackId	Name	Composer
1	7	BitChill	Kobus leche robertsi
2	11	Emulation Rock	Abshire Inc
3	17	Grassbrassroots	Adams McKenzie and Auer
4	18	ljh	Anderson Group
5	22	Inverse	AndersonMurazik
6	23	freshthinking	Ankunding-Jones
7	27	Fine Fran	Aufderhar-Raynor
8	29	Selfenabling	Bahringer Group

```
Execution finished without errors.
Result: 1519 rows returned in 88ms
```