SQL Assignment

Sumit Thakur Barahi
Course: SQL Fundamentals

May 24, 2025

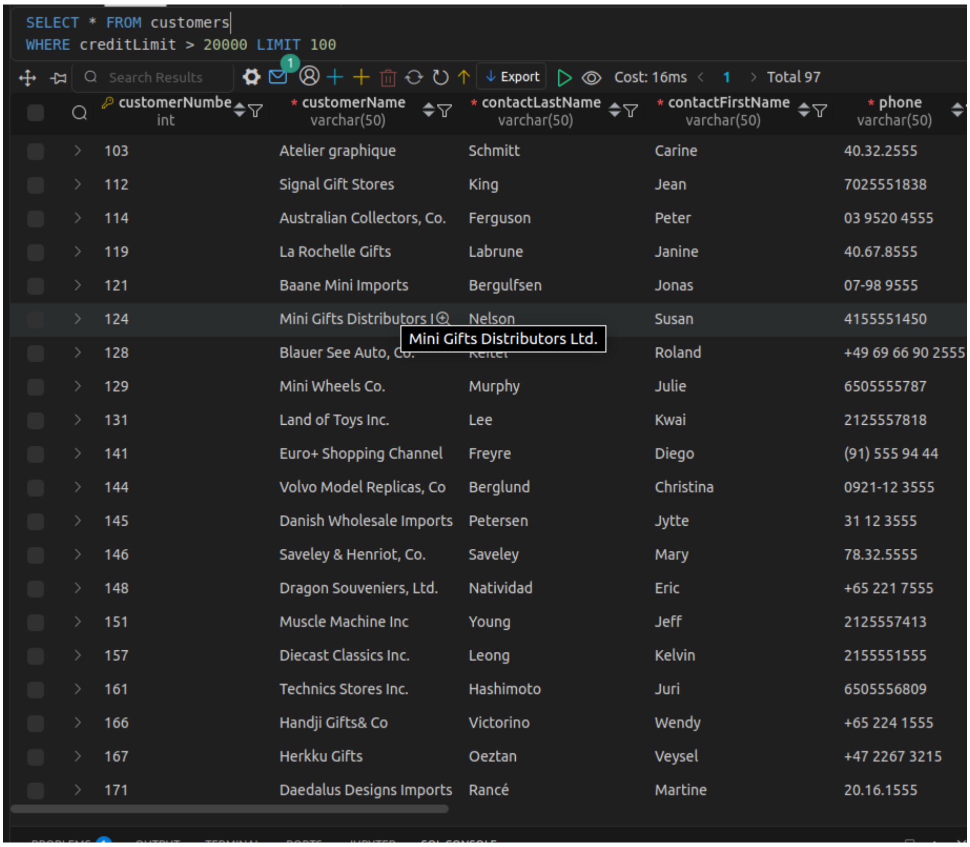
Contents

1 Task : Show all the customers whose creditLimit is greater than 20000

SQL Query

```
SELECT * FROM customers
WHERE creditLimit > 20000;
```

Output Screenshot



```
SELECT * FROM customers
WHERE creditLimit > 20000 LIMIT 100
```

customerNumbe int	customerName varchar(50)	contactLastName varchar(50)	contactFirstName varchar(50)	phone varchar(50)
103	Atelier graphique	Schmitt	Carine	40.32.2555
112	Signal Gift Stores	King	Jean	7025551838
114	Australian Collectors, Co.	Ferguson	Peter	03 9520 4555
119	La Rochelle Gifts	Labrune	Janine	40.67.8555
121	Baane Mini Imports	Bergulfsen	Jonas	07-98 9555
124	Mini Gifts Distributors l	Nelson	Susan	4155551450
128	Blauer See Auto, Co.	Keitel	Roland	+49 69 66 90 2555
129	Mini Wheels Co.	Murphy	Julie	6505555787
131	Land of Toys Inc.	Lee	Kwai	2125557818
141	Euro+ Shopping Channel	Freyre	Diego	(91) 555 94 44
144	Volvo Model Replicas, Co	Berglund	Christina	0921-12 3555
145	Danish Wholesale Imports	Petersen	Jytte	31 12 3555
146	Saveley & Henriot, Co.	Saveley	Mary	78.32.5555
148	Dragon Souveniers, Ltd.	Natividad	Eric	+65 221 7555
151	Muscle Machine Inc	Young	Jeff	2125557413
157	Diecast Classics Inc.	Leong	Kelvin	2155551555
161	Technics Stores Inc.	Hashimoto	Juri	6505556809
166	Handji Gifts& Co	Victorino	Wendy	+65 224 1555
167	Herkku Gifts	Oeztan	Veysel	+47 2267 3215
171	Daedalus Designs Imports	Rancé	Martine	20.16.1555

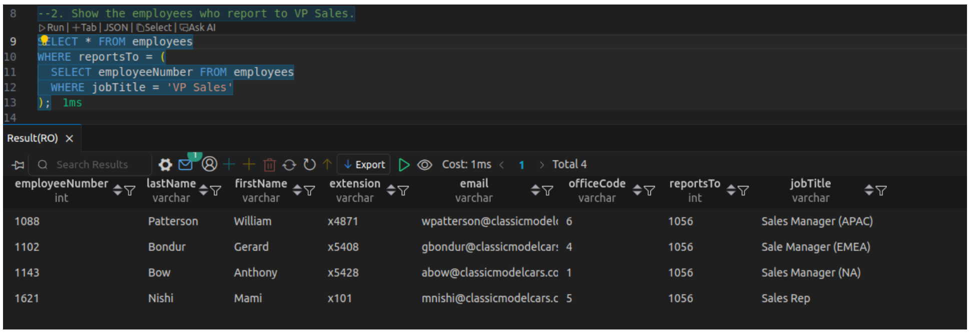
Mini Gifts Distributors Ltd.

2 Task : Show the employees who report to VP Sales.

SQL Query

```
SELECT * FROM employees
WHERE reportsTo = (
  SELECT employeeNumber FROM employees
  WHERE jobTitle = 'VP Sales'
);
```

Output Screenshot



```
--2. Show the employees who report to VP Sales.
SELECT * FROM employees
WHERE reportsTo = (
  SELECT employeeNumber FROM employees
  WHERE jobTitle = 'VP Sales'
);  1ms
```

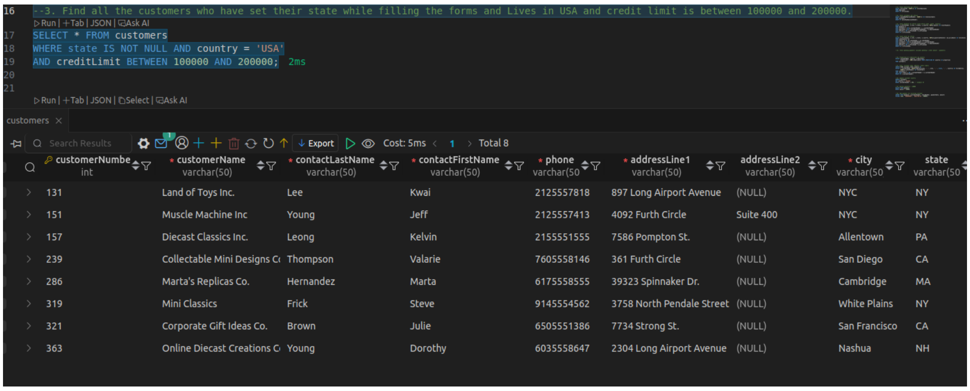
employeeNumber	lastName	firstName	extension	email	officeCode	reportsTo	jobTitle
int	varchar	varchar	varchar	varchar	varchar	int	varchar
1088	Patterson	William	x4871	wpatterson@classicmodel	6	1056	Sales Manager (APAC)
1102	Bondur	Gerard	x5408	gbondur@classicmodelcar:	4	1056	Sale Manager (EMEA)
1143	Bow	Anthony	x5428	abow@classicmodelcars.co	1	1056	Sales Manager (NA)
1621	Nishi	Mami	x101	mnishi@classicmodelcars.c	5	1056	Sales Rep

3 Task : Find all the customers who have set their state while filling the forms and Lives in USA and credit limit is between 100000 and 200000.

SQL Query

```
SELECT * FROM customers
WHERE state IS NOT NULL AND country = 'USA'
AND creditLimit BETWEEN 100000 AND 200000;
```

Output Screenshot



```
--3. Find all the customers who have set their state while filling the forms and Lives in USA and credit limit is between 100000 and 200000.
SELECT * FROM customers
WHERE state IS NOT NULL AND country = 'USA'
AND creditLimit BETWEEN 100000 AND 200000;  2ms
```

▷Run | ＋Tab | JSON | Ask AI
▷Run | ＋Tab | JSON | ⎘Select | Ask AI
customers ×
Search Results
Export
Cost: 5ms
1
Total 8

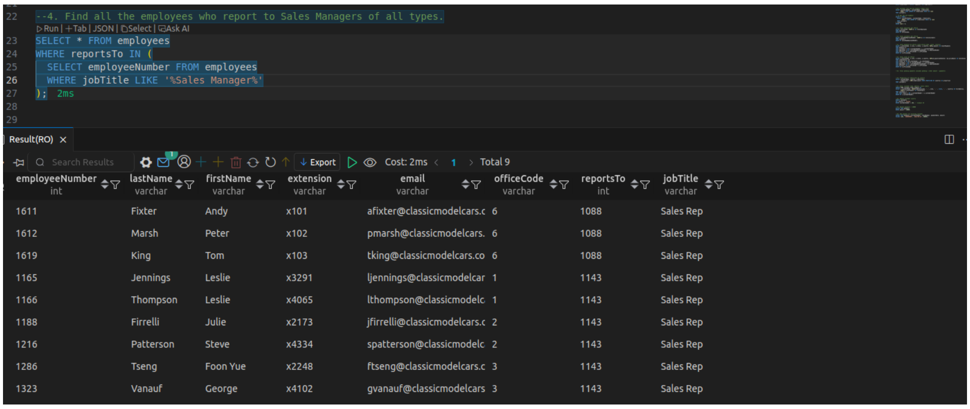
customerNumbe	customerName	contactLastName	contactFirstName	phone	addressLine1	addressLine2	city	state
int	varchar(50)	varchar(50)	varchar(50)	varchar(50	varchar(50)	varchar(50)	varchar(50	varchar(50
131	Land of Toys Inc.	Lee	Kwai	2125557818	897 Long Airport Avenue	(NULL)	NYC	NY
151	Muscle Machine Inc	Young	Jeff	2125557413	4092 Furth Circle	Suite 400	NYC	NY
157	Diecast Classics Inc.	Leong	Kelvin	2155551555	7586 Pompton St.	(NULL)	Allentown	PA
239	Collectable Mini Designs Co	Thompson	Valarie	7605558146	361 Furth Circle	(NULL)	San Diego	CA
286	Marta's Replicas Co.	Hernandez	Marta	6175558555	39323 Spinnaker Dr.	(NULL)	Cambridge	MA
319	Mini Classics	Frick	Steve	9145554562	3758 North Pendale Street	(NULL)	White Plains	NY
321	Corporate Gift Ideas Co.	Brown	Julie	6505551386	7734 Strong St.	(NULL)	San Francisco	CA
363	Online Diecast Creations Co	Young	Dorothy	6035558647	2304 Long Airport Avenue	(NULL)	Nashua	NH

4 Task : Find all the employees who report to Sales Managers of all types

SQL Query

```
SELECT * FROM employees
WHERE reportsTo IN (
  SELECT employeeNumber FROM employees
  WHERE jobTitle LIKE '%Sales Manager%'
);
```

Output Screenshot



```
--4. Find all the employees who report to Sales Managers of all types.
SELECT * FROM employees
WHERE reportsTo IN (
  SELECT employeeNumber FROM employees
  WHERE jobTitle LIKE '%Sales Manager%'
);  2ms
```

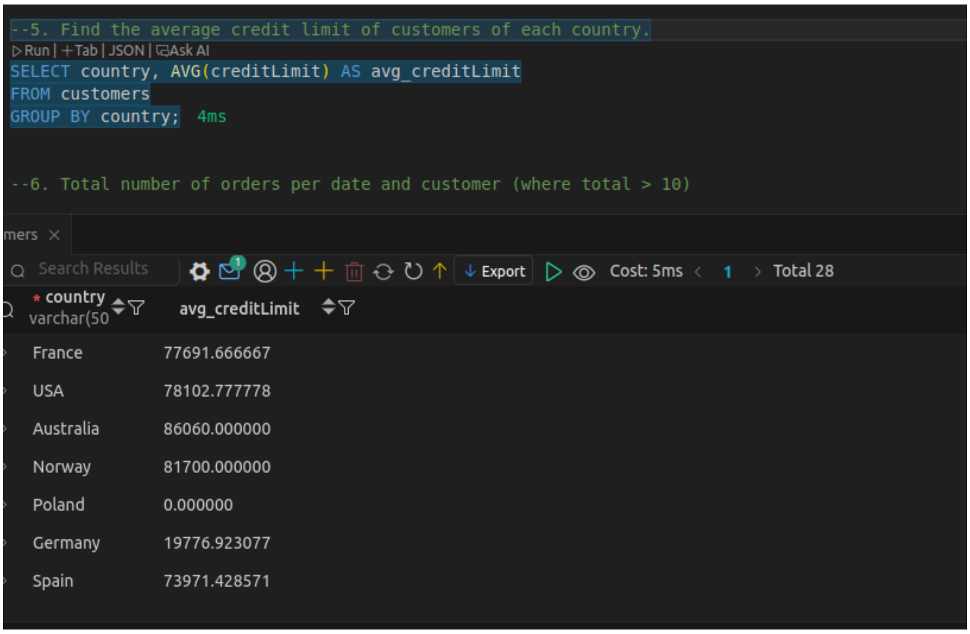
employeeNumber int	lastName varchar	firstName varchar	extension varchar	email varchar	officeCode varchar	reportsTo int	jobTitle varchar
1611	Fixter	Andy	x101	afixter@classicmodelcars.c	6	1088	Sales Rep
1612	Marsh	Peter	x102	pmarsh@classicmodelcars.	6	1088	Sales Rep
1619	King	Tom	x103	tking@classicmodelcars.co	6	1088	Sales Rep
1165	Jennings	Leslie	x3291	ljennings@classicmodelcar	1	1143	Sales Rep
1166	Thompson	Leslie	x4065	lthompson@classicmodelc	1	1143	Sales Rep
1188	Firrelli	Julie	x2173	jfirrelli@classicmodelcars.c	2	1143	Sales Rep
1216	Patterson	Steve	x4334	spatterson@classicmodelc	2	1143	Sales Rep
1286	Tseng	Foon Yue	x2248	ftseng@classicmodelcars.c	3	1143	Sales Rep
1323	Vanauf	George	x4102	gvanauf@classicmodelcars	3	1143	Sales Rep

5 Task : Find the average credit limit of customers of each country.

SQL Query

```
SELECT country, AVG(creditLimit) AS avg_creditLimit
FROM customers
GROUP BY country;
```

Output Screenshot



```
--5. Find the average credit limit of customers of each country.
SELECT country, AVG(creditLimit) AS avg_creditLimit
FROM customers
GROUP BY country;  4ms


--6. Total number of orders per date and customer (where total > 10)
```

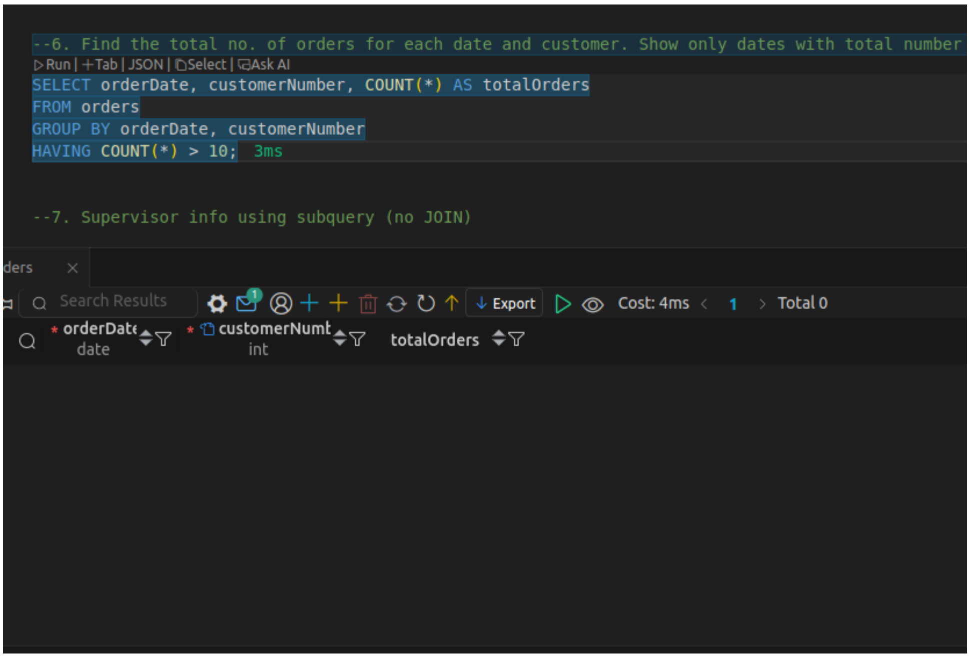
* country varchar(50	avg_creditLimit
France	77691.666667
USA	78102.777778
Australia	86060.000000
Norway	81700.000000
Poland	0.000000
Germany	19776.923077
Spain	73971.428571

6 Task : Find the total no. of orders for each date and customer. Show only dates with total number of orders greater than 10 for date and customer.

SQL Query

```
SELECT orderDate, customerNumber, COUNT(*) AS totalOrders
FROM orders
GROUP BY orderDate, customerNumber
HAVING COUNT(*) > 10;
```

Output Screenshot



```
--6. Find the total no. of orders for each date and customer. Show only dates with total number
SELECT orderDate, customerNumber, COUNT(*) AS totalOrders
FROM orders
GROUP BY orderDate, customerNumber
HAVING COUNT(*) > 10;  3ms


--7. Supervisor info using subquery (no JOIN)
```

orderDate date	customerNumber int	totalOrders

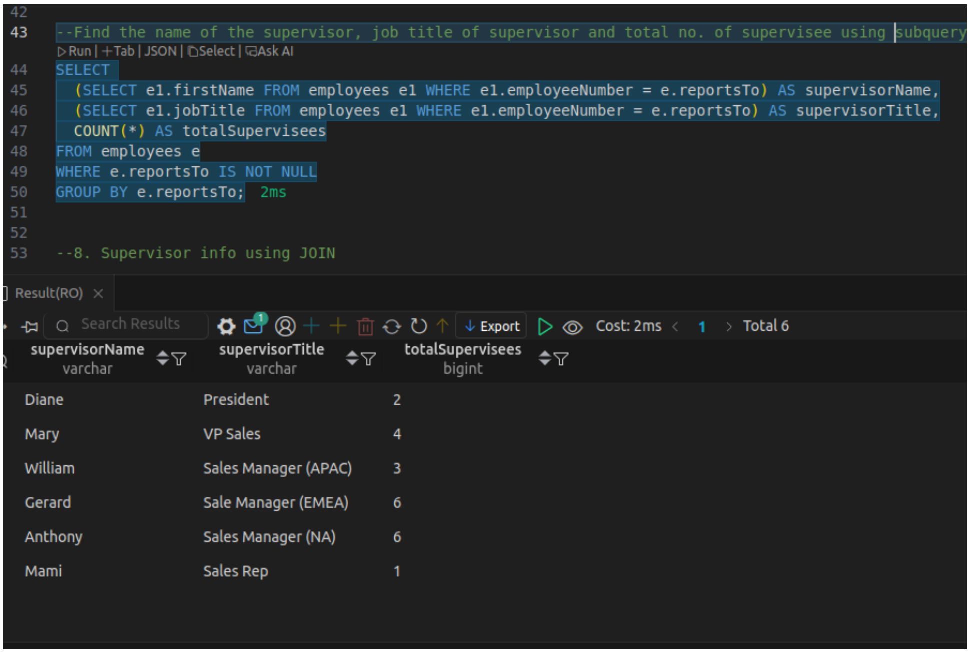
7 Task : Find the name of the supervisor, job title of supervisor and total no. of supervisee using subquery. (With out using Join operation)

SQL Query

```
SELECT
  (SELECT e1.firstName FROM employees e1 WHERE e1.employeeNumber
     = e.reportsTo) AS supervisorName,
  (SELECT e1.jobTitle FROM employees e1 WHERE e1.employeeNumber =
     e.reportsTo) AS supervisorTitle,
  COUNT(*) AS totalSupervisees
FROM employees e
WHERE e.reportsTo IS NOT NULL
```

```sql
GROUP BY e.reportsTo;
```

Output Screenshot



```sql
--Find the name of the supervisor, job title of supervisor and total no. of supervisee using subquery
SELECT
  (SELECT e1.firstName FROM employees e1 WHERE e1.employeeNumber = e.reportsTo) AS supervisorName,
  (SELECT e1.jobTitle FROM employees e1 WHERE e1.employeeNumber = e.reportsTo) AS supervisorTitle,
  COUNT(*) AS totalSupervisees
FROM employees e
WHERE e.reportsTo IS NOT NULL
GROUP BY e.reportsTo;  2ms


--8. Supervisor info using JOIN
```

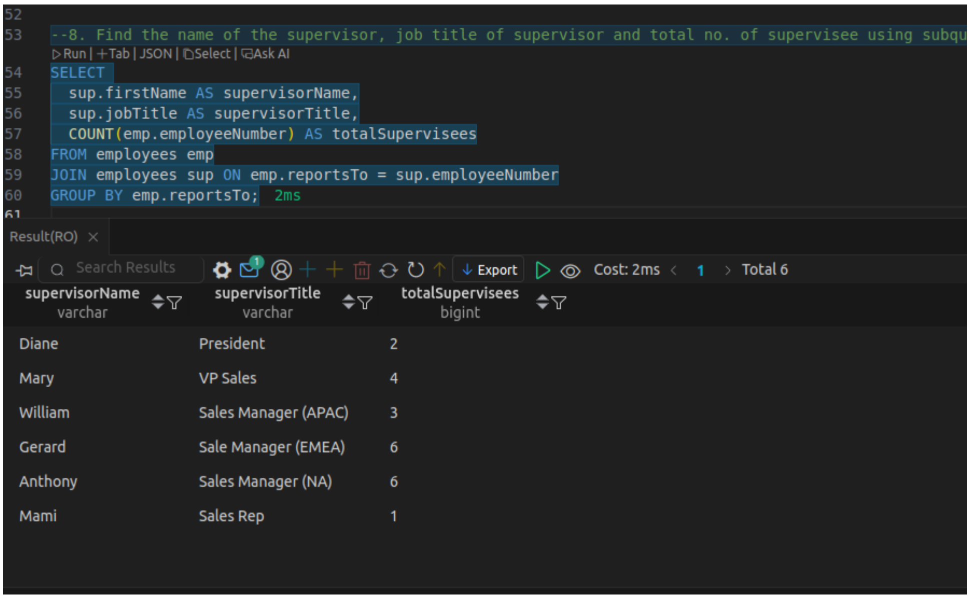
supervisorName varchar	supervisorTitle varchar	totalSupervisees bigint
Diane	President	2
Mary	VP Sales	4
William	Sales Manager (APAC)	3
Gerard	Sale Manager (EMEA)	6
Anthony	Sales Manager (NA)	6
Mami	Sales Rep	1

8 Task : Find the name of the supervisor, job title of supervisor and total no. of supervisee using subquery. (With using Join operation)

SQL Query

```sql
SELECT
  sup.firstName AS supervisorName,
  sup.jobTitle AS supervisorTitle,
  COUNT(emp.employeeNumber) AS totalSupervisees
FROM employees emp
JOIN employees sup ON emp.reportsTo = sup.employeeNumber
GROUP BY emp.reportsTo;
```

Output Screenshot



```
--8. Find the name of the supervisor, job title of supervisor and total no. of supervisee using subqu
SELECT
  sup.firstName AS supervisorName,
  sup.jobTitle AS supervisorTitle,
  COUNT(emp.employeeNumber) AS totalSupervisees
FROM employees emp
JOIN employees sup ON emp.reportsTo = sup.employeeNumber
GROUP BY emp.reportsTo;  2ms
```

Result(RO) Cost: 2ms Total 6

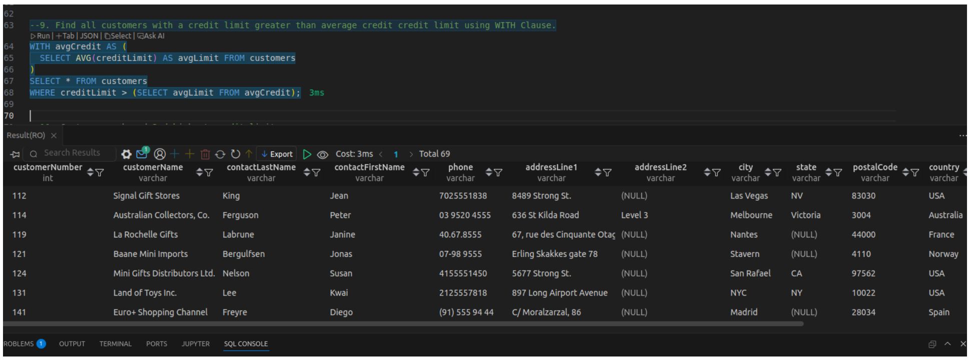
supervisorName	supervisorTitle	totalSupervisees
varchar	varchar	bigint
Diane	President	2
Mary	VP Sales	4
William	Sales Manager (APAC)	3
Gerard	Sale Manager (EMEA)	6
Anthony	Sales Manager (NA)	6
Mami	Sales Rep	1

9 Task :Find all customers with a credit limit greater than average credit credit limit using WITH Clause.

SQL Query

```
WITH avgCredit AS (
  SELECT AVG(creditLimit) AS avgLimit FROM customers
)
SELECT * FROM customers
WHERE creditLimit > (SELECT avgLimit FROM avgCredit);
```

Output Screenshot



```
--9. Find all customers with a credit limit greater than average credit credit limit using WITH Clause.
WITH avgCredit AS (
  SELECT AVG(creditLimit) AS avgLimit FROM customers
)
SELECT * FROM customers
WHERE creditLimit > (SELECT avgLimit FROM avgCredit);  3ms
```

Result(RO) Cost: 3ms Total 69

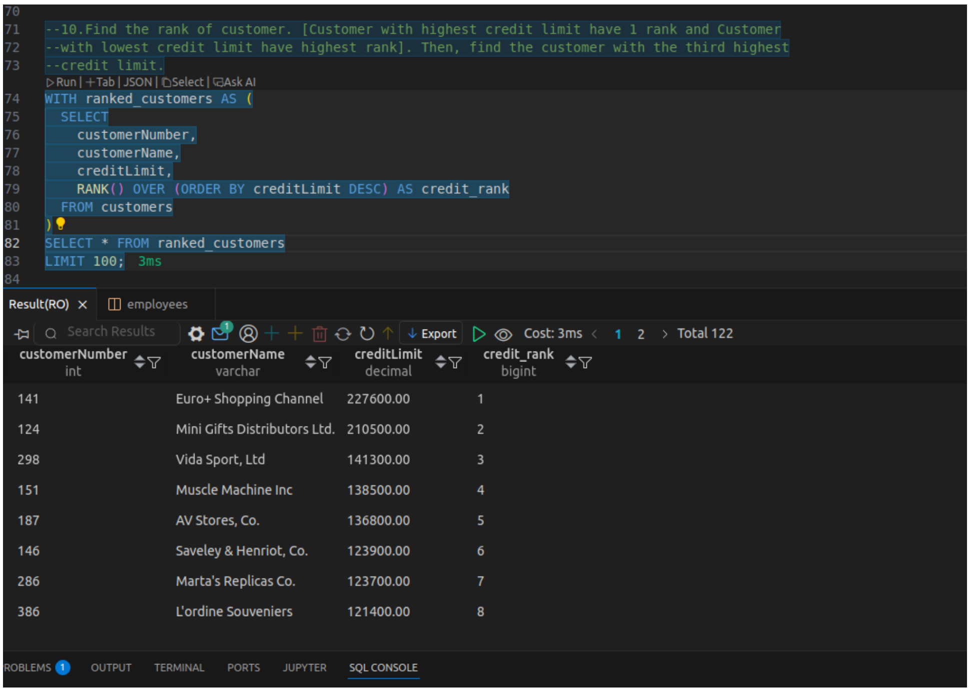
customerNumber	customerName	contactLastName	contactFirstName	phone	addressLine1	addressLine2	city	state	postalCode	country
int	varchar	varchar	varchar	varchar	varchar	varchar	varchar	varchar	varchar	varchar
112	Signal Gift Stores	King	Jean	7025551838	8489 Strong St.	(NULL)	Las Vegas	NV	83030	USA
114	Australian Collectors, Co.	Ferguson	Peter	03 9520 4555	636 St Kilda Road	Level 3	Melbourne	Victoria	3004	Australia
119	La Rochelle Gifts	Labrune	Janine	40.67.8555	67, rue des Cinquante Otag	(NULL)	Nantes	(NULL)	44000	France
121	Baane Mini Imports	Bergulfsen	Jonas	07-98 9555	Erling Skakkes gate 78	(NULL)	Stavern	(NULL)	4110	Norway
124	Mini Gifts Distributors Ltd.	Nelson	Susan	4155551450	5677 Strong St.	(NULL)	San Rafael	CA	97562	USA
131	Land of Toys Inc.	Lee	Kwai	2125557818	897 Long Airport Avenue	(NULL)	NYC	NY	10022	USA
141	Euro+ Shopping Channel	Freyre	Diego	(91) 555 94 44	C/ Moralzarzal, 86	(NULL)	Madrid	(NULL)	28034	Spain

10 Task :Find the rank of customer. [Customer with highest credit limit have 1 rank and Customer with lowest credit limit have highest rank]. Then, find the customer with the third highest credit limit.

SQL Query

```
WITH ranked_customers AS (
  SELECT
    customerNumber,
    customerName,
    creditLimit,
    RANK() OVER (ORDER BY creditLimit DESC) AS credit_rank
  FROM customers
)
SELECT * FROM ranked_customers
LIMIT 100;
```

Output Screenshot



```
--10.Find the rank of customer. [Customer with highest credit limit have 1 rank and Customer
--with lowest credit limit have highest rank]. Then, find the customer with the third highest
--credit limit.
WITH ranked_customers AS (
  SELECT
    customerNumber,
    customerName,
    creditLimit,
    RANK() OVER (ORDER BY creditLimit DESC) AS credit_rank
  FROM customers
)
SELECT * FROM ranked_customers
LIMIT 100;  3ms
```

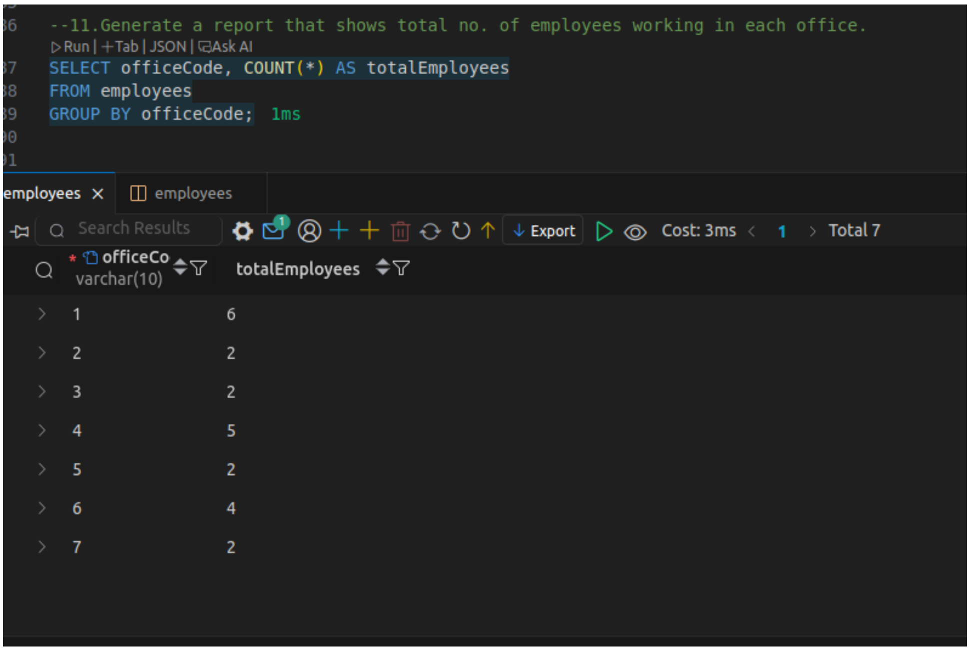
customerNumber int	customerName varchar	creditLimit decimal	credit_rank bigint
141	Euro+ Shopping Channel	227600.00	1
124	Mini Gifts Distributors Ltd.	210500.00	2
298	Vida Sport, Ltd	141300.00	3
151	Muscle Machine Inc	138500.00	4
187	AV Stores, Co.	136800.00	5
146	Saveley & Henriot, Co.	123900.00	6
286	Marta's Replicas Co.	123700.00	7
386	L'ordine Souveniers	121400.00	8

11 Task :Generate a report that shows total no. of employees working in each office.

SQL Query

```sql
SELECT officeCode, COUNT(*) AS totalEmployees
FROM employees
GROUP BY officeCode;
```

Output Screenshot



```sql
--11.Generate a report that shows total no. of employees working in each office.
SELECT officeCode, COUNT(*) AS totalEmployees
FROM employees
GROUP BY officeCode;  1ms
```

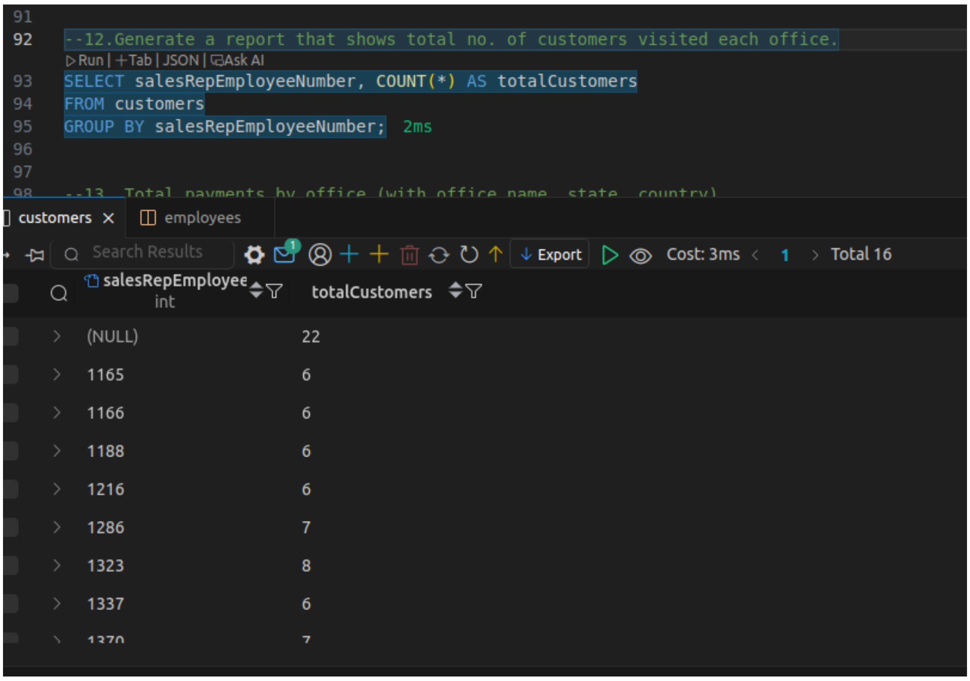
officeCo varchar(10)	totalEmployees
1	6
2	2
3	2
4	5
5	2
6	4
7	2

12 Task :Generate a report that shows total no. of customers visited each office.

SQL Query

```sql
SELECT salesRepEmployeeNumber, COUNT(*) AS totalCustomers
FROM customers
GROUP BY salesRepEmployeeNumber;
```

Output Screenshot



```
--12.Generate a report that shows total no. of customers visited each office.
SELECT salesRepEmployeeNumber, COUNT(*) AS totalCustomers
FROM customers
GROUP BY salesRepEmployeeNumber;  2ms

--13 Total payments by office (with office name, state, country)
```

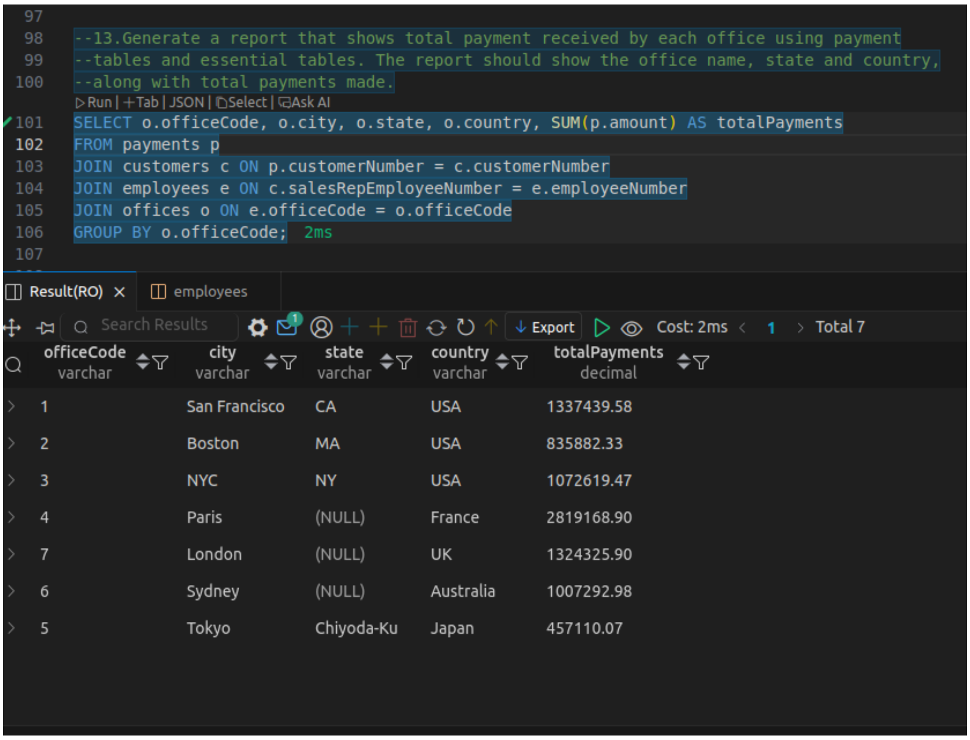
salesRepEmployee int	totalCustomers
(NULL)	22
1165	6
1166	6
1188	6
1216	6
1286	7
1323	8
1337	6
1370	7

13 Task :Generate a report that shows total payment received by each office using payment tables and essential tables. The report should show the office name, state and country, along with total payments made.

SQL Query

```sql
SELECT o.officeCode, o.city, o.state, o.country, SUM(p.amount) AS
    totalPayments
FROM payments p
JOIN customers c ON p.customerNumber = c.customerNumber
JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber
JOIN offices o ON e.officeCode = o.officeCode
GROUP BY o.officeCode;
```

Output Screenshot



```
--13.Generate a report that shows total payment received by each office using payment
--tables and essential tables. The report should show the office name, state and country,
--along with total payments made.
SELECT o.officeCode, o.city, o.state, o.country, SUM(p.amount) AS totalPayments
FROM payments p
JOIN customers c ON p.customerNumber = c.customerNumber
JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber
JOIN offices o ON e.officeCode = o.officeCode
GROUP BY o.officeCode;  2ms
```

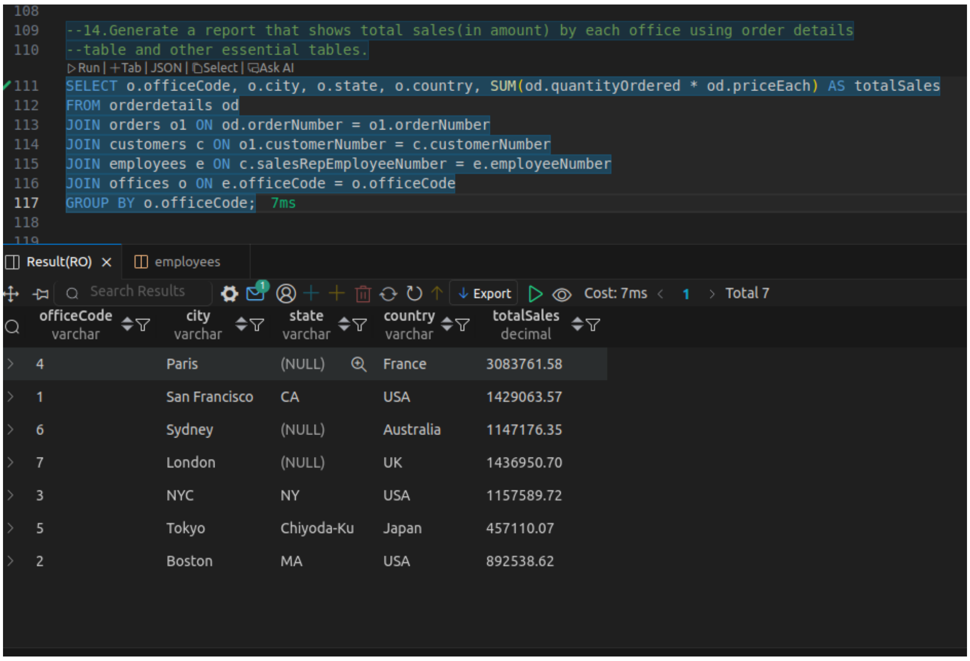
officeCode varchar	city varchar	state varchar	country varchar	totalPayments decimal
1	San Francisco	CA	USA	1337439.58
2	Boston	MA	USA	835882.33
3	NYC	NY	USA	1072619.47
4	Paris	(NULL)	France	2819168.90
7	London	(NULL)	UK	1324325.90
6	Sydney	(NULL)	Australia	1007292.98
5	Tokyo	Chiyoda-Ku	Japan	457110.07

14 Task :Generate a report that shows total sales(in amount) by each office using order details table and other essential tables

SQL Query

```
SELECT o.officeCode, o.city, o.state, o.country, SUM(od.
   quantityOrdered * od.priceEach) AS totalSales
FROM orderdetails od
JOIN orders o1 ON od.orderNumber = o1.orderNumber
JOIN customers c ON o1.customerNumber = c.customerNumber
JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber
JOIN offices o ON e.officeCode = o.officeCode
GROUP BY o.officeCode;
```

Output Screenshot



```
--14.Generate a report that shows total sales(in amount) by each office using order details
--table and other essential tables.
SELECT o.officeCode, o.city, o.state, o.country, SUM(od.quantityOrdered * od.priceEach) AS totalSales
FROM orderdetails od
JOIN orders o1 ON od.orderNumber = o1.orderNumber
JOIN customers c ON o1.customerNumber = c.customerNumber
JOIN employees e ON c.salesRepEmployeeNumber = e.employeeNumber
JOIN offices o ON e.officeCode = o.officeCode
GROUP BY o.officeCode;
```

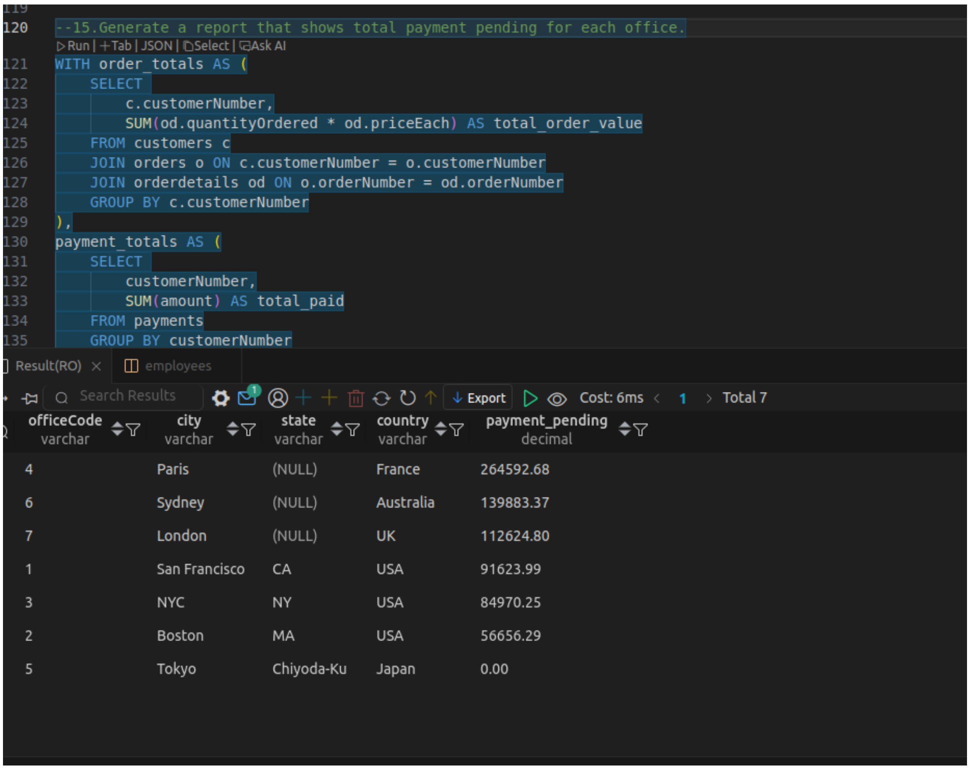
officeCode varchar	city varchar	state varchar	country varchar	totalSales decimal
4	Paris	(NULL)	France	3083761.58
1	San Francisco	CA	USA	1429063.57
6	Sydney	(NULL)	Australia	1147176.35
7	London	(NULL)	UK	1436950.70
3	NYC	NY	USA	1157589.72
5	Tokyo	Chiyoda-Ku	Japan	457110.07
2	Boston	MA	USA	892538.62

15 Task :Generate a report that shows total payment pending for each office.

SQL Query

```
WITH order_totals AS (
    SELECT
        c.customerNumber,
        SUM(od.quantityOrdered * od.priceEach) AS
           total_order_value
    FROM customers c
    JOIN orders o ON c.customerNumber = o.customerNumber
    JOIN orderdetails od ON o.orderNumber = od.orderNumber
    GROUP BY c.customerNumber
),
payment_totals AS (
    SELECT
        customerNumber,
        SUM(amount) AS total_paid
    FROM payments
    GROUP BY customerNumber
),
customer_balances AS (
    SELECT
        c.customerNumber,
        c.salesRepEmployeeNumber,
        COALESCE(ot.total_order_value, 0) - COALESCE(pt.
           total_paid, 0) AS payment_pending
    FROM customers c
    LEFT JOIN order_totals ot ON c.customerNumber = ot.
       customerNumber
    LEFT JOIN payment_totals pt ON c.customerNumber = pt.
       customerNumber
)
SELECT
    o.officeCode,
    o.city,
    o.state,
    o.country,
    SUM(cb.payment_pending) AS payment_pending
FROM offices o
JOIN employees e ON o.officeCode = e.officeCode
JOIN customer_balances cb ON e.employeeNumber = cb.
   salesRepEmployeeNumber
GROUP BY o.officeCode, o.city, o.state, o.country
ORDER BY payment_pending DESC;
```

Output Screenshot



```
--15.Generate a report that shows total payment pending for each office.
WITH order_totals AS (
    SELECT
        c.customerNumber,
        SUM(od.quantityOrdered * od.priceEach) AS total_order_value
    FROM customers c
    JOIN orders o ON c.customerNumber = o.customerNumber
    JOIN orderdetails od ON o.orderNumber = od.orderNumber
    GROUP BY c.customerNumber
),
payment_totals AS (
    SELECT
        customerNumber,
        SUM(amount) AS total_paid
    FROM payments
    GROUP BY customerNumber
```

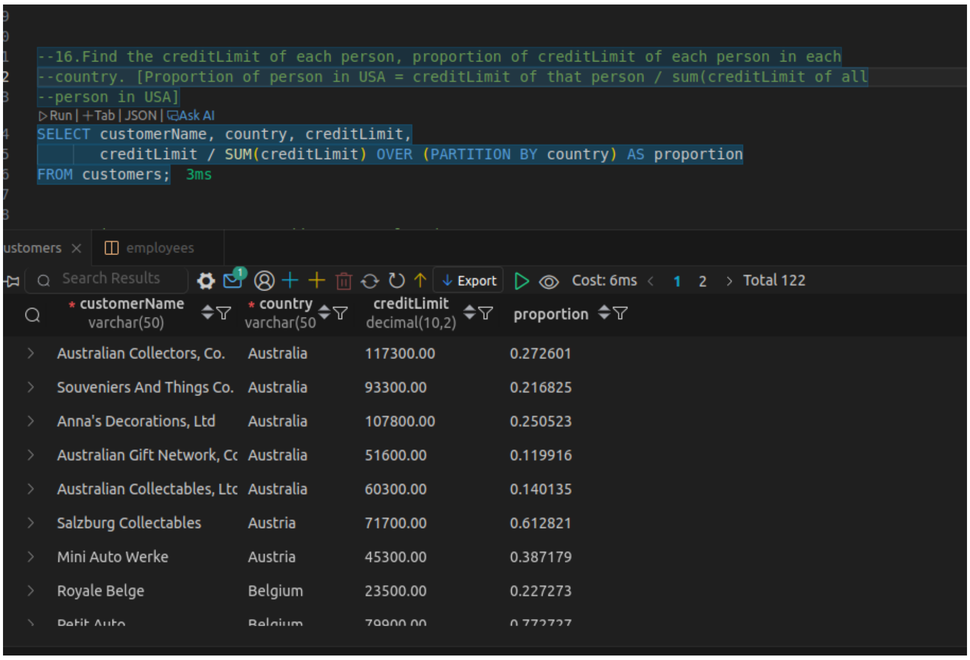
officeCode varchar	city varchar	state varchar	country varchar	payment_pending decimal
4	Paris	(NULL)	France	264592.68
6	Sydney	(NULL)	Australia	139883.37
7	London	(NULL)	UK	112624.80
1	San Francisco	CA	USA	91623.99
3	NYC	NY	USA	84970.25
2	Boston	MA	USA	56656.29
5	Tokyo	Chiyoda-Ku	Japan	0.00

16 Task : Find the creditLimit of each person, proportion of creditLimit of each person in each country. [Proportion of person in USA = creditLimit of that person / sum(creditLimit of all person in USA].

SQL Query

```sql
SELECT customerName, country, creditLimit,
       creditLimit / SUM(creditLimit) OVER (PARTITION BY country)
           AS proportion
FROM customers;
```

Output Screenshot



```sql
--16.Find the creditLimit of each person, proportion of creditLimit of each person in each
--country. [Proportion of person in USA = creditLimit of that person / sum(creditLimit of all
--person in USA]
SELECT customerName, country, creditLimit,
       creditLimit / SUM(creditLimit) OVER (PARTITION BY country) AS proportion
FROM customers;  3ms
```

* customerName varchar(50)	* country varchar(50	creditLimit decimal(10,2)	proportion
Australian Collectors, Co.	Australia	117300.00	0.272601
Souveniers And Things Co.	Australia	93300.00	0.216825
Anna's Decorations, Ltd	Australia	107800.00	0.250523
Australian Gift Network, Cc	Australia	51600.00	0.119916
Australian Collectables, Ltc	Australia	60300.00	0.140135
Salzburg Collectables	Austria	71700.00	0.612821
Mini Auto Werke	Austria	45300.00	0.387179
Royale Belge	Belgium	23500.00	0.227273
Petit Auto	Belgium	79900.00	0.772727

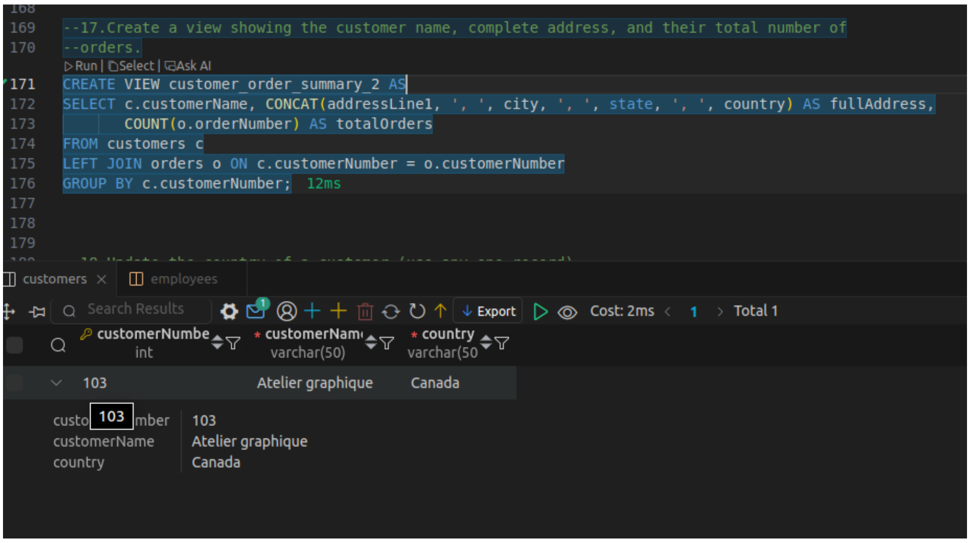
17 Task :Create a view showing the customer name, complete address, and their total number of orders.

SQL Query

```sql
CREATE VIEW customer_order_summary_2 AS
SELECT c.customerName, CONCAT(addressLine1, ’, ’, city, ’, ’,
   state, ’, ’, country) AS fullAddress,
       COUNT(o.orderNumber) AS totalOrders
```

```
FROM customers c
LEFT JOIN orders o ON c.customerNumber = o.customerNumber
GROUP BY c.customerNumber;
```

Output Screenshot



```
--17.Create a view showing the customer name, complete address, and their total number of
--orders.
CREATE VIEW customer_order_summary_2 AS
SELECT c.customerName, CONCAT(addressLine1, ', ', city, ', ', state, ', ', country) AS fullAddress,
        COUNT(o.orderNumber) AS totalOrders
FROM customers c
LEFT JOIN orders o ON c.customerNumber = o.customerNumber
GROUP BY c.customerNumber;  12ms
```

```
customerNumbe  customerNam     country
int            varchar(50)     varchar(50
103            Atelier graphique  Canada

customerNumber  103
customerName    Atelier graphique
country         Canada
```

18 Task :Update the country of a customer (use any one record).

SQL Query

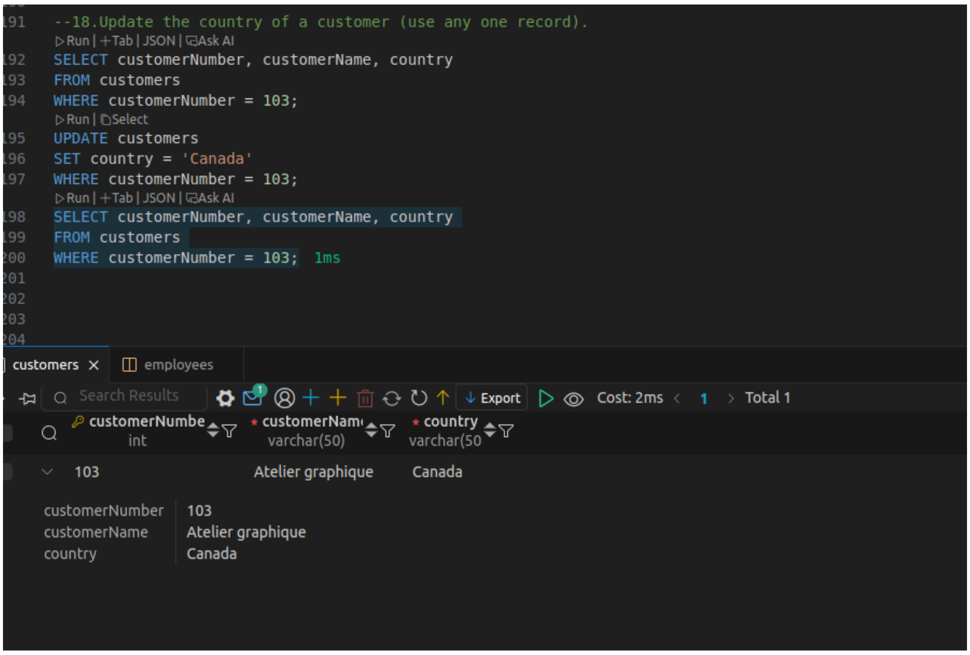
```
SELECT customerNumber, customerName, country
FROM customers
WHERE customerNumber = 103;
UPDATE customers
SET country = 'Canada'
WHERE customerNumber = 103;
SELECT customerNumber, customerName, country
FROM customers
WHERE customerNumber = 103;





CREATE VIEW customer_order_summary AS
SELECT
    c.customerNumber,
    c.customerName,
    CONCAT(c.addressLine1,
           CASE WHEN c.addressLine2 IS NOT NULL THEN CONCAT(', ',
               c.addressLine2) ELSE '' END,
           ', ', c.city,
           CASE WHEN c.state IS NOT NULL THEN CONCAT(', ', c.
              state) ELSE '' END,
           ', ', c.postalCode,
           ', ', c.country) as complete_address,
    COUNT(o.orderNumber) as total_orders
FROM customers c
LEFT JOIN orders o ON c.customerNumber = o.customerNumber
GROUP BY c.customerNumber, c.customerName, complete_address
ORDER BY total_orders DESC; -- example ID
```

Output Screenshot



```
--18.Update the country of a customer (use any one record).
SELECT customerNumber, customerName, country
FROM customers
WHERE customerNumber = 103;
UPDATE customers
SET country = 'Canada'
WHERE customerNumber = 103;
SELECT customerNumber, customerName, country
FROM customers
WHERE customerNumber = 103;  1ms
```

customerNumbe int	customerNam varchar(50)	country varchar(50
103	Atelier graphique	Canada

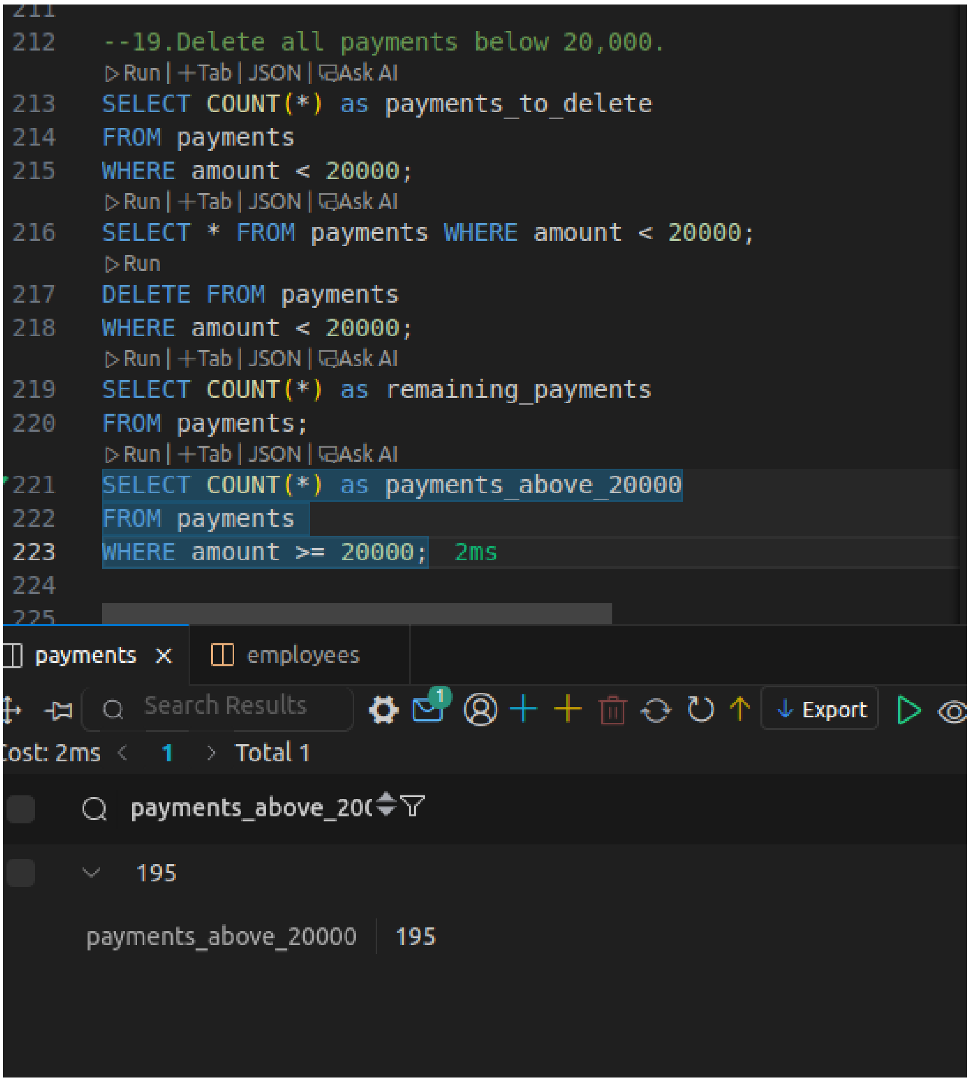
```
customerNumber  103
customerName    Atelier graphique
country         Canada
```

19 Task :Delete all payments below 20,000.

SQL Query

```
SELECT COUNT(*) as payments_to_delete
FROM payments
WHERE amount < 20000;
SELECT * FROM payments WHERE amount < 20000;
DELETE FROM payments
WHERE amount < 20000;
SELECT COUNT(*) as remaining_payments
FROM payments;
SELECT COUNT(*) as payments_above_20000
FROM payments
WHERE amount >= 20000;
```

Output Screenshot



```
--19.Delete all payments below 20,000.
SELECT COUNT(*) as payments_to_delete
FROM payments
WHERE amount < 20000;
SELECT * FROM payments WHERE amount < 20000;
DELETE FROM payments
WHERE amount < 20000;
SELECT COUNT(*) as remaining_payments
FROM payments;
SELECT COUNT(*) as payments_above_20000
FROM payments
WHERE amount >= 20000;  2ms
```

```
195
payments_above_20000 | 195
```

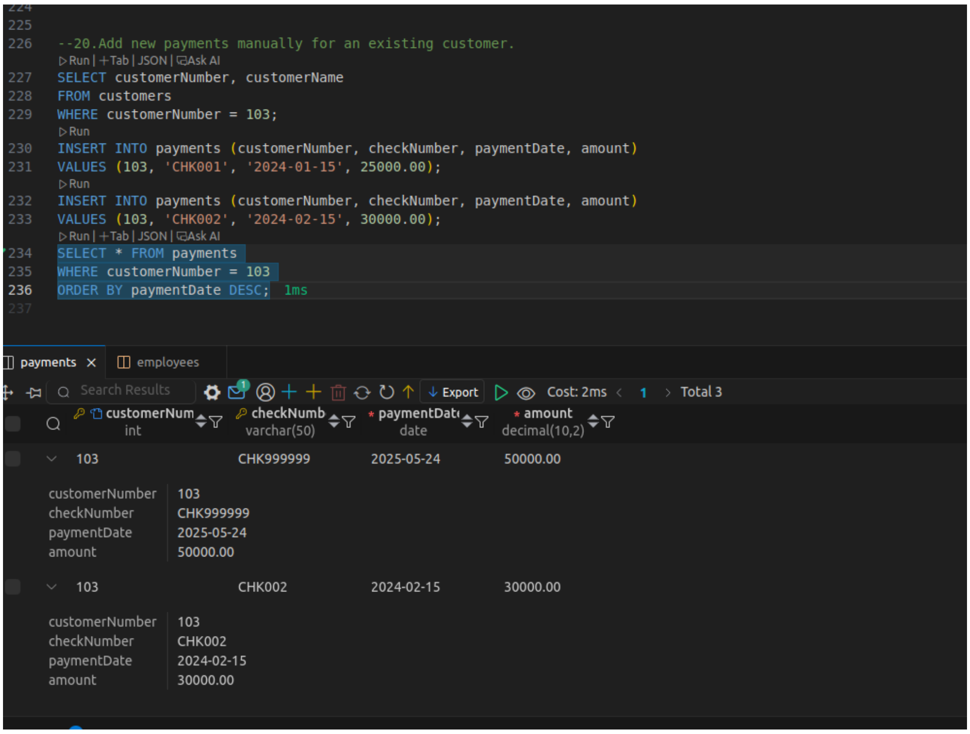
20 Task : Add new payments manually for an existing customer.

SQL Query

```sql
SELECT customerNumber, customerName
FROM customers
WHERE customerNumber = 103;
INSERT INTO payments (customerNumber, checkNumber, paymentDate,
   amount)
VALUES (103, 'CHK001', '2024-01-15', 25000.00);
INSERT INTO payments (customerNumber, checkNumber, paymentDate,
   amount)
VALUES (103, 'CHK002', '2024-02-15', 30000.00);
SELECT * FROM payments
WHERE customerNumber = 103
```

```sql
ORDER BY paymentDate DESC;
```

Output Screenshot



```sql
--20.Add new payments manually for an existing customer.
SELECT customerNumber, customerName
FROM customers
WHERE customerNumber = 103;
INSERT INTO payments (customerNumber, checkNumber, paymentDate, amount)
VALUES (103, 'CHK001', '2024-01-15', 25000.00);
INSERT INTO payments (customerNumber, checkNumber, paymentDate, amount)
VALUES (103, 'CHK002', '2024-02-15', 30000.00);
SELECT * FROM payments
WHERE customerNumber = 103
ORDER BY paymentDate DESC;  1ms
```

```
customerNum    checkNumb      paymentDat    amount
int            varchar(50)    date          decimal(10,2)
103            CHK999999      2025-05-24    50000.00

customerNumber  103
checkNumber     CHK999999
paymentDate     2025-05-24
amount          50000.00

103            CHK002         2024-02-15    30000.00

customerNumber  103
checkNumber     CHK002
paymentDate     2024-02-15
amount          30000.00
```