**Table of Contents**

[1. Show all customers whose creditLimit is greater than 20000 1](#_Toc198969049)

[2. Show the employees who report to VP Sales 1](#_Toc198969050)

[3. Find all customers who have set their state, live in USA, and credit limit is between 100000 and 200000 2](#_Toc198969051)

[4. Find all employees who report to Sales Managers of all types 2](#_Toc198969052)

[5. Find the average credit limit of customers of each country 3](#_Toc198969053)

[6. Find total orders for each date and customer (only dates with >10 orders) 3](#_Toc198969054)

[7. Find supervisor info and supervisee count using subquery (without JOIN) 4](#_Toc198969055)

[8. Find supervisor info and supervisee count using JOIN 4](#_Toc198969056)

[9. Find customers with credit limit greater than average using WITH clause 5](#_Toc198969057)

[10. Find customer rank by credit limit and get 3rd highest 6](#_Toc198969058)

[11. Generate report showing total employees in each office 7](#_Toc198969059)

[12. Generate report showing total customers visited each office 8](#_Toc198969060)

[13. Generate report showing total payment received by each office 9](#_Toc198969061)

[14. Generate report showing total sales by each office 10](#_Toc198969062)

[15. Generate report showing total payment pending for each office 11](#_Toc198969063)

[16. Find creditLimit proportion of each person in each country 12](#_Toc198969064)

[17. Create a view showing customer name, complete address, and total orders 13](#_Toc198969065)

[18. Update the country of a customer 14](#_Toc198969066)

[19. Delete all payments below 20,000 15](#_Toc198969067)

[20. Add new payments manually for an existing customer 16](#_Toc198969068)

SQL Assignment Solutions - Classic Models Database (Query Solutions)

# 1. Show all customers whose creditLimit is greater than 20000

SELECT \*

FROM customers

WHERE creditLimit > 20000;

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 1 Q1

# 2. Show the employees who report to VP Sales

SELECT e.\*

FROM employees e

JOIN employees m ON e.reportsTo = m.employeeNumber

WHERE m.jobTitle = 'VP Sales';

A screen shot of a computer

AI-generated content may be incorrect.

Figure 2 Q2

# 3. Find all customers who have set their state, live 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;

A black screen with white text

AI-generated content may be incorrect.

Figure 3 Q3

# 4. Find all employees who report to Sales Managers of all types

SELECT e.\*

FROM employees e

JOIN employees m ON e.reportsTo = m.employeeNumber

WHERE m.jobTitle LIKE '%Sales Manager%';

A black and white image of a building

AI-generated content may be incorrect.

Figure 4 Q4

# 5. Find the average credit limit of customers of each country

SELECT country, AVG(creditLimit) as average\_credit\_limit

FROM customers

GROUP BY country

ORDER BY average\_credit\_limit DESC;



Figure 5 Q5

# 6. Find total orders for each date and customer (only dates with >10 orders)

SELECT orderDate, customerNumber, COUNT(\*) as total\_orders

FROM orders

GROUP BY orderDate, customerNumber

HAVING COUNT(\*) > 10

ORDER BY orderDate, total\_orders DESC;



Figure 6 Q6

# 7. Find supervisor info and supervisee count using subquery (without JOIN)

SELECT

employeeNumber,

CONCAT(firstName, ' ', lastName) as supervisor\_name,

jobTitle as supervisor\_job\_title,

(SELECT COUNT(\*)

FROM employees e2

WHERE e2.reportsTo = e1.employeeNumber) as total\_supervisees

FROM employees e1

WHERE employeeNumber IN (SELECT DISTINCT reportsTo FROM employees WHERE reportsTo IS NOT NULL)

ORDER BY total\_supervisees DESC;

A black screen with white text

AI-generated content may be incorrect.

Figure 7 Q7

# 8. Find supervisor info and supervisee count using JOIN

SELECT

m.employeeNumber,

CONCAT(m.firstName, ' ', m.lastName) as supervisor\_name,

m.jobTitle as supervisor\_job\_title,

COUNT(e.employeeNumber) as total\_supervisees

FROM employees m

JOIN employees e ON m.employeeNumber = e.reportsTo

GROUP BY m.employeeNumber, m.firstName, m.lastName, m.jobTitle

ORDER BY total\_supervisees DESC;

A black screen with a black background

AI-generated content may be incorrect.

Figure 8 Q8

# 9. Find customers with credit limit greater than average using WITH clause

WITH avg\_credit AS (

SELECT AVG(creditLimit) as avg\_limit

FROM customers

)

SELECT c.\*

FROM customers c, avg\_credit

WHERE c.creditLimit > avg\_credit.avg\_limit

ORDER BY c.creditLimit DESC;

A screenshot of a computer screen

AI-generated content may be incorrect.

Figure 9 Q9

# 10. Find customer rank by credit limit and get 3rd highest

-- Customer rankings

SELECT

customerNumber,

customerName,

creditLimit,

RANK() OVER (ORDER BY creditLimit DESC) as credit\_rank

FROM customers

ORDER BY creditLimit DESC;

-- Customer with 3rd highest credit limit

SELECT

customerNumber,

customerName,

creditLimit

FROM (

SELECT

customerNumber,

customerName,

creditLimit,

RANK() OVER (ORDER BY creditLimit DESC) as credit\_rank

FROM customers

) ranked\_customers

WHERE credit\_rank = 3;

A black screen with a black background

AI-generated content may be incorrect.

Figure 10 Q10

# 11. Generate report showing total employees in each office

SELECT

o.officeCode,

o.city,

o.state,

o.country,

COUNT(e.employeeNumber) as total\_employees

FROM offices o

LEFT JOIN employees e ON o.officeCode = e.officeCode

GROUP BY o.officeCode, o.city, o.state, o.country

ORDER BY total\_employees DESC;

A black screen with a black background

AI-generated content may be incorrect.

Figure 11 Q11

# 12. Generate report showing total customers visited each office

SELECT

o.officeCode,

o.city,

o.state,

o.country,

COUNT(DISTINCT c.customerNumber) as total\_customers

FROM offices o

LEFT JOIN employees e ON o.officeCode = e.officeCode

LEFT JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber

GROUP BY o.officeCode, o.city, o.state, o.country

ORDER BY total\_customers DESC;

A black screen with white text

AI-generated content may be incorrect.

Figure 12 Q12

# 13. Generate report showing total payment received by each office

SELECT

o.officeCode,

o.city,

o.state,

o.country,

COALESCE(SUM(p.amount), 0) as total\_payments

FROM offices o

LEFT JOIN employees e ON o.officeCode = e.officeCode

LEFT JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber

LEFT JOIN payments p ON c.customerNumber = p.customerNumber

GROUP BY o.officeCode, o.city, o.state, o.country

ORDER BY total\_payments DESC;

A black screen with white lines

AI-generated content may be incorrect.

Figure 13 Q13

# 14. Generate report showing total sales by each office

SELECT

o.officeCode,

o.city,

o.state,

o.country,

COALESCE(SUM(od.quantityOrdered \* od.priceEach), 0) as total\_sales

FROM offices o

LEFT JOIN employees e ON o.officeCode = e.officeCode

LEFT JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber

LEFT JOIN orders ord ON c.customerNumber = ord.customerNumber

LEFT JOIN orderdetails od ON ord.orderNumber = od.orderNumber

GROUP BY o.officeCode, o.city, o.state, o.country

ORDER BY total\_sales DESC;

A screenshot of a computer

AI-generated content may be incorrect.

Figure 14 Q14

# 15. Generate report showing total payment pending for each office

SELECT

o.officeCode,

o.city,

o.state,

o.country,

COALESCE(SUM(od.quantityOrdered \* od.priceEach), 0) - COALESCE(SUM(p.amount), 0) as payment\_pending

FROM offices o

LEFT JOIN employees e ON o.officeCode = e.officeCode

LEFT JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber

LEFT JOIN orders ord ON c.customerNumber = ord.customerNumber

LEFT JOIN orderdetails od ON ord.orderNumber = od.orderNumber

LEFT JOIN payments p ON c.customerNumber = p.customerNumber

GROUP BY o.officeCode, o.city, o.state, o.country

ORDER BY payment\_pending DESC;

A computer screen with white text

AI-generated content may be incorrect.

Figure 15 Q15

# 16. Find creditLimit proportion of each person in each country

SELECT

customerNumber,

customerName,

country,

creditLimit,

SUM(creditLimit) OVER (PARTITION BY country) as country\_total\_credit,

ROUND(creditLimit / SUM(creditLimit) OVER (PARTITION BY country) \* 100, 2) as credit\_proportion\_percentage

FROM customers

ORDER BY country, credit\_proportion\_percentage DESC;

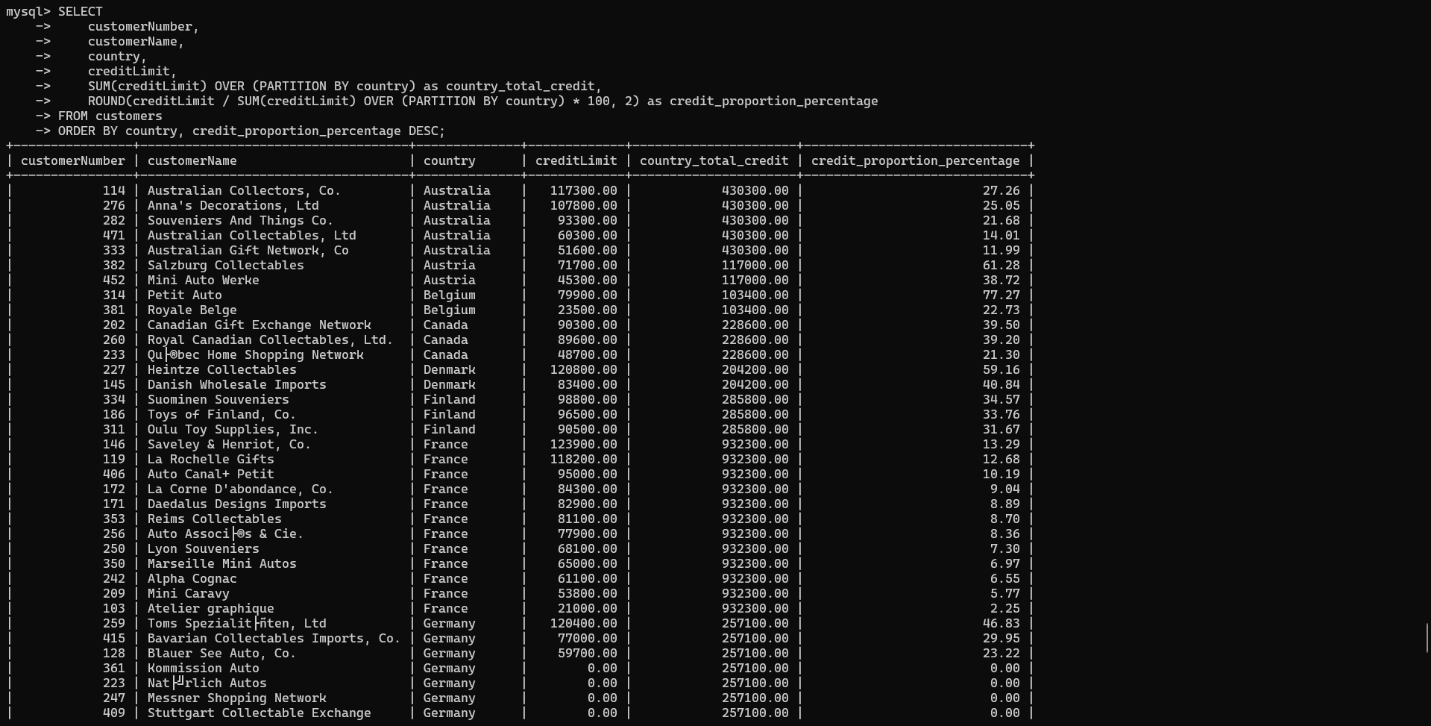


Figure 16 Q16

# 17. Create a view showing customer name, complete address, and total orders

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;

-- To view the created view:

SELECT \* FROM customer\_order\_summary;

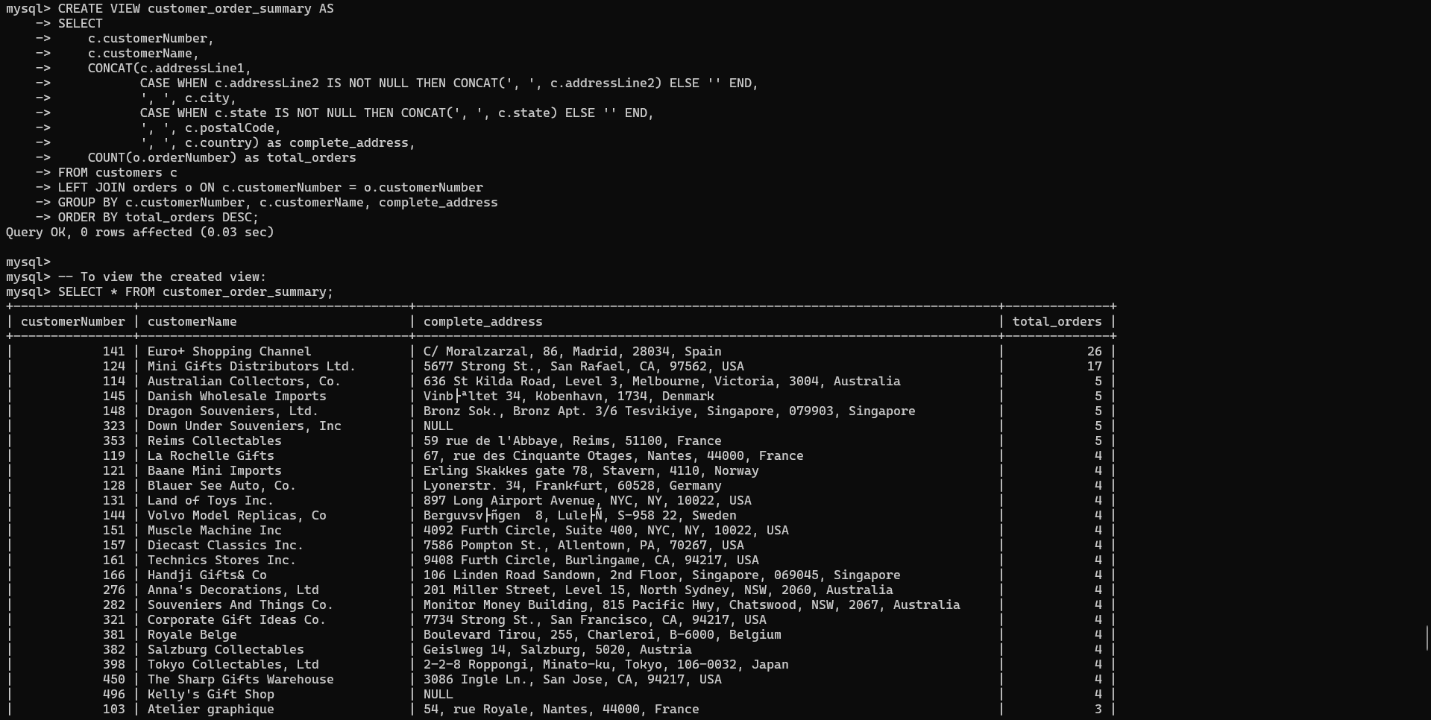


Figure 17 Q17

# 18. Update the country of a customer

-- First, check current data

SELECT customerNumber, customerName, country

FROM customers

WHERE customerNumber = 103;

-- Update the country

UPDATE customers

SET country = 'Canada'

WHERE customerNumber = 103;

-- Verify the update

SELECT customerNumber, customerName, country

FROM customers

WHERE customerNumber = 103;

A black screen with a black background

AI-generated content may be incorrect.

Figure 18 Q18

# 19. Delete all payments below 20,000

-- First, check how many payments will be affected

SELECT COUNT(\*) as payments\_to\_delete

FROM payments

WHERE amount < 20000;

-- Delete payments below 20,000

DELETE FROM payments

WHERE amount < 20000;

-- Verify deletion

SELECT COUNT(\*) as remaining\_payments

FROM payments;

A black screen with a black background

AI-generated content may be incorrect.

Figure 19 Q19

# 20. Add new payments manually for an existing customer

-- Check existing customer

SELECT customerNumber, customerName

FROM customers

WHERE customerNumber = 103;

-- Insert new payment

INSERT INTO payments (customerNumber, checkNumber, paymentDate, amount)

VALUES (103, 'CHK001', '2024-01-15', 25000.00);

-- Insert another payment

INSERT INTO payments (customerNumber, checkNumber, paymentDate, amount)

VALUES (103, 'CHK002', '2024-02-15', 30000.00);

-- Verify the new payments

SELECT \* FROM payments

WHERE customerNumber = 103

ORDER BY paymentDate DESC;

A black screen with white lines

AI-generated content may be incorrect.

Figure 20 Q20