

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

Database Server Tools Scripting Help

SQL File 4*

Limit to 1000 rows

```
1 • select *
2 from customers
3 where creditLimit > 20000
4 order by creditLimit
```

Result Grid

customerNumber	customerName	contactLastName	contactFirstName	phone	addressLine1	addressLine2	city	state	postalCode	country	salesRepEmployeeNumber	creditLimit
103	Atelier graphique	Schmitt	Carrie	40.32.2555	54, rue Royale		Nantes		44000	France	1370	21000.00
198	Auto-Moto Classics Inc.	Taylor	Leslie	6175558428	16780 Pompton St.		Bridgwater	MA	08339	USA	1216	23000.00
381	Royale Belge	Cartran	Pascale	(071) 23 67 2555	Boulevard Trou, 255		Charleroi		B-6000	Belgium	1401	23000.00
473	Prau da Colazione	Ruotti	Franco	+39 022515555	20093 Cologno Monzese	Alessandro Volta 36	Milan			Italy	1401	34800.00
456	Microscale Inc.	Choi	Yu	2125551957	5290 North Pendale Street	Suite 200	NYC	NY	10022	USA	1286	39800.00
362	Gifts4Allages.com	Yoshida	Juri	6175559555	8616 Spinnaker Dr.		Boston	MA	01003	USA	1216	41900.00
173	Tekni Collectables Inc.	Brown	William	2015559300	7476 Moss Rd.		Newark	NJ	07102	USA	1323	43000.00
489	Double Decker Gift Stores, Ltd	Smith	Thomas	(171) 555-7555	120 Hanover Sq.		London		W1A 1DP	UK	1501	43300.00
173	Cambridge Collectables Co.	Tsang	Jerry	6175555555	4638 Belden Av.		Cambridge	MA	02147	USA	1189	49400.00
452	Mini Auto Werke	Mendel	Roland	7675-3555	Kirchgasse 6		Graz		8010	Austria	1401	45300.00
233	Quebec Home Shopping Netw...	Fresnais	Jean	(514) 555-8054	43 rue St. Laurent		Montréal	Qué...	H3J 1C3	Canada	1286	48700.00
447	Gift Ideas Corp.	Lewis	Dan	2035554407	2440 Pompton St.		Glendale	CT	07061	USA	1323	49700.00
333	Australian Gift Network, Co	Callaghan	Ben	61-7-3844-6555	31 Duncan St. West End		South Bris...	Que...	4101	Australia	1611	51600.00
144	Volvo Model Restoras, Co	Berglund	Christina	0801-12 3555	Berguvägen 8		Luleå		S-958 22	Sweden	1504	53100.00
209	Mini Caravay	Citeaux	Frédérique	88.60.1555	24, place Kléber		Strasbourg		67000	France	1370	53800.00
475	West Coast Collectables Co.	Thompson	Steve	3105553722	3675 Furth Circle		Burbank	CA	94019	USA	1166	55400.00
347	Men R' US Retailers, Ltd.	Chandler	Brian	2155554369	6047 Douglas Av.		Los Angeles	CA	91003	USA	1166	57700.00
211	King Kong Collectables, Co	Gao	Hile	+852 2251 1555	Bank of China Tower	1 Garden Road	Central H...			Hong K...	1621	58600.00
144	CAF Imports	Fernandez	Jesus	+34 913 728 555	Merchants House	27-30 Merchant's ...	Madrid		28023	Spain	1702	59600.00
128	Blauer See Auto, Co.	Keitel	Roland	+49 69 66 90 2555	Lyonenstr. 34		Frankfurt		60528	Germany	1504	59700.00
471	Australian Collectables, Ltd	Clenahan	Sean	61-9-3844-6555	7 Allen Street		Glen Warr...	Vict...	3190	Australia	1611	60300.00
216	Enaco Distributors	Saavedra	Eduardo	(93) 203 4555	Rambla de Cataluña, 23		Barcelona		08022	Spain	1702	60300.00
487	Signal Collectables Ltd.	Taylor	Sue	4155554312	2793 Furth Circle		Brisbane	CA	94217	USA	1165	60300.00
242	Alpha Cognac	Roulet	Annette	61.77.6555	1 rue Alsace-Lorraine		Toulouse		31000	France	1370	61100.00
129	Mini Wheels Co.	Murphy	Julie	6505555787	5557 North Pendale Street		San Franc...	CA	94217	USA	1165	64600.00
350	Marseille Mini Autos	Lebihan	Laurence	91.24.4555	12, rue des Bouchers		Marseille		13008	France	1337	65000.00
484	Iberia Gift Imports, Corp.	Rioel	Jose Pedro	(95) 555 82 82	C/ Romero, 33		Sevilla		41101	Spain	1702	65700.00
424	Classic Legends Inc.	Hernandez	Maria	2125559493	5905 Pompton St.	Suite 750	NYC	NY	10022	USA	1286	67500.00
250	Lyon Souveniers	Da Silva	Daniel	+33 1 46 62 7555	27 rue du Colonel Pierre A...		Paris		75508	France	1337	68100.00
164	Palace Mini Collectables	Bonville	Marc	4155554312	2793 Furth Circle		Brisbane	CA	94217	USA	1165	64600.00

2. Show the employees who report to VP Sales.

SQL File 5*

Limit to 1000 rows

```
1 • select e.employeeNumber, e.firstName, e.lastName, e.jobTitle
2 from employees e
3 where e.reportsTo = (
4     select employeeNumber
5     from employees
6     where jobTitle = 'VP Sales'
7 );
8
9
```

Result Grid

employeeNumber	firstName	lastName	jobTitle
1088	William	Patterson	Sales Manager (APAC)
1102	Gerard	Bondur	Sale Manager (EMEA)
1143	Anthony	Bow	Sales Manager (NA)
1621	Mami	Nishi	Sales Rep
NULL	NULL	NULL	NULL

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.

SQL File 6*

```
1 • select *
2   from customers
3  where state is not Null
4        and country = "USA"
5        and creditLimit between 100000 and 200000;
```

Result Grid

customerNumber	customerName	contactLastName	contactFirstName	phone	addressLine1	addressLine2	city	state	postalCode	country	salesRepEmployeeNumber	creditLimit
131	Land of Toys Inc.	Lee	Kwai	2125557818	897 Long Airport Avenue		NYC	NY	10022	USA	1323	114900.00
151	Musde Machine Inc.	Young	Jeff	2125557413	4092 Furth Circle	Suite 400	NYC	NY	10022	USA	1286	138500.00
157	Diecast Classics Inc.	Leong	Kelvin	2155551555	7586 Pompton St.		Allentown	PA	70267	USA	1216	100600.00
239	Collectable Mini Designs Co.	Thompson	Valarie	7605558146	361 Furth Circle		San Diego	CA	91217	USA	1166	105000.00
286	Marta's Replicas Co.	Hernandez	Marta	6175558555	39323 Spinnaker Dr.		Cambridge	MA	51247	USA	1216	123700.00
319	Mini Classics	Prick	Steve	9145554562	3758 North Pendale Street		White Plains	NY	24067	USA	1323	102700.00
321	Corporate Gift Ideas Co.	Brown	Julie	6505551386	7734 Strong St.		San Francisco	CA	94217	USA	1165	105000.00
363	Online Diecast Creations Co.	Young	Dorothy	6035558647	2304 Long Airport Avenue		Nashua	NH	62005	USA	1216	114200.00

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

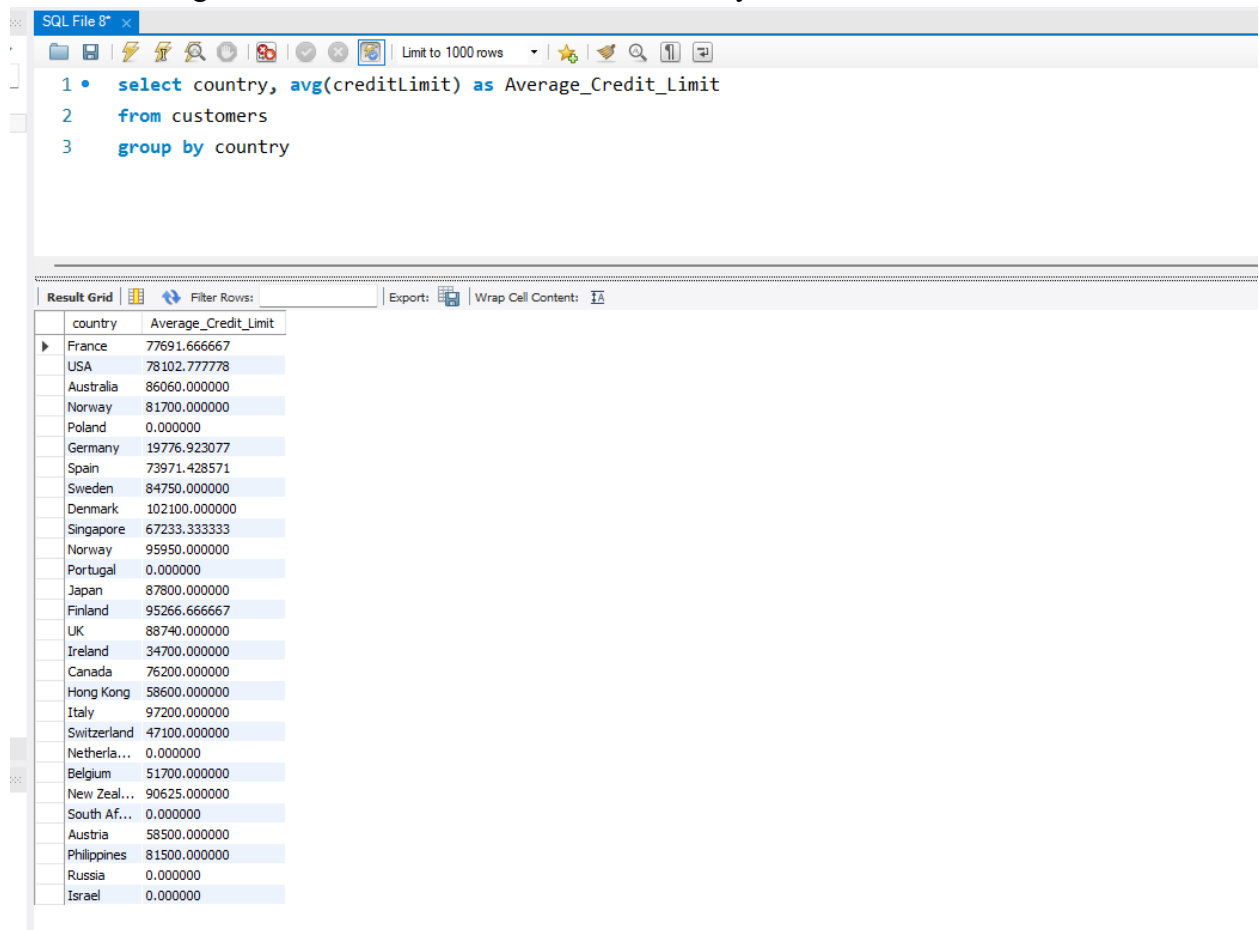
SQL File 7*

```
1 • select *
2   from employees
3  where reportsTo In (
4    select employeeNumber
5    from employees
6    where jobtitle like 'Sales Manager%'
7  )
```

Result Grid

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

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



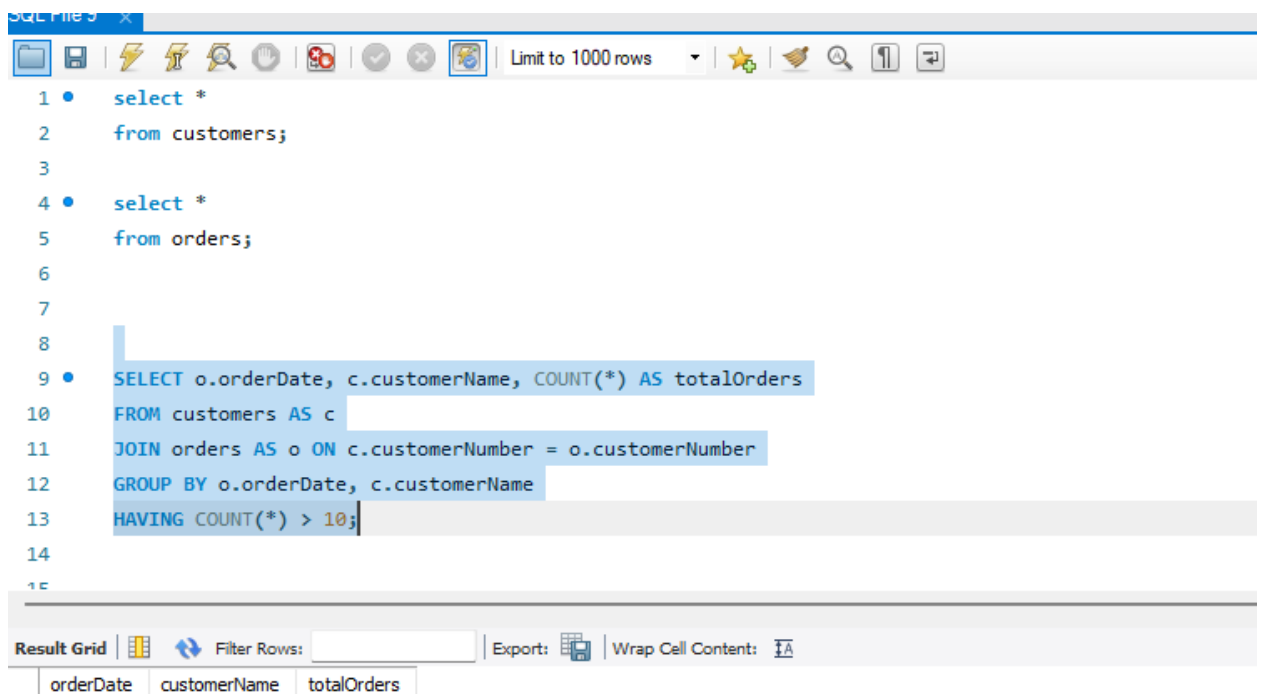
The screenshot shows a SQL IDE window titled "SQL File 8". The query editor contains the following SQL code:

```
1 • select country, avg(creditLimit) as Average_Credit_Limit
2   from customers
3  group by country
```

Below the query editor, the "Result Grid" is displayed, showing the results of the query. The grid has two columns: "country" and "Average_Credit_Limit". The results are as follows:

country	Average_Credit_Limit
France	77691.666667
USA	78102.777778
Australia	86060.000000
Norway	81700.000000
Poland	0.000000
Germany	19776.923077
Spain	73971.428571
Sweden	84750.000000
Denmark	102100.000000
Singapore	67233.333333
Norway	95950.000000
Portugal	0.000000
Japan	87800.000000
Finland	95266.666667
UK	88740.000000
Ireland	34700.000000
Canada	76200.000000
Hong Kong	58600.000000
Italy	97200.000000
Switzerland	47100.000000
Netherla...	0.000000
Belgium	51700.000000
New Zeal...	90625.000000
South Af...	0.000000
Austria	58500.000000
Philippines	81500.000000
Russia	0.000000
Israel	0.000000

6. 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.



The screenshot shows a SQL IDE window titled "SQL File 9". The query editor contains the following SQL code:

```
1 • select *
2   from customers;
3
4 • select *
5   from orders;
6
7
8
9 • SELECT o.orderDate, c.customerName, COUNT(*) AS totalOrders
10  FROM customers AS c
11  JOIN orders AS o ON c.customerNumber = o.customerNumber
12  GROUP BY o.orderDate, c.customerName
13  HAVING COUNT(*) > 10;
```

Below the query editor, the "Result Grid" is displayed, showing the columns "orderDate", "customerName", and "totalOrders".

orderDate	customerName	totalOrders
-----------	--------------	-------------

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

The screenshot shows the SQL Developer interface with a query window titled 'SQL File 10*'. The query is as follows:

```
4 • SELECT
5     CONCAT(firstName, ' ', lastName) AS supervisorName,
6     jobTitle,
7     (
8         SELECT COUNT(*)
9         FROM employees AS e2
10        WHERE e2.reportsTo = e1.employeeNumber
11    ) AS totalSupervisees
12 FROM employees AS e1
13 WHERE employeeNumber IN (
14     SELECT reportsTo FROM employees WHERE reportsTo IS NOT NULL
15 );
```

Below the query window, the 'Result Grid' is displayed, showing the results of the query. The columns are 'supervisorName', 'jobTitle', and 'totalSupervisees'. The results are as follows:

supervisorName	jobTitle	totalSupervisees
Diane Murphy	President	2
Mary Patterson	VP Sales	4
William Patterson	Sales Manager (APAC)	3
Gerard Bondur	Sale Manager (EMEA)	6
Anthony Bow	Sales Manager (NA)	6
Mami Nishi	Sales Rep	1

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

The screenshot shows the SQL Developer interface with a query window titled 'SQL File 3*'. The query is as follows:

```
1 • SELECT
2     CONCAT(s.firstName, ' ', s.lastName) AS supervisorName,
3     s.jobTitle,
4     COUNT(e.employeeNumber) AS totalSupervisees
5 FROM employees e
6 JOIN employees s
7     ON e.reportsTo = s.employeeNumber
8 GROUP BY s.employeeNumber;
```

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

The screenshot shows a SQL IDE window titled "SQL File 4*". The query editor contains the following SQL code:

```
1 • select *
2   from customers;
3
4 • with Credit_Limit as
5   (
6     select avg(creditLimit) as avgCL
7     from customers
8   )
9
10  select customerName, creditLimit
11  from customers, Credit_Limit
12  where creditLimit > Credit_Limit.avgCL;
```

The IDE interface includes a toolbar with icons for file operations, a "Limit to 1000 rows" dropdown, and a "Result Grid" tab. The "Result Grid" is active, displaying the following data:

customerName	creditLimit
Signal Gift Stores	71800.00
Australian Collectors, Co.	117300.00
La Rochelle Gifts	118200.00
Baane Mini Imports	81700.00
Mini Gifts Distributors Ltd.	210500.00
Land of Toys Inc.	114900.00
Euro+ Shopping Channel	227600.00
Danish Wholesale Imports	83400.00
Saveley & Henriot, Co.	123900.00
Dragon Souvenirs, Ltd.	103800.00
Muscle Machine Inc	138500.00
Diecast Classics Inc.	100600.00
Technics Stores Inc.	84600.00
Handji Gifts & Co	97900.00
Herkku Gifts	96800.00
Daedalus Designs Imports	82900.00
La Corne D'abondance, Co.	84300.00

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.

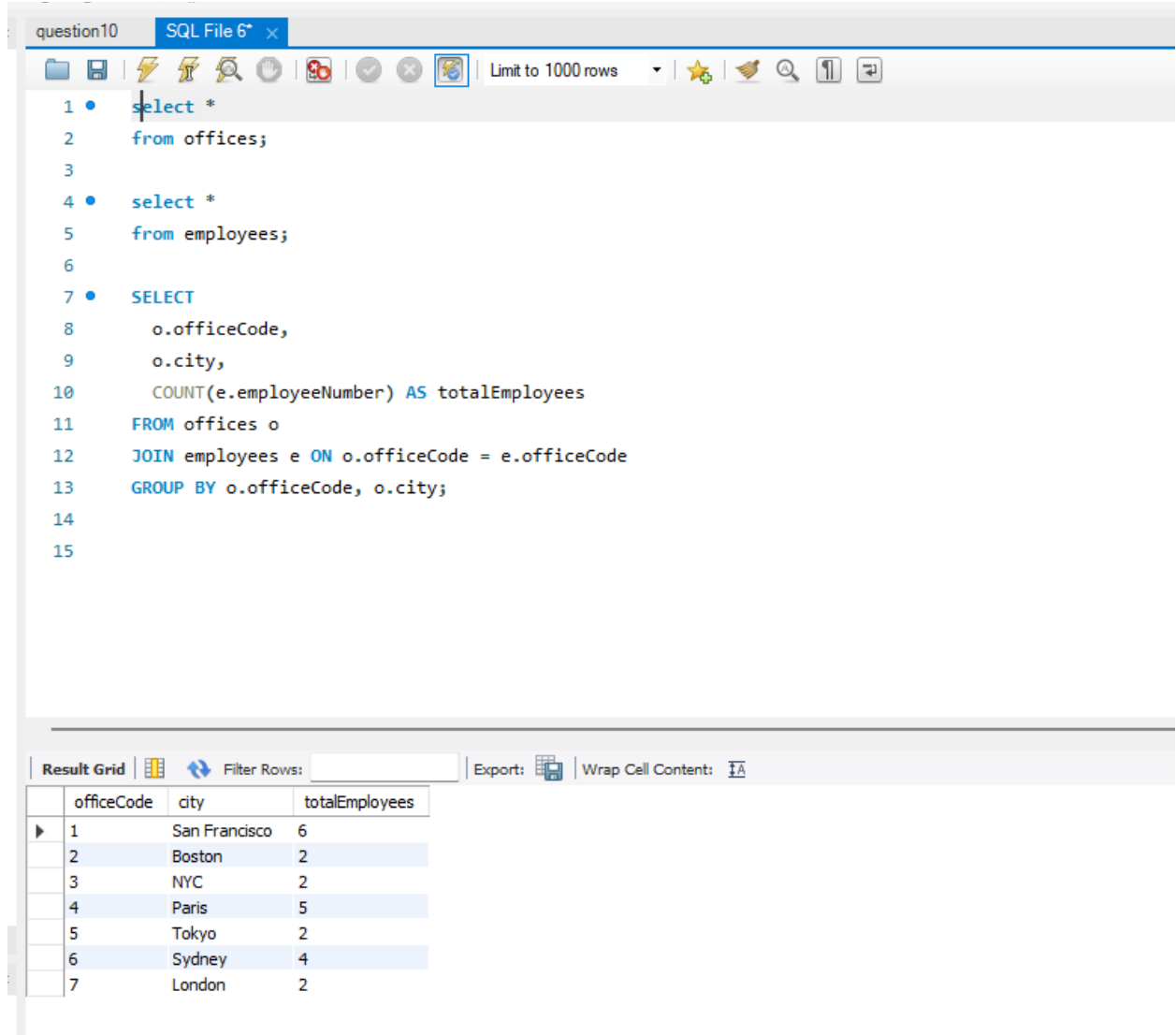
The screenshot shows a SQL IDE window titled "SQL File 5". The query editor contains the following SQL code:

```
1 • SELECT
2     customerName,
3     creditLimit,
4     RANK() OVER (ORDER BY creditLimit DESC) AS rankPosition
5 FROM customers;
6
7 • WITH ranked_customers AS (
8     SELECT
9         customerName,
10        creditLimit,
11        RANK() OVER (ORDER BY creditLimit DESC) AS rankPosition
12    FROM customers
13 )
14 SELECT *
15 FROM ranked_customers
16 WHERE rankPosition = 3;
```

The bottom of the window shows the "Result Grid" with the following data:

	customerName	creditLimit	rankPosition
▶	Vida Sport, Ltd	141300.00	3

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



The screenshot shows a SQL IDE window titled "question10" and "SQL File 6". The query editor contains the following SQL code:

```
1 • select *
2   from offices;
3
4 • select *
5   from employees;
6
7 • SELECT
8     o.officeCode,
9     o.city,
10    COUNT(e.employeeNumber) AS totalEmployees
11  FROM offices o
12  JOIN employees e ON o.officeCode = e.officeCode
13  GROUP BY o.officeCode, o.city;
14
15
```

The results pane shows a table with the following data:

	officeCode	city	totalEmployees
▶	1	San Francisco	6
	2	Boston	2
	3	NYC	2
	4	Paris	5
	5	Tokyo	2
	6	Sydney	4
	7	London	2

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

The screenshot shows a SQL IDE window titled "SQL File 4* x". The query editor contains the following SQL code:



```
1 • select *
2   from customers;
3
4 • select *
5   from offices;
6
7 • select *
8   from employees;
9
10 • SELECT
11     o.officeCode,
12     o.city,
13     COUNT(DISTINCT c.customerNumber) AS totalCustomers
14   FROM offices o
15   JOIN employees e ON o.officeCode = e.officeCode
16   JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber
17   GROUP BY o.officeCode, o.city;
18
```

Below the query editor, the "Result Grid" tab is active, displaying the results of the query. The grid has columns for "officeCode", "city", and "totalCustomers". The results are as follows:

	officeCode	city	totalCustomers
▶	1	San Francisco	12
	2	Boston	12
	3	NYC	15
	4	Paris	29
	5	Tokyo	5
	6	Sydney	10
	7	London	17

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

```
12
13
14 • select
15     o.officeCode,
16     o.city as office_name,
17     o.state,
18     o.country,
19     sum(p.amount) as total_payments
20 from offices o
21 join employees e On o.officeCode = e.officeCode
22 join customers c On e.employeeNumber = c.salesRepEmployeeNumber
23 join payments p On c.customerNumber = p.customerNumber
24 group by o.officeCode,o.state,o.country
25
```

Result Grid					
Filter Rows: <input type="text"/>					
Export:  Wrap Cell Content: 					
	officeCode	office_name	state	country	total_payments
▶	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. Generate a report that shows total sales (in amount) by each office using order details table and other essential tables.

```
20 • select
21     o.officeCode,
22     o.country,
23     sum(od.quantityOrdered * od.priceEach) as total_sales
24 from offices o
25 join employees e on e.officeCode = o.officeCode
26 join customers c on e.employeeNumber = c.salesRepEmployeeNumber
27 join orders ord on c.customerNumber = ord.customerNumber
28 join orderdetails od on ord.orderNumber = od.orderNumber
29 group by o.officeCode, o.country
30
31
32
```

Result Grid				Filter Rows:		Export:	Wrap Cell Content:
	officeCode	country	total_sales				
▶	4	France	3083761.58				
	1	USA	1429063.57				
	6	Australia	1147176.35				
	7	UK	1436950.70				
	3	USA	1157589.72				
	5	Japan	457110.07				
	2	USA	892538.62				

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

```
question15 x
Limit to 1000 rows
1 • SELECT
2     o.officeCode,
3     o.city AS officeName,
4     o.country,
5     s.total_sales,
6     IFNULL(p.total_payments, 0) AS total_payments,
7     s.total_sales - IFNULL(p.total_payments, 0) AS total_pending
8 FROM offices o
9 JOIN (
10     SELECT o.officeCode, SUM(od.quantityOrdered * od.priceEach) AS total_sales
11     FROM offices o
12     JOIN employees e ON o.officeCode = e.officeCode
13     JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber
14     JOIN orders ord ON c.customerNumber = ord.customerNumber
15     JOIN orderdetails od ON ord.orderNumber = od.orderNumber
16     GROUP BY o.officeCode
17 ) s ON o.officeCode = s.officeCode
18 LEFT JOIN (
19     SELECT o.officeCode, SUM(p.amount) AS total_payments
20     FROM offices o
21     JOIN employees e ON o.officeCode = e.officeCode
22     JOIN customers c ON e.employeeNumber = c.salesRepEmployeeNumber
23     JOIN payments p ON c.customerNumber = p.customerNumber
24     GROUP BY o.officeCode
25 ) p ON o.officeCode = p.officeCode;
26
```

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)].

SQL File 8*

Limit to 1000 rows

```
1 • SELECT
2     customerName,
3     c.country,
4     creditLimit,
5     ROUND(creditLimit / country_total.totalCreditLimit, 4) AS credit_proportion
6 FROM customers c
7 JOIN (
8     SELECT country, SUM(creditLimit) AS totalCreditLimit
9     FROM customers
10    GROUP BY country
11 ) AS country_total
12 ON c.country = country_total.country;
```

Result Grid

Filter Rows:

Export: | Wrap Cell Content: 1.5

customerName	country	creditLimit	credit_proportion
Atelier graphique	France	21000.00	0.0225
Signal Gift Stores	USA	71800.00	0.0255
Australian Collectors, Co.	Australia	117300.00	0.2726
La Rochelle Gifts	France	118200.00	0.1268
Baane Mini Imports	Norway	81700.00	1.0000
Mini Gifts Distributors Ltd.	USA	210500.00	0.0749
Havel & Zbyszek Co	Poland	0.00	NULL
Blauer See Auto, Co.	Germany	59700.00	0.2322
Mini Wheels Co.	USA	64600.00	0.0230
Land of Toys Inc.	USA	114900.00	0.0409
Euro+ Shopping Channel	Spain	227600.00	0.4396
Volvo Model Replicas, Co	Sweden	53100.00	0.3133
Danish Wholesale Imports	Denmark	83400.00	0.4084
Saveley & Henriot, Co.	France	123900.00	0.1329
Dragon Souvenirs, Ltd.	Singapore	103800.00	0.5146
Muscle Machine Inc	USA	138500.00	0.0493
Diecast Classics Inc.	USA	100600.00	0.0358
Technics Stores Inc.	USA	84600.00	0.0301
Handji Gifts & Co	Singapore	97900.00	0.4854
Herku Gifts	Norway	96800.00	0.5044
American Souvenirs Inc	USA	0.00	0.0000
Porto Imports Co.	Portugal	0.00	NULL

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

The screenshot shows a SQL IDE with a script editor and a result grid. The script editor contains the following SQL code:

```
1 CREATE VIEW customer_order_summary AS
2 SELECT
3     c.customerName,
4     CONCAT_WS(' ',
5         c.addressLine1,
6         c.addressLine2,
7         c.city,
8         c.state,
9         c.postalCode,
10        c.country
11    ) AS completeAddress,
12     COUNT(o.orderNumber) AS totalOrders
13 FROM customers c
14 LEFT JOIN orders o ON c.customerNumber = o.customerNumber
15 GROUP BY c.customerNumber;
16
17 SELECT * FROM customer_order_summary;
18
```

The result grid displays the data returned by the second query. It has three columns: customerName, completeAddress, and totalOrders. The data is as follows:

customerName	completeAddress	totalOrders
Atelier graphique	54, rue Royale, Nantes, 44000, France	3
Signal Gift Stores	8489 Strong St., Las Vegas, NV, 83030, USA	3
Australian Collectors, Co.	636 St Kilda Road, Level 3, Melbourne, Victoria, ...	5
La Rochelle Gifts	67, rue des Cinquante Otages, Nantes, 44000, ...	4
Baane Mini Imports	Erling Skakkes gate 78, Stavern, 4110, Norway	4
Mini Gifts Distributors Ltd.	5677 Strong St., San Rafael, CA, 97562, USA	17
Havel & Zbyszek Co	ul. Filtrawa 68, Warszawa, 01-012, Poland	0
Blauer See Auto, Co.	Lyonerstr. 34, Frankfurt, 60528, Germany	4
Mini Wheels Co.	5557 North Pendale Street, San Francisco, CA, ...	3
Land of Toys Inc.	897 Long Airport Avenue, NYC, NY, 10022, USA	4

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

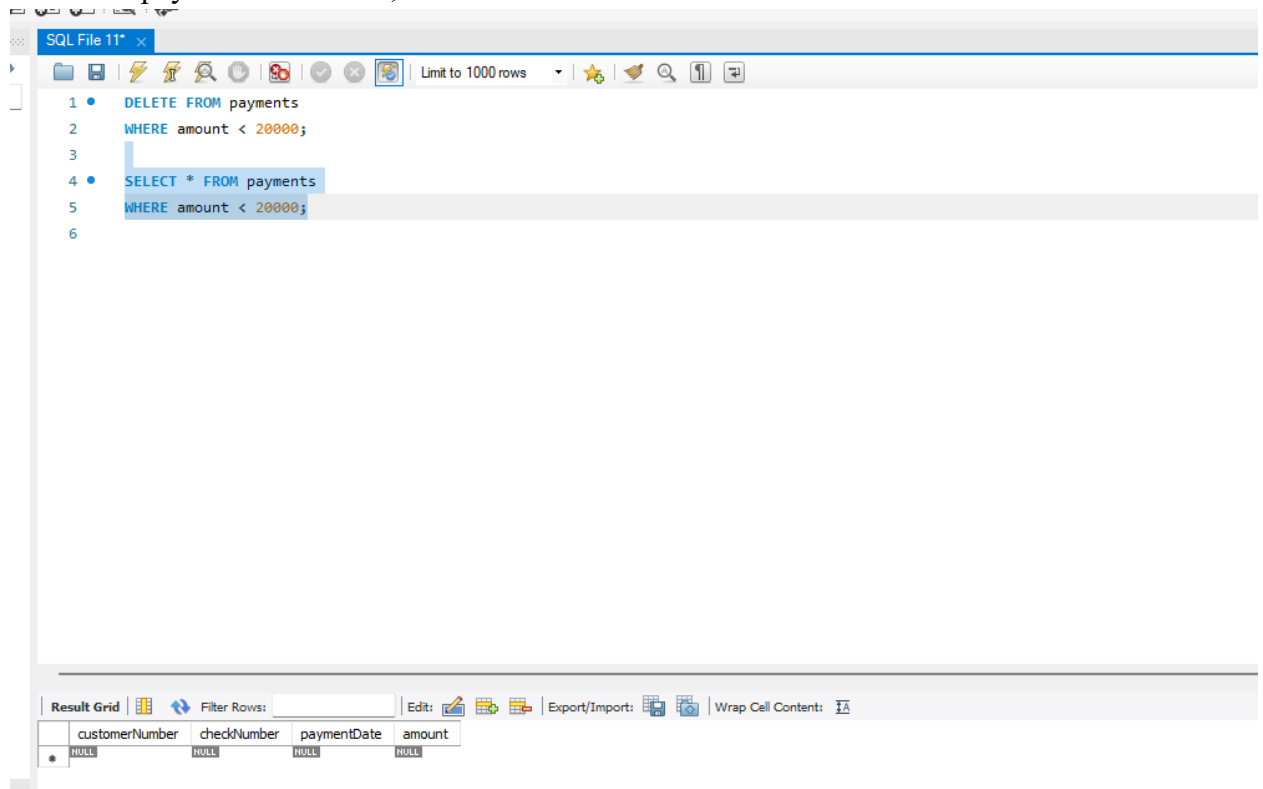
The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and search. The main editor area contains the following SQL code:

```
1 • UPDATE customers
2   SET country = 'Nepal'
3   WHERE customerNumber = 114;
4
5 • SELECT customerName, country
6   FROM customers
7   WHERE customerNumber = 114;
```

Below the editor is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The result grid displays the following data:

customerName	country
Australian Collectors, Co.	Nepal

19. Delete all payments below 20,000.



The screenshot shows a SQL IDE window titled "SQL File 11*". The script contains the following SQL statements:

```
1 • DELETE FROM payments
2   WHERE amount < 20000;
3
4 • SELECT * FROM payments
5   WHERE amount < 20000;
6
```

The interface includes a toolbar with icons for file operations, execution, and search. A "Limit to 1000 rows" dropdown is visible. Below the script editor is a "Result Grid" section with a "Filter Rows:" input field and an "Edit:" button. The result grid displays the following columns: customerNumber, checkNumber, paymentDate, and amount. The first row shows values: 10000, 10000, 10000, 10000.

customerNumber	checkNumber	paymentDate	amount
10000	10000	10000	10000

20. Add new payments manually for an existing customer.

The screenshot shows a SQL IDE window titled "SQL File 3*". The script contains two SQL statements: an INSERT statement to add a new payment record and a SELECT statement to retrieve all records for customer 103, ordered by payment date in descending order.

```
1 • INSERT INTO payments (customerNumber, checkNumber, paymentDate, amount)
2   VALUES (103, 'CHK6969', '2025-05-23', 696969.00);
3
4 • SELECT * FROM payments
5   WHERE customerNumber = 103
6   ORDER BY paymentDate DESC;
7
```

Below the script, the "Result Grid" shows the results of the SELECT statement. It displays a table with four columns: customerNumber, checkNumber, paymentDate, and amount. The first row shows the data for customer 103, and the second row shows NULL values.

customerNumber	checkNumber	paymentDate	amount
103	CHK6969	2025-05-23	696969.00
NULL	NULL	NULL	NULL