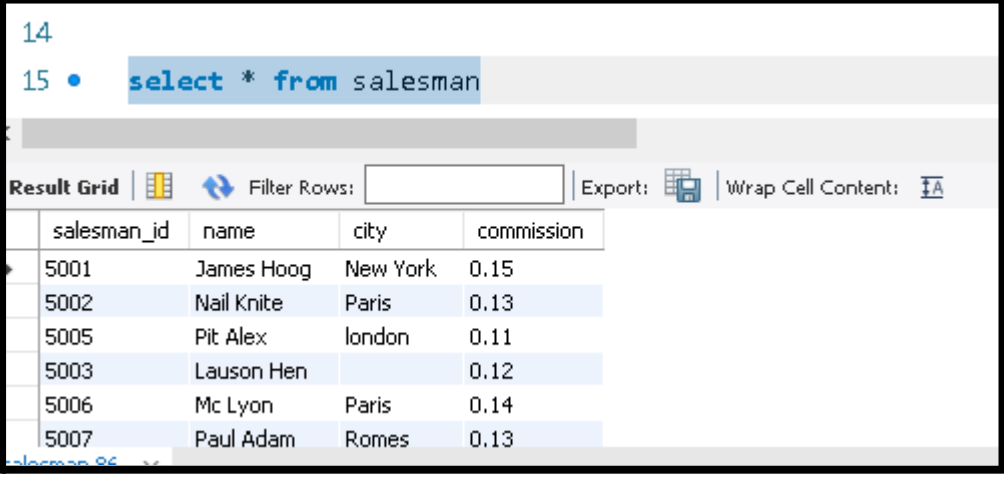


1. Write a SQL statement that displays all the information about all salespeople.

```
select * from salesman
```

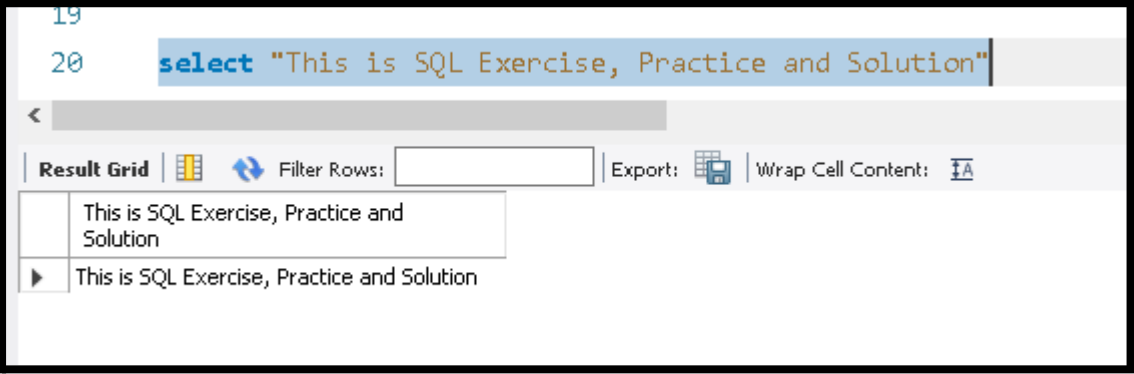


The screenshot shows a SQL IDE interface. The query editor at the top contains the text `select * from salesman`. Below the editor is a toolbar with icons for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. The 'Result Grid' is displayed below the toolbar, showing a table with the following data:

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	london	0.11
5003	Lauson Hen		0.12
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Romes	0.13

2. Write a SQL statement to display a string "This is SQL Exercise, Practice and Solution".

```
select "This is SQL Exercise, Practice and Solution"
```

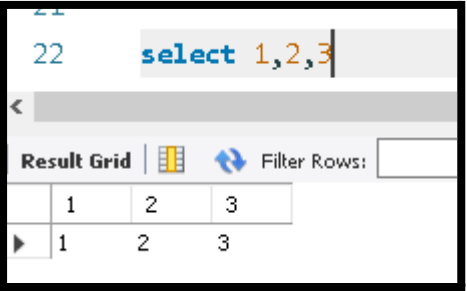


The screenshot shows a SQL IDE interface. The query editor at the top contains the text `select "This is SQL Exercise, Practice and Solution"`. Below the editor is a toolbar with icons for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. The 'Result Grid' is displayed below the toolbar, showing a table with the following data:

This is SQL Exercise, Practice and Solution

3. Write a SQL query to display three numbers in three columns.

```
Select 1,2,3
```

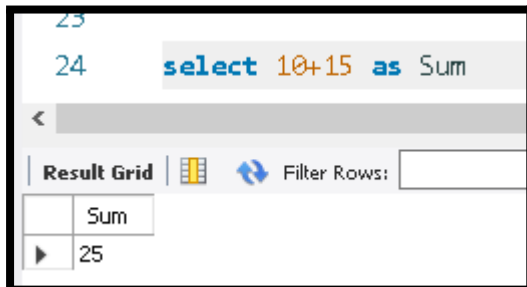


The screenshot shows a SQL IDE interface. The query editor at the top contains the text `select 1,2,3`. Below the editor is a toolbar with icons for 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. The 'Result Grid' is displayed below the toolbar, showing a table with the following data:

1	2	3
1	2	3

4. Write a SQL query to display the sum of two numbers 10 and 15 from the RDBMS server.

```
select 10+15 as Sum
```

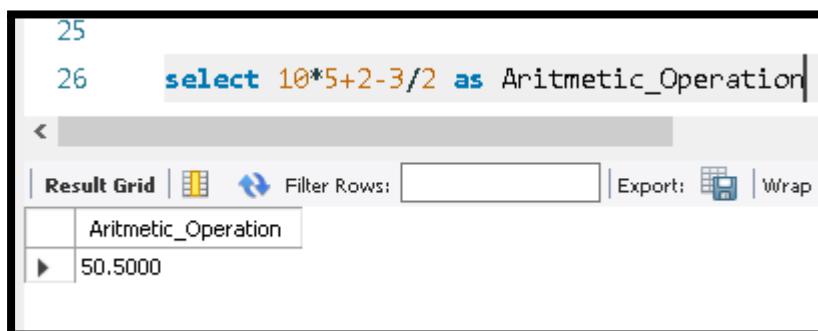


The screenshot shows a SQL query editor with the query `select 10+15 as Sum` entered on line 24. Below the query, the 'Result Grid' is displayed, showing a single row with the column header 'Sum' and the value '25'.

Sum
25

5. Write an SQL query to display the result of an arithmetic expression.

```
select 10*5+2-3/2 as Aritmetic_Operation
```

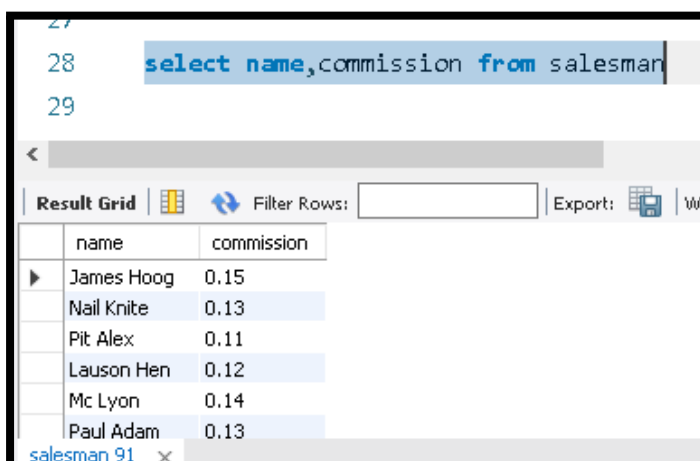


The screenshot shows a SQL query editor with the query `select 10*5+2-3/2 as Aritmetic_Operation` entered on line 26. Below the query, the 'Result Grid' is displayed, showing a single row with the column header 'Aritmetic_Operation' and the value '50.5000'.

Aritmetic_Operation
50.5000

6. Write a SQL statement to display specific columns such as names and commissions for all salespeople.

```
select name,commission from salesman
```



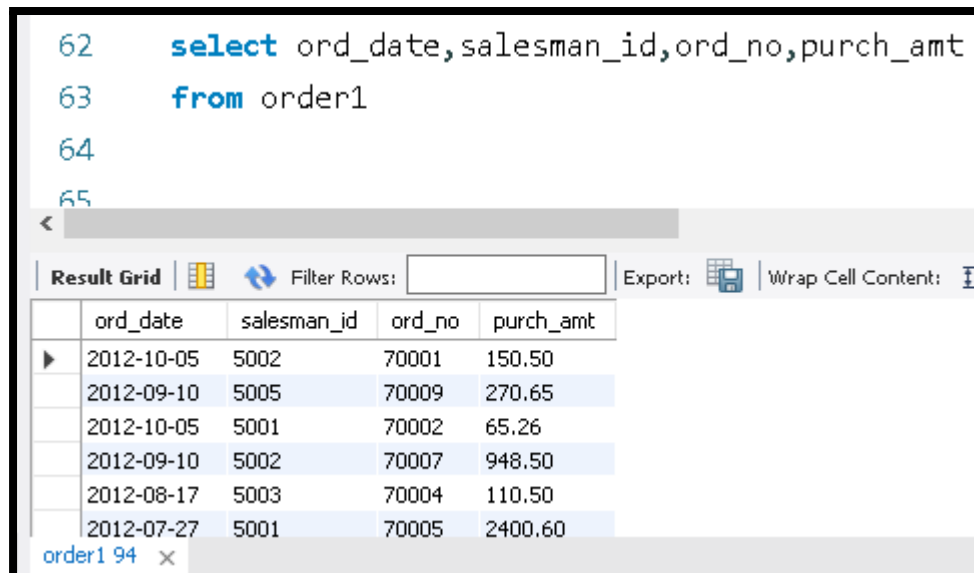
The screenshot shows a SQL query editor with the query `select name,commission from salesman` entered on line 28. Below the query, the 'Result Grid' is displayed, showing a table with two columns: 'name' and 'commission'. The table contains six rows of data.

name	commission
James Hoog	0.15
Nail Knite	0.13
Pit Alex	0.11
Lauson Hen	0.12
Mc Lyon	0.14
Paul Adam	0.13

7. Write a query to display the columns in a specific order, such as order date, salesman ID, order number, and purchase amount for all orders.

```
select ord_date,salesman_id,ord_no,purch_amt
```

```
from order1
```



The screenshot shows a SQL query editor with the following code:

```
62 select ord_date,salesman_id,ord_no,purch_amt
63 from order1
64
65
```

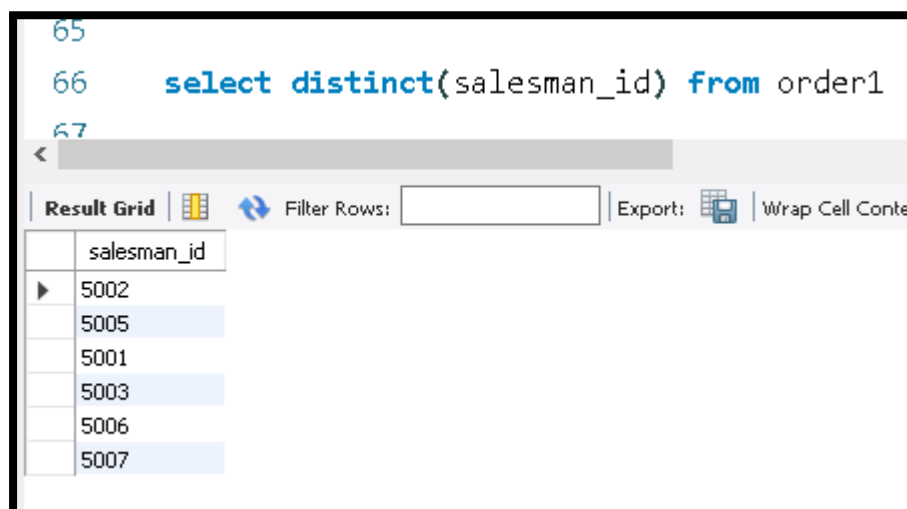
Below the code is a "Result Grid" showing the results of the query. The grid has a toolbar with "Filter Rows:", "Export:", and "Wrap Cell Content:". The results are as follows:

	ord_date	salesman_id	ord_no	purch_amt
▶	2012-10-05	5002	70001	150.50
	2012-09-10	5005	70009	270.65
	2012-10-05	5001	70002	65.26
	2012-09-10	5002	70007	948.50
	2012-08-17	5003	70004	110.50
	2012-07-27	5001	70005	2400.60

At the bottom, there is a tab labeled "order1 94" with a close button.

8. From the following table, write a SQL query to identify the unique salespeople ID. Return salesman_id.

```
select distinct(salesman_id) from order1
```



The screenshot shows a SQL query editor with the following code:

```
65
66 select distinct(salesman_id) from order1
67
```

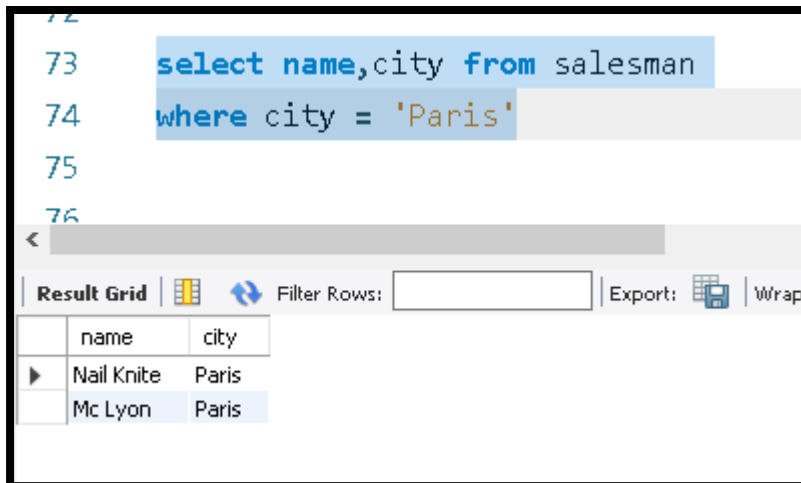
Below the code is a "Result Grid" showing the results of the query. The grid has a toolbar with "Filter Rows:", "Export:", and "Wrap Cell Conte". The results are as follows:

	salesman_id
▶	5002
	5005
	5001
	5003
	5006
	5007

9. From the following table, write a SQL query to locate salespeople who live in the city of 'Paris'. Return salesperson's name, city.

```
select name,city from salesman
```

```
where city = 'Paris'
```



The screenshot shows a SQL query editor with the following text:

```
72  
73 select name,city from salesman  
74 where city = 'Paris'  
75  
76
```

Below the editor is a 'Result Grid' with the following data:

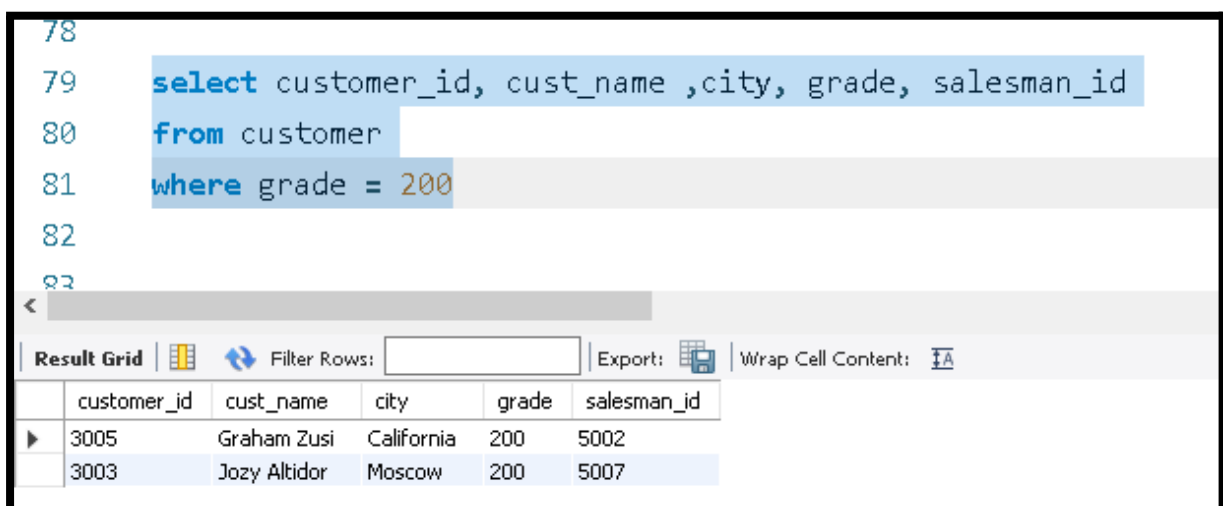
	name	city
▶	Nail Knite	Paris
	Mc Lyon	Paris

10. From the following table, write a SQL query to find customers whose grade is 200. Return customer_id, cust_name, city, grade, salesman_id.

```
select customer_id, cust_name ,city, grade, salesman_id
```

```
from customer
```

```
where grade = 200
```



The screenshot shows a SQL query editor with the following text:

```
78  
79 select customer_id, cust_name ,city, grade, salesman_id  
80 from customer  
81 where grade = 200  
82  
83
```

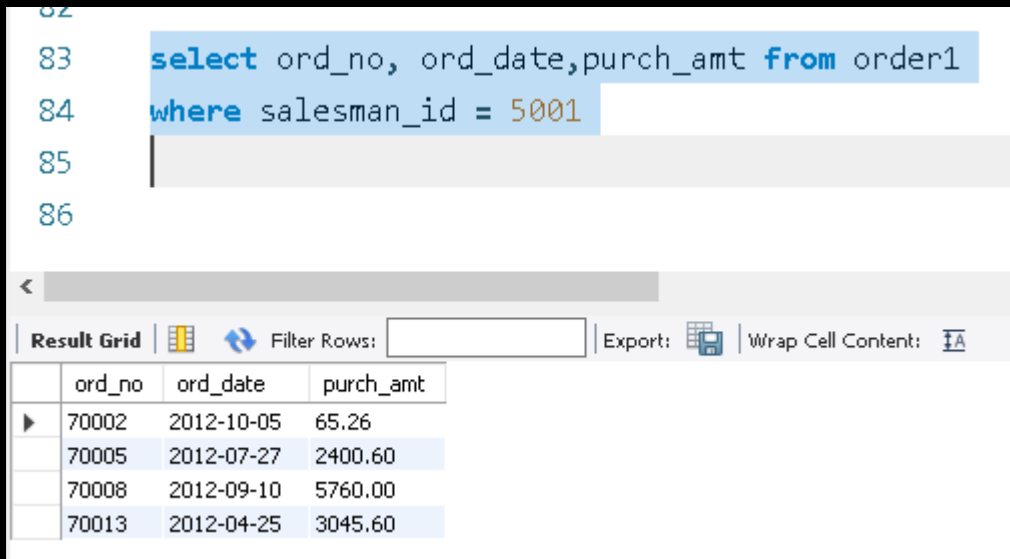
Below the editor is a 'Result Grid' with the following data:

	customer_id	cust_name	city	grade	salesman_id
▶	3005	Graham Zusi	California	200	5002
	3003	Jozy Altidor	Moscow	200	5007

11. From the following table, write a SQL query to find orders that are delivered by a salesperson with ID. 5001. Return ord_no, ord_date, purch_amt.

```
select ord_no, ord_date, purch_amt from order1
```

```
where salesman_id = 5001
```



The screenshot shows a SQL query editor with the following code:

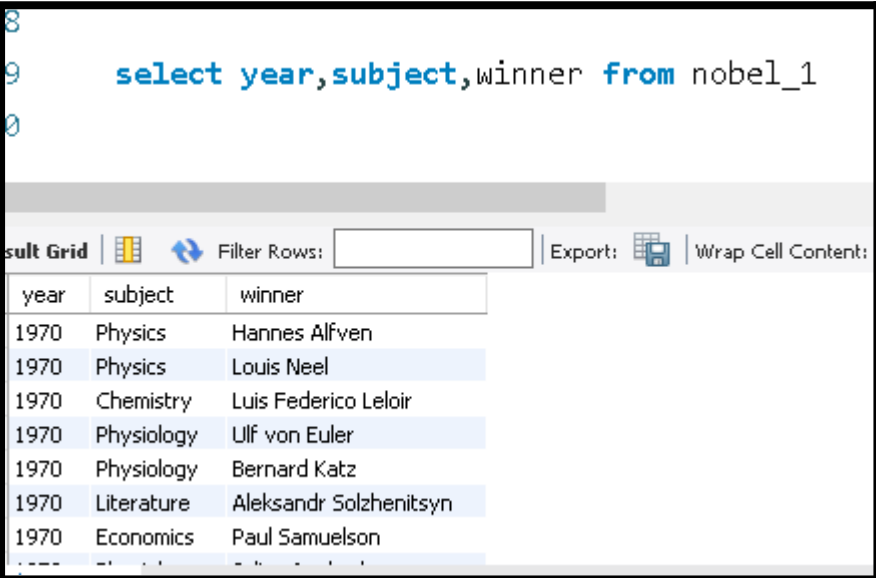
```
82  
83 select ord_no, ord_date, purch_amt from order1  
84 where salesman_id = 5001  
85  
86
```

Below the editor is a result grid with the following data:

ord_no	ord_date	purch_amt
70002	2012-10-05	65.26
70005	2012-07-27	2400.60
70008	2012-09-10	5760.00
70013	2012-04-25	3045.60

12. From the following table, write a SQL query to find the Nobel Prize winner(s) for the year 1970. Return year, subject and winner.

```
select year, subject, winner from nobel_1
```



The screenshot shows a SQL query editor with the following code:

```
8  
9 select year, subject, winner from nobel_1  
10
```

Below the editor is a result grid with the following data:

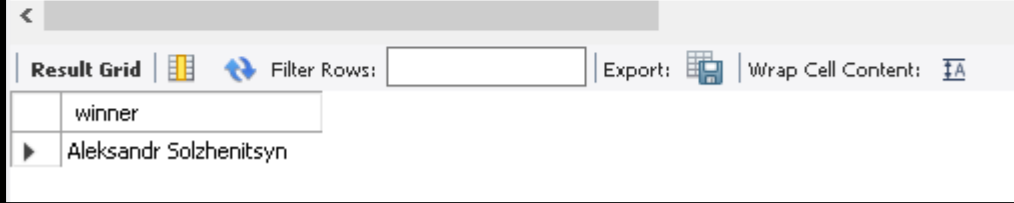
year	subject	winner
1970	Physics	Hannes Alfven
1970	Physics	Louis Neel
1970	Chemistry	Luis Federico Leloir
1970	Physiology	Ulf von Euler
1970	Physiology	Bernard Katz
1970	Literature	Aleksandr Solzhenitsyn
1970	Economics	Paul Samuelson

13. From the following table, write a SQL query to find the Nobel Prize winner in 'Literature' for 1970. Return winner.

Select winner from nobel_1

Where winner= 'Literature' and year = 1970

```
125
126 Select winner from nobel_1
127 Where subject= 'Literature' and year = 1970
128
129
```



The screenshot shows a SQL query editor with a query window and a result grid. The query window contains the following SQL query:

```
Select winner from nobel_1
Where subject= 'Literature' and year = 1970
```

The result grid shows the following data:

winner
Aleksandr Solzhenitsyn


14. From the following table, write a SQL query to locate the Nobel Prize winner 'Dennis Gabor'. Return year, subject.

Select year, subject

From nobel_1

Where winner = "Dennis Gabor"

```
129
130 Select year, subject
131 From nobel_1
132 Where winner = "Dennis Gabor"
133
134
```



The screenshot shows a SQL query editor with a query window and a result grid. The query window contains the following SQL query:

```
Select year, subject
From nobel_1
Where winner = "Dennis Gabor"
```

The result grid shows the following data:

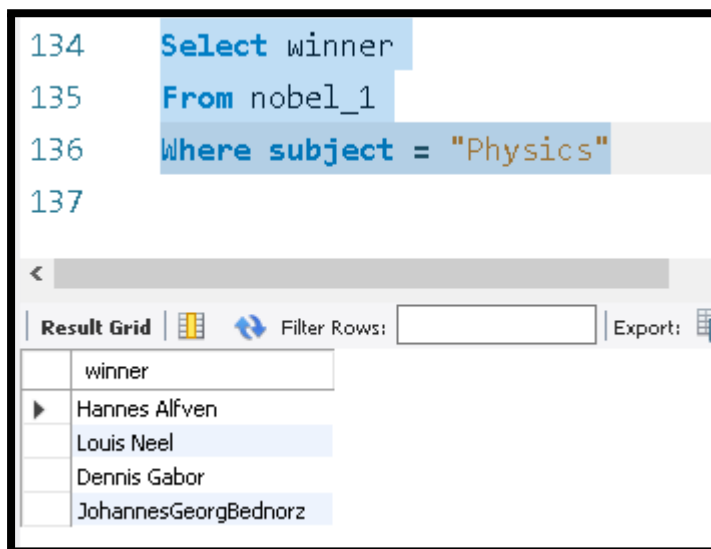
year	subject
1971	Physics

15. From the following table, write a SQL query to find the Nobel Prize winners in the field of 'Physics' since 1950. Return winner.

Select winner

From nobel_1

Where subject = "Physics"



The screenshot shows a SQL query editor with the following text:

```
134 Select winner
135 From nobel_1
136 Where subject = "Physics"
137
```

Below the editor is a 'Result Grid' showing the results of the query. The grid has a header row with the column 'winner' and four data rows with the names of the winners.

winner
Hannes Alfven
Louis Neel
Dennis Gabor
JohannesGeorgBednorz





16. From the following table, write a SQL query to find the Nobel Prize winners in 'Chemistry' between the years 1965 and 1975. Begin and end values are included. Return year, subject, winner, and country.

Select year,subject,winner,country

From nobel_1

Where subject = "Chemistry" and year between 1965 and 1975

```
137
138 Select year,subject,winner,country
139 From nobel_1
140 Where subject = "Chemistry" and year between 1965 and 1975
141
```

<   Filter Rows: Export:  Wrap Cell Content: 





	year	subject	winner	country
▶	1970	Chemistry	Luis Federico Leloir	France
	1971	Chemistry	Gerhard Herzberg	Germany

17. Write a SQL query to display all details of the Prime Ministerial winners after 1972 of Menachem Begin and Yitzhak Rabin.

Select * from nobel_1

Where category = "PrimeMinister" and year>1972

```
142 Select * from nobel_1
143 Where category = "PrimeMinister" and year>1972
144
145
```

<   Filter Rows: Export:  Wrap Cell Content: 

	YEAR	SUBJECT	WINNER	COUNTRY	CATEGORY
▶	1978	Peace	MenachemBegin	Israel	PrimeMinister
	1994	Peace	YitzhakRabin	Israel	PrimeMinister

18. From the following table, write a SQL query to retrieve the details of the winners whose first names match with the string 'Louis'. Return year, subject, winner, country, and category.

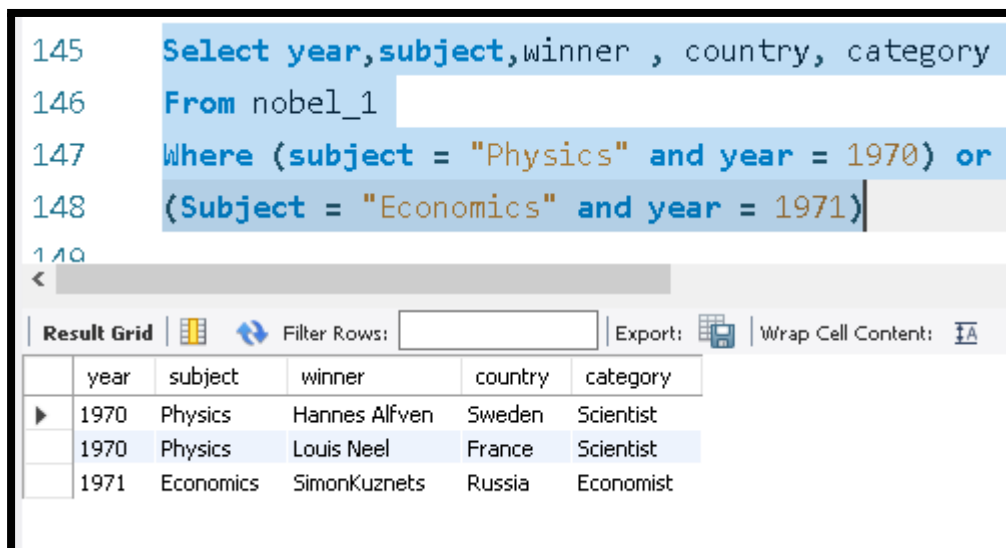
19. From the following table, write a SQL query that combines the winners in Physics, 1970 and in Economics, 1971. Return year, subject, winner, country, and category.

Select year,subject,winner , country, category

From nobel_1

Where (subject = "Physics" and year = 1970) or

(Subject = "Economics" and year = 1971)



```
145 Select year,subject,winner , country, category
146 From nobel_1
147 Where (subject = "Physics" and year = 1970) or
148 (Subject = "Economics" and year = 1971)
```

year	subject	winner	country	category
1970	Physics	Hannes Alfven	Sweden	Scientist
1970	Physics	Louis Neel	France	Scientist
1971	Economics	SimonKuznets	Russia	Economist

20. From the following table, write a SQL query to find the Nobel Prize winners in 1970 excluding the subjects of Physiology and Economics. Return year, subject, winner, country, and category.

Select year,subject,winner,country,category

From nobel_1

Where year != 1970 and subject not in ("Physiology","Economics")

```

150 Select year,subject,winner,country,category
151 From nobel_1
152 Where year != 1970 and subject not in ("Physiology","Economics")
153
154

```

Result Grid

year	subject	winner	country	category
1971	Physics	Dennis Gabor	Hungary	Scientist
1971	Chemistry	Gerhard Herzberg	Germany	Scientist
1971	Peace	Willy Brandt	Germany	Chancellor
1971	Literature	Pablo Neruda	Chile	Linguist
1978	Peace	Anwaral-Sadat	Egypt	President
1978	Peace	MenachemBegin	Israel	PrimeMinister
1987	Chemistry	DonaldJ.Cram	USA	Scientist

nobel_1 110 x

21. From the following table, write a SQL query to combine the winners in 'Physiology' before 1971 and winners in 'Peace' on or after 1974. Return year, subject, winner, country, and category.

Select year,subject,winner,country,category

From nobel_1

Where (subject = "Physiology" and year<1971) or

(subject = "Peace" and year>=1974)

```

154 Select year,subject,winner,country,category
155 From nobel_1
156 Where (subject = "Physiology" and year<1971) or
157 (subject = "Peace" and year>=1974)
158

```

Result Grid

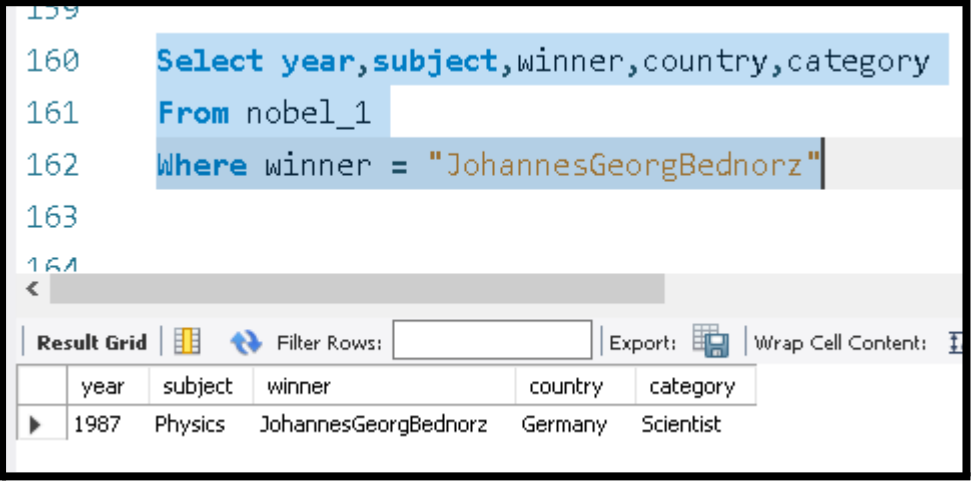
year	subject	winner	country	category
1970	Physiology	Ulf von Euler	Sweden	Scientist
1970	Physiology	Bernard Katz	Germany	Scientist
1970	Physiology	Julius Axelrod	USA	Scientist
1978	Peace	Anwaral-Sadat	Egypt	President
1978	Peace	MenachemBegin	Israel	PrimeMinister
1994	Peace	YitzhakRabin	Israel	PrimeMinister

22. From the following table, write a SQL query to find the details of the Nobel Prize winner 'Johannes Georg Bednorz'. Return year, subject, winner, country, and category.

Select year,subject,winner,country,category

From nobel_1

Where winner = "JohannesGeorgBednorz"



The screenshot shows a SQL query editor with the following text:

```
159  
160 Select year,subject,winner,country,category  
161 From nobel_1  
162 Where winner = "JohannesGeorgBednorz"  
163  
164
```

Below the editor is a 'Result Grid' with the following data:

	year	subject	winner	country	category
▶	1987	Physics	JohannesGeorgBednorz	Germany	Scientist

23. From the following table, write a SQL query to find Nobel Prize winners for the subject that does not begin with the letter 'P'. Return year, subject, winner, country, and category. Order the result by year, descending and winner in ascending.

Select year,subject,winner,country,category

From nobel_1

Where winner like "P%"

```

164  Select year,subject,winner,country,category
165  From nobel_1
166  Where winner like "P%"
167
168

```

Result Grid

	year	subject	winner	country	category
▶	1970	Economics	Paul Samuelson	USA	Economist
	1971	Literature	Pablo Neruda	Chile	Linguist

24. From the following table, write a SQL query to find the details of 1970 Nobel Prize winners. Order the results by subject, ascending except for 'Chemistry' and 'Economics' which will come at the end of the result set. Return year, subject, winner, country, and category.

Select year,subject,winner,country,category

25. From the following table, write a SQL query to select a range of products whose price is in the range Rs.200 to Rs.600. Begin and end values are included. Return pro_id, pro_name, pro_price, and pro_com.

Select pro_id, pro_name, pro_price, pro_com

From item_mast

Where pro_price between 200 and 600

```

169  Select pro_id, pro_name, pro_price, pro_com
170  From item_mast
171  Where pro_price between 200 and 600
172
173

```

Result Grid

	pro_id	pro_name	pro_price	pro_com
▶	102	Key Board	450	16
	103	ZIP drive	250	14
	104	Speaker	550	16
	109	Refill cartridge	350	13
	110	Mouse	250	12

26. From the following table, write a SQL query to calculate the average price for a manufacturer code of 16. Return avg.

Select avg(pro_price) from item_mast

where pro_cost = 16

```

173  Select avg(pro_price) from item_mast
174  where pro_com = 16
175
176
177

```

Result Grid

	avg(pro_price)
▶	500

27. From the following table, write a SQL query to display the pro_name as 'Item Name' and pro_priceas 'Price in Rs.'

Select pro_name as Item_name , pro_price as 'Price in Rs'

from item_mast

```

176 Select pro_name as Item_name , pro_price as 'Price in Rs'
177 from item_mast
178

```

Result Grid

	Item_name	Price in Rs
▶	Mother Board	3200
	Key Board	450
	ZIP drive	250
	Speaker	550
	Monitor	5000
	DVD drive	900
	CD drive	800

item_mast 118 x

28. From the following table, write a SQL query to find the items whose prices are higher than or equal to \$250. Order the result by product price in descending, then product name in ascending. Return pro_name and pro_price.

Select pro_name, pro_price

From item_mast

Where pro_price >= 250

Order by pro_price desc , pro_name

```

179 Select pro_name, pro_price
180 From item_mast
181 Where pro_price >= 250
182 Order by pro_price desc , pro_name
183
184

```

Result Grid

	pro_name	pro_price
▶	Monitor	5000
	Mother Board	3200
	Printer	2600
	DVD drive	900
	CD drive	800
	Speaker	550
	Key Board	450

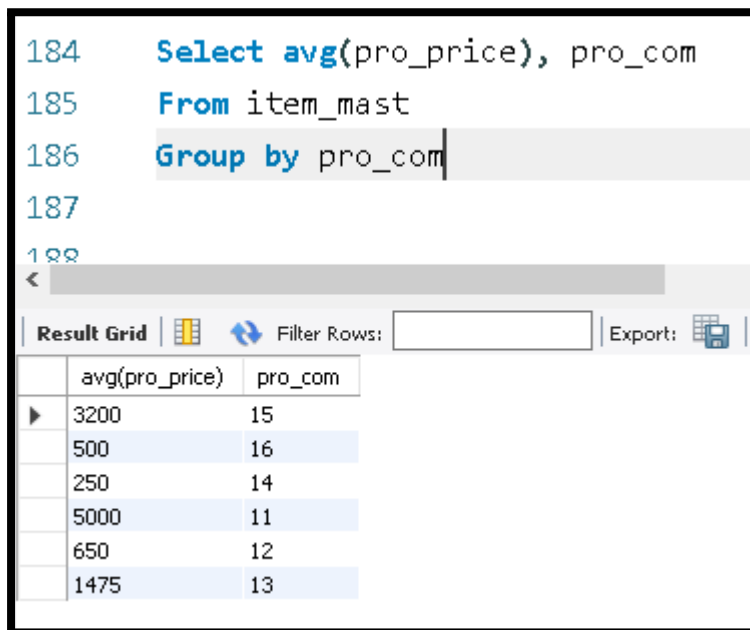
item_mast 121 x

29. From the following table, write a SQL query to calculate average price of the items for each company. Return average price and company code.

Select avg(pro_price), pro_com

From item_mast

Group by pro_com



The screenshot shows a SQL query editor with the following text:

```
184  Select avg(pro_price), pro_com
185  From item_mast
186  Group by pro_com
187
188
```

Below the editor is a 'Result Grid' showing the output of the query. The grid has two columns: 'avg(pro_price)' and 'pro_com'. The data is as follows:

avg(pro_price)	pro_com
3200	15
500	16
250	14
5000	11
650	12
1475	13

30. From the following table, write a SQL query to find the cheapest item(s). Return pro_name and, pro_price.

Select pro_name , pro_price

From item_mast

Where pro_price = (select min(pro_price) from item_mast)

```
187
188 Select pro_name , pro_price
189 From item_mast
190 Where pro_price = (select min(pro_price) from item_mast )
191
```

<

Result Grid Filter Rows: Export: Wrap Cell Content:

	pro_name	pro_price
▶	ZIP drive	250
	Mouse	250

31. From the following table, write a SQL query to find the unique last name of all employees. Return emp_lname.

Select distinct emp_lname

From emp_details

```
192 Select distinct emp_lname
193 From emp_details
194
195
196
```

<

Result Grid Filter Rows:

	emp_lname
	Sitaraman
	Gabriel
	Manuel
	Mardy
	Saule
	Snappy
	Foster

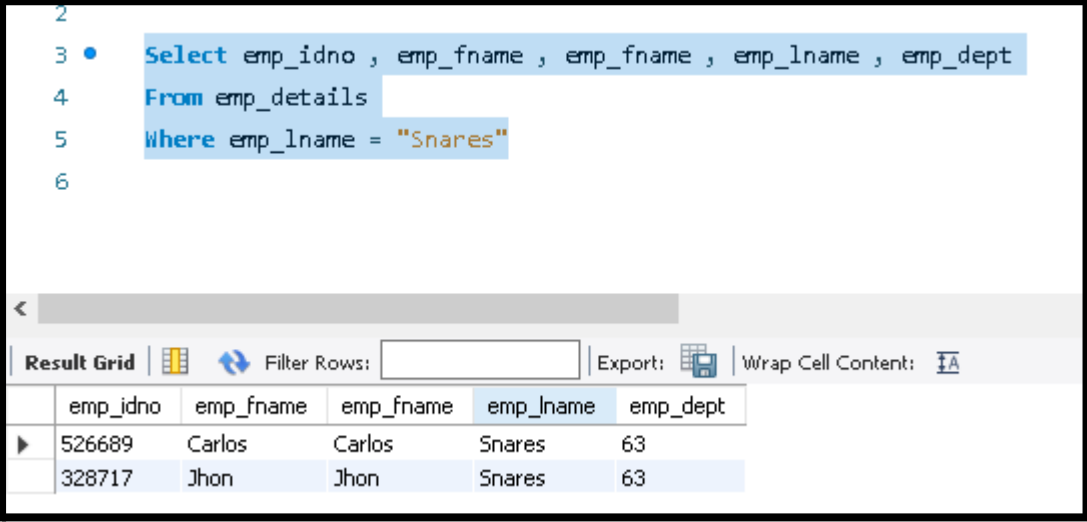
emp_details.124 x

32. From the following table, write a SQL query to find the details of employees whose last name is 'Snares'. Return emp_idno, emp_fname, emp_lname, and emp_dept.

Select emp_idno , emp_fname , emp_fname , emp_lname , emp_dept

From emp_details

Where emp_lname = "Snares"



The screenshot shows a SQL query editor with the following text:

```
2
3 • Select emp_idno , emp_fname , emp_fname , emp_lname , emp_dept
4 From emp_details
5 Where emp_lname = "Snares"
6
```

Below the editor is a toolbar with options: Result Grid, Filter Rows (with a search box), Export, and Wrap Cell Content. Below the toolbar is a table with the following data:

	emp_idno	emp_fname	emp_fname	emp_lname	emp_dept
▶	526689	Carlos	Carlos	Snares	63
	328717	Jhon	Jhon	Snares	63



33. From the following table, write a SQL query to retrieve the details of the employees who work in the department 57. Return emp_idno, emp_fname, emp_lname and emp_dept..

Select emp_idno, emp_fname, emp_lname, emp_dept

From emp_details

Where emp_dept = 57

```
6
7 • Select emp_idno, emp_fname, emp_lname, emp_dept
8 From emp_details
9 Where emp_dept = 57
10
```

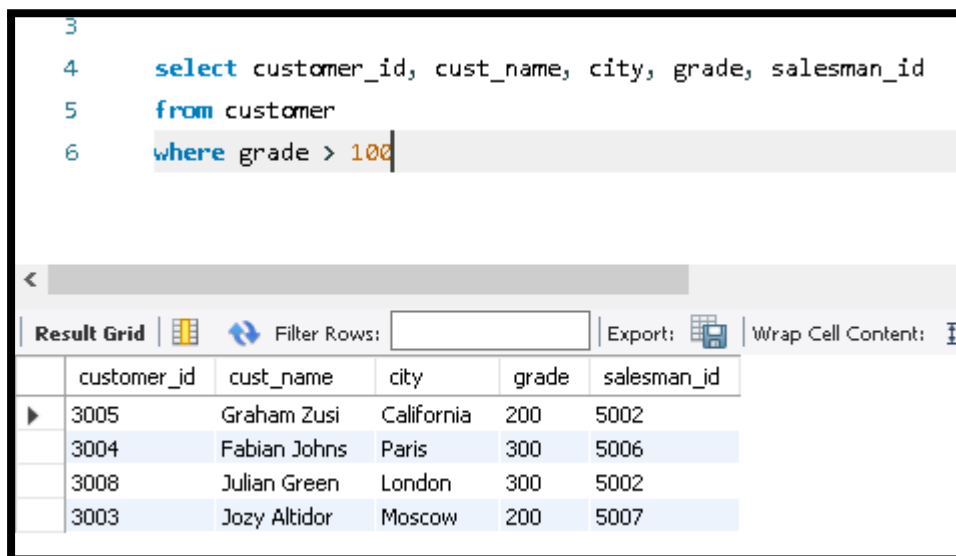
Result Grid				
Filter Rows: <input type="text"/>				
Export:  Wrap Cell Content: 				
	emp_idno	emp_fname	emp_lname	emp_dept
▶	127323	Michale	Robbin	57
	843795	Enric	Dosio	57
	847674	Kuleswar	Sitaraman	57
	555935	Alex	Manuel	57
	839139	Maria	Foster	57

1. From the following table, write a SQL query to locate the details of customers with grade values above 100. Return customer_id, cust_name, city, grade, and salesman_id.

```
select customer_id, cust_name, city, grade, salesman_id
```

```
from customer
```

```
where grade > 100
```



The screenshot shows a SQL query editor with the following code:

```
3  
4  select customer_id, cust_name, city, grade, salesman_id  
5  from customer  
6  where grade > 100
```

Below the editor is a 'Result Grid' showing the results of the query. The grid has columns: customer_id, cust_name, city, grade, and salesman_id. The results are as follows:

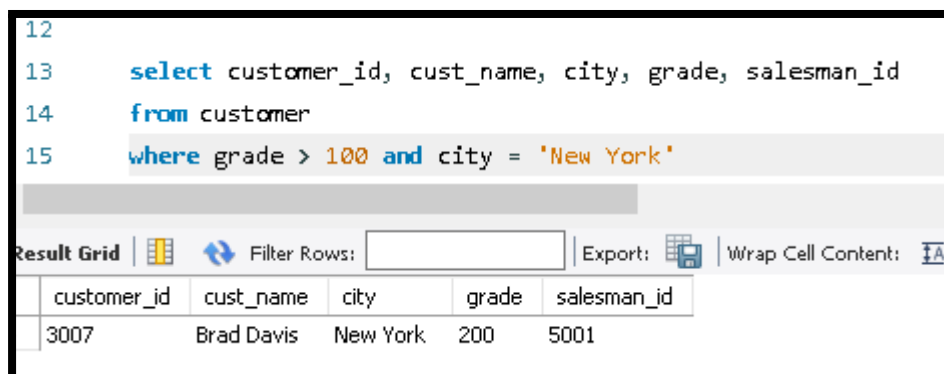
customer_id	cust_name	city	grade	salesman_id
3005	Graham Zusi	California	200	5002
3004	Fabian Johns	Paris	300	5006
3008	Julian Green	London	300	5002
3003	Jozy Altidor	Moscow	200	5007

2. From the following table, write a SQL query to find all the customers in 'New York' city who have a grade value above 100. Return customer_id, cust_name, city, grade, and salesman_id.

```
select customer_id, cust_name, city, grade, salesman_id
```

```
from customer
```

```
where grade > 100 and city = 'New York'
```



The screenshot shows a SQL query editor with the following code:

```
12  
13  select customer_id, cust_name, city, grade, salesman_id  
14  from customer  
15  where grade > 100 and city = 'New York'
```

Below the editor is a 'Result Grid' showing the results of the query. The grid has columns: customer_id, cust_name, city, grade, and salesman_id. The results are as follows:

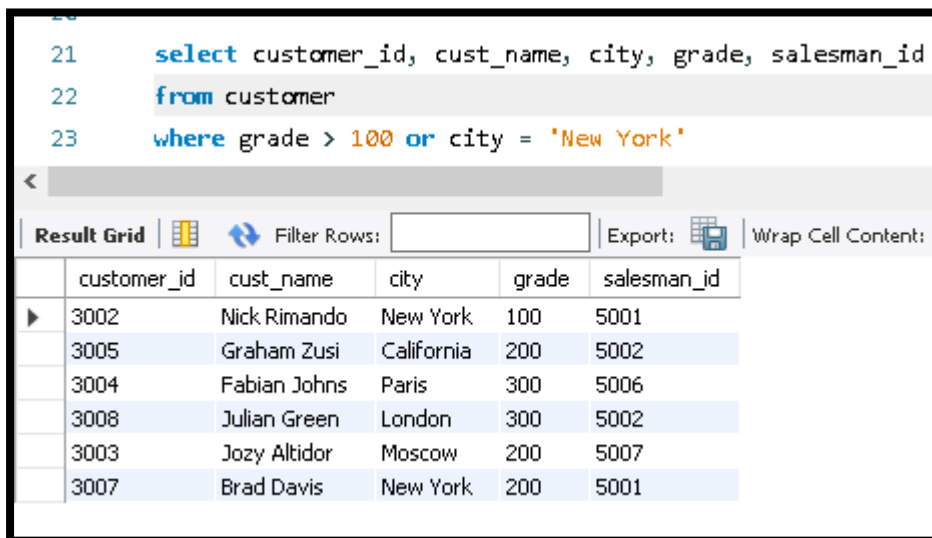
customer_id	cust_name	city	grade	salesman_id
3007	Brad Davis	New York	200	5001

3. From the following table, write a SQL query to find customers who are from the city of New York or have a grade of over 100. Return customer_id, cust_name, city, grade, and salesman_id.

```
select customer_id, cust_name, city, grade, salesman_id
```

```
from customer
```

```
where grade > 100 or city = 'New York'
```



The screenshot shows a SQL query editor with the following code:

```
21 select customer_id, cust_name, city, grade, salesman_id
22 from customer
23 where grade > 100 or city = 'New York'
```

Below the query editor is a 'Result Grid' with the following data:

	customer_id	cust_name	city	grade	salesman_id
▶	3002	Nick Rimando	New York	100	5001
	3005	Graham Zusi	California	200	5002
	3004	Fabian Johns	Paris	300	5006
	3008	Julian Green	London	300	5002
	3003	Jozy Altidor	Moscow	200	5007
	3007	Brad Davis	New York	200	5001

4. From the following table, write a SQL query to find customers who are either from the city 'New York' or who do not have a grade greater than 100. Return customer_id, cust_name, city, grade, and salesman_id.

```
select customer_id, cust_name, city, grade, salesman_id
```

```
from customer
```

```
where grade < 100 or city = 'New York'
```

```

29
30 select customer_id, cust_name, city, grade, salesman_id
31 from customer
32 where grade < 100 or city = 'New York'

```

	customer_id	cust_name	city	grade	salesman_id
▶	3002	Nick Rimando	New York	100	5001
	3001	Brad Guzan	London	0	5005
	3001	Brad Guzan	London	0	5005
	3001	Brad Guzan	London	0	5005
	3007	Brad Davis	New York	200	5001

5. From the following table, write a SQL query to identify customers who do not belong to the city of 'New York' or have a grade value that exceeds 100. Return customer_id, cust_name, city, grade, and salesman_id.

select customer_id, cust_name, city, grade, salesman_id

from customer

where grade < 100 or city != 'New York'

```

38
39 select customer_id, cust_name, city, grade, salesman_id
40 from customer
41 where grade < 100 or city != 'New York'
42
43
44

```

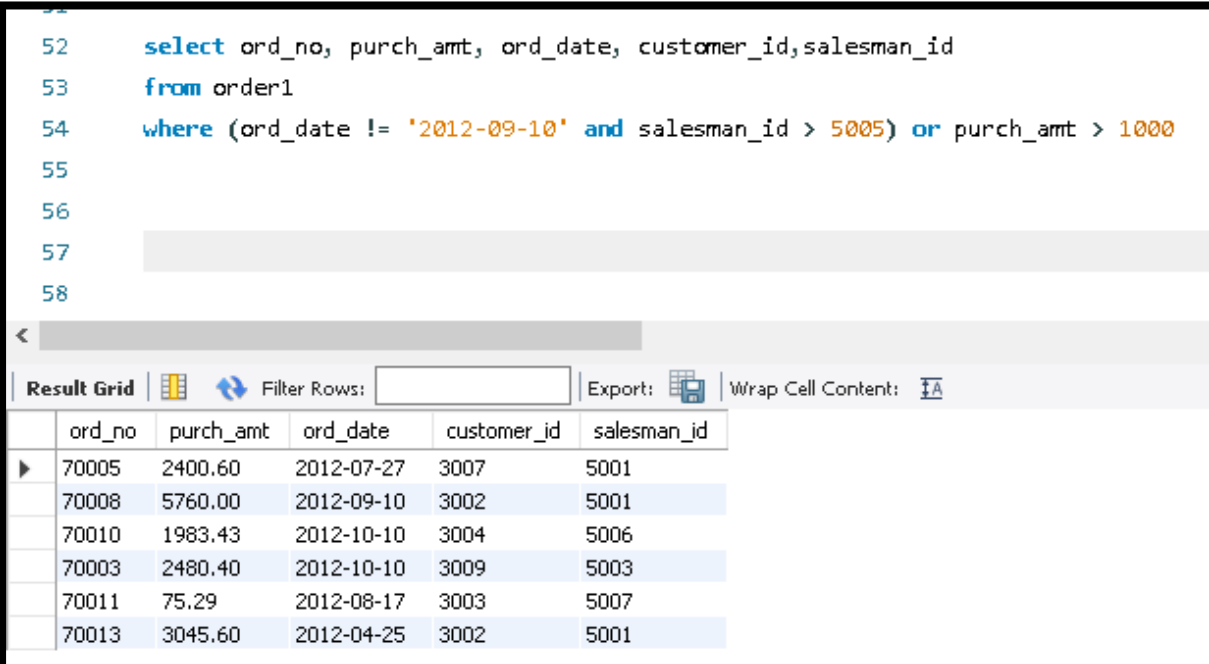
	customer_id	cust_name	city	grade	salesman_id
▶	3005	Graham Zusi	California	200	5002
	3001	Brad Guzan	London	0	5005
	3004	Fabian Johns	Paris	300	5006
	3001	Brad Guzan	London	0	5005
	3001	Brad Guzan	London	0	5005
	3009	Geoff Camero	Berlin	100	5003
	3008	Julian Green	London	300	5002
	3003	Jozy Altidor	Moscow	200	5007

6. From the following table, write a SQL query to find details of all orders excluding those with ord_date equal to '2012-09-10' and salesman_id higher than 5005 or purch_amt greater than 1000. Return ord_no, purch_amt, ord_date, customer_id and salesman_id.

```
select ord_no, purch_amt, ord_date, customer_id, salesman_id
```

```
from order1
```

```
where (ord_date != '2012-09-10' and salesman_id > 5005) or purch_amt > 1000
```



The screenshot shows a SQL query editor with the following code:

```
52 select ord_no, purch_amt, ord_date, customer_id, salesman_id
53 from order1
54 where (ord_date != '2012-09-10' and salesman_id > 5005) or purch_amt > 1000
55
56
57
58
```

Below the editor is a 'Result Grid' with the following data:

	ord_no	purch_amt	ord_date	customer_id	salesman_id
▶	70005	2400.60	2012-07-27	3007	5001
	70008	5760.00	2012-09-10	3002	5001
	70010	1983.43	2012-10-10	3004	5006
	70003	2480.40	2012-10-10	3009	5003
	70011	75.29	2012-08-17	3003	5007
	70013	3045.60	2012-04-25	3002	5001

7. From the following table, write a SQL query to find the details of those salespeople whose commissions range from 0.10 to 0.12. Return salesman_id, name, city, and commission.

```
select salesman_id, name, city, commission
```

```
from salesman
```

```
where commission between 0.10 and 0.12
```

60	
61	<code>select salesman_id, name, city, commission</code>
62	<code>from salesman</code>
63	<code>where commission between 0.10 and 0.12</code>

Result Grid				
Filter Rows: <input type="text"/>				
Export:				
	salesman_id	name	city	commission
▶	5005	Pit Alex	london	0.11
	5003	Lauson Hen		0.12

8. From the following table, write a SQL query to find details of all orders with a purchase amount less than 200 or exclude orders with an order date greater than or equal to '2012-02-10' and a customer ID less than 3009. Return ord_no, purch_amt, ord_date, customer_id and salesman_id.

select ord_no, purch_amt, ord_date, customer_id, salesman_id

from orderl

where purch_amt <200 or

NOT(ord_date >= '2012-02-10' and customer_id < 3009)

```

71
72 select ord_no, purch_amt, ord_date, customer_id, salesman_id
73 from order1
74 where purch_amt <200 or
75 NOT(ord_date >= '2012-02-10' and customer_id < 3009)
76
77

```

Result Grid

	ord_no	purch_amt	ord_date	customer_id	salesman_id
▶	70001	150.50	2012-10-05	3005	5002
	70002	65.26	2012-10-05	3002	5001
	70004	110.50	2012-08-17	3009	5003
	70003	110.50	2012-10-10	3009	5003
	70011	75.29	2012-08-17	3003	5007

9. From the following table, write a SQL query to find all orders that meet the following conditions. Exclude combinations of order date equal to '2012-08-17' or customer ID greater than 3005 and purchase amount less than 1000.




select ord_no, purch_amt, ord_date, customer_id, salesman_id

from order1

where purch_amt <1000 and

NOT(ord_date >= '2012-08-17' and customer_id > 3005)

81	
82	<code>select ord_no, purch_amt, ord_date, customer_id, salesman_id</code>
83	<code>from order1</code>
84	<code>where purch_amt <1000 and</code>
85	<code>NOT(ord_date >= '2012-08-17' and customer_id > 3005)</code>
86	
87	
88	

Result Grid		Filter Rows: <input type="text"/>	Export: 	Wrap Cell Content: 
ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.50	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70007	948.50	2012-09-10	3005	5002
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007

10. Write a SQL query that displays order number, purchase amount, and the achieved and unachieved percentage (%) for those orders that exceed 50% of the target value of 6000.

SELECT ord_no,purch_amt,

(100*purch_amt)/6000 AS "Achieved %",

(100*(6000-purch_amt)/6000) AS "Unachieved %"

FROM order1

WHERE (100*purch_amt)/6000>50;

```
93
94 SELECT ord_no,purch_amt,
95 (100*purch_amt)/6000 AS "Achieved %",
96 (100*(6000-purch_amt)/6000) AS "Unachieved %"
97 FROM order1
```

ord_no	purch_amt	Achieved %	Unachieved %
70008	5760.00	96.000000	4.000000
70013	3045.60	50.760000	49.240000

11. From the following table, write a SQL query to find the details of all employees whose last name is 'Dosni' or 'Mardy'. Return emp_idno, emp_fname, emp_lname, and emp_dept.

select emp_idno, emp_fname, emp_lname, emp_dept

from emp_details

where emp_lname = "Dosni" or emp_lname = "Mardy"

```
106
107 select emp_idno, emp_fname, emp_lname, emp_dept
108 from emp_details
109 where emp_lname = "Dosni" or emp_lname = "Mardy"
110
111
112
113
```

emp_idno	emp_fname	emp_lname	emp_dept
444527	Joseph	Dosni	47
539569	George	Mardy	27

12. From the following table, write a SQL query to find the employees who work at depart 47 or 63. Return emp_idno, emp_fname, emp_lname, and emp_dept.

```
select emp_idno, emp_fname, emp_lname, emp_dept
```

```
from emp_details
```

```
where emp_dept in (47,63)
```

```
115
116 select emp_idno, emp_fname, emp_lname, emp_dept
117 from emp_details
118 where emp_dept in (47,63)
119
120
```

	emp_idno	emp_fname	emp_lname	emp_dept
▶	526689	Carlos	Snares	63
	328717	Jhon	Snares	63
	444527	Joseph	Dosni	47
	659831	Zanifer	Emily	47
	748681	Henrey	Gabriel	47
	733843	Mario	Saule	63