HR

DEPARTMENTS
department_id
department_name
manager_id
location_id

LOCATIONS
location_id
street_address
postal_code
city
state_province
country_id

JOB_HISTORY
employee_id
start_date
end_date
job_id
department_id

EMPLOYEES
employee_id
first_name
last_name
email
phone_number
hire_date
job_id
salary
commission_pct
manager_id
department_id

COUNTRIES
country_id
country_name
region_id

JOBS
job_id
job_title
min_salary
max_salary

REGIONS
region_id
region_name

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

```
1 •   use humanresource;
2 •   select * from employees where job_id = 'SA_REP' or job_id = 'SA_MAN' ;
```

Result Grid | 🔢 | 🔄 Filter Rows: [   ] | Edit: 🖋 📲 📲 | Export/Import: 📇 📇 | Wrap Cell Content: 🔤

employee_id	first_name	last_name	email	phone_number	hire_date	job_id	salary	commission_pct	manager_id	department_id
▶ 145	John	Russell	JRUSSEL	011.44.1344.429268	1996-10-01	SA_MAN	14000.00	0.40	100	80
146	Karen	Partners	KPARTNER	011.44.1344.467268	1997-01-05	SA_MAN	13500.00	0.30	100	80
147	Alberto	Errazuriz	AERRAZUR	011.44.1344.429278	1997-03-10	SA_MAN	12000.00	0.30	100	80
148	Gerald	Cambrault	GCAMBRAU	011.44.1344.619268	1999-10-15	SA_MAN	11000.00	0.30	100	80

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

```
1 •   use humanresource;
2 •   select "This is SQL Exercise, Practice and Solution" as ' ';
```

Result Grid | ⊞ ⟲ Filter Rows: | Export: ⊞ | Wrap Cell Content: 🅐
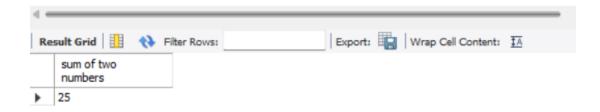
▶   This is SQL Exercise, Practice and Solution

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

```
1 •   use humanresource;
2 •   select 420 as col1, 820 as col2, 99 as col3;
```
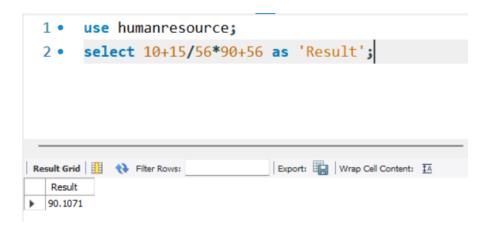
Result Grid | ⊞ ⟲ Filter Rows: | Export: ⊞ | Wrap Cell Content: 🅐

col1	col2	col3
420	820	99

4. Write a SQL query to display the sum of two numbers 10 and 15 from the RDBMS server.
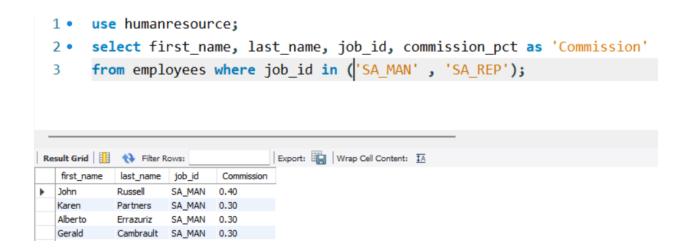
```
1 •   use humanresource;
2 •   select 10+15 as 'sum of two numbers';
```
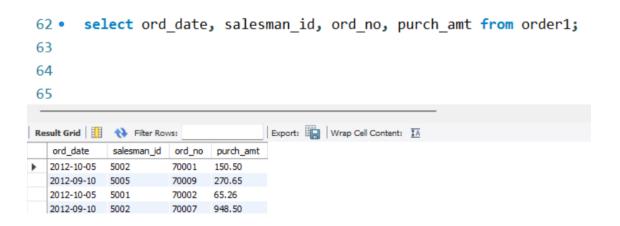
| Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

sum of two numbers
25

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

```
1 •   use humanresource;
2 •   select 10+15/56*90+56 as 'Result';
```

| Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

Result
90.1071
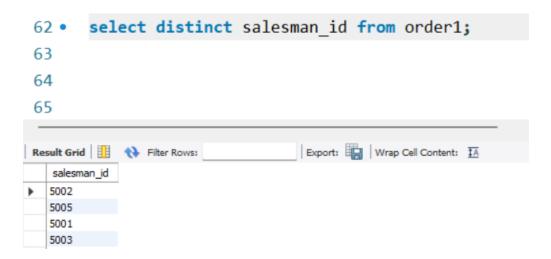
6. Write a SQL statement to display specific columns such as names and commissions for all salespeople.

```
1 •  use humanresource;
2 •  select first_name, last_name, job_id, commission_pct as 'Commission'
3     from employees where job_id in ('SA_MAN' , 'SA_REP');
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

first_name	last_name	job_id	Commission
John	Russell	SA_MAN	0.40
Karen	Partners	SA_MAN	0.30
Alberto	Errazuriz	SA_MAN	0.30
Gerald	Cambrault	SA_MAN	0.30

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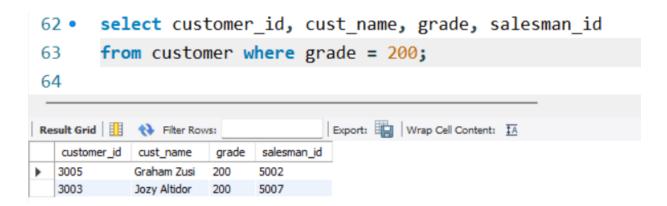
```
62 •  select ord_date, salesman_id, ord_no, purch_amt from order1;
63
64
65
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

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

```
62 •    select distinct salesman_id from order1;
63
64
65
```

salesman_id
5002
5005
5001
5003

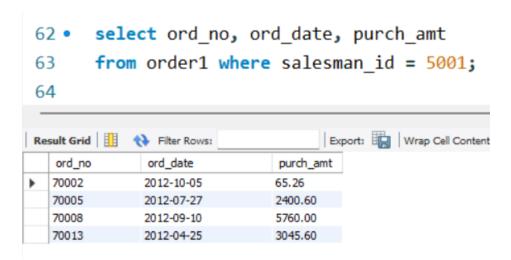**9.** From the following table, write a SQL query to locate salespeople who live in the city of 'Paris'. Return salesperson's name, city.

```
62 •    select name from salesman where city = 'paris';
63
```
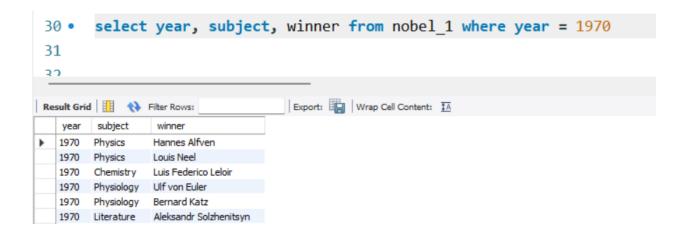
name
Nail Knite
Mc Lyon

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.

```
62 •    select customer_id, cust_name, grade, salesman_id
63       from customer where grade = 200;
64
```

customer_id	cust_name	grade	salesman_id
3005	Graham Zusi	200	5002
3003	Jozy Altidor	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.
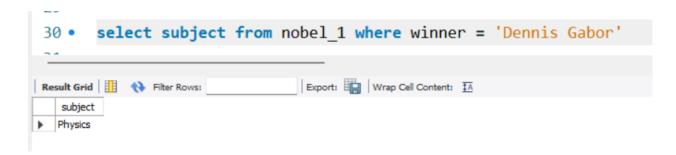
```
62 •   select ord_no, ord_date, purch_amt
63     from order1 where salesman_id = 5001;
64
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content

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.

```
30 •   select year, subject, winner from nobel_1 where year = 1970
31
32
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

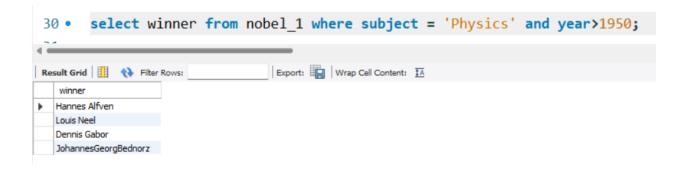| year	subject	winner
1970	Physics	Hannes Alfven
1970	Physics	Louis Neel
1970	Chemistry	Luis Federico Leloir
1970	Physiology	Ulf von Euler
1970	Physiology	Bernard Katz
1970	Literature	Aleksandr Solzhenitsyn

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

```
30 •   select year, subject, winner
31      from nobel_1 where year = 1970 and category = 'Literature'
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:

year	subject	winner

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

```
30 •   select subject from nobel_1 where winner = 'Dennis Gabor'
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:

subject
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.

```
30 •   select winner from nobel_1 where subject = 'Physics' and year>1950;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:

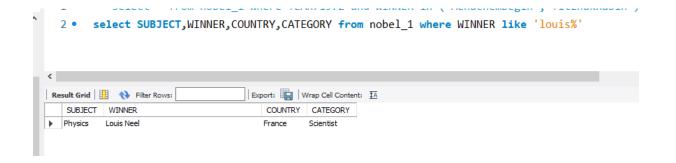| winner |
|---|
| Hannes Alfven |
| Louis Neel |
| Dennis Gabor |
| JohannesGeorgBednorz |

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.

```
30 •   select subject, winner, country from nobel_1 where subject = 'Chemistry' and
31      (year>=1965 and year<=1975)
```

subject	winner	country
Chemistry	Luis Federico Leloir	France
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.

```
1 •   select * from nobel_1 where YEAR>1972 and WINNER in ('MenachemBegin','YitzhakRabin');
2
```

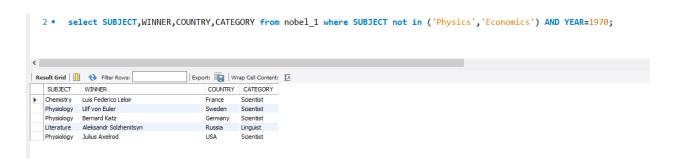| YEAR	SUBJECT	WINNER	COUNTRY	CATEGORY
1978	Peace	MenachemBegin	Israel	PrimeMinister
1994	Peace	YitzhakRabin	Israel	PrimeMinister

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.

```
2 •   select SUBJECT,WINNER,COUNTRY,CATEGORY from nobel_1 where WINNER like 'louis%'
```

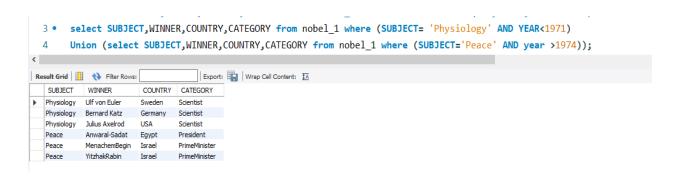| SUBJECT	WINNER	COUNTRY	CATEGORY
Physics	Louis Neel	France	Scientist

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.

```
2 •   select SUBJECT,WINNER,COUNTRY,CATEGORY from nobel_1 where SUBJECT in ('Physics','Economics') AND YEAR in (1970,1971);
```

SUBJECT	WINNER	COUNTRY	CATEGORY
Physics	Hannes Alfven	Sweden	Scientist
Physics	Louis Neel	France	Scientist
Economics	Paul Samuelson	USA	Economist
Physics	Dennis Gabor	Hungary	Scientist
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.
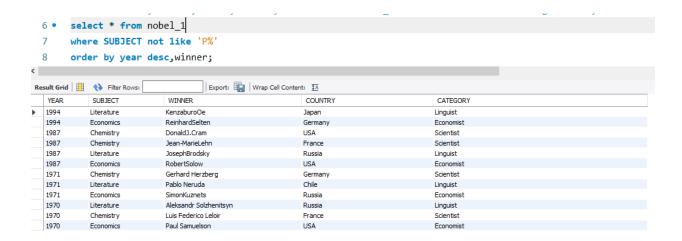
```
2 •   select SUBJECT,WINNER,COUNTRY,CATEGORY from nobel_1 where SUBJECT not in ('Physics','Economics') AND YEAR=1970;
```

SUBJECT	WINNER	COUNTRY	CATEGORY
Chemistry	Luis Federico Leloir	France	Scientist
Physiology	Ulf von Euler	Sweden	Scientist
Physiology	Bernard Katz	Germany	Scientist
Literature	Aleksandr Solzhenitsyn	Russia	Linguist
Physiology	Julius Axelrod	USA	Scientist

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.

```
3 •   select SUBJECT,WINNER,COUNTRY,CATEGORY from nobel_1 where (SUBJECT= 'Physiology' AND YEAR<1971)
4     Union (select SUBJECT,WINNER,COUNTRY,CATEGORY from nobel_1 where (SUBJECT='Peace' AND year >1974));
```

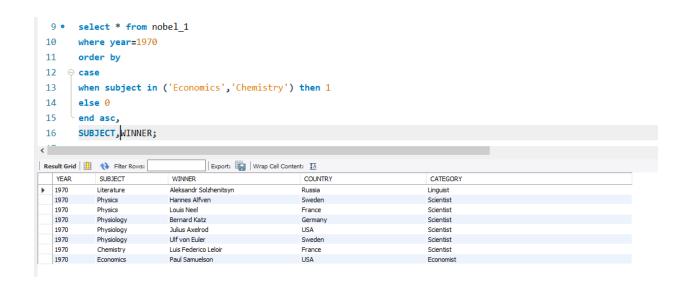| SUBJECT	WINNER	COUNTRY	CATEGORY
Physiology	Ulf von Euler	Sweden	Scientist
Physiology	Bernard Katz	Germany	Scientist
Physiology	Julius Axelrod	USA	Scientist
Peace	Anwaral-Sadat	Egypt	President
Peace	MenachemBegin	Israel	PrimeMinister
Peace	YitzhakRabin	Israel	PrimeMinister

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.
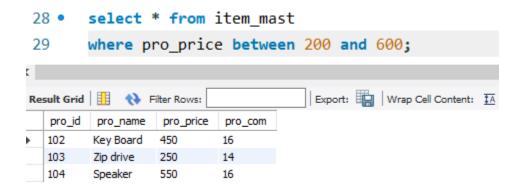
```
5 •   select YEAR,SUBJECT,WINNER,COUNTRY,CATEGORY from nobel_1 where WINNER='JohannesGeorgBednorz';
```

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.

```
6 •   select * from nobel_1
7     where SUBJECT not like 'P%'
8     order by year desc,winner;
```

YEAR	SUBJECT	WINNER	COUNTRY	CATEGORY
1994	Literature	KenzaburoOe	Japan	Linguist
1994	Economics	ReinhardSelten	Germany	Economist
1987	Chemistry	DonaldJ.Cram	USA	Scientist
1987	Chemistry	Jean-MarieLehn	France	Scientist
1987	Literature	JosephBrodsky	Russia	Linguist
1987	Economics	RobertSolow	USA	Economist
1971	Chemistry	Gerhard Herzberg	Germany	Scientist
1971	Literature	Pablo Neruda	Chile	Linguist
1971	Economics	SimonKuznets	Russia	Economist
1970	Literature	Aleksandr Solzhenitsyn	Russia	Linguist
1970	Chemistry	Luis Federico Leloir	France	Scientist
1970	Economics	Paul Samuelson	USA	Economist

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.

```
 9 •   select * from nobel_1
10     where year=1970
11     order by
12  ⊖  case
13     when subject in ('Economics','Chemistry') then 1
14     else 0
15     end asc,
16     SUBJECT,WINNER;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: 𝐼A

YEAR	SUBJECT	WINNER	COUNTRY	CATEGORY
1970	Literature	Aleksandr Solzhenitsyn	Russia	Linguist
1970	Physics	Hannes Alfven	Sweden	Scientist
1970	Physics	Louis Neel	France	Scientist
1970	Physiology	Bernard Katz	Germany	Scientist
1970	Physiology	Julius Axelrod	USA	Scientist
1970	Physiology	Ulf von Euler	Sweden	Scientist
1970	Chemistry	Luis Federico Leloir	France	Scientist
1970	Economics	Paul Samuelson	USA	Economist

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.
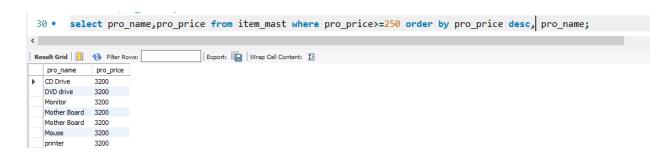
```
28 •   select * from item_mast
29     where pro_price between 200 and 600;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: 𝐼A

pro_id	pro_name	pro_price	pro_com
102	Key Board	450	16
103	Zip drive	250	14
104	Speaker	550	16

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

```
28 •    select avg(pro_price) from item_mast
29       where pro_com=16;
```

Result Grid | 🔢 ↔ Filter Rows: [   ] | Export: 💾 | Wrap Cell Content: 🛱

avg(pro_price)
▶ 500.0000

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

```
30 •    SELECT pro_name as "Item Name", pro_price AS "Price in Rs."
31          FROM item_mast;
```

Result Grid | 🔢 ↔ Filter Rows: [   ] | Export: 💾 | Wrap Cell Content: 🛱

Item Name	Price in Rs.
▶ Mother Board	3200
Mother Board	3200
Key Board	450
Zip drive	250
Speaker	550
Monitor	3200
DVD drive	3200
CD Drive	3200
printer	3200
Refill cartrige	3200
Mouse	3200

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.

```
30 •    select pro_name,pro_price from item_mast where pro_price>=250 order by pro_price desc, pro_name;
```

Result Grid | 🔢 ↔ Filter Rows: [   ] | Export: 💾 | Wrap Cell Content: 🛱

pro_name	pro_price
▶ CD Drive	3200
DVD drive	3200
Monitor	3200
Mother Board	3200
Mother Board	3200
Mouse	3200
printer	3200

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.
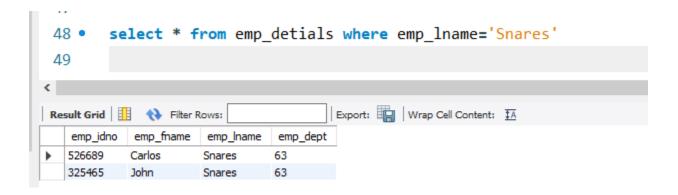
```
31 •   select avg(pro_price) as 'average',pro_com from item_mast group by pro_com;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

average	pro_com
3200.0000	15
500.0000	16
250.0000	14
3200.0000	11
3200.0000	12
3200.0000	13

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

```
32 •   select pro_price,pro_name from item_mast where pro_price=(select min(pro_price) from item_mast)
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

pro_price	pro_name
250	Zip drive

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

47

48 • `select distinct emp_lname from emp_detials;`

<

Result Grid | Filter Rows: [ ] | Export: | Wrap Cell Content: A

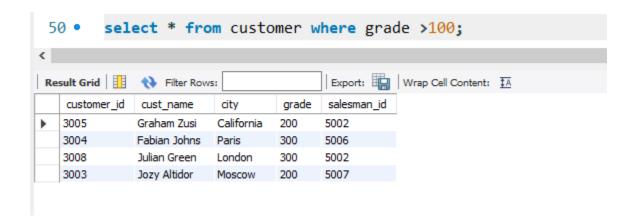| emp_lname |
| --- |
| ▶ Robbin |
| Snares |
| Dosio |
| Dosni |
| Emily |
| Sitaraman |
| Gabriel |

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.
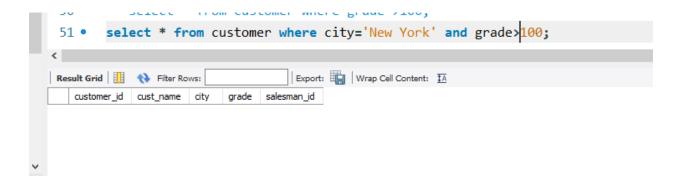
48 • `select * from emp_detials where emp_lname='Snares'`

49

<

Result Grid | Filter Rows: [ ] | Export: | Wrap Cell Content: A

emp_idno	emp_fname	emp_lname	emp_dept
▶ 526689	Carlos	Snares	63
325465	John	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..

49 • `select * from emp_detials where emp_dept=57;`

50

<

Result Grid | Filter Rows: [ ] | Export: | Wrap Cell Content: A

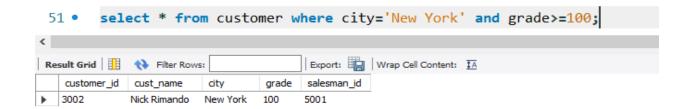| emp_idno	emp_fname	emp_lname	emp_dept
▶ 127323	Michale	Robbin	57
832983	Enric	Dosio	57
1273423423	Kuleswar	Sitaraman	57

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.
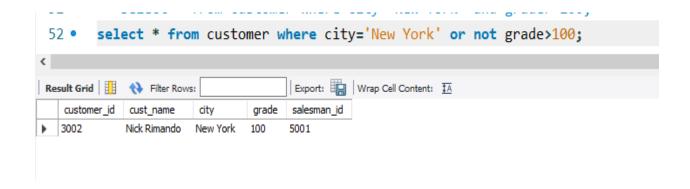
```
50 •   select * from customer where grade >100;
```

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

Result Grid | Filter Rows: | Export: | Wrap Cell Content: $\overline{I}A$

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.

```
51 •   select * from customer where city='New York' and grade>100;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: $\overline{I}A$

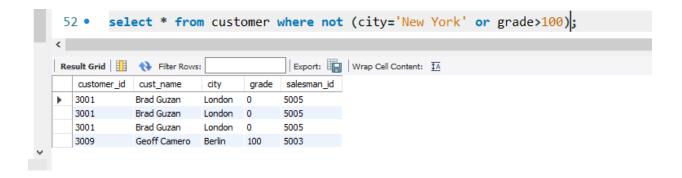| customer_id	cust_name	city	grade	salesman_id

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.

```
51 •   select * from customer where city='New York' and grade>=100;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: $\overline{I}A$

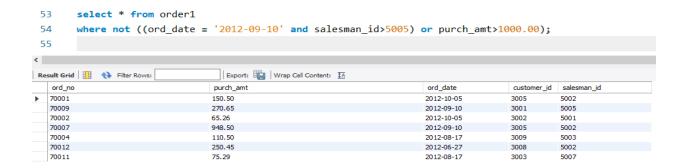| customer_id	cust_name	city	grade	salesman_id
▶ 3002	Nick Rimando	New York	100	5001

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.

```
52 •   select * from customer where city='New York' or not grade>100;
```

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	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.

```
52 •   select * from customer where not (city='New York' or grade>100);
```

customer_id	cust_name	city	grade	salesman_id
3001	Brad Guzan	London	0	5005
3001	Brad Guzan	London	0	5005
3001	Brad Guzan	London	0	5005
3009	Geoff Camero	Berlin	100	5003

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.
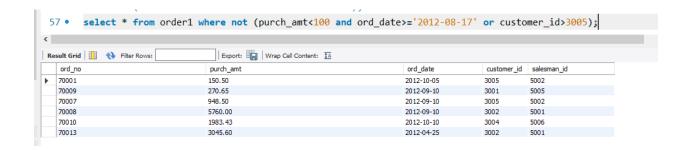
```
53    select * from order1
54    where not ((ord_date = '2012-09-10' and salesman_id>5005) or purch_amt>1000.00);
55
```

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
70004	110.50	2012-08-17	3009	5003
70012	250.45	2012-06-27	3008	5002
70011	75.29	2012-08-17	3003	5007

7. From the following table, write a SQL query to find the details of those salespeople whose commissions range from 0.10 to0.12. Return salesman_id, name, city, and commission.

```
56    where (commission > 0.10 and commission < 0.12);
```

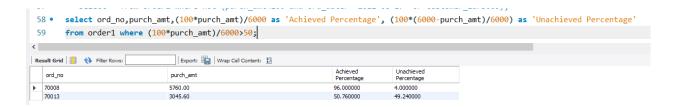| salesman_id	name	city	commission
5005	Pit Alex	london	0.11

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.

```
57 •  select * from orders where(purch_amt<200 or not(ord_date>='2012-02-10' and customer_id<3009));
58
```

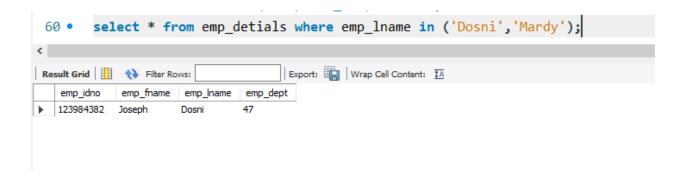| salesman_id	name	city	commission
5005	Pit Alex	london	0.11

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.

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
70007	948.50	2012-09-10	3005	5002
70008	5760.00	2012-09-10	3002	5001
70010	1983.43	2012-10-10	3004	5006
70013	3045.60	2012-04-25	3002	5001

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.

58 • select ord_no,purch_amt,(100*purch_amt)/6000 as 'Achieved Percentage', (100*(6000-purch_amt)/6000) as 'Unachieved Percentage'
59   from order1 where (100*purch_amt)/6000>50;

ord_no	purch_amt	Achieved Percentage	Unachieved Percentage
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.

60 • select * from emp_detials where emp_lname in ('Dosni','Mardy');

emp_idno	emp_fname	emp_lname	emp_dept
123984382	Joseph	Dosni	47

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.

```sql
61 •    select * from emp_detials where emp_dept=47 or emp_dept=63;
```

emp_idno	emp_fname	emp_lname	emp_dept
526689	Carlos	Snares	63
325465	John	Snares	63
123984382	Joseph	Dosni	47
127324323	Zanifer	Emily	47
127234323	Henrey	Gabriel	47

Result Grid | Filter Rows: | Export: | Wrap Cell Content: