

1. Write an SQL query to print all Worker details from the Worker table order by FIRST\_NAME Ascending and DEPARTMENT Descending.

### Ascending

The screenshot shows the SQL Developer interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL Editor contains the following queries:

```
22 (7,'Satish','Kumar',75000,'2014-01-20 9:00','Account'),
23 (8,'Geetika','Chauhan',90000,'2014-04-11 9:00','Admin');
24
25 • select *from workers;
26
27 • select * from workers
28   order by first_name asc;
29
30 • select * from workers
31   order by department desc;
32
```

The Results pane shows the output of the second query, ordered by first\_name ascending. The table has 6 columns: worker\_id, first\_name, last\_name, salary, joining\_date, and department.

worker_id	first_name	last_name	salary	joining_date	department
4	Amitabh	Singh	500000	2014-02-20 09:00:00	Admin
8	Geetika	Chauhan	90000	2014-04-11 09:00:00	Admin
1	Monika	Arora	100000	2014-02-20 09:00:00	HR
2	Nihanika	Verma	80000	2014-06-11 09:00:00	Admin
7	Satish	Kumar	75000	2014-01-20 09:00:00	Account
9	Shruti	Lad	100000	2014-04-11 09:00:00	HR
6	Vipul	Diwan	200000	2014-06-11 09:00:00	Account

### descending

The screenshot shows the SQL Developer interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL Editor contains the following queries:

```
22 (7,'Satish','Kumar',75000,'2014-01-20 9:00','Account'),
23 (8,'Geetika','Chauhan',90000,'2014-04-11 9:00','Admin');
24
25 • select *from workers;
26
27 • select * from workers
28   order by first_name asc;
29
30 • select * from workers
31   order by department desc;
32
```

The Results pane shows the output of the second query, ordered by first\_name ascending. The table has 6 columns: worker\_id, first\_name, last\_name, salary, joining\_date, and department.

worker_id	first_name	last_name	salary	joining_date	department
1	Monika	Arora	100000	2014-02-20 09:00:00	HR
3	Vishal	Singhal	300000	2014-02-20 09:00:00	HR
9	Shruti	Lad	100000	2014-04-11 09:00:00	HR
2	Nihanika	Verma	80000	2014-06-11 09:00:00	Admin
4	Amitabh	Singh	500000	2014-02-20 09:00:00	Admin
5	Vivek	Bhati	500000	2014-06-11 09:00:00	Admin
8	Geetika	Chauhan	90000	2014-04-11 09:00:00	Admin

2. Write an SQL query to print details for Workers with the first names from the Worker table. "Vipul" and "Satish"

<

3. Write an SQL query to print details of the Workers whose FIRST\_NAME ends with 'h' and contains six alphabets.

Limit to 1000 rows

```

34  where first_name in ('Vipul','Satish');
35
36  •  select * from workers
37  where first_name like '____h';
38
39
40
41

```

Result Grid

Filter Rows:

Edit:

Export/Import:

Wrap Cell Content:

	worker_id	first_name	last_name	salary	joining_date	department
▶	7	Satish	Kumar	75000	2014-01-20 09:00:00	Account
*	NULL	NULL	NULL	NULL	NULL	NULL

4. Write an SQL query to print details of the Workers whose SALARY lies between 1.

Limit to 1000 rows

```

39 • select * from workers
40   where salary between 1 and 100000
41
42
43
44
45
46

```

Result Grid

worker_id	first_name	last_name	salary	joining_date	department
1	Monika	Arora	100000	2014-02-20 09:00:00	HR
2	Nihanrika	Verma	80000	2014-06-11 09:00:00	Admin
7	Satish	Kumar	75000	2014-01-20 09:00:00	Account
8	Geetika	Chauhan	90000	2014-04-11 09:00:00	Admin
9	Shruti	Lad	100000	2014-04-11 09:00:00	HR
NULL	NULL	NULL	NULL	NULL	NULL

5. Write an SQL query to fetch duplicate records having matching data in some fields of a table.

Limit to 1000 rows

```

41
42 • select salary,department,count(*) from workers
43   group by salary,department
44   having count(*)>1;
45
46
47
48

```

Result Grid

salary	department	count(*)
100000	HR	2
500000	Admin	2

6. Write an SQL query to show the top 6 records of a table.

Limit to 1000 rows

```

45
46 • select * from workers
47   limit 6;
48
49
50
51
52

```

Result Grid

worker_id	first_name	last_name	salary	joining_date	department
1	Monika	Arora	100000	2014-02-20 09:00:00	HR
2	Nihanrika	Verma	80000	2014-06-11 09:00:00	Admin
3	Vishal	Singhal	300000	2014-02-20 09:00:00	HR
4	Amitabh	Singh	500000	2014-02-20 09:00:00	Admin
5	Vivek	Bhati	500000	2014-06-11 09:00:00	Admin
6	Vipul	Diwan	200000	2014-06-11 09:00:00	Account
NULL	NULL	NULL	NULL	NULL	NULL

Delete selected rows

7. Write an SQL query to fetch the departments that have less than five people in them.

The screenshot shows a SQL IDE interface. The query editor contains the following SQL query:

```
48
49 • select department, count(*) as people from workers
50   group by department
51   having count(*) <5;
52
53
54
55
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query in a table:

department	people
HR	3
Admin	4
Account	2

8. Write an SQL query to show all departments along with the number of people in there.

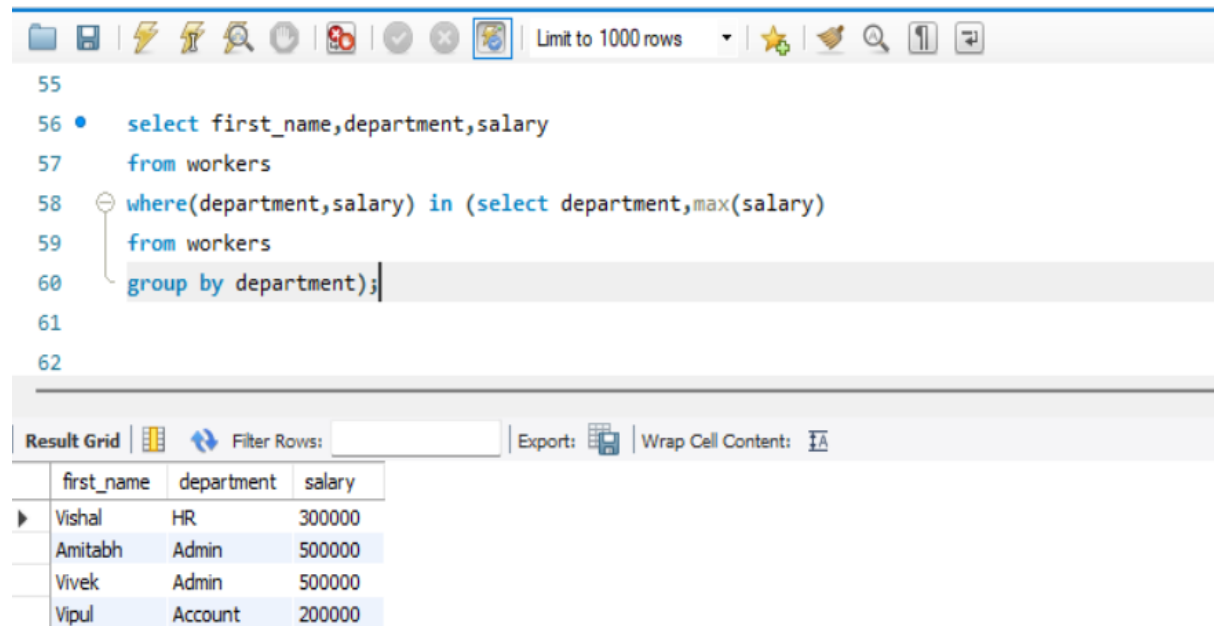
The screenshot shows a SQL IDE interface. The query editor contains the following SQL query:

```
48
49 • select department, count(*) as people from workers
50   group by department
51   having count(*) <5;
52
53 • select department, count(*) as people from workers
54   group by department;
55
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query in a table:

department	people
HR	3
Admin	4
Account	2

9. Write an SQL query to print the name of employees having the highest salary in each department.



The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. The SQL editor contains the following query:

```
55
56 • select first_name,department,salary
57 from workers
58 where(department,salary) in (select department,max(salary)
59 from workers
60 group by department);
61
62
```

Below the editor is the 'Result Grid' section, which includes a 'Filter Rows' input and 'Export' and 'Wrap Cell Content' options. The results are displayed in a table with the following data:

	first_name	department	salary
▶	Vishal	HR	300000
	Amitabh	Admin	500000
	Vivek	Admin	500000
	Vipul	Account	200000