

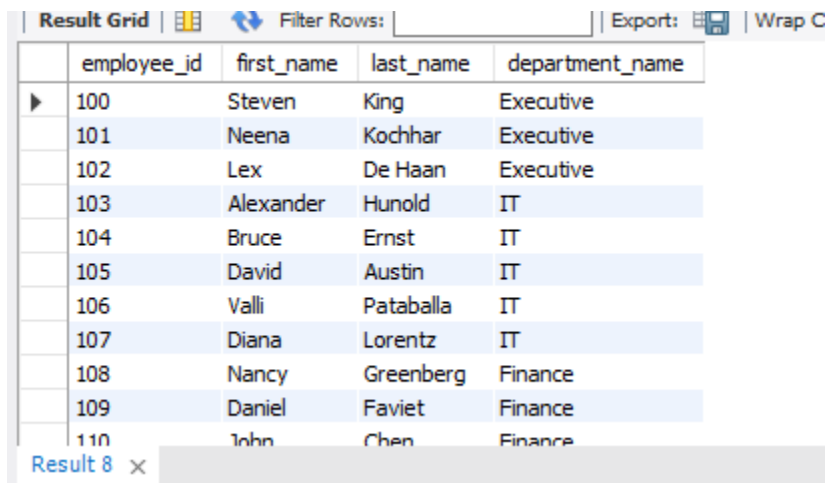
use hr;

```
/* Task 1 : Display the employee ID, first names,last names, and department names of all employees */
```

```
select employee_id,first_name,last_name,
```

```
(select department_name from departments where  
departments.department_id=employees.department_id) as department_name
```

```
from employees;
```



The screenshot shows a database query result grid with the following columns: employee\_id, first\_name, last\_name, and department\_name. The results are displayed in a table with alternating light blue and white rows. The data includes employees with IDs 100 through 110, with their respective first names, last names, and department names. The interface includes a 'Result Grid' header, a 'Filter Rows' input field, and an 'Export' button. The bottom of the window shows 'Result 8' and a close button.

employee_id	first_name	last_name	department_name
100	Steven	King	Executive
101	Neena	Kochhar	Executive
102	Lex	De Haan	Executive
103	Alexander	Hunold	IT
104	Bruce	Ernst	IT
105	David	Austin	IT
106	Valli	Pataballa	IT
107	Diana	Lorentz	IT
108	Nancy	Greenberg	Finance
109	Daniel	Faviet	Finance
110	John	Chen	Finance

```
/* Task 2 : Write a query to find the names & salaries of employees whose salaries are greater than  
average salary */
```

```
select first_name,last_name,salary from employees
```

```
where salary >
```

```
(select avg(salary) from employees);
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
first_name	last_name	salary	
Steven	King	24000.00	
Neena	Kochhar	17000.00	
Lex	De Haan	17000.00	
Alexander	Hunold	9000.00	
Nancy	Greenberg	12000.00	
Daniel	Faviet	9000.00	
John	Chen	8200.00	
Ismael	Sciarra	7700.00	
Jose Manuel	Urman	7800.00	
Luis	Popp	6900.00	
Den	Ranhaelv	11000.00	



/\* Task 3 : Write a query to find the names (first\_name,last\_name) and salaries of the employees who earn less than the average salary & work in any of the sales department \*/

select @z:=department\_id from departments where department\_name='sales' ;

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
@z:=department_id			
80			

select first\_name,last\_name from employees where salary > (select avg(salary) from employees) and department\_id=@z;

<

Result Grid   Filter Rows:



	first_name	last_name
▶	John	Russell
	Karen	Partners
	Alberto	Errazuriz
	Gerald	Cambrault
	Eleni	Zlotkey
	Peter	Tucker
	David	Bernstein
	Peter	Hall
	Christopher	Olsen
	Nanette	Cambrault
	Oliver	Tuvault

employees 11 x

Output

/\* Task 4 : Write a query to find the name and salary of the employees who have salary higher than the salary of all IT programmers (JOB ID = 'IT\_PROG'). Sort the salary results lowest to higher \*/

```
select first_name,last_name,salary from employees where salary >
(select max(salary) from employees where job_id='IT_PROG')
order by 3;
```

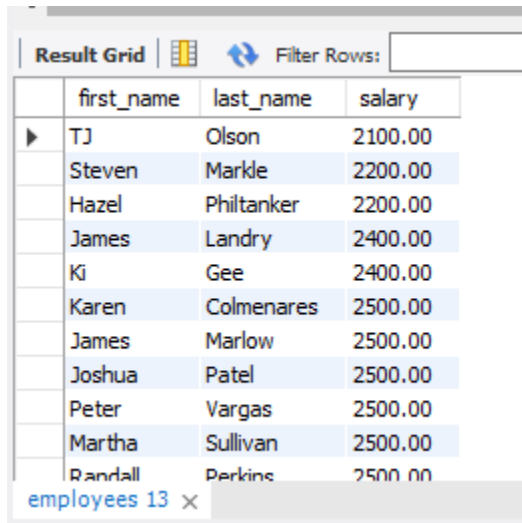
Result Grid   Filter Rows:

	first_name	last_name	salary
▶	David	Bernstein	9500.00
	Patrick	Sully	9500.00
	Danielle	Greene	9500.00
	Tayler	Fox	9600.00
	Peter	Tucker	10000.00
	Janette	King	10000.00
	Harrison	Bloom	10000.00
	Hermann	Baer	10000.00
	Eleni	Zlotkey	10500.00
	Clara	Vishney	10500.00
	Den	Ranhaelv	11000.00

employees 12 x

/\* Task 5 : Find the employee with the minimum salary in the employees table. Arrange the records in ascending order \*/

```
select first_name,last_name,salary from employees
order by 3;
```



The screenshot shows a database interface with a 'Result Grid' tab. It displays a table with columns 'first\_name', 'last\_name', and 'salary'. The data is sorted in ascending order of salary. The first row is TJ Olson with a salary of 2100.00. The last row is Randall Perkins with a salary of 2500.00. There are 13 rows in total. A 'Filter Rows' input field is visible at the top right of the grid.

	first_name	last_name	salary
▶	TJ	Olson	2100.00
	Steven	Markle	2200.00
	Hazel	Philtanker	2200.00
	James	Landry	2400.00
	Ki	Gee	2400.00
	Karen	Colmenares	2500.00
	James	Marlow	2500.00
	Joshua	Patel	2500.00
	Peter	Vargas	2500.00
	Martha	Sullivan	2500.00
	Randall	Perkins	2500.00

employees 13 x

-- Interpretation : TJ Olson has minimum salary

**/\* Task 6 : Find the names of employees whose salary is greater than 60 % of their departments total salary bill \*/**

```
create view Job_details as
select department_id,60*sum(salary)/100 as salary_perc from employees
group by department_id;
select * from Job_details;
```

Result Grid		Filter Rows:
	department_id	salary_perc
▶	NULL	4200.000000
	10	2640.000000
	20	11400.000000
	30	14940.000000
	40	3900.000000
	50	93840.000000
	60	17280.000000
	70	6000.000000
	80	182700.000000
	90	34800.000000
	100	30960.000000

Job\_details14 x

select first\_name,last\_name,salary\_perc,department\_id,salary from employees join Job\_details  
using(department\_id) where salary > salary\_perc;

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	first_name	last_name	salary_perc	department_id	salary
▶	Jennifer	Whalen	2640.000000	10	4400.00
	Michael	Hartstein	11400.000000	20	13000.00
	Susan	Mavris	3900.000000	40	6500.00
	Hermann	Baer	6000.000000	70	10000.00

**/\* Task 7 : Write a query to find the first name and last name of the employees who have manager and worked in UK-base department \*/**

select first\_name,last\_name,manager\_id from employees where department\_id in  
(select department\_id from departments where location\_id in  
(select location\_id from locations where country\_id in  
(select country\_id from countries where country\_name="United Kingdom")));

Result Grid			
		Filter Rows:	
		Export:	
		Wrap Cell Content:	
	first_name	last_name	manager_id
▶	Susan	Mavris	101
	John	Russell	100
	Karen	Partners	100
	Alberto	Errazuriz	100
	Gerald	Cambrault	100
	Eleni	Zlotkey	100
	Peter	Tucker	145
	David	Bernstein	145
	Peter	Hall	145
	Christopher	Olsen	145
	Nanette	Cambrault	145

/\* Task 8 : Write salary, first name, last name of the highest paid employees and export the output as a csv file \*/

show variables like "secure\_file\_priv";

select first\_name,last\_name,salary from employees

order by 3 desc

into outfile 'C:\\ProgramData\\MySQL\\MySQL Server 8.0\\Uploads\\Myfile5.csv'

fields terminated by ','

lines terminated by'\n';

F18				
	A	B	C	D
1	Steven	King	24000	
2	Neena	Kochhar	17000	
3	Lex	De Haan	17000	
4	John	Russell	14000	
5	Karen	Partners	13500	
6	Michael	Hartstein	13000	
7	Nancy	Greenberg	12000	
8	Alberto	Errazuriz	12000	
9	Shelley	Higgins	12000	
10	Lisa	Ozer	11500	
11	Den	Raphaely	11000	
12	Gerald	Cambrault	11000	
13	Ellen	Abel	11000	
14	Eleni	Zlotkey	10500	
15	Clara	Vishney	10500	
16	Peter	Tucker	10000	
17	Janette	King	10000	
18	Harrison	Bloom	10000	
19	Hermann	Baer	10000	
20	Tayler	Fox	9600	
21	David	Bernstein	9500	
22	Patrick	Sully	9500	