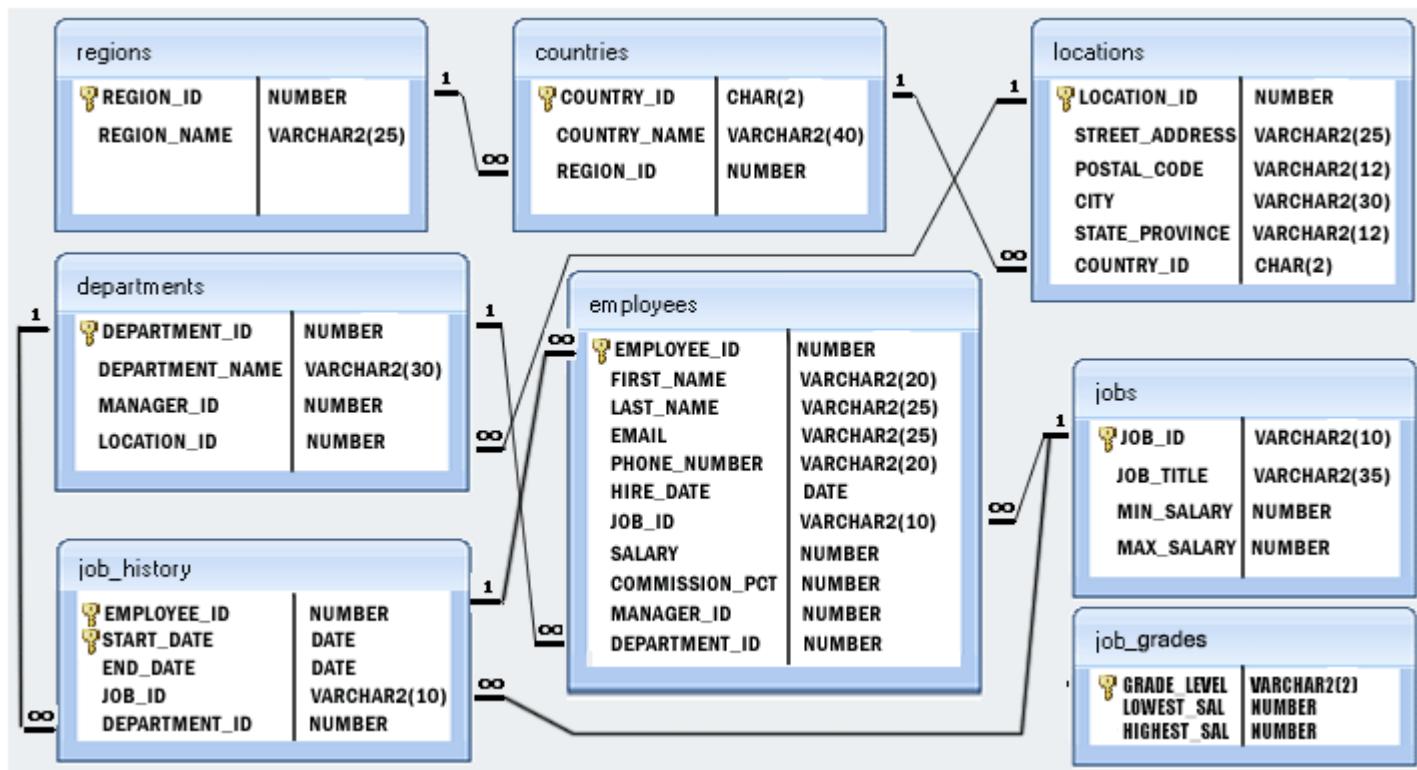


SELECT Statement Examples



The Human Resources (HR) Database Sampling

This sample database consists of 8 tables, as depicted in the following Entity-Relationship (ER) Diagram.



Run the SQL Scripts

Before we proceed any further, please go ahead and import the SQL Script ([click here](https://bryanuniversity.instructure.com/courses/10152/files/744673/download?wrap=1)) (<https://bryanuniversity.instructure.com/courses/10152/files/744673/download?wrap=1>) to download the script file) into MySQL Work Bench and/or PostgreSQL applications.



Basic SQL SELECT Statement: Employee Table

employee_id	first_name	last_name	email	phone_number	department_id
100	Steven	King	SKING	515.123.4567	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	90
102	Lex	De Haan	LDEHAAN	515.123.4569	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	60
104	Bruce	Ernst	BERNST	590.423.4568	60
105	David	Austin	DAUSTIN	590.423.4569	60
106	Valli	Pataballa	VPATABAL	590.423.4560	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	60
108	Nancy	Greenberg	NGREENBE	515.124.4569	100
109	Daniel	Faviet	DFAVIET	515.124.4169	100

**SELECT first_name "First Name", last_name "Last Name"
FROM employees;**

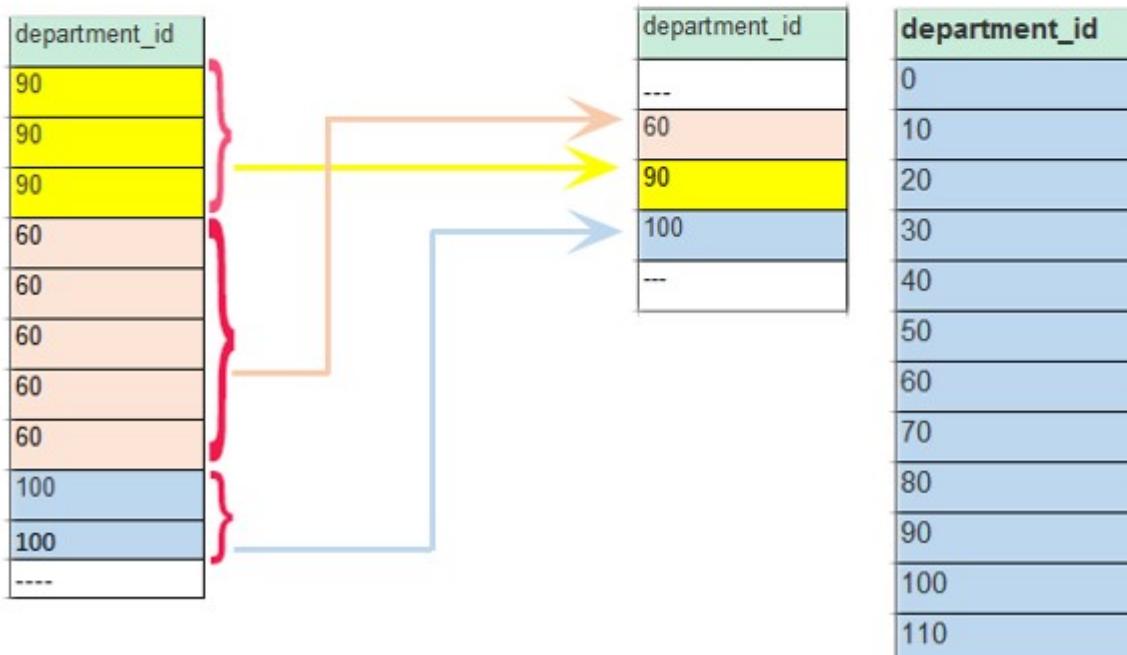
First Name	Last Name
Ellen	Abel
Sundar	Ande
Mozhe	Atkinson
David	Austin
Hermann	Baer
Shelli	Baida
Amit	Banda
Elizabeth	Bates
Sarah	Bell
David	Berman

- Write a query to get unique department ID from the employee table.

```
SELECT DISTINCT department_id  
FROM employees;
```

employee_id	first_name	last_name	email	phone_number	-----	department_id
100	Steven	King	SKING	515.123.4567	-----	90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	-----	90
102	Lex	De Haan	LDEHAAN	515.123.4569	-----	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	-----	60
104	Bruce	Ernst	BERNST	590.423.4568	-----	60
105	David	Austin	DAUSTIN	590.423.4569	-----	60
106	Valli	Pataballa	VPATABAL	590.423.4560	-----	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	-----	60
108	Nancy	Greenberg	NGREENBE	515.124.4569	-----	100
109	Daniel	Faviet	DFAVIET	515.124.4169	-----	100
-----	-----	-----	-----	-----	-----	-----

**SELECT DISTINCT department_id
FROM employees;**



- Write a query to get all employee details from the employee table order by first name, descending.

```
SELECT *
FROM employees
ORDER BY first_name DESC;
```

The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema with tables like employees, departments, and locations. The main area contains a SQL editor with the following code:

```

10 • SELECT first_name, last_name, LENGTH(first_name) + LENGTH(last_name) AS 'Length of Names' FROM employees;
11
12 -- 4. Write a query to select first 10 records from a table.
13 • SELECT employee_id, first_name, last_name FROM employees LIMIT 10;
14
15 -- 5. Write a query to display the names (first_name, last_name) and salary for all employees whose salary is not in the range $10,000 through $15,000.
16 • SELECT first_name, last_name, salary FROM employees WHERE salary NOT BETWEEN
17
18 -- 6. Write a query to display the name (first_name, last_name) and department ID of all employees in departments 30 or 100 in ascending order.
19
20
21 -- 7. Write a query to display the name (first_name, last_name) and salary for all employees whose salary is not in the range $10,000 through $15,000 and are in department 30 or 100.
22
23
24 -- 8. Write a query to display the name (first_name, last_name) and hire date for all employees who were hired in 1987.

```

The results grid shows the following data:

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

- Write a query to display the names (first_name, last_name) using alias name "First Name", "Last Name" respectively.

```
SELECT first_name "First Name", last_name "Last Name"
FROM employees;
```

employee_id	first_name	last_name	email	phone_number	-----
100	Steven	King	SKING	515.123.4567	-----
101	Neena	Kochhar	NKOCHHAR	515.123.4568	-----
102	Lex	De Haan	LDEHAAN	515.123.4569	-----
103	Alexander	Hunold	AHUNOLD	590.423.4567	-----
104	Bruce	Ernst	BERNST	590.423.4568	-----
105	David	Austin	DAUSTIN	590.423.4569	-----
106	Valli	Pataballa	VPATABAL	590.423.4560	-----
107	Diana	Lorentz	DLORENTZ	590.423.5567	-----
108	Nancy	Greenberg	NGREENBE	515.124.4569	-----
109	Daniel	Faviet	DFAVIET	515.124.4169	-----
-----	-----	-----	-----	-----	-----

```
SELECT *
FROM employees
ORDER BY first_name DESC ;
```

the first_name have been arranged in descendingly

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NU
180	Winston	Taylor	WTAYLOR	650.507.987
171	William	Smith	WSMITH	011.44.134
206	William	Gietz	WGIETZ	515.123.8
195	Vance	Jones	VJONES	650.501.48
106	Valli	Pataballa	VPATABAL	590.423.456
141	Trenna	Rajs	TRAJS	650.121.800
132	TJ	Olson	TJOLSON	650.124.823
190	Timothy	Gates	TGATES	650.505.38
170	Tayler	Fox	TFOX	011.44.134
203	Susan	Mavris	SMAVRIS	515.123.7
173	Sundita	Kumar	SKUMAR	011.44.1
166	Sund	Ande	SANDE	011.44.1

- Write a query to get the names (first_name, last_name), salary, PF of all the employees (ex: Pre-Funded Salary or PF is calculated as 15% of salary).

```
SELECT first_name, last_name, salary, salary*.15 "PF"
FROM employees;
```

employee_id	first_name	last_name	email	phone_number		salary	
100	Steven	King	SKING	515.123.4567	-----	24000	-----
101	Neena	Kochhar	NKOCHHAR	515.123.4568	-----	17000	-----
102	Lex	De Haan	LDEHAAN	515.123.4569	-----	17000	-----
103	Alexander	Hunold	AHUNOLD	590.423.4567	-----	9000	-----
104	Bruce	Ernst	BERNST	590.423.4568	-----	6000	-----
105	David	Austin	DAUSTIN	590.423.4569	-----	4800	-----
106	Valli	Pataballa	VPATABAL	590.423.4560	-----	4800	-----
107	Diana	Lorentz	DLORENTZ	590.423.5567	-----	4200	-----
108	Nancy	Greenberg	NGREENBE	515.124.4569	-----	12000	-----
109	Daniel	Faviet	DFAVIET	515.124.4169	-----	9000	-----
-----	-----	-----	-----	-----	-----	-----	-----

**SELECT first_name, last_name, salary, salary*.15 PF
FROM employees;**

PF calculated by 15% of salary

salary	PF
24000	3600
17000	2550
17000	2550
9000	1350
6000	900
4800	720
4800	720
4200	630
12000	1800
9000	1350
---	---

first_name	last_name	salary	PF
Steven	King	24000	3600
Neena	Kochhar	17000	2550
Lex	De Haan	17000	2550
Alexander	Hunold	9000	1350
Bruce	Ernst	6000	900
David	Austin	4800	720
Valli	Pataballa	4800	720
Diana	Lorentz	4200	630
Nancy	Greenberg	12000	1800
Daniel	Faviet	9000	1350
-----	-----	-----	-----

- Write a query to get the employee ID, names (first_name, last_name), salary in ascending order of salary.

```
SELECT employee_id, first_name, last_name, salary
FROM employees
ORDER BY salary;
```



employee_id	first_name	last_name	email	phone_number		salary	
100	Steven	King	SKING	515.123.4567	-----	24000	-----
101	Neena	Kochhar	NKOCHHAR	515.123.4568	-----	17000	-----
102	Lex	De Haan	LDEHAAN	515.123.4569	-----	17000	-----
103	Alexander	Hunold	AHUNOLD	590.423.4567	-----	9000	-----
104	Bruce	Ernst	BERNST	590.423.4568	-----	6000	-----
105	David	Austin	DAUSTIN	590.423.4569	-----	4800	-----
106	Valli	Pataballa	VPATABAL	590.423.4560	-----	4800	-----
107	Diana	Lorentz	DLORENTZ	590.423.5567	-----	4200	-----
108	Nancy	Greenberg	NGREENBE	515.124.4569	-----	12000	-----
109	Daniel	Faviet	DFAVIET	515.124.4169	-----	9000	-----
-----	-----	-----	-----	-----	-----	-----	-----

```
SELECT employee_id, first_name, last_name, salary
FROM employees
ORDER BY salary ;
```

salaries have been arranged in ascending order



salary

24000
17000
17000
9000
6000
4800
4800
4200
12000
9000



salary

4200
4800
4800
6000
9000
9000
12000
17000
17000
24000



employee_id	first_name	last_name	salary
132	TJ	Olson	2100
128	Steven	Markle	2200
136	Hazel	Philtanker	2200
127	James	Landry	2400
135	Ki	Gee	2400
119	Karen	Colmenares	2500
131	James	Marlow	2500
140	Joshua	Patel	2500
144	Peter	Vargas	2500
182	Martha	Sullivan	2500
191	Randall	Perkins	2500
118	Guy	Himuro	2600
143	Randall	Matos	2600
198	Donald	OConnell	2600

- Write a query to get the total salaries payable to employees.

```
SELECT SUM(salary)
FROM employees;
```

employee_id	first_name	last_name	email	phone_number		salary
100	Steven	King	SKING	515.123.4567	-----	24000
101	Neena	Kochhar	NKOCHHAR	515.123.4568	-----	17000
102	Lex	De Haan	LDEHAAN	515.123.4569	-----	17000
103	Alexander	Hunold	AHUNOLD	590.423.4567	-----	9000
104	Bruce	Ernst	BERNST	590.423.4568	-----	6000
105	David	Austin	DAUSTIN	590.423.4569	-----	4800
106	Valli	Pataballa	VPATABAL	590.423.4560	-----	4800
107	Diana	Lorentz	DLORENTZ	590.423.5567	-----	4200
108	Nancy	Greenberg	NGREENBE	515.124.4569	-----	12000
109	Daniel	Faviet	DFAVIET	515.124.4169	-----	9000
-----	-----	-----	-----	-----	-----	-----

SELECT **SUM(salary)**
FROM employees;

SUM(salary)

691400

- Write a query to get the maximum and minimum salary from employees table.

```
SELECT MAX(salary), MIN(salary)
FROM employees;
```

employee_id	first_name	last_name	email	phone_number		salary	
100	Steven	King	SKING	515.123.4567	-----	24000	-----
101	Neena	Kochhar	NKOCHHAR	515.123.4568	-----	17000	-----
102	Lex	De Haan	LDEHAAN	515.123.4569	-----	17000	-----
103	Alexander	Hunold	AHUNOLD	590.423.4567	-----	9000	-----
104	Bruce	Ernst	BERNST	590.423.4568	-----	6000	-----
105	David	Austin	DAUSTIN	590.423.4569	-----	4800	-----
106	Valli	Pataballa	VPATABAL	590.423.4560	-----	4800	-----
107	Diana	Lorentz	DLORENTZ	590.423.5567	-----	4200	-----
108	Nancy	Greenberg	NGREENBE	515.124.4569	-----	12000	-----
109	Daniel	Faviet	DFAVIET	515.124.4169	-----	9000	-----
-----	-----	-----	-----	-----	-----	-----	-----

SELECT MAX(salary), MIN(salary)
FROM employees;

Maximum Salary

salary
24000
17000
17000
9000
6000
4800
4800
4200
12000
9000

2100

Minimum Salary

MAX(salary)	MIN(salary)
24000	2100

(<https://bryanuniversity.instructure.com/courses/10152/files/744634/download?wrap=1>)

- Write a query to get the average salary and number of employees in the employees table.

```
SELECT AVG(salary), COUNT(*)
FROM employees;
```

employee_id	first_name	last_name	email	phone_number		salary	
100	Steven	King	SKING	515.123.4567	-----	24000	-----
101	Neena	Kochhar	NKOCHHAR	515.123.4568	-----	17000	-----
102	Lex	De Haan	LDEHAAN	515.123.4569	-----	17000	-----
103	Alexander	Hunold	AHUNOLD	590.423.4567	-----	9000	-----
104	Bruce	Ernst	BERNST	590.423.4568	-----	6000	-----
105	David	Austin	DAUSTIN	590.423.4569	-----	4800	-----
106	Valli	Pataballa	VPATABAL	590.423.4560	-----	4800	-----
107	Diana	Lorentz	DLORENTZ	590.423.5567	-----	4200	-----
108	Nancy	Greenberg	NGREENBE	515.124.4569	-----	12000	-----
109	Daniel	Faviet	DFAVIET	515.124.4169	-----	9000	-----
-----	-----	-----	-----	-----	-----	-----	-----

SELECT **AVG(salary), COUNT(*)**
FROM employees;

AVG(salary)	COUNT(*)
6461.682243	107

- Write a query to get the number of employees working with the company.

```
SELECT COUNT(*)
FROM employees;
```

employee_id	first_name	last_name	email	phone_number	-----
100	Steven	King	SKING	515.123.4567	-----
101	Neena	Kochhar	NKOCHHAR	515.123.4568	-----
102	Lex	De Haan	LDEHAAN	515.123.4569	-----
103	Alexander	Hunold	AHUNOLD	590.423.4567	-----
104	Bruce	Ernst	BERNST	590.423.4568	-----
105	David	Austin	DAUSTIN	590.423.4569	-----
106	Valli	Pataballa	VPATABAL	590.423.4560	-----
107	Diana	Lorentz	DLORENTZ	590.423.5567	-----
108	Nancy	Greenberg	NGREENBE	515.124.4569	-----
109	Daniel	Faviet	DFAVIET	515.124.4169	-----
-----	-----	-----	-----	-----	-----

SELECT **COUNT(*)**
FROM employees;

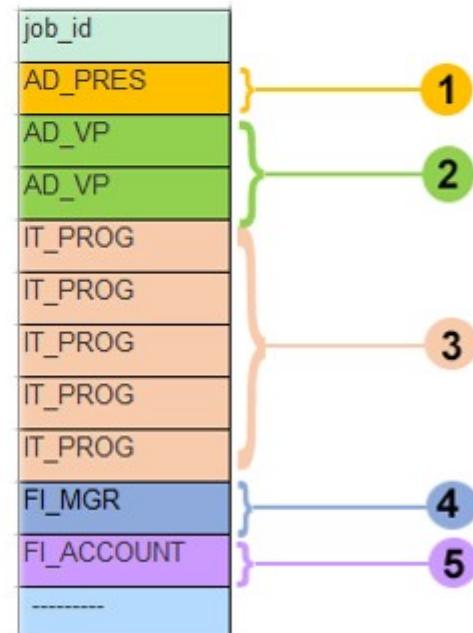
COUNT(*)
107

- Write a query to get the number of designations available in the employees table.

```
SELECT COUNT(DISTINCT job_id)
FROM employees;
```

employee_id	first_name	last_name	email	phone_number	-----	job_id	-----
100	Steven	King	SKING	515.123.4567	-----	AD_PRES	-----
101	Neena	Kochhar	NKOCHHAR	515.123.4568	-----	AD_VP	-----
102	Lex	De Haan	LDEHAAN	515.123.4569	-----	AD_VP	-----
103	Alexander	Hunold	AHUNOLD	590.423.4567	-----	IT_PROG	-----
104	Bruce	Ernst	BERNST	590.423.4568	-----	IT_PROG	-----
105	David	Austin	DAUSTIN	590.423.4569	-----	IT_PROG	-----
106	Valli	Pataballa	VPATABAL	590.423.4560	-----	IT_PROG	-----
107	Diana	Lorentz	DLORENTZ	590.423.5567	-----	IT_PROG	-----
108	Nancy	Greenberg	NGREENBE	515.124.4569	-----	FI_MGR	-----
109	Daniel	Faviet	DFAVIET	515.124.4169	-----	FI_ACCOUNT	-----
-----	-----	-----	-----	-----	-----	-----	-----

SELECT COUNT(DISTINCT job_id)
FROM employees;



COUNT(DISTINCT job_id)
19

- Write a query get all first name from employees table in upper case.

```
SELECT UPPER(first_name)
FROM employees;
```

employee_id	first_name	last_name	email	phone_number	-----
100	Steven	King	SKING	515.123.4567	-----
101	Neena	Kochhar	NKOCHHAR	515.123.4568	-----
102	Lex	De Haan	LDEHAAN	515.123.4569	-----
103	Alexander	Hunold	AHUNOLD	590.423.4567	-----
104	Bruce	Ernst	BERNST	590.423.4568	-----
105	David	Austin	DAUSTIN	590.423.4569	-----
106	Valli	Pataballa	VPATABAL	590.423.4560	-----
107	Diana	Lorentz	DLORENTZ	590.423.5567	-----
108	Nancy	Greenberg	NGREENBE	515.124.4569	-----
109	Daniel	Faviet	DFAVIET	515.124.4169	-----
-----	-----	-----	-----	-----	-----

```
SELECT UPPER(first_name)
FROM employees;
```

The diagram illustrates the transformation of first names from their original form in the employees table to their uppercase equivalents using the **UPPER** function.

Employees Table:

FIRST_NAME
Ellen
Sundar
Mozhe
David
Hermann
Shelli
Amit
Elizabeth
Sarah
David

Resulting Table:

UPPER(first_name)
ELLEN
SUNDAR
MOZHE
DAVID
HERMANN
SHELLI
AMIT
ELIZABETH
SARAH
DAVID

- Write a query to get the first 3 characters of first name from employees table.

```
SELECT SUBSTRING(first_name,1,3)
FROM employees;
```

employee_id	first_name	last_name	email	phone_number	-----
100	Steven	King	SKING	515.123.4567	-----
101	Neena	Kochhar	NKOCHHAR	515.123.4568	-----
102	Lex	De Haan	LDEHAAN	515.123.4569	-----
103	Alexander	Hunold	AHUNOLD	590.423.4567	-----
104	Bruce	Ernst	BERNST	590.423.4568	-----
105	David	Austin	DAUSTIN	590.423.4569	-----
106	Valli	Pataballa	VPATABAL	590.423.4560	-----
107	Diana	Lorentz	DLORENTZ	590.423.5567	-----
108	Nancy	Greenberg	NGREENBE	515.124.4569	-----
109	Daniel	Faviet	DFAVIET	515.124.4169	-----
-----	-----	-----	-----	-----	-----

**SELECT SUBSTRING(first_name,1,3)
FROM employees;**

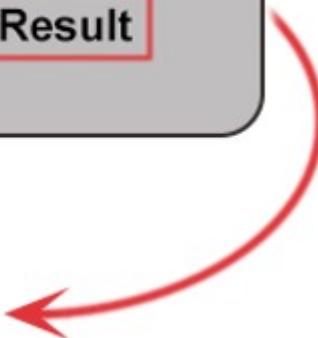
FIRST_NAME	SUBSTRING (first_name,1,3)
Ellen	Ell
Sundar	Sun
Mozhe	Moz
David	Dav
Hermann	Her
Shelli	She
Amit	Ami
Elizabeth	Eli
Sarah	Sar
David	Dav
---	----

- Write a query to calculate $171*214+625$.

SELECT 171*214+625 Result;

SELECT **171*214+625 Result**

Result
37219



- Write a query to get the name (for example Ellen Abel, Sundar Ande etc.) of all the employees from employees table.

```
SELECT CONCAT(first_name, ' ', last_name) 'Employee Name'  
FROM employees;
```

employee_id	first_name	last_name	email	phone_number	-----
100	Steven	King	SKING	515.123.4567	-----
101	Neena	Kochhar	NKOCHHAR	515.123.4568	-----
102	Lex	De Haan	LDEHAAN	515.123.4569	-----
103	Alexander	Hunold	AHUNOLD	590.423.4567	-----
104	Bruce	Ernst	BERNST	590.423.4568	-----
105	David	Austin	DAUSTIN	590.423.4569	-----
106	Valli	Pataballa	VPATABAL	590.423.4560	-----
107	Diana	Lorentz	DLORENTZ	590.423.5567	-----
108	Nancy	Greenberg	NGREENBE	515.124.4569	-----
109	Daniel	Faviet	DFAVIET	515.124.4169	-----
-----	-----	-----	-----	-----	-----

```
SELECT CONCAT(first_name, ' ', last_name) 'Employee Name'
FROM employees;
```

FIRST_NAME	LAST_NAME
Ellen	Abel
Sundar	Ande
Mozhe	Atkinson
David	Austin
Hermann	Baer
Shelli	Baida
Amit	Banda
Elizabeth	Bates
Sarah	Bell
David	Bernstein
---	---

first_name and
last_name have
concatenated →

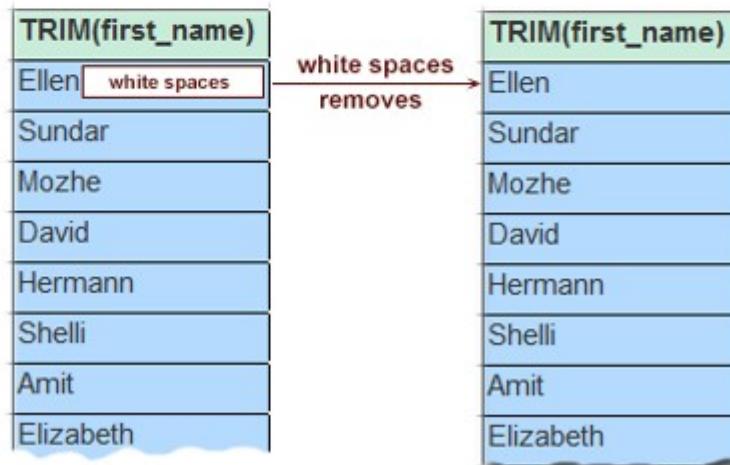
Employee Name
Ellen Abel
Sundar Ande
Mozhe Atkinson
David Austin
Hermann Baer
Shelli Baida
Amit Banda
Elizabeth Bates
Sarah Bell
David Bernstein

- Write a query to get first name of all employees table after removing white spaces from both side.

```
SELECT TRIM(first_name)
FROM employees;
```

employee_id	first_name	last_name	email	phone_number	
100	Steven	King	SKING	515.123.4567	-----
101	Neena	Kochhar	NKOCHHAR	515.123.4568	-----
102	Lex	De Haan	LDEHAAN	515.123.4569	-----
103	Alexander	Hunold	AHUNOLD	590.423.4567	-----
104	Bruce	Ernst	BERNST	590.423.4568	-----
105	David	Austin	DAUSTIN	590.423.4569	-----
106	Valli	Pataballa	VPATABAL	590.423.4560	-----
107	Diana	Lorentz	DLORENTZ	590.423.5567	-----
108	Nancy	Greenberg	NGREENBE	515.124.4569	-----
109	Daniel	Faviet	DFAVIET	515.124.4169	-----
-----	-----	-----	-----	-----	-----

SELECT **TRIM(first_name)**
FROM employees;



- Write a query to get the length of the employee names (first_name, last_name) from employees table.

```
SELECT first_name, last_name, LENGTH(first_name)+LENGTH(last_name) 'Length of Names'  
FROM employees;
```

employee_id	first_name	last_name	email	phone_number	-----
100	Steven	King	SKING	515.123.4567	-----
101	Neena	Kochhar	NKOCHHAR	515.123.4568	-----
102	Lex	De Haan	LDEHAAN	515.123.4569	-----
103	Alexander	Hunold	AHUNOLD	590.423.4567	-----
104	Bruce	Ernst	BERNST	590.423.4568	-----
105	David	Austin	DAUSTIN	590.423.4569	-----
106	Valli	Pataballa	VPATABAL	590.423.4560	-----
107	Diana	Lorentz	DLORENTZ	590.423.5567	-----
108	Nancy	Greenberg	NGREENBE	515.124.4569	-----
109	Daniel	Faviet	DFAVIET	515.124.4169	-----
-----	-----	-----	-----	-----	-----

```
SELECT first_name, last_name, LENGTH(first_name)
+LENGTH(last_name) 'Length of Names'
FROM employees;
```

first_name	last_name	Length of Names
Ellen → 5	Abel → 4	9
Sundar	Ande	10
Mozhe	Atkinson	13
David	Austin	11
Hermann	Baer	11
Shelli	Baida	11
Amit	Banda	9
Elizabeth	Bates	14
Sarah	Bell	9
David		

- Write a query to select first 10 records from a table.

```
SELECT employee_id, first_name
FROM employees LIMIT 10;
```

employee_id	first_name	last_name	email	phone_number	
100	Steven	King	SKING	515.123.4567	-----
101	Neena	Kochhar	NKOCHHAR	515.123.4568	-----
102	Lex	De Haan	LDEHAAN	515.123.4569	-----
103	Alexander	Hunold	AHUNOLD	590.423.4567	-----
104	Bruce	Ernst	BERNST	590.423.4568	-----
105	David	Austin	DAUSTIN	590.423.4569	-----
106	Valli	Pataballa	VPATABAL	590.423.4560	-----
107	Diana	Lorentz	DLORENTZ	590.423.5567	-----
108	Nancy	Greenberg	NGREENBE	515.124.4569	-----
109	Daniel	Faviet	DFAVIET	515.124.4169	-----
-----	-----	-----	-----	-----	-----

```
SELECT employee_id, first_name
FROM employees
LIMIT 10 ;
```

employee_id	first_name
100	Steven
101	Neena
102	Lex
103	Alexander
104	Bruce
105	David
106	Valli
107	Diana
108	Nancy
109	Daniel

- Write a query to get monthly salary (round 2 decimal places) of all employees.

```
SELECT first_name, last_name, round(salary/12,2) as 'Monthly Salary'
FROM employees;
```

employee_id	first_name	last_name	email	phone_number		salary	
100	Steven	King	SKING	515.123.4567	-----	24000	-----
101	Neena	Kochhar	NKOCHHAR	515.123.4568	-----	17000	-----
102	Lex	De Haan	LDEHAAN	515.123.4569	-----	17000	-----
103	Alexander	Hunold	AHUNOLD	590.423.4567	-----	9000	-----
104	Bruce	Ernst	BERNST	590.423.4568	-----	6000	-----
105	David	Austin	DAUSTIN	590.423.4569	-----	4800	-----
106	Valli	Pataballa	VPATABAL	590.423.4560	-----	4800	-----
107	Diana	Lorentz	DLORENTZ	590.423.5567	-----	4200	-----
108	Nancy	Greenberg	NGREENBE	515.124.4569	-----	12000	-----
109	Daniel	Faviet	DFAVIET	515.124.4169	-----	9000	-----
-----	-----	-----	-----	-----	-----	-----	-----

```
SELECT first_name, last_name, round(salary/12,2) as 'Monthly Salary'
FROM employees;
```

salary	round (salary/12, 2)	first_name	last_name	Monthly
24000	2000	Steven	King	2000
17000	1416.67	Neena	Kochhar	1416.67
17000	1416.67	Lex	De Haan	1416.67
9000	750	Alexander	Hunold	750
6000	500	Bruce	Ernst	500
4800	400	David	Austin	400
4800	400	Valli	Pataballa	400



Restructuring and Sorting Data: Employees Table

- Write a query to display the names (first_name, last_name) and salary for all employees whose salary is not in the range \$10,000 through \$15,000.

```
SELECT first_name, last_name, salary
FROM employees
WHERE salary NOT BETWEEN 10000 AND 15000;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
100	Steven	King	24000	90
101	Neena	Kochhar	17000	90
102	Lex	De Haan	17000	90
103	Alexander	Hunold	9000	60
104	Bruce	Ernst	6000	60
105	David	Austin	4800	60
106	Valli	Pataballa	4800	60
107	Diana	Lorentz	4200	60
108	Nancy	Greenberg	12000	100

```
SELECT first_name, last_name, salary
FROM employees
WHERE salary NOT BETWEEN 10000 AND 15000 ;
```

salary arranged in not between 10000 and 15000

FIRST_NAME	LAST_NAME	SALARY
Steven	King	24000
Neena	Kochhar	17000
Lex	De Haan	17000
Alexander	Hunold	9000
Bruce	Ernst	6000
David	Austin	4800
Valli	Pataballa	4800
Diana	Lorentz	4200
Nancy	Greenberg	12000

first_name	last_name	salary
Steven	King	24000
Neena	Kochhar	17000
Lex	De Haan	17000
Alexander	Hunold	9000
Bruce	Ernst	6000
David	Austin	4800
Valli	Pataballa	4800
Diana	Lorentz	4200
Daniel	Faviet	9000
-----	-----	-----

- Write a query to display the name (first_name, last_name) and department ID of all employees in departments 30 or 100 in ascending order.

```
SELECT first_name, last_name, department_id
FROM employees
WHERE department_id IN (30, 100)
ORDER BY department_id ASC;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT_ID
100	Steven	King	90
101	Neena	Kochhar	90
102	Lex	De Haan	90
103	Alexander	Hunold	60
104	Bruce	Ernst	60
-	-	-	-
108	Nancy	Greenberg	100
-	-	-	-
114	Den	Raphaely	30
-	-	-	-

```

SELECT first_name, last_name, department_id
  FROM employees
 WHERE department_id IN (30, 100)
 ORDER BY department_id ASC ;
  
```

Departments 30 or 100 in ascending alphabetical order by department ID

FIRST_NAME	LAST_NAME	DEPARTMENT_ID
Steven	King	90
Neena	Kochhar	90
Lex	De Haan	90
Alexander	Hunold	60
Bruce	Ernst	60
-	-	-
Nancy	Greenberg	100
-	-	-
Den	Raphaely	30
-	-	-

first_name	last_name	department_id
Den	Raphaely	30
Alexander	Khoo	30
Shelli	Baida	30
Sigal	Tobias	30
Guy	Himuro	30
Karen	Colmenares	30
Nancy	Greenberg	100
Daniel	Faviet	100
John	Chen	100
Ismael	Sciarra	100
Jose Manuel	Urman	100
Luis	Popp	100

- Write a query to display the name (first_name, last_name) and salary for all employees whose salary is not in the range \$10,000 through \$15,000 and are in department 30 or 100.

```
SELECT first_name, last_name, salary, department_id
FROM employees
WHERE salary NOT BETWEEN 10000 AND 15000
AND department_id IN (30, 100);
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME		SALARY		DEPARTMENT_ID
100	Steven	King	—	24000	—	90
101	Neena	Kochhar	—	17000	—	90
102	Lex	De Haan	—	17000	—	90
103	Alexander	Hunold	—	9000	—	60
104	Bruce	Ernst	—	6000	—	60
—	—	—	—	—	—	—
108	Nancy	Greenberg	—	12000	—	100
109	Daniel	Faviet	—	9000	—	100
—	—	—	—	—	—	—
114	Den	Raphaely	—	11000	—	30
—	—	—	—	—	—	—

SELECT first_name, last_name, salary, department_id
 FROM employees
 WHERE salary NOT BETWEEN 10000 AND 15000
 AND department_id IN (30, 100) ;

Explain:

Greenberg	12000	100	X
Faviet	9000	100	✓

salary is within the 10000 to 15000 return false
 department_id return true

salary is out of 10000 to 15000 return true
 department_id return true

first_name	last_name	salary	department_id
Alexander	Khoo	3100	30
Shelli	Baida	2900	30
Sigal	Tobias	2800	30
Guy	Himuro	2600	30
Karen	Colmenares	2500	30
Daniel	Faviet	9000	100
John	Chen	8200	100
Janet	King	21000	90

- Write a query to display the name (first_name, last_name) and hire date for all employees who were hired in 1987.

```
SELECT first_name, last_name, hire_date
FROM employees
WHERE YEAR(hire_date) LIKE '1987%';
```

employee_id	first_name	last_name	email	phone_number	hire_date	---
100	Steven	King	SKING	515.123.4567	1987/06/17	----
101	Neena	Kochhar	NKOCHHA D	515.123.4568	1987/06/18	----
102	Lex	De Haan	LDEHAAN	515.123.4569	1987/06/19	----
103	Alexander	Hunold	AHUNOLD	590.423.4567	1987/06/20	----
104	Bruce	Ernst	BERNST	590.423.4568	1987/06/21	----
105	David	Austin	DAUSTIN	590.423.4569	1987/06/22	----
106	Valli	Pataballa	VPATABAL	590.423.4560	1987/06/23	----
107	Diana	Lorentz	DLORENTZ	590.423.5567	1987/06/24	----
108	Nancy	Greenberg	NGREENB E	515.124.4569	1987/06/25	----
109	Daniel	Faviet	DFAVIET	515.124.4169	1987/06/26	----
----	-----	-----	-----	-----	-----	-----

SELECT **first_name, last_name, hire_date**
FROM employees
WHERE YEAR(hire_date) LIKE '%1987' ;

first_name	last_name	hire_date
Steven	King	1987/06/17
Neena	Kochhar	1987/06/18
Lex	De Haan	1987/06/19
Alexander	Hunold	1987/06/20
Bruce	Ernst	1987/06/21
David	Austin	1987/06/22
Valli	Pataballa	1987/06/23
Diana	Lorentz	1987/06/24
Nancy	Greenberg	1987/06/25
Daniel	Faviet	1987/06/26
-----	-----	-----

- Write a query to display the first_name of all employees who have both "b" and "c" in their first name.

```
SELECT first_name
FROM employees
WHERE first_name LIKE '%b%'
    AND first_name LIKE '%c%';
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
100	Steven	King	24000	90
101	Neena	Kochhar	17000	90
102	Lex	De Haan	17000	90
103	Alexander	Hunold	9000	60
104	Bruce	Ernst	6000	60
-	-	-	-	-

**SELECT first_name
FROM employees
WHERE first_name LIKE '%b%'
AND first_name LIKE '%c%' ;**

first_name
Steven
Neena
Lex
Alexander
Bruce
David
Valli
Diana
Nancy
Daniel

first_name
Bruce

the first_name of all employees who have both "b" and "c" in their first name

- Write a query to display the last name, job, and salary for all employees whose job is that of a Programmer or a Shipping Clerk, and salary is not equal to \$4,500, \$10,000, or \$15,000.

```
SELECT last_name, job_id, salary
FROM employees
```

```
WHERE job_id IN ('IT_PROG', 'SH_CLERK')
AND salary NOT IN (4500,10000, 15000);
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOB_ID	SALARY	DEPARTMENT_ID
100	Steven	King	AD_PRES	24000	90
—	—	—	—	—	—
103	Alexander	Hunold	IT_PROG	9000	90
104	Bruce	Ernst	IT_PROG	6000	60
105	David	Austin	IT_PROG	4800	60
106	Valli	Pataballa	IT_PROG	4800	60
107	Diana	Lorentz	IT_PROG	4200	60
—	—	—	—	—	—
181	Jean	Fleaur	SH_CLERK	3100	100
—	—	—	—	—	—

SELECT last_name, job_id, salary
 FROM employees
 WHERE job_id IN ('IT_PROG', 'SH_CLERK')
 AND salary NOT IN (4500,10000, 15000) ;

job_id IN ('IT_PROG', 'SH_CLERK')
 AND salary NOT IN (4500,10000, 15000)

LAST_NAME	JOB_ID	SALARY
King	AD_PRES	24000
—	—	—
Hunold	IT_PROG	9000
Ernst	IT_PROG	6000
Austin	IT_PROG	4800
Pataballa	IT_PROG	4800
Lorentz	IT_PROG	4200
—	—	—
Fleaur	SH_CLERK	3100
—	—	—

last_name	job_id	salary
Hunold	IT_PROG	9000
Ernst	IT_PROG	6000
Austin	IT_PROG	4800
Pataballa	IT_PROG	4800
Lorentz	IT_PROG	4200
Taylor	SH_CLERK	3200
Fleaur	SH_CLERK	3100
Sullivan	SH_CLERK	2500
Geoni	SH_CLERK	2800
Sarchand	SH_CLERK	4200
---	---	---

- Write a query to display the last name of employees whose name have exactly 6 characters.

```
SELECT last_name FROM employees WHERE last_name LIKE '_____';
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
100	Steven	King	24000	90
101	Neena	Kochhar	17000	90
102	Lex	De Haan	17000	90
103	Alexander	Hunold	9000	60
104	Bruce	Ernst	6000	60
105	David	Austin	4800	60
106	Valli	Pataballa	4800	60
-	-	-	-	-

SELECT last_name FROM employees
WHERE last_name LIKE '_____';

whose last_names have exactly 6 characters

LAST_NAME
King
Kochhar
De Haan
Hunold
Ernst
Austin
Pataballa
-

last_name arranged in ascending order

last_name
Austin
Bissot
Cabrio
Davies
Faviet
Feeney
Fleur
Greene
Himuro
Hunold

- Write a query to display the last name of employees having 'e' as the third character.

SELECT last_name FROM employees WHERE last_name LIKE '__e%';

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
100	Steven	King	24000	90
101	Neena	Kochhar	17000	90
102	Lex	De Haan	17000	90
103	Alexander	Hunold	9000	60
104	Bruce	Ernst	6000	60
105	David	Austin	4800	60
106	Valli	Pataballa	4800	60
—	—	—	—	—
108	Nancy	Greenberg	12000	100
—	—	—	—	—

```
SELECT last_name
FROM employees
WHERE last_name LIKE '__e%' ;
```

last_name
Austin
De Haan
Ernst
Faviet
Greenberg
Hunold
King
Kochhar
Lorentz
Pataballa
—

last_name
Abel
Baer
Chen
Everett
Feeney
Fleur
Gee
Gietz
Greenberg
Greene
—

last_name arranged in ascending order

the last names of employees
having 'e' as the third character

- Write a query to display the jobs/designations available in the employees table.

```
SELECT DISTINCT job_id FROM employees;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	Steven	King	AD_PRES	90
101	Neena	Kochhar	AD_VP	90
102	Lex	De Haan	AD_VP	90
103	Alexander	Hunold	IT_PROG	60
104	Bruce	Ernst	IT_PROG	60
105	David	Austin	IT_PROG	60
106	Valli	Pataballa	IT_PROG	60
—	—	—	—	—

**SELECT DISTINCT job_id
FROM employees;**

job_id
AD_PRES
AD_VP
AD_VP
IT_PROG
FI_MGR
FI_ACCOUNT

job_id
AC_ACCOUNT
AC_MGR
AD_ASST
AD_PRES
AD_VP
FI_ACCOUNT
FI_MGR
HR_REP
IT_PROG
MK_MAN

job_id arranged in distincting

- Write a query to select all records from employees where last name in 'JONES', 'BLAKE', 'SCOTT', 'KING' and 'FORD'.

```
SELECT *
FROM employees
WHERE last_name IN('JONES', 'BLAKE', 'SCOTT', 'KING', 'FORD');
```



EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
100	Steven	King	24000	90
101	Neena	Kochhar	17000	90
102	Lex	De Haan	17000	90
103	Alexander	Hunold	9000	60
104	Bruce	Ernst	6000	60
105	David	Austin	4800	60
106	Valli	Pataballa	4800	60
—	—	—	—	—
156	Janette	King	10000	80
—	—	—	—	—

SELECT

FROM employees

WHERE last_name IN('JONES', 'BLAKE', 'SCOTT', 'KING', 'FOR')

last_name IN('JONES', 'BLAKE', 'SCOTT', 'KING', 'FORD')



LAST_NAME	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MGR_ID
King	195	Vance	Jones	VJONES	650.501.4876	17-Mar-07	SH_CLERK	2800	0	123
Kochhar	156	Janette	King	JKING	011.44.1345.429268	30-Jan-04	SA_REP	10000	0.35	146
De Haan	100	Steven	King	SKING	515.123.4567	17-Jun-03	AD_PRES	24000	0	0
Hunold										
Ernst										
Austin										
Pataballa										
—										
King										
—										



Aggregate Functions and Group By: Employees Table

- Write a query to list the number of jobs available in the employees table.

```
SELECT COUNT(DISTINCT job_id)
FROM employees;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	Steven	King	AD_PRES	90
101	Neena	Kochhar	AD_VP	90
102	Lex	De Haan	AD_VP	90
103	Alexander	Hunold	IT_PROG	60
104	Bruce	Ernst	IT_PROG	60
105	David	Austin	IT_PROG	60
106	Valli	Pataballa	IT_PROG	60
107	Diana	Lorentz	IT_PROG	60
108	Nancy	Greenberg	FI_MGR	100
-	-	-	-	-

SELECT COUNT(DISTINCT job_id)
FROM employees;

job_id
AC_ACCOUNT
AC_MGR
AD_ASST
AD_PRES
AD_VP
FI_ACCOUNT
FI_MGR
HR REP
IT_PROG
MK_MAN
MK_REP
PR_REP
PU_CLERK
PU_MAN
SA_MAN
SA_REP
SH_CLERK
ST_CLERK
ST_MAN

number distinct jobs available

COUNT(DISTINCT job_id)

19

- Write a query to get the minimum salary from employees table.

```
SELECT MIN(salary)
FROM employees;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
100	Steven	King	24000	90
101	Neena	Kochhar	17000	90
102	Lex	De Haan	17000	90
103	Alexander	Hunold	9000	60
104	Bruce	Ernst	6000	60
105	David	Austin	4800	60
—	—	—	—	—
132	TJ	Olson	2100	50
—	—	—	—	—

SELECT MIN(salary)
FROM employees;

SALARY
24000
17000
17000
9000
6000
4800
—
2100
—

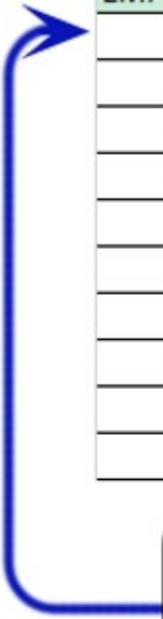
MIN(salary)

2100

Minimum Salary

- Write a query to get the maximum salary of an employee working as a Programmer.

```
SELECT MAX(salary)
FROM employees
WHERE job_id = 'IT_PROG';
```



EMPLOYEE_ID	FIRST_NAME	LAST_NAME		JOB_ID	SALARY		DEPARTMENT
100	Steven	King	-	AD_PRES	24000	-	90
101	Neena	Kochhar	-	AD_VP	17000	-	90
102	Lex	De Haan	-	AD_VP	17000	-	90
103	Alexander	Hunold	-	IT_PROG	9000	-	60
104	Bruce	Ernst	-	IT_PROG	6000	-	60
105	David	Austin	-	IT_PROG	4800	-	60
106	Valli	Pataballa	-	IT_PROG	4800	-	60
107	Diana	Lorentz	-	IT_PROG	4200	-	60
108	Nancy	Greenberg	-	FI_MGR	12000	-	100
-	-	-	-	-	-	-	-

```
SELECT MAX(salary)
FROM employees
WHERE job_id = 'IT_PROG' ;
```

JOB_ID	SALARY
AD_PRES	24000
AD_VP	17000
AD_VP	17000
IT_PROG	9000
IT_PROG	6000
IT_PROG	4800
IT_PROG	4800
IT_PROG	4200
FI_MGR	12000
-	-

MAX(salary)
9000

maximum salary

- Write a query to get the average salary and number of employees working the department 90.

```
SELECT AVG(salary), count(*)
FROM employees
WHERE department_id = 90;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
100	Steven	King	24000	90
101	Neena	Kochhar	17000	90
102	Lex	De Haan	17000	90
103	Alexander	Hunold	9000	60
104	Bruce	Ernst	6000	60
105	David	Austin	4800	60
106	Valli	Pataballa	4800	60
107	Diana	Lorentz	4200	60
108	Nancy	Greenberg	12000	100
-	-	-	-	-

```
SELECT AVG(salary),count(*)
FROM employees
WHERE department_id = 90 ;
```

SALARY	DEPARTMENT_ID
24000	90
17000	90
17000	90
9000	60
6000	60
4800	60
4800	60
4200	60
12000	100
-	-

Average salary and number of employees working in the department ID 90

1
2
3

AVG(salary)	count(*)
19333.33333	3

- Write a query to get the highest, lowest, sum, and average salary of all employees.

```
SELECT ROUND(MAX(salary),0) 'Maximum',
ROUND(MIN(salary),0) 'Minimum',
ROUND(SUM(salary),0) 'Sum',
ROUND(AVG(salary),0) 'Average'
FROM employees;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
100	Steven	King	24000	90
101	Neena	Kochhar	17000	90
102	Lex	De Haan	17000	90
103	Alexander	Hunold	9000	60
104	Bruce	Ernst	6000	60
105	David	Austin	4800	60
106	Valli	Pataballa	4800	60
107	Diana	Lorentz	4200	60
108	Nancy	Greenberg	12000	100
-	-	-	-	-

```
SELECT ROUND(MAX(salary),0) "Maximum",
       ROUND(MIN(salary),0) "Minimum",
       ROUND(SUM(salary),0) "Sum",
       ROUND(AVG(salary),0) "Average"
  FROM employees;
```

Maximum	Minimum	Sum	Average
24000	2100	691400	6462

- Write a query to get the number of employees with the same job.

```
SELECT job_id, COUNT(*)
  FROM employees
 GROUP BY job_id;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	Steven	King	AD_PRES	90
101	Neena	Kochhar	AD_VP	90
102	Lex	De Haan	AD_VP	90
103	Alexander	Hunold	IT_PROG	60
104	Bruce	Ernst	IT_PROG	60
105	David	Austin	IT_PROG	60
106	Valli	Pataballa	IT_PROG	60
107	Diana	Lorentz	IT_PROG	60
108	Nancy	Greenberg	FI_MGR	100

```
SELECT job_id, COUNT(*)
FROM employees
GROUP BY job_id;
```

job_id have been arranged as a group

job_id
AD_PRES
AD_VP
AD_VP
IT_PROG

1

2

5

job_id	COUNT(*)
AC_ACCOUNT	1
AC_MGR	1
AD_ASST	1
AD_PRES	1
AD_VP	2
FI_ACCOUNT	5
FI_MGR	1
HR_REP	1
IT_PROG	5
MK_MAN	1
MK_REP	1
PR_REP	1
PU_CLERK	5

- Write a query to get the difference between the highest and lowest salaries.

```
SELECT MAX(salary) - MIN(salary) DIFFERENCE
FROM employees;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
100	Steven	King	24000	90
101	Neena	Kochhar	17000	90
102	Lex	De Haan	17000	90
103	Alexander	Hunold	9000	60
104	Bruce	Ernst	6000	60
105	David	Austin	4800	60
106	Valli	Pataballa	4800	60
—	—	—	—	—
132	TJ	Olson	2100	50
—	—	—	—	—

**SELECT MAX(salary) - MIN(salary) DIFFERENCE
FROM employees;**

salary
24000
17000
17000
9000

2100

→ Maximum Salary

DIFFERENCE

21900

→ Minimum Salary

- Write a query to find the manager ID and the salary of the lowest-paid employee for that manager.

```
SELECT manager_id, MIN(salary)
FROM employees
WHERE manager_id IS NOT NULL
GROUP BY manager_id
ORDER BY MIN(salary) DESC;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	MANAGER_ID	DEPARTMENT
100	Steven	King	24000	0	90
101	Neena	Kochhar	17000	100	90
102	Lex	De Haan	17000	100	90
103	Alexander	Hunold	9000	102	60
104	Bruce	Ernst	6000	103	60
105	David	Austin	4800	103	60
106	Valli	Pataballa	4800	103	60
107	Diana	Lorentz	4200	103	60
108	Nancy	Greenberg	12000	101	100
—	—	—	—	—	—

```
SELECT manager_id, MIN(salary)
FROM employees
WHERE manager_id IS NOT NULL
GROUP BY manager_id
ORDER BY MIN(salary) DESC ;
```

manager_id have arranged in a group
and minimum salaries have arranged in descending order

MANAGER_ID	SALARY
0	24000
100	17000
100	17000
102	9000
103	6000
103	4800
103	4800
103	4200

manager_id	MIN(salary)
0	24000
102	9000
205	8300
145	7000
146	7000
108	6900
147	6200
149	6200
148	6100

- Write a query to get the department ID and the total salary payable in each department.

```
SELECT department_id, SUM(salary)
FROM employees
GROUP BY department_id;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME		SALARY	—	DEPARTMENT_ID
100	Steven	King		24000	—	90
101	Neena	Kochhar		17000	—	90
102	Lex	De Haan		17000	—	90
103	Alexander	Hunold		9000	—	60
104	Bruce	Ernst		6000	—	60
105	David	Austin		4800	—	60
106	Valli	Pataballa		4800	—	60
107	Diana	Lorentz		4200	—	60
108	Nancy	Greenberg		12000	—	100
—	—	—	—	—	—	—

```
SELECT department_id, SUM(salary)
FROM employees
GROUP BY department_id ;
```

DEPARTMENT_ID	SALARY
90	24000
90	17000
90	17000
60	9000
60	6000
60	4800
60	4800
60	4200

58000

department_id	SUM(salary)
0	7000
10	4400
20	19000
30	24900
40	6500
50	156400
60	28800
70	10000
80	304500
90	58000
100	51600

- Write a query to get the average salary for each job ID excluding programmer.

```
SELECT job_id, AVG(salary)
FROM employees
WHERE job_id <> 'IT_PROG'
GROUP BY job_id;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME		JOB_ID	SALARY		DEPARTMENT_ID
100	Steven	King	—	AD_PRES	24000	—	90
101	Neena	Kochhar	—	AD_VP	17000	—	90
102	Lex	De Haan	—	AD_VP	17000	—	90
103	Alexander	Hunold	—	IT_PROG	9000	—	60
104	Bruce	Ernst	—	IT_PROG	6000	—	60
105	David	Austin	—	IT_PROG	4800	—	60
106	Valli	Pataballa	—	IT_PROG	4800	—	60
107	Diana	Lorentz	—	IT_PROG	4200	—	60
108	Nancy	Greenberg	—	FI_MGR	12000	—	100
—	—	—	—	—	—	—	—

SELECT job_id, AVG(salary)

FROM employees

WHERE job_id <> 'IT_PROG'

GROUP BY job_id ;

job_id have been arranged
in a group

JOB_ID	SALARY
AD_PRES	24000
AD_VP	17000
AD_VP	17000
IT_PROG	9000
IT_PROG	6000
IT_PROG	4800
IT_PROG	4800
IT_PROG	4200
FI_MGR	12000

job_id	AVG(salary)
AC_ACCOUNT	8300
AC_MGR	12000
AD_ASST	4400
AD_PRES	24000
AD_VP	17000
FI_ACCOUNT	7920
FI_MGR	12000
HR_REP	6500

- Write a query to get the total salary, maximum, minimum, average salary of employees (job ID wise), for department ID 90 only.

```
SELECT job_id, SUM(salary), AVG(salary), MAX(salary), MIN(salary)
FROM employees
```

```
WHERE department_id = '90'
GROUP BY job_id;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME		JOB_ID	SALARY		DEPARTMENT_ID
100	Steven	King	—	AD_PRES	24000	—	90
101	Neena	Kochhar	—	AD_VP	17000	—	90
102	Lex	De Haan	—	AD_VP	17000	—	90
103	Alexander	Hunold	—	IT_PROG	9000	—	60
104	Bruce	Ernst	—	IT_PROG	6000	—	60
105	David	Austin	—	IT_PROG	4800	—	60
106	Valli	Pataballa	—	IT_PROG	4800	—	60
107	Diana	Lorentz	—	IT_PROG	4200	—	60
108	Nancy	Greenberg	—	FI_MGR	12000	—	100
—	—	—	—	—	—	—	—

```
SELECT job_id, SUM(salary), AVG(salary),
MAX(salary), MIN(salary)
FROM employees
WHERE department_id = '90'
GROUP BY job_id ;
```

job have been arranged in a group
for department_id 90

JOB_ID	SALARY	DEPARTMENT_ID
AD_PRES	24000	90
AD_VP	17000	90
AD_VP	17000	90
IT_PROG	9000	60

job_id	SUM(salary)	AVG(salary)	MAX(salary)	MIN(salary)
AD_PRES	24000	24000	24000	24000
AD_VP	34000	17000	17000	17000

- Write a query to get the job ID and maximum salary of the employees where maximum salary is greater than or equal to \$4000.

```
SELECT job_id, MAX(salary)
FROM employees
GROUP BY job_id
HAVING MAX(salary) >= 4000;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME		JOB_ID	SALARY		DEPARTMENT_ID
100	Steven	King	—	AD_PRES	24000	—	90
101	Neena	Kochhar	—	AD_VP	17000	—	90
102	Lex	De Haan	—	AD_VP	17000	—	90
103	Alexander	Hunold	—	IT_PROG	9000	—	60
104	Bruce	Ernst	—	IT_PROG	6000	—	60
105	David	Austin	—	IT_PROG	4800	—	60
106	Valli	Pataballa	—	IT_PROG	4800	—	60
107	Diana	Lorentz	—	IT_PROG	4200	—	60
108	Nancy	Greenberg	—	FI_MGR	12000	—	100
—	—	—	—	—	—	—	—

```
SELECT job_id, MAX(salary)
FROM employees
GROUP BY job_id
HAVING MAX(salary) >=4000 ;
```

job_id have been arranged
in a group

JOB_ID	SALARY
AD PRES	24000
AD VP	17000
AD VP	17000
IT PROG	9000
IT PROG	6000
IT PROG	4800
IT PROG	4800
IT PROG	4200
FI MGR	12000

each are
>=4000

job_id	MAX(salary)
AC_ACCOUNT	8300
AC_MGR	12000
AD_ASST	4400
AD_PRES	24000
AD_VP	17000
FI_ACCOUNT	9000
FI_MGR	12000

- Write a query to get the average salary for all departments employing more than 10 employees.

```
SELECT department_id, AVG(salary), COUNT(*)
FROM employees
GROUP BY department_id
HAVING COUNT(*) > 10;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME		JOB_ID	SALARY		DEPARTMENT_ID
100	Steven	King	—	AD_PRES	24000	—	90
101	Neena	Kochhar	—	AD_VP	17000	—	90
—	—	—	—	—	—	—	—
120	Matthew	Weiss	—	ST_MAN	8000	—	50
121	Adam	Fripp	—	ST_MAN	8200	—	50
122	Payam	Kaufling	—	ST_MAN	7900	—	50
123	Shanta	Vollman	—	ST_MAN	6500	—	50
124	Kevin	Mourgos	—	ST_MAN	5800	—	50
—	—	—	—	—	—	—	—

```

SELECT department_id, AVG(salary), COUNT(*)
FROM employees
GROUP BY department_id
HAVING COUNT(*) > 10;

```

department_id have been arranged
in a group with average salary
and number of employees in each group

department_id	avg(salary)	count(*)
0	7000.00	1
10	4400.00	1
20	9500.00	2
3	4150.00	6
50	3475.555556	45
50	3475.55	45
60	5760.00	5
70	10000.00	1
80	8955.88235	34
90	19333.33	3
100	8600.00	6
110	10150.00	2

having count(*) > 10

department_id	avg(salary)	count(*)
50	3475.55	45
80	8955.88	34