SQL Queries

Create a database worker that should contain **first name, last name email, department, salary, Join Date** with 50 employees.

Project_1 Database created

```
3 •  CREATE DATABASE project_1;
4 •  SHOW DATABASES;
5
```

Result Grid | Filter Rows: | Export:

Database
employees
information_schema
mysql
performance_schema
project_1
sales
sys

Table employees created

```
4
5 •  Select * from employees;
6
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	HIRE_DATE
101	Donald	OConnell	DOCONNEL	PRODUCTION	2600	6/21/2007
102	Douglas	Grant	DGRANT	PRODUCTION	2600	1/13/2008
103	Jennifer	Whalen	JWHALEN	HR	4400	9/17/2003
104	Michael	Hartstein	MHARTSTE	ANALYTICS	13000	2/17/2004
105	Pat	Fay	PFAY	TESTING	6000	8/17/2005
106	Susan	Mavris	SMAVRIS	TESTING	6500	6/7/2002
107	Hermann	Baer	HBAER	TESTING	10000	6/7/2002
108	Shelley	Higgins	SHIGGINS	HR	12008	6/7/2002
109	William	Gietz	WGIETZ	ACCOUNTS	8300	6/7/2002
110	Steven	King	SKING	HR	24000	6/17/2003
111	Neena	Kochhar	NKOCHHAR	TESTING	17000	9/21/2005
112	Lex	De Haan	LDEHAAN	TESTING	17000	1/13/2001
113	Alexander	Hunold	AHUNOLD	HR	9000	1/3/2006
114	Bruce	Ernst	BERNST	HR	6000	5/21/2007
115	David	Austin	DAUSTIN	HR	4800	6/25/2005

Task-1

1. Write an SQL query to fetch “FIRST_NAME” from the Worker table using the alias name as <WORKER_NAME>.

```
4
5 •  Select FIRST_NAME as WORKER_NAME from employees;
6
```

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

WORKER_NAME
Donald
Douglas
Jennifer
Michael
Pat
Susan
Hermann
Shelley
William
Steven
Neena

2. Write an SQL query to fetch unique values of DEPARTMENT from the Worker table.

```
5 •  SELECT DISTINCT DEPARTMENT FROM employees;
6
```

Result Grid | Filter Rows: | Export: | Wrap Cel

DEPARTMENT
PRODUCTION
HR
ANALYTICS
TESTING
ACCOUNTS

3. Write an SQL query to show the last 5 records from a table.

```
5 • (SELECT * FROM employees
6    ORDER BY EMPLOYEE_ID DESC
7    LIMIT 5)
8    ORDER BY EMPLOYEE_ID ASC;
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	HIRE_DATE
146	Hazel	Philtanker	HPHILTAN	PRODUCTION	2200	2/6/2008
147	Renske	Ladwig	RLADWIG	PRODUCTION	3600	7/14/2003
148	Stephen	Stiles	SSTILES	PRODUCTION	3200	10/26/2005
149	John	Seo	JSEO	PRODUCTION	2700	2/12/2006
150	Joshua	Patel	JPATEL	PRODUCTION	2500	4/6/2006

Task-2

1. Write an SQL query to print the first three characters of FIRST_NAME from Worker.

```
5 •  SELECT LEFT(FIRST_NAME,3) AS FIRST_NAME FROM employees;
6
```

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

FIRST_NAME
Don
Dou
Jen
Mic
Pat
Sus
Her
She
Wil
Ste
Nee
Lex
Ale
Bru

2. Write an SQL query to find the position of the alphabet (‘a’) in the first name

```
5 •   SELECT FIRST_NAME, POSITION('a'in FIRST_NAME) AS POSITION_OF_a FROM employees;
6
```

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

FIRST_NAME	POSITION_OF_a
Donald	4
Douglas	6
Jennifer	0
Michael	5
Pat	2
Susan	4
Hermann	5
Shelley	0
William	6
Steven	0

3. Write an SQL query to print the name of employees who have the highest salary in each department.

```
5 •   SELECT * FROM
6     (SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, DEPARTMENT, SALARY,
7     RANK() OVER(
8     PARTITION BY DEPARTMENT
9     ORDER BY salary desc) as HIGHEST_SALARY from employees) employees
10    WHERE HIGHEST_SALARY = 1;
11
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT	SALARY	HIGHEST_SALARY
119	Daniel	Faviet	ACCOUNTS	9000	1
104	Michael	Hartstein	ANALYTICS	13000	1
110	Steven	King	HR	24000	1
147	Renske	Ladwig	PRODUCTION	3600	1
111	Neena	Kochhar	TESTING	17000	1
112	Lex	De Haan	TESTING	17000	1

Task-3

1. Write an SQL query to print the FIRST_NAME from the Worker table after removing white spaces from the right side.

```
7 •   SELECT RTRIM(FIRST_NAME) AS FIRST_NAME FROM employees;
8
```

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

FIRST_NAME
Donald
Douglas
Jennifer
Michael
Pat
Susan
Hermann
Shelley
William
Steven
Neena
Lex

2. Write an SQL query that fetches the unique values of DEPARTMENT from the Worker table and prints its length.

```
7 •   SELECT DISTINCT DEPARTMENT, LENGTH(DEPARTMENT) AS LEN_DEPARTMENT FROM employees;
8
```

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

DEPARTMENT	LEN_DEPARTMENT
PRODUCTION	10
HR	2
ANALYTICS	9
TESTING	7
ACCOUNTS	8

3. Write an SQL query to fetch nth max salaries from a table.

```
58 •  SELECT EMPLOYEE_ID,FIRST_NAME, LAST_NAME, SALARY FROM employees
59    ORDER BY SALARY DESC
60    LIMIT 5;
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY
110	Steven	King	24000
111	Neena	Kochhar	17000
112	Lex	De Haan	17000
104	Michael	Hartstein	13000
118	Nancy	Greenberg	12008

Task-4

1. Write an SQL query to print the FIRST_NAME from the Worker table after replacing ‘a’ with ‘A’.

```
30 •  SELECT FIRST_NAME, REPLACE(FIRST_NAME,'a','A') AS FIRST_NAME_A FROM employees;
31
```

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

FIRST_NAME	FIRST_NAME_A
Donald	DonAld
Douglas	DouglAs
Jennifer	Jennifer
Michael	MichAel
Pat	PAt
Susan	SusAn
Hermann	HermAnn
Shelley	Shelley
William	WilliAm
Steven	Steven
Neena	NeenA
Lex	Lex

2. Write an SQL query to print all Worker details from the Worker table order FIRST_NAME Ascending and DEPARTMENT Descending.

```
35 •    SELECT * FROM employees
36      ORDER BY FIRST_NAME ASC, DEPARTMENT DESC;
37
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	HIRE_DATE
131	Adam	Fripp	AFRIPP	ANALYTICS	8200	4/10/2005
125	Alexander	Khoo	AKHOO	PRODUCTION	3100	5/18/2003
113	Alexander	Hunold	AHUNOLD	HR	9000	1/3/2006
114	Bruce	Ernst	BERNST	HR	6000	5/21/2007
119	Daniel	Faviet	DFAVIET	ACCOUNTS	9000	8/16/2002
115	David	Austin	DAUSTIN	HR	4800	6/25/2005
124	Den	Raphaely	DRAPHEAL	ANALYTICS	11000	12/7/2002
117	Diana	Lorentz	DLORENTZ	HR	4200	2/7/2007
101	Donald	OConnell	DOCONNEL	PRODUCTION	2600	6/21/2007
102	Douglas	Grant	DGRANT	PRODUCTION	2600	1/13/2008
128	Guy	Himuro	GHIMURO	PRODUCTION	2600	11/15/2006
146	Hazel	Philtanker	HPHILTAN	PRODUCTION	2200	2/6/2008
107	Hermann	Baer	HBAER	TESTING	10000	6/7/2002

3. Write an SQL query to fetch the names of workers who earn the highest salary.

```
41 •    SELECT FIRST_NAME, LAST_NAME, SALARY FROM employees
42      ORDER BY SALARY DESC;
43
```

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

FIRST_NAME	LAST_NAME	SALARY
Steven	King	24000
Neena	Kochhar	17000
Lex	De Haan	17000
Michael	Hartstein	13000
Nancy	Greenberg	12008
Shelley	Higgins	12008
Den	Raphaely	11000
Hermann	Baer	10000
Alexander	Hunold	9000
Daniel	Faviet	9000
William	Gietz	8300
Adam	Fripp	8200

Task-5

1. Write an SQL query to print details of workers excluding first names, “Pat” and “Donald” from the Worker table.

```
47 •  SELECT * FROM employees
48    WHERE FIRST_NAME NOT IN ('Pat','Donald');
49
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	HIRE_DATE
102	Douglas	Grant	DGRANT	PRODUCTION	2600	1/13/2008
103	Jennifer	Whalen	JWHALEN	HR	4400	9/17/2003
104	Michael	Hartstein	MHARTSTE	ANALYTICS	13000	2/17/2004
106	Susan	Mavris	SMAVRIS	TESTING	6500	6/7/2002
107	Hermann	Baer	HBAER	TESTING	10000	6/7/2002
108	Shelley	Higgins	SHIGGINS	HR	12008	6/7/2002
109	William	Gietz	WGIETZ	ACCOUNTS	8300	6/7/2002
110	Steven	King	SKING	HR	24000	6/17/2003
111	Neena	Kochhar	NKOCHHAR	TESTING	17000	9/21/2005
112	Lex	De Haan	LDEHAAN	TESTING	17000	1/13/2001
113	Alexander	Hunold	AHUNOLD	HR	9000	1/3/2006

2. Write an SQL query to print details of the Workers whose FIRST_NAME ends with ‘a’ and contains six alphabets.

```
53 •  SELECT * FROM employees
54    WHERE FIRST_NAME LIKE ('%____a');
55
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	HIRE_DATE
133	Shanta	Vollman	SVOLLMAN	ANALYTICS	6500	10/10/2005
150	Joshua	Patel	JPATEL	PRODUCTION	2500	4/6/2006

3. Write a query to validate Email of Employee (email should have first name last name and guvi.com example (first name=Kamal last name= raja and the mail id should be kamalraja@guvi.com).

```
70 •    SELECT * FROM employees_dup
71      WHERE (EMAIL REGEXP CONCAT(LOWER(RTRIM(FIRST_NAME)), LOWER(LAST_NAME), '@guvi.com'));
72
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	HIRE_DATE
152	Ram	Kumar	ramkumar@guvi.com	TESTING	3000	9/17/2005
153	Alok	Das	alokdas@guvi.com	HR	2000	10/17/2007
154	Jenny	Fer	jennyfer@guvi.com	PRODUCTION	9000	2/17/2009
155	Lucas	Linda	lucaslinda@guvi.com	ACCOUNTS	11000	4/17/2006
156	Ajay	Khanna	ajaykhanna@guvi.com	ANALYTICS	14000	5/17/2004

Task-6

1. Write an SQL query to print details of the Workers who have joined in March ’2021.

```
70 •    SELECT * FROM employees
71      WHERE HIRE_DATE BETWEEN '2002-08-01' AND  '2002-08-31';
72
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	HIRE_DATE
118	Nancy	Greenberg	NGREENBE	HR	12008	2002-08-17 00:00:00
119	Daniel	Faviet	DFAVIET	ACCOUNTS	9000	2002-08-16 00:00:00

2. Write an SQL query to fetch duplicates that have matching data in some fields of a table.

```
77 •  SELECT * , count(FIRST_NAME) AS COUNT_DUP FROM employees
78    GROUP BY FIRST_NAME
79    HAVING count(FIRST_NAME) > 1;
80
81
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	HIRE_DATE	COUNT_DUP
104	Michael	Hartstein	MHARTSTE	ANALYTICS	13000	2004-02-17 00:00:00	2
110	Steven	King	SKING	HR	24000	2003-06-17 00:00:00	2
113	Alexander	Hunold	AHUNOLD	HR	9000	2006-01-03 00:00:00	2
120	John	Chen	JCHEN	ACCOUNTS	8200	2005-09-28 00:00:00	2
137	James	Landry	JLANDRY	PRODUCTION	2400	2007-01-14 00:00:00	2

3. How to remove duplicate rows from the Employees table.

   NOTE: Removed 1 DUPLICATE RECORD

```
93    # Fetching duplicate records
94 •  SELECT EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, DEPARTMENT, SALARY, count(*) AS COUNT_DUPLICATE FROM employees_dup
95    GROUP BY FIRST_NAME, LAST_NAME, EMAIL, DEPARTMENT, SALARY
96    HAVING count(*) > 1 ;
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	COUNT_DUPLICATE
160	Pat	Fay	PFAY	TESTING	6000	2

```
101   # Deleting duplicate records
102 • DELETE FROM employees_dup
103   WHERE EMPLOYEE_ID NOT IN(
104       SELECT MAX(EMPLOYEE_ID) FROM (SELECT * FROM employees_dup) AS dup
105       GROUP BY FIRST_NAME, LAST_NAME, EMAIL, DEPARTMENT, SALARY);
106
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	HIRE_DATE
141	James	Marlow	JAMRLOW	PRODUCTION	2500	2/16/2005
142	TJ	Olson	TJOLSON	PRODUCTION	2100	4/10/2007
143	Jason	Mallin	JMALLIN	PRODUCTION	3300	6/14/2004
144	Michael	Rogers	MROGERS	PRODUCTION	2900	8/26/2006
145	Ki	Gee	KGEE	PRODUCTION	2400	12/12/2007
146	Hazel	Philtanker	HPHILTAN	PRODUCTION	2200	2/6/2008
147	Renske	Ladwig	RLADWIG	PRODUCTION	3600	7/14/2003
148	Stephen	Stiles	SSTILES	PRODUCTION	3200	10/26/2005
149	John	Seo	JSEO	PRODUCTION	2700	2/12/2006
150	Joshua	Patel	JPATEL	PRODUCTION	2500	4/6/2006
151	Pat	Fay	PFAY	TESTING	6000	8/17/2005

Task-7

1. Write an SQL query to show only odd rows from a table.

```
112 •   SELECT * FROM employees
113     WHERE MOD(EMPLOYEE_ID,2) != 0;
114
115
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	HIRE_DATE
101	Donald	OConnell	DOCONNEL	PRODUCTION	2600	2007-06-21 00:00:00
103	Jennifer	Whalen	JWHALEN	HR	4400	2003-09-17 00:00:00
105	Pat	Fay	PFAY	TESTING	6000	2005-08-17 00:00:00
107	Hermann	Baer	HBAER	TESTING	10000	2002-06-07 00:00:00
109	William	Gietz	WGIETZ	ACCOUNTS	8300	2002-06-07 00:00:00
111	Neena	Kochhar	NKOCHHAR	TESTING	17000	2005-09-21 00:00:00
113	Alexander	Hunold	AHUNOLD	HR	9000	2006-01-03 00:00:00
115	David	Austin	DAUSTIN	HR	4800	2005-06-25 00:00:00
117	Diana	Lorentz	DLORENTZ	HR	4200	2007-02-07 00:00:00

2. Write an SQL query to clone a new table from another table.

```
119  CREATE TABLE `emp_hiring_detail` (
120    `EMPLOYEE_ID` int DEFAULT NULL,
121    `FIRST_NAME` text,
122    `LAST_NAME` text,
123    `EMAIL` text,
124    `DEPARTMENT` text,
125    `SALARY` int DEFAULT NULL,
126    `HIRE_DATE` datetime
127  );
128  Insert into emp_hiring_detail(
129  EMPLOYEE_ID,
130    FIRST_NAME,
131    LAST_NAME,
132    EMAIL,
133    DEPARTMENT,
134    SALARY,
135    HIRE_DATE)
136  select * from  employees;
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	HIRE_DATE
101	Donald	OConnell	DOCONNEL	PRODUCTION	2600	2007-06-21 00:00:00
102	Douglas	Grant	DGRANT	PRODUCTION	2600	2008-01-13 00:00:00
103	Jennifer	Whalen	JWHALEN	HR	4400	2003-09-17 00:00:00
104	Michael	Hartstein	MHARTSTE	ANALYTICS	13000	2004-02-17 00:00:00
105	Pat	Fay	PFAY	TESTING	6000	2005-08-17 00:00:00
106	Susan	Mavris	SMAVRIS	TESTING	6500	2002-06-07 00:00:00
107	Hermann	Baer	HBAER	TESTING	10000	2002-06-07 00:00:00
108	Shelley	Higgins	SHIGGINS	HR	12008	2002-06-07 00:00:00

emp_hiring_detail 45

Task-8

1. Write an SQL query to fetch intersecting records of two tables.

NOTE: Created 2 new Tables. “emp_hiring_detail” and “emp_salary”

```
146 •   Select * from emp_salary;
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	SALARY
101	Donald	OConnell	DOCONNEL	2600
102	Douglas	Grant	DGRANT	2600
103	Jennifer	Whalen	JWHALEN	4400
104	Michael	Hartstein	MHARTSTE	13000
105	Pat	Fay	PFAY	6000

```
148 •   Select * from emp_hiring_detail;
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT	HIRE_DATE
101	Donald	OConnell	PRODUCTION	2007-06-21 00:00:00
102	Douglas	Grant	PRODUCTION	2008-01-13 00:00:00
103	Jennifer	Whalen	HR	2003-09-17 00:00:00
104	Michael	Hartstein	ANALYTICS	2004-02-17 00:00:00
105	Pat	Fay	TESTING	2005-08-17 00:00:00

Fetching the intersecting records between the two:

```
158 •   SELECT es.EMPLOYEE_ID, es.SALARY, eh.DEPARTMENT FROM emp_salary AS es
159     INNER JOIN emp_hiring_detail AS eh
160     ON es.EMPLOYEE_ID = eh.EMPLOYEE_ID;
161
```

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

EMPLOYEE_ID	SALARY	DEPARTMENT
101	2600	PRODUCTION
102	2600	PRODUCTION
103	4400	HR
104	13000	ANALYTICS
105	6000	TESTING
106	6500	TESTING
107	10000	TESTING
108	12008	HR

2. Write an SQL query to show records from one table that another table does not have.

```
165 •   SELECT eh.FIRST_NAME, es.SALARY FROM emp_salary AS es
166     RIGHT JOIN emp_hiring_detail AS eh
167     ON eh.EMPLOYEE_ID = es.EMPLOYEE_ID;
168
```

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

FIRST_NAME	SALARY
Michael	2900
Ki	2400
Hazel	2200
Renske	3600
Stephen	3200
John	2700
Joshua	2500
Dan	NULL
Garry	NULL

Task-9

1. Write an SQL query to show the top n (say 15) records of a table.

```
172 •   SELECT * FROM employees
173     LIMIT 15;
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	HIRE_DATE
101	Donald	OConnell	DOCONNEL	PRODUCTION	2600	2007-06-21 00:00:00
102	Douglas	Grant	DGRANT	PRODUCTION	2600	2008-01-13 00:00:00
103	Jennifer	Whalen	JWHALEN	HR	4400	2003-09-17 00:00:00
104	Michael	Hartstein	MHARTSTE	ANALYTICS	13000	2004-02-17 00:00:00
105	Pat	Fay	PFAY	TESTING	6000	2005-08-17 00:00:00
106	Susan	Mavris	SMAVRIS	TESTING	6500	2002-06-07 00:00:00
107	Hermann	Baer	HBAER	TESTING	10000	2002-06-07 00:00:00
108	Shelley	Higgins	SHIGGINS	HR	12008	2002-06-07 00:00:00
109	William	Gietz	WGIETZ	ACCOUNTS	8300	2002-06-07 00:00:00
110	Steven	King	SKING	HR	24000	2003-06-17 00:00:00
111	Neena	Kochhar	NKOCHHAR	TESTING	17000	2005-09-21 00:00:00
112	Lex	De Haan	LDEHAAN	TESTING	17000	2001-01-13 00:00:00
113	Alexander	Hunold	AHUNOLD	HR	9000	2006-01-03 00:00:00
114	Bruce	Ernst	BERNST	HR	6000	2007-05-21 00:00:00
115	David	Austin	DAUSTIN	HR	4800	2005-06-25 00:00:00

2. Write an SQL query to determine the nth (say n=10) highest salary from a table.

```
177 •  SELECT * FROM
178        (SELECT * FROM employees
179        ORDER BY SALARY DESC
180        LIMIT 10) as Nth_highest
181    ORDER BY SALARY ASC
182    LIMIT 1;
183
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	HIRE_DATE
113	Alexander	Hunold	AHUNOLD	HR	9000	2006-01-03 00:00:00

Task-10

1. Write an SQL query to determine the 8th highest salary without using TOP or LIMIT methods.

```
186 •  SELECT * FROM(
187        SELECT EMPLOYEE_ID, FIRST_NAME, SALARY,
188        DENSE_RANK() OVER(
189        ORDER BY salary DESC) AS HIGHEST_SALARY FROM emp_salary) emp_salary
190    WHERE HIGHEST_SALARY = 8;
191
```

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

EMPLOYEE_ID	FIRST_NAME	SALARY	HIGHEST_SALARY
109	William	8300	8

2. Write an SQL query to fetch the list of employees with the same salary.

```
195 •   SELECT * , COUNT(SALARY) AS COUNT_DUP FROM emp_salary
196     GROUP BY SALARY
197     HAVING COUNT(SALARY) > 1;
198
```

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

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	SALARY	COUNT_DUP
101	Donald	OConnell	DOCONNEL	2600	3
105	Pat	Fay	PFAY	6000	2
106	Susan	Mavris	SMAVRIS	6500	2
108	Shelley	Higgins	SHIGGINS	12008	2
111	Neena	Kochhar	NKOCHHAR	17000	2
113	Alexander	Hunold	AHUNOLD	9000	2
115	David	Austin	DAUSTIN	4800	2
120	John	Chen	JCHEN	8200	2
126	Shelli	Baida	SBAIDA	2900	2
127	Sigal	Tobias	STOBIAS	2800	2
129	Karen	Colmenares	KCOLMENA	2500	3
135	Julia	Nayer	JNAYER	3200	2
136	Irene	Mikkilineni	IMIKKILI	2700	2
137	James	Landry	JLANDRY	2400	2
138	Steven	Markle	SMARKLE	2200	2
139	Laura	Bissot	LBISSOT	3300	2