

SQL ASSIGNMENTS

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

Worker Table:

The screenshot shows a database application interface. At the top, a toolbar includes icons for file operations, a search icon, and a 'Limit to 1000 rows' dropdown. Below the toolbar, a SQL editor contains the query: `SELECT * FROM Worker;`. To the right of the editor is an 'Autom help' button. Below the editor is a 'Result Grid' section. It features a toolbar with 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. The main area displays a table with 7 columns: ID, FIRST_NAME, LAST_NAME, EMAIL, DEPARTMENT, SALARY, and JOINDATE. The table contains 15 rows of employee data. To the right of the table is a vertical sidebar with buttons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. Below the table are 'Worker 1' and 'Apply' buttons. At the bottom, an 'Output' section shows 'Action Output' with a table containing one row: `SELECT * FROM Worker LIMIT 0, 1000`. A 'Message' box indicates '50 row(s) returned'.

ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	JOINDATE
1	Yogesh	Vaishnav	yogeshvaishnav@guvi.com	Accounts	30000	2019-10-03
2	Vishal	Vishwakarma	vishalvishwakarma@guvi.com	IT	20000	2019-11-07
3	Ajit	Yadav	ajityadav@guvi.com	Admin	15000	2019-12-12
4	Ashish	Yadav	ashishyadav@guvi.com	HR	10000	2019-12-25
5	Tanvi	Thakur	tanvithakur@guvi.com	IT	25000	2020-01-20
6	Sam	Parker	samparker@guvi.com	IT	20000	2020-03-03
7	Ron	Wesley	ronwesley@guvi.com	Accounts	25000	2020-05-16
8	Sara	Rogers	sararogers@guvi.com	HR	20000	2020-07-01
9	Yoji	Seo	yojiseo@guvi.com	Admin	20000	2020-03-10
10	Zara	Vargas	zaravargas@guvi.com	IT	30000	2020-08-20
11	Steven	King	stevenking@guvi.com	IT	25000	2020-09-25
12	Bruce	Ernst	bruceernst@guvi.com	HR	20000	2019-06-21
13	David	Austin	davidAustin@guvi.com	IT	40000	2020-07-10
14	Diana	Lorentz	dianalorentz@guvi.com	HR	10000	2021-08-03
15	Nancv	Greenbera	nancvreenbera@guvi.com	IT	25000	2020-01-20

#	Time	Action	Message
1	20:38:58	SELECT * FROM Worker LIMIT 0, 1000	50 row(s) returned

Task-1

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

The screenshot displays a database query tool interface. At the top, a query editor window titled "Query 1" contains the SQL statement: `SELECT FIRST_NAME AS WORKER_NAME FROM Worker;`. Below the editor, a toolbar includes icons for file operations, a "Limit to 1000 rows" dropdown, and other utility icons. The main area shows the "Result Grid" with a list of names: Yogesh, Vishal, Ajit, Ashish, Tanvi, Sam, Ron, Sara, Yoji, Zara, Steven, Bruce, and David. To the right of the result grid is a vertical toolbar with buttons for "Result Grid", "Form Editor", and "Field Types". At the bottom, an "Output" section shows the "Action Output" for the query execution, indicating that 50 rows were returned.

Query 1 x

SQLAddit

Limit to 1000 rows

1 • SELECT FIRST_NAME AS WORKER_NAME FROM Worker;

Aut

Result Grid

Filter Rows: [] Export: [] Wrap Cell Content: []

WORKER_NAME
Yogesh
Vishal
Ajit
Ashish
Tanvi
Sam
Ron
Sara
Yoji
Zara
Steven
Bruce
David

Worker 9 x Read Only Context H

Output

Action Output

#	Time	Action	Message
1	00:05:38	SELECT FIRST_NAME AS WORKER_NAME FROM Worker LIMIT 0, 1000	50 row(s) returned

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

The screenshot shows a database management interface. At the top, a tab labeled 'Query 1' contains the SQL query: `SELECT DISTINCT DEPARTMENT FROM Worker;`. Below the query editor, a 'Result Grid' displays the results of the query. The grid has a single column titled 'DEPARTMENT' with five rows: 'Accounts', 'IT', 'Admin', 'HR', and 'Operations'. To the right of the grid, a vertical toolbar contains icons for 'Result Grid', 'Form Editor', and 'Field Types'. Below the grid, a status bar indicates 'Worker 8' and 'Read Only'. At the bottom, an 'Output' section shows a log of actions, including the execution of the query and the message '5 row(s) returned'.

Query 1 x

SQLAddi

Limit to 1000 rows

1 • `SELECT DISTINCT DEPARTMENT FROM Worker;`

Au

Result Grid

Filter Rows: Export: Wrap Cell Content:

DEPARTMENT
Accounts
IT
Admin
HR
Operations

Form Editor

Field Types

Worker 8 x

Read Only

Context i

Output

Action Output

#	Time	Action	Message
1	00:04:16	SELECT DISTINCT DEPARTMENT FROM Worker LIMIT 0, 1000	5 row(s) returned

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

The screenshot shows a SQL query editor window titled "Query 1" with the following query:

```
1 (SELECT * FROM Worker ORDER BY ID DESC LIMIT 5)
2 ORDER BY ID ASC;
```

Below the query editor is a "Result Grid" showing the results of the query. The grid has columns: ID, FIRST_NAME, LAST_NAME, EMAIL, DEPARTMENT, SALARY, and JOINDATE. The results are as follows:

ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	JOINDATE
46	Harrison	Bloom	harrisonbloom@guvi.com	Accounts	30000	2020-04-02
47	Elizabeth	Bates	elizabethbates@guvi.com	HR	25000	2021-04-02
48	Alyssa	Hutton	alyssahutton@guvi.com	HR	15000	2021-04-02
49	Jack	Livingston	jacklivingston@guvi.com	IT	35000	2019-05-20
50	Somesh	Bandavath	someshbandavath@guvi.com	IT	25000	2019-05-20

At the bottom of the screenshot is an "Output" window showing the execution details of the query:

#	Time	Action	Message
1	00:06:57	(SELECT * FROM Worker ORDER BY ID DESC LIMIT 5) ORDER BY ID ASC	5 row(s) returned

Task-2

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

The screenshot shows the SQL Developer interface. The query editor at the top contains the following SQL query:

```
1 • Select substring(FIRST_NAME,1,3) from Worker;
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query. The results are as follows:

substring(FIRST_NAME,1,3)
Yog
Vis
Aji
Ash
Tan
Sam
Ron
Sar
Yoj
Zar
Ste
Bru
Nav

At the bottom of the interface, the 'Output' window shows the execution details:

#	Time	Action	Message
✓ 1	00:09:05	Select substring(FIRST_NAME,1,3) from Worker LIMIT 0, 1000	50 row(s) returned

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

The screenshot shows the SQLAdditor application interface. At the top, a tab labeled 'Query 1' is active. Below it, a toolbar contains various icons for file operations, execution, and settings. The main query editor displays the following SQL query:

```
1 Select INSTR(FIRST_NAME, BINARY'a') from Worker;
```

Below the query editor, the 'Result Grid' is visible, showing the results of the query. The grid has two columns: the first column contains the values of the INSTR function, and the second column contains the first names of the workers. The results are as follows:

INSTR(FIRST_NAME, BINARY'a')	FIRST_NAME
0	...
5	...
1	...
1	...
2	...
2	...
0	...
2	...
0	...
2	...
0	...
0	...

At the bottom of the interface, the 'Output' section is visible, showing the execution log. The log entry is as follows:

#	Time	Action	Message
1	00:11:32	Select INSTR(FIRST_NAME, BINARY'a') from Worker LIMIT 0, 1000	50 row(s) returned

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

The screenshot shows a SQL query editor window titled "Query 1" with a toolbar and a "Limit to 1000 rows" dropdown. The query is as follows:

```
1 SELECT t.DEPARTMENT,t.FIRST_NAME,t.SALARY from(SELECT max(SALARY) as
2 TotalSalary,DEPARTMENT from Worker group by DEPARTMENT) as TempNew
3 Inner Join Worker t on TempNew.DEPARTMENT=t.DEPARTMENT
4 and TempNew.TotalSalary=t.SALARY;
```

Below the query editor is the "Result Grid" tab, which displays the query results in a table with columns DEPARTMENT, FIRST_NAME, and SALARY. The results are:

DEPARTMENT	FIRST_NAME	SALARY
Accounts	Santhosh	40000
Operations	Shanta	30000
Admin	Kevin	40000
IT	Sarath	50000
HR	Danielle	40000

On the right side of the interface, there are buttons for "Result Grid", "Form Editor", and "Field Types". At the bottom, the "Output" window shows the "Action Output" for the query, indicating that 5 row(s) were returned.

Result 15 x Read Only Context Help

Output

Action Output

#	Time	Action	Message
1	00:19:51	SELECT t.DEPARTMENT,t.FIRST_NAME,t.SALARY from(SELECT max(SALARY) as TotalSalary,DEPARTME...	5 row(s) returned

Task-3

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

The screenshot shows the SQL Developer interface. The query editor at the top contains the following SQL query:

```
1 • SELECT RTRIM(FIRST_NAME) FROM Worker;
```

Below the query editor, the 'Result Grid' tab is active, displaying the results of the query. The results are as follows:

RTRIM(FIRST_NAME)
Yogesh
Vishal
Ajit
Ashish
Tanvi
Sam
Ron
Sara
Yoji
Zara
Steven
Bruce
David

At the bottom of the interface, the 'Output' window shows the 'Action Output' for the query execution:

#	Time	Action	Message
✓ 1	00:22:21	SELECT RTRIM(FIRST_NAME) FROM Worker LIMIT 0, 1000	50 row(s) returned

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

Query 1 x

SQLAdditions v

Limit to 1000 rows

1 • `SELECT DISTINCT length(DEPARTMENT) FROM Worker;`

Autom help

Result Grid

length(DEPARTMENT)
8
2
5
10

Filter Rows: Export: Wrap Cell Content:

Result Grid

Form Editor

Field Types

Result 17 x

Read Only

Context Help

Output

Action Output

#	Time	Action	Message
1	00:24:41	SELECT DISTINCT length(DEPARTMENT) FROM Worker LIMIT 0, 1000	4 row(s) returned

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

The screenshot shows a database query editor interface. The query editor window, titled 'Query 1', contains the following SQL query:

```
1 SELECT distinct Salary from worker a WHERE 5 >= (SELECT count(distinct Salary)
2 from worker b WHERE a.Salary <= b.Salary) order by a.Salary desc;
```

Below the query editor, the 'Result Grid' is displayed, showing the results of the query. The grid has a single column labeled 'Salary' and contains five rows of data:

Salary
50000
40000
35000
30000
25000

At the bottom of the interface, the 'Output' section shows the execution log. It includes a table with columns for '#', 'Time', 'Action', and 'Message'.

#	Time	Action	Message
1	00:45:26	SELECT distinct Salary from worker a WHERE 5 >= (SELECT count(distinct Salary) from worker b WHERE a.Sa...	5 row(s) returned

Task-4

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

The screenshot shows a SQL IDE interface. At the top, a tab labeled 'Query 1' contains the SQL query: `SELECT REPLACE(FIRST_NAME, 'a', 'A') FROM WORKER;`. The query is limited to 1000 rows. Below the query editor, the 'Result Grid' is displayed, showing the results of the query. The first column is labeled `REPLACE(FIRST_NAME, 'a', 'A')`. The results are as follows:

REPLACE(FIRST_NAME, 'a', 'A')
Yogesh
VishAl
Ajit
Ashish
TAnvi
SAm
Ron
SArA
Yoji
ZArA
Steven
Bruce
DAvid

Below the result grid, the 'Output' section is visible, showing the 'Action Output' for the query execution. The output indicates that the query was executed successfully at 00:35:32, returning 50 row(s).

#	Time	Action	Message
1	00:35:32	SELECT REPLACE(FIRST_NAME, 'a', 'A') FROM WORKER LIMIT 0, 1000	50 row(s) returned

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

Query 1 x

SQLAdditions

Limit to 1000 rows

1 SELECT * FROM WORKER ORDER BY FIRST_NAME ASC,DEPARTMENT DESC;

Auton
hel

Result Grid

ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	JOINDATE
22	Adam	Fripp	adamfripp@guvi.com	Accounts	30000	2021-03-01
3	Ajit	Yadav	ajityadav@guvi.com	Admin	15000	2019-12-12
39	Allan	McEwen	allanmcewen@guvi.com	IT	40000	2021-03-11
48	Alyssa	Hutton	alyssahutton@guvi.com	HR	15000	2021-04-02
45	Amit	Banda	amitbanda@guvi.com	IT	25000	2020-04-02
37	Andrew	Hall	andrewhall@guvi.com	Admin	30000	2019-05-20
4	Ashish	Yadav	ashishyadav@guvi.com	HR	10000	2019-12-25
12	Bruce	Ernst	bruceernst@guvi.com	HR	20000	2019-06-21
32	Curtis	Davies	curtisdavies@guvi.com	IT	17000	2021-03-02
43	Danielle	Green	daniellegreen@guvi.com	HR	40000	2019-07-01
13	David	Austin	davidaustin@guvi.com	IT	40000	2020-07-10
44	David	Lee	davidlee@guvi.com	IT	30000	2019-07-01
14	Diana	Lorentz	dianalorentz@guvi.com	HR	10000	2021-08-03

WORKER 24 x

Apply Revert Context Help

Output

Action Output

#	Time	Action	Message
1	00:40:19	SELECT * FROM WORKER ORDER BY FIRST_NAME ASC,DEPARTMENT DESC LIMIT 0, 1000	50 row(s) returned

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

The screenshot shows a database query tool interface. At the top, a tab labeled "Query 1" is active. Below the tab is a toolbar with various icons, including a "Limit to 1000 rows" dropdown. The main area contains the following SQL query:

```
1 • SELECT FIRST_NAME, SALARY from Worker WHERE SALARY=(SELECT max(SALARY) from Worker);
```

Below the query editor, there is a "Result Grid" section. It includes a "Filter Rows" input field, an "Export" button, and a "Wrap Cell Content" checkbox. The result grid displays the following data:

FIRST_NAME	SALARY
Sarath	50000

On the right side of the interface, there is a sidebar with the text "SQLAdditions:" at the top, followed by "Autom help". Below this, there are icons for "Result Grid", "Form Editor", and "Field Types". At the bottom of the sidebar, there are "Read Only" and "Context Help" links.

At the bottom of the window, there is an "Output" section. It shows a table with columns "#", "Time", "Action", and "Message". The first row of the output table is:

#	Time	Action	Message
1	00:47:29	SELECT FIRST_NAME, SALARY from Worker WHERE SALARY=(SELECT max(SALARY) from Worker) LIMIT ...	1 row(s) returned

Task-5

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

The screenshot shows a database application interface. At the top, a tab labeled "Query 1" is active. Below it, a toolbar contains various icons for file operations, execution, and settings. The SQL query editor displays the following query:

```
1 SELECT * FROM Worker WHERE FIRST_NAME not in ('Ramesh','Santhosh');
```

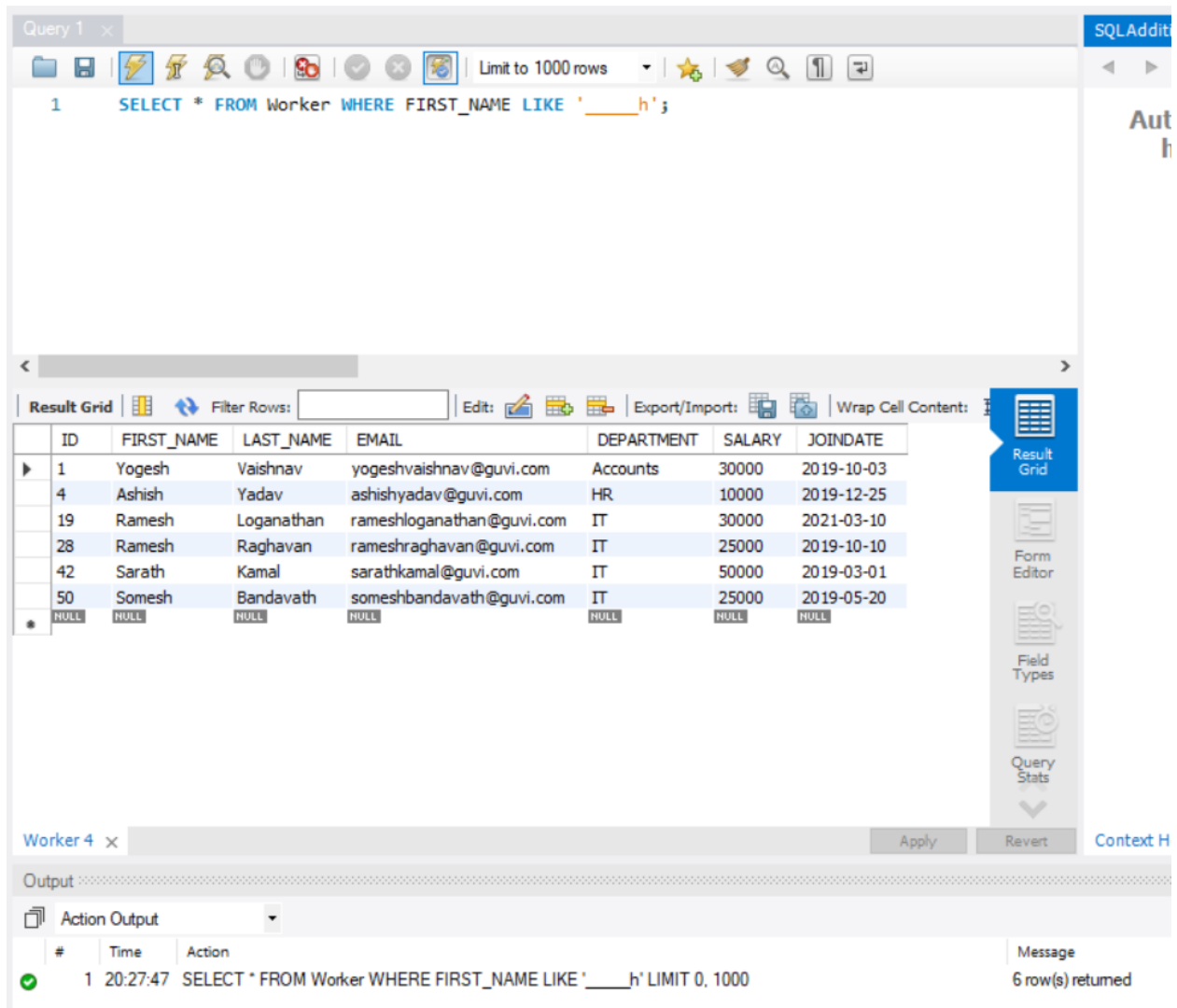
Below the query editor, a "Result Grid" tab is selected, showing a table with 7 columns: ID, FIRST_NAME, LAST_NAME, EMAIL, DEPARTMENT, SALARY, and JOINDATE. The table contains 14 rows of data. To the right of the table, a vertical toolbar offers options like "Result Grid", "Form Editor", "Field Types", and "Query Stats".

At the bottom, an "Output" section shows the execution log. It includes a table with columns for status, time, action, and message. The log shows a successful execution of the query at 20:23:54, returning 47 rows.

ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	JOINDATE
1	Yogesh	Vaishnav	yogeshvaishnav@guvi.com	Accounts	30000	2019-10-03
2	Vishal	Vishwakarma	vishalvishwakarma@guvi.com	IT	20000	2019-11-07
3	Ajit	Yadav	ajityadav@guvi.com	Admin	15000	2019-12-12
4	Ashish	Yadav	ashishyadav@guvi.com	HR	10000	2019-12-25
5	Tanvi	Thakur	tanvithakur@guvi.com	IT	25000	2020-01-20
6	Sam	Parker	samparker@guvi.com	IT	20000	2020-03-03
7	Ron	Wesley	ronwesley@guvi.com	Accounts	25000	2020-05-16
8	Sara	Rogers	sararogers@guvi.com	HR	20000	2020-07-01
9	Yoji	Seo	yojiseo@guvi.com	Admin	20000	2020-03-10
10	Zara	Vargas	zaravargas@guvi.com	IT	30000	2020-08-20
11	Steven	King	stevenking@guvi.com	IT	25000	2020-09-25
12	Bruce	Ernst	bruceernst@guvi.com	HR	20000	2019-06-21
13	David	Austin	davidaustin@guvi.com	IT	40000	2020-07-10
14	Diana	Lorentz	dianalorentz@guvi.com	HR	10000	2021-08-03

#	Time	Action	Message
1	20:23:54	SELECT * FROM Worker WHERE FIRST_NAME not in ('Ramesh','Santhosh') LIMIT 0, 1000	47 row(s) returned

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



The screenshot shows a SQL query editor with a query window and a result grid. The query is:

```
1 SELECT * FROM Worker WHERE FIRST_NAME LIKE '____h';
```

The result grid displays the following data:

ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	JOINDATE
1	Yogesh	Vaishnav	yogeshvaishnav@guvi.com	Accounts	30000	2019-10-03
4	Ashish	Yadav	ashishyadav@guvi.com	HR	10000	2019-12-25
19	Ramesh	Loganathan	rameshlogathan@guvi.com	IT	30000	2021-03-10
28	Ramesh	Raghavan	rameshraghavan@guvi.com	IT	25000	2019-10-10
42	Sarath	Kamal	sarathkamal@guvi.com	IT	50000	2019-03-01
50	Somesh	Bandavath	someshbandavath@guvi.com	IT	25000	2019-05-20
NULL	NULL	NULL	NULL	NULL	NULL	NULL

The bottom section shows the output of the query, indicating that 6 rows were returned.

Worker 4 x Apply Revert Context H

Output

Action Output

#	Time	Action	Message
1	20:27:47	SELECT * FROM Worker WHERE FIRST_NAME LIKE '____h' LIMIT 0, 1000	6 row(s) returned

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).

The screenshot shows a SQL query editor with a query to validate email addresses. The query is as follows:

```
1 SELECT EMAIL AS ValidEmail
2 FROM WORKER WORK
3 WHERE EMAIL LIKE lower(CONCAT(WORK.FIRST_NAME,WORK.LAST_NAME,'@','guvi.com'));
```

The results are displayed in a table with the following data:

ValidEmail
yogeshvaishnav@guvi.com
vishalvishwakarma@guvi.com
ajityadav@guvi.com
ashishyadav@guvi.com
tanvithakur@guvi.com
samparker@guvi.com
ronwesley@guvi.com
sararogers@guvi.com
yojiseo@guvi.com
zaravargas@guvi.com
stevenking@guvi.com
bruceernst@guvi.com
davidAustin@guvi.com
dianalorentz@guvi.com

The bottom of the screenshot shows the output of the query, indicating that 50 row(s) returned.

#	Time	Action	Message
1	20:43:11	SELECT EMAIL AS ValidEmail FROM WORKER WORK WHERE EMAIL LIKE lower(CONCAT(WORK.FIRST_...	50 row(s) returned

Task-6

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

The screenshot displays a database query tool interface. At the top, a query editor shows the SQL statement: `SELECT * FROM WORKER WHERE JOINDATE LIKE '2021-03%';`. Below the editor, a 'Result Grid' tab is active, showing a table with 7 rows of worker data. The columns are ID, FIRST_NAME, LAST_NAME, EMAIL, DEPARTMENT, SALARY, and JOINDATE. The data includes workers like Shell Baida, Ramesh Loganathan, Adam Fripp, Kevin Mourgos, Curtis Davies, Allan McEwen, and Louise Doran, all of whom joined in March 2021. A toolbar on the right offers options like 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. At the bottom, an 'Output' section shows the query execution log, indicating that 7 rows were returned.

ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	JOINDATE
18	Shell	Baida	shellibaida@guvi.com	IT	20000	2021-03-05
19	Ramesh	Loganathan	rameshloganathan@guvi.com	IT	30000	2021-03-10
22	Adam	Fripp	adamfripp@guvi.com	Accounts	30000	2021-03-01
24	Kevin	Mourgos	kevinmourgos@guvi.com	Admin	40000	2021-03-20
32	Curtis	Davies	curtsdavies@guvi.com	IT	17000	2021-03-02
39	Allan	McEwen	allanmcewen@guvi.com	IT	40000	2021-03-11
41	Louise	Doran	louisedoran@guvi.com	Accounts	35000	2021-03-30
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Output

#	Time	Action	Message
1	20:48:20	SELECT * FROM WORKER WHERE JOINDATE LIKE '2021-03%' LIMIT 0, 1000	7 row(s) returned

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

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

```
1 SELECT FIRST_NAME, DEPARTMENT, COUNT(*)
2 FROM WORKER
3 GROUP BY FIRST_NAME, DEPARTMENT
4 HAVING COUNT(*) > 1;
```

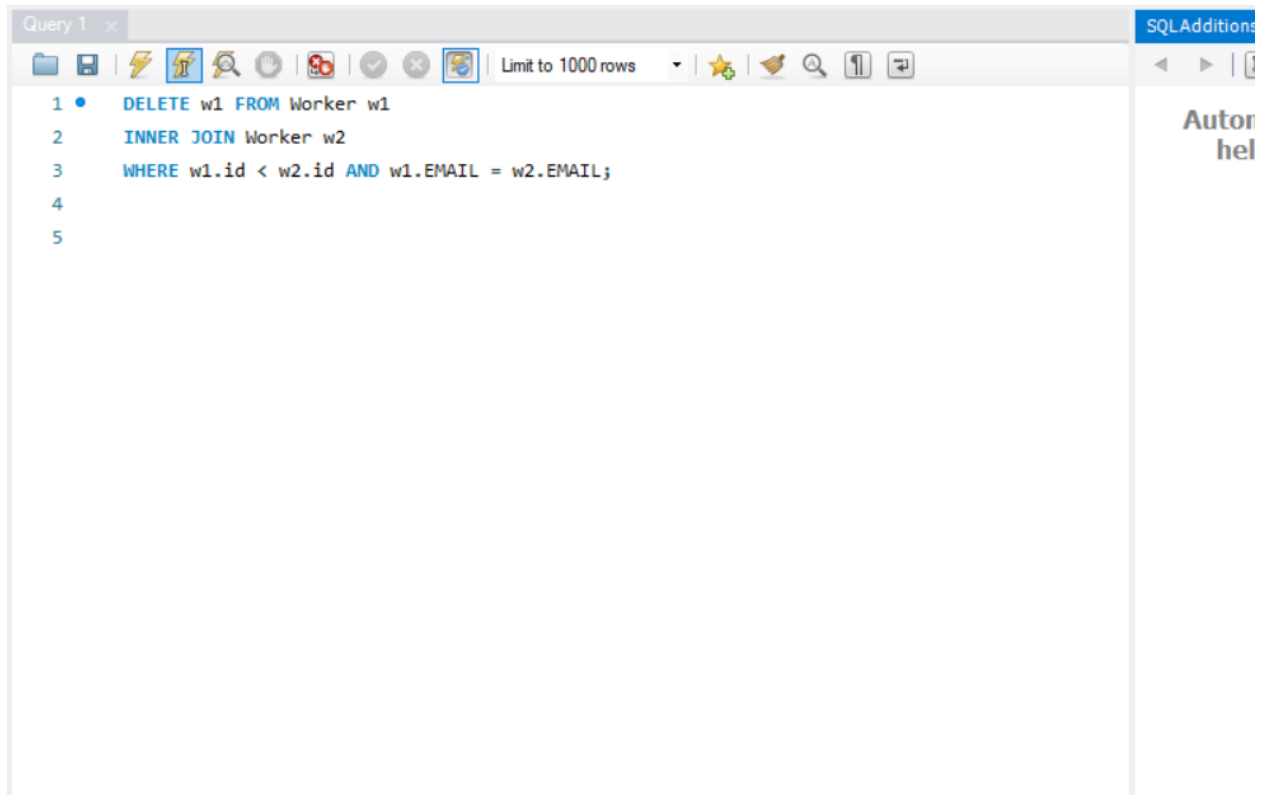
The bottom pane displays the 'Result Grid' with the following data:

FIRST_NAME	DEPARTMENT	COUNT(*)
David	IT	2
Ramesh	IT	2

Below the result grid, the 'Output' pane shows the execution log:

#	Time	Action	Message
1	20:57:01	SELECT FIRST_NAME, DEPARTMENT, COUNT(*) FROM WORKER GROUP BY FIRST_NAME, DEPARTME...	2 row(s) returned

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



Task-7

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

The screenshot shows a database management tool interface. At the top, a query editor displays the SQL query: `SELECT * FROM Worker WHERE MOD (ID, 2) <> 0;`. Below the query editor, a toolbar includes options like 'Limit to 1000 rows', 'Filter Rows', 'Edit', 'Export/Import', and 'Wrap Cell Content'. The main area displays a 'Result Grid' with a table of data. The table has columns: ID, FIRST_NAME, LAST_NAME, EMAIL, DEPARTMENT, SALARY, and JOINDATE. The data is filtered to show only odd-numbered rows (ID 1, 3, 5, etc.). On the right side, there is a sidebar with buttons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. At the bottom, an 'Output' section shows the execution details: 'Action Output' with a green checkmark, 'Time' 21:06:23, 'Action' `SELECT * FROM Worker WHERE MOD (ID, 2) <> 0 LIMIT 0, 1000`, and 'Message' '25 row(s) returned'.

ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	JOINDATE
1	Yogesh	Vaishnav	yogeshvaishnav@guvi.com	Accounts	30000	2019-10-03
3	Ajit	Yadav	ajityadav@guvi.com	Admin	15000	2019-12-12
5	Tanvi	Thakur	tanvithakur@guvi.com	IT	25000	2020-01-20
7	Ron	Wesley	ronwesley@guvi.com	Accounts	25000	2020-05-16
9	Yoji	Seo	yojiseo@guvi.com	Admin	20000	2020-03-10
11	Steven	King	stevenking@guvi.com	IT	25000	2020-09-25
13	David	Austin	davidaustin@guvi.com	IT	40000	2020-07-10
15	Nancy	Greenberg	nancygreenberg@guvi.com	IT	25000	2020-01-20
17	Luis	Popp	luispopp@guvi.com	Accounts	25000	2019-12-25
19	Ramesh	Loganathan	rameshloganathan@guvi.com	IT	30000	2021-03-10
21	Karen	Colmenares	karencolmenares@guvi.com	HR	22000	2019-10-03
23	Shanta	Vollman	shantavollman@guvi.com	Operations	30000	2021-07-10
25	Julia	Mayers	juliamayers@guvi.com	Operations	25000	2020-07-20
27	Steven	Markle	stevenmarkle@guvi.com	HR	30000	2020-08-31

Output

#	Time	Action	Message
1	21:06:23	SELECT * FROM Worker WHERE MOD (ID, 2) <> 0 LIMIT 0, 1000	25 row(s) returned

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

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

```
1 CREATE TABLE WorkerClone LIKE Worker;
```

The IDE's toolbar includes icons for file operations, execution, and a 'Limit to 1000 rows' dropdown. On the right side, there are buttons for 'SQLAddition' and 'Auto h'.

Below the query editor is the 'Output' pane, which displays the execution results in a table format:

#	Time	Action	Message
1	21:09:26	CREATE TABLE WorkerClone LIKE Worker	0 row(s) affected

Task-8

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

The screenshot shows a SQL IDE interface. The main editor window displays the following SQL query:

```
1 • SELECT * FROM Worker W
2 INNER JOIN WorkerClone WC ON
3 W.ID = WC.ID;
4
5
6
```

Below the query editor is a toolbar with icons for saving, running, and other functions. A dropdown menu shows "Limit to 1000 rows".

Below the toolbar is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The grid itself is empty, showing only the column headers:

ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	JOINDATE	ID	FIRST_NAME	LAST_NAI
----	------------	-----------	-------	------------	--------	----------	----	------------	----------

On the right side of the IDE, there is a vertical toolbar with icons for "Result Grid", "Form Editor", "Field Types", and "Query Stats".

At the bottom of the IDE, there is an "Output" section. It includes a "Read Only" status indicator and a "Context Help" link. Below this is an "Action Output" section with a table showing the results of actions:

#	Time	Action	Message
---	------	--------	---------

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

The screenshot shows a SQL query editor with the following query:

```
1 SELECT ID, FIRST_NAME
2 FROM WORKER
3 WHERE NOT EXISTS
4 (SELECT *
5  FROM WORKERCLONE
6   WHERE WORKER.ID = WORKERCLONE.ID);
7
8
```

The query is executed, and the results are displayed in a grid. The grid shows 14 rows of data:

ID	FIRST_NAME
1	Yogesh
2	Vishal
3	Ajit
4	Ashish
5	Tanvi
6	Sam
7	Ron
8	Sara
9	Yoji
10	Zara
11	Steven
12	Bruce
13	David
14	Diana

The bottom of the screenshot shows the "Output" section with the following message:

```
1 21:38:32 SELECT ID, FIRST_NAME FROM WORKER WHERE NOT EXISTS (SELECT * FROM WORKERCLONE ... 50 row(s) returned
```

Task-9

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

The screenshot shows a database management interface with a query editor and a result grid. The query editor contains the following SQL query:

```
1 • SELECT * FROM Worker ORDER BY SALARY DESC LIMIT 15;
```

The result grid displays the top 15 records from the Worker table, ordered by salary in descending order. The columns are ID, FIRST_NAME, LAST_NAME, EMAIL, DEPARTMENT, SALARY, and JOINDATE. The results are as follows:

ID	FIRST_NAME	LAST_NAME	EMAIL	DEPARTMENT	SALARY	JOINDATE
42	Sarath	Kamal	sarathkamal@guvi.com	IT	50000	2019-03-01
39	Allan	McEwen	allanmcewen@guvi.com	IT	40000	2021-03-11
20	Santhosh	Padmanabhan	santhoshpadmanabhan@guvi.com	Accounts	40000	2020-01-20
24	Kevin	Mourgos	kevinmourgos@guvi.com	Admin	40000	2021-03-20
26	James	Landry	jameslandry@guvi.com	IT	40000	2019-05-31
13	David	Austin	davidaustin@guvi.com	IT	40000	2020-07-10
43	Danielle	Green	daniellegreen@guvi.com	HR	40000	2019-07-01
38	Patrick	Sully	patrickully@guvi.com	HR	35000	2020-08-11
49	Jack	Livingston	jacklivingston@guvi.com	IT	35000	2019-05-20
41	Louise	Doran	louisedoran@guvi.com	Accounts	35000	2021-03-30
1	Yogesh	Vaishnav	yogeshvaishnav@guvi.com	Accounts	30000	2019-10-03
19	Ramesh	Loganathan	rameshloganathan@guvi.com	IT	30000	2021-03-10
23	Shanta	Vollman	shantavollman@guvi.com	Operations	30000	2021-07-10
10	Zara	Vargas	zaravargas@guvi.com	IT	30000	2020-08-20
27	Steven	Markle	stevenmarkle@guvi.com	HR	30000	2020-08-31
NULL	NULL	NULL	NULL	NULL	NULL	NULL

The output section shows the action output and the message:

```
1 21:41:23 SELECT * FROM Worker ORDER BY SALARY DESC LIMIT 15
```

Message: 15 row(s) returned

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

The screenshot shows a database management interface with a query editor and a results pane. The query editor contains the following SQL query:

```
1 • SELECT SALARY FROM Worker ORDER BY SALARY DESC LIMIT 9,1;
```

The results pane displays a single row of data:

SALARY
35000

The interface also includes a toolbar with various icons, a "Limit to 1000 rows" dropdown, and a "Result Grid" button. The bottom pane shows the "Output" section with a table of action logs:

#	Time	Action	Message
1	21:46:46	SELECT SALARY FROM Worker ORDER BY SALARY DESC LIMIT 9,1	1 row(s) returned

Task-10

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

The screenshot shows a database query editor with the following SQL query:

```
1 SELECT DISTINCT Salary
2 FROM Worker W1
3 WHERE 7 = (
4     SELECT COUNT( DISTINCT ( W2.Salary ) )
5     FROM Worker W2
6     WHERE W2.Salary >= W1.Salary
7 );
8
```

The query is executed, and the result grid shows the following data:

Salary
20000

The bottom of the screenshot shows the output log with the following message:

```
1 21:50:55 SELECT DISTINCT Salary FROM Worker W1 WHERE 7 = ( SELECT COUNT( DISTINCT ( W2.Salary ) ) FRO... 1 row(s) returned
```

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

The screenshot shows a SQL query editor window titled "Query 1". The query is as follows:

```
1 • Select DISTINCT W.ID, W.FIRST_NAME, W.SALARY
2 FROM Worker W, Worker W1
3 WHERE W.Salary = W1.Salary
4 AND W.ID != W1.ID;
5
6
7
```

Below the query editor is a "Result Grid" showing the results of the query. The grid has columns for ID, FIRST_NAME, and SALARY. The results are as follows:

ID	FIRST_NAME	SALARY
46	Harrison	30000
44	David	30000
37	Andrew	30000
27	Steven	30000
23	Shanta	30000
22	Adam	30000
19	Ramesh	30000
10	Zara	30000
35	Gerald	20000
29	Laura	20000
18	Shelli	20000
16	John	20000
12	Bruce	20000
9	Yoji	20000
8	Sara	20000

At the bottom of the window, there is an "Output" section showing the execution of the query. The output is as follows:

#	Time	Action	Message
1	21:54:32	Select DISTINCT W.ID, W.FIRST_NAME, W.SALARY FROM Worker W, Worker W1 WHERE W.Salary = W...	47 row(s) returned