

SQL ASSIGNMENTS

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

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 SQL IDE interface. At the top, a tab labeled 'Query 1' is active. The query editor contains the following SQL statement:

```
1 • SELECT FIRST_NAME AS WORKER_NAME FROM Worker;
```

Below the query editor, the 'Result Grid' is visible, showing a table with one column, 'WORKER_NAME', and 50 rows of employee names. The names listed are: Yogesh, Vishal, Ajit, Ashish, Tanvi, Sam, Ron, Sara, Yoji, Zara, Steven, Bruce, and David. The interface includes various toolbars for editing, executing, and viewing results. On the right side, there are buttons for 'Result Grid', 'Form Editor', and 'Field Types'. At the bottom, the 'Output' pane shows the execution log with the following entry:

#	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 of data: "Accounts", "IT", "Admin", "HR", and "Operations". To the right of the result grid is a vertical toolbar with icons for "Result Grid", "Form Editor", and "Field Types". Below the result 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

Limit to 1000 rows

1 • `SELECT DISTINCT DEPARTMENT FROM Worker;`

Result Grid

DEPARTMENT
Accounts
IT
Admin
HR
Operations

Worker 8 x

Read Only

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 database management interface. At the top, a query editor window titled 'Query 1' contains the following SQL query:

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

Below the query editor, a 'Result Grid' displays the results of the query. The grid has columns for 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	alysahutton@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 interface, an 'Output' section shows 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

Query 1 x

Limit to 1000 rows

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

Result Grid

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

Result 11 x

Read Only

Context Hel

Output

Action Output

#	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 SQL Developer interface. The top pane, titled 'Query 1', contains the SQL query: `1 Select INSTR(FIRST_NAME, BINARY'a') from Worker;`. The bottom pane displays the 'Result Grid' with the column header `INSTR(FIRST_NAME, BINARY'a')` and 50 rows of data. The values in the rows are: 0, 5, 1, 1, 2, 2, 0, 2, 0, 2, 0, 0. The right sidebar includes 'SQLAddition', 'Auto he', 'Result Grid', 'Form Editor', and 'Field Types'. The bottom status bar shows 'Result 12 x', 'Read Only', and 'Context Help'. The 'Output' pane at the very bottom shows a log entry: `1 00:11:32 Select INSTR(FIRST_NAME, BINARY'a') from Worker LIMIT 0, 1000` with the message '50 row(s) returned'.

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

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 with the following query:

```
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;
```

The results are displayed in a grid with the following columns: DEPARTMENT, FIRST_NAME, SALARY.

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

The bottom of the screenshot shows the 'Output' section with the following 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.

Query 1 x SQLAdditions x

Limit to 1000 rows

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

Automa help

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

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

Result 16 x Read Only Context Help

Output

Action Output

#	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

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.

Query 1 x SQLAdd

Limit to 1000 rows

```
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;
```

Result Grid

Salary
50000
40000
35000
30000
25000

worker 25 x Read Only Context

Output

Action Output

#	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'.

Query 1 x

SQLAdditions:

Limit to 1000 rows

1 `SELECT REPLACE(FIRST_NAME,'a','A') FROM WORKER;`

Autom help

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

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

Result 21 x Read Only Context Help

Output

Action Output

#	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;**

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.

Task-5

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

Query 1 x SQLAdditions

Limit to 1000 rows

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

Autom help

Result Grid

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

Worker 2 x Apply Revert Context Help

Output

Action Output

#	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.

Query 1 x

SQL Addit

Limit to 1000 rows

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

Result Grid

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	rameshloganathan@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

Form Editor

Field Types

Query Stats

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 database query tool interface. The top section displays a SQL query in a text editor:

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

Below the query editor, the "Result Grid" is visible, showing a list of email addresses under the column "ValidEmail". The results are:

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

At the bottom, the "Output" section shows the execution details:

#	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 shows a database management 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 JOINDATE LIKE '2021-03%';
```

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 7 rows of data, with the last row showing NULL values. A vertical toolbar on the right side of the result grid includes buttons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'.

At the bottom of the interface, there is an 'Output' section with a dropdown menu set to 'Action Output'. Below this, a table shows the execution log:

#	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 query editor with the following query:

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

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

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

The interface also shows a toolbar with various icons, a 'Limit to 1000 rows' dropdown, and a 'Result Grid' button. The bottom section shows the 'Output' pane with a message: 'SELECT FIRST_NAME, DEPARTMENT, COUNT(*) FROM WORKER GROUP BY FIRST_NAME, DEPARTMENT... 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 SQL IDE interface. The query editor at the top contains the following SQL query:

```
1 • SELECT * FROM Worker WHERE MOD (ID, 2) <> 0;
```

Below the query editor is a "Result Grid" showing the results of the query. The grid has 8 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
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

At the bottom of the IDE, the "Output" pane shows the execution log:

#	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 query editor window titled "Query 1". The query text is: `1 CREATE TABLE WorkerClone LIKE Worker;`. The editor has a toolbar with various icons, including a "Limit to 1000 rows" dropdown. On the right side, there is a "SQLAdditi" button and a partially visible "Auto h" label. Below the query editor is an "Output" section with a dropdown menu set to "Action Output". The output table has four columns: "#", "Time", "Action", and "Message". It contains one row with a green checkmark in the "#" column, the value "1" in the "Time" column, the text "21:09:26 CREATE TABLE WorkerClone LIKE Worker" in the "Action" column, and "0 row(s) affected" in the "Message" column.

#	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 query editor window titled "Query 1". The query is as follows:

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

The editor includes a toolbar with various icons and a "Limit to 1000 rows" dropdown. Below the query editor is a "Result Grid" section with a table header showing columns: ID, FIRST_NAME, LAST_NAME, EMAIL, DEPARTMENT, SALARY, JOINDATE, ID, FIRST_NAME, LAST_NAME. The grid is currently empty. To the right of the grid is a vertical toolbar with icons for "Result Grid", "Form Editor", "Field Types", and "Query Stats". At the bottom of the window, there is an "Output" section with a dropdown menu set to "Action Output" and a table with columns: #, Time, Action, and Message. The status bar at the bottom right indicates "Read Only" and "Context Help".

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

The screenshot shows a database management tool interface. At the top, a tab labeled "Query 1" is active. The SQL query editor contains the following code:

```
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
```

Below the query editor, the "Result Grid" is displayed, showing a table with two columns: "ID" and "FIRST_NAME". The table contains 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

At the bottom of the interface, the "Output" pane shows the "Action Output" for the query execution. It displays a single row with a green checkmark, indicating successful execution:

#	Time	Action	Message
1	21:38:32	SELECT ID, FIRST_NAME FROM WORKER WHERE NOT EXISTS (SELECT * FROM WORKERCLONE WHERE WORKER.ID = WORKERCLONE.ID);	50 row(s) returned

Task-9

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

Query 1 x

SQLAddition

Limit to 1000 rows

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

2

3

4

Result Grid

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	kevinmorgos@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	danielegreen@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

Worker 18 x

Apply Revert Context Help

Output

Action Output

#	Time	Action	Message
1	21:41:23	SELECT * FROM Worker ORDER BY SALARY DESC LIMIT 15	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 tool interface. At the top, a tab labeled "Query 1" is active. The SQL query editor contains the following query:

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

Below the query editor, the "Result Grid" is displayed, showing a single row of results:

SALARY
35000

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

At the bottom, the "Worker 19" tab is active, and the "Output" section shows the execution results:

#	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 management tool interface. The main window displays an SQL query in a text editor. The query is as follows:

```
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
```

Below the query editor, there is a 'Result Grid' section. It shows a table with one column 'Salary' and one row with the value '20000'.

At the bottom of the interface, there is an 'Output' section. It shows a table with columns '#', 'Time', 'Action', and 'Message'. The first row indicates that the query was executed successfully and returned 1 row(s).

#	Time	Action	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 database query tool interface. The top section displays the SQL query for 'Query 1':

```
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 the 'Result Grid' showing the results of the query. The grid has three columns: 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

On the right side of the interface, there is a vertical toolbar with icons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. Below the toolbar, there are buttons for 'Read Only' and 'Context Help'.

At the bottom of the interface is the 'Output' section, which shows the execution details of the query:

#	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