

## GUVI ASSIGNMENT

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

The screenshot shows the MySQL Workbench interface with the 'worker' table selected. The table has 50 rows of data, each representing an employee with columns: First\_name, Last\_name, Email, Department, Salary, DOJ, and ID. The data includes names like Alex, Arjun, Andrew, Anil, Brian, Bruce, Cam, Darren, David, Ferry, Ganesh, Gloria, Guntu, Hardik, Jal, Joe, Joey, Justin, Kishore, Lex, Levy, Mahesh, Mari, Mayank, Michael, Mitch, Pandy, Rajan, Rohit, Priyanka, Rachel, Rahul, Ram, Ramesh, Ravindra, Rekha, Ross, Sai, Shawn, Sudesh, Suresh, Surya, Thala, and Vrat. The 'DOJ' column shows various dates from 2001-03-21 to 2018-07-05.

## TASK-1

1. Write an SQL query to fetch “**FIRST\_NAME**” from the Worker table using the alias name as **<WORKER\_NAME>**.

ANS : select First\_name as worker\_name from worker;

The screenshot shows the MySQL Workbench interface with the result of the SQL query: "select First\_name as worker\_name from workers". The result grid displays the 'worker\_name' column with 50 entries corresponding to the first names listed in the previous screenshot.

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

ANS : SELECT DISTINCT Department from worker;

The screenshot shows the MySQL Workbench interface. In the 'Query Editor' tab, the query `SELECT DISTINCT Department from worker` is run, resulting in the following output:

Department
IT
Marketing
UX
R&D
UI
Analyst
Dev
HR

In the bottom left, the 'Information' pane displays the structure of the 'worker' table:

Columns:
First_name varchar(50)
Last_name varchar(30)
Email varchar(50)
Department varchar(20)
Salary int
DOJ date
ID int AI PK

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

ANS : (select \* from worker order by ID desc limit 5) order by ID asc;

The screenshot shows the MySQL Workbench interface. In the 'Query Editor' tab, the query `(select * from worker order by ID desc limit 5) order by ID asc` is run, resulting in the following output:

First_name	Last_name	Email	Department	Salary	DOJ	ID
Suresh	Jada	jada@gmail.com	BT	22222	2001-09-19	46
Surya	Aditya	aditya@gmail.com	Marketing	30000	2001-09-21	47
Thala	Mognath	mognath@gmail.com	HR	30000	2006-07-17	48
Wat	Kohli	kohli@gmail.com	Dev	22222	2007-09-12	49
Elon	Musk	musk@gmail.com	UK	20908	1986-03-04	50

In the bottom left, the 'Information' pane displays the structure of the 'worker' table:

Columns:
First_name varchar(50)
Last_name varchar(30)
Email varchar(50)
Department varchar(20)
Salary int
DOJ date
ID int AI PK

## **TASK-2**

1. Write an SQL query to print the first three characters of FIRST\_NAME from Worker.

ANS : SELECT SUBSTRING(First\_name,1,3) FROM worker;

The screenshot shows the MySQL Workbench interface. In the central Query Editor window, the following SQL query is run:

```
1  SELECT SUBSTRING(First_name,1,3) FROM worker;
```

The Result Grid displays the output of the query:

SUBSTRING(First_name,1,3)
Ale
All
And
Ari
Br
Bru
Cam
Dar
Dav
Ela
Fer
Gan
Glo
Gun
Har
Jai
Joe
Jus
Kru
Lex
Lex
Mah
Man
May
Mc

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

ANS : Select INSTR(FIRST\_NAME, BINARY 'a') from Worker

The screenshot shows the MySQL Workbench interface. In the central Query Editor window, the following SQL query is run:

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

The Result Grid displays the output of the query:

INSTR(FIRST_NAME, BINARY 'a')
1
1
1
4
0
2
2
0
2
6
0
2
2
0
0
5
0
0
2
2
2
2

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

ANS :SELECT Department, MAX(salary) FROM worker GROUP BY Department;

The screenshot shows the MySQL Workbench interface. In the top-left, the 'Navigator' pane shows the 'Schemas' tree with 'ganesha' selected, and the 'Tables' node under it, which contains 'worker'. The 'Table: worker' details pane shows columns: First\_name (varchar(50)), Last\_name (varchar(50)), Email (varchar(50)), Department (varchar(20)), Salary (int), and DOJ (date). The 'ID' column is marked as the Primary Key (PK). In the top-right, the 'Query Editor' pane displays the query: 'SELECT Department, MAX(salary) FROM worker GROUP BY Department;'. The 'Result Grid' pane shows the output:

Department	MAX(salary)
IT	652323
Marketing	332522
UX	23000
BI	562643
UI	30000
Analyst	474558
Dev	24521
HR	34969

## TASK-3

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

ANS : Select RTRIM(First\_name) from worker;

The screenshot shows the MySQL Workbench interface. In the top-left, the 'Navigator' pane shows the 'Schemas' tree with 'ganesha' selected, and the 'Tables' node under it, which contains 'worker'. The 'Table: worker' details pane shows columns: First\_name (varchar(50)), Last\_name (varchar(50)), Email (varchar(50)), Department (varchar(20)), Salary (int), and DOJ (date). The 'ID' column is marked as the Primary Key (PK). In the top-right, the 'Query Editor' pane displays the query: 'Select RTRIM(First\_name) from worker;'. The 'Result Grid' pane shows the output:

RTRIM(First_name)
Ale
Allu
Andrew
Anil
Brian
Bruce
Cam
Daren
David
Elon
Ferry
Ganesh
Gloria
Guntu
Hardik
Jai
Joe
Joey
Justin
Krunal
Lex
Lexy
Mahesh

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

ANS : SELECT DISTINCT (Department), length(Department) FROM worker;

The screenshot shows the MySQL Workbench interface. In the top-left pane, the 'Schemas' tree shows the 'ganesha' database with a 'Tables' node expanded, containing 'worker'. The 'Tables' node has sub-options like 'Columns', 'Indexes', 'Foreign Keys', 'Triggers', 'Views', 'Stored Procedures', and 'Functions'. The 'sys' schema is also listed. In the top-right pane, the 'Query 1' editor contains the SQL query:

```
1 • SELECT DISTINCT (department), length(Department) FROM worker;
```

The bottom pane displays the results in a 'Result Grid' table:

department	length(Department)
IT	2
Marketing	9
UX	2
BI	2
UI	2
Analyst	7
Dev	3
HR	2

On the right side of the interface, there's a vertical toolbar with icons for 'Result Grid', 'Form Editor', and 'Field Types'. A status bar at the bottom right shows the date and time: 15-11-2022 15:02.

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

ANS : SELECT Salary FROM worker ORDER BY Salary DESC Limit 0,1

The screenshot shows the MySQL Workbench interface. In the top-left pane, the 'Schemas' tree shows the 'ganesha' database with a 'Tables' node expanded, containing 'worker'. The 'Tables' node has sub-options like 'Columns', 'Indexes', 'Foreign Keys', 'Triggers', 'Views', 'Stored Procedures', and 'Functions'. The 'sys' schema is also listed. In the top-right pane, the 'Query 1' editor contains the SQL query:

```
1 • SELECT Salary FROM worker ORDER BY Salary DESC LIMIT 0,1;
```

The bottom pane displays the results in a 'Result Grid' table:

Salary
652323

On the right side of the interface, there's a vertical toolbar with icons for 'Result Grid', 'Form Editor', and 'Field Types'. A status bar at the bottom right shows the date and time: 15-11-2022 14:26.

## TASK-4

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

ANS : SELECT REPLACE (First\_name, 'a','A') FROM worker;

The screenshot shows the MySQL Workbench interface. In the central pane, the results of the query `SELECT REPLACE(First_name, 'a', 'A') FROM worker;` are displayed in a grid. The results show the first names from the 'worker' table where 'a' has been replaced by 'A'. The names listed are Alex, Aliu, Andrew, Anil, Bruce, Cam, Darren, David, Elon, Ferry, Ganesh, Gloria, Guntu, Hardik, Jal, Joe, Joey, Justin, Krunal, Lex, Lexy, Mahesh, Mari, Mayank.

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

ANS : SELECT \* FROM worker ORDER BY First\_name ASC, Department DESC;

The screenshot shows the MySQL Workbench interface. In the central pane, the results of the query `SELECT * FROM worker ORDER BY First_name ASC, department DESC;` are displayed in a grid. The results show all columns for each worker, ordered by First\_name in ascending order and department in descending order. The data includes columns like First\_name, Last\_name, Email, Department, Salary, DOJ, and ID.

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

ANS : SELECT First\_name,salary from worker where salary = (SELECT max(Salary) from worker);

The screenshot shows the MySQL Workbench interface. In the 'Query 1' editor, the following SQL query is run:

```
1 • SELECT First_name, Salary from worker WHERE Salary=(SELECT max(Salary) from worker)
```

The results are displayed in the 'Result Grid' tab:

First_name	Salary
Ross	65232

The 'Information' pane on the left shows the 'worker' table definition:

**Table: worker**

**Columns:**

- First\_name varchar(50)
- Last\_name varchar(30)
- Email varchar(50)
- Department varchar(20)
- Salary int
- DOJ date
- ID int AI PK

The 'Object Info' pane shows the current session and connection details.

## TASK-5

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

ANS : SELECT \* FROM worker WHERE First\_name NOT IN (“RAMESH”,“SANTOSH”);

The screenshot shows the MySQL Workbench interface. In the 'Query 1' editor, the following SQL query is run:

```
1 • SELECT * FROM worker  
2 WHERE first_name NOT IN('Ramesh', 'Santhosh')  
3  
4
```

The results are displayed in the 'Result Grid' tab:

First_name	Last_name	Email	Department	Salary	DOJ	ID
Mahesh	Frodo	frof@gmail.com	HR	30030	2006-07-15	25
Mitch	Pritch	pritch@gmail.com	HR	34039	2012-03-13	26
Monica	Geller	geller@gmail.com	HR	34969	2013-03-11	27
Naren	Jalle	jala@gmail.com	IT	12000	2024-09-07	28
Nicholas	Pooran	poorna@gmail.com	Marketing	30402	2001-03-21	29
Parvesh	Raina	raina@gmail.com	IT	12001	2024-09-01	30
Paul	Valthay	valberrnan@gmail.com	Marketing	33202	2003-03-12	31
Perry	Mathew	perry@gmail.com	Dev	23541	1996-03-24	32
Phoebe	Buffay	buffay@gmail.com	BI	56264	2005-05-14	33
Priyvili	Shaw	shaw@gmail.com	Dev	23121	1998-08-24	34
Rachel	Green	green@gmail.com	Marketing	33252	2003-09-10	35
Rahul	David	dravid@gmail.com	BI	25643	2004-05-04	36
Ram	Charan	charan@gmail.com	Dev	23421	1996-08-24	37
Ravindra	Jadeja	jadeja@gmail.com	HR	30929	2003-07-13	38
Rishabh	Puri	puri@gmail.com	Analyst	45648	1997-09-22	40
Ross	Geller	ross@gmail.com	IT	65232	2001-01-01	41
Sai	Cherry	cherry@gmail.com	Dev	23521	1996-07-24	42
Shawn	Mendoza	mendoza@gmail.com	BI	22843	2002-05-01	43
Sudeesh	Prabhu	prabhu@gmail.com	Analyst	45645	1997-09-23	44
Sudha	Poornima	poornima@gmail.com	BI	22222	2001-09-19	45
Surya	Yadav	yadav@gmail.com	Marketing	300083	2001-09-21	47
Thalia	Mognath	mognath@gmail.com	HR	30989	2006-07-17	48
Virat	Kohli	kohli@gmail.com	Dev	22222	2007-09-12	49
Elo	Musk	musk@gmail.com	UX	20908	1986-03-04	50

The 'Information' pane on the left shows the 'worker' table definition:

**Table: worker**

**Columns:**

- First\_name varchar(50)
- Last\_name varchar(30)
- Email varchar(50)
- Department varchar(20)
- Salary int
- DOJ date
- ID int AI PK

The 'Object Info' pane shows the current session and connection details.

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

ANS : SELECT \* FROM worker WHERE First\_name like '\_\_\_\_\_h';

```

MySQL Workbench - Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator: Schemas
SCHEMAS
  Filter objects
  general
    Tables
      worker
        Columns
        Indexes
        Foreign Keys
        Triggers
      Views
      Stored Procedures
      Functions
  sys
Query 1 SQL File 3*
1 • SELECT * FROM worker
2 WHERE First_name like '_____h'
3
4
Result Grid | Filter Rows | Edit: | Export/Import: | Wrap Cell Contents: 
First_name Last_name Email Department Salary DOJ ID
Ganesh cheerla gan123@gmail.com ANALYST 25000 2020-03-12 11
Mahesh Babu babu@gmail.com HR 34929 2002-07-13 22
Naresh Jala Jla@gmail.com IT 12000 0204-09-07 28
Ramesh Malla malla121@gmail.com Marketing 30002 2001-03-22 38
sudeesh prabhu prabhu@gmail.com Analyst 45678 1997-09-23 44
Suresh Yedla yedla@gmail.com R 22222 2002-09-19 46
Rakesh Kishore rakesh_kishore@gmail.com Sales 33333 2003-09-20 47
Rakesh Kishore rakesh_kishore@gmail.com Sales 33333 2003-09-20 48
Rakesh Kishore rakesh_kishore@gmail.com Sales 33333 2003-09-20 49

```

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](mailto:kamalraja@guvi.com)).

ANS : Select \* from worker where (First\_name ="Allu" AND Last\_name = "Arjun") AND  
(Email REGEXP '^[A-Z0-9\_-]+[A-Z0-9\_-]+@[A-Z0-9.-]+\.[A-Z]{2,4}\$')

```

MySQL Workbench - Local instance MySQL80
File Edit View Query Database Server Tools Scripting Help
Navigator: Schemas
SCHEMAS
  Filter objects
  general
    Tables
      employee_done
      worker
      Views
      Stored Procedures
      Functions
  sys
Query 1 SQL File 3* SQL File 4*
1 • Select * from worker where (First_name = "Allu" AND Last_name = "Arjun") AND
2 (Email REGEXP '^[A-Z0-9_-]+[A-Z0-9_-]+@[A-Z0-9.-]+\.[A-Z]{2,4}$')
3
4
5
Result Grid | Filter Rows | Edit: | Export/Import: | Wrap Cell Contents: 
First_name Last_name Email Department Salary DOJ ID
Allu Arjun AlluArjun@govi.com IT 12723 2001-09-29 2

```

## TASK-6

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

ANS : SELECT \* FROM worker WHERE (DOJ) = 2021 and month(DOJ) =3;

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema 'ganesha' with a table 'worker' selected.
- Query Editor (Query 1):** Contains the SQL query:

```
1 • SELECT * FROM worker
2 WHERE year(DOJ) = 2021 and month(DOJ) = 3;
3
```
- Result Grid:** Displays the results of the query, showing three rows of data from the 'worker' table. The columns are First\_name, Last\_name, Email, Department, Salary, DOJ, and ID.
- Information:** Shows the table structure for 'worker' with columns: First\_name (varchar(50)), Last\_name (varchar(30)), Email (varchar(50)), Department (varchar(20)), Salary (int), DOJ (date), and ID (int AI PK).
- Toolbar:** Includes standard MySQL Workbench icons for file operations, database navigation, and query execution.
- System Tray:** Shows the date and time as 15-11-2022 15:37.

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

ANS : SELECT DEPARTMENT,Last\_name,COUNT(\*) FROM worker GROUP BY Department,Last\_name HAVING count(\*) > 1;

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema 'ganesha' with a table 'worker' selected.
- Query Editor (Query 1):** Contains the SQL query:

```
1 • SELECT DEPARTMENT,Last_name,COUNT(*) FROM worker GROUP BY DEPARTMENT,Last_name
2 HAVING COUNT(*) > 1;
```
- Result Grid:** Displays the results of the query, showing one row where IT department has 2 entries for the last name 'de'.
- Information:** Shows the table structure for 'worker' with columns: First\_name (varchar(50)), Last\_name (varchar(30)), Email (varchar(50)), Department (varchar(20)), Salary (int), DOJ (date), and ID (int AI PK).
- Toolbar:** Includes standard MySQL Workbench icons for file operations, database navigation, and query execution.
- System Tray:** Shows the date and time as 15-11-2022 16:00:02.

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

ANS : Here you can see the query is executed below in action output.

The screenshot shows the MySQL Workbench interface. In the Navigator pane, under the 'ganes' schema, the 'Tables' section contains 'employee\_clone', 'worker', and 'sys'. The 'worker' table is selected. In the Query Editor (Query 1), the following SQL code is written:

```
1
2 • SELECT
3     email, COUNT(email)
4   FROM
5     worker
6  GROUP BY
7     email
8  HAVING
9     COUNT(email) > 1;
10
11 • DELETE t1 FROM worker t1
12   INNER JOIN worker t2
13   WHERE
14     t1.ID < t2.ID AND
15     t1.Email = t2.Email;
16
17
```

The Output pane shows the execution results:

#	Time	Action	Message	Duration / Fetch
1	21:00:10	SELECT email, COUNT(email) FROM worker GROUP BY email HAVING COUNT(email) > 1 LIMIT ...	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the S...	0.000 sec
2	21:00:26	SELECT * from worker LIMIT 0, 1000	Error Code: 1046. No database selected Select the default DB to be used by double-clicking its name in the S...	0.000 sec
3	21:00:33	use ganes	0 row(s) affected	0.000 sec
4	21:00:46	SELECT email, COUNT(email) FROM worker GROUP BY email HAVING COUNT(email) > 1 LIMIT ...	3 row(s) returned	0.000 sec / 0.000 sec
5	21:00:51	DELETEt1 FROM workert1 INNER JOIN worker t2 WHERE t1.ID < t2.ID AND t1.Email = t2.Email	4 row(s) affected	0.015 sec
6	21:01:12	SELECT * from worker LIMIT 0, 1000	48 row(s) returned	0.000 sec / 0.000 sec
7	21:01:37	SELECT count(*) from worker LIMIT 0, 1000	1 row(s) returned	0.016 sec / 0.000 sec
8	21:02:14	SELECT email, COUNT(email) FROM worker GROUP BY email HAVING COUNT(email) > 1 LIMIT ...	0 row(s) returned	0.000 sec / 0.000 sec
9	21:02:22	DELETEt1 FROM workert1 INNER JOIN worker t2 WHERE t1.ID < t2.ID AND t1.Email = t2.Email	0 row(s) affected	0.000 sec

## TASK-7

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

ANS : SELECT \* FROM worker WHERE MOD (ID,2) <>0;

The screenshot shows the MySQL Workbench interface. In the Navigator pane, under the 'ganes' schema, the 'Tables' section contains 'worker', 'Views', 'Stored Procedures', and 'Functions'. The 'worker' table is selected. In the Query Editor (Query 1), the following SQL code is written:

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

The Output pane shows the results of the query:

First_name	Last_name	Email	Department	Salary	DOJ	ID
Andrew	Huberman	huberman@gmail.com	Marketing	30202	2001-03-12	3
Brian	Chantl	chantl@gmail.com	IT	12323	2001-01-29	5
Cam	Tucker	tucker@gmail.com	IT	17323	2001-01-29	7
David	Austin	austin23@gmail.com	IT	30000	2021-03-20	9
Ganesh	cheerla	gan123@gmail.com	ANALYST	25000	2020-03-12	11
Guntu	Pavan	guntu@gmail.com	HR	34939	2002-03-13	13
Jai	Kumar	jai@gmail.com	Analyst	47658	1993-01-22	15
Joey	Trbiani	trbiani@gmail.com	Analyst	40458	1993-01-15	17
Krystal	Pandya	pandya@gmail.com	HR	30000	2003-01-19	19
Levy	Friedman	levy@gmail.com	IT	12323	2004-09-29	21
Mari	Kanta	kanta@gmail.com	Dev	23121	1998-08-21	23
Michael	Fran	fran@gmail.com	HR	30939	2006-07-16	25
Monica	Geller	geller@gmail.com	HR	34969	2013-03-11	27
Nicholas	Pooran	pooran@gmail.com	Marketing	20402	2001-03-21	29
Paul	Valthay	valthay@gmail.com	Marketing	33202	2003-03-12	31
Phoebe	Buffay	buffay@gmail.com	IT	562643	2005-05-14	33
Rachel	Green	green@gmail.com	Marketing	332522	2003-09-10	35
Ram	Charan	charan@gmail.com	Dev	23421	1996-08-24	37
Ravindra	Jadeja	jadeja@gmail.com	HR	30929	2003-07-13	39
Ross	Geller	ross@gmail.com	IT	652323	2001-01-06	41
Shawn	Mendoza	mendoza@gmail.com	IT	22643	2002-05-04	43
Santhosh	Babu	babu@gmail.com	Analyst	47648	1993-09-22	45
Surya	Yadav	yadav@gmail.com	Marketing	300083	2001-09-21	47
Virat	Kohli	kohli@gmail.com	Dev	22222	2007-09-12	49

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

ANS : CREATE TABLE employee\_clone LIKE worker  
INSERT INTO employee\_clone SELECT \* FROM worker;

The screenshot shows the MySQL Workbench interface. In the Navigator pane, under the 'ganesh' schema, the 'Tables' section contains the 'worker' table. The 'Query 1' editor window displays the following SQL code:

```
1 • CREATE TABLE employee_clone
2   LIKE worker;
3 • INSERT INTO employee_clone
4   SELECT *
5   FROM worker;
```

The 'Output' pane shows the execution of these statements. The first two statements succeed, while the third fails with an error message:

Error Code: 1146. Table 'ganesh.employee\_clone' doesn't exist

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Action	Time	Message	Duration / Fetch
CREATE TABLE employee_clone	177 16:31:27	0.000 sec	0.000 sec
LIKE worker;	178 16:32:04	Error Code: 1050. Table 'employee_clone' already exists	0.016 sec / 0.000 sec
INSERT INTO employee_clone	179 16:33:00	50 rows returned	0.031 sec
SELECT *	180 16:33:21	0 row(s) affected	0.047 sec
FROM worker;	181 16:33:25	0 row(s) affected	0.047 sec
CREATE TABLE employee_clone	182 16:33:25	50 rows affected Records: 50 Duplicates: 0 Warnings: 0	0.015 sec
LIKE worker;	183 16:34:27	50 rows returned	0.000 sec / 0.000 sec
SELECT * FROM employee_clone	184 16:35:03	0 row(s) affected	0.031 sec
DROP TABLE ganesh.employee_clone	185 16:35:03	0 row(s) affected	0.047 sec
CREATE TABLE employee_clone LIKE worker	186 16:35:03	50 rows affected Records: 50 Duplicates: 0 Warnings: 0	0.016 sec

The screenshot shows the MySQL Workbench interface. In the Navigator pane, under the 'ganesh' schema, the 'Tables' section contains the 'employee\_clone' table. The 'Query 1' editor window displays the following SQL code:

```
1 • SELECT * FROM employee_clone;
```

The 'Output' pane shows the results of the SELECT query, displaying a grid of employee data. The columns are: First\_name, Last\_name, Email, Department, Salary, DOJ, and ID. The data includes entries for various employees like Alex de, Allu Arjun, Andrew Huberman, Anil Kumble, Brian Chant, Bruce Ernst, Cam Tucker, Darren Sammy, David Austin, Ferry Muscan, Ganesh cheerla, Gloria Delgado, Guntu Pavan, Hardik Pandya, Jal Kumar, Joe Rogan, Joey Roman, Justin Bieber, Kunal Pandya, Lexi de, Lexy Friedman, Mahesh Babu, Mari Kanta, Mayank Agarwal, Michael Fran, and many others.

First_name	Last_name	Email	Department	Salary	DOJ	ID
Alex	de	alex123@gmail.com	IT	30000	2021-03-21	1
Allu	Arjun	Arjun@gmail.com	IT	12723	2001-09-29	2
Andrew	Huberman	huberman@gmail.com	Marketing	30202	2001-03-12	3
Anil	Kumble	Kumble@gmail.com	Marketing	332422	2003-04-12	4
Brian	Chant	chant@gmail.com	IT	12323	2001-01-29	5
Bruce	Ernst	Bruce121@gmail.com	UX	23000	2018-07-05	6
Cam	Tucker	tucker@gmail.com	IT	17323	2001-01-29	7
Darren	Sammy	sammy@gmail.com	BI	223222	2002-09-23	8
David	Austin	austin2@gmail.com	UI	30000	2021-03-20	9
Ferry	Muscan	Muscan@gmail.com	Analyst	474558	1992-01-22	10
Ganesh	cheerla	gan123@gmail.com	ANALYST	25000	2020-03-12	11
Gloria	Delgado	delgado@gmail.com	Dev	24521	1986-07-24	12
Guntu	Pavan	guntu@gmail.com	HR	34939	2002-03-13	13
Hardik	Pandya	hpandya@gmail.com	IT	12023	2020-09-19	14
Jal	Kumar	jal@gmail.com	Analyst	47658	1993-01-22	15
Joe	Rogan	rogan@gmail.com	BI	22322	2002-09-23	16
Joey	Roman	joeyroman@gmail.com	Analyst	46559	1993-01-15	17
Justin	Bieber	ieber@gmail.com	Marketing	332222	2003-04-12	18
Kunal	Pandya	pandya@gmail.com	HR	30989	2003-07-19	19
Lexi	de	lexi123@gmail.com	IT	30000	2021-03-21	20
Lexy	Friedman	freddy@gmail.com	IT	12223	2004-09-29	21
Mahesh	Babu	babu@gmail.com	HR	34929	2002-07-13	22
Mari	Kanta	kanta@gmail.com	Dev	23121	1998-08-21	23
Mayank	Agarwal	agarwal@gmail.com	BI	22622	2002-09-04	24
Michael	Fran	fran@gmail.com	HR	30939	2006-07-16	25

## TASK-8

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

ANS :  
SELECT ID,First\_name,Last\_name,Email,Department,salary,DOJ FROM worker  
WHERE EXISTS(SELECT \* FROM employee\_clone WHERE worker.First\_name = employee\_clone.First\_name);

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema 'ganesha' with tables 'employee\_clone' and 'worker' selected.
- Query Editor (Query 1):** Contains the SQL query:

```
1 • SELECT ID,First_name,Last_name,Email,Department,salary,DOJ
  2   FROM worker
  3   WHERE EXISTS
  4     (SELECT *
  5      FROM employee_clone
  6     WHERE worker.First_name = employee_clone.First_name);
```
- Result Grid:** Displays the results of the query, showing 32 rows of data from the 'worker' table.
- Toolbar:** Includes standard MySQL Workbench icons for file operations, database navigation, and query execution.
- Status Bar:** Shows the date and time as 15-11-2022 17:40.

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

ANS :  
SELECT ID,First\_name,Last\_name,Email,Department,salary,DOJ FROM worker

WHERE NOT EXISTS(SELECT \* FROM employee\_clone WHERE worker.First\_name = employee\_clone.First\_name);

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema 'ganesha' with tables 'employee\_clone' and 'worker' selected.
- Query Editor (Query 1):** Contains the SQL query:

```
1 • SELECT First_name, ID, Email, Department, Salary, DOJ
  2   FROM worker
  3   WHERE NOT EXISTS
  4     (SELECT *
  5       FROM employee_clone
  6      WHERE worker.First_name = employee_clone.First_name);
```
- Result Grid:** Displays the results of the query, showing 1 row of data for Mitch.
- Action Output:** Shows the history of actions taken in the session, including the execution of the previous query and various error messages related to MySQL syntax errors.
- Toolbar:** Includes standard MySQL Workbench icons for file operations, database navigation, and query execution.
- Status Bar:** Shows the date and time as 15-11-2022 17:56.

## TASK-9

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

ANS : SELECT \* FROM worker ORDER BY Salary DESC LIMIT 10;

The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `ganesha` with the table `employee\_clone` selected. The table has columns: First\_name, Last\_name, Email, Department, Salary, DOJ, and ID.
- Query Editor:** Displays the SQL query: `SELECT \* FROM worker ORDER BY Salary DESC LIMIT 10;`
- Result Grid:** Shows the top 10 records from the `worker` table, ordered by salary in descending order. The records are as follows:

First_name	Last_name	Email	Department	Salary	DOJ	ID
Ross	Geller	ross@gmail.com	IT	652323	1993-01-06	41
Phoebe	Buffay	phoebe@gmail.com	IT	562323	2005-05-14	33
Felicity	Green	felicity@gmail.com	Analyst	474558	1992-01-22	10
Rachel	Kirkland	rachel@gmail.com	Marketing	323222	2003-09-10	35
Anil	Kumble	kumble@gmail.com	Marketing	324222	2003-04-12	4
Justin	Beiber	beiber@gmail.com	Marketing	322222	2003-04-12	18
Surya	Yadav	yadav@gmail.com	Marketing	300088	2001-09-21	47
Daren	Sammy	sammy@gmail.com	IT	223222	2002-09-23	8
Jai	Kumar	jai@gmail.com	Analyst	47558	1993-01-22	15
Santhosh	Babu	babu@gmail.com	Analyst	47648	1993-09-22	45

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

SELECT salary from worker ORDER BY salary DESC LIMIT 9, 1;

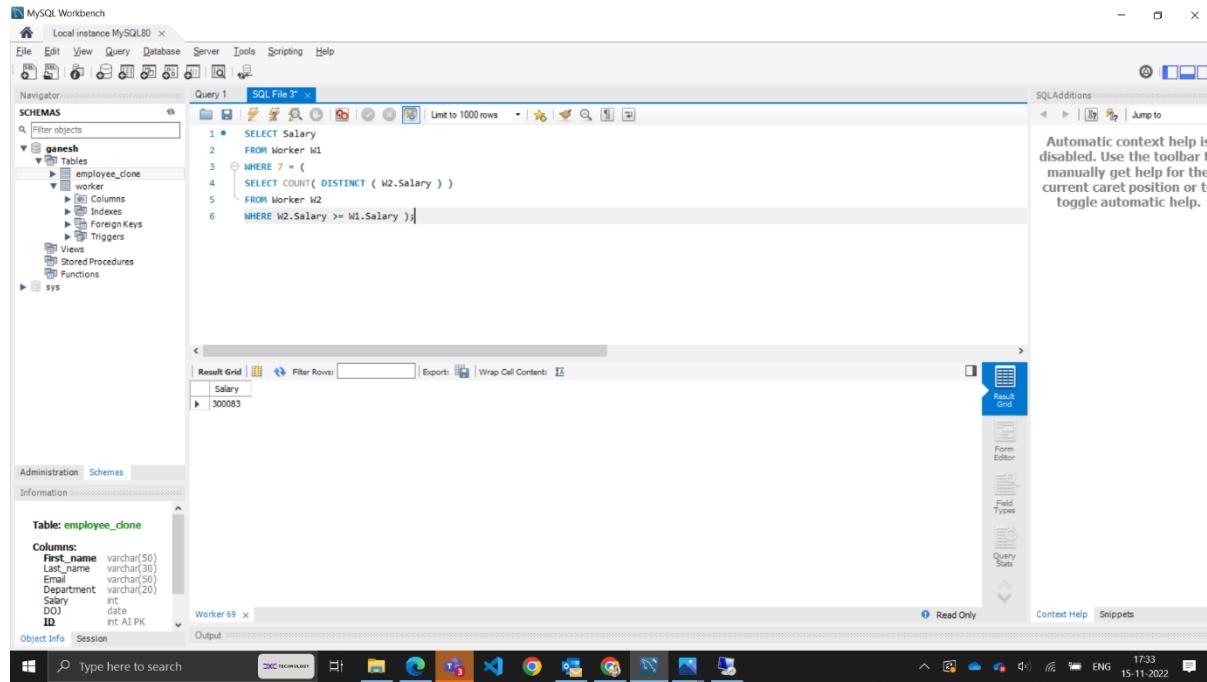
The screenshot shows the MySQL Workbench interface with the following details:

- Navigator:** Shows the schema `ganesha` with the table `employee\_clone` selected. The table has columns: First\_name, Last\_name, Email, Department, Salary, DOJ, and ID.
- Query Editor:** Displays the SQL query: `SELECT Salary FROM Worker ORDER BY Salary DESC LIMIT 9, 1;`
- Result Grid:** Shows the 10th highest salary from the `worker` table, ordered by salary in descending order. The result is: Salary = 47648.

## TASK-10

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

ANS : SELECT Salary FROM Worker W1 WHERE 7 = (SELECT COUNT( DISTINCT ( W2.Salary ) )FROM Worker W2 WHERE W2.Salary >= W1.Salary );

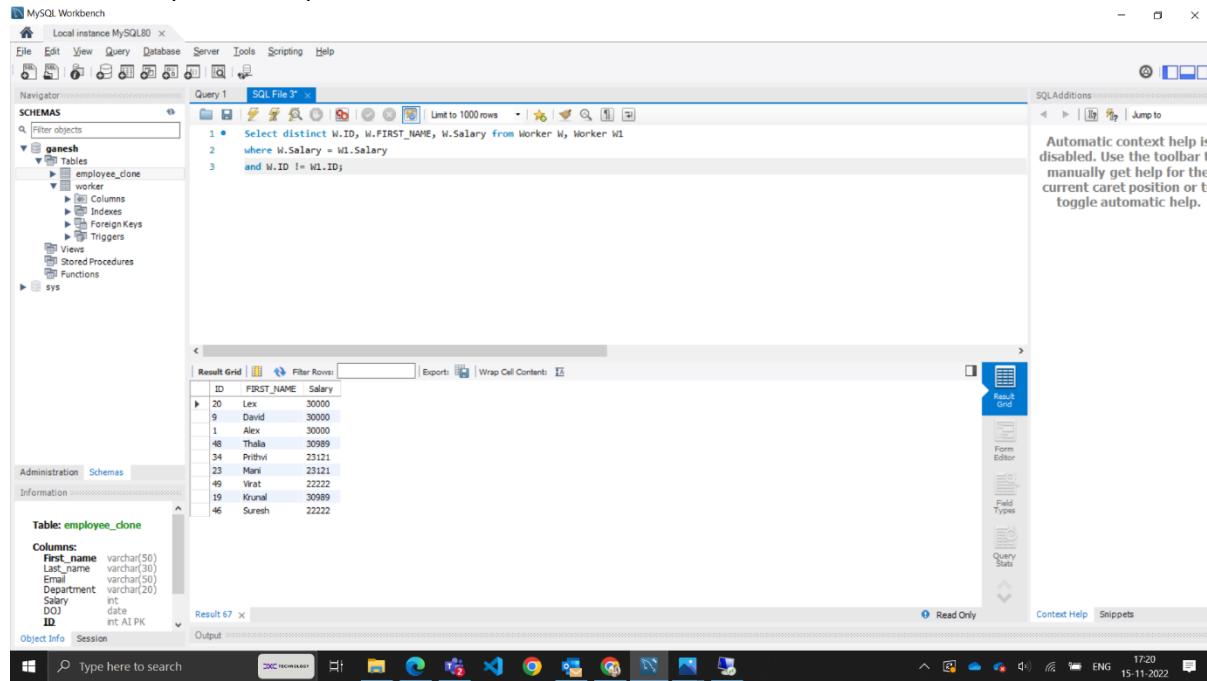


```
1 • SELECT Salary
2   FROM Worker W1
3   WHERE 7 = (
4     SELECT COUNT( DISTINCT ( W2.Salary ) )
5       FROM Worker W2
6      WHERE W2.Salary >= W1.Salary );
```

The screenshot shows the MySQL Workbench interface. The Query Editor window contains the provided SQL query. Below it, the Results Grid shows a single row with the value '300083'. The Navigator pane on the left shows the schema 'ganesha' with tables 'employee\_clone' and 'Worker'. The Information pane displays details for the 'employee\_clone' table, including columns like First\_name, Last\_name, Department, Salary, DOJ, and ID.

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

ANS : Select distinct W.ID, W.First\_name, W.Salary from Worker W, Worker W1 where W.Salary = W1.Salary and W.ID != W1.ID



```
1 • Select distinct W.ID, W.FIRST_NAME, W.Salary from Worker W, Worker W1
2   where W.Salary = W1.Salary
3   and W.ID != W1.ID;
```

The screenshot shows the MySQL Workbench interface. The Query Editor window contains the provided SQL query. Below it, the Results Grid displays a list of employees with the same salary but different IDs. The columns shown are ID, FIRST\_NAME, and Salary. The results include multiple rows for each unique salary value, such as Lex, David, Alex, Thala, Praveen, Mani, Vrat, Kunal, and Suresh, all having a salary of 30000.