

15

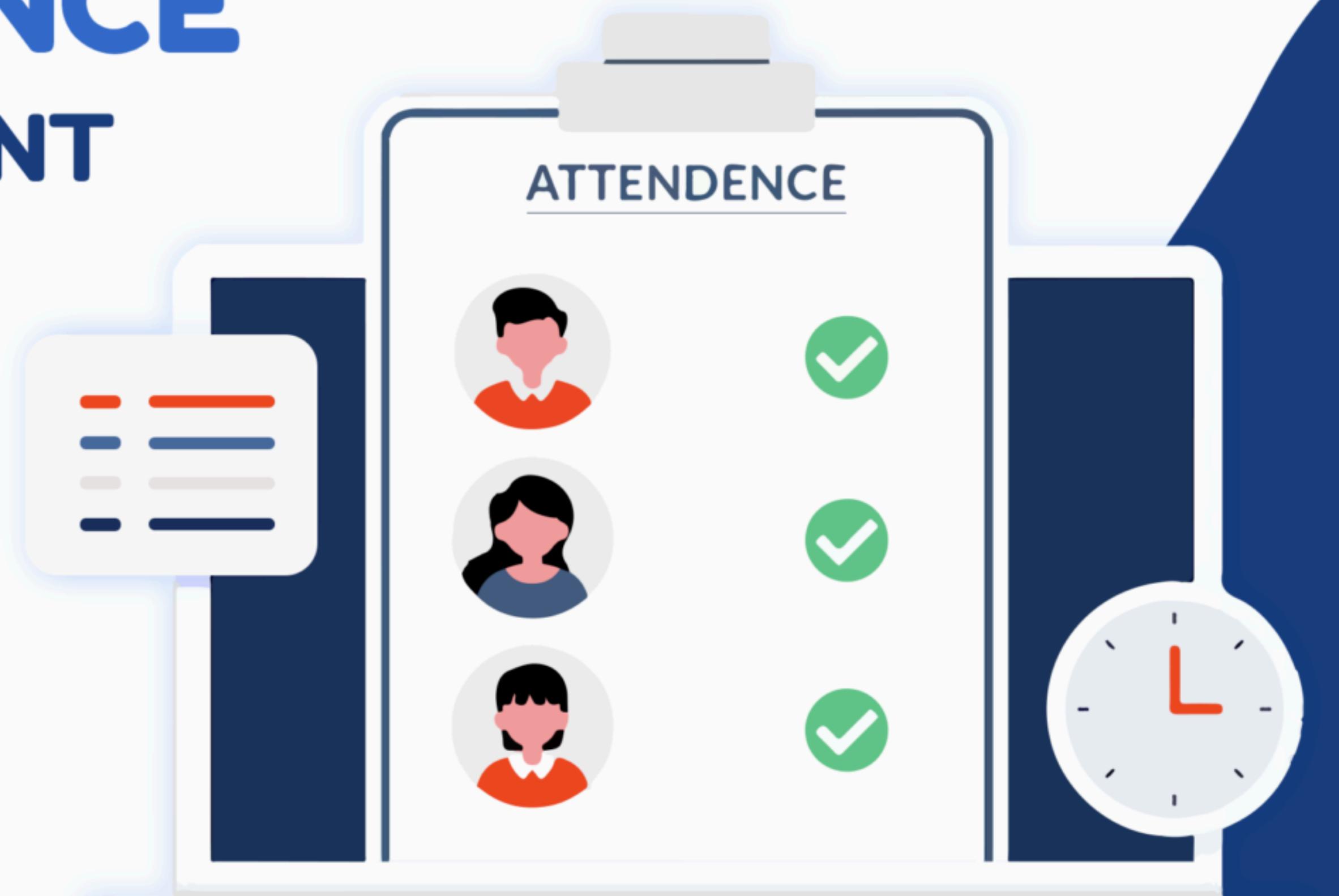
# Besant Technologies

Prepared By:

VIMAL RAJ.V



# ATTENDANCE MANAGEMENT SYSTEM



# VIEW TABLE

## TABLE 1:Departments

10 • describe Departments;

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

	Field	Type	Null	Key	Default	Extra
▶	Dept_id	int	NO	PRI	NULL	auto_increment
	Dept_Name	varchar(30)	YES		NULL	

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

	Dept_id	Dept_Name
▶	1	sales
	2	marketing
	3	analyst
	4	developer
	5	sales
	6	sales
	7	analyst
	8	marketing
	9	developer
	10	analyst

Departments 3 ×

Output

Action Output

## Table 2:Employees

29 | Emp\_id int,

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

	Field	Type	Null	Key	Default	Extra
▶	Emp_id	int	NO	PRI	NULL	auto_increment
	Emp_name	varchar(100)	YES		NULL	
	Dept_id	int	YES	MUL	NULL	
	Date_Of_Joining	date	YES		NULL	

Result 4 x

82 • select \* from Employees;

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

	Emp_id	Emp_name	Dept_id	Date_Of_Joining
▶	1	jeeva	1	2018-04-12
	2	babu	2	2017-01-29
	3	santhosh	3	2016-03-18
	4	kaviya	4	2012-12-12
	5	sabapathi	5	2019-11-30
	6	diwash	6	2020-04-24
	7	madhumitha	7	2021-06-12
	8	adithiyan	8	2023-08-19
	9	suji	9	2015-02-12
	10	gautham	10	2010-04-16

Employees 5 x

### Table 3:AttendanceRecords

34 • `describe AttendanceRecords;`

---

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

	Field	Type	Null	Key	Default	Extra
▶	Attendance_id	int	NO	PRI	NULL	auto_increment
	Emp_id	int	YES	MUL	NULL	
	ClockIn_Time	datetime	YES		NULL	
	ClockOut_Time	datetime	YES		NULL	
	WorkHours	decimal(5,2)	YES		NULL	
	Attendance_Date	date	YES		NULL	

94

95 • `select * from AttendanceRecords;`

---

Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

	Attendance_id	Emp_id	ClockIn_Time	ClockOut_Time	WorkHours	Attendance_Date
▶	1	1	2025-01-01 09:00:00	2025-01-01 17:00:00	8.00	2025-01-01
	2	2	2025-01-01 09:00:00	2025-01-01 17:00:00	8.70	2025-01-01
	3	3	2025-01-01 09:00:00	2025-01-01 17:00:00	8.00	2025-01-01
	4	4	2025-01-01 09:00:00	2025-01-01 17:00:00	8.00	2025-01-01
	5	5	2025-01-01 09:00:00	2025-01-01 17:00:00	8.75	2025-01-01
	6	6	2025-01-01 09:00:00	2025-01-01 17:00:00	8.00	2025-01-01
	7	7	2025-01-01 09:00:00	2025-01-01 17:00:00	8.00	2025-01-01
	8	8	2025-01-01 09:00:00	2025-01-01 17:00:00	8.65	2025-01-01
	9	9	2025-01-01 09:00:00	2025-01-01 17:00:00	8.55	2025-01-01
	10	10	2025-01-01 09:00:00	2025-01-01 17:00:00	8.45	2025-01-01

AttendanceRecords 7 ×

On behalf

## Table4:Leaverequests

```
44    );
45 • describe LeaveRequests;
```

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

	Field	Type	Null	Key	Default	Extra
▶	LeaveRequest_ID	int	NO	PRI	NULL	auto_increment
	Emp_id	int	YES	MUL	NULL	
	LeaveType	varchar(50)	YES		NULL	
	LeaveStartDate	date	YES		NULL	
	LeaveEndDate	date	YES		NULL	
	LeaveStatus	varchar(50)	YES		NULL	

108

```
109 • select * from LeaveRequests;
```

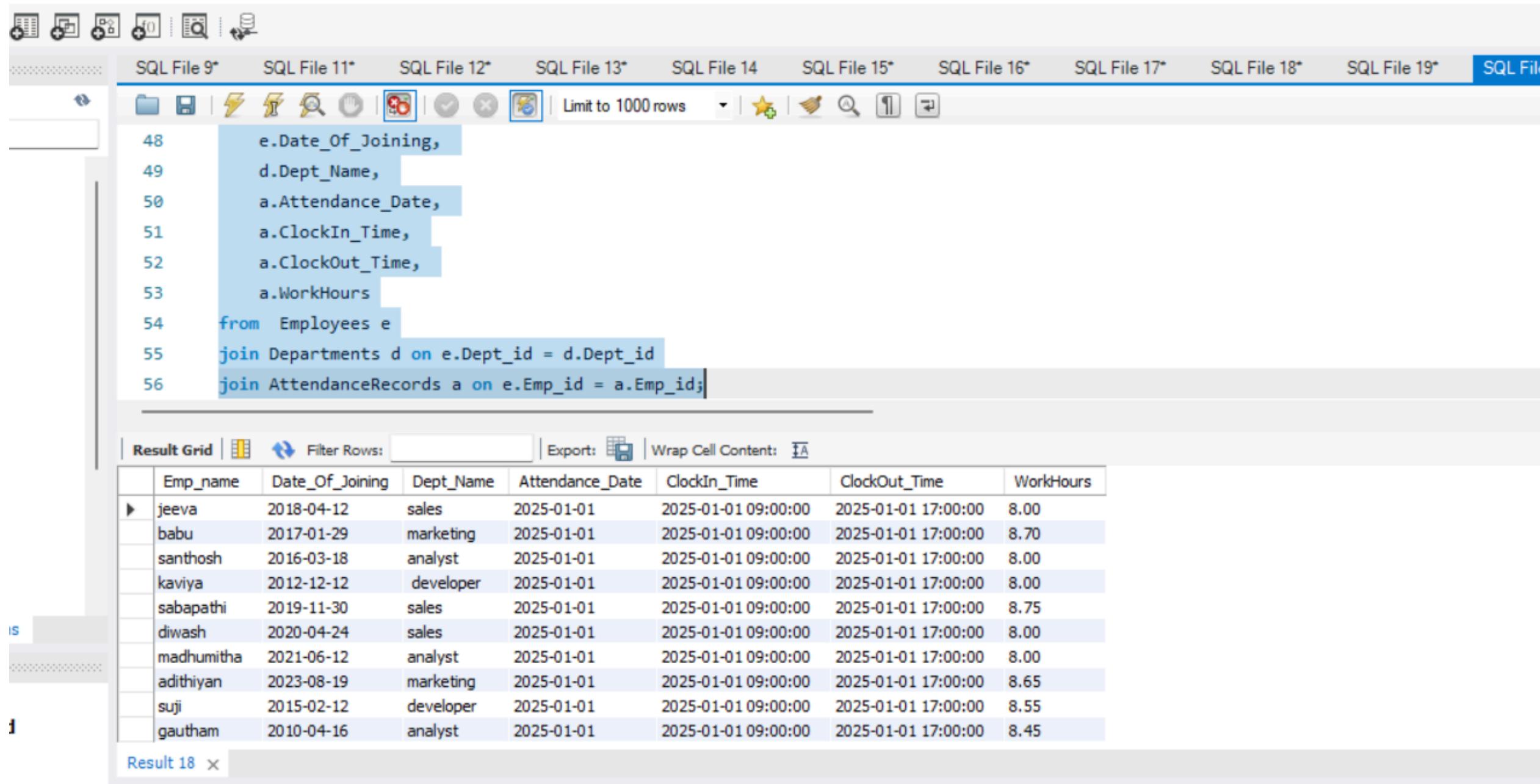
Result Grid | Filter Rows: Edit: Export/Import: Wrap Cell Content:

	LeaveRequest_ID	Emp_id	LeaveType	LeaveStartDate	LeaveEndDate	LeaveStatus
▶	1	1	helath issue	2025-01-10	2025-02-10	informed
	2	2	vocation	2025-01-21	2024-01-28	not informed
	3	3	vocation	2025-01-22	2024-01-28	not informed
	4	4	health issue	2025-01-19	2024-01-25	informed
	5	5	marriage	2025-01-18	2024-01-20	informed
	6	6	vocation	2025-01-19	2024-01-29	not informed
	7	7	marriage	2025-01-01	2024-01-26	informed
	8	8	relation death	2025-01-09	2024-01-16	informed
	9	9	marriage	2025-01-24	2024-01-29	informed
	10	10	health issue	2025-01-30	2024-02-12	informed

LeaveRequests 9 ×

Output :::::::::::::::::::::

# QUERY:to fetch employee details along with their department and attendance records.



The screenshot shows a SQL Server Management Studio (SSMS) interface. The top menu bar has tabs for 'SQL File 9\*' through 'SQL File 20\*'. Below the menu is a toolbar with various icons. The main area contains a SQL query:

```
48     e.Date_Of_Joining,  
49     d.Dept_Name,  
50     a.Attendance_Date,  
51     a.ClockIn_Time,  
52     a.ClockOut_Time,  
53     a.WorkHours  
54   from Employees e  
55   join Departments d on e.Dept_id = d.Dept_id  
56   join AttendanceRecords a on e.Emp_id = a.Emp_id;
```

Below the query is a 'Result Grid' with the following columns: Emp\_name, Date\_Of\_Joining, Dept\_Name, Attendance\_Date, ClockIn\_Time, ClockOut\_Time, and WorkHours. The data grid displays 18 rows of employee information:

	Emp_name	Date_Of_Joining	Dept_Name	Attendance_Date	ClockIn_Time	ClockOut_Time	WorkHours
▶	jeeva	2018-04-12	sales	2025-01-01	2025-01-01 09:00:00	2025-01-01 17:00:00	8.00
	babu	2017-01-29	marketing	2025-01-01	2025-01-01 09:00:00	2025-01-01 17:00:00	8.70
	santhosh	2016-03-18	analyst	2025-01-01	2025-01-01 09:00:00	2025-01-01 17:00:00	8.00
	kaviya	2012-12-12	developer	2025-01-01	2025-01-01 09:00:00	2025-01-01 17:00:00	8.00
	sabapathi	2019-11-30	sales	2025-01-01	2025-01-01 09:00:00	2025-01-01 17:00:00	8.75
	diwash	2020-04-24	sales	2025-01-01	2025-01-01 09:00:00	2025-01-01 17:00:00	8.00
	madhumitha	2021-06-12	analyst	2025-01-01	2025-01-01 09:00:00	2025-01-01 17:00:00	8.00
	adithiyan	2023-08-19	marketing	2025-01-01	2025-01-01 09:00:00	2025-01-01 17:00:00	8.65
	suji	2015-02-12	developer	2025-01-01	2025-01-01 09:00:00	2025-01-01 17:00:00	8.55
	gautham	2010-04-16	analyst	2025-01-01	2025-01-01 09:00:00	2025-01-01 17:00:00	8.45

At the bottom left of the result grid, it says 'Result 18'.

## Q2: Get Employees Who Took Leave in the Same Period as Another Employee

The screenshot shows the MySQL Workbench interface. The main window displays a SQL editor with the following query:

```
118 select 1
119 from LeaveRequests L2
120 where L2.Emp_id <> E.Emp_id
121 and (
122     (L1.LeaveStartDate between L2.LeaveStartDate and L2.LeaveEndDate)
123     or (L1.LeaveEndDate between L2.LeaveStartDate and L2.LeaveEndDate)
124 )
125 )
126 );
```

The result grid shows the following data:

Emp_name
babu
santhosh
kaviya
sabapathi
diwash
suji
gautham

A context help message on the right side of the interface states: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."

## Q3: Find employees who have taken leave on a specific date

The screenshot shows the MySQL Workbench interface. The main window displays a SQL editor with the following query:

```
135 );
```

The result grid shows the following data:

Emp_name
jeeva

# Q4:Find employees with more than 8 hours of work in a day

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Includes icons for Home, Schema, Table, View, Insert, Delete, Update, Select, and others.
- Navigator:** Schemas section lists databases: agg, college, company, customers, delivery, employee, food, for\_joins, groupbydemo, hospitalsystem, hostel, info, joins, list, members, my\_database, pg, pratice, purchase, sakila.
- SQL Editor:** Contains the following SQL code:

```
136
137
138 • select Emp_name
139   from Employees
140   where Emp_id in (
141     select Emp_id
142       from AttendanceRecords
143      where WorkHours > 8 and Attendance_Date = '2025-01-01'
144    )
```
- Result Grid:** Shows the results of the query in a table format:

Emp_name
babu
sabapathi
adithyan
saji
gautham
- Message Bar:** Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
- Status Bar:** Employees 13 x, Read Only, Context Help, Snippets.

# Q4: Count the Number of Employees in Each Department

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

- File Bar:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** Includes icons for Home, Schema, Table, View, Insert, Delete, Update, Select, and others.
- Navigator:** Schemas section lists databases: agg, college, company, customers, delivery, employee, food, for\_joins, groupbydemo, hospitalsystem, hostel, info, joins, list, members, my\_database, pg, pratice, purchase, sakila.
- SQL Editor:** Contains the following SQL code:

```
154
155 •  select ucname(Emp_name) from Employees;
156 •  show index from Employees;
157 •  create index index_doj on Employees (Date_of_Joining);
158
159 •  SELECT d.Dept_Name, COUNT(e.Emp_id) AS Employee_Count
160   FROM Departments d
161  LEFT JOIN Employees e ON d.Dept_id = e.Dept_id
162  GROUP BY d.Dept_Name;
```
- Result Grid:** Shows the results of the query in a table format:

Dept_Name	Employee_Count
sales	3
marketing	2
analyst	3
developer	1
developer	1
- Message Bar:** Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
- Status Bar:** Result 15 x, Read Only, Context Help, Snippets.

## Q5:Calculate the Total Work Hours for Each Employee

The screenshot shows the MySQL Workbench interface with the 'Schemas' tab selected. A result grid displays the following data:

Emp_name	Total_Work_Hours
jeeva	8.00
babu	8.70
santhosh	8.00
kaviya	8.00
sabapathi	8.75
diwash	8.00
madhumitha	8.00
adithyan	8.65
suji	8.55
gautham	8.45

## Q6:Find the Maximum and Minimum Work Hours for a Given Day

The screenshot shows the MySQL Workbench interface with the 'SQL' tab selected. The SQL editor contains the following query:

```
165 FROM Execute the statement under the keyboard cursor
166 JOIN AttendanceRecords a ON e.emp_id = a.emp_id
167 GROUP BY e.Emp_name;
168
169 • SELECT
170     MAX(WorkHours) AS Max_Work_Hours,
171     MIN(Workhours) AS Min_Work_Hours
172 FROM AttendanceRecords
173 WHERE Attendance_Date = '2025-01-01';
```

The results grid shows the output of the query:

Max_Work_Hours	Min_Work_Hours
8.75	8.00

The 'Output' pane at the bottom shows the execution log:

#	Time	Action	Message	Duration / Fetch
27	18:27:22	SELECT Emp_name from Employees where Emp_id in ( select Emp_id from LeaveRequests where ...)	1 row(s) returned	0.000 sec / 0.000 sec
28	18:27:38	select Emp_name from Employees where Emp_id in ( select Emp_id from AttendanceRecords where ...)	5 row(s) returned	0.000 sec / 0.000 sec
29	18:28:17	select ucase(Emp_name) from Employees LIMIT 0, 1000	10 row(s) returned	0.000 sec / 0.000 sec
30	18:28:35	SELECT d.Dept_Name, COUNT(e.Emp_id) AS Employee_Count FROM Departments d LEFT JOIN Employees e ON d.Dept_ID = e.Dept_ID WHERE e.Emp_id IN (SELECT Emp_id FROM AttendanceRecords WHERE Attendance_Date = '2025-01-01') GROUP BY d.Dept_Name	5 row(s) returned	0.000 sec / 0.000 sec

## Q7:ucase

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

- Navigator:** Shows the "SCHEMAS" tree with various databases listed.
- SQL Editor:** Displays the following SQL code:

```
151 •    select * from Employees;
152 •    select * from AttendanceRecords;
153 •    select * from LeaveRequests;
154
155 •    select ucase(Emp_name) from Employees;
156 •    show index from Employees;
157 •    create index index_doj on Employees (Date_of_Joining);
158
159 •    SELECT d.Dept_Name, COUNT(e.Emp_id) AS Employee_Count
```
- Result Grid:** Shows the output of the `select ucase(Emp_name)` query, displaying the names JEEVA, BABU, SANTHOSH, KAVIYA, SABAPATHI, DIWASH, MADHUMITHA, ADITHIYAN, SUJI, and GAUTHAM.
- Status Bar:** Shows "Result 14" and "Output".
- Help Message:** A message on the right states: "Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help."