Wage Manager

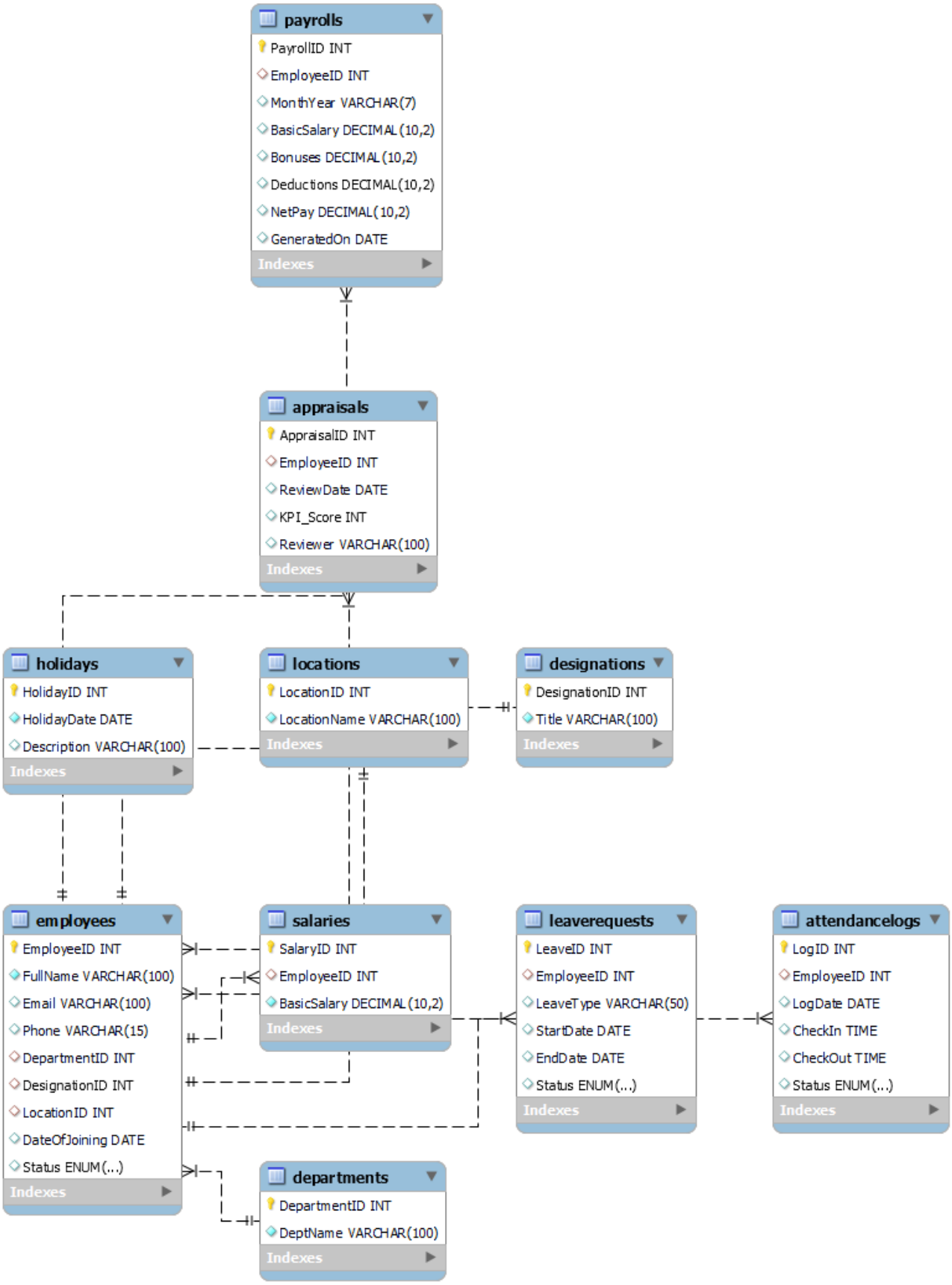
Name – Jaya Prajapati

INTRODUCTION

The HR & Payroll Management System is a structured SQL-based project developed to manage essential human resource data for an organization. It includes modules for handling employee records, department structures, attendance logs, leave requests, salary details, and performance appraisals.

The system is designed using MySQL with fully normalized relational tables to ensure data consistency and integrity. This project focuses on database creation and applies a wide range of SQL operations-covering DDL, DML, and DQL queries such as joins, subqueries, views, and aggregate functions—to demonstrate practical implementation of core database.

ER DIAGRAM


payrolls
PayrollID INT
EmployeeID INT
MonthYear VARCHAR(7)
BasicSalary DECIMAL(10,2)
Bonuses DECIMAL(10,2)
Deductions DECIMAL(10,2)
NetPay DECIMAL(10,2)
GeneratedOn DATE
Indexes
appraisals
AppraisalID INT
EmployeeID INT
ReviewDate DATE
KPI_Score INT
Reviewer VARCHAR(100)
Indexes
holidays
HolidayID INT
HolidayDate DATE
Description VARCHAR(100)
Indexes
locations
LocationID INT
LocationName VARCHAR(100)
Indexes
designations
DesignationID INT
Title VARCHAR(100)
Indexes
employees
EmployeeID INT
FullName VARCHAR(100)
Email VARCHAR(100)
Phone VARCHAR(15)
DepartmentID INT
DesignationID INT
LocationID INT
DateOfJoining DATE
Status ENUM(...)
Indexes
salaries
SalaryID INT
EmployeeID INT
BasicSalary DECIMAL(10,2)
Indexes
leaverequests
LeaveID INT
EmployeeID INT
LeaveType VARCHAR(50)
StartDate DATE
EndDate DATE
Status ENUM(...)
Indexes
attendancelogs
LogID INT
EmployeeID INT
LogDate DATE
CheckIn TIME
CheckOut TIME
Status ENUM(...)
Indexes
departments
DepartmentID INT
DeptName VARCHAR(100)
Indexes

Databases :

```
Create database wage_manager;

use wage_manager;

show databases;
```

Database
information_schema
jayadb
mysql
performance_schema
pizza_sales_analysis
sys
wage_manager

Tables in wage_manager database :

```
show tables;
```

Tables_in_wage_manager
appraisals
attendancelogs
departments
designations
employees
holidays
leaverequests
locations
payrolls
salaries

1. DATA DEFINITION LANGUAGE (DDL):

1.Creating Tables :

A. Departments

```
CREATE TABLE Departments (
  DepartmentID INT PRIMARY KEY AUTO_INCREMENT,
  DeptName VARCHAR(100) NOT NULL
);
desc departments;
```

Field	Type	Null	Key	Default	Extra
DepartmentID	int	NO	PRI	NULL	auto_increment
DeptName	varchar(100)	NO		NULL	

B. Designations

```
CREATE TABLE Designations (
  DesignationID INT PRIMARY KEY AUTO_INCREMENT,
  Title VARCHAR(100) NOT NULL
);
desc designations;
```

Field	Type	Null	Key	Default	Extra
DesignationID	int	NO	PRI	NULL	auto_increment
Title	varchar(100)	NO		NULL	

C. Locations

```
CREATE TABLE Locations (
  LocationID INT PRIMARY KEY AUTO_INCREMENT,
  LocationName VARCHAR(100) NOT NULL
);
desc Locations;
```

Field	Type	Null	Key	Default	Extra
LocationID	int	NO	PRI	NULL	auto_increment
LocationName	varchar(100)	NO		NULL	

D. Employees

```
CREATE TABLE Employees (
  EmployeeID INT PRIMARY KEY AUTO_INCREMENT,
  FullName VARCHAR(100) NOT NULL,
  Email VARCHAR(100) UNIQUE,
  Phone VARCHAR(15),
  DepartmentID INT,
  DesignationID INT,
  LocationID INT,
  DateOfJoining DATE,
  Status ENUM('Active', 'Inactive', 'Resigned') DEFAULT 'Active',
  FOREIGN KEY (DepartmentID) REFERENCES
Departments(DepartmentID),
  FOREIGN KEY (DesignationID) REFERENCES
Designations(DesignationID),
```

```
  FOREIGN KEY (LocationID) REFERENCES Locations(LocationID)
);
desc Employees;
```

Field	Type	Null	Key	Default	Extra
EmployeeID	int	NO	PRI	NULL	auto_increment
FullName	varchar(100)	NO		NULL	
Email	varchar(100)	YES	UNI	NULL	
Phone	varchar(15)	YES		NULL	
DepartmentID	int	YES	MUL	NULL	
DesignationID	int	YES	MUL	NULL	
LocationID	int	YES	MUL	NULL	
DateOfJoining	date	YES		NULL	
Status	enum('Active','Inactive','Resigned')	YES		Active	

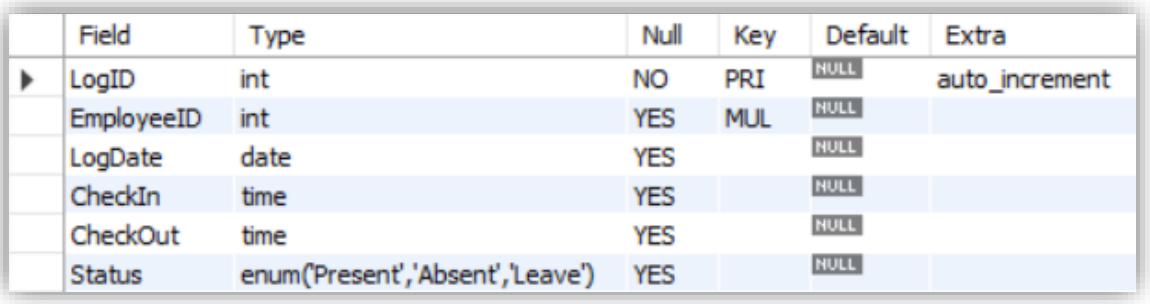
E. Salaries

```
CREATE TABLE Salaries (
  SalaryID INT PRIMARY KEY AUTO_INCREMENT,
  EmployeeID INT UNIQUE,
  BasicSalary DECIMAL(10,2) NOT NULL,
  FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)
);
desc Salaries;
```

Field	Type	Null	Key	Default	Extra
SalaryID	int	NO	PRI	NULL	auto_increment
EmployeeID	int	YES	UNI	NULL	
BasicSalary	decimal(10,2)	NO		NULL	

F. AttendanceLogs

```
CREATE TABLE AttendanceLogs (
  LogID INT PRIMARY KEY AUTO_INCREMENT,
  EmployeeID INT,
  LogDate DATE,
  CheckIn TIME,
  CheckOut TIME,
  Status ENUM('Present', 'Absent', 'Leave'),
  FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)
);
desc AttendanceLogs;
```

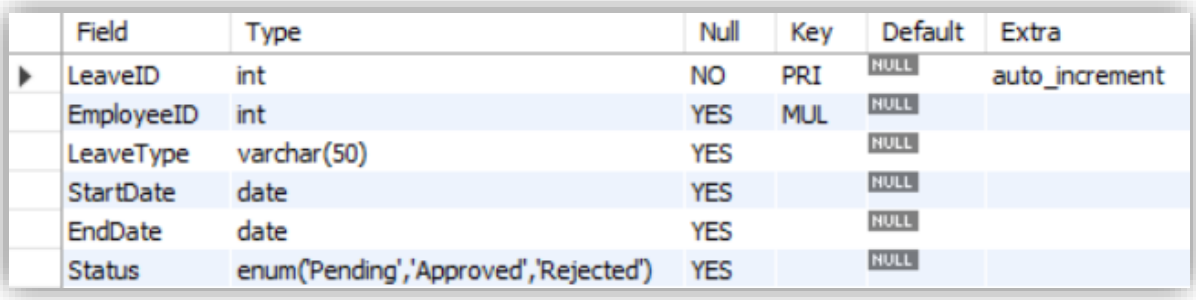


Field	Type	Null	Key	Default	Extra
LogID	int	NO	PRI	NULL	auto_increment
EmployeeID	int	YES	MUL	NULL	
LogDate	date	YES		NULL	
CheckIn	time	YES		NULL	
CheckOut	time	YES		NULL	
Status	enum('Present','Absent','Leave')	YES		NULL	

G. LeaveRequests

```
CREATE TABLE LeaveRequests (
  LeaveID INT PRIMARY KEY AUTO_INCREMENT,
  EmployeeID INT,
  LeaveType VARCHAR(50),
  StartDate DATE,
  EndDate DATE,
  Status ENUM('Pending', 'Approved', 'Rejected'),
  FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)
);
```

```
desc LeaveRequests;
```



	Field	Type	Null	Key	Default	Extra
▶	LeaveID	int	NO	PRI	NULL	auto_increment
	EmployeeID	int	YES	MUL	NULL	
	LeaveType	varchar(50)	YES		NULL	
	StartDate	date	YES		NULL	
	EndDate	date	YES		NULL	
	Status	enum('Pending','Approved','Rejected')	YES		NULL	

H. Payrolls

```
CREATE TABLE Payrolls (
  PayrollID INT PRIMARY KEY AUTO_INCREMENT,
  EmployeeID INT,
  MonthYear VARCHAR(7), -- Format: 'YYYY-MM'
  BasicSalary DECIMAL(10,2),
  Bonuses DECIMAL(10,2),
  Deductions DECIMAL(10,2),
  NetPay DECIMAL(10,2),
  GeneratedOn DATE DEFAULT (CURDATE()),
  FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)
);
desc Payrolls;
```

Field	Type	Null	Key	Default	Extra
PayrollID	int	NO	PRI	NULL	auto_increment
EmployeeID	int	YES	MUL	NULL	
MonthYear	varchar(7)	YES		NULL	
BasicSalary	decimal(10,2)	YES		NULL	
Bonuses	decimal(10,2)	YES		NULL	
Deductions	decimal(10,2)	YES		NULL	
NetPay	decimal(10,2)	YES		NULL	
GeneratedOn	date	YES		curdate()	DEFAULT_GENERATED

I. Appraisals

```
CREATE TABLE Appraisals (
  AppraisalID INT PRIMARY KEY AUTO_INCREMENT,
  EmployeeID INT,
  ReviewDate DATE,
  KPI_Score INT CHECK (KPI_Score BETWEEN 0 AND 100),
  Reviewer VARCHAR(100),
  FOREIGN KEY (EmployeeID) REFERENCES Employees(EmployeeID)
);
desc Appraisals;
```

Field	Type	Null	Key	Default	Extra
AppraisalID	int	NO	PRI	NULL	auto_increment
EmployeeID	int	YES	MUL	NULL	
ReviewDate	date	YES		NULL	
KPI_Score	int	YES		NULL	
Reviewer	varchar(100)	YES		NULL	

J. Holidays

```
CREATE TABLE Holidays (
  HolidayID INT PRIMARY KEY AUTO_INCREMENT,
  HolidayDate DATE NOT NULL,
  Description VARCHAR(100)
);
desc Holidays;
```

Field	Type	Null	Key	Default	Extra
HolidayID	int	NO	PRI	NULL	auto_increment
HolidayDate	date	NO		NULL	
Description	varchar(100)	YES		NULL	

2.Alter Table :

- **Alter Table: Add Column**

```
ALTER TABLE Employees ADD COLUMN Gender ENUM('Male', 'Female',
'Other');
```

Field	Type	Null	Key	Default	Extra
EmployeeID	int	NO	PRI	NULL	auto_increment
FullName	varchar(100)	NO		NULL	
Email	varchar(100)	YES	UNI	NULL	
Phone	varchar(15)	YES		NULL	
DepartmentID	int	YES	MUL	NULL	
DesignationID	int	YES	MUL	NULL	
LocationID	int	YES	MUL	NULL	
DateOfJoining	date	YES		NULL	
Status	enum('Active','Inactive','Resigned')	YES		Active	
Gender	enum('Male','Female','Other')	YES		NULL	

- **Alter Table: Modify Datatype**

ALTER TABLE Employees MODIFY COLUMN Phone VARCHAR(20);

Field	Type	Null	Key	Default	Extra
EmployeeID	int	NO	PRI	NULL	auto_increment
FullName	varchar(100)	NO		NULL	
Email	varchar(100)	YES	UNI	NULL	
Phone	varchar(20)	YES		NULL	
DepartmentID	int	YES	MUL	NULL	
DesignationID	int	YES	MUL	NULL	
LocationID	int	YES	MUL	NULL	
DateOfJoining	date	YES		NULL	
Status	enum('Active','Inactive','Resigned')	YES		Active	
Gender	enum('Male','Female','Other')	YES		NULL	

- **Alter Table: Rename Column**

ALTER TABLE Employees CHANGE COLUMN FullName Full_Name VARCHAR(100);

Field	Type	Null	Key	Default	Extra
EmployeeID	int	NO	PRI	NULL	auto_increment
Full_Name	varchar(100)	YES		NULL	
Email	varchar(100)	YES	UNI	NULL	
Phone	varchar(20)	YES		NULL	
DepartmentID	int	YES	MUL	NULL	
DesignationID	int	YES	MUL	NULL	
LocationID	int	YES	MUL	NULL	
DateOfJoining	date	YES		NULL	
Status	enum('Active','Inactive','Resigned')	YES		Active	
Gender	enum('Male','Female','Other')	YES		NULL	

- **Alter Table: Drop Column**

ALTER TABLE Employees DROP COLUMN Gender;

Field	Type	Null	Key	Default	Extra
EmployeeID	int	NO	PRI	NULL	auto_increment
Full_Name	varchar(100)	YES		NULL	
Email	varchar(100)	YES	UNI	NULL	
Phone	varchar(20)	YES		NULL	
DepartmentID	int	YES	MUL	NULL	
DesignationID	int	YES	MUL	NULL	
LocationID	int	YES	MUL	NULL	
DateOfJoining	date	YES		NULL	
Status	enum('Active','Inactive','Resigned')	YES		Active	

- **Alter Table: Rename Table**

RENAME TABLE Employees TO Staff;

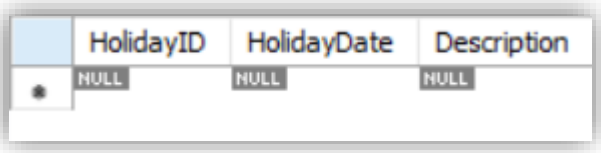
desc staff;

Field	Type	Null	Key	Default	Extra
EmployeeID	int	NO	PRI	NULL	auto_increment
Full_Name	varchar(100)	YES		NULL	
Email	varchar(100)	YES	UNI	NULL	
Phone	varchar(20)	YES		NULL	
DepartmentID	int	YES	MUL	NULL	
DesignationID	int	YES	MUL	NULL	
LocationID	int	YES	MUL	NULL	
DateOfJoining	date	YES		NULL	
Status	enum('Active','Inactive','Resigned')	YES		Active	

RENAME TABLE Staff TO Employees; -- Optional revert

3.Truncate Table :

```
TRUNCATE TABLE Holidays;
```



	HolidayID	HolidayDate	Description
*	NULL	NULL	NULL

4.Drop table :

```
DROP TABLE Holidays;
```

Tables_in_wage_manager
appraisals
attendancelogs
departments
designations
employees
leaverequests
locations
payrolls
salaries

2. DATA MANIPULATION LANGUAGE (DML):

1.Insert into Table

```
INSERT INTO Departments (DeptName) VALUES
('Human Resources'), ('Finance'), ('IT'), ('Marketing');
```

DepartmentID	DeptName
1	Human Resources
2	Finance
3	IT
4	Marketing
NULL	NULL

2.Update into Table

```
-- Update salary for a specific employee
UPDATE Salaries SET BasicSalary = 48000.00 WHERE EmployeeID = 1;
```

SalaryID	EmployeeID	BasicSalary
1	1	48000.00
2	2	60000.00
3	3	40000.00
4	4	75000.00
NULL	NULL	NULL

3.Delete from Table

```
-- Delete a leave request (e.g., cancelled leave)
DELETE FROM LeaveRequests WHERE LeaveID = 2;
```

	LeaveID	EmployeeID	LeaveType	StartDate	EndDate	Status
▶	1	3	Sick Leave	2025-07-01	2025-07-03	Approved
*	NULL	NULL	NULL	NULL	NULL	NULL

3. DATA QUERY LANGUAGE (DQL):

1.Select query

Q. Select Query for entire data

```
Select * from Appraisals;
```

	AppraisalID	EmployeeID	ReviewDate	KPI_Score	Reviewer
▶	1	1	2025-06-30	85	HR Manager
	2	2	2025-06-30	90	CTO
	3	3	2025-06-30	70	Finance Head
*	NULL	NULL	NULL	NULL	NULL

Q. Select specific data like employee ID and their KPI score

```
select employeeID, KPI_score from Appraisals;
```

	employeeID	KPI_score
▶	1	85
	2	90
	3	70

Q. Select query with changing column name

```
select Email as Email_IDs_of_Employees from Employees;
```

	Email_IDs_of_Employees
▶	amit@company.com
	jayaa@company.com
	neha@company.com
	ravi@company.com

2.Order by & Limit

Q. Select 1st two name of joiners and their joining date

```
SELECT Full_Name, DateOfJoining FROM Employees ORDER BY
DateOfJoining asc LIMIT 2;
```

	Full_Name	DateOfJoining
▶	Amit Patel	2020-07-10
	Ravi Mehta	2021-09-15

3.Distinct

Q. Distinct status from employee

```
SELECT DISTINCT Status FROM Employees;
```

	Status
▶	Active
	Resigned

4.Like

Q. Select email ending with @company.com

```
SELECT Full_Name, Email FROM Employees WHERE Email LIKE
'%@company.com';
```

Full_Name	Email
Jayaa Sharma	jayaa@company.com
Ravi Mehta	ravi@company.com
Neha Arora	neha@company.com
Amit Patel	amit@company.com

5.WHERE Clause

I. With comparison Operator

Q. Find employees who joined after 1st Jan 2023

```
SELECT Full_Name, DateOfJoining FROM Employees WHERE
DateOfJoining > '2023-01-01';
```

Full_Name	DateOfJoining
Neha Arora	2023-01-20

II. With Logical Operator

➢ AND

Q. Active employees in IT department

```
SELECT Full_Name FROM Employees WHERE Status = 'Active' AND
DepartmentID = 3;
```

Full_Name
Ravi Mehta

➢ OR

Q. Select Employee ID and leave type if request is for Sick leave or week off

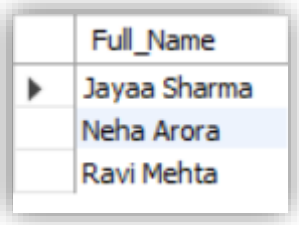
```
Select EmployeeID, LeaveType FROM LeaveRequests WHERE LeaveType = 'Sick Leave' OR 'WEEK OFF';
```

EmployeeID	LeaveType
3	Sick Leave

➢ NOT

Q. Employees not in 'Marketing' or 'Resigned'

```
SELECT Full_Name FROM Employees WHERE DepartmentID != 4 AND Status != 'Resigned';
```



Full_Name
Jayaa Sharma
Neha Arora
Ravi Mehta

➢ Not null

Q. Employees who joined after 1st Jan 2023 and have a phone number

```
SELECT Full_Name, DateOfJoining, Phone FROM Employees WHERE DateOfJoining > '2023-01-01' AND Phone IS NOT NULL;
```

	Full_Name	DateOfJoining	Phone
▶	Neha Arora	2023-01-20	7777766666

➢ BETWEEN

Q. Salaries between ₹40,000 and ₹60,000

```
SELECT * FROM Salaries WHERE BasicSalary BETWEEN 40000 AND 60000;
```

	SalaryID	EmployeeID	BasicSalary
▶	1	1	48000.00
	2	2	60000.00
	3	3	40000.00
*	NULL	NULL	NULL

➢ IN

Q. Employees from HR or IT (DeptID 1 and 3)

```
SELECT * FROM Employees WHERE DepartmentID IN (1, 3);
```

	EmployeeID	Full_Name	Email	Phone	DepartmentID	DesignationID	LocationID	DateOfJoining	Status
▶	1	Jayaa Sharma	jayaa@company.com	9999988888	1	2	1	2022-05-01	Active
	2	Ravi Mehta	ravi@company.com	8888877777	3	3	2	2021-09-15	Active
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

➢ ANY

Q. Salaries more than any of Employee 2 or 3

```
SELECT * FROM Salaries WHERE BasicSalary > ANY (SELECT BasicSalary FROM Salaries WHERE EmployeeID IN (2, 3));
```

SalaryID	EmployeeID	BasicSalary
1	1	48000.00
2	2	60000.00
4	4	75000.00
NULL	NULL	NULL

➢ ALL

Q. Salaries more than both Employee 2 and 3

```
SELECT * FROM Salaries WHERE BasicSalary > ALL (SELECT
BasicSalary FROM Salaries WHERE EmployeeID IN (2, 3));
```

SalaryID	EmployeeID	BasicSalary
4	4	75000.00
NULL	NULL	NULL

6.Aggregates

1. Count

Q. How many employees are in the company?

```
SELECT COUNT(*) AS TotalEmployees FROM Employees;
```

TotalEmployees
4

2. Sum

Q. What is the total payroll paid last month?

```
SELECT SUM(NetPay) AS TotalPayroll FROM Payrolls WHERE MonthYear
= '2025-06';
```

	TotalPayroll
▶	149500.00

3. Average

Q. What is the average salary of all employees?

```
SELECT AVG(BasicSalary) AS AverageSalary FROM Salaries;
```

	AverageSalary
▶	55750.000000

4. Min & Max

Q. What is the highest & lowest basic salary

```
SELECT MAX(BasicSalary) AS HighestSalary, MIN(BasicSalary) AS
LowestSalary FROM Salaries;
```

	HighestSalary	LowestSalary
▶	75000.00	40000.00

7. Group by clause

Q. Count employees per department

```
SELECT DepartmentID, COUNT(*) AS EmployeeCount FROM Employees
GROUP BY DepartmentID;
```

	DepartmentID	EmployeeCount
▶	1	1
	2	1
	3	1
	4	1

8.Having clause

Q. Departments with more than 1 employee

```
SELECT DepartmentID, COUNT(*) AS Total FROM Employees GROUP BY DepartmentID HAVING Total > 1;
```

	DepartmentID	Total

9.Joins

➢ Inner join

Q. Employee names with their department names

```
SELECT e.Full_Name, d.DeptName FROM Employees e INNER JOIN Departments d ON e.DepartmentID = d.DepartmentID;
```

	Full_Name	DeptName
▶	Jayaa Sharma	Human Resources
	Ravi Mehta	IT
	Neha Arora	Finance
	Amit Patel	Marketing

➢ Left join

Q. All employees, even if department missing

```
SELECT e.Full_Name, d.DeptName FROM Employees e LEFT JOIN
Departments d ON e.DepartmentID = d.DepartmentID;
```

Full_Name	DeptName
Jayaa Sharma	Human Resources
Ravi Mehta	IT
Neha Arora	Finance
Amit Patel	Marketing

➢ Right join

Q. All departments, even if no employees

```
SELECT d.DeptName, e.Full_Name FROM Departments d RIGHT JOIN
Employees e ON d.DepartmentID = e.DepartmentID;
```

DeptName	Full_Name
Human Resources	Jayaa Sharma
IT	Ravi Mehta
Finance	Neha Arora
Marketing	Amit Patel

➢ Self join

Q. Select Query for entire data

```
SELECT e1.FullName AS Employee1, e2.FullName AS Employee2,
e1.DepartmentID FROM Employees e1 JOIN Employees e2 ON
e1.DepartmentID = e2.DepartmentID WHERE e1.EmployeeID <
e2.EmployeeID;
```

Employee1	Employee2	DepartmentID

➢ Cross join

Q. Select Query for entire data

```
SELECT e.Full_Name, l.LocationName FROM Employees e CROSS
JOIN Locations l;
```

Full_Name	LocationName
Jayaa Sharma	Branch Office - Mumbai
Jayaa Sharma	Branch Office - Delhi
Jayaa Sharma	Head Office
Ravi Mehta	Branch Office - Mumbai
Ravi Mehta	Branch Office - Delhi
Ravi Mehta	Head Office
Neha Arora	Branch Office - Mumbai
Neha Arora	Branch Office - Delhi
Neha Arora	Head Office
Amit Patel	Branch Office - Mumbai
Amit Patel	Branch Office - Delhi
Amit Patel	Head Office

10. Subqueries

1. Single row Subquery

Q. Who is earning the highest salary?

```
SELECT * FROM Employees WHERE EmployeeID = (SELECT EmployeeID
FROM Salaries  WHERE BasicSalary = (SELECT MAX(BasicSalary) FROM
Salaries));
```

EmployeeID	Full_Name	Email	Phone	DepartmentID	DesignationID	LocationID	DateOfJoining	Status
4	Amit Patel	amit@company.com	6666655555	4	1	1	2020-07-10	Resigned
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

2. Multiple row Subquery

Q. Show employees who have taken any leave.

```
SELECT * FROM Employees WHERE EmployeeID IN (SELECT DISTINCT EmployeeID FROM LeaveRequests);
```

EmployeeID	Full_Name	Email	Phone	DepartmentID	DesignationID	LocationID	DateOfJoining	Status
3	Neha Arora	neha@company.com	7777766666	2	4	3	2023-01-20	Active
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3. Multiple column Subquery

Q. Show salary details of employees who received appraisals on June 30, 2025.

```
SELECT * FROM Salaries WHERE EmployeeID IN (SELECT EmployeeID FROM Appraisals WHERE ReviewDate = '2025-06-30');
```

SalaryID	EmployeeID	BasicSalary
1	1	48000.00
2	2	60000.00
3	3	40000.00
NULL	NULL	NULL

11. UNION

Q. Combine names of employees and reviewers

```
SELECT Full_Name AS Person FROM Employees UNION SELECT Reviewer FROM Appraisals;
```

Person
Jayaa Sharma
Ravi Mehta
Neha Arora
Amit Patel
HR Manager
CTO
Finance Head

12. View Creation

Q. Create a view – Active employees with salary

```
CREATE VIEW ActiveEmployeesWithSalary AS SELECT e.Full_Name,
s.BasicSalary FROM Employees e JOIN Salaries s ON e.EmployeeID =
s.EmployeeID WHERE e.Status = 'Active';
```

Q. Query the view

```
SELECT * FROM ActiveEmployeesWithSalary;
```

Full_Name	BasicSalary
Jayaa Sharma	48000.00
Ravi Mehta	60000.00
Neha Arora	40000.00