

**Faculty of Engineering and Natural Sciences Software Engineering**

### Table of Contents

[INTRODUCTION TO THE Employee Company 2](#_bookmark0)

[THE DATABASE DESIGN 4](#_bookmark1)

-[An Entity Relationship Diagram (ERD( 4](#_bookmark2)

-[An Enhanced Entity Relationship Diagram (EERD) 5](#_bookmark3)

[Data Dictionary 6](#_bookmark4)

-[THE TABLES IN THE DATABASE OF THE Employee Company 7](#_bookmark5)

[The Implementation and Loading 10](#_bookmark6)

-[The create statements: 10](#_bookmark7)

-[THE INSERT STATEMENTS 19](#_bookmark8)

-[Trigger function 23](#_bookmark9)

[-Virtually generated column 23](#_TOC_250000)

Show data using select 24

[Testing and Evaluation 26](#_bookmark10)

Conclusion 42

[REFERENCES 43](#_bookmark11)

# INTRODUCTION TO THE EMPLOYEE COMPANY

**INTRODUCTION**

The "EmployeeCompany" database is a comprehensive system designed to manage various aspects of an organization's workforce. It consists of tables that capture critical information about employees, departments, job roles, projects, and attendance.

Here's a breakdown of the main tables:

* Manager (Manager): This table stores data about managers, including unique ManagerIDs and their corresponding names (ManegeName).
* Employee: This table holds detailed employee information such as names, gender, date of birth, contact details (phone numbers and email addresses), age, and their association with managers through the Maneger\_ID.
* Department: Contains information about different organizational units, defining them by unique DepartmentIDs and descriptive DepartmentNames.
* JobRole: Represents various job roles available within the company, identified uniquely.
* JobRole\_Department: Links job roles to departments and tracks employees' tenures in specific roles within departments, recording start and end dates for each role.
* Attendance: Logs employee attendance by date and time, capturing clock-in and clock-out data.
* Project: Records project details, including names and start/end dates.
* ProjectTeam: Connects employees to projects, detailing the composition of teams involved in various projects.

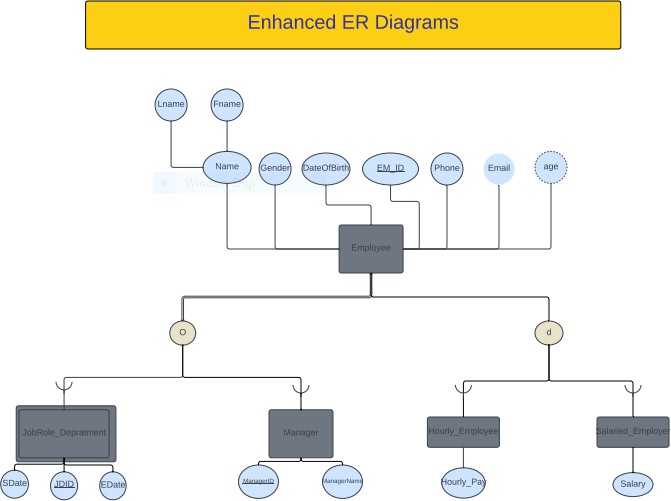
This structured database enables efficient management of employee information, departmental structuring, job role assignments, project allocations, and accurate attendance tracking, contributing to smoother organizational operations.

### THE DATABASE DESIGN

### An Entity Relationship Diagram (ERD)



An Enhanced Entity Relationship Diagram (EERD) of one entity



**Data Dictionary**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **TABLE NAME** | **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| **Project** | ProjectName ProjectStartDate ProjectEndData | Varchar(45) Date  Date | NO NO YES | PRI | NULL NULL NULL |  |
| **ProjectTeam** | TeamID Employee\_idEmployee Project\_ProjectName | Int Int  Varchar(45) | NO NO NO | PRI PRI PRI | NULL NULL NULL |  |
| **Hourly\_Employee** | Hourly\_Pay Employee\_idEmployee | Int int | NO NO |  | NULL NULL |  |
|  | PRI |  |
| **Manager** | ManagerID ManageName | Int Varchar(45) | NO NO | PRI | NULL NULL |  |
| **Employee** | EM\_ID  FirstName LastName Gender DateOfBirth Phone Email  Age Manager\_ID | Int Varchar(45) Varchar(45) Varchar(45) Date  Int Varchar(45) Int  Int | NO NO NO NO NO YES YES NO  YES | PRI  MUL | NULL NULL NULL NULL NULL NULL NULL NULL  NULL |  |
| **Department** | DepartmentID DepartmentName | Varchar(5) Varchar(255) | NO NO | PRI | NULL NULL |  |
| **JobRole\_Department** | JobRoleDepartmentID JobRole\_JobName Department\_ID Employee\_idEmployee StartDate  EndDate | Int Varchar(45) Varchar(5) Int  Date Date | NO NO NO NO NO  NO | PRI PRI PRI PRI | NULL NULL NULL NULL NULL  NULL |  |
| **Attandance** | AID  Date ClockInTime ClockOutTime  Employee\_idEmployee | Int Date  Varchar(45)  Varchar(45) Int | NO NO NO NO  NO | PRI  PRI | NULL NULL NULL NULL  NULL |  |
| **Salaried\_employee** | Salary  Employee\_idEmployee | Int  Int | NO  NO |  | NULL  NULL |  |
|  | PRI |  |
| **JobRole** | JobName | Varchar(45) | NO | PRI | NULL |  |

### THE TABLES IN THE DATABASE OF THE Employee Company

**Employee**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **EM\_ ID** | **FirstN ame** | **LastNa me** | **Gen der** | **DateOf Birth** | **Phone** | **Email** | **Ag e** | **Manag er\_ID** |
| **1** | John | Doe | Male | 1990-  01-15 | 123456  7890 | [John.doe@example](mailto:John.doe@example.com)  [.com](mailto:John.doe@example.com) | 32 | NULL |
| **2** | Jane | Smith | Fem  ale | 1995-  05-20 | 505271  638 | [Jane.smith@exampl](mailto:Jane.smith@example.com)  [e.com](mailto:Jane.smith@example.com) | 28 | 1 |
| **3** | Bob | Johnso  n | Male | 1985-  09-10 | 508271  938 | [bob.johnson@exam](mailto:bob.johnson@example.com)  [ple.com](mailto:bob.johnson@example.com) | 38 | 5 |
| **4** | Alice | Willia  ms | Fem  ale | 1992-  12-05 | 539127  933 | [alice.williams@exa](mailto:alice.williams@example.com)  [mple.com](mailto:alice.williams@example.com) | 31 | 5 |
| **5** | Charlie | Brown | Male | 1988-  07-25 | 599274  392 | charlie.brown@exa  mple.com | 35 | NULL |

**Manager**

|  |  |
| --- | --- |
| **ManagerID** | **ManagerName** |
| 1 | John Doe |
| 2 | Charli Brown |

**Department**

|  |  |
| --- | --- |
| **DepartmentID** | **DepartmentName** |
| FM | Finance management |
| HR | Human Resources |
| IT | Information Technology |

**Jobrole\_department**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **JobRoleDepart**  **mentID** | **JobRole\_Job**  **Name** | **Departme**  **nt\_ID** | **Employee\_idEm**  **ployee** | **StartD**  **ate** | **EndD**  **ate** |
| **1** | IT Specialist | IT | 1 | 2016-  11-16 | 2020-  04-12 |
| **2** | HR  Coordinator | HR | 2 | 2017-  06-09 | 2019-  07-19 |
| **3** | Accountant | FM | 4 | 2019-  04-12 | 2021-  10-01 |
| **4** | HR  Coordinator | HR | 3 | 2018-  05-20 | 2020-  08-03 |
| **5** | Accountant | FM | 5 | 2015-  07-12 | 2022-  04-28 |

**Salaried\_employee**

|  |  |
| --- | --- |
| **Salary** | **Employee\_idEmployee** |
| 120000 | 1 |
| 55000 | 2 |
| 70000 | 4 |

**Hourly\_employee**

|  |  |
| --- | --- |
| **Hourly\_Pay** | **Employee\_idEmployee** |
| 20 | 3 |
| 18 | 5 |

**Project**

|  |  |  |
| --- | --- | --- |
| **ProjectName** | **ProjectStartDate** | **ProjectEndDate** |
| Project A | 2023-02-01 | 2023-04-30 |
| Project B | 2023-03-15 | 2023-06-30 |
| Project C | 2023-04-10 | 2023-07-15 |

**Projectteam**

|  |  |  |
| --- | --- | --- |
| **TeamID** | **Employee\_idEmployee** | **Project\_ProjectName** |
| 1 | 1 | Project A |
| 1 | 2 | Project A |
| 2 | 3 | Project B |
| 2 | 4 | Project B |
| 2 | 5 | Project C |

**Attandance**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **AID** | **Date** | **ClockInTime** | **ClockOutTime** | **Employee\_idEmployee** |
| 1 | 2023-01-  15 | 09:00:00 | 17:00:00 | 1 |
| 2 | 2023-01-  15 | 09:30:00 | 18:00:00 | 2 |
| 3 | 2023-01-  15 | 10:00:00 | 16:30:00 | 3 |
| 4 | 2023-01-  15 | 08:45:00 | 17:15:00 | 4 |
| 5 | 2023-01-  15 | 09:15:00 | 17:45:00 | 5 |

**JobRole**

|  |
| --- |
| **JobName** |
| Accountant |
| HR Coordinator |
| IT Specialist |

### The Implementation and Loading The create statements:

These are the create MYSQL statements that are used to create the database (Employee Company) and its tables which include the

Project,ProjectTeam,Hourly\_Employee,Manager,Employee,Department,JobRole\_ Department,Attandance,Salaried\_employee.

Below is the list of the create statements for the database and its tables:

CREATE SCHEMA IF NOT EXISTS `EmployeeCompany` DEFAULT CHARACTER

SET utf8;

USE `EmployeeCompany`;

CREATE TABLE IF NOT EXISTS `EmployeeCompany`.`Maneger` (

`ManagerID` INT(5) NOT NULL,

`ManegeName` VARCHAR(45) NOT NULL, PRIMARY KEY (`ManagerID`)

) ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `EmployeeCompany`.`Employee` (

##### `EM\_ID` INT(5) NOT NULL,

`FirstName` VARCHAR(45) NOT NULL,

`LastName` VARCHAR(45) NOT NULL,

`Gender` VARCHAR(45) NOT NULL,

`DateOfBirth` DATE NOT NULL,

`Phone` INT(10) NULL,

`Email` VARCHAR(45) NULL,

`Age` INT(2) NOT NULL,

`Maneger\_ID` INT(5) NULL, PRIMARY KEY (`EM\_ID`),

CONSTRAINT `fk\_Employee\_Maneger1` FOREIGN KEY (`Maneger\_ID`)

REFERENCES `EmployeeCompany`.`Maneger` (`ManagerID`) ON DELETE SET NULL

##### ON UPDATE CASCADE

) ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `EmployeeCompany`.`Department` (

`DepartmentID` VARCHAR(5) NOT NULL,

`DepartmentName` VARCHAR(255) NOT NULL, PRIMARY KEY (`DepartmentID`)

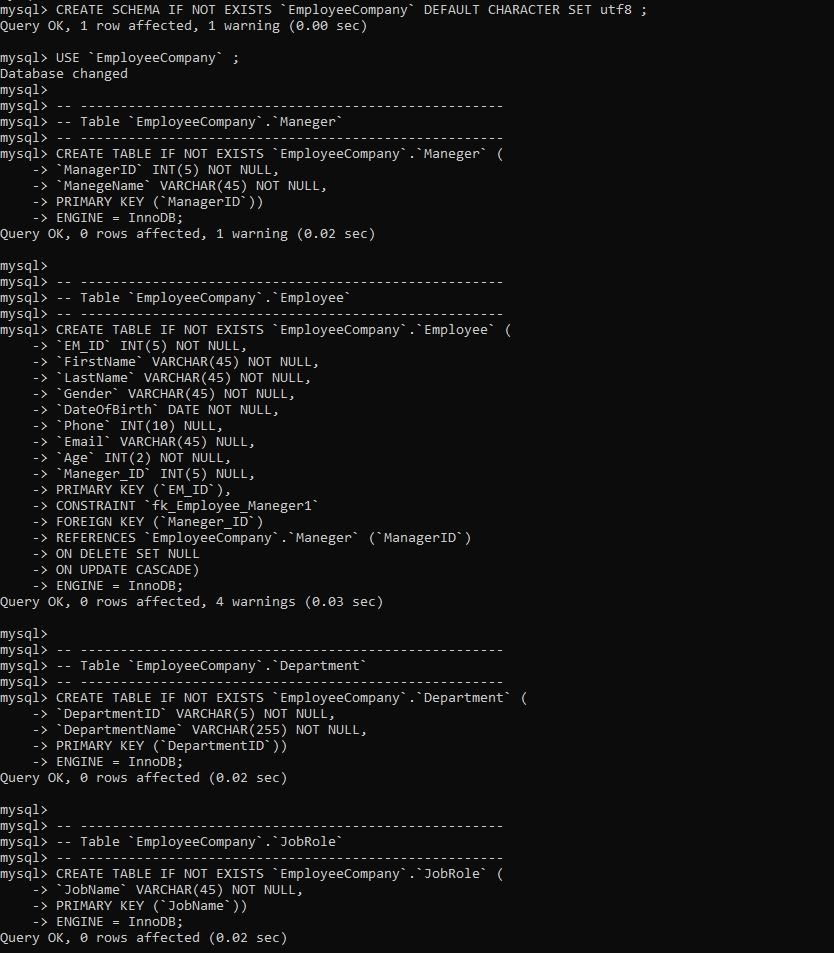
) ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `EmployeeCompany`.`JobRole` (

`JobName` VARCHAR(45) NOT NULL,

PRIMARY KEY (`JobName`)

) ENGINE = InnoDB;



CREATE TABLE IF NOT EXISTS `EmployeeCompany`.`JobRole\_Department` (

`JobRoleDepartmentID` INT(5) NOT NULL,

`JobRole\_JobName` VARCHAR(45) NOT NULL,

`Department\_ID` VARCHAR(5) NOT NULL,

`Employee\_idEmployee` INT(5) NOT NULL,

`StartDate` DATE NOT NULL,

`EndDate` DATE NOT NULL,

PRIMARY KEY (`JobRoleDepartmentID`, `JobRole\_JobName`, `Department\_ID`,

`Employee\_idEmployee`),

CONSTRAINT `fk\_EmployeeDepartment\_JobRole1` FOREIGN KEY (`JobRole\_JobName`)

REFERENCES `EmployeeCompany`.`JobRole` (`JobName`) ON DELETE CASCADE

##### ON UPDATE CASCADE,

CONSTRAINT `fk\_EmployeeDepartment\_Department1` FOREIGN KEY (`Department\_ID`)

REFERENCES `EmployeeCompany`.`Department` (`DepartmentID`) ON DELETE CASCADE

##### ON UPDATE CASCADE,

CONSTRAINT `fk\_EmployeeDepartment\_Employee1` FOREIGN KEY (`Employee\_idEmployee`)

REFERENCES `EmployeeCompany`.`Employee` (`EM\_ID`) ON DELETE CASCADE

##### ON UPDATE CASCADE

) ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `EmployeeCompany`.`Attandance` (

##### `AID` INT(5) NOT NULL,

`Date` DATE NOT NULL,

`ClockInTime` VARCHAR(45) NOT NULL,

`ClockOutTime` VARCHAR(45) NOT NULL,

`Employee\_idEmployee` INT(10) NOT NULL, PRIMARY KEY (`AID`, `Employee\_idEmployee`), CONSTRAINT `fk\_Attandance\_Employee1` FOREIGN KEY (`Employee\_idEmployee`)

REFERENCES `EmployeeCompany`.`Employee` (`EM\_ID`) ON DELETE CASCADE

##### ON UPDATE CASCADE

) ENGINE = InnoDB;



CREATE TABLE IF NOT EXISTS `EmployeeCompany`.`Project` (

`ProjectName` VARCHAR(45) NOT NULL,

`ProjectStartDate` DATE NOT NULL,

`ProjectEndDate` DATE NULL, PRIMARY KEY (`ProjectName`)

) ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `EmployeeCompany`.`ProjectTeam` (

`TeamID` INT(5) NOT NULL,

`Employee\_idEmployee` INT(5) NOT NULL,

`Project\_ProjectName` VARCHAR(45) NOT NULL,

PRIMARY KEY (`TeamID`, `Employee\_idEmployee`, `Project\_ProjectName`), CONSTRAINT `fk\_ProjectTeam\_Employee1`

FOREIGN KEY (`Employee\_idEmployee`)

REFERENCES `EmployeeCompany`.`Employee` (`EM\_ID`) ON DELETE CASCADE

##### ON UPDATE NO ACTION,

CONSTRAINT `fk\_ProjectTeam\_Project1` FOREIGN KEY (`Project\_ProjectName`)

REFERENCES `EmployeeCompany`.`Project` (`ProjectName`) ON DELETE CASCADE

##### ON UPDATE NO ACTION

) ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `EmployeeCompany`.`Salaried\_Employee` (

`Salary` INT(5) NOT NULL,

`Employee\_idEmployee` INT(5) NOT NULL, PRIMARY KEY (`Employee\_idEmployee`), CONSTRAINT `fk\_Salaried\_Employee\_Employee` FOREIGN KEY (`Employee\_idEmployee`)

REFERENCES `EmployeeCompany`.`Employee` (`EM\_ID`) ON DELETE CASCADE

##### ON UPDATE CASCADE

) ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `EmployeeCompany`.`Hourly\_Employee` (

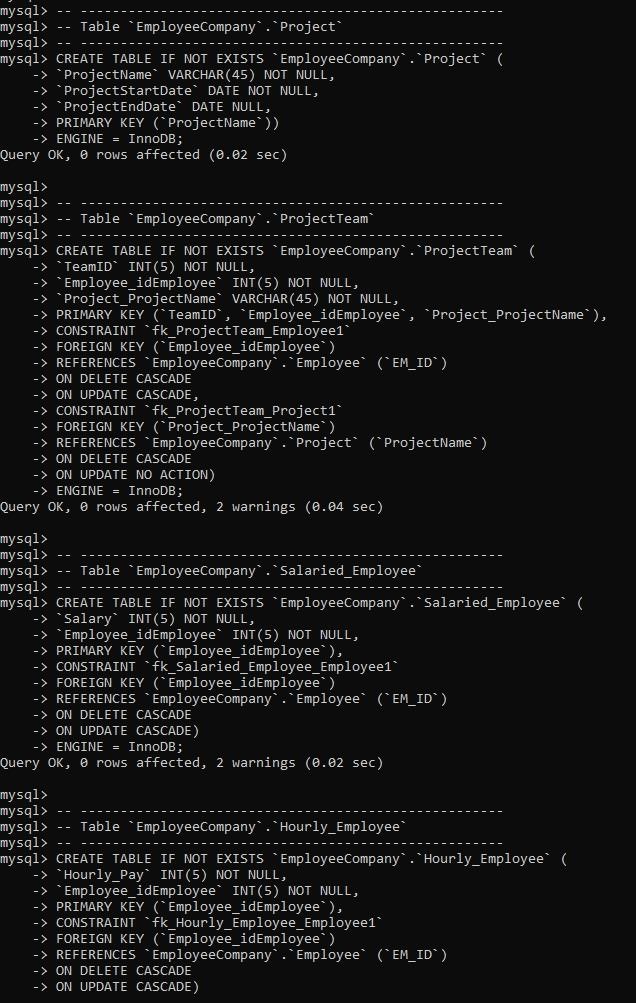
`Hourly\_Pay` INT(5) NOT NULL,

`Employee\_idEmployee` INT(5) NOT NULL, PRIMARY KEY (`Employee\_idEmployee`), CONSTRAINT `fk\_Hourly\_Employee\_Employee` FOREIGN KEY (`Employee\_idEmployee`)

REFERENCES `EmployeeCompany`.`Employee` (`EM\_ID`) ON DELETE CASCADE

##### ON UPDATE CASCADE

) ENGINE = InnoDB;



### THE INSERT STATEMENTS

These are the MYSQL statements that are used to insert a value into the tables,

**Project,ProjectTeam,Hourly\_Employee,Manager,Employee,Department,JobRole\_Depart mentAttandance,Salaried\_employee.**

**TABLE NAME: Project**

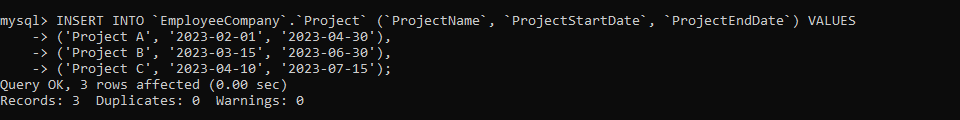
INSERT INTO `EmployeeCompany`.`Project` (`ProjectName`, `ProjectStartDate`,

`ProjectEndDate`) VALUES

('Project A', '2023-02-01', '2023-04-30'),

('Project B', '2023-03-15', '2023-06-30'),

('Project C', '2023-04-10', '2023-07-15');



**TABLE NAME: ProjectTeam**

INSERT INTO `EmployeeCompany`.`ProjectTeam` (`TeamID`, `Employee\_idEmployee`,

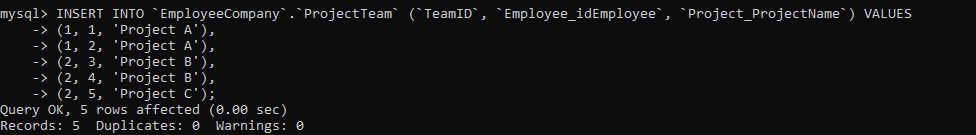
`Project\_ProjectName`) VALUES (1, 1, 'Project A'),

(1, 2, 'Project A'),

(2, 3, 'Project B'),

(2, 4, 'Project B'),

(2, 5, 'Project C');

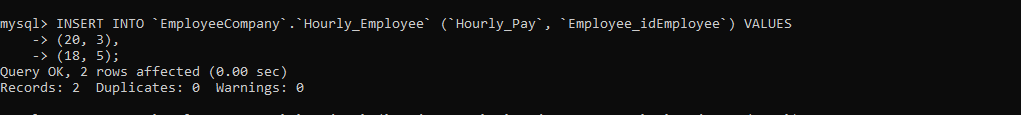


**TABLE NAME: Hourly\_Employee**

INSERT INTO `EmployeeCompany`.`Hourly\_Employee` (`Hourly\_Pay`,

`Employee\_idEmployee`) VALUES (20, 3),

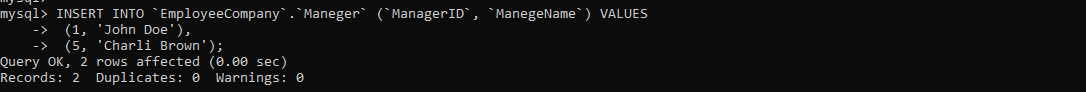
(18, 5);



**TABLE NAME: Manager**

INSERT INTO `EmployeeCompany`.`Maneger` (`ManagerID`, `ManegeName`) VALUES (1, 'John Doe'),

(5, 'Charli Brown');



**TABLE NAME: Employee**

INSERT INTO `EmployeeCompany`.`Employee` (`EM\_ID`, `FirstName`, `LastName`,

`Gender`, `DateOfBirth`, `Phone`, `Email`, `Age`, `Maneger\_ID`) VALUES

(1, 'John', 'Doe', 'Male', '1990-01-15', 1234567890, 'john.doe@example.com', 32, Null),

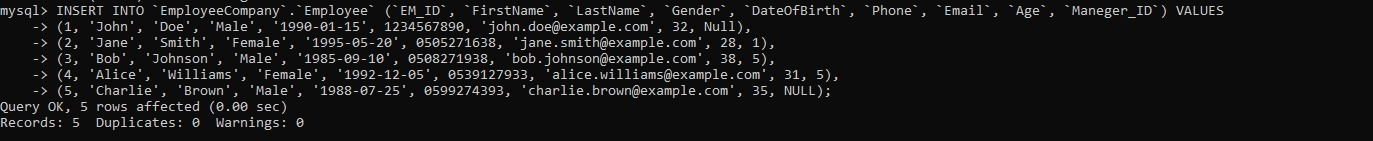
(2, 'Jane', 'Smith', 'Female', '1995-05-20', 0505271638, 'jane.smith@example.com', 28, 1),

(3, 'Bob', 'Johnson', 'Male', '1985-09-10', 0508271938, 'bob.johnson@example.com', 38, 5),

(4, 'Alice', 'Williams', 'Female', '1992-12-05', 0539127933, 'alice.williams@example.com', 31,

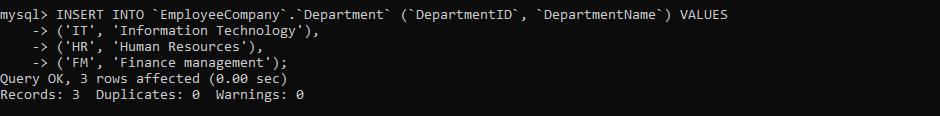
5),

(5, 'Charlie', 'Brown', 'Male', '1988-07-25', 0599274393, 'charlie.brown@example.com', 35, NULL);



**TABLE NAME: Department**

INSERT INTO `EmployeeCompany`.`Department` (`DepartmentID`, `DepartmentName`) VALUES

('IT', 'Information Technology'), ('HR', 'Human Resources'), ('FM', 'Finance management');

**TABLE NAME: JobRole\_Department**

INSERT INTO `EmployeeCompany`.`JobRole\_Department` (`JobRoleDepartmentID`,

`JobRole\_JobName`, `Department\_ID`, `Employee\_idEmployee`, `StartDate`, `EndDate`) VALUES

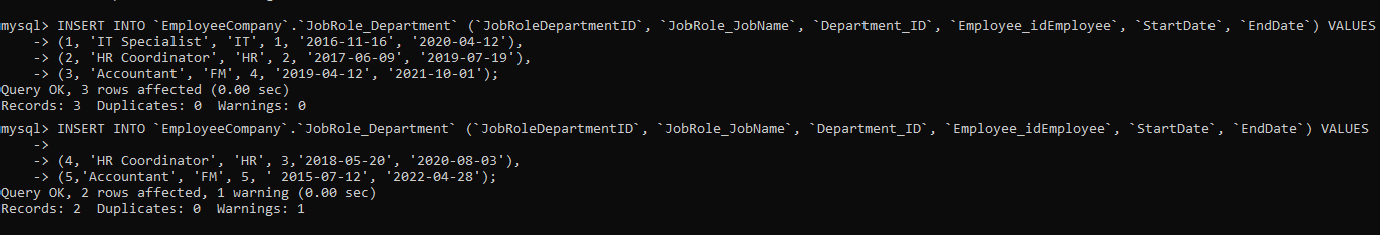
(1, 'IT Specialist', 'IT', 1, '2016-11-16', '2020-04-12'),

(2, 'HR Coordinator', 'HR', 2, '2017-06-09', '2019-07-19'),

(3, 'Accountant', 'FM', 4, '2019-04-12', '2021-10-01'),

(4, 'HR Coordinator', 'HR', 3,'2018-05-20', '2020-08-03'),

(5,'Accountant', 'FM', 5, ' 2015-07-12', '2022-04-28');



**TABLE NAME: Attandance**

INSERT INTO `EmployeeCompany`.`Attandance` (`AID`, `Date`, `ClockInTime`,

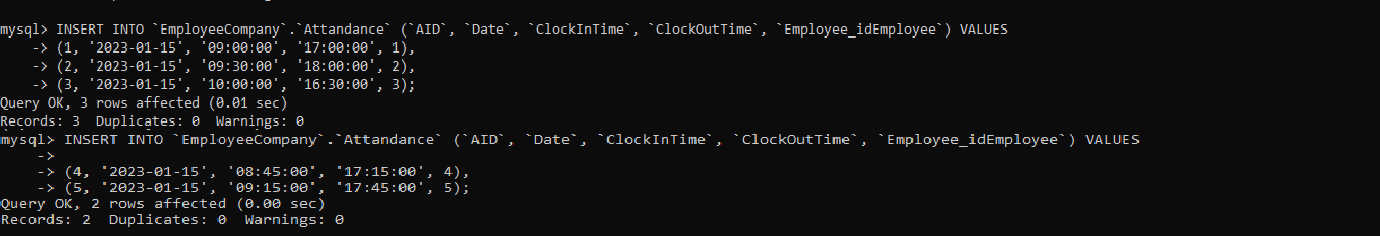
`ClockOutTime`, `Employee\_idEmployee`) VALUES (1, '2023-01-15', '09:00:00', '17:00:00', 1),

(2, '2023-01-15', '09:30:00', '18:00:00', 2),

(3, '2023-01-15', '10:00:00', '16:30:00', 3),

(4, '2023-01-15', '08:45:00', '17:15:00', 4),

(5, '2023-01-15', '09:15:00', '17:45:00', 5);



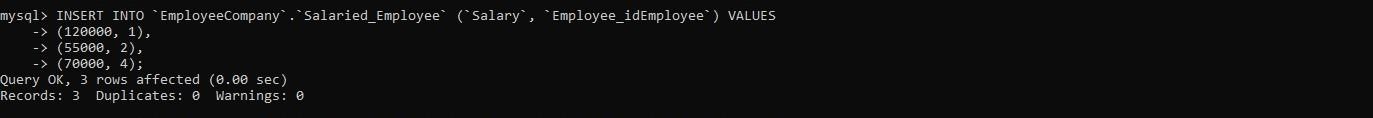
**TABLE NAME: Salaried\_employee**

INSERT INTO `EmployeeCompany`.`Salaried\_Employee` (`Salary`, `Employee\_idEmployee`) VALUES

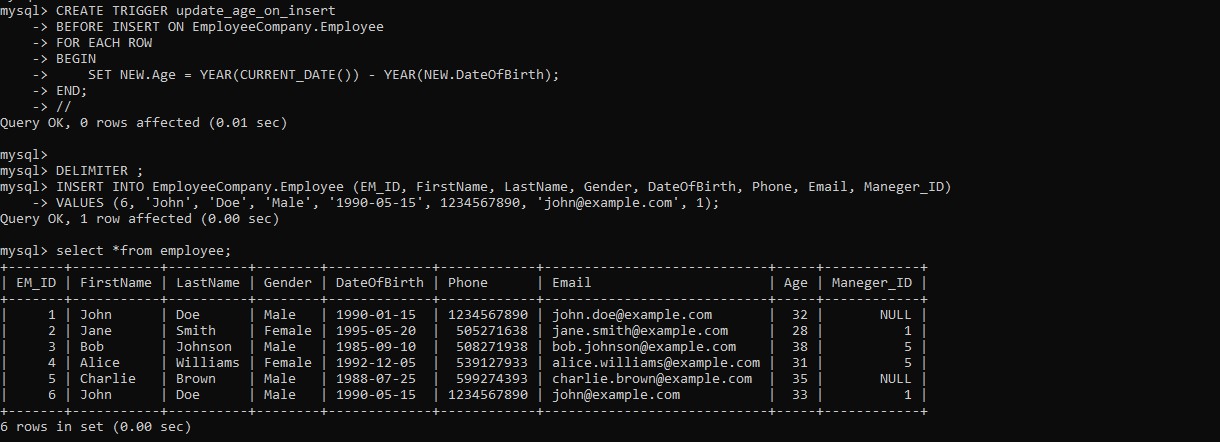
(120000, 1),

(55000, 2),

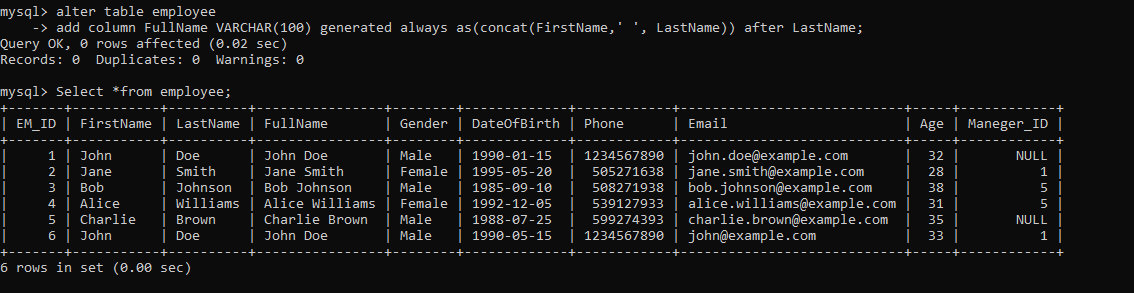
(70000, 4);



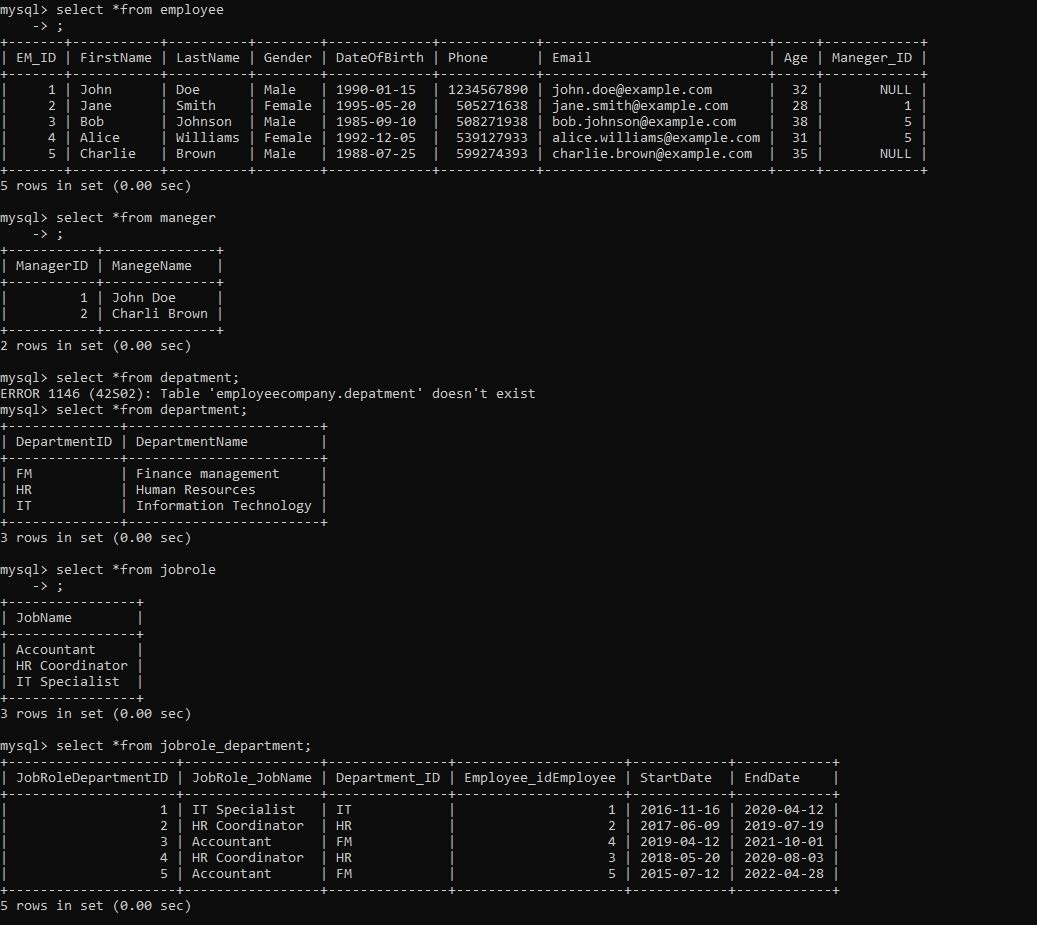
### Trigger function:

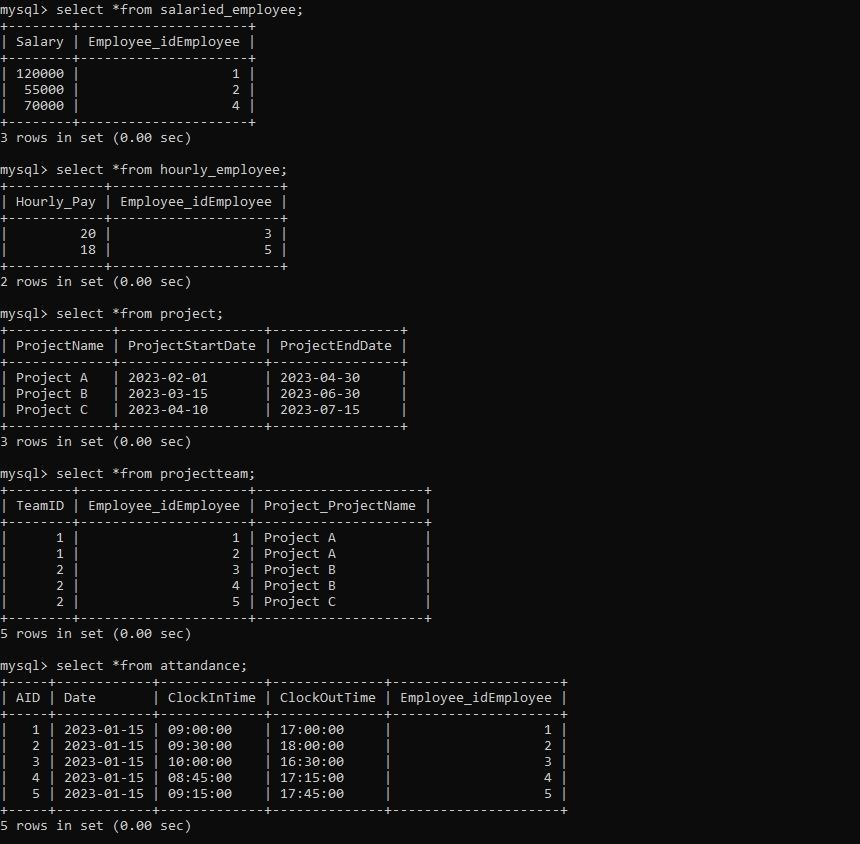


### virtually generated column:



**\*Here we are using select to show the data we inserted into our tables:**



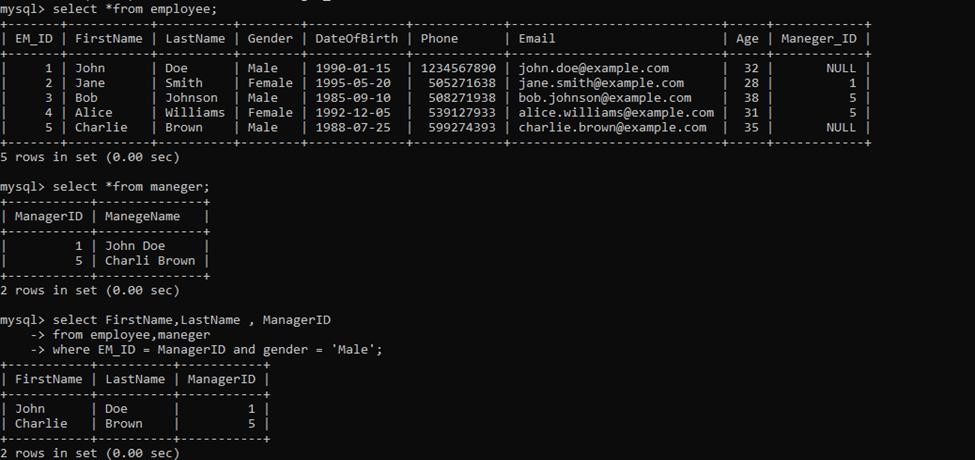


### Testing and Evaluation

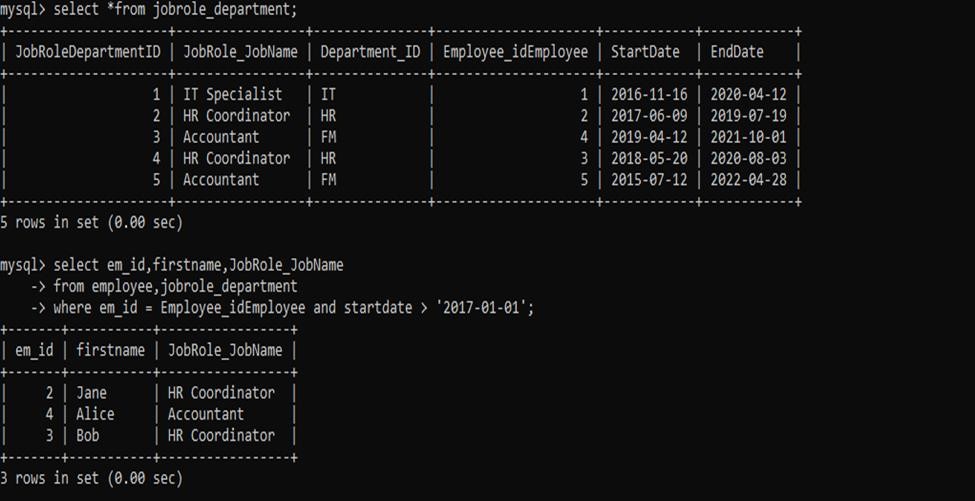


AND, OR:

* 1#AND:



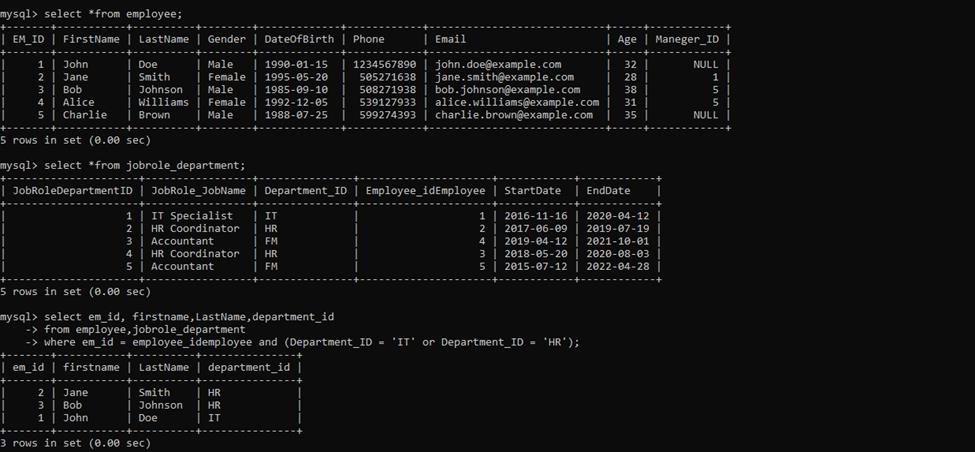
* 2#AND:



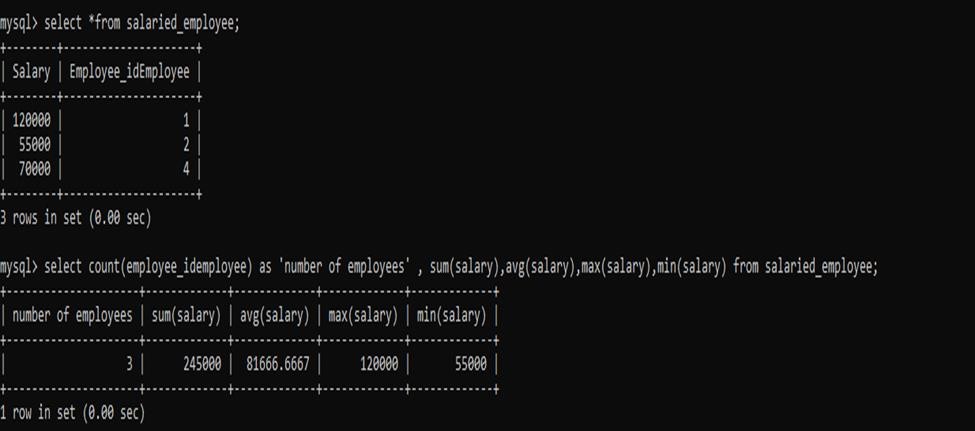
- 1#OR:



SUM, COUNT, AVG, MAX, MIN:

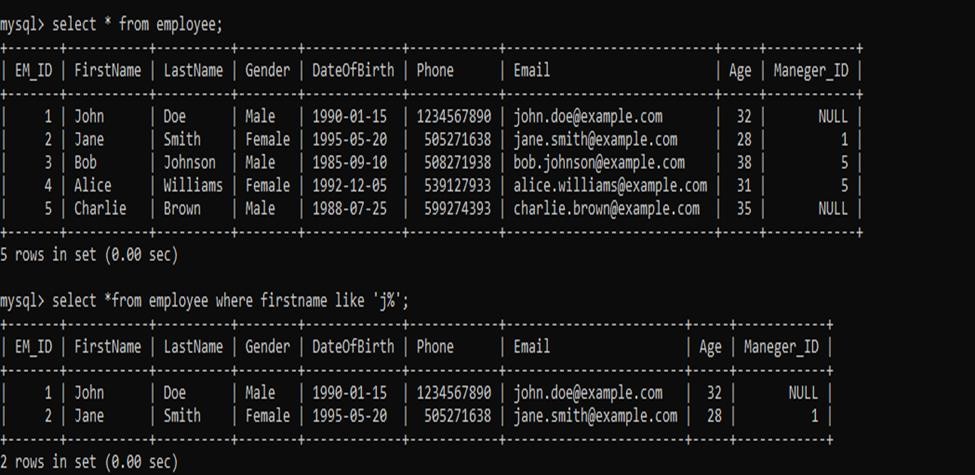


-

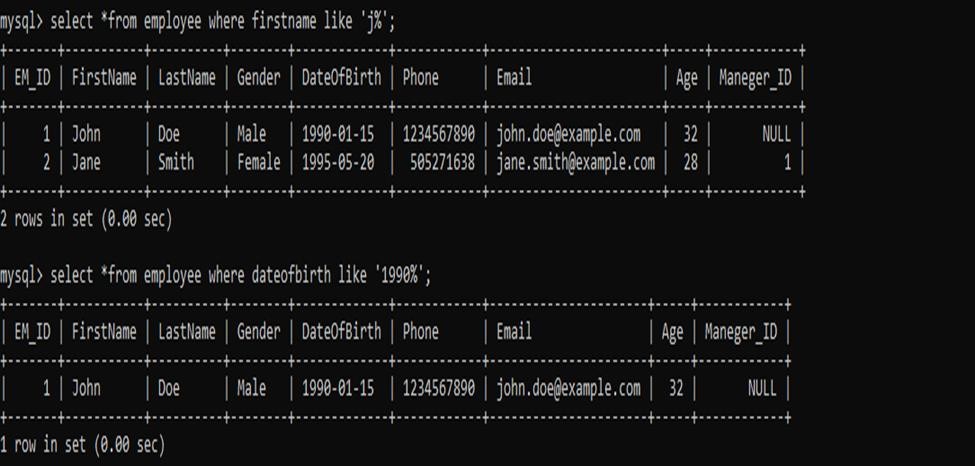


Like & IN:

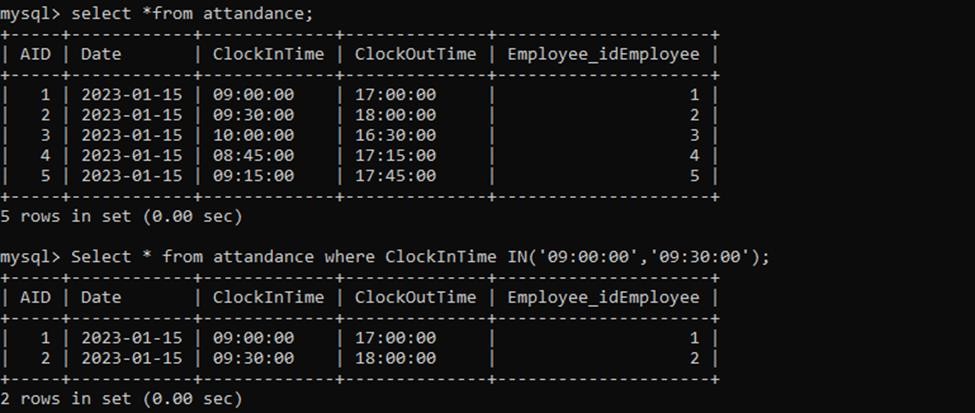
* 1# LIKE:



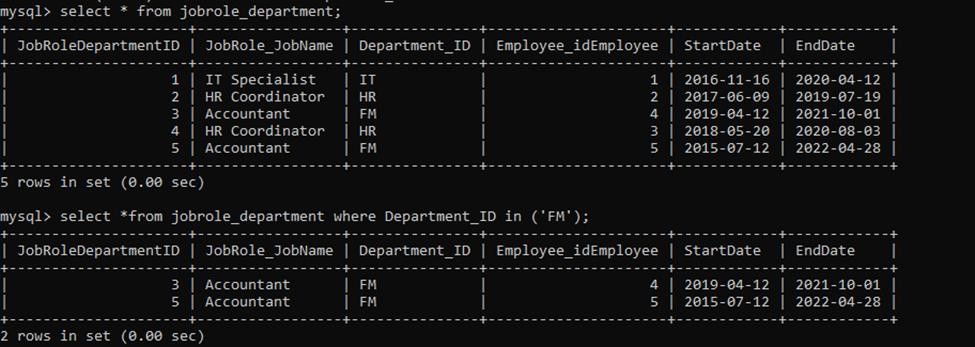
- 2#LIKE:



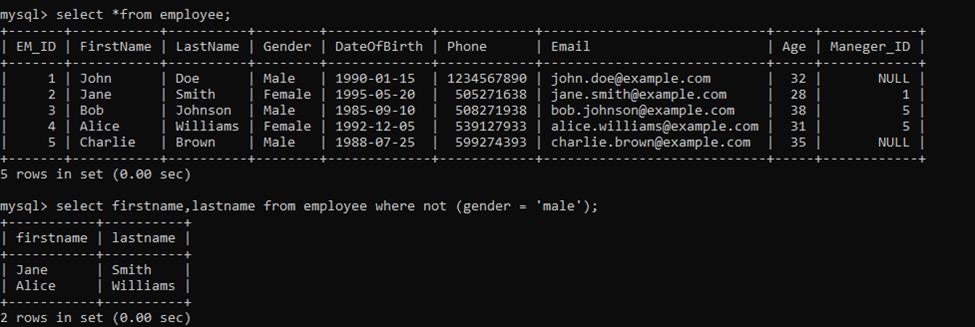
- 1#IN:



- 2#IN:



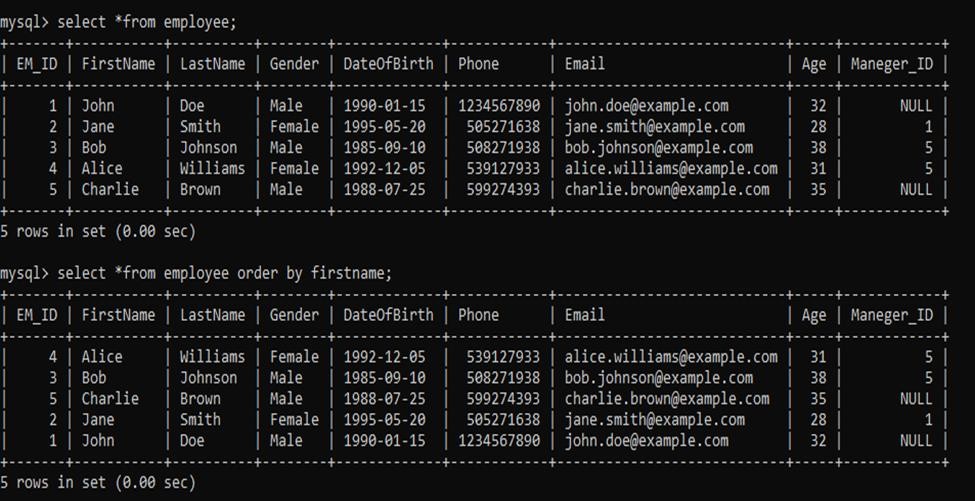
* NOT:



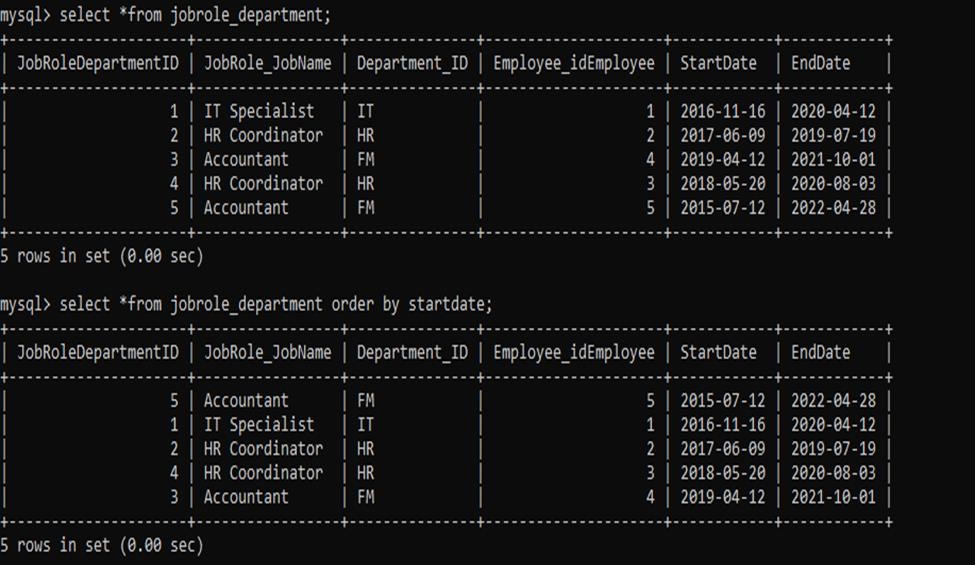


ORDER BY, GROUP BY, HAVING:

* 1#ORDER BY:



* 2#ORDER BY:



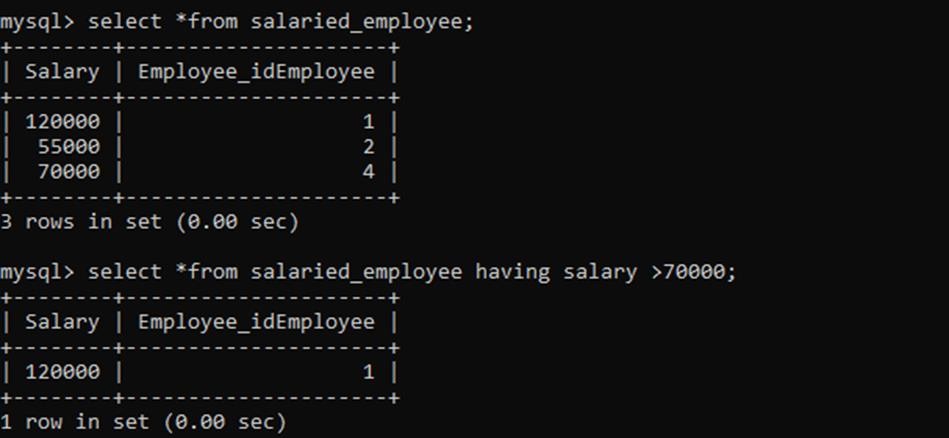
* 1#GROUP BY:



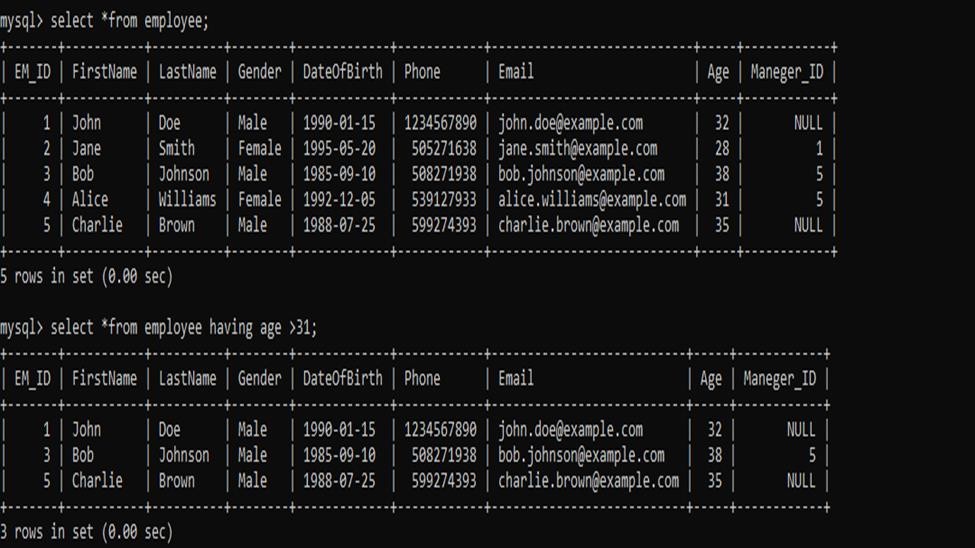
* 2#GROUP BY:



* 1#HAVING:



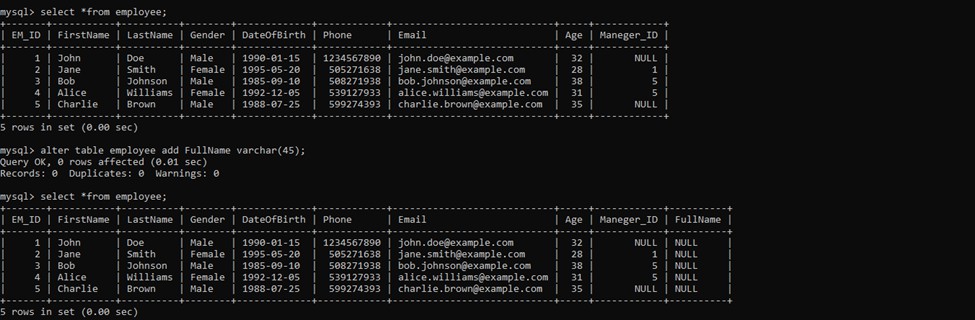
* 2#HAVING:



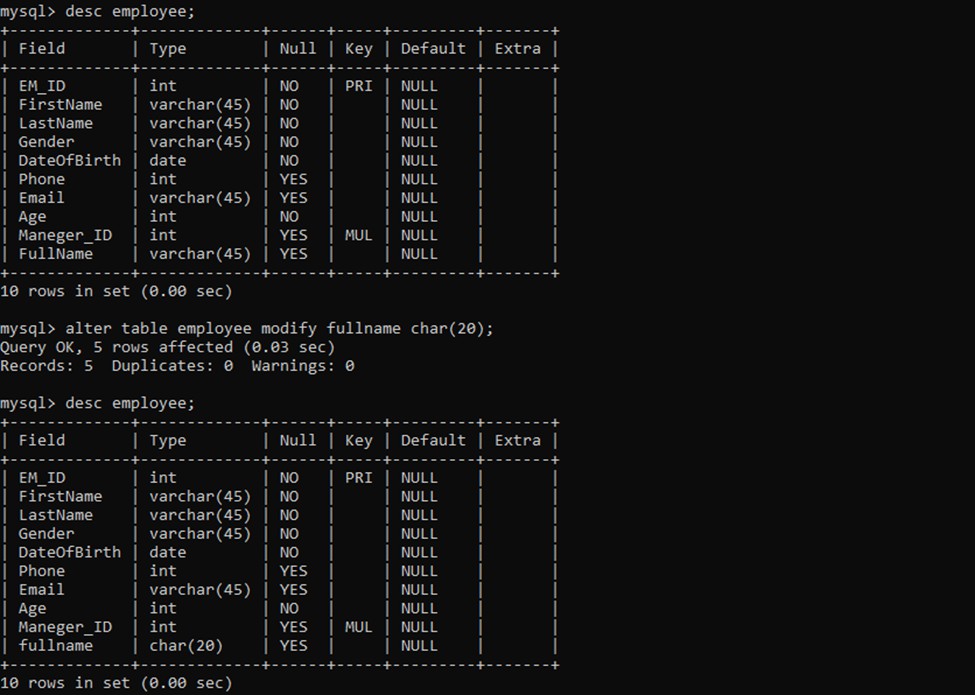


Drop Column, Add Column, Modify Column, Unique and Auto\_increment keywords:

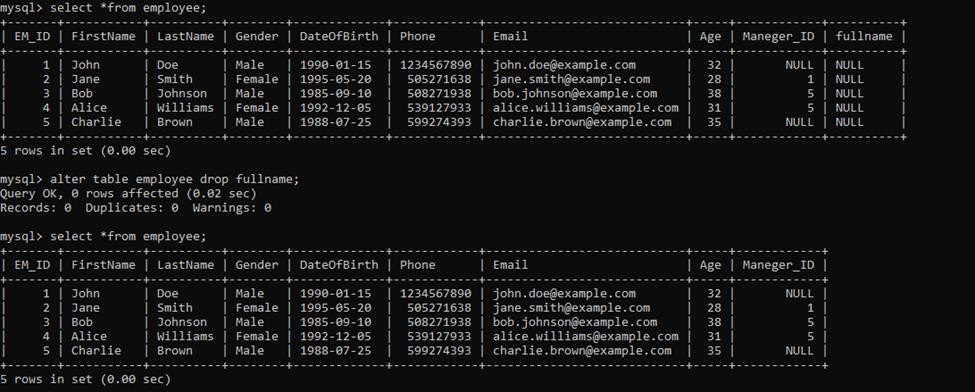
* Add:



* MODIFY:

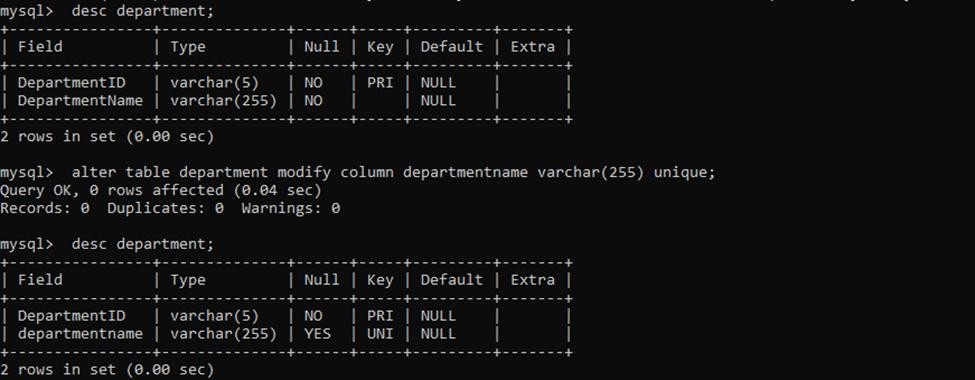






- Drop:

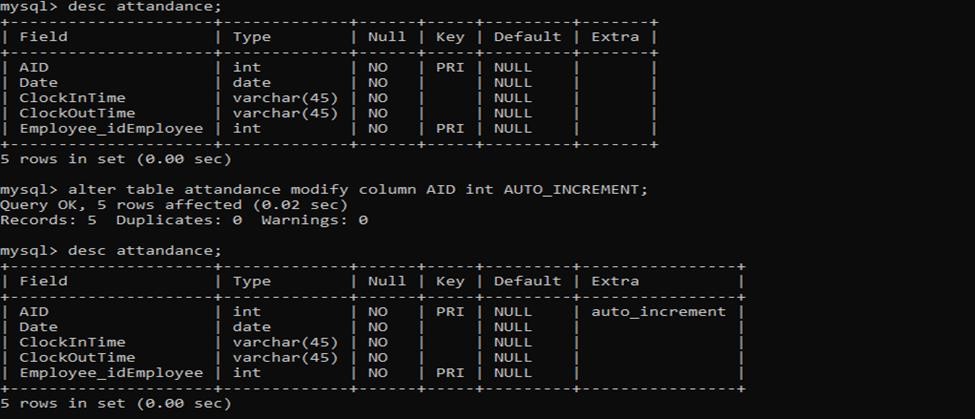
- Unique:



-



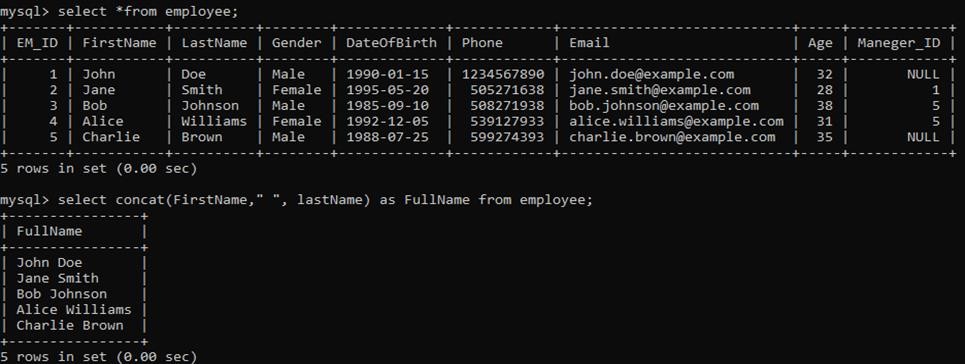
AUTO\_INCREMENT:



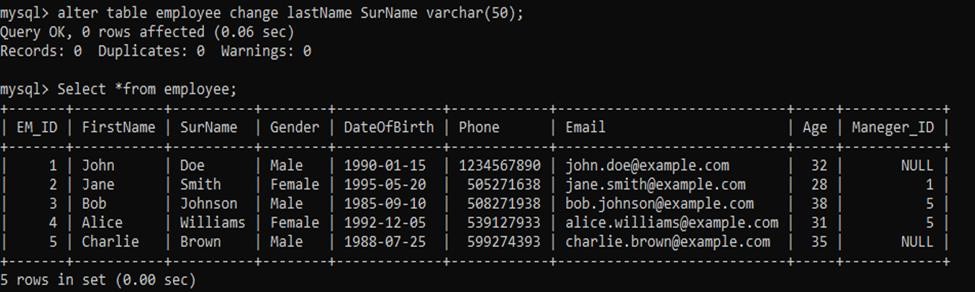


CONCAT, CHANGE, UPDATE and DELETE:

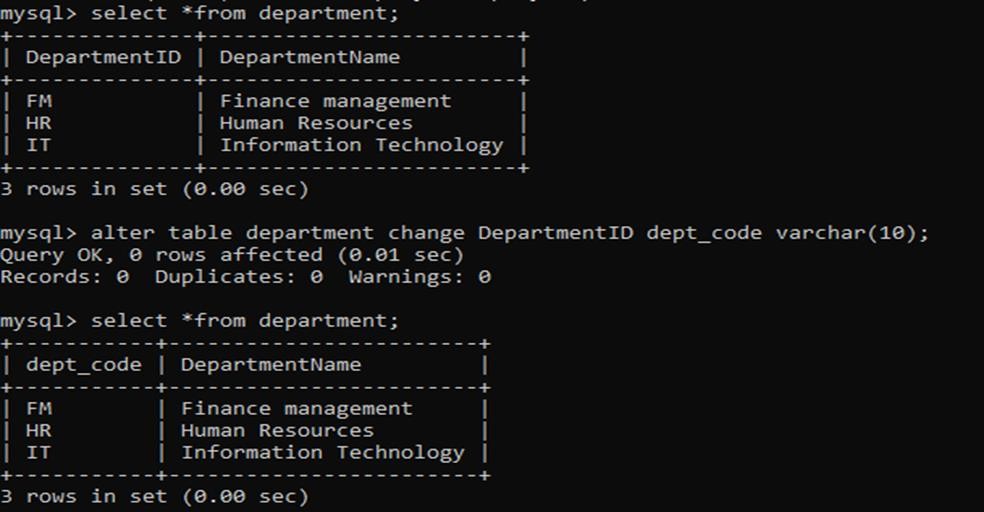
* CONCAT:



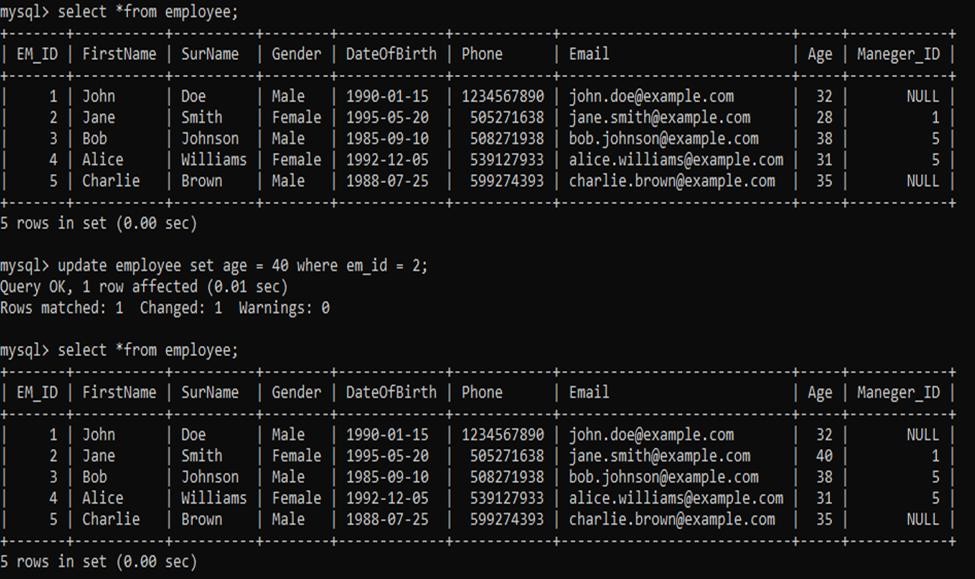
* 1#CHANGE:



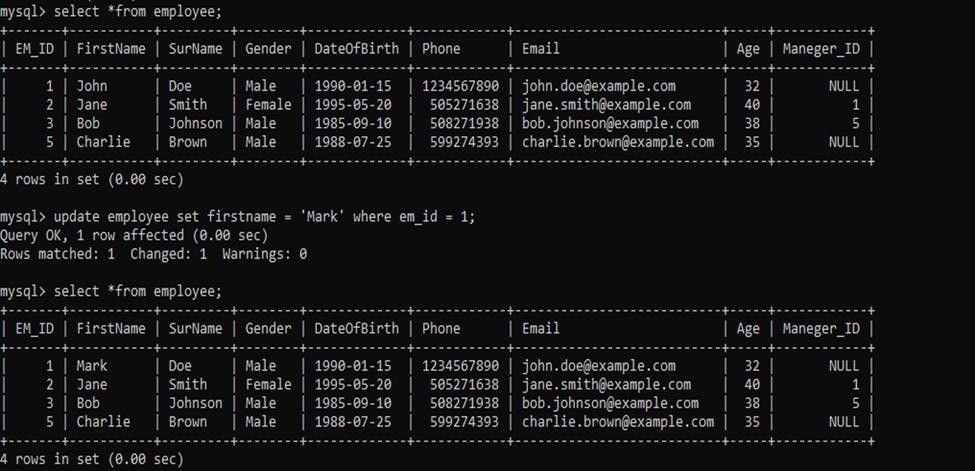
* 2#CHANGE:



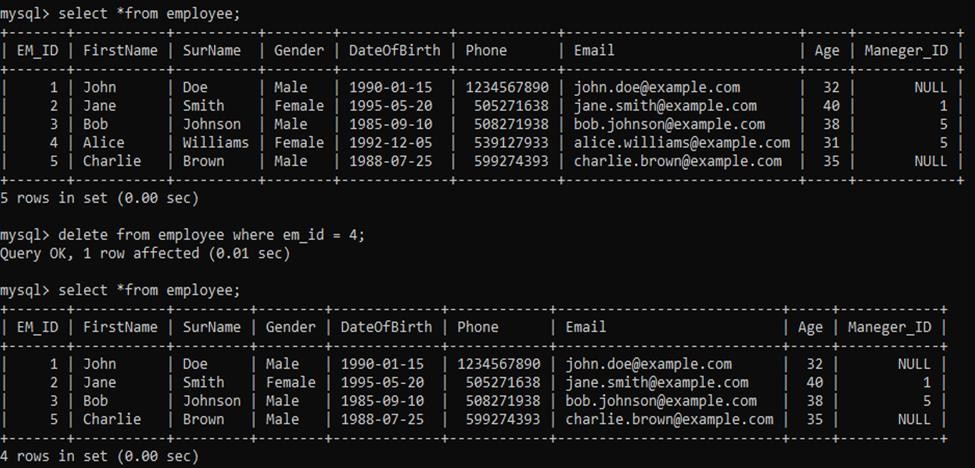
* 1#UPDATE:



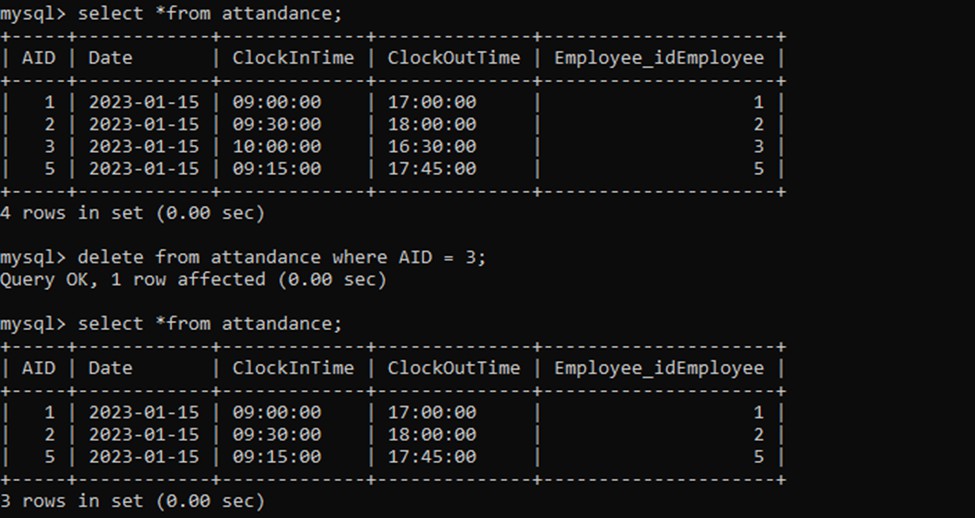
* 2#UPDATE:



* 1#DELETE:



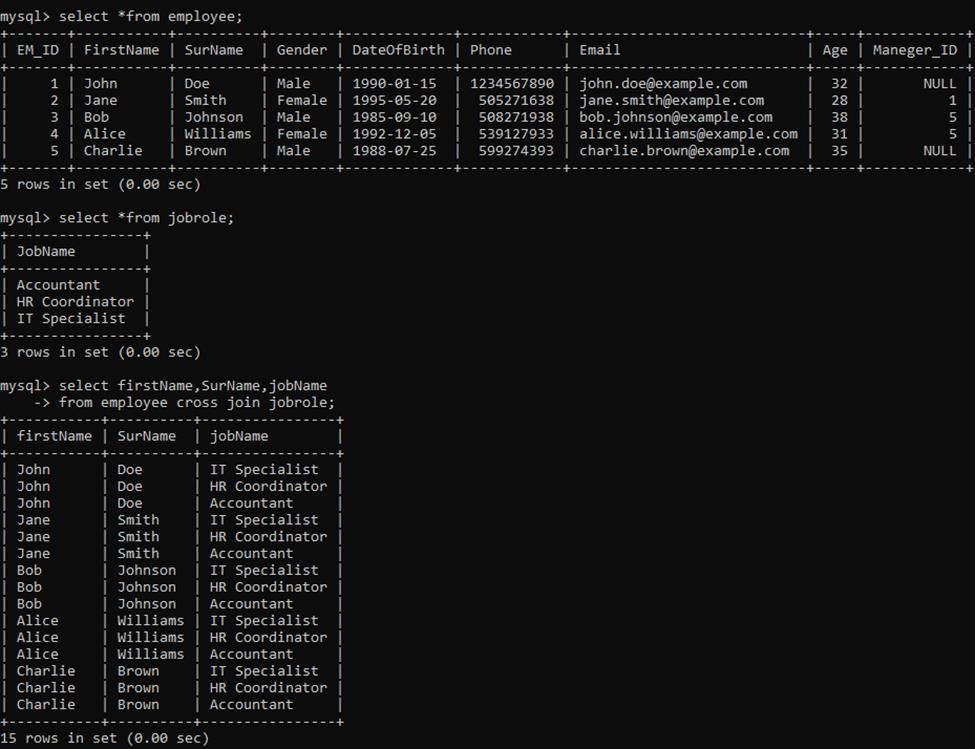
* 2#DELETE:



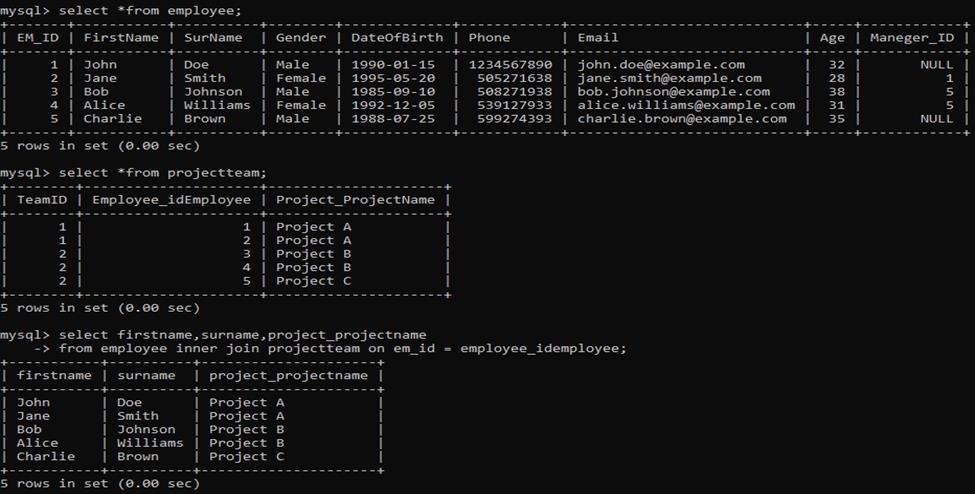


INNER, LEFT, RIGHT and CROSS:

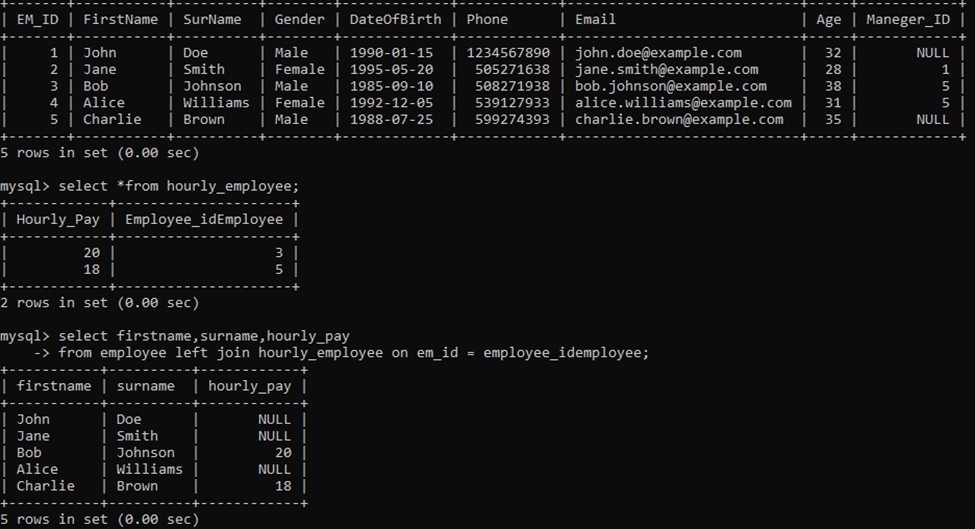
* CORSS JOIN:



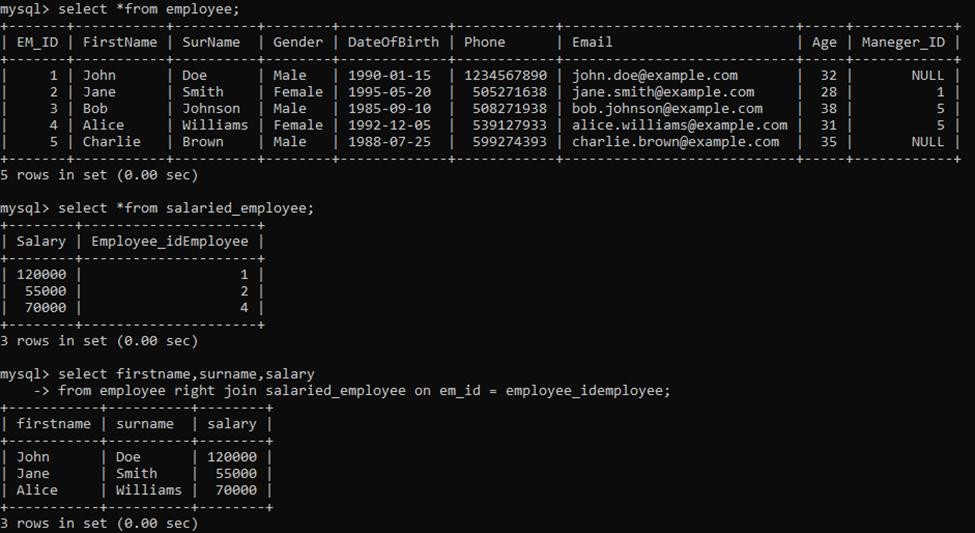
* INNER JOIN:



* LEFT JOIN:



* RIGHT JOIN:



#### Conclusion and Expected Benefits of the System

The "EmployeeCompany" database serves as a robust and organized system for managing crucial aspects of an organization's workforce. Its structured tables provide a comprehensive overview of employees, managers, departments, job roles, attendance, projects, and their interconnections. This database enhances the efficiency of workforce management, enabling the organization to make informed decisions and streamline various processes

#### Expected Benefits of the System

* Efficient Workforce Management: The system enables the efficient management of employee information, departmental structuring, and job role assignments, leading to smoother day-to-day operations.
* Accurate Attendance Tracking: The attendance module ensures precise recording of employee clock-in and clock-out times, facilitating payroll processing and compliance with labor regulations.
* Optimized Project Allocation: The ability to connect employees to projects through the "ProjectTeam" table ensures optimal utilization of skills and resources, improving project outcomes.
* Data-Driven Decision Making: With comprehensive data on employees, departments, and projects, the organization can make informed decisions regarding resource allocation, workforce planning, and project management.
* Enhanced Communication: The database facilitates improved communication by providing a centralized platform for accessing and sharing critical workforce information, fostering collaboration across the organization.

In summary, the "EmployeeCompany" database is a valuable tool for organizations seeking efficient and organized management of their workforce. Future enhancements can further elevate its capabilities, ensuring adaptability to evolving organizational needs and industry trends.

## REFERENCES

[1]. Donaldson, Molla S., and Kathleen N. Lohr. "Health databases and health database organizations: uses, benefits, and concerns." *Health Data in the information age: use, disclosure, and privacy*. National Academies Press (US), 1994.

[2]. Elmasri, Ramez, and Shamkant B. Navathe. Fundamentals of database systems. Addison-Wesley, 2011.

[3]. Chen, Peter Pin-Shan. "The entity-relationship model: a basis for the enterprise view of data." *Proceedings of the June 13-16, 1977, national computer conference*. 1977.

[4]. Chen, Peter Pin-Shan. "The entity-relationship model—toward a unified view of data." *ACM transactions on database systems (TODS)* 1.1 (1976): 9-36.

[5]. Alkoshman, Manal Mahmoud. "Unified modeling language and enhanced entity relationship: an empirical study." *Int. J. Database Theory Appl* 8.3 (2015): 215-227.

[6]. Do Nascimento Fidalgo, Robson, et al. "EERMM: a metamodel for the enhanced entity-relationship model." *Conceptual Modeling: 31st International Conference ER 2012, Florence, Italy, October 15-18, 2012. Proceedings 31*. Springer Berlin Heidelberg, 2012.

[7]. Ezema, Chukwuedozie N., Chinazam C. Ezemab, and Chukwuebuka B. Umezinwac. "Employee Performance Appraisal Database Management System." *development* 4 (2017): 7.

[8]. Bachman, Charles W. "Data structure diagrams." *ACM SIGMIS Database: The DATABASE for Advances in Information Systems* 1.2 (1969): 4-10.

[9]. Mohammed, A. M. "Design and implementation of employee management system." *Baze University, Abuja* (2020