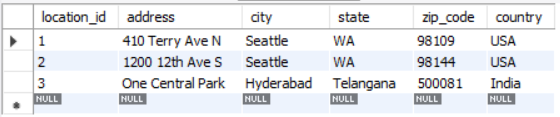
BDDA – 2 Project 1: **Database Management using MySQL & SQL**

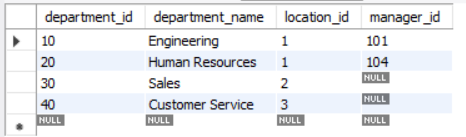
**Submitted by: Arushi Khanna, Domain: HR – Employee Database on Amazon**

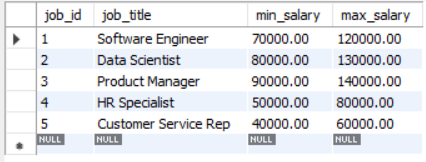
Database Images (before CRUD Operations and put Dummy Data)

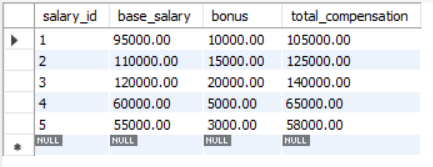
Location

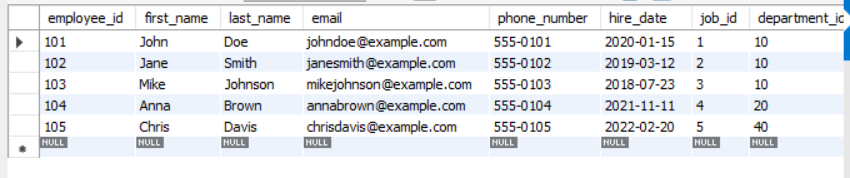


Department









Before CRUD Operations:

CREATE DATABASE amazon\_employee\_db;

USE amazon\_employee\_db;

-- Table: Locations

CREATE TABLE Locations (

location\_id INT PRIMARY KEY,

address VARCHAR(255),

city VARCHAR(100),

state VARCHAR(100),

zip\_code VARCHAR(20),

country VARCHAR(100)

);

-- Table: Departments

CREATE TABLE Departments (

department\_id INT PRIMARY KEY,

department\_name VARCHAR(100),

location\_id INT,

manager\_id INT,

FOREIGN KEY (location\_id) REFERENCES Locations(location\_id),

FOREIGN KEY (manager\_id) REFERENCES Employees(employee\_id) -- Self-reference for manager

);

-- Table: Jobs

CREATE TABLE Jobs (

job\_id INT PRIMARY KEY,

job\_title VARCHAR(100),

min\_salary DECIMAL(10, 2),

max\_salary DECIMAL(10, 2)

);

-- Table: Salaries

CREATE TABLE Salaries (

salary\_id INT PRIMARY KEY,

base\_salary DECIMAL(10, 2),

bonus DECIMAL(10, 2),

total\_compensation DECIMAL(10, 2) AS (base\_salary + bonus) STORED

);

-- Table: Employees

CREATE TABLE Employees (

employee\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100) UNIQUE,

phone\_number VARCHAR(20),

hire\_date DATE,

job\_id INT,

department\_id INT,

manager\_id INT,

salary\_id INT,

FOREIGN KEY (job\_id) REFERENCES Jobs(job\_id),

FOREIGN KEY (department\_id) REFERENCES Departments(department\_id),

FOREIGN KEY (manager\_id) REFERENCES Employees(employee\_id), -- Self-reference for manager

FOREIGN KEY (salary\_id) REFERENCES Salaries(salary\_id)

);

DROP DATABASE amazon\_employee\_db;

CREATE DATABASE amazon\_employee\_db;

USE amazon\_employee\_db;

-- Table: Locations

CREATE TABLE Locations (

location\_id INT PRIMARY KEY,

address VARCHAR(255),

city VARCHAR(100),

state VARCHAR(100),

zip\_code VARCHAR(20),

country VARCHAR(100)

);

-- Table: Departments (initial creation without manager\_id FK)

CREATE TABLE Departments (

department\_id INT PRIMARY KEY,

department\_name VARCHAR(100),

location\_id INT,

FOREIGN KEY (location\_id) REFERENCES Locations(location\_id)

);

-- Table: Jobs

CREATE TABLE Jobs (

job\_id INT PRIMARY KEY,

job\_title VARCHAR(100),

min\_salary DECIMAL(10, 2),

max\_salary DECIMAL(10, 2)

);

-- Table: Salaries

CREATE TABLE Salaries (

salary\_id INT PRIMARY KEY,

base\_salary DECIMAL(10, 2),

bonus DECIMAL(10, 2),

total\_compensation DECIMAL(10, 2) AS (base\_salary + bonus) STORED

);

-- Table: Employees (initial creation without foreign key constraints)

CREATE TABLE Employees (

employee\_id INT PRIMARY KEY,

first\_name VARCHAR(50),

last\_name VARCHAR(50),

email VARCHAR(100) UNIQUE,

phone\_number VARCHAR(20),

hire\_date DATE,

job\_id INT,

department\_id INT,

manager\_id INT,

salary\_id INT

);

-- Add foreign key constraints for Employees after all tables have been created

ALTER TABLE Employees

ADD FOREIGN KEY (job\_id) REFERENCES Jobs(job\_id),

ADD FOREIGN KEY (department\_id) REFERENCES Departments(department\_id),

ADD FOREIGN KEY (manager\_id) REFERENCES Employees(employee\_id), -- Self-reference for manager

ADD FOREIGN KEY (salary\_id) REFERENCES Salaries(salary\_id);

-- Add the manager\_id foreign key to Departments after Employees table is created

ALTER TABLE Departments

ADD manager\_id INT,

ADD FOREIGN KEY (manager\_id) REFERENCES Employees(employee\_id);

-- Insert dummy data into Locations

INSERT INTO Locations (location\_id, address, city, state, zip\_code, country)

VALUES

(1, '410 Terry Ave N', 'Seattle', 'WA', '98109', 'USA'),

(2, '1200 12th Ave S', 'Seattle', 'WA', '98144', 'USA'),

(3, 'One Central Park', 'Hyderabad', 'Telangana', '500081', 'India');

-- Insert dummy data into Departments

INSERT INTO Departments (department\_id, department\_name, location\_id)

VALUES

(10, 'Engineering', 1),

(20, 'Human Resources', 1),

(30, 'Sales', 2),

(40, 'Customer Service', 3);

-- Insert dummy data into Jobs

INSERT INTO Jobs (job\_id, job\_title, min\_salary, max\_salary)

VALUES

(1, 'Software Engineer', 70000, 120000),

(2, 'Data Scientist', 80000, 130000),

(3, 'Product Manager', 90000, 140000),

(4, 'HR Specialist', 50000, 80000),

(5, 'Customer Service Rep', 40000, 60000);

-- Insert dummy data into Salaries

INSERT INTO Salaries (salary\_id, base\_salary, bonus)

VALUES

(1, 95000, 10000),

(2, 110000, 15000),

(3, 120000, 20000),

(4, 60000, 5000),

(5, 55000, 3000);

-- Insert dummy data into Employees

INSERT INTO Employees (employee\_id, first\_name, last\_name, email, phone\_number, hire\_date, job\_id, department\_id, manager\_id, salary\_id)

VALUES

(101, 'John', 'Doe', 'johndoe@example.com', '555-0101', '2020-01-15', 1, 10, NULL, 1),

(102, 'Jane', 'Smith', 'janesmith@example.com', '555-0102', '2019-03-12', 2, 10, 101, 2),

(103, 'Mike', 'Johnson', 'mikejohnson@example.com', '555-0103', '2018-07-23', 3, 10, 101, 3),

(104, 'Anna', 'Brown', 'annabrown@example.com', '555-0104', '2021-11-11', 4, 20, NULL, 4),

(105, 'Chris', 'Davis', 'chrisdavis@example.com', '555-0105', '2022-02-20', 5, 40, NULL, 5);

-- Update Departments to assign manager\_id for each department where applicable

UPDATE Departments SET manager\_id = 101 WHERE department\_id = 10;

UPDATE Departments SET manager\_id = 104 WHERE department\_id = 20;

SELECT \* FROM Locations;

SELECT \* FROM Departments;

SELECT \* FROM Jobs;

SELECT \* FROM Salaries;

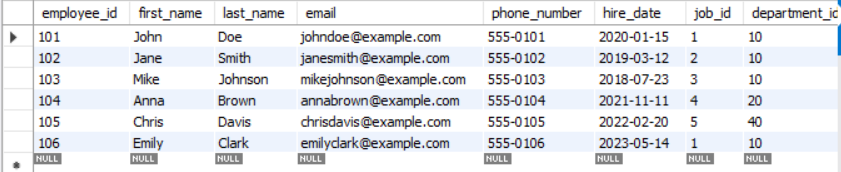
SELECT \* FROM Employees;

After CRUD Operations:

CREATE

INSERT INTO Employees (employee\_id, first\_name, last\_name, email, phone\_number, hire\_date, job\_id, department\_id, manager\_id, salary\_id)

VALUES (106, 'Emily', 'Clark', 'emilyclark@example.com', '555-0106', '2023-05-14', 1, 10, 101, 1);



READ

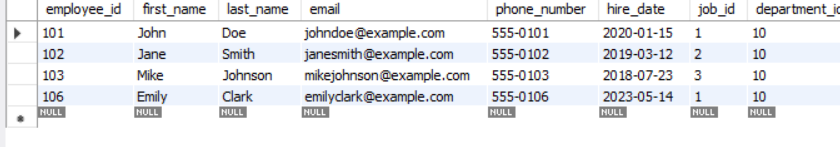
SELECT \* FROM Employees;

-- Select specific columns

SELECT first\_name, last\_name, email FROM Employees;

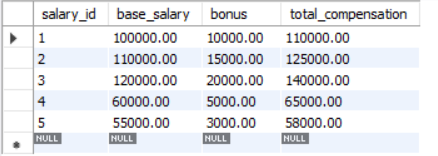
-- Select employees with specific criteria (e.g., from Engineering department)

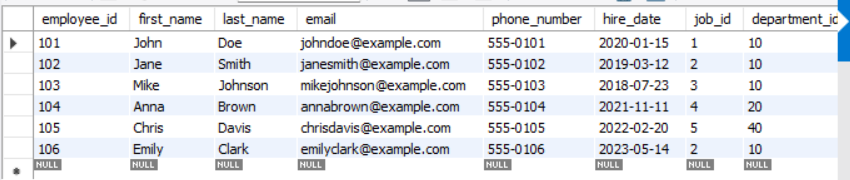
SELECT \* FROM Employees WHERE department\_id = 10;

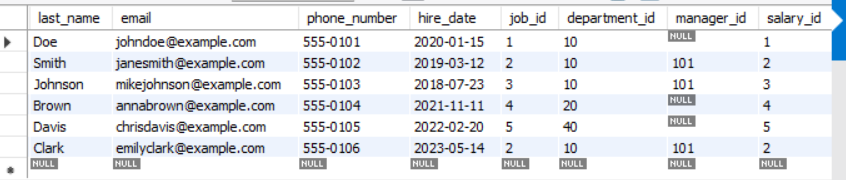


UPDATE

UPDATE Employees SET job\_id = 2, salary\_id = 2 WHERE employee\_id = 106; -- Increase the base salary in Salaries table for a specific employee's salary entry UPDATE Salaries SET base\_salary = base\_salary + 5000 WHERE salary\_id = 1;







DELETE

DELETE FROM Employees WHERE employee\_id = 106; -- Delete all employees from a specific department (e.g., department\_id = 20) DELETE FROM Employees WHERE department\_id = 20;

