

# Day 32 Practice Problem - Included ER Diagram

1 . Employee Payroll Problem

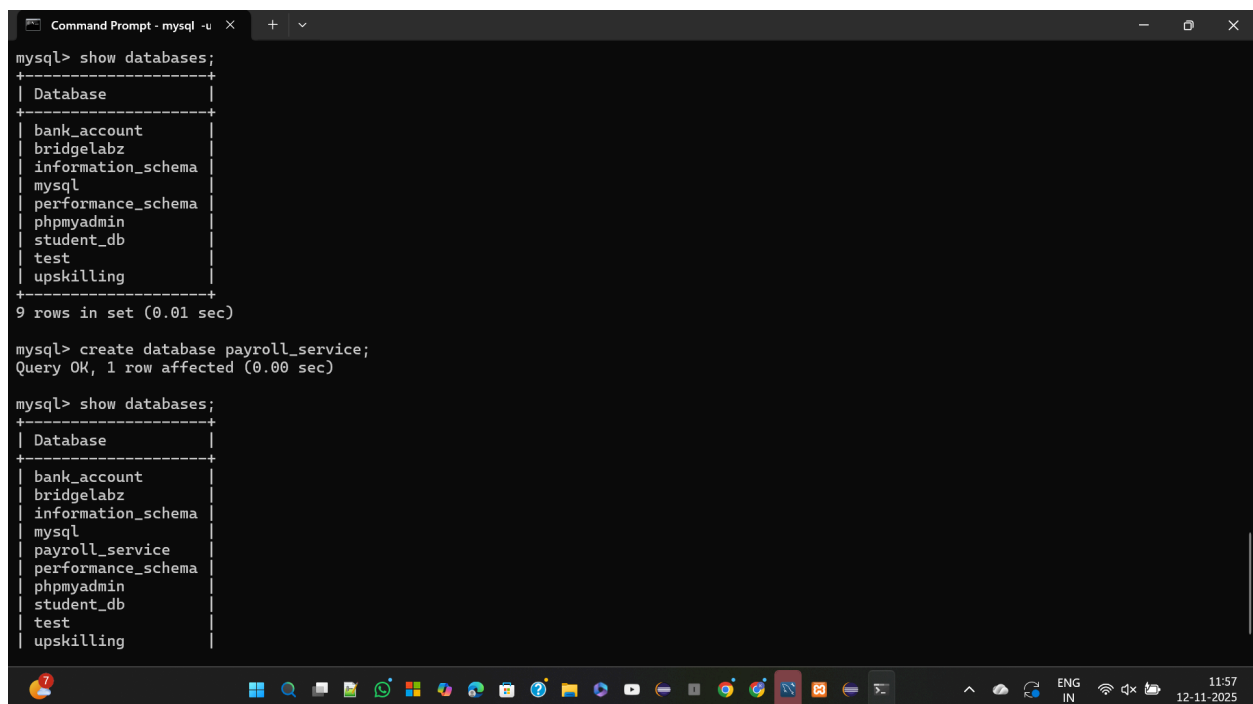
2 . AddressBook Problem

## 1 . EMPLOYEE PAYROLL PROBLEM :

### Use Case 1 :

#### Ability to create a payroll service database :

- Using MySQL Client use **create database** query to create a payroll\_service database
- Also you can see the DB created by using **show database** query
- And go to the database created by using **use payroll\_service** query



```
mysql> show databases;
+-----+
| Database |
+-----+
| bank_account |
| bridgelabz |
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| student_db |
| test |
| upskilling |
+-----+
9 rows in set (0.01 sec)

mysql> create database payroll_service;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| bank_account |
| bridgelabz |
| information_schema |
| mysql |
| payroll_service |
| performance_schema |
| phpmyadmin |
| student_db |
| test |
| upskilling |
+-----+
```

## Use Case 2 :

**Ability to create a employee payroll table in the payroll service database to manage employee payrolls :**

- Use payroll\_service database in MySQL Client
- Use **Create Table employee\_payroll** Query to create employee payroll table with columns id, name, salary and start date as column. Note Id is set to auto increment.
- Understand the SQL data types to be used for the table
- Note: SQL Queries as case insensitive

```
information_schema |
mysql              |
payroll_service    |
performance_schema |
phpmyadmin         |
student_db         |
test               |
upskilling         |
+-----+
10 rows in set (0.00 sec)

mysql> use payroll_service;
Database changed
mysql> create table employee_payroll(id INT NOT NULL AUTO_INCREMENT, name VARCHAR(50) NOT NULL, salary DECIMAL(10,2) NOT NULL, start
DATE NOT NULL, PRIMARY KEY (id));
Query OK, 0 rows affected (0.02 sec)
```

## Use Case 3 :

**Ability to create employee payroll data in the payroll service database as part of CRUD Operation :**

- Use payroll\_service database in MySQL Client
- Use **INSERT INTO employee\_payroll** Query to create employees payroll data into the employee\_payroll table

```
mysql> use payroll_service;
Database changed
mysql> create table employee_payroll(id INT NOT NULL AUTO_INCREMENT, name VARCHAR(50) NOT NULL, salary DECIMAL(10,2) NOT NULL, start
DATE NOT NULL, PRIMARY KEY (id));
Query OK, 0 rows affected (0.02 sec)

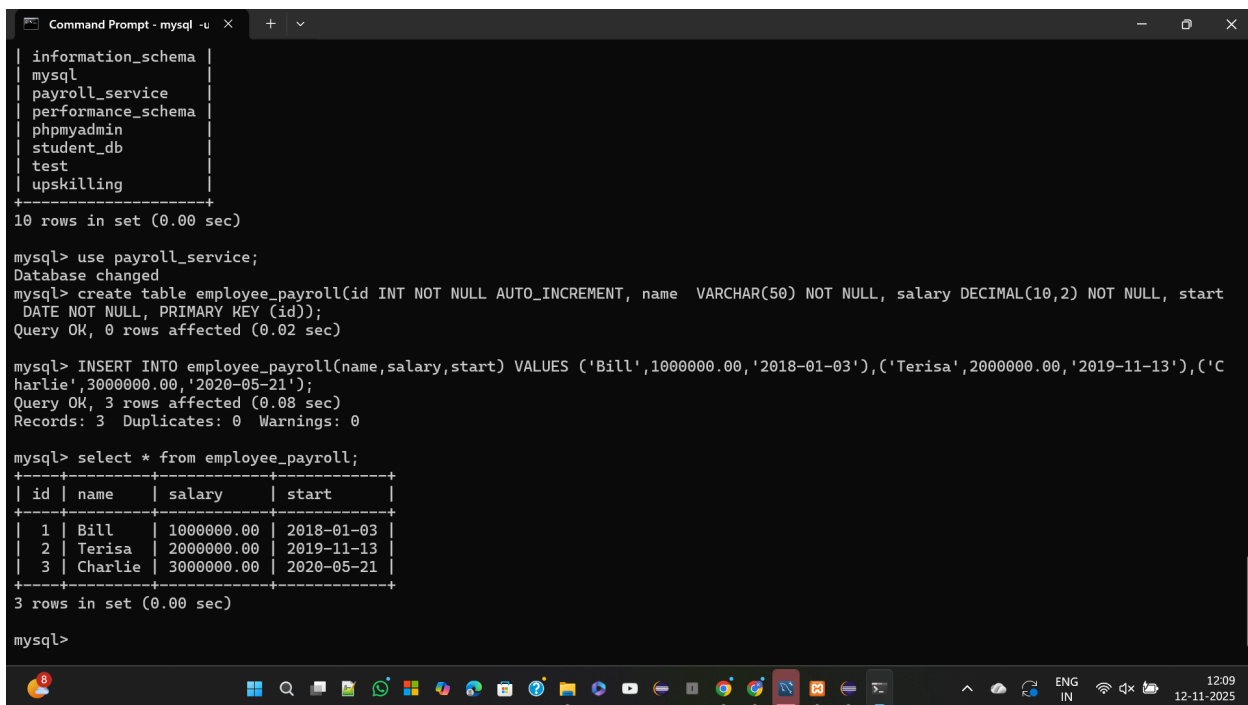
mysql> INSERT INTO employee_payroll(name,salary,start) VALUES ('Bill',1000000.00,'2018-01-03'),('Terisa',2000000.00,'2019-11-13'),('C
harlie',3000000.00,'2020-05-21');
Query OK, 3 rows affected (0.08 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> |
```

## Use Case 4 :

Ability to retrieve all the employee payroll data that is added to payroll service Database :

- Use payroll\_service database in MySQL Client
- Use **SELECT \* FROM employee\_payroll** to retrieve all the data from the employee\_payroll table



```
Command Prompt - mysql -u <user>
+-----+
| information_schema |
| mysql              |
| payroll_service    |
| performance_schema |
| phpmyadmin         |
| student_db        |
| test               |
| upskilling         |
+-----+
10 rows in set (0.00 sec)

mysql> use payroll_service;
Database changed
mysql> create table employee_payroll(id INT NOT NULL AUTO_INCREMENT, name VARCHAR(50) NOT NULL, salary DECIMAL(10,2) NOT NULL, start
DATE NOT NULL, PRIMARY KEY (id));
Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO employee_payroll(name,salary,start) VALUES ('Bill',1000000.00,'2018-01-03'),('Terisa',2000000.00,'2019-11-13'),('C
harlie',3000000.00,'2020-05-21');
Query OK, 3 rows affected (0.08 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> select * from employee_payroll;
+----+-----+-----+-----+
| id | name  | salary | start |
+----+-----+-----+-----+
| 1  | Bill  | 1000000.00 | 2018-01-03 |
| 2  | Terisa | 2000000.00 | 2019-11-13 |
| 3  | Charlie | 3000000.00 | 2020-05-21 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

## Use Case 5 :

Ability to retrieve salary data for a particular employee as well as all employees who have joined in a particular data range from the payroll service database :

- Use **SELECT salary FROM employee\_payroll**  
**WHERE name = 'Bill'** Query to View Bill's salary
- Use **Select** query with **Where** condition View employees between start dates
- Query: **WHERE start BETWEEN CAST('2018-01-01' AS DATE) AND DATE(NOW());**
- Note: Where Condition Clause is used to retrieve the row needed from the table
- Note: Use of Database Functions like **CAST()** and **NOW()** in the Query

```
Command Prompt - mysql -u x + v
harlie',3000000.00,'2020-05-21');
Query OK, 3 rows affected (0.08 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> select * from employee_payroll;
+----+-----+-----+-----+
| id | name  | salary | start |
+----+-----+-----+-----+
| 1  | Bill  | 1000000.00 | 2018-01-03 |
| 2  | Terisa | 2000000.00 | 2019-11-13 |
| 3  | Charlie | 3000000.00 | 2020-05-21 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> select salary from employee_payroll where name = 'Bill';
+-----+
| salary |
+-----+
| 1000000.00 |
+-----+
1 row in set (0.00 sec)

mysql> select * from employee_payroll where start BETWEEN CAST('2018-01-01' AS DATE) and DATE(NOW());
+----+-----+-----+-----+
| id | name  | salary | start |
+----+-----+-----+-----+
| 1  | Bill  | 1000000.00 | 2018-01-03 |
| 2  | Terisa | 2000000.00 | 2019-11-13 |
| 3  | Charlie | 3000000.00 | 2020-05-21 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

## Use Case 6 :

**Ability to add Gender to Employee Payroll Table and Update the Rows to reflect the correct Employee Gender :**

- Use payroll\_service database in MySQL Client
- Use Alter Table Command to add Field gender after the name field
- Use Update Query to set the gender using where condition with the employee name
- E.g. **UPDATE employee\_payroll set gender = 'M' where name = 'Bill' or name = 'Charlie';**

```
mysql> select * from employee_payroll where start BETWEEN CAST('2018-01-01' AS DATE) and DATE(NOW());
+----+-----+-----+-----+
| id | name  | salary | start |
+----+-----+-----+-----+
| 1  | Bill  | 1000000.00 | 2018-01-03 |
| 2  | Terisa | 2000000.00 | 2019-11-13 |
| 3  | Charlie | 3000000.00 | 2020-05-21 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> Alter table employee_payroll add gender CHAR(1) after name;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> update employee_payroll set gender = 'M' where name = 'Bill' or name = 'Charlie';
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2 Changed: 2 Warnings: 0

mysql> update employee_payroll set gender = 'F' where name = 'Terisa';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from employee_payroll;
+----+-----+-----+-----+
| id | name  | gender | salary | start |
+----+-----+-----+-----+
| 1  | Bill  | M      | 1000000.00 | 2018-01-03 |
| 2  | Terisa | F      | 2000000.00 | 2019-11-13 |
| 3  | Charlie | M      | 3000000.00 | 2020-05-21 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

## Use Case 7 :

**Ability to find sum, average, min, max and number of male and female employees :**

- Use payroll\_service database in MySQL Client
- Use Database Function **SUM**, **AVG**, **MIN**, **MAX**, **COUNT** to do analysis by Male or Female.
- Note: You will need to use GROUP BY GENDER grouping to get the result
- E.g. **SELECT SUM(salary) FROM employee\_payroll WHERE gender = 'F' GROUP BY gender;**

```
mysql> select gender, SUM(salary) from employee_payroll group by gender;
+-----+
| gender | SUM(salary) |
+-----+
| F      | 2000000.00  |
| M      | 4000000.00  |
+-----+
2 rows in set (0.00 sec)

mysql> select gender, AVG(salary) from employee_payroll group by gender;
+-----+
| gender | AVG(salary) |
+-----+
| F      | 2000000.000000 |
| M      | 2000000.000000 |
+-----+
2 rows in set (0.00 sec)

mysql> select gender, MIN(salary) from employee_payroll group by gender;
+-----+
| gender | MIN(salary) |
+-----+
| F      | 2000000.00  |
| M      | 1000000.00  |
+-----+
2 rows in set (0.00 sec)

mysql> select gender, MAX(salary) from employee_payroll group by gender;
+-----+
| gender | MAX(salary) |
+-----+
| F      | 2000000.00  |
+-----+
```

```
mysql> select gender, MAX(salary) from employee_payroll group by gender;
+-----+
| gender | MAX(salary) |
+-----+
| F      | 2000000.00  |
| M      | 3000000.00  |
+-----+
2 rows in set (0.00 sec)

mysql> select gender, COUNT(*) AS count from employee_payroll group by gender;
+-----+
| gender | count |
+-----+
| F      | 1     |
| M      | 2     |
+-----+
2 rows in set (0.00 sec)

mysql>
```

## Use Case 8 :

Ability to extend employee\_payroll data to store employee information like employee phone, address and department - Ensure employee department is non nullable fields.

- Add Default Value for address field.

```
Command Prompt - mysql -u <user>
+-----+
| bridgelabz |
| information_schema |
| mysql |
| payroll_service |
| performance_schema |
| phpmyadmin |
| student_db |
| test |
| upskilling |
+-----+
11 rows in set (0.00 sec)

mysql> use payroll_service
Database changed
mysql> alter table employee_payroll add phone VARCHAR(20);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table employee_payroll add address VARCHAR(150) DEFAULT 'Bangalore';
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table employee_payroll add department VARCHAR(50) NOT NULL;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> select * from employee_payroll;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | name | gender | salary | start | phone | address | department |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Bill | M | 1000000.00 | 2018-01-03 | NULL | Bangalore |  |
| 2 | Terisa | F | 2000000.00 | 2019-11-13 | NULL | Bangalore |  |
| 3 | Charlie | M | 3000000.00 | 2020-05-21 | NULL | Bangalore |  |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

## Use Case 9 :

Ability to extend employee\_payroll table to have Basic Pay, Deductions, Taxable Pay, Income Tax, Net Pay

```
Command Prompt - mysql -u x
mysql> select * from employee_payroll;
+----+-----+-----+-----+-----+-----+-----+-----+
| id | name  | gender | salary | start | phone | address | department |
+----+-----+-----+-----+-----+-----+-----+-----+
| 1  | Bill  | M      | 1000000.00 | 2018-01-03 | NULL | Bangalore |             |
| 2  | Terisa | F      | 2000000.00 | 2019-11-13 | NULL | Bangalore |             |
| 3  | Charlie | M      | 3000000.00 | 2020-05-21 | NULL | Bangalore |             |
+----+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> alter table employee_payroll add basic_pay DOUBLE, add deductions DOUBLE, add taxable_pay DOUBLE, add income_tax DOUBLE, add net_pay DOUBLE;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> create table department (department_id INT AUTO_INCREMENT PRIMARY KEY, department_name VARCHAR(50) NOT NULL UNIQUE);
Query OK, 0 rows affected (0.04 sec)

mysql> create table employee (employee_id INT AUTO_INCREMENT PRIMARY KEY, name varchar(50) NOT NULL, gender CHAR(1), phone varchar(20), address varchar(150) default 'Bangalore');
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE payroll ( payroll_id INT AUTO_INCREMENT PRIMARY KEY, employee_id INT NOT NULL, basic_pay DOUBLE, deductions DOUBLE, taxable_pay DOUBLE, income_tax DOUBLE, net_pay DOUBLE, FOREIGN KEY (employee_id) REFERENCES employee(employee_id) );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE employee_department ( employee_id INT NOT NULL, department_id INT NOT NULL, PRIMARY KEY(employee_id, department_id), FOREIGN KEY (employee_id) REFERENCES employee(employee_id), FOREIGN KEY (department_id) REFERENCES department(department_id) );
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO department (department_name) VALUES ('Sales'), ('Marketing'), ('Finance'), ('HR'), ('Engineering');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

**Use Case 10 :**

**Ability to make Terissa as part of Sales and Marketing Department**

- Note: The Complete employee payroll roll need to be Inserted
- If a Salary is now going to be updated multiple rows has to be updated creating unnecessary redundancy
- Further There is 2 Employee ID so according to Database there is two different Terissa

```
Command Prompt - mysql -u x + v
mysql> select * from employee_payroll;
+-----+-----+-----+-----+-----+-----+-----+
| id | name | gender | salary | start | phone | address | department |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Bill | M | 1000000.00 | 2018-01-03 | NULL | Bangalore |
| 2 | Terisa | F | 2000000.00 | 2019-11-13 | NULL | Bangalore |
| 3 | Charlie | M | 3000000.00 | 2020-05-21 | NULL | Bangalore |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

mysql> alter table employee_payroll add basic_pay DOUBLE, add deductions DOUBLE, add taxable_pay DOUBLE, add income_tax DOUBLE, add net_pay DOUBLE;
Query OK, 0 rows affected (0.01 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> create table department (department_id INT AUTO_INCREMENT PRIMARY KEY, department_name VARCHAR(50) NOT NULL UNIQUE);
Query OK, 0 rows affected (0.04 sec)

mysql> create table employee (employee_id INT AUTO_INCREMENT PRIMARY KEY, name varchar(50) NOT NULL, gender CHAR(1), phone varchar(20), address varchar(150) default 'Bangalore');
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE payroll ( payroll_id INT AUTO_INCREMENT PRIMARY KEY, employee_id INT NOT NULL, basic_pay DOUBLE, deductions DOUBLE, taxable_pay DOUBLE, income_tax DOUBLE, net_pay DOUBLE, FOREIGN KEY (employee_id) REFERENCES employee(employee_id) );
Query OK, 0 rows affected (0.04 sec)

mysql> CREATE TABLE employee_department ( employee_id INT NOT NULL, department_id INT NOT NULL, PRIMARY KEY(employee_id, department_id), FOREIGN KEY (employee_id) REFERENCES employee(employee_id), FOREIGN KEY (department_id) REFERENCES department(department_id) );
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO department (department_name) VALUES ('Sales'), ('Marketing'), ('Finance'), ('HR'), ('Engineering');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0
```

```
Command Prompt - mysql -u x + v
mysql> INSERT INTO department (department_name) VALUES ('Sales'), ('Marketing'), ('Finance'), ('HR'), ('Engineering');
Query OK, 5 rows affected (0.01 sec)
Records: 5 Duplicates: 0 Warnings: 0

mysql> INSERT INTO employee (name, gender, phone) VALUES ('Terissa', 'F', '9874563210');
Query OK, 1 row affected (0.01 sec)

mysql> SELECT department_id FROM department WHERE department_name='Sales';
+-----+
| department_id |
+-----+
| 1 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT department_id FROM department WHERE department_name='Marketing';
+-----+
| department_id |
+-----+
| 2 |
+-----+
1 row in set (0.00 sec)

mysql> INSERT INTO employee_department VALUES (1, 1);
Query OK, 1 row affected (0.01 sec)

mysql> INSERT INTO employee_department VALUES (1, 2);
Query OK, 1 row affected (0.00 sec)

mysql> UPDATE payroll SET basic_pay = 60000, deductions = 5000, taxable_pay = 55000, income_tax = 2500, net_pay = 52500 WHERE employee_id = 1;
Query OK, 0 rows affected (0.00 sec)
```

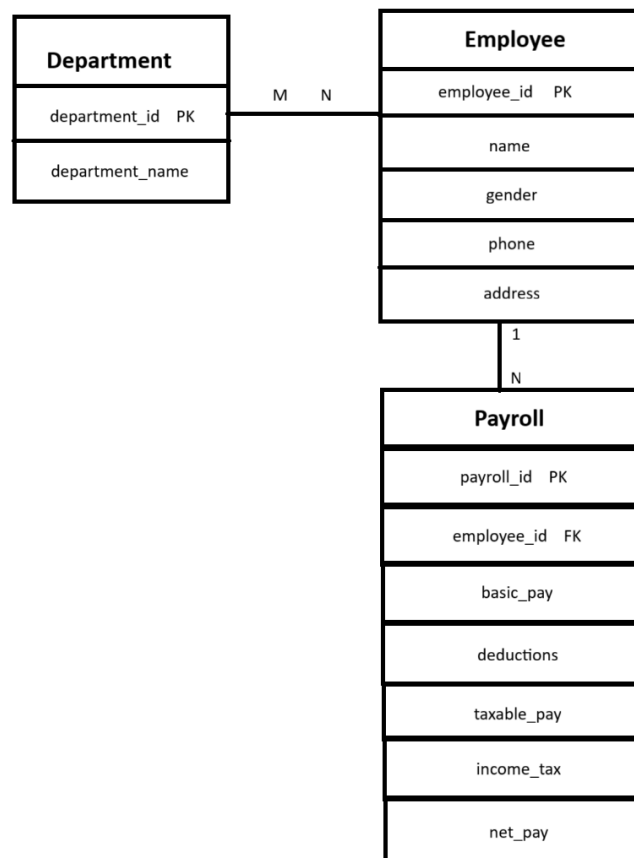
Draw the ER Diagram for  
Payroll Service DB

- Identifies the Entities – Entities can be Identified using Normalization Technique
- Check each attribute and see if they are Composite or Multi-Valued

### Use Case 11 :

#### Implement the ER Diagram into Payroll Service DB

- Create the corresponding tables as identified in the ER Diagram along with attributes
- For Many to Many relationship, create new Table called Employee Department having Employee Id and Department ID and redo the UC 7



## Use Case 12 :

Ensure all retrieve queries done especially in UC 4, UC 5 and UC 7 are working with new table structure

```
mysql> INSERT INTO payroll (employee_id, basic_pay, deductions, taxable_pay, income_tax, net_pay) VALUES (1, 0, 0, 0, 0, 0);
Query OK, 1 row affected (0.01 sec)

mysql> UPDATE payroll SET basic_pay = 60000, deductions = 5000, taxable_pay = 55000, income_tax = 2500, net_pay = 52500 WHERE employee_id = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> INSERT INTO employee (employee_id, name, gender, phone) VALUES (10, 'Mark', 'M', '9876543210');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO employee_department (employee_id, department_id) VALUES (10, 1);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO payroll (employee_id, basic_pay, deductions, taxable_pay, income_tax, net_pay) VALUES (10, 50000, 3000, 47000, 2000, 45000);
Query OK, 1 row affected (0.00 sec)

mysql> select * from employee;
+-----+-----+-----+-----+-----+
| employee_id | name  | gender | phone  | address |
+-----+-----+-----+-----+-----+
|          1 | Terissa | F      | 9874563210 | Bangalore |
|          10 | Mark   | M      | 9876543210 | Bangalore |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> select * from payroll;
+-----+-----+-----+-----+-----+-----+
| payroll_id | employee_id | basic_pay | deductions | taxable_pay | income_tax | net_pay |
+-----+-----+-----+-----+-----+-----+
|          1 |          1 |      60000 |         5000 |         55000 |         2500 |      52500 |
+-----+-----+-----+-----+-----+-----+

```

## 2 . AddressBook Problem

### Use case 1 :

**Ability to create a Address Book Service DB**

- Use SQL Client to create DB and DB records

```
Command Prompt - mysql -u x + v
+-----+-----+
2 rows in set (0.00 sec)

mysql> create database address_book_service;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+-----+-----+
| Database |
+-----+-----+
| address_book_service |
| bank_account |
| bridgelabz |
| information_schema |
| mysql |
| payroll_service |
| performance_schema |
| phpmyadmin |
| student_db |
| test |
| upskilling |
+-----+-----+
11 rows in set (0.00 sec)

mysql> use address_book_service;
Database changed
mysql> create table address_book(id INT NOT NULL AUTO_INCREMENT,first_name VARCHAR(50) NOT NULL,last_name VARCHAR(50) NOT NULL,addres
s VARCHAR(200),city VARCHAR(50),state VARCHAR(50),zip VARCHAR(10),phone_number VARCHAR(15),email VARCHAR(100),PRIMARY KEY (id));
Query OK, 0 rows affected (0.03 sec)

mysql> INSERT INTO address_book
-> (first_name, last_name, address, city, state, zip, phone_number, email)
-> VALUES
```

## Use case 2 :

Ability to create a Address Book Table with first and last names, address, city, state, zip, phone number and email as its attributes

```
Command Prompt - mysql -u x + v

mysql
payroll_service
performance_schema
phpmyadmin
student_db
test
upskilling
+-----+-----+
11 rows in set (0.00 sec)

mysql> use address_book_service;
Database changed
mysql> create table address_book(id INT NOT NULL AUTO_INCREMENT,first_name VARCHAR(50) NOT NULL,last_name VARCHAR(50) NOT NULL,addres
s VARCHAR(200),city VARCHAR(50),state VARCHAR(50),zip VARCHAR(10),phone_number VARCHAR(15),email VARCHAR(100),PRIMARY KEY (id));
Query OK, 0 rows affected (0.03 sec)
```

## Use case 3 :

Ability to insert new Contacts to Address Book

```
mysql> INSERT INTO address_book
-> (first_name, last_name, address, city, state, zip, phone_number, email)
-> VALUES
-> ('Bill', 'Gates', '108 Microsoft Road', 'Seattle', 'Washington', '98052', '9999999999', 'bill@microsoft.com'),
-> ('Elon', 'Musk', '370 Tesla Street', 'Austin', 'Texas', '73301', '8888888888', 'elon@tesla.com');
Query OK, 2 rows affected (0.00 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql> select * from address_book;
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | first_name | last_name | address | city | state | zip | phone_number | email |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Bill | Gates | 108 Microsoft Road | Seattle | Washington | 98052 | 9999999999 | bill@microsoft.com |
| 2 | Elon | Musk | 370 Tesla Street | Austin | Texas | 73301 | 8888888888 | elon@tesla.com |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

## Use case 4 :

Ability to edit existing contact person using their name

```
Command Prompt - mysql -u <user>
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | first_name | last_name | address | city | state | zip | phone_number | email |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Bill | Gates | 108 Microsoft Road | Seattle | Washington | 98052 | 9999999999 | bill@microsoft.com |
| 2 | Elon | Musk | 370 Tesla Street | Austin | Texas | 73301 | 8888888888 | elon@tesla.com |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> update address_book set phone_number = '7777777777' where first_name = 'Bill' AND last_name = 'Gates';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

## Use case 5 :

Ability to delete a person using person's name

```
mysql> Delete from address_book where first_name = 'Elon' and last_name = 'Musk';
Query OK, 1 row affected (0.01 sec)

mysql> select * from address_book where city = 'Seattle';
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | first_name | last_name | address | city | state | zip | phone_number | email |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Bill | Gates | 108 Microsoft Road | Seattle | Washington | 98052 | 7777777777 | bill@microsoft.com |
+----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

## Use case 6 :

Ability to Retrieve Person belonging to a City or State from the Address Book

```
mysql> select * from address_book where city = 'Seattle';
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | first_name | last_name | address | city | state | zip | phone_number | email |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Bill | Gates | 108 Microsoft Road | Seattle | Washington | 98052 | 7777777777 | bill@microsoft.com |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> INSERT INTO address_book (first_name, last_name, address,city,state,zip,phone_number,email) VALUES ('Heath','Ledger','560 Dark Knight','Austin','Texas','73302','8888888888','ledger@joker.com');
Query OK, 1 row affected (0.01 sec)

mysql> select * from address_book where state = 'Texas';
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | first_name | last_name | address | city | state | zip | phone_number | email |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 3 | Heath | Ledger | 560 Dark Knight | Austin | Texas | 73302 | 8888888888 | ledger@joker.com |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> select city,COUNT(*) AS count from address_book group by city;
```

### Use case 7 :

### Ability to understand the size of address book by City and State

- Here size indicates the count

```

Command Prompt - mysql -u x
+-----+
1 row in set (0.00 sec)

mysql> select city,COUNT(*) AS count from address_book group by city;
+-----+
| city | count |
+-----+
| Austin | 1 |
| Seattle | 1 |
+-----+
2 rows in set (0.00 sec)

mysql> select state,COUNT(*) AS count from address_book group by state;
+-----+
| state | count |
+-----+
| Texas | 1 |
| Washington | 1 |
+-----+
2 rows in set (0.00 sec)

```

### Use case 8 :

Ability to retrieve entries sorted alphabetically by Person's name for a given city

```
Command Prompt - mysql -u x + v
mysql> select * from address_book where city = 'Seattle' order by first_name ASC, last_name ASC;
```

id	first_name	last_name	address	city	state	zip	phone_number	email
1	Bill	Gates	108 Microsoft Road	Seattle	Washington	98052	7777777777	bill@microsoft.com

```
1 row in set (0.00 sec)
```

## Use case 9 :

### Ability to identify each Address Book with name and Type.

- Here the type could Family, Friends, Profession, etc
- Alter Address Book to add name and type

```
mysql> select * from address_book where city = 'Seattle' order by first_name ASC, last_name ASC;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | first_name | last_name | address | city | state | zip | phone_number | email |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Bill | Gates | 108 Microsoft Road | Seattle | Washington | 98052 | 7777777777 | bill@microsoft.com |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> alter table address_book add address_book_name VARCHAR(50);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table address_book add address_book_type VARCHAR(50);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> update address_book set address_book_name = 'Personal', address_book_type = 'Friends' where first_name = 'Bill';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update address_book set address_book_name = 'Office', address_book_type = 'Colleague' where first_name = 'Heath';
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from address_book;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | first_name | last_name | address | city | state | zip | phone_number | email | address_book_name | address_book_type |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Bill | Gates | 108 Microsoft Road | Seattle | Washington | 98052 | 7777777777 | bill@microsoft.com | Personal | Friends |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

## Use case 10 :

### Ability to get number of contact persons i.e. count by type

```
mysql> select address_book_type, COUNT(*) As count from address_book group by address_book_type;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'address_book_type' at line 1
mysql> select address_book_type, COUNT(*) as count from address_book group by address_book_type;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'address_book_type' at line 1
mysql> select address_book_type, COUNT(*) as count from address_book group by address_book_type;
+-----+-----+
| address_book_type | count |
+-----+-----+
| Colleague | 1 |
| Family | 2 |
| Friends | 1 |
+-----+-----+
3 rows in set (0.00 sec)

mysql>
```

## Use case 11 :

Ability to add person to both Friend and Family

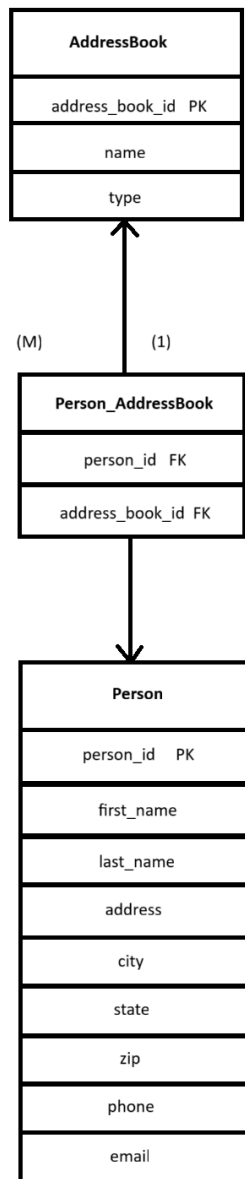
```
mysql> INSERT INTO address_book (first_name, last_name, address,city,state,zip,phone_number,email,address_book_name,address_book_type
) select first_name,last_name,address,city,state,zip,phone_number,email,'Personal','Family' from address_book where first_name = 'Bil
l' and last_name = 'Gates';
Query OK, 1 row affected (0.01 sec)
Records: 1 Duplicates: 0 Warnings: 0

mysql> select * from address_book;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | first_name | last_name | address          | city   | state   | zip   | phone_number | email          | address_book_name | address_book_type |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Bill      | Gates    | 108 Microsoft Road | Seattle | Washington | 98052 | 7777777777 | bill@microsoft.com | Personal          | Family            |
| 3 | Heath    | Ledger   | 560 Dark Knight   | Austin | Texas     | 73302 | 8888888888 | ledger@joker.com   | Personal          | Family            |
| 4 | Christopher | Nolan   | 780 The Bat Man   | Austin | Texas     | 73202 | 9898989898 | chris@nolan.com    | Office            | Family            |
| 5 | Bill      | Gates    | 108 Microsoft Road | Seattle | Washington | 98052 | 7777777777 | bill@microsoft.com | Personal          | Family            |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

## Use case 12 :

Draw the ER Diagram for Address Book Service DB

- Identifies the Entities – Entities can be Identified using Normalization Technique
- Check each attribute and see if they are Composite or Multi-Valued



### Use case 13 :

Ensure all retrieve queries done especially in UC 6, UC 7, UC 8 and UC 10 are working with new table structure

```
Command Prompt - mysql -u x + v

mysql> use address_book_service;
Database changed
mysql> show tables;
+-----+
| Tables_in_address_book_service |
+-----+
| address_book                    |
+-----+
1 row in set (0.01 sec)

mysql> select * from address_book;
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| id | first_name | last_name | address          | city   | state   | zip   | phone_number | email          | address_book |
|_name | address_book_type |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 1 | Bill      | Gates    | 108 Microsoft Road | Seattle | Washington | 98052 | 7777777777 | bill@microsoft.com | Personal
| 3 | Heath    | Ledger   | 560 Dark Knight   | Austin | Texas     | 73302 | 8888888888 | ledger@joker.com   | Personal
| 4 | Christopher | Nolan   | 780 The Bat Man   | Austin | Texas     | 73202 | 9898989898 | chris@nolan.com    | Office
| 5 | Bill      | Gates    | 108 Microsoft Road | Seattle | Washington | 98052 | 7777777777 | bill@microsoft.com | Personal
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.03 sec)

mysql> CREATE TABLE person ( person_id INT AUTO_INCREMENT PRIMARY KEY, first_name VARCHAR(50), last_name VARCHAR(50), address VARCHAR(100), city VARCHAR(50), state VARCHAR(50), zip VARCHAR(10), phone_number VARCHAR(15), email VARCHAR(100) );
Query OK, 0 rows affected (0.02 sec)
```

```
Command Prompt - mysql -u x + v

mysql> CREATE TABLE person ( person_id INT AUTO_INCREMENT PRIMARY KEY, first_name VARCHAR(50), last_name VARCHAR(50), address VARCHAR(100), city VARCHAR(50), state VARCHAR(50), zip VARCHAR(10), phone_number VARCHAR(15), email VARCHAR(100) );
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----+
| Tables_in_address_book_service |
+-----+
| address_book                    |
| person                         |
+-----+
2 rows in set (0.00 sec)

mysql> CREATE TABLE addressbook ( address_book_id INT AUTO_INCREMENT PRIMARY KEY, address_book_name VARCHAR(50), address_book_type VARCHAR(50) );
Query OK, 0 rows affected (0.01 sec)

mysql> CREATE TABLE person_addressbook ( person_id INT, address_book_id INT, FOREIGN KEY (person_id) REFERENCES person(person_id) ON DELETE CASCADE, FOREIGN KEY (address_book_id) REFERENCES addressbook(address_book_id) ON DELETE CASCADE );
Query OK, 0 rows affected (0.05 sec)

mysql> INSERT INTO addressbook (address_book_name, address_book_type) VALUES ('Friends', 'Personal'), ('Family', 'Personal'), ('Colleague', 'Office');
Query OK, 3 rows affected (0.01 sec)
Records: 3 Duplicates: 0 Warnings: 0

mysql> INSERT INTO person (first_name, last_name, address, city, state, zip, phone_number, email) VALUES ('Bill', 'Gates', '108 Microsoft Road', 'Seattle', 'Washington', '98052', '7777777777', 'bill@microsoft.com'), ('Heath', 'Ledger', '560 Dark Knight', 'Austin', 'Texas', '73302', '8888888888', 'ledger@joker.com'), ('Christopher', 'Nolan', '780 The Bat Man', 'Austin', 'Texas', '73202', '9898989898', 'chris@nolan.com');
Query OK, 3 rows affected (0.01 sec)
Records: 3 Duplicates: 0 Warnings: 0
```

```
Command Prompt - mysql -u x + v

mysql> SELECT * FROM person_addressbook;
+-----+-----+
| person_id | address_book_id |
+-----+-----+
| 1 | 1 |
| 1 | 2 |
| 2 | 2 |
| 3 | 3 |
+-----+-----+
4 rows in set (0.00 sec)

mysql> SELECT p.first_name, p.last_name, p.city, p.state, ab.address_book_name, ab.address_book_type FROM person p JOIN person_addressbook pa ON p.person_id = pa.person_id JOIN addressbook ab ON pa.address_book_id = ab.address_book_id WHERE p.city = 'Seattle' OR p.state = 'Washington';
+-----+-----+-----+-----+-----+-----+
| first_name | last_name | city | state | address_book_name | address_book_type |
+-----+-----+-----+-----+-----+-----+
| Bill | Gates | Seattle | Washington | Friends | Personal |
| Bill | Gates | Seattle | Washington | Family | Personal |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> SELECT p.first_name, p.last_name, p.city, p.state FROM person p WHERE p.city = 'Austin' OR p.state = 'Texas';
+-----+-----+-----+-----+
| first_name | last_name | city | state |
+-----+-----+-----+-----+
| Heath | Ledger | Austin | Texas |
| Christopher | Nolan | Austin | Texas |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> SELECT city, state, COUNT(*) AS total_contacts FROM person GROUP BY city, state;
```

```
Command Prompt - mysql -u x + v

mysql> SELECT city, state, COUNT(*) AS total_contacts FROM person GROUP BY city, state;
+-----+-----+-----+
| city | state | total_contacts |
+-----+-----+-----+
| Austin | Texas | 2 |
| Seattle | Washington | 1 |
+-----+-----+-----+
2 rows in set (0.01 sec)

mysql> SELECT first_name, last_name, city FROM person WHERE city = 'Austin' ORDER BY first_name, last_name;
+-----+-----+-----+
| first_name | last_name | city |
+-----+-----+-----+
| Christopher | Nolan | Austin |
| Heath | Ledger | Austin |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> SELECT ab.address_book_type, COUNT(pa.person_id) AS total_contacts FROM addressbook ab JOIN person_addressbook pa ON ab.address_book_id = pa.address_book_id GROUP BY ab.address_book_type;
+-----+-----+
| address_book_type | total_contacts |
+-----+-----+
| Office | 1 |
| Personal | 3 |
+-----+-----+
2 rows in set (0.00 sec)

mysql> INSERT INTO person_addressbook (person_id, address_book_id) VALUES (1, 1), (1, 3);
Query OK, 2 rows affected (0.03 sec)
Records: 2 Duplicates: 0 Warnings: 0

mysql>
```