

Lab-1: Basics of DDL and DML Statements

Prepare the Lab Sheet of MYSQL Statements for the following.

1. Create a database named “Yourname_Roll_COMPANY” e.g.: Atiz_02_Company and then create the following tables within the database. Specify proper primary keys and the needed constraints while defining the tables. Use appropriate data types for the attributes.

ANSWER: Creating a database named ‘Prabhat_22_Company’

SQL QUERY:

```
--  
CREATE DATABASE Prabhat_22_COMPANY;
```

OUTPUT:

Name	Value
Updated Rows	1
Query	CREATE DATABASE Prabhat_22_COMPANY
Start time	Wed Apr 03 00:07:19 NPT 2024
Finish time	Wed Apr 03 00:07:19 NPT 2024

1. a. Employee (SSN, Ename, Gender, Bdate, Address, Salary, Ono, Years_of_experience); whereOno is a foreign key referencing the Office table. Set the default value of salary to 0.00. The Ename should not be null. Set SSN to auto increment. The name and address should be varchar, Gender should be char(1), Bdate should be date type, and Salary should be decimal type with two digits after the decimal. Years_of_experience should be integer. Use Check constraint for gender as CHECK (Gender IN ('M', 'F'))

Creating Table name EMPLOYEE

SQL QUERY:

The screenshot shows the MySQL Workbench interface. On the left, there's a tree view of databases, tables, and other schema objects under the 'Prabhat_22_COMPANY' database. A context menu is open over the 'Tables' node of the 'Prabhat_22_COMPANY' database. The menu item 'CREATE TABLE' is highlighted in blue. To the right of the menu, the SQL query for creating the 'EMPLOYEE' table is displayed:

```
CREATE TABLE EMPLOYEE (
    SSN INT AUTO_INCREMENT PRIMARY KEY,
    Ename VARCHAR(255) NOT NULL,
    Gender CHAR(1) NOT NULL,
    CONSTRAINT check_gender CHECK (Gender IN ('M', 'F')),
    Bdate DATE,
    Address VARCHAR(255),
    Salary DECIMAL(10, 2) DEFAULT '0.00',
    Ono INT,
    FOREIGN KEY (Ono) REFERENCES Office(ONumber),
    Years_of_experience INT
);
```

OUTPUT:

Name	Value
Updated Rows 0	
Query	CREATE TABLE EMPLOYEE (SSN INT AUTO_INCREMENT PRIMARY KEY, Ename VARCHAR(255) NOT NULL, Gender CHAR(1) NOT NULL, CONSTRAINT check_gender CHECK (Gender IN ('M', 'F')), Bdate DATE, Address VARCHAR(255), Salary DECIMAL(10, 2) DEFAULT '0.00', Ono INT, FOREIGN KEY (Ono) REFERENCES Office(ONumber), Years_of_experience INT)
Start time	Sun Mar 31 19:16:13 NPT 2024
Finish time	Sun Mar 31 19:16:13 NPT 2024

1. b. Office (Onumber, Oname, Country); where Oname should not be NULL. The country should be varchar.

SQL QUERY

```
•CREATE TABLE Office (
    Onumber INT PRIMARY KEY,
    Oname VARCHAR(255) NOT NULL,
    Country VARCHAR(255)
);
```

OUTPUT

Name	Value
Updated Rows	0
Query	CREATE TABLE Office (Onumber INT PRIMARY KEY, Oname VARCHAR(255) NOT NULL, Country VARCHAR(255))
Start time	Sun Mar 31 19:07:55 NPT 2024
Finish time	Sun Mar 31 19:07:55 NPT 2024

c. Project (Pnumber, Pname, Plocation, Onumber); where Onumber is a foreign key referencing Office table. Create a constraint name fk_pro for the foreign key. The pname should be unique and should not be null. Both Pname and Plocations should be of type varchar(40).

SQL QUERY:

```
•CREATE TABLE Project (
    Pnumber INT PRIMARY KEY,
    Pname VARCHAR(40) NOT NULL UNIQUE,
    Plocation VARCHAR(40),
    Onumber INT,
    CONSTRAINT fk_pro FOREIGN KEY (Onumber) REFERENCES Office(Onumber)
)
```

OUTPUT:

Name	Value
Updated Rows	0
Query	<pre>CREATE TABLE Project (Pnumber INT PRIMARY KEY, Pname VARCHAR(40) NOT NULL UNIQUE, Plocation VARCHAR(40), Onumber INT, CONSTRAINT fk_pro FOREIGN KEY (Onumber) REFERENCES Office(Onumber))</pre>
Start time	Sun Mar 31 20:42:03 NPT 2024
Finish time	Sun Mar 31 20:42:03 NPT 2024

1. d. Works_on(ESSN, Pno); where ESSN references Employee SSN and Pno references to Pnumber from Project. Set cascade on update and cascade on delete to both

SQL QUERY:

```
•CREATE TABLE Works_on (
ESSN INT,
Pno INT,
FOREIGN KEY (ESSN) REFERENCES EMPLOYEE(SSN) ON UPDATE CASCADE ON DELETE CASCADE,
FOREIGN KEY (Pno) REFERENCES Project(Pnumber) ON UPDATE CASCADE ON DELETE CASCADE,
PRIMARY KEY (ESSN, Pno)
)
```

OUTPUT:

Name	Value
Updated Rows	0
Query	<pre>CREATE TABLE Works_on(ESSN INT, Pno INT, FOREIGN KEY (ESSN) REFERENCES EMPLOYEE(SSN) ON UPDATE CASCADE ON DELETE CASCADE, FOREIGN KEY (Pno) REFERENCES Project(Pnumber) ON UPDATE CASCADE ON DELETE CASCADE, PRIMARY KEY (ESSN, Pno))</pre>
Start time	Sun Mar 31 20:54:07 NPT 2024
Finish time	Sun Mar 31 20:54:08 NPT 2024

1. e. Dependents(Did, Dname, Dage, SSN); where SSN is a Foreign key referencing the employee. Set NULL on delete and on update to the foreign key. Add constraint age_constraint using CHECK(Dage<16).

SQL QUERY:

```
>CREATE TABLE Dependents (
    Did INT PRIMARY KEY,
    Dname VARCHAR(255),
    Dage INT,
    CONSTRAINT age_constraint CHECK (Dage < 16),
    SSN INT,
    FOREIGN KEY (SSN) REFERENCES EMPLOYEE(SSN) ON DELETE SET NULL ON UPDATE SET NULL
)
```

OUTPUT:

Name	Value
Updated Rows	0
Query	CREATE TABLE Dependents (Did INT PRIMARY KEY, Dname VARCHAR(255), Dage INT, CONSTRAINT age_constraint CHECK (Dage < 16), SSN INT, FOREIGN KEY (SSN) REFERENCES EMPLOYEE(SSN) ON DELETE SET NULL ON UPDATE SET NULL)
Start time	Sun Mar 31 21:44:17 NPT 2024
Finish time	Sun Mar 31 21:44:17 NPT 2024

2. Alter table Dependents and add an attribute Drelation of type Char(50).

SQL QUERY:

```
ALTER TABLE Dependents
ADD Drelation CHAR(50);
```

OUTPUT:

Name	Value
Updated Rows	0
	ALTER TABLE Dependents ADD Drelation CHAR(50)
Start time	Sun Mar 31 21:48:09 NPT 2024
Finish time	Sun Mar 31 21:48:09 NPT 2024

Before Making Alter Statement, the Dependents table schema structure was like this:

COLUMNS 1 X		
	SELECT COLUMN_NAME, DATA_TYPE FROM INFORMATION_SCHEMA.COLUMNS	
	COLUMN_NAME	DATA_TYPE
1	Did	int
2	Dname	varchar
3	Dage	int
4	SSN	int

After Making Alter Statement, the Dependents table schema structure is like this:

COLUMNS 1 X		
	SELECT COLUMN_NAME, DATA_TYPE FROM INFORMATION_SCHEMA.COLUMNS	
	COLUMN_NAME	DATA_TYPE
1	Did	int
2	Dname	varchar
3	Dage	int
4	SSN	int
5	Drelation	char

Impact: A new column name called ‘Drelation’ with char(50) data type is added here.

3. Alter table Dependent and modify the attribute Drelation of type Char(50) to Varchar(50)

SQL QUERY:

```
•ALTER TABLE Dependents  
    MODIFY Drelation VARCHAR(50);
```

OUTPUT:

Name	Value
Updated Rows	0
Query	ALTER TABLE Dependents MODIFY Drelation VARCHAR(50)
Start time	Sun Mar 31 21:54:00 NPT 2024
Finish time	Sun Mar 31 21:54:01 NPT 2024

SELECTING THE NAME OF COLUMNS AS WELL AS THEIR DATA TYPES FROM Dependents Table Before & After Modifying the Dependents table schema.

```
•SELECT COLUMN_NAME, DATA_TYPE FROM  
INFORMATION_SCHEMA.COLUMNS  
WHERE TABLE_NAME = "Dependents";
```

COLUMN NAMES OF Dependent Table Along with Their Data Types Before Applying Modify Operation:

	COLUMN_NAME	DATA_TYPE
1	Did	int
2	Dname	varchar
3	Dage	int
4	SSN	int
5	Drelation	char

COLUMN NAMES OF Dependent Table Along with Their Data Types AFTER Applying to Modify Operation

COLUMN 1 X		
	ABC COLUMN NAME	ABC DATA TYPE
1	Did	int
2	Dname	varchar
3	Dage	int
4	SSN	int
5	Drelation	varchar

4. Insert at least five tuples into the tables. (Illustrate insertion of single tuple and multiple tuples both). During insertion insert the following as well.

One of the tuples in the Office table should have the office name "Yourname_Office_Roll" i.e. Deric_Office_06. Similarly one of the tuple in an employee should have a salary of 30000.

In addition, there should be one tuple in the office table having the office name Yourname_Ncell_Roll.

Single Insertion:

SQL QUERY:

```
•INSERT INTO Office(Onumber, Oname, Country)  
VALUES (22, 'Prabhat_Office_22', 'Nepal');
```

OUTPUT:

Statistics 1 X	
Name	Value
Updated Rows	1
	INSERT INTO Office(Onumber, Oname, Country)
	VALUES (22, 'Prabhat_Office_22', 'Nepal')
Start time	Sun Mar 31 22:14:44 NPT 2024
Finish time	Sun Mar 31 22:14:44 NPT 2024

SQL QUERY: (Single Insertion)

```
•INSERT INTO Office(Onumber, Oname, Country)  
  VALUES (11, 'Prabhat_Ncell_22', 'India');
```

OUTPUT:

Statistics 1 X	
Name	Value
Updated Rows	1
	INSERT INTO Office(Onumber, Oname, Country) VALUES (11, 'Prabhat_Ncell_22', 'India')
Start time	Sun Mar 31 23:09:31 NPT 2024
Finish time	Sun Mar 31 23:09:31 NPT 2024

INSERTION OF MULTIPLE TUPLES.

SQL QUERY:

```
•INSERT INTO Office(Onumber, Oname, Country)  
  VALUES  
    (2, 'Anish_Office_2', 'Canada'),  
    (33, 'Suman_Office_33', 'USA'),  
    (12, 'Gaurav_Office_12', 'Germany');
```

OUTPUT:

Statistics 1 X	
Name	Value
Updated Rows	3
Query	INSERT INTO Office(Onumber, Oname, Country) VALUES (2, 'Anish_Office_2', 'Canada'), (33, 'Suman_Office_33', 'USA'), (12, 'Gaurav_Office_12', 'Germany')
Start time	Sun Mar 31 23:13:17 NPT 2024
Finish time	Sun Mar 31 23:13:17 NPT 2024

Office Table After Inserting 5 rows:

The screenshot shows a database interface with a query editor and a results grid. The query editor contains the SQL command: `SELECT * FROM Office;`. The results grid, titled "Office 1 X", displays the following data:

	Onumber	Oname	Country
1	2	Anish_Office_2	Canada
2	11	Prabhat_Ncell_22	India
3	12	Gaurav_Office_12	Germany
4	22	Prabhat_Office_22	Nepal
5	33	Suman_Office_33	USA

There should be one record in the Employee table having Ename “Your name” i. e. Deric and SSN “Your roll number” e.g. 6

SQL QUERY:

```
•INSERT INTO EMPLOYEE(Ename, Gender, Bdate, Address, Salary, Ono, Years_of_experience)
VALUES
('Prabhat', 'M', '2054-12-15', 'Kritipur', 200000, 22, 4),
('Gaurav', 'M', '2056-07-24', 'MacheGaun', 30000, 12, 1);
```

OUTPUT:

The screenshot shows a database interface with a results grid titled "Statistics 1 X". The grid displays the following information:

Name	Value
Updated Rows	2
	INSERT INTO EMPLOYEE(Ename, Gender, Bdate, Address, Salary, Ono, Years_of_experience) VALUES ('Prabhat', 'M', '2054-12-15', 'Kritipur', 200000, 22, 4), ('Gaurav', 'M', '2056-07-24', 'MacheGaun', 30000, 12, 1)
Start time	Tue Apr 02 19:10:29 NPT 2024
Finish time	Tue Apr 02 19:10:29 NPT 2024

INSERTION OF MULTIPLE TUPLES

SQL QUERY:

```

•INSERT INTO EMPLOYEE(Ename, Gender, Bdate, Address, Salary, Ono, Years_of_experience)
VALUES
('Anish', 'M', '2053-12-05', 'Budhanilkandha', 95000, 2, 3),
('Prabhat', 'M', '2054-12-15', 'Nawalparasi', 80000, 11, 2),
('Suman', 'M', '2052-07-06', 'Sunwal', 75000, 33, 3);

```

OUTPUT:

Name	Value
Updated Rows 3	
INSERT INTO EMPLOYEE(Ename, Gender, Bdate, Address, Salary, Ono, Years_of_experience)	
VALUES	
	('Anish', 'M', '2053-12-05', 'Budhanilkandha', 95000, 2, 3),
	('Prabhat', 'M', '2054-12-15', 'Nawalparasi', 80000, 11, 2),
	('Suman', 'M', '2052-07-06', 'Sunwal', 75000, 33, 3)
Start time	Tue Apr 02 19:46:10 NPT 2024
Finish time	Tue Apr 02 19:46:10 NPT 2024

Employee Table After Inserting 5 rows:

SSN	Ename	Gender	Bdate	Address	Salary	Ono	Years_of_experience
1	Prabhat	M	2054-12-15	Kritipur	200,000	22	4
2	Gaurav	M	2056-07-24	MacheGaun	30,000	12	1
3	Anish	M	2053-12-05	Budhanilkandha	95,000	2	3
4	Prabhat	M	2054-12-15	Nawalparasi	80,000	11	2
5	Suman	M	2052-07-06	Sunwal	75,000	33	3

There should be one record in the Project table having Pname =“Your name_ProjMDS” and Pnumber = 2*Your Roll number.

SQL QUERY:

```

•INSERT INTO Project(Pnumber, Pname, Plocation, Onumber)
VALUES
(44, 'Prabhat_ProjMDS', 'Kathmandu', 22),
(22, 'Prabhat_ProjDBMS', 'USA', 11),
(24, 'Gaurav_ProjPython', 'Butwal', 12),
(4, 'Anish_ProjPandas', 'Chitwan', 2),
(66, 'Suman_ProjML', 'UK', 33);

```

OUTPUT:

Name	Value
Updated Rows	5
Query	INSERT INTO Project(Pnumber, Pname, Plocation, Onumber) VALUES (44, 'Prabhat_ProjMDS', 'Kathmandu', 22), (22, 'Prabhat_ProjDBMS', 'USA', 11), (24, 'Gaurav_ProjPython', 'Butwal', 12), (4, 'Anish_ProjPandas', 'Chitwan', 2), (66, 'Suman_ProjML', 'UK', 33)
Start time	Tue Apr 02 22:34:50 NPT 2024
Finish time	Tue Apr 02 22:34:50 NPT 2024

Project Table After Inserting 5 rows:

	Pnumber	Pname	Plocation	Onumber
1	4	Anish_ProjPandas	Chitwan	2
2	22	Prabhat_ProjDBMS	USA	11
3	24	Gaurav_ProjPython	Butwal	12
4	44	Prabhat_ProjMDS	Kathmandu	22
5	66	Suman_ProjML	UK	33

Insert the five tuples into Works_on table.

SQL QUERY:

```
• INSERT INTO Works_on(ESSN, Pno)
VALUES
(1, 44),
(2, 24),
(3, 4),
(4, 22),
(5, 66);
```

OUTPUT:

Name	Value
Updated Rows 5	
Query	INSERT INTO Works_on(ESSN, Pno) VALUES (1, 44), (2, 24), (3, 4), (4, 22), (5, 66)
Start time	Tue Apr 02 22:48:09 NPT 2024
Finish time	Tue Apr 02 22:48:09 NPT 2024

Works_on Table After Inserting 5 rows:

	ESSN	Pno
1	3	4
2	4	22
3	2	24
4	1	44
5	5	66

In the dependents table insert the rows with Dname and Drelation having values from your family. For example, Deric has his elder brother and mother as his dependents. So the table will have records with values Dname=Denish and Drelation=Brother and Dname=Gayatri and Drelation=Mother. Take assumptions based on your family members while inserting the values

Answer:

Since we have age constraint which restricts the dependent age to be less than 16 that's why it is really hard to select Drelation for Monther as Mother is always less than 16.

SQL QUERY:

```

•INSERT INTO Dependents(Did, Dname, Dage, SSN, Drelation)
VALUES
(1, "Rihans", 6, 1, "Brother"),
(2, "Kabya", 8, 1, "Sister"),
(3, "Kabir", 3, 4, "Brother"),
(4, "Subin", 2, 5, "Son"),
(5, "Anisha", 1, 3, "Daughter"),
(6, "Gauri", 9, 2, "Sister");

```

OUTPUT:

Name	Value	
Updated Rows 6		
Query	INSERT INTO Dependents(Did, Dname, Dage, SSN, Drelation) VALUES (1, "Rihans", 6, 1, "Brother"), (2, "Kabya", 8, 1, "Sister"), (3, "Kabir", 3, 4, "Brother"), (4, "Subin", 2, 5, "Son"), (5, "Anisha", 1, 3, "Daughter"), (6, "Gauri", 9, 2, "Sister")	
Start time	Tue Apr 02 23:01:03 NPT 2024	
Finish time	Tue Apr 02 23:01:03 NPT 2024	

Dependent Table After Inserting 5 rows:

Did	Dname	Dage	SSN	Drelation
1	Rihans	6	1	Brother
2	Kabya	8	1	Sister
3	Kabir	3	4	Brother
4	Subin	2	5	Son
5	Anisha	1	3	Daughter
6	Gauri	9	2	Sister

5. Update the name of the office having office name “Yourname_Ncell_Roll” to “Yourname_Ntc_Roll”.

SQL QUERY:

```
•UPDATE Office  
SET Oname = "Prabhat_Ntc_22"  
WHERE Oname = "Prabhat_Ncell_22";
```

OUTPUT:

Name	Value
Updated Rows	1
Query	UPDATE Office SET Oname = "Prabhat_Ntc_22" WHERE Oname = "Prabhat_Ncell_22"
Start time	Tue Apr 02 23:10:27 NPT 2024
Finish time	Tue Apr 02 23:10:27 NPT 2024

Office Table Before Update:

SELECT * FROM Office;			
Grid	Onumber	Oname	Country
1	2	Anish_Office_2	Canada
2	11	Prabhat_Ncell_22	India
3	12	Gaurav_Office_12	Germany
4	22	Prabhat_Office_22	Nepal
5	33	Suman_Office_33	USA

Office Table After Update:

The screenshot shows a database interface with a title bar 'Office 1 X'. Below it is a search bar with the query 'SELECT * FROM Office' and a placeholder 'Enter a SQL expression to filter results (use Ctrl+Space)'. The main area displays a table with three columns: 'Onumber', 'Oname', and 'Country'. The data is as follows:

	Onumber	Oname	Country
1	2	Anish_Office_2	Canada
2	11	Prabhat_Ntc_22	India
3	12	Gaurav_Office_12	Germany
4	22	Prabhat_Office_22	Nepal
5	33	Suman_Office_33	USA

6. Delete employees whose SSN is 1.

SQL QUERY:

```
•DELETE FROM EMPLOYEE  
WHERE SSN = 1;
```

OUTPUT:

The screenshot shows a statistics window titled 'Statistics 1 X'. It displays the following information:

Name	Value
Updated Rows	1
DELETE FROM EMPLOYEE	
WHERE SSN = 1	
Start time	Tue Apr 02 23:15:24 NPT 2024
Finish time	Tue Apr 02 23:15:24 NPT 2024

EMPLOYEE TABLE BEFORE DELETE OPERATION:

	SSN	Ename	Gender	Bdate	Address	Salary	Ono	Years_of_experience
1	1	Prabhat	M	2054-12-15	Kritipur	200,000	22	4
2	2	Gaurav	M	2056-07-24	MacheGaun	30,000	12	1
3	3	Anish	M	2053-12-05	Budhanilkandha	95,000	2	3
4	4	Prabhat	M	2054-12-15	Nawalparasi	80,000	11	2
5	5	Suman	M	2052-07-06	Sunwal	75,000	33	3

EMPLOYEE TABLE AFTER DELETE OPERATION:

	SSN	Ename	Gender	Bdate	Address	Salary	Ono	Years_of_experience
1	2	Gaurav	M	2056-07-24	MacheGaun	30,000	12	1
2	3	Anish	M	2053-12-05	Budhanilkandha	95,000	2	3
3	4	Prabhat	M	2054-12-15	Nawalparasi	80,000	11	2
4	5	Suman	M	2052-07-06	Sunwal	75,000	33	3

7. Alter table Project to rename the attribute in Plocation to Proj_location

SQL QUERY:

```
•ALTER TABLE Project
  CHANGE Plocation Proj_location VARCHAR(255);
```

OUTPUT:

Name	Value
Updated Rows	5
	ALTER TABLE Project
	CHANGE Plocation Proj_location VARCHAR(255)
Start time	Tue Apr 02 23:20:16 NPT 2024
Finish time	Tue Apr 02 23:20:16 NPT 2024

SQL TABLE BEFORE COLUMN RENAMING:

Project 1 X

SELECT * FROM Project Enter a SQL expression to filter results (use Ctrl+Space)

	Pnumber	Pname	Plocation	Onumber
1	4	Anish_ProjPandas	Chitwan	2
2	22	Prabhat_ProjDBMS	USA	11
3	24	Gaurav_ProjPython	Butwal	12
4	44	Prabhat_ProjMDS	Kathmandu	22
5	66	Suman_ProjML	UK	33

SQL TABLE AFTER COLUMN RENAMING:

SELECT * FROM Project

	Pnumber	Pname	Proj_location	Onumber
1	4	Anish_ProjPandas	Chitwan	2
2	22	Prabhat_ProjDBMS	USA	11
3	24	Gaurav_ProjPython	Butwal	12
4	44	Prabhat_ProjMDS	Kathmandu	22
5	66	Suman_ProjML	UK	33

8. Select tuples from all of the tables individually

EMPLOYEE TABLE:

SQL QUERY:

```
SELECT * FROM EMPLOYEE;
```

OUTPUT:

	SSN	Ename	Gender	Bdate	Address	Salary	Ono	Years_of_experience
1	2	Gaurav	M	2056-07-24	MacheGaun	30,000	12	1
2	3	Anish	M	2053-12-05	Budhanilkandha	95,000	2	3
3	4	Prabhat	M	2054-12-15	Nawalparasi	80,000	11	2
4	5	Suman	M	2052-07-06	Sunwal	75,000	33	3

OFFICE TABLE:

SQL QUERY:

```
SELECT * FROM Office;
```

OUTPUT:

Office 1 X

SELECT * FROM Office Enter a SQL expression to filter results (use Ctrl+Space)

	Onumber	Oname	Country
1	2	Anish_Office_2	Canada
2	11	Prabhat_Ntc_22	India
3	12	Gaurav_Office_12	Germany
4	22	Prabhat_Office_22	Nepal
5	33	Suman_Office_33	USA

PROJECT TABLE:

SQL QUERY:

```
SELECT * FROM Project;
```

OUTPUT:

Project 1 X

SELECT * FROM Project Enter a SQL expression to filter results (use Ctrl+Space)

	Pnumber	Pname	Proj_location	Onumber
1	4	Anish_ProjPandas	Chitwan	2
2	22	Prabhat_ProjDBMS	USA	11
3	24	Gaurav_ProjPython	Butwal	12
4	44	Prabhat_ProjMDS	Kathmandu	22
5	66	Suman_ProjML	UK	33

Works_on Table

SQL QUERY:

```
SELECT * FROM Works_on;
```

OUTPUT:

Works_on 1 ×

SELECT * FROM Works_on Enter a SQL expression

Grid	ESSN	Pno
1	3	4
2	4	22
3	2	24
4	5	66

Dependents Table:

SQL QUERY:

```
SELECT * FROM Dependents;
```

OUTPUT:

Dependents 1 ×

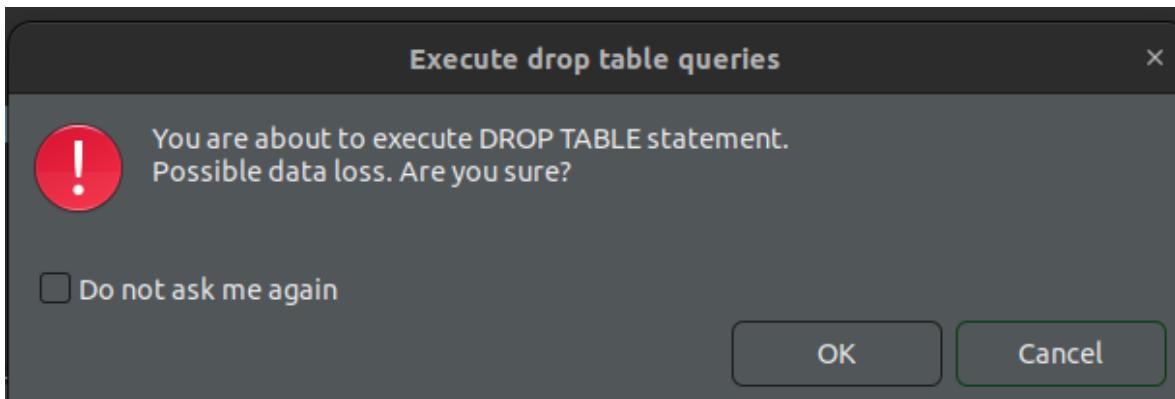
SELECT * FROM Dependents Enter a SQL expression to filter results (use Ctrl+Space)

Grid	Did	Dname	Dage	SSN	Drelation
1	1	Rihans	6	[NULL]	Brother
2	2	Kabya	8	[NULL]	Sister
3	3	Kabir	3	4	Brother
4	4	Subin	2	5	Son
5	5	Anisha	1	3	Daughter
6	6	Gauri	9	2	Sister

9. Drop the table `Works_on`. Make sure to export your database before you drop it so that you can recover.

`DROP TABLE Works_on`

```
DROP TABLE Works_on;
```



Statistics 1 X	
Name	Value
Updated Rows	0
Query	<code>DROP TABLE Works_on</code>
Start time	Wed Apr 03 00:00:29 NPT 2024
Finish time	Wed Apr 03 00:00:29 NPT 2024

10. Drop the constraint `age_constraint` for the dependent table

SQL QUERY:

```
ALTER TABLE Dependents DROP CONSTRAINT age_constraint;
```

OUTPUT:

Name	Value
Updated Rows	0
	ALTER TABLE Dependents DROP CONSTRAINT age_constraint
Start time	Wed Apr 03 00:03:17 NPT 2024
Finish time	Wed Apr 03 00:03:17 NPT 2024

11. Drop the database COMPANY. Make sure to export your database before you drop it so that you can recover.

SQL QUERY:

```
DROP DATABASE Prabhat_22_COMPANY;
```

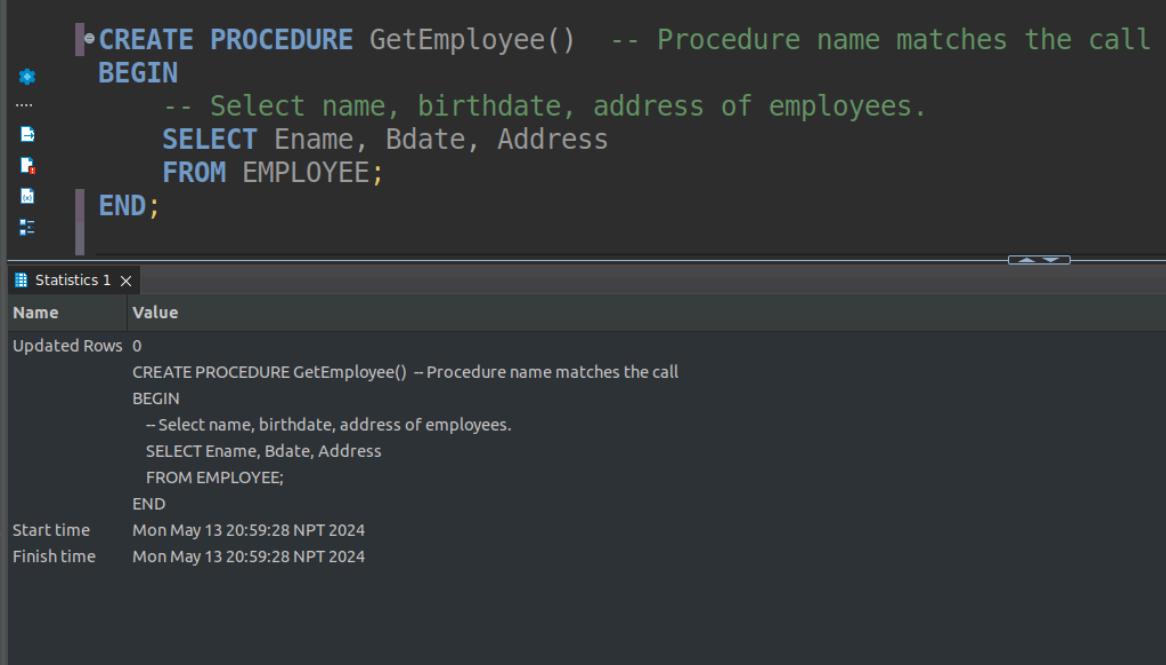
OUTPUT:

Name	Value
Updated Rows	4
Query	DROP DATABASE Prabhat_22_COMPANY
Start time	Wed Apr 03 00:06:13 NPT 2024
Finish time	Wed Apr 03 00:06:13 NPT 2024

Prabhat Ale

Roll No:22

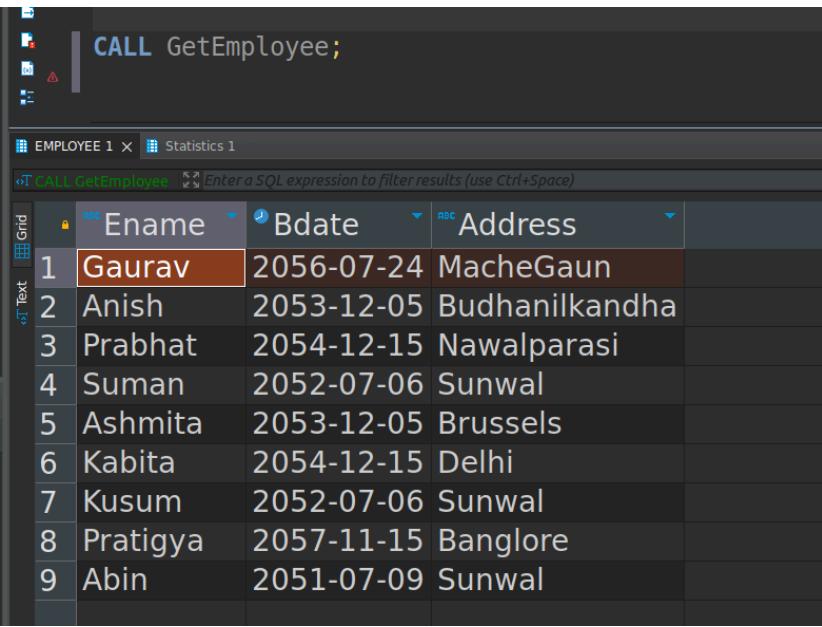
1. Write a stored procedure named “GetEmployee()” to get the name, birthdate, and address of employees.



```
CREATE PROCEDURE GetEmployee() -- Procedure name matches the call
BEGIN
    -- Select name, birthdate, address of employees.
    SELECT Ename, Bdate, Address
    FROM EMPLOYEE;
END;
```

Name	Value
Updated Rows	0
CREATE PROCEDURE GetEmployee() – Procedure name matches the call	
BEGIN	
– Select name, birthdate, address of employees.	
SELECT Ename, Bdate, Address	
FROM EMPLOYEE;	
END	
Start time	Mon May 13 20:59:28 NPT 2024
Finish time	Mon May 13 20:59:28 NPT 2024

2. Execute the procedure in Q.1 and show the result.



```
CALL GetEmployee;
```

	Ename	Bdate	Address
1	Gaurav	2056-07-24	MacheGaun
2	Anish	2053-12-05	Budhanilkandha
3	Prabhat	2054-12-15	Nawalparasi
4	Suman	2052-07-06	Sunwal
5	Ashmita	2053-12-05	Brussels
6	Kabita	2054-12-15	Delhi
7	Kusum	2052-07-06	Sunwal
8	Pratigya	2057-11-15	Banglore
9	Abin	2051-07-09	Sunwal

3. Write a stored procedure to get the PF category name, Amount, and start date where the amount is greater than the provided input value. Your procedure should contain an IN parameter named amt to take the input value of the amount. Call the procedure with inputs 1000 and 3000 respectively.

```
CREATE PROCEDURE GetPFDetails(IN amt DECIMAL(10, 2))
BEGIN
    SELECT PFCategoryName, Amount, Start_date
    FROM PF
    WHERE Amount > amt;
END
```

Statistics 1 X

Name	Value
Updated Rows	0
CREATE PROCEDURE GetPFDetails(IN amt DECIMAL(10, 2))	
BEGIN	
SELECT PFCategoryName, Amount, Start_date	
FROM PF	
WHERE Amount > amt;	
END	
Start time	Mon May 13 21:31:42 NPT 2024
Finish time	Mon May 13 21:31:42 NPT 2024

CALL GetPFDetails(1000);

PF 1 X Statistics 1

Enter a SQL expression to filter results (use Ctrl+Space)

	PFCategoryName	Amount	Start_date
1	Employee_Provident_Fund	4,500	2079-12-05
2	Pension_Fund	5,500	2078-12-15
3	Retirement_Fund	4,700	2076-07-06
4	Children_Savings_Fund	5,800	2080-11-15
5	General_Provident_Fund	2,500	2074-07-09
6	Employee_Provident_Fund	3,000	2078-12-05
7	Pension_Fund	6,000	2077-12-15
8	Retirement_Fund	4,000	2075-07-06
9	Children_Savings_Fund	5,000	2079-11-15
10	General_Provident_Fund	3,500	2073-07-09

```

CALL GetPFDetails(3000);

```

PF 1 × Statistics 1

CALL GetPFDetails(3000) Enter a SQL expression to filter results (use Ctrl+Space)

	PFCategoryName	Amount	Start_date
1	Employee_Provident_Fund	4,500	2079-12-05
2	Pension_Fund	5,500	2078-12-15
3	Retirement_Fund	4,700	2076-07-06
4	Children_Savings_Fund	5,800	2080-11-15
5	Pension_Fund	6,000	2077-12-15
6	Retirement_Fund	4,000	2075-07-06
7	Children_Savings_Fund	5,000	2079-11-15
8	General_Provident_Fund	3,500	2073-07-09

4. Write a stored procedure to get a number of PF records where the amount of PF is equal to the provided input value. Your procedure should contain an IN parameter named amt to take the input value of the amount and should contain an OUT parameter named total to return the total number of PF records satisfying the condition.

```

CREATE PROCEDURE GetPFTotal(IN amt DECIMAL(10,2), OUT total INT)
BEGIN
    SELECT COUNT(*) INTO TOTAL FROM PF
    WHERE Amount = amt;
END

```

Statistics 1 ×

Name	Value
Updated Rows	0
Query	CREATE PROCEDURE GetPFTotal(IN amt DECIMAL(10,2), OUT total INT) BEGIN SELECT COUNT(*) INTO TOTAL FROM PF WHERE Amount = amt; END
Start time	Mon May 13 21:41:20 NPT 2024
Finish time	Mon May 13 21:41:20 NPT 2024

5. Call the procedure in Q4. with input of 3000 and print the @total.

```

CALL GetPFTotal(3000, @total)

```

Statistics 1 ×

Name	Value
Updated Rows	1
Query	CALL GetPFTotal(3000, @total)
Start time	Mon May 13 21:44:22 NPT 2024
Finish time	Mon May 13 21:44:22 NPT 2024

```

SELECT @total AS total_number_of_pf_records;

```

Results 1 ×

	total_number_of_pf_records
1	1

```

SELECT * FROM PF WHERE Amount = 3000;

```

PF 1 ×

PFID	SSN	PFC categoryName	Amount	Start_date	Remarks
1	1,111	Employee_Provident_Fund	3,000	2078-12-05	[NULL]

6. Write a before insert the trigger before inserting a record into the Employee table. Show some action on the event.

```

CREATE TRIGGER before_insert_trigger BEFORE
INSERT ON EMPLOYEE
FOR EACH ROW
BEGIN
    SET NEW.Salary = NEW.Salary * 1.3;
END

```

Statistics 1 ×

Name	Value
Updated Rows	0
Query	CREATE TRIGGER before_insert_trigger BEFORE INSERT ON EMPLOYEE FOR EACH ROW BEGIN SET NEW.Salary = NEW.Salary * 1.3; END
Start time	Mon May 13 21:52:04 NPT 2024
Finish time	Mon May 13 21:52:04 NPT 2024

To execute a trigger, we must first insert data into the EMPLOYEE table. However, the EMPLOYEE table depends on the OFFICE table, as the primary key of the OFFICE table is referenced as a foreign key in the EMPLOYEE table. Therefore, we need to perform an insert operation in the OFFICE table beforehand.

```

... • INSERT INTO Office(Onumber, Oname, Country)
VALUES
(36, 'Sunil_Office_36', 'Ireland');

Statistics 1 X
Name Value
Updated Rows 1
Query      INSERT INTO Office(Onumber, Oname, Country)
VALUES
(36, 'Sunil_Office_36', 'Ireland')
Start time Mon May 13 22:04:02 NPT 2024
Finish time Mon May 13 22:04:02 NPT 2024

```

Then we can insert into the EMPLOYEE table with the appropriate details.

```

... • INSERT INTO EMPLOYEE (SSN, Ename, Gender, Bdate, Address, Salary, Ono, Years_of_experience, Marital_Status)
VALUES(11, 'Sunil', 'M', '2055-07-10', 'Gangabu', 100000, 36, 4, 'Single');

Statistics 1 X
Name Value
Updated Rows 1
Query      INSERT INTO EMPLOYEE (SSN, Ename, Gender, Bdate, Address, Salary, Ono, Years_of_experience, Marital_Status)
VALUES(11, 'Sunil', 'M', '2055-07-10', 'Gangabu', 100000, 36, 4, 'Single')
Start time Mon May 13 22:05:17 NPT 2024
Finish time Mon May 13 22:05:17 NPT 2024

```

NOTE: We have inserted the Sunil information where his salary was inserted as 100000 but the trigger automatically gets executed before the insert operation and hiked the salary by 30% to 130000.

EMPLOYEE 1 X									
	SSN	Ename	Gender	Bdate	Address	Salary	Ono	Years_of_experience	Marital_Status
1	2	Gaurav	M	2056-07-24	MacheGaun	30,000	10	4	Married
2	3	Anish	M	2053-12-05	Budhanilkandha	95,000	2	5	Married
3	4	Prabhat	M	2054-12-15	Nawalparasi	80,000	11	6	Divorced
4	5	Suman	M	2052-07-06	Sunwal	75,000	33	7	Divorced
5	6	Ashmita	F	2053-12-05	Brussels	95,000	4	8	Divorced
6	7	Kabita	F	2054-12-15	Delhi	80,000	12	9	Single
7	8	Kusum	F	2052-07-06	Sunwal	75,000	15	10	Single
8	9	Pratigya	F	2057-11-15	Banglore	80,000	27	11	Single
9	10	Abin	M	2051-07-09	Sunwal	75,000	1	12	Single
10	11	Sunil	M	2055-07-10	Gangabu	130,000	36	13	Single

7. Write after the delete trigger on the PF table during the delete operation. Print “It is deleted”.

```
#### Create a new table named  
trigger_log to store the log message so  
that we can print the trigger message
```

The screenshot shows the MySQL Workbench interface. In the main query editor window, the following SQL code is written:

```
•CREATE TABLE trigger_log(  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    message VARCHAR(255),  
    created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP  
)
```

Below the code, the "Statistics 1" tab is selected, displaying the following information:

Name	Value
Updated Rows	0
Query	CREATE TABLE trigger_log(id INT AUTO_INCREMENT PRIMARY KEY, message VARCHAR(255), created_at TIMESTAMP DEFAULT CURRENT_TIMESTAMP)
Start time	Mon May 13 22:13:20 NPT 2024
Finish time	Mon May 13 22:13:20 NPT 2024

```
#### Create an after-delete trigger on  
the PF table
```

The screenshot shows the MySQL Workbench interface. In the main query editor window, the following SQL code is written:

```
•CREATE TRIGGER after_delete_pf_table AFTER DELETE  
ON PF  
FOR EACH ROW  
BEGIN  
    SELECT CONCAT('It is deleted from ', OLD.PFCategoryName) INTO @message;  
    •INSERT INTO trigger_log(message) VALUES(@message);  
END
```

Below the code, the "Statistics 1" tab is selected, displaying the following information:

Name	Value
Updated Rows	0
Query	CREATE TRIGGER after_delete_pf_table AFTER DELETE ON PF FOR EACH ROW BEGIN SELECT CONCAT('It is deleted from ', OLD.PFCategoryName) INTO @message; INSERT INTO trigger_log(message) VALUES(@message); END
Start time	Mon May 13 22:17:02 NPT 2024
Finish time	Mon May 13 22:17:02 NPT 2024

```
#### Delete records From the PF table to  
see whether the trigger message gets  
populated in the trigger_log table
```

The screenshot shows a MySQL Workbench interface. In the top query editor, the following SQL command is entered:

```
DELETE FROM PF WHERE PFID = 111;
```

In the bottom statistics window, the following information is displayed:

Name	Value
Updated Rows	1
Query	DELETE FROM PF WHERE PFID = 111
Start time	Mon May 13 22:20:15 NPT 2024
Finish time	Mon May 13 22:20:15 NPT 2024

```
#### Check logs in the trigger_log table
```

The screenshot shows a MySQL Workbench interface. In the top query editor, the following SQL command is entered:

```
SELECT * FROM trigger_log;
```

In the bottom results grid, the following data is shown:

id	message	created_at
1	It is deleted from Employee_Provident_Fund	2024-05-13 22:20:15

Note: Displaying a message After the Delete Trigger message is not possible in MYSQL, so creating a separate trigger_log table and creating after the delete trigger to store the message in this table.

1. Insert at least 5 tuples in each of the tables of the Yourname_Roll_COMPANY database in LAB-1

Office Table

The screenshot shows the MySQL Workbench interface. On the left, the 'Schemas' tree displays the 'Prabhat_22_COMPANY' schema with its tables: Dependents, EMPLOYEE, Office, Project, and Works_on. The 'Tables' section shows the 'Office' table with 16K rows. The central pane contains a SQL editor with the following code:

```
## lab 2
•INSERT INTO Office(Onumber, Oname, Country)
Values
(4, 'Ashmita_Office_4', 'Belgium'),
(12, 'Kabita_Office_12', 'India'),
(15, 'Kusum_Office_15', 'Nepal'),
(27, 'Pratigya_Office_27', 'India'),
(1, 'Abin_Office_1', 'Nepal');
```

Below the SQL editor is a 'Statistics 1' panel showing the execution details:

Name	Value
Updated Rows	5
Query	INSERT INTO Office(Onumber, Oname, Country) Values (4, 'Ashmita_Office_4', 'Belgium'), (12, 'Kabita_Office_12', 'India'), (15, 'Kusum_Office_15', 'Nepal'), (27, 'Pratigya_Office_27', 'India'), (1, 'Abin_Office_1', 'Nepal')
Start time	Thu Apr 18 21:31:42 NPT 2024
Finish time	Thu Apr 18 21:31:42 NPT 2024

Employee Table

The screenshot shows the MySQL Workbench interface. On the left, the 'Schemas' tree displays the 'Prabhat_22_COMPANY' schema with its tables: Dependents, EMPLOYEE, Office, Project, and Works_on. The 'Tables' section shows the 'EMPLOYEE' table with 32K rows. The central pane contains a SQL editor with the following code:

```
•INSERT INTO EMPLOYEE(Ename, Gender, Bdate, Address, Salary, Ono, Years_of_experience)
VALUES
('Ashmita', 'F', '2053-12-05', 'Brussels', 95000, 4, 5),
('Kabita', 'F', '2054-12-15', 'Delhi', 80000, 12, 4),
('Kusum', 'F', '2052-07-06', 'Sunwal', 75000, 15, 2),
('Pratigya', 'F', '2057-11-15', 'Banglore', 80000, 27, 3),
('Abin', 'M', '2051-07-09', 'Sunwal', 75000, 1, 2);
```

Below the SQL editor is a 'Statistics 1' panel showing the execution details:

Name	Value
Updated Rows	5
Query	INSERT INTO EMPLOYEE(Ename, Gender, Bdate, Address, Salary, Ono, Years_of_experience) VALUES ('Ashmita', 'F', '2053-12-05', 'Brussels', 95000, 4, 5), ('Kabita', 'F', '2054-12-15', 'Delhi', 80000, 12, 4), ('Kusum', 'F', '2052-07-06', 'Sunwal', 75000, 15, 2), ('Pratigya', 'F', '2057-11-15', 'Banglore', 80000, 27, 3), ('Abin', 'M', '2051-07-09', 'Sunwal', 75000, 1, 2)
Start time	Thu Apr 18 21:42:18 NPT 2024
Finish time	Thu Apr 18 21:42:18 NPT 2024

Project Table

```
•INSERT INTO Project(Pnumber, Pname, Proj_location, Onumber)
VALUES (8, 'Ashmita_ProjMDS', 'Belgium', 4),
(24, 'Kabita_ProjDBMS', 'Delhi', 12),
(30, 'Kusum_ProjPython', 'Nepal', 15),
(54, 'Pratigya_ProjPandas', 'Banglore', 27),
(2, 'Abin_ProjML', 'Lalitpur', 1);
```

Statistics 1 X

Name	Value
Updated Rows	5

```
INSERT INTO Project(Pnumber, Pname, Proj_location, Onumber)
VALUES (8, 'Ashmita_ProjMDS', 'Belgium', 4),
(24, 'Kabita_ProjDBMS', 'Delhi', 12),
(30, 'Kusum_ProjPython', 'Nepal', 15),
(54, 'Pratigya_ProjPandas', 'Banglore', 27),
(2, 'Abin_ProjML', 'Lalitpur', 1)
```

Start time Fri Apr 19 10:20:58 NPT 2024
Finish time Fri Apr 19 10:20:58 NPT 2024

Works_On Table

```
•INSERT INTO Works_on(ESSN, Pno)
VALUES
(6, 8),
(7, 24),
(8, 30),
(9, 54),
(10, 2);
```

Statistics 1 X

Name	Value
Updated Rows	5

```
INSERT INTO Works_on(ESSN, Pno)
VALUES
(6, 8),
(7, 24),
(8, 30),
(9, 54),
(10, 2)
```

Start time Fri Apr 19 10:22:15 NPT 2024
Finish time Fri Apr 19 10:22:15 NPT 2024

Dependents Table:

```
•INSERT INTO Dependents(Did, Dname, Dage, SSN, Drelation)
VALUES
(7, "Alisha", 8, 6, "Sister"),
(8, "Atul", 3, 7, "Brother"),
(9, "Kaushal", 72, 8, "Father"),
(10, "Jenisa", 1, 9, "Daughter"),
.... (11, "Asbina", 67, 10, "Mother");
```

The screenshot shows the execution results in the Statistics window. It displays the query executed, the number of updated rows (5), and the start and finish times of the operation.

Name	Value
Updated Rows	5
Query	INSERT INTO Dependents(Did, Dname, Dage, SSN, Drelation) VALUES (7, "Alisha", 8, 6, "Sister"), (8, "Atul", 3, 7, "Brother"), (9, "Kaushal", 72, 8, "Father"), (10, "Jenisa", 1, 9, "Daughter"), (11, "Asbina", 67, 10, "Mother")
Start time	Fri Apr 19 10:24:59 NPT 2024
Finish time	Fri Apr 19 10:24:59 NPT 2024

2. In the database Yourname_Roll_COMPANY in LAB-1, Create a table PF(PFID, SSN, PFCategoryName, Amount, Start_date, Remarks); where SSN is a foreign key referencing Employee. The start date should be of date type.

```
•CREATE TABLE PF
(
PFID INT PRIMARY KEY NOT NULL UNIQUE,
SSN INT,
PFCategoryName VARCHAR(255),
Amount DECIMAL(10, 2),
Start date DATE,
Remarks VARCHAR(255),
FOREIGN KEY (SSN) REFERENCES EMPLOYEE(SSN) ON UPDATE CASCADE ON DELETE CASCADE
);
```

The screenshot shows the creation of the PF table. The table structure is defined with columns PFID, SSN, PFCategoryName, Amount, Start date, and Remarks. A foreign key constraint is defined on the SSN column, referencing the SSN column in the EMPLOYEE table with cascade update and delete options.

Name	Value
Updated Rows	0
Query	CREATE TABLE PF (PFID INT PRIMARY KEY NOT NULL UNIQUE, SSN INT, PFCategoryName VARCHAR(255), Amount DECIMAL(10, 2), Start date DATE, Remarks VARCHAR(255), FOREIGN KEY (SSN) REFERENCES EMPLOYEE(SSN) ON UPDATE CASCADE ON DELETE CASCADE);
Start time	Thu Apr 18 22:10:13 NPT 2024
Finish time	Thu Apr 18 22:10:13 NPT 2024

3. In the database Yourname_Roll_COMPANY in LAB-1, alter the table Employee and add an attribute Marital_status of type varchar. Update the records in the table and set values of status to “Married”, “Single”, and “Divorced”. At least three records should have status married.

```
•ALTER TABLE EMPLOYEE
ADD Marital_Status VARCHAR(50)
CONSTRAINT check_marital_status CHECK (Marital_Status IN ('Married', 'Single', 'Divorced'));
```

Name	Value
Updated Rows	5

ALTER TABLE EMPLOYEE
ADD Marital_Status VARCHAR(50)
CONSTRAINT check_marital_status CHECK (Marital_Status IN ('Married', 'Single', 'Divorced'))
Start time Thu Apr 18 22:22:20 NPT 2024
Finish time Thu Apr 18 22:22:20 NPT 2024

```
•UPDATE EMPLOYEE
SET Marital_Status = "Married"
WHERE SSN IN (1, 2, 3);
```

Name	Value
Updated Rows	3

UPDATE EMPLOYEE
SET Marital_Status = "Married"
WHERE SSN IN (1, 2, 3)
Start time Thu Apr 18 22:28:41 NPT 2024
Finish time Thu Apr 18 22:28:41 NPT 2024

```
•UPDATE EMPLOYEE
SET Marital_Status = "Divorced"
WHERE SSN IN (4, 5, 6);
```

Name	Value
Updated Rows	3

UPDATE EMPLOYEE
SET Marital_Status = "Divorced"
WHERE SSN IN (4, 5, 6)
Start time Fri Apr 19 10:29:11 NPT 2024
Finish time Fri Apr 19 10:29:11 NPT 2024

```

• UPDATE EMPLOYEE
  SET Marital_status = "Single"
  WHERE SSN IN (7, 8, 9, 10);

Statistics 1 X
Name Value
Updated Rows 4
  UPDATE EMPLOYEE
  SET Marital_status = "Single"
  WHERE SSN IN (7, 8, 9, 10)
Start time Fri Apr 19 10:30:50 NPT 2024
Finish time Fri Apr 19 10:30:50 NPT 2024

```

4. Insert ten records in the table PF, where at least two records have the Remarks field NULL.

```

• INSERT INTO PF(PFID, SSN, PFCategoryName, Amount, Start_date, Remarks)
  VALUES
  (1111, 2, 'Employee_Provident_Fund', 3000, '2078-12-05', NULL),
  (2222, 2, 'Pension_Fund', 6000, '2077-12-15', NULL),
  (3333, 3, 'Retirement_Fund', 4000, '2075-07-06', NULL),
  (4444, 4, 'Children_Savings_Fund', 5000, '2079-11-15', NULL),
  (5555, 5, 'General_Provident_Fund', 3500, '2073-07-09', NULL),
  (111, 6, 'Employee_Provident_Fund', 4500, '2079-12-05', 'Regular_Monthly_Contribution'),
  (222, 7, 'Pension_Fund', 5500, '2078-12-15', 'Monthly_Contribution_For_Pension'),
  (333, 8, 'Retirement_Fund', 4700, '2076-07-06', 'Contribution_For_Retirement'),
  (444, 9, 'Children_Savings_Fund', 5800, '2080-11-15', 'Contribution_For_Children_Education'),
  (555, 10, 'General_Provident_Fund', 2500, '2074-07-09', 'Normal_Monthly_Contribution');

Statistics 1 X
Name Value
Updated Rows 10
Query INSERT INTO PF(PFID, SSN, PFCategoryName, Amount, Start_date, Remarks)
VALUES
(1111, 2, 'Employee_Provident_Fund', 3000, '2078-12-05', NULL),
(2222, 2, 'Pension_Fund', 6000, '2077-12-15', NULL),
(3333, 3, 'Retirement_Fund', 4000, '2075-07-06', NULL),
(4444, 4, 'Children_Savings_Fund', 5000, '2079-11-15', NULL),
(5555, 5, 'General_Provident_Fund', 3500, '2073-07-09', NULL),
(111, 6, 'Employee_Provident_Fund', 4500, '2079-12-05', 'Regular_Monthly_Contribution'),
(222, 7, 'Pension_Fund', 5500, '2078-12-15', 'Monthly_Contribution_For_Pension'),
(333, 8, 'Retirement_Fund', 4700, '2076-07-06', 'Contribution_For_Retirement'),
(444, 9, 'Children_Savings_Fund', 5800, '2080-11-15', 'Contribution_For_Children_Education'),
(555, 10, 'General_Provident_Fund', 2500, '2074-07-09', 'Normal_Monthly_Contribution')
Start time Fri Apr 19 10:33:59 NPT 2024
Finish time Fri Apr 19 10:33:59 NPT 2024

```

5. Select all employees.

SELECT * FROM EMPLOYEE;

	SSN	Ename	Gender	Bdate	Address	Salary	Ono	Years_of_experience	Marital_Status
1	2	Gaurav	M	2056-07-24	MacheGaun	30,000	10	1	Married
2	3	Anish	M	2053-12-05	Budhanilkandha	95,000	2	3	Married
3	4	Prabhat	M	2054-12-15	Nawalparasi	80,000	11	2	Divorced
4	5	Suman	M	2052-07-06	Sunwal	75,000	33	3	Divorced
5	6	Ashmita	F	2053-12-05	Brussels	95,000	4	5	Divorced
6	7	Kabita	F	2054-12-15	Delhi	80,000	12	4	Single
7	8	Kusum	F	2052-07-06	Sunwal	75,000	15	2	Single
8	9	Pratigya	F	2057-11-15	Banglore	80,000	27	3	Single
9	10	Abin	M	2051-07-09	Sunwal	75,000	1	2	Single

6. Select employees having a salary greater than 30000 and list the results in descending order of Ename.

```
# 6. Select employees having salary greater than 30000 and list the results in descending order of Ename.
```

```
SELECT * FROM EMPLOYEE
WHERE Salary > 30000
ORDER BY Ename DESC;
```

	SSN	Ename	Gender	Bdate	Address	Salary	Ono	Years_of_experience	Marital_Status
1	5	Suman	M	2052-07-06	Sunwal	75,000	33	3	Divorced
2	9	Pratigya	F	2057-11-15	Banglore	80,000	27	3	Single
3	4	Prabhat	M	2054-12-15	Nawalparasi	80,000	11	2	Divorced
4	8	Kusum	F	2052-07-06	Sunwal	75,000	15	2	Single
5	7	Kabita	F	2054-12-15	Delhi	80,000	12	4	Single
6	6	Ashmita	F	2053-12-05	Brussels	95,000	4	5	Divorced
7	3	Anish	M	2053-12-05	Budhanilkandha	95,000	2	3	Married
8	10	Abin	M	2051-07-09	Sunwal	75,000	1	2	Single

7. Retrieve the tuples from the project table. Sort the tuples based on Pname.

The screenshot shows a database interface with a query window and a results grid. The query is:

```
SELECT * FROM Project  
ORDER BY Pname ASC;
```

The results grid has columns: Pnumber, Pname, Proj_location, and Onumber. The data is:

Pnumber	Pname	Proj_location	Onumber
1	2 Abin_ProjML	Lalitpur	1
2	4 Anish_ProjPandas	Chitwan	2
3	8 Ashmita_ProjMDS	Belgium	4
4	20 Gaurav_ProjPython	Butwal	10
5	24 Kabita_ProjDBMS	Delhi	12
6	30 Kusum_ProjPython	Nepal	15
7	22 Prabhat_ProjDBMS	USA	11
8	44 Prabhat_ProjMDS	Kathmandu	22
9	54 Pratigya_ProjPandas	Banglore	27
10	66 Suman_ProjML	UK	33

8. Select the employees having a salary greater than 30000 and years of experience less than 3 years.

The screenshot shows a database interface with a query window and a results grid. The query is:

```
SELECT * FROM EMPLOYEE  
WHERE Salary > 30000 AND Years_of_experience < 3;
```

The results grid has columns: SSN, Ename, Gender, Bdate, Address, Salary, Ono, Years_of_experience, and Marital_Status. The data is:

SSN	Ename	Gender	Bdate	Address	Salary	Ono	Years_of_experience	Marital_Status
1	4 Prabhat	M	2054-12-15	Nawalparasi	80,000	11	2	Divorced
2	8 Kusum	F	2052-07-06	Sunwal	75,000	15	2	Single
3	10 Abin	M	2051-07-09	Sunwal	75,000	1	2	Single

9. Select the name, address, and salary of employees having a salary greater than 30000 or years of experience less than 3 years.

```

SELECT Ename, Address, Salary
FROM EMPLOYEE
WHERE Salary > 30000
OR Years_of_experience < 3;

```

EMPLOYEE 1 ×

SELECT Ename, Address, Salary FROM EMPLOYEE WHERE Salary > 30000 Enter a SQL expression to filter results

	Ename	Address	Salary
1	Gaurav	MacheGaun	30,000
2	Anish	Budhanilkandha	95,000
3	Prabhat	Nawalparasi	80,000
4	Suman	Sunwal	75,000
5	Ashmita	Brussels	95,000
6	Kabita	Delhi	80,000
7	Kusum	Sunwal	75,000
8	Pratigya	Banglore	80,000
9	Abin	Sunwal	75,000

10. Select all dependents.

```

SELECT * FROM Dependents;

```

Dependents 1 ×

SELECT * FROM Dependents Enter a SQL expression to filter results (use Ctrl+Space)

	Did	Dname	Dage	SSN	Drelation
1	1	Rihans	6	[NULL]	Brother
2	2	Kabya	8	[NULL]	Sister
3	3	Kabir	3	4	Brother
4	4	Subin	2	5	Son
5	5	Anisha	1	3	Daughter
6	6	Gauri	9	2	Sister
7	7	Alisha	8	6	Sister
8	8	Atul	3	7	Brother
9	9	Kaushal	72	8	Father
10	10	Jenisa	1	9	Daughter
11	11	Asbina	67	10	Mother

11. Select the name and age of the dependents an age between 5 to 60.

```
•SELECT Dname, Dage FROM Dependents  
WHERE Dage BETWEEN 5 AND 60;
```

Dependents 1 X

SELECT Dname, Dage FROM Dependents WHERE Dage BETWEEN Enter a SQL expression

	Dname	Dage
1	Rihans	6
2	Kabya	8
3	Gauri	9
4	Alisha	8

12. Select the offices having office names like “%Nt%” as a substring.

```
•SELECT * FROM Office  
WHERE Oname LIKE '%NT%';
```

Office 1 X

SELECT * FROM Office WHERE Oname LIKE '%NT%' Enter a SQL expression to filter results (use C

	Onumber	Oname	Country
1	11	Prabhat_Ntc_22	India

13. Select the offices having office numbers in (1, 2, 3).

```
•SELECT * FROM Office
  WHERE Onumber IN (1, 2, 3);
```

Office 1 X

SELECT * FROM Office WHERE Onumber IN (1, 2, 3) Enter a SQL expression to filter results (use Ctrl+Space)

	Onumber	Oname	Country
1	1	Abin_Office_1	Nepal
2	2	Anish_Office_2	Canada

14. Select the records from the PF table where remarks are NULL

PF 1 X

SELECT * FROM PF WHERE Remarks IS NULL Enter a SQL expression to filter results (use Ctrl+Space)

	PFID	SSN	PFCategoryName	Amount	Start_date	Remarks
1	1,111	2	Employee_Provident_Fund	3,000	2078-12-05	[NULL]
2	2,222	2	Pension_Fund	6,000	2077-12-15	[NULL]
3	3,333	3	Retirement_Fund	4,000	2075-07-06	[NULL]
4	4,444	4	Children_Savings_Fund	5,000	2079-11-15	[NULL]
5	5,555	5	General_Provident_Fund	3,500	2073-07-09	[NULL]

15. Select PF category name, amount, start date, and remarks from PF where remarks are not NULL

PF 1 X

SELECT PFCategoryName, Amount, Start_date, Remarks FROM PF WHERE Remarks IS NOT NULL Enter a SQL expression to filter results (use Ctrl+Space)

	PFCategoryName	Amount	Start_date	Remarks
1	Employee_Provident_Fund	4,500	2079-12-05	Regular_Monthly_Contribution
2	Pension_Fund	5,500	2078-12-15	Monthly_Contribution_For_Pension
3	Retirement_Fund	4,700	2076-07-06	Contribution_For_Retirement
4	Children_Savings_Fund	5,800	2080-11-15	Contribution_For_Children_Education
5	General_Provident_Fund	2,500	2074-07-09	Normal_Monthly_Contribution

16. Select the five records from the PF table using the LIMIT Clause.

The screenshot shows the MySQL Workbench interface. In the top-left pane, a SQL editor window contains the query: `SELECT * FROM PF LIMIT 5;`. In the bottom-right pane, a results grid titled "PF 1" displays the following data:

PFID	SSN	PFCategoryName	Amount	Start_date	Remarks
1	111	Employee_Provident_Fund	4,500	2079-12-05	Regular_Monthly_Contribution
2	222	Pension_Fund	5,500	2078-12-15	Monthly_Contribution_For_Pension
3	333	Retirement_Fund	4,700	2076-07-06	Contribution_For_Retirement
4	444	Children_Savings_Fund	5,800	2080-11-15	Contribution_For_Children_Education
5	555	General_Provident_Fund	2,500	2074-07-09	Normal_Monthly_Contribution

17. Select the category name of PF where the amount is not equal to 3000.

The screenshot shows the MySQL Workbench interface. In the top-left pane, a SQL editor window contains the query: `SELECT PFCategoryName FROM PF WHERE Amount != 3000;`. In the bottom-right pane, a results grid titled "PF 1" displays the following data:

PFCategoryName
Employee_Provident_Fund
Pension_Fund
Retirement_Fund
Children_Savings_Fund
General_Provident_Fund
Pension_Fund
Retirement_Fund
Children_Savings_Fund
General_Provident_Fund

18. Select all employees who work on a project no 2.

```
--> SELECT * FROM EMPLOYEE e
JOIN Works_on wo ON
e.SSN = wo.ESSN
WHERE wo.Pno = 2;
```

EMPLOYEE(+) 1 X

Enter SQL expression to filter results (use Ctrl+Space)

	SSN	Ename	Gender	Bdate	Address	Salary	Ono	Years_of_exp	Marital_Status	ESSN	Pno
1	10	Abin	M	2051-07-09	Sunwal	75,000	1	2	Single	10	2

1. Select the names of employees and their dependents without using JOIN.

The screenshot shows a database interface with a query editor and a results grid. The query in the editor is:

```
# Lab3
SELECT e.Ename, d.Dname FROM EMPLOYEE e, Dependents d
WHERE e.SSN = d.SSN;
```

The results grid displays the following data:

	Ename	Dname
1	Prabhat	Kabir
2	Suman	Subin
3	Anish	Anisha
4	Gaurav	Gauri
5	Ashmita	Alisha
6	Kabita	Atul
7	Kusum	Kaushal
8	Pratigya	Jenisa
9	Abin	Asbina

2. Select the names of employees and their dependents without using INNER JOIN and order the result based on the dependents' names.

The screenshot shows a database interface with a query editor and a results grid. The query in the editor is:

```
•SELECT e.Ename, d.Dname FROM EMPLOYEE e, Dependents d
WHERE e.SSN = d.SSN
ORDER BY d.Dname;
```

The results grid displays the following data, ordered by Dname:

	Ename	Dname
1	Ashmita	Alisha
2	Anish	Anisha
3	Abin	Asbina
4	Kabita	Atul
5	Gaurav	Gauri
6	Pratigya	Jenisa
7	Prabhat	Kabir
8	Kusum	Kaushal
9	Suman	Subin

- 3. Use JOIN between Employee, Project, and Works_on and retrieve the names of employees and the projects on which they work.**

```

•SELECT e.Ename, p.Pname FROM EMPLOYEE e JOIN Works_on w
  ON e.SSN = w.ESSN
  JOIN Project p
  ON w.Pno = p.Pnumber;
  
```

	Ename	Pname
1	Abin	Abin_ProjML
2	Anish	Anish_ProjPandas
3	Ashmita	Ashmita_ProjMDS
4	Gaurav	Gaurav_ProjPython
5	Kabita	Kabita_ProjDBMS
6	Kusum	Kusum_ProjPython
7	Prabhat	Prabhat_ProjDBMS
8	Pratigya	Pratigya_ProjPandas
9	Suman	Suman_ProjML

- 4. Use Inner join between Employee and PF table with the join condition, Employee.SSN=PF.SSN and Employee.Salary>PF.Amount**

```

SELECT * FROM EMPLOYEE e INNER JOIN PF p
WHERE e.SSN = p.SSN AND
e.Salary > p.Amount;
  
```

EMPLOYEE(+1) X

Enter a SQL expression to filter results (use Ctrl+Space)

EMPLOYEE e INNER JOIN PF p
e.SSN = p.SSN AND
e.Salary > p.Amount;

Enter a SQL expression to filter results (use Ctrl+Space)

Enter a SQL expression to filter results (use Ctrl+Space)

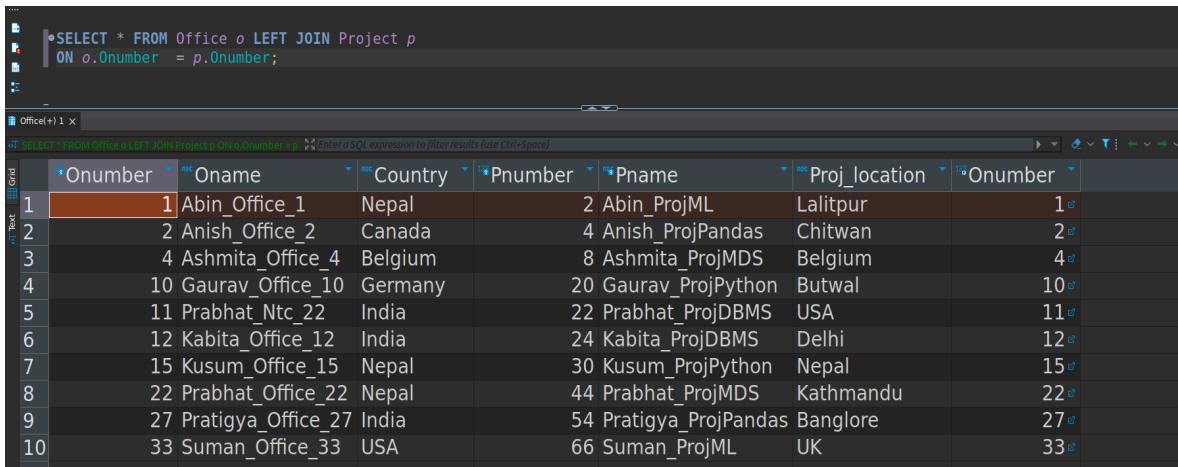
Enter a SQL expression to filter results (use Ctrl+Space)

#	SSN	Ename	Gender	Bdate	Address	Salary	Ono	Years_of_experience	Marital_Status	PFID	SSN	PFCategoryName	Amount	Start_date	Remarks
1	6	Ashmita	F	2053-12-05	Brussels	95,000	4	5	Divorced	111	6	Employee_Provident_Fund	4,500	2079-12-05	Regular_Monthly_Contribution
2	7	Kabita	F	2054-12-15	Delhi	80,000	12	4	Single	222	7	Pension_Fund	5,500	2078-12-15	Monthly_Contribution_For_Pension
3	8	Kusum	F	2052-07-06	Sunwal	75,000	15	2	Single	333	8	Retirement_Fund	4,700	2076-07-06	Contribution_For_Retirement
4	9	Pratigya	F	2057-11-15	Banglore	80,000	27	3	Single	444	9	Children_Savings_Fund	5,800	2080-11-15	Contribution_For_Children_Education
5	10	Abin	M	2051-07-09	Sunwal	75,000	1	2	Single	555	10	General_Provident_Fund	2,500	2074-07-09	Normal_Monthly_Contribution
6	2	Gaurav	M	2056-07-24	MacheGaun	30,000	10	1	Married	1,111	2	Employee_Provident_Fund	3,000	2078-12-20	[NULL]
7	2	Gaurav	M	2056-07-24	MacheGaun	30,000	10	1	Married	2,222	2	Pension_Fund	6,000	2077-12-15	[NULL]
8	3	Anish	M	2053-12-05	Budhanilkanda	95,000	2	3	Married	3,333	3	Retirement_Fund	4,000	2075-07-06	[NULL]
9	4	Prabhat	M	2054-12-15	Nawalparasi	80,000	11	2	Divorced	4,444	4	Children_Savings_Fund	5,000	2079-11-15	[NULL]
10	5	Suman	M	2052-07-06	Sunwal	75,000	33	3	Divorced	5,555	5	General_Provident_Fund	3,500	2073-07-09	[NULL]

5. Write a query to show the results of Left and Right Join between Office and Project.

Left Join:

```
SELECT * FROM Office o LEFT JOIN Project p  
ON o.Onumber = p.Onumber;
```



The screenshot shows the SQL Server Management Studio interface with a query window containing the following code:

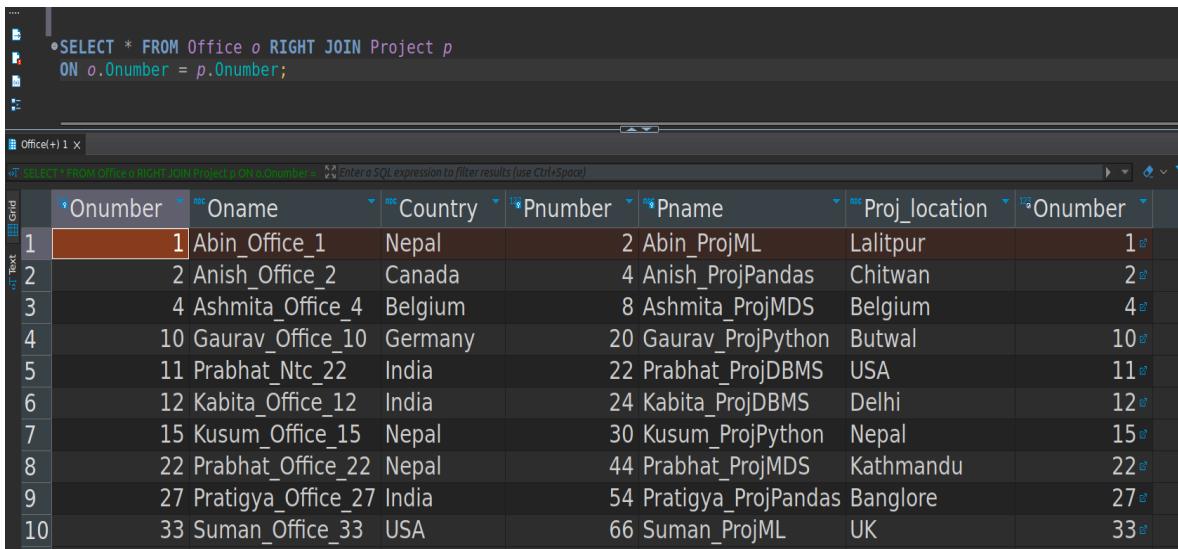
```
SELECT * FROM Office o LEFT JOIN Project p  
ON o.Onumber = p.Onumber;
```

Below the code, the results are displayed in a grid. The grid has columns: Onumber, Oname, Country, Pnumber, Pname, Proj_location, and Onumber. The data consists of 10 rows from the Office table and 10 rows from the Project table, joined on the Onumber column. The Project table rows are listed below the Office table rows in the grid.

	Onumber	Oname	Country	Pnumber	Pname	Proj_location	Onumber
1	1	Abin_Office_1	Nepal	2	Abin_ProjML	Lalitpur	1
2	2	Anish_Office_2	Canada	4	Anish_ProjPandas	Chitwan	2
3	4	Ashmita_Office_4	Belgium	8	Ashmita_ProjMDS	Belgium	4
4	10	Gaurav_Office_10	Germany	20	Gaurav_ProjPython	Butwal	10
5	11	Prabhat_Ntc_22	India	22	Prabhat_ProjDBMS	USA	11
6	12	Kabita_Office_12	India	24	Kabita_ProjDBMS	Delhi	12
7	15	Kusum_Office_15	Nepal	30	Kusum_ProjPython	Nepal	15
8	22	Prabhat_Office_22	Nepal	44	Prabhat_ProjMDS	Kathmandu	22
9	27	Pratigya_Office_27	India	54	Pratigya_ProjPandas	Banglore	27
10	33	Suman_Office_33	USA	66	Suman_ProjML	UK	33

Right Join:

```
SELECT * FROM Office o RIGHT JOIN Project p  
ON o.Onumber = p.Onumber;
```



The screenshot shows the SQL Server Management Studio interface with a query window containing the following code:

```
SELECT * FROM Office o RIGHT JOIN Project p  
ON o.Onumber = p.Onumber;
```

Below the code, the results are displayed in a grid. The grid has columns: Onumber, Oname, Country, Pnumber, Pname, Proj_location, and Onumber. The data consists of 10 rows from the Office table and 10 rows from the Project table, joined on the Onumber column. The Office table rows are listed below the Project table rows in the grid.

	Onumber	Oname	Country	Pnumber	Pname	Proj_location	Onumber
1	1	Abin_Office_1	Nepal	2	Abin_ProjML	Lalitpur	1
2	2	Anish_Office_2	Canada	4	Anish_ProjPandas	Chitwan	2
3	4	Ashmita_Office_4	Belgium	8	Ashmita_ProjMDS	Belgium	4
4	10	Gaurav_Office_10	Germany	20	Gaurav_ProjPython	Butwal	10
5	11	Prabhat_Ntc_22	India	22	Prabhat_ProjDBMS	USA	11
6	12	Kabita_Office_12	India	24	Kabita_ProjDBMS	Delhi	12
7	15	Kusum_Office_15	Nepal	30	Kusum_ProjPython	Nepal	15
8	22	Prabhat_Office_22	Nepal	44	Prabhat_ProjMDS	Kathmandu	22
9	27	Pratigya_Office_27	India	54	Pratigya_ProjPandas	Banglore	27
10	33	Suman_Office_33	USA	66	Suman_ProjML	UK	33

6. Write a query to show the results of Cross Join between Employee and PF tables.

```
SELECT * FROM EMPLOYEE e CROSS JOIN PF p  
ON e.SSN = p.SSN;
```

The screenshot shows the Oracle SQL Developer interface with two panes. The top pane displays the SQL query:

SELECT * FROM EMPLOYEE e CROSS JOIN PF p
ON e.SSN = p.SSN;

The bottom pane shows the resulting grid of data from the cross join. The columns are labeled: SSN, Ename, Gender, Bdate, Address, Salary, Ono, Years_of_experience, Marital_Status, PFID, SSN, PFCategoryName, Amount, Start_date, Remarks. The data consists of 10 rows, each pairing an employee from the EMPLOYEE table with a corresponding entry in the PF table.

	SSN	Ename	Gender	Bdate	Address	Salary	Ono	Years_of_experience	Marital_Status	PFID	SSN	PFCategoryName	Amount	Start_date	Remarks
1	2	Gaurav	M	2056-07-24	MacheGaun	30,000	10	1	Married	1,111	2	Employee_Provident_Fund	3,000	2078-12-05	(NULL)
2	2	Gaurav	M	2056-07-24	MacheGaun	30,000	10	1	Married	2,222	2	Pension_Fund	6,000	2077-12-15	(NULL)
3	3	Anish	M	2053-12-05	Budhanilkandha	95,000	2	3	Married	3,333	3	Retirement_Fund	4,000	2075-07-06	(NULL)
4	4	Prabhat	M	2054-12-15	Nawalparsi	80,000	11	2	Divorced	4,444	4	Children_Savings_Fund	5,000	2079-11-15	(NULL)
5	5	Suman	M	2052-07-06	Sunwal	75,000	33	3	Divorced	5,555	5	General_Provident_Fund	3,500	2073-07-09	(NULL)
6	6	Ashmita	F	2053-12-05	Brussels	95,000	4	5	Divorced	111	6	Employee_Provident_Fund	4,500	2079-12-05	Regular_Monthly_Contribution
7	7	Kabita	F	2054-12-15	Delhi	80,000	12	4	Single	222	7	Pension_Fund	5,500	2078-12-15	Monthly_Contribution_For_Pension
8	8	Kusum	F	2052-07-06	Sunwal	75,000	15	2	Single	333	8	Retirement_Fund	4,700	2076-07-06	Contribution_For_Retirement
9	9	Pratigya	F	2057-11-15	Banglore	80,000	27	3	Single	444	9	Children_Savings_Fund	5,800	2080-11-15	Contribution_For_Children_Education
10	10	Abin	M	2051-07-09	Sunwal	75,000	1	2	Single	555	10	General_Provident_Fund	2,500	2074-07-09	Normal_Monthly_Contribution

7. Show results of using natural join between Employee and PF.

```
SELECT * FROM EMPLOYEE e NATURAL JOIN PF;
```

SELECT * FROM EMPLOYEE e NATURAL JOIN PF;

SSN	Ename	Gender	Bdate	Address	Salary	Ono	Years_of_experience	Marital_Status	PFID	PFCategoryName	Amount	Start_date	Remarks
2	Gaurav	M	2056-07-24	MacheGaun	30,000	10	10	Married	1,111	Employee_Provident_Fund	3,000	2078-12-05	[NULL]
2	Gaurav	M	2056-07-24	MacheGaun	30,000	10	10	Married	2,222	Pension_Fund	6,000	2077-12-15	[NULL]
3	Anish	M	2053-12-05	Budhanikandha	95,000	2	2	Married	3,333	Retirement_Fund	4,000	2075-07-06	[NULL]
4	Prabhat	M	2054-12-15	Nawalparasi	80,000	11	11	Divorced	4,444	Children_Savings_Fund	5,000	2079-11-15	[NULL]
5	Suman	M	2052-07-06	Sunwal	75,000	33	33	Divorced	5,555	General_Provident_Fund	3,500	2073-07-09	[NULL]
6	Ashmita	F	2053-12-05	Brussels	95,000	4	4	Divorced	111	Employee_Provident_Fund	4,500	2079-12-05	Regular_Monthly_Contribution
7	Kabita	F	2054-12-15	Delhi	80,000	12	12	Single	222	Pension_Fund	5,500	2078-12-15	Monthly_Contribution_For_Pension
8	Kusum	F	2052-07-06	Sunwal	75,000	15	15	Single	333	Retirement_Fund	4,700	2076-07-06	Contribution_For_Retirement
9	Pratigya	F	2057-11-15	Banglore	80,000	27	27	Single	444	Children_Savings_Fund	5,800	2080-11-15	Contribution_For_Children_Education
10	Abin	M	2051-07-09	Sunwal	75,000	1	1	Single	555	General_Provident_Fund	2,500	2074-07-09	Normal_Monthly_Contribution

8. Find the number of employees and status in each status of “Married”, “Single”, or “Divorced”. Use the COUNT function with the GROUP BY clause with status.

EMPLOYEE 1 ×

```
•SELECT COUNT(*) as number_of_employees, Marital_Status  FROM  EMPLOYEE
GROUP BY Marital_Status;
```

SELECT COUNT(*) as number_of_employees, Marital_Status FROM EMPLOYEE

	number_of_employees	Marital_Status
1	2	Married
2		Divorced
3		Single

9. Find the number of employees and status in each status of “Married” OR “Single”. Use the COUNT function with the GROUP BY clause with status and Having clause with status= “Married” OR “Single”

EMPLOYEE 1 ×

```
•SELECT COUNT(*) as number_of_employees, Marital_Status  FROM EMPLOYEE e
GROUP BY Marital_Status HAVING
Marital_Status IN ('Married', 'Single');
```

SELECT COUNT(*) as number_of_employees, Marital_Status FROM EMPLOYEE

	number_of_employees	Marital_Status
1	2	Married
2		Single

10. Using sub query, select the name and location of projects whose Onumber is in the Onumber of the offices located in countries Nepal and India.

The screenshot shows a database interface with a SQL editor and a results grid. The SQL query is:

```
•SELECT Pname, Proj_location  
FROM Project p  
WHERE p.Onumber IN (SELECT o.Onumber FROM Office o WHERE o.Country IN ('Nepal', 'India'));
```

The results grid displays the following data:

	Pname	Proj_location
1	Abin_ProjML	Lalitpur
2	Prabhat_ProjDBMS	USA
3	Kabita_ProjDBMS	Delhi
4	Kusum_ProjPython	Nepal
5	Prabhat_ProjMDS	Kathmandu
6	Pratigya_ProjPandas	Banglore

LAB 4 SET OPERATIONS & AGGREGATE FUNCTIONS

Prepare Lab Sheet of MYSQL Statements for the following.

1. Create tables Teacher (Id INT PRIMARY KEY, Tname VARCHAR(20)) and Student (id INT PRIMARY KEY, Sname VARCHAR(20));

Teacher Table:

The screenshot shows the MySQL Workbench interface with a query editor window. The SQL statement to create the Teacher table is displayed:

```
CREATE TABLE Teacher (
    Id INT PRIMARY KEY,
    Tname VARCHAR(20)
);
```

Below the SQL statement, the "Statistics 1" tab is open, showing the following information:

Name	Value
Updated Rows	0
CREATE TABLE Teacher (
Id INT PRIMARY KEY,	
Tname VARCHAR(20)	
)	
Start time	Thu May 09 04:59:31 NPT 2024
Finish time	Thu May 09 04:59:31 NPT 2024

Student Table:

The screenshot shows the MySQL Workbench interface with a query editor window. The SQL statement to create the Student table is displayed:

```
CREATE TABLE Student(
    id INT PRIMARY KEY,
    Sname VARCHAR(20)
)
```

Below the SQL statement, the "Statistics 1" tab is open, showing the following information:

Name	Value
Updated Rows	0
CREATE TABLE Student(
id INT PRIMARY KEY,	
Sname VARCHAR(20)	
)	
Start time	Thu May 09 05:02:46 NPT 2024
Finish time	Thu May 09 05:02:46 NPT 2024

2. Insert values like {(1, "Ram"), (2, "Hari"), (3, "Sita")} in Teacher and {"2, "Hari"}, (3, "Sita"), (4, "Gita")} in Student.

```

    • INSERT INTO Teacher(Id, Tname) VALUES
      (1, "Ram"),
      (2, "Hari"),
      (3, "Sita");
  
```

Statistics 1 X	
Name	Value
Updated Rows	3
INSERT INTO Teacher(Id, Tname) VALUES (1, "Ram"), (2, "Hari"), (3, "Sita")	
Start time	Thu May 09 05:10:11 NPT 2024
Finish time	Thu May 09 05:10:11 NPT 2024

```

    • INSERT INTO Student(id, Sname) VALUES
      (2, "Hari"),
      (3, "Sita"),
      (4, "Gita");
  
```

Statistics 1 X	
Name	Value
Updated Rows	3
INSERT INTO Student(id, Sname) VALUES (2, "Hari"), (3, "Sita"), (4, "Gita")	
Start time	Thu May 09 05:12:03 NPT 2024
Finish time	Thu May 09 05:12:03 NPT 2024

3. Write a query to find the Union of “Teacher” and “Student” tables.

```

    ... • SELECT Id AS ID, Tname AS Name FROM Teacher
        UNION
        SELECT id AS ID, Sname AS Name FROM Student;
  
```

Results 1 X		
SELECT Id AS ID, Tname AS Name FROM Teacher UNION SELECT id AS ID, Sname AS Name FROM Student; Enter a SQL expression to filter results		
Grid	ID	Name
1	1	Ram
2	2	Hari
3	3	Sita
4	4	Gita

```
•SELECT Id AS ID, Tname AS Name FROM Teacher  
INTERSECT  
SELECT id AS ID, Sname AS Name FROM Student;
```

ID	Name
1	Hari
2	Sita

4. Write a query to find the Intersection of Teacher and Student.

5. Write a query to find the intersection of the names Teacher and Student using Distinct and Inner Join.

```
•SELECT DISTINCT Teacher.Tname AS Name  
FROM Teacher  
INNER JOIN Student ON Teacher.Tname = Student.Sname;
```

ID	Name
1	Hari
2	Sita

6. Write a query to find the intersection of names Teacher and Student using IN and Sub query

```
•SELECT Tname FROM Teacher WHERE  
TName IN (  
SELECT SName FROM Student  
) ;
```

Tname
Hari
Sita

7. Write a query to find Teacher MINUS Student using Left Join

```
SELECT Teacher.* FROM Teacher LEFT JOIN Student  
ON Teacher.Id = Student.id WHERE Student.id IS NULL;
```

Teacher 1 X

SELECT Teacher.* FROM Teacher LEFT JOIN Student ON Teacher.

	Id	Tname
1	1	Ram

Enter a SQL expression to filter results (use Ctrl+Space)

8. Find the number of offices in the Office table from the COMPANY Database in Lab-1 using the COUNT function.

```
SELECT COUNT(Oname) AS number_of_offices FROM Office;
```

Results 1 X

SELECT COUNT(Oname) FROM Office

COUNT(Oname)
10

Enter a SQL expression to filter results (use Ctrl+Space)

9. Write a query to count the distinct names of Employees.

```
SELECT DISTINCT COUNT(Ename) AS distinct_names_count FROM EMPLOYEE e;
```

Results 1 X

SELECT DISTINCT COUNT(Ename) AS distinct_names_count FROM EMPLOYEE e

distinct_names_count
9

Enter a SQL expression to filter results (use Ctrl+Space)

10. Write a query to find the sum of the salary of Employees.

```
| SELECT SUM(Salary) AS salary_sum FROM EMPLOYEE;
```

Results 1 X

```
T SELECT SUM(Salary) AS salary_sum FROM EMPLOYEE | Enter a SQL expression to filter results (use Ctrl+Space)
```

	salary_sum
1	685,000

11. Write a query to find the average salary of Employees.

```
| SELECT AVG(Salary) AS average_salary FROM EMPLOYEE;
```

Results 1 X

```
T SELECT AVG(Salary) AS average_salary FROM EMPLOYEE | Enter a SQL expression to filter results (use Ctrl+Space)
```

	average_salary
1	76,111.111111

12. Write a query to find the Maximum PF Amount from the PF Table.

```
| SELECT MAX(Amount) AS maximum_pf_amount FROM PF;
```

Results 1 X

```
T SELECT MAX(Amount) AS maximum_pf_amount FROM PF | Enter a SQL expression to filter results (use Ctrl+Space)
```

	maximum_pf_amount
1	6,000

13. Write a query to find the Minimum PF Amount from the PF Table.

```
SELECT MIN(Amount) AS minimum_pf_amount FROM PF;
```

Results 1 X

SELECT MIN(Amount) AS minimum_pf_amount FROM PF Enter a SQL expression to filter results (use C

	minimum_pf_amount
1	2,500