Database Assignment 1

Question No. 01

Create tables

a. Make a student table
b. Make a Library table
c. Make a Fees table

Create tables with proper relations.

Answer No. 01

a.

```
CREATE TABLE Students (
   StudentID INT PRIMARY KEY,
   StudentName VARCHAR(255),
   Age INT,
   Father VARCHAR(255),
   Mother VARCHAR(255),
   DateOfBirth DATE
);
```

b.

```
CREATE TABLE Library (
   ItemID INT PRIMARY KEY,
   BookName VARCHAR(255),
   Author VARCHAR(255),
   PublicationYear YEAR
);
```

c.

```
CREATE TABLE Fees (
   FeeID INT PRIMARY KEY,
   StudentID INT,
   Amount DECIMAL(10, 2),
   DueDate DATE,
   Status VARCHAR(50),
   FOREIGN KEY (StudentID) REFERENCES Students(StudentID)
);
```

Proper Relation ::

```
CREATE TABLE BorrowedBooks (
    BorrowID INT PRIMARY KEY,
    StudentID INT,
    ItemID INT,
    CheckoutDate DATE,
    ReturnDate DATE,
    FOREIGN KEY (StudentID) REFERENCES Students(StudentID),
    FOREIGN KEY (ItemID) REFERENCES Library(ItemID)
);
```

Question No. 02

Add proper constraints with the No 1 question

Answer No. 02

Constraints Name ::

Student table ::
StudentID > PRIMARY KEY: Ensures uniqueness and non-nullability.

Library Table ::
ItemID > PRIMARY KEY: Ensures uniqueness and non-nullability.

Fees Table ::
FeeID > PRIMARY KEY: Ensures uniqueness and non-nullability.
StudentID > FOREIGN KEY: References StudentID in the Students table, ensuring referential integrity. It implies that it cannot be NULL since it must match an existing StudentID.
Amount > Not NULL
Status > Not NULL

BorrowedBooks Table ::
BorrowID > PRIMARY KEY: Ensures uniqueness and non-nullability.

StudentID > FOREIGN KEY: References StudentID in the Students table. This implies non-nullability to maintain referential integrity.
ItemID > FOREIGN KEY: References ItemID in the Library table. This also implies non-nullability for referential integrity.
CheckoutDate > Not Null
ReturnDate > Null

Question No. 03

Write the differences between data and information

Answer No. 03

Data is an individual unit that contains raw materials which do not carry any specific meaning. example : **(Mehedi Hasan**) ,(**12**), (**Barisal**) -- Individually unit data

Information is a group of data that collectively carries a logical meaning. Data doesn't depend on information. Information depends on data. example : **(Mehedi Hasan,12,Barisal)** -- group all information

Question No. 04

In MySQL, Update and Delete query wasn't executing, what was the reason and how to run those query? Write the code to enable the feature. (If you followed the class, you should know this).

Answer the following questions with this table data. Table name Employee.

```
+-----------+------------+--------+--------+------------+
| EmployeeID | FirstName   | LastName | Age     | Department |
+-----------+------------+--------+--------+------------+
| 1          | John        | Doe     | 28      | Sales       |
| 2          | Jane        | Smith   | 32      | Marketing   |
| 3          | Michael     | Johnson| 35      | Finance     |
| 4          | Sarah       | Brown   | 30      | HR          |
| 5          | William     | Davis   | 25      | Engineering|
| 6          | Emily       | Wilson  | 28      | Sales       |
| 7          | Robert      | Lee     | 33      | Marketing   |
| 8          | Laura       | Hall    | 29      | Finance     |
| 9          | Thomas      | White   | 31      | HR          |
| 10         | Olivia      | Clark   | 27      | Engineering|
+-----------+------------+--------+--------+------------+
```

Answer No. 04

Start

SET SQL_SAFE_UPDATES= 0;

End

SET SQL_SAFE_UPDATES= 1;

Question No. 05

Write a query to show the distinct department names

Answer No. 05

SELECT DISTINCT Department FROM Employee;

Question No. 06

Write a query to show the LastNames of the employees sorted by descending ages

Answer No. 06

```
SELECT LastName FROM Employee ORDER BY age DESC;
```

Question No. 07

Write a query to show the employee LastName whose age is greater than 30 and works in the Marketing department.

Answer No. 07

```
SELECT LastName FROM Employee WHERE age > 30 AND Department = 'Marketing';
```

Question No. 08

Write a query to select all the employees

Answer No. 08

```
SELECT * FROM Employee;
```

Question No. 09

Write a query to get employees whose names includes ‘son’

Answer No. 09

```
SELECT * FROM Employee WHERE name LIKE '%son%';
```

Question No. 10

Write a query to get the engineers

Answer No. 10

```
SELECT * FROM Employee WHERE department = 'Engineering';
```