Database: Assignment-01

Answer- 01(a, b, c)

```
CREATE DATABASE assignment;
USE assignment;

-- a) Student Table:

CREATE TABLE Student (
        Name VARCHAR(20),
        Id INT PRIMARY KEY,
        Department VARCHAR(20),
        Section VARCHAR(20)
);

-- b) Library Table:

CREATE TABLE Library (
        BookName VARCHAR(20) PRIMARY KEY,
        IdWhoHired INT,
        FOREIGN KEY(IdWhoHired) REFERENCES Student(Id)
);

-- b) Fees Table:

CREATE TABLE Fees (
        AccountNo INT PRIMARY KEY,
        IdWhoPaid INT,
        PaidAmount FLOAT,
        FOREIGN KEY(IdWhoPaid) REFERENCES Student(Id)
);
```

Answer - 02

-- a) Student Table with Constraint:

```
CREATE TABLE Student (
        Name VARCHAR(20) NOT NULL,
        Id INT,
        Department VARCHAR(20) NOT NULL,
        Section VARCHAR(20) NOT NULL,
        CONSTRAINT id_rule PRIMARY KEY(Id)
);
```

-- b) Library Table with Constraint:

```
CREATE TABLE Library (
        BookName VARCHAR(20),
        IdWhoHired INT,
        CONSTRAINT book_name PRIMARY KEY(BookName),
        CONSTRAINT id_rule FOREIGN KEY(IdWhoHired)
REFERENCES Student(Id)
);
```

-- a) Fees Table with Constraint:

```
CREATE TABLE Fees (
        AccountNo INT,
        IdWhoPaid INT,
        PaidAmount FLOAT,
        CONSTRAINT acc_no PRIMARY KEY(AccountNo),
        CONSTRAINT who_hired FOREIGN KEY(IdWhoPaid)
REFERENCES Student(Id)
);
```

Answer - 03

Difference between **Data and Information**: Data is basic units of meaning which can be either organised or unorganised, when it comes in an organised way then it provides information.

Answer - 04

When we need to update or delete any information then we need to turn off sql_safe_updates otherwise you can't modify any data. Given below the code to enable this feature:

SET SQL_SAFE_UPDATES = 0;

When your modification is completed then you need to turn it on again otherwise it can be a reason for security issues. To turn on it again you need to put **1** instead of **0**.

Answer - 05

-- to show the distinct department names:

```
SELECT DISTINCT Department
FROM Employee;
```

Answer - 06

-- to show the LastNames of the employees sorted by descending
-- ages:

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

Answer - 07

-- to show the employee LastName whose age is greater than 30
-- and works in the Marketing department:

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

Answer - 08

-- to select all the employees:

```
SELECT *
FROM Employee;
```

Answer - 09

-- to get employees whose names includes ‘son’:

```
SELECT *
FROM Employee
WHERE FirstName LIKE '%son%' OR LastName LIKE '%son%';
```

Answer - 10

-- to get the engineers:

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