**Answer Script**

| **Question No. 01** |
| --- |
| Create tables   * 1. Make a student table   2. Make a Library table   3. Make a Fees table   Create table with proper relations. |
| **Answer No. 01** |
| **-- TABLE A**  **CREATE TABLE STUDENT(**  **Roll CHAR(4) PRIMARY KEY,**  **Name VARCHAR(50) NOT NULL,**  **Email VARCHAR(100) UNIQUE,**  **Adress VARCHAR(250),**  **Age int CHECK(Age > 10)**  **);**  **-- TABLE B**  **CREATE TABLE LIBRARY(**  **Library\_id CHAR(4) PRIMARY KEY,**  **Books VARCHAR(100) UNIQUE,**  **Student\_Roll CHAR(4) UNIQUE,**  **FOREIGN KEY(Student\_Roll) REFERENCES STUDENT(Roll)**  **);**  **-- TABLE C**  **CREATE TABLE FEES(**  **Fees\_id CHAR(4) PRIMARY KEY,**  **Students\_Roll CHAR(4) UNIQUE,**  **Monthly\_Fees FLOAT ,**  **Books\_Fees FLOAT,**  **Library\_fine FLOAT,**  **Total\_Fees FLOAT NOT NULL,**  **FOREIGN KEY(Students\_Roll) REFERENCES STUDENT(Roll)**  **);** |

| **Question No. 02** |
| --- |
| Add proper constraints with the No 1 question |
| **Answer No. 02** |
| **-- TABLE A**  **CREATE TABLE STUDENT(**  **Roll CHAR(4) PRIMARY KEY,**  **Name VARCHAR(50) NOT NULL,**  **Email VARCHAR(100) UNIQUE,**  **Adress VARCHAR(250),**  **Age int CHECK(Age > 10)**  **);**  **-- TABLE B**  **CREATE TABLE LIBRARY(**  **Library\_id CHAR(4) PRIMARY KEY,**  **Books VARCHAR(100) UNIQUE,**  **Student\_Roll CHAR(4) UNIQUE,**  **FOREIGN KEY(Student\_Roll) REFERENCES STUDENT(Roll)**  **);**  **-- TABLE C**  **CREATE TABLE FEES(**  **Fees\_id CHAR(4) PRIMARY KEY,**  **Students\_Roll CHAR(4) UNIQUE,**  **Monthly\_Fees FLOAT ,**  **Books\_Fees FLOAT,**  **Library\_fine FLOAT,**  **Total\_Fees FLOAT NOT NULL,**  **FOREIGN KEY(Students\_Roll) REFERENCES STUDENT(Roll)**  **);** |

| **Question No. 03** |
| --- |
| Write the differences between data and information |
| **Answer No. 03** |
| **Data is raw material that has not been optimized,organized and not arranged. On the other hand, information is organized and arranged. Data doesn’t depend on information, but information depend on data.** |

| **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. |
| **Answer No. 04** |
| **For this issu main reason is security. But we able to solve this issu and can enable the feature by following some operation.**  **Here is the code :**  **SET SQL\_SAFE\_UPDATES = 0;**  **-- Update and Delete query**  **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 EMLOYEE**  **ORDER BY Age DESC;** |

| **Question No. 07** |
| --- |
| Write a query to show the employee LastName whose age is greater than 30 and works in 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 FastName LIKE '%son%' OR LastName LIKE '%son%';** |

| **Question No. 10** |
| --- |
| Write a query to get the engineers |
| **Answer No. 10** |
| **SELECT \***  **FROM EMPLOYEE**  **WHERE DEPARTMENT = 'Engineering';** |