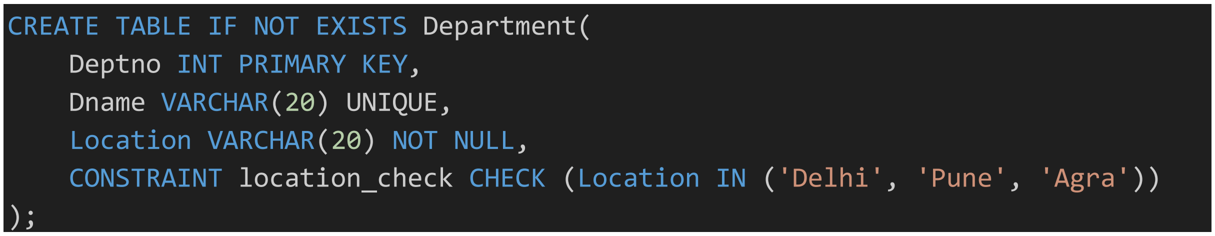
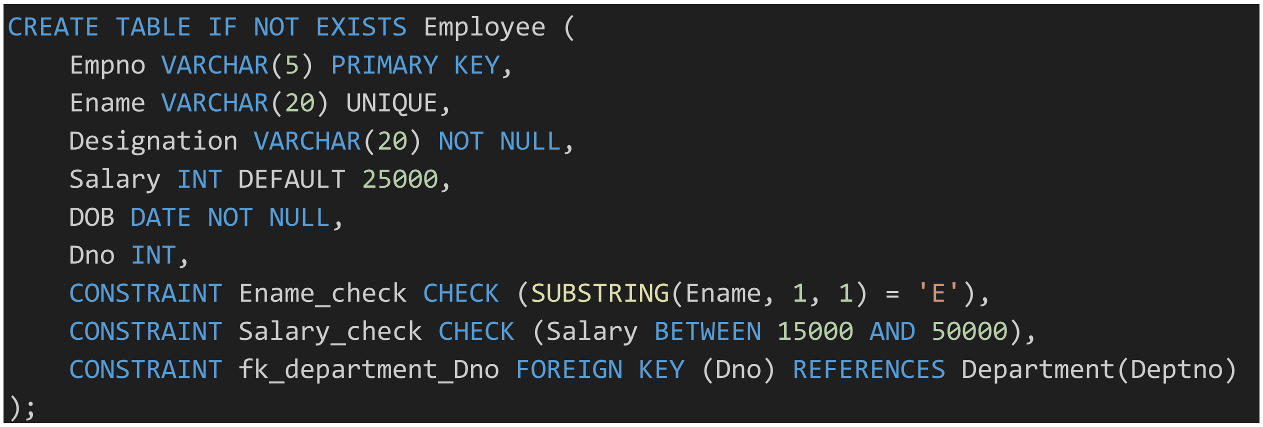
Practical Assignment – 2

- Name: Hardik Arora
- Branch : Btech - CS
- Program : AIML
- University Roll No. : 2215500071
- Section : 2AC
- Class Roll No. : 28

1. Create the following tables and specify constraints at the time of creation:



```sql
CREATE TABLE IF NOT EXISTS Department(
    Deptno INT PRIMARY KEY,
    Dname VARCHAR(20) UNIQUE,
    Location VARCHAR(20) NOT NULL,
    CONSTRAINT location_check CHECK (Location IN ('Delhi', 'Pune', 'Agra'))
);
```



```sql
CREATE TABLE IF NOT EXISTS Employee (
    Empno VARCHAR(5) PRIMARY KEY,
    Ename VARCHAR(20) UNIQUE,
    Designation VARCHAR(20) NOT NULL,
    Salary INT DEFAULT 25000,
    DOB DATE NOT NULL,
    Dno INT,
    CONSTRAINT Ename_check CHECK (SUBSTRING(Ename, 1, 1) = 'E'),
    CONSTRAINT Salary_check CHECK (Salary BETWEEN 15000 AND 50000),
    CONSTRAINT fk_department_Dno FOREIGN KEY (Dno) REFERENCES Department(Deptno)
);
```

```sql
CREATE TABLE IF NOT EXISTS Candidate(
    Candidate_ID INT PRIMARY KEY,
    Candidate_Name VARCHAR(20) NOT NULL,
    Candidate_Email VARCHAR(20) UNIQUE,
    Candidate_Dept VARCHAR(20) DEFAULT 'HR',
    Manager_Id INT,
    CONSTRAINT email_format_check CHECK (
        POSITION('@' IN Candidate_Email) > 0 AND
        POSITION('.' IN Candidate_Email) > POSITION('@' IN Candidate_Email)
    ),
    CONSTRAINT fk_manager_candidate FOREIGN KEY (Manager_ID) REFERENCES
Candidate(Candidate_ID)
);
```

2. Create the schemas as specified above without specifying any constraints.

```sql
CREATE TABLE IF NOT EXISTS College(
    cName VARCHAR(10),
    state VARCHAR(10),
    enrollment INT
);
```

```sql
CREATE TABLE IF NOT EXISTS Student(
    sID INT,
    sName VARCHAR(10),
    GPA FLOAT,
    sizeHS INT
);
```

```sql

CREATE TABLE IF NOT EXISTS Apply(
    sID INT,
    cName VARCHAR(10),
    major VARCHAR(20)
);
```

Q1. Add cName as Primary key in College.

```
ALTER TABLE College ADD PRIMARY KEY (cName);
```

Q2. Add sID as Primary Key in student.

```
ALTER TABLE Student ADD PRIMARY KEY (sID);
```

Q3. Add sID, cName, major as Primary Key in Apply.

```
ALTER TABLE Apply ADD PRIMARY KEY (sID, cName, major);
```

Q4. Make sID in Apply foreign key referring table student and cName reffering table college.

```
ALTER TABLE Apply ADD CONSTRAINT fk_student FOREIGN KEY (sID) REFERENCES
student(sID);

ALTER TABLE Apply ADD CONSTRAINT fk_college FOREIGN KEY (cName) REFERENCES
college(cName);
```

Q5. Increase data type size of major from 20 to 25.

```
ALTER TABLE Apply MODIFY COLUMN major VARCHAR(25);
```

Q6. Add a new column decision in the Apply table keeping a constraint of not null for this column with data type varchar(3).

```
ALTER TABLE Apply ADD COLUMN decision VARCHAR(3) NOT NULL;
```

Q7. Change data type of decision in Apply to char(1).

```
ALTER TABLE Apply MODIFY COLUMN decision CHAR(1);
```

Q8. Drop foreign key on column name cName from Apply table.

```
ALTER TABLE Apply DROP FOREIGN KEY fk_college;
```

Q9. Remove Column sizeHS from Student table.

```
ALTER TABLE Student DROP COLUMN sizeHS;
```

Q10. Drop primary key from college.

```
ALTER TABLE College DROP PRIMARY KEY;
```

Q11. Make cName, major unique pairwise such as Stanford CS, Stanford EE.

```
ALTER TABLE Apply ADD CONSTRAINT uc_cName_major UNIQUE (cName, major);
```

Q12. Add cName as Foreign Key in Apply table referring table College using on delete cascade.

```
ALTER TABLE College ADD INDEX idx_cName (cName);
ALTER TABLE Apply ADD CONSTRAINT fk_college FOREIGN KEY (cName) REFERENCES
College(cName) ON DELETE CASCADE;
```

Q13. Rename Column enrollment to enroll in college table.

```
ALTER TABLE Student CHANGE COLUMN enrollment enroll VARCHAR(255);
```

Exercise:

Customer:

```
CREATE TABLE IF NOT EXISTS CUSTOMER(
    CustomerId VARCHAR(6) PRIMARY KEY,
    CustomerName VARCHAR(30) NOT NULL,
    DateOfReg DATE,
    UserId VARCHAR(15) UNIQUE,
    Password VARCHAR(15) NOT NULL,
    CONSTRAINT  CustomerId_check CHECK (SUBSTRING(CustomerId, 1, 1) = 'C')

);
```

BankInfo:

```
CREATE TABLE IF NOT EXISTS BankInfo(
    AccountNo INT,
    CustomerId VARCHAR(6),
    CONSTRAINT PK_BankInfo PRIMARY KEY (AccountNo, CustomerId),
    CONSTRAINT CustomerId_fk FOREIGN KEY(CustomerId) REFERENCES
Customer(CustomerId) ON DELETE CASCADE
);
```

Billing:

```
CREATE TABLE IF NOT EXISTS Billing(
    Billd INT PRIMARY KEY,
    AccountNo INT,
    CustomerId VARCHAR(6),
    BillDate DATE DEFAULT CURRENT_DATE,
    PaymentType ENUM('creditcard', 'debitcard'),
    CONSTRAINT FK_AccountNo_CustomerId FOREIGN KEY(AccountNo, CustomerId)
REFERENCES BankInfo(AccountNo, CustomerId) ON DELETE CASCADE
);
INSERT INTO Billing (Billd, AccountNo, CustomerId, PaymentType) VALUES (1, 12345,
'ABCDEF', 'creditcard');
```

Item:

```
CREATE TABLE IF NOT EXISTS Item(
    ItemId VARCHAR(6) PRIMARY KEY,
    ItemName VARCHAR(30) NOT NULL,
    QtyOnHand INT CHECK (QtyOnHand > 0),
    UnitPrice INT CHECK (UnitPrice > 0),
    Class CHAR(1),
    UnitOfMeasurement VARCHAR(12),
    ReOrderLevel INT CHECK (ReOrderLevel > 0),
    ReOrderQty INT CHECK (ReOrderQty > 0),
    Discount INT,
    CONSTRAINT QtyOnHand_check CHECK(QtyOnHand > ReOrderLevel),
    CONSTRAINT Class_check CHECK(
        (Class = 'A' AND UnitPrice < 100) OR
        (Class = 'B' AND UnitPrice < 1000 AND UnitPrice >= 100) OR
        (Class = 'C' AND UnitPrice >= 1000)
    )
);
```

- Submitted By: Hardik Arora
- Branch :   Btech - CS
- Program : AIML
- University Roll No. : 2215500071
- Section : 2AC
- Class Roll No. : 28
- Submitted to: Ayushi Mam