

Electricity Grid Database Management System DDL Scripts

202303005 : Hemil Patel
202303015 : Akshat Shah
202303001 : Preet Siddhapura

1 Creating Schema

We will name our Schema **Electricity_Grid_Management**. For that we will Execute Script

```
CREATE SCHEMA Electricity_Grid_Management;
```

2 Creating Tables

2.1 Setting Search Path

First we will set our Search Path to our Schema so that we won't have to write **Electricity_Grid_Management** every time we create a Table. For that we will Execute Script

```
SET SEARCH_PATH TO Electricity_Grid_Management;
```

2.2 Creating tables

Now we will Create tables of our Database.

1. First we will Create **Customer_Type** Table. For that we will Execute Script

```
CREATE TABLE Customer_Type (  
    Customer_Type_ID INT Primary Key,  
    Type_Name VARCHAR(100) NOT NULL UNIQUE,  
    Description VARCHAR(200)  
);
```

2. Now we will Create **Pin_Code** Table. For that we will Execute Script

```
CREATE TABLE Pin_Code(  
    Pin_Code CHAR(6) PRIMARY KEY CHECK (Pin_Code ~ '^[0-9]{6}'),  
    City VARCHAR(50),  
    District VARCHAR(50),  
    State VARCHAR(50)  
);
```

3. Now we will Create **Customer** Table. For that we will Execute Script

```
CREATE TABLE Customer (  
    Customer_ID VARCHAR(12) PRIMARY KEY,  
    Customer_Name VARCHAR(200) NOT NULL,  
    Phone_Number VARCHAR(10) CHECK (Phone_Number ~ '^[0-9]{10}'),  
    UNIQUE,  
    Block_Flat_No VARCHAR(50),  
    Street VARCHAR(200),  
    Billing_Cycle INT CHECK (Billing_Cycle > 0),  
    Connection_Status VARCHAR(50) CHECK (Connection_Status IN ('Connected',  
'Disconnected')),  
    Customer_Type_ID INT NOT NULL,  
    Pin_Code CHAR(6) CHECK (Pin_Code ~ '^[0-9]{6}'),  
    FOREIGN KEY (Pin_Code) REFERENCES Pin_Code(Pin_Code) ON DELETE  
    RESTRICT ON UPDATE CASCADE,  
    FOREIGN KEY (Customer_Type_ID) REFERENCES Customer_Type(Customer_Type_ID)  
    ON DELETE RESTRICT,  
    CONSTRAINT unique_address UNIQUE (Block_Flat_No, Street, Pin_Code)  
);
```

4. Now we will Create **Substation** Table. For that we will Execute Script

```
CREATE TABLE Substation (  
    Substation_ID CHAR(10) PRIMARY KEY,  
    Area VARCHAR(200),  
    Voltage_Level DECIMAL(5, 2) CHECK (Voltage_Level >= 0),  
    Capacity DECIMAL(10, 2),  
    Transformer_Capacity DECIMAL(10, 2),  
    Circuit_Breakers VARCHAR(50),  
    Status VARCHAR(50) CHECK (Status IN ('Active', 'Inactive')),  
    Pin_Code CHAR(6) CHECK (Pin_Code ~ '^[0-9]{6}'),  
    FOREIGN KEY (Pin_Code) REFERENCES Pin_Code(Pin_Code) ON DELETE  
    RESTRICT ON UPDATE CASCADE,  
    CONSTRAINT unique_area UNIQUE (Area, Pin_Code)  
);
```

5. Now we will Create **Feeder** Table. For that we will Execute Script

```
CREATE TABLE Feeder (  
    Feeder_ID CHAR(12) PRIMARY KEY,  
    Area_Name VARCHAR(200),  
    Voltage_Level DECIMAL(5, 2) CHECK (Voltage_Level >= 0),
```

```

        Capacity DECIMAL(10, 2),
        Load_Profile DECIMAL(10, 2),
        Circuit_Breaker_Rating DECIMAL(5, 2),
        Number_Of_Meters INT CHECK (Number_Of_Meters >= 0),
        Substation_ID CHAR(10) NOT NULL,
        CONSTRAINT unique_feeder_substation UNIQUE (Substation_ID,
Area_Name),
        FOREIGN KEY (Substation_ID) REFERENCES Substation(Substation_ID)
ON DELETE RESTRICT
);

```

6. Now we will Create **Meter** Table. For that we will Execute Script

```

CREATE TABLE Meter (
    Meter_ID CHAR(12) PRIMARY KEY,
    Current_Reading DECIMAL(10, 2) CHECK (Current_Reading >= 0),
    Status VARCHAR(50) CHECK (Status IN ('Active', 'Inactive')),
    Installation_Date DATE CHECK (Installation_Date <= CURRENT_DATE),
    Last_Reading_Date DATE CHECK (Last_Reading_Date <= CURRENT_DATE),
    Feeder_ID CHAR(12) NOT NULL,
    Customer_ID CHAR(12) NOT NULL,
    FOREIGN KEY (Feeder_ID) REFERENCES Feeder(Feeder_ID) ON DELETE
RESTRICT,
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
ON DELETE CASCADE
);

```

7. Now we will Create **Electricity Rate** Table. For that we will Execute Script

```

CREATE TABLE Electricity_Rate (
    Rate_ID CHAR(12) PRIMARY KEY,
    Rate_Start_Date DATE CHECK (Rate_Start_Date <= CURRENT_DATE),
    Rate_End_Date DATE CHECK (Rate_End_Date <= CURRENT_DATE),
    Electricity_Rate DECIMAL(10, 2) CHECK (Electricity_Rate >=
0),
    Customer_Type_ID INT NOT NULL,
    CONSTRAINT unique_rate_date_range UNIQUE (Customer_Type_ID,
Rate_Start_Date, Rate_End_Date),
    FOREIGN KEY (Customer_Type_ID) REFERENCES Customer_Type(Customer_Type_ID)
ON DELETE RESTRICT
);

```

8. Now we will Create **Bill** Table. For that we will Execute Script

```

CREATE TABLE Bill (
    Bill_ID CHAR(16) PRIMARY KEY,
    Total_Price DECIMAL(10, 2) CHECK (Total_Price >= 0),
    Billing_Date DATE CHECK (Billing_Date <= CURRENT_DATE),
    Payment_Status VARCHAR(50) CHECK (Payment_Status IN ('Paid',
'Unpaid')),

```

```

        Meter_ID CHAR(12) NOT NULL,
        Rate_ID CHAR(12) NOT NULL,
        CONSTRAINT unique_customer_meter_billing UNIQUE (Meter_ID, Billing_Date),
        FOREIGN KEY (Meter_ID) REFERENCES Meter(Meter_ID) ON DELETE
CASCADE,
        FOREIGN KEY (Rate_ID) REFERENCES Electricity_Rate(Rate_ID)
ON DELETE RESTRICT
);

```

9. Now we will Create **Power Source Owner** Table. For that we will Execute Script

```

CREATE TABLE Power_Source_Owner (
    Owner_ID CHAR(3) PRIMARY KEY CHECK (Owner_ID ~ '^[A-Za-z]{3}'),
    Name VARCHAR(200) NOT NULL,
    Office_No VARCHAR(50),
    Street VARCHAR(200),
    Contact_Info VARCHAR(10) CHECK (Contact_Info ~ '[0-9]{10}')
UNIQUE,
    Power_Source_Ownership_Details VARCHAR(200),
    Pin_Code CHAR(6) CHECK (Pin_Code ~ '[0-9]{6}'),
    FOREIGN KEY (Pin_Code) REFERENCES Pin_Code(Pin_Code) ON DELETE
RESTRICT ON UPDATE CASCADE,
    CONSTRAINT unique_address_owner UNIQUE (Office_No, Street,
Pin_Code)
);

```

10. Now we will Create **Power Source** Table. For that we will Execute Script

```

CREATE TABLE Power_Source (
    Power_Source_ID CHAR(6) PRIMARY KEY,
    Type VARCHAR(50) CHECK (Type IN ('Solar', 'Wind', 'Hydro',
'Thermal')),
    Capacity DECIMAL(10, 2),
    Area VARCHAR(200),
    Generation_Data DECIMAL(10, 2),
    Status VARCHAR(50) CHECK (Status IN ('Active', 'Inactive')),
    Pin_Code CHAR(6) CHECK (Pin_Code ~ '[0-9]{6}'),
    Substation_ID CHAR(10) NOT NULL,
    Owner_ID CHAR(3) NOT NULL,
    CONSTRAINT unique_power_source UNIQUE (Substation_ID, Area),
    FOREIGN KEY (Pin_Code) REFERENCES Pin_Code(Pin_Code) ON DELETE
RESTRICT ON UPDATE CASCADE,
    FOREIGN KEY (Substation_ID) REFERENCES Substation(Substation_ID)
ON DELETE RESTRICT,
    FOREIGN KEY (Owner_ID) REFERENCES Power_Source_Owner(Owner_ID)
ON DELETE RESTRICT
);

```

11. Now we will Create **Maintenance Team** Table. For that we will Execute Script

```
CREATE TABLE Maintenance_Team (
    Team_ID CHAR(8) PRIMARY KEY,
    Team_Type VARCHAR(100) NOT NULL
);
```

12. Now we will Create **Outage** Table. For that we will Execute Script

```
CREATE TABLE Outage (
    Outage_ID CHAR(12) PRIMARY KEY,
    Start_Date_Time TIMESTAMP,
    End_Date_Time TIMESTAMP,
    Status VARCHAR(50) CHECK (Status IN ('Scheduled', 'Ongoing',
'Resolved')),
    Outage_Type VARCHAR(50) CHECK (Outage_Type IN ('Scheduled',
'Unexpected')),
    Cause VARCHAR(200),
    Maintenance_ID CHAR(8) NOT NULL,
    FOREIGN KEY (Maintenance_ID) REFERENCES Maintenance_Schedule(Maintenance_ID)
ON DELETE CASCADE
);
```

13. Now we will Create **Maintenance Schedule** Table. For that we will Execute Script

```
CREATE TABLE Maintenance_Schedule (
    Maintenance_ID CHAR(12) PRIMARY KEY,
    Start_Date_Time TIMESTAMP,
    End_Date_Time TIMESTAMP,
    Maintenance_Type VARCHAR(100),
    Status VARCHAR(50) CHECK (Status IN ('Scheduled', 'Completed',
'Cancelled')),
    Substation_ID CHAR(10),
    Team_ID CHAR(8) NOT NULL,
    FOREIGN KEY (Substation_ID) REFERENCES Substation(Substation_ID)
ON DELETE SET NULL,
    FOREIGN KEY (Team_ID) REFERENCES Maintenance_Team(Team_ID)
ON DELETE SET NULL,
    CONSTRAINT unique_start_team UNIQUE (Start_Date_Time, Team_ID),
    CONSTRAINT unique_end_team UNIQUE (End_Date_Time, Team_ID),
    CONSTRAINT unique_type_substation UNIQUE (Maintenance_Type,
Start_Date_Time, Substation_ID)
);
```

14. Now we will Create **Employee** Table. For that we will Execute Script

```
CREATE TABLE Employee (
    Employee_ID CHAR(8) PRIMARY KEY,
    Employee_Name VARCHAR(200) NOT NULL,
    Role VARCHAR(100),
    Department VARCHAR(100),
```

```

Salary DECIMAL(10, 2) CHECK (Salary > 0),
Contact_Info VARCHAR(10) CHECK (Contact_Info ~ '[0-9]{10}'),
UNIQUE,
Substation_ID CHAR(10),
Team_ID CHAR(8),
FOREIGN KEY (Substation_ID) REFERENCES Substation(Substation_ID)
ON DELETE SET NULL,
FOREIGN KEY (Team_ID) REFERENCES Maintenance_Team(Team_ID)
ON DELETE SET NULL
);

```

15. Now we will Create **Affected Area** Table. For that we will Execute Script

```

CREATE TABLE Affected_Area (
    Affected_Area_ID CHAR(12) PRIMARY KEY,
    Area_Type VARCHAR(100),
    Area VARCHAR(200),
    Pin_Code CHAR(6) CHECK (Pin_Code ~ '[0-9]{6}'),
    Outage_ID INT NOT NULL,
    FOREIGN KEY (Pin_Code) REFERENCES Pin_Code(Pin_Code) ON DELETE
RESTRICT ON UPDATE CASCADE,
    FOREIGN KEY (Outage_ID) REFERENCES Outage(Outage_ID) ON DELETE
CASCADE,
    CONSTRAINT unique_area_pincode UNIQUE (Area, Pin_Code, Outage_ID)
);

```

16. Now we will Create **Inter Connects** Table for the Relation between multiple Substations. For that we will Execute Script

```

CREATE TABLE Substation_Connections (
    Substation1_ID CHAR(10) NOT NULL,
    Substation2_ID CHAR(10) NOT NULL,
    PRIMARY KEY (Substation1_ID, Substation2_ID),
    FOREIGN KEY (Substation1_ID) REFERENCES Substation(Substation_ID)
ON DELETE CASCADE,
    FOREIGN KEY (Substation2_ID) REFERENCES Substation(Substation_ID)
ON DELETE CASCADE
);

```

3 Constraints that cannot be Implemented by DDL

(a) **Non-overlapping Maintenance Schedules**

Constraint : Maintenance schedules for the same substation or feeder must not overlap in time.

Reason : SQL DDL can't check overlapping date-time ranges

(b) **Meter Readings Validation**

Constraint : The current meter reading must always be greater than

or equal to the previous meter reading.

Reason : SQL DDL cannot enforce constraints based on the previous row's data.

(c) **Customer Deactivation Rules**

Constraint : A customer can only be deactivated if their outstanding bills are paid.

Reason : SQL DDL can't easily enforce rules based on related records like outstanding bills.

(d) **Employee Assignment Validation**

Constraint : An employee can only be assigned to one substation or feeder at a time.

Reason : SQL DDL cannot enforce exclusive relationships between multiple columns.