SQL Code

```
-- 1. AUDIT AND NOTIFICATION TABLES
CREATE TABLE AuditLogging (
    LogID INT PRIMARY KEY AUTO_INCREMENT,
    UserID INT NOT NULL,
    Action VARCHAR(20) NOT NULL,
    TableName VARCHAR(50) NOT NULL,
    RecordID INT NOT NULL,
    Timestamp DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    FOREIGN KEY (UserID) REFERENCES User(UserID)
);

CREATE TABLE Notification (
    NotificationID INT PRIMARY KEY AUTO_INCREMENT,
    UserID INT NOT NULL,
    Message TEXT NOT NULL,
    Timestamp DATETIME NOT NULL DEFAULT CURRENT_TIMESTAMP,
    IsRead BOOLEAN NOT NULL DEFAULT FALSE,
    FOREIGN KEY (UserID) REFERENCES User(UserID)
);

-- 2. LOOKUP TABLES
CREATE TABLE City (
    CityID INT PRIMARY KEY AUTO_INCREMENT,
    CityName VARCHAR(50) UNIQUE NOT NULL
);

INSERT INTO City (CityName) VALUES
('Alex'),
('Aswan'),
('Cairo'),
('Suez');
```

```
CREATE TABLE PlanStatus (
   StatusID INT PRIMARY KEY AUTO_INCREMENT,
   StatusName VARCHAR(50) UNIQUE NOT NULL
);
INSERT INTO PlanStatus (StatusName) VALUES
('Draft'),
('InProgress'),
('Complete');

-- 3. CORE TABLES
CREATE TABLE Branch (
   BranchID INT PRIMARY KEY AUTO_INCREMENT,
   CityID INT NOT NULL,
   Location VARCHAR(100) NOT NULL,
   NumberOfEmployees INT CHECK (NumberOfEmployees > 0), -- KEPT with validation
   FOREIGN KEY (CityID) REFERENCES City(CityID)
);
INSERT INTO Branch (CityID, Location, NumberOfEmployees) VALUES
-- Cairo branches
(3, 'Maadi', 35),
(3, 'Nasr City', 38),
(3, 'New Cairo', 40),
(3, 'Badr City', 32),

-- Aswan branches
(2, 'New Aswan', 24),
(2, 'Edfu', 20),
(2, 'Tushka', 20),
(2, 'Nag el Kurkur', 22),

-- Alex branches
(1, 'El Laban', 28),
(1, 'Borg El Arab', 25),
(1, 'Al Amreya', 29),
(1, 'El Agamy', 27),
```

```
-- Suez branches
(4, 'Al Arbaeen', 15),
(4, 'Al Adabeya', 18),
(4, 'Faisal District', 18),
(4, 'El Ganayen', 20);

CREATE TABLE User (
    UserID INT PRIMARY KEY AUTO_INCREMENT,
    BranchID INT NULL,
    UserName VARCHAR(100) NOT NULL,
    UserRole ENUM('BranchUser', 'OPManager', 'CIO', 'CEO') NOT NULL,
    UserEmail VARCHAR(100) UNIQUE NOT NULL,
    Password VARCHAR(255) NOT NULL,
    FOREIGN KEY (BranchID) REFERENCES Branch(BranchID),
    CONSTRAINT CHK_UserRole CHECK (
        (UserRole = 'BranchUser' AND BranchID IS NOT NULL) OR
        (UserRole IN ('OPManager', 'CIO', 'CEO') AND BranchID IS NULL)
    )
);
INSERT INTO User (BranchID, UserName, UserRole, UserEmail, Password) VALUES
(1, 'Amr Azzam', 'BranchUser', 'amr.azzam@gmail.com', 'AZZ89754'),
(2, 'Mehad Mohamed', 'BranchUser', 'mehad.mohamed@gmail.com', 'ADD-5648'),
(3, 'Ahmed Maher', 'BranchUser', 'ahmed.maher@gmail.com', 'AH.M1236'),
(4, 'Shawkat Eldin', 'BranchUser', 'shawkat.eldin@gmail.com', 'DIN/89745'),
(5, 'Lujain Hesham', 'BranchUser', 'lujain.hesham@gmail.com', 'LOJ*9874'),
(6, 'Nagi Eid', 'BranchUser', 'nagi.eid@gmail.com', 'NAG/8974'),
(7, 'Baraa Ahmed', 'BranchUser', 'baraa.ahmed@gmail.com', 'BAR54789'),
(8, 'Abdallah Amr', 'BranchUser', 'abdallah.amr@gmail.com', 'ABDA**987'),
(9, 'Essam Elghandour', 'BranchUser', 'essam.elghandour@gmail.com', 'EGH**987'),
(10, 'Yehia Homaimy', 'BranchUser', 'yehia.homaimy@gmail.com', 'EGH7-098'),
(11, 'Mohab Elkandil', 'BranchUser', 'mohab.elkandil@gmail.com', 'moh-0987'),
(12, 'Nussaiba Khaled', 'BranchUser', 'nussaiba.khaled@gmail.com', 'NKH/8974'),
(13, 'Farouk Mahmoud', 'BranchUser', 'farouk.mahmoud@gmail.com', 'FKM-0987'),
(14, 'Karen Adel', 'BranchUser', 'karen.adel@gmail.com', 'krn**879'),
```

(15, 'Sarah Ahmed', 'BranchUser', 'sarah.ahmed@gmail.com', 'SAR//897'),

(16, 'Nada Ibrahim', 'BranchUser', 'nada.ibrahim@gmail.com', 'NAD-1234'),

(NULL, 'Ahmed Samir', 'OPManager', 'ahmed.samir@gmail.com', 'OPM*8974'),

(NULL, 'Amr Daba', 'CIO', 'amr.daba@gmail.com', 'CIO**789'),

(NULL, 'Osama Hanafy', 'CEO', 'osama.hanafy@gmail.com', 'CEO-1598');

```
-- 4. BRANCH-SPECIFIC TABLES (WITH SUPPLIER KEPT)
CREATE TABLE CoffeeDistribution (
    DistributionID INT PRIMARY KEY AUTO_INCREMENT,
    BranchID INT NOT NULL,
    UserID INT NOT NULL,
    VehicleType ENUM('Minivan', 'Pickup Truck') NOT NULL,
    NumberOfVehicles INT CHECK (NumberOfVehicles > 0),
    DistancePerVehicle_KM DECIMAL(10,2) CHECK (DistancePerVehicle_KM > 0),
    TotalDistance_KM DECIMAL(10,2) AS (NumberOfVehicles * DistancePerVehicle_KM),
    FuelEfficiency DECIMAL(10,2) GENERATED ALWAYS AS (
        CASE
            WHEN VehicleType = 'Minivan' THEN 10
            WHEN VehicleType = 'Pickup Truck' THEN 15
        END
    ) STORED,
    V_CarbonEmissions_Kg DECIMAL(10,2) GENERATED ALWAYS AS (
        ROUND((TotalDistance_KM / FuelEfficiency) * 2.68, 2)
    ) STORED,
    FOREIGN KEY (BranchID) REFERENCES Branch(BranchID),
    FOREIGN KEY (UserID) REFERENCES User(UserID)
);
INSERT INTO CoffeeDistribution (BranchID, UserID, VehicleType, NumberOfVehicles, DistancePerVehicle_KM) VALUES
-- Cairo branches (highest emissions)
(1, 1, 'Minivan', 5, 150.0),
(2, 2, 'Pickup Truck', 4, 120.0),
(3, 3, 'Minivan', 6, 130.0),
(4, 4, 'Pickup Truck', 5, 140.0),

-- Aswan branches (medium-high emissions)
(5, 5, 'Pickup Truck', 3, 180.0),
(6, 6, 'Minivan', 4, 160.0),
(7, 7, 'Pickup Truck', 2, 200.0),
(8, 8, 'Minivan', 3, 170.0),
-- Alex branches (medium emissions)
(9, 9, 'Minivan', 4, 100.0),
```

```
(10, 10, 'Pickup Truck', 3, 110.0),
(11, 11, 'Minivan', 3, 90.0),
(12, 12, 'Pickup Truck', 2, 120.0),

-- Suez branches (lowest emissions)
(13, 13, 'Pickup Truck', 2, 80.0),
(14, 14, 'Minivan', 3, 70.0),
(15, 15, 'Pickup Truck', 1, 90.0),
(16, 16, 'Minivan', 2, 60.0);

CREATE TABLE CoffeePackaging (
    PackagingID INT PRIMARY KEY AUTO_INCREMENT,
    BranchID INT NOT NULL,
    UserID INT NOT NULL,
    PackagingWaste_KG DECIMAL(10,2) CHECK (PackagingWaste_KG >= 0),
    Pa_CarbonEmissions_KG DECIMAL(10,2) GENERATED ALWAYS AS (
        ROUND(PackagingWaste_KG * 6, 2)
    ) STORED,
    FOREIGN KEY (BranchID) REFERENCES Branch(BranchID),
    FOREIGN KEY (UserID) REFERENCES User(UserID));
CREATE TABLE CoffeeProduction (
    ProductionID INT PRIMARY KEY AUTO_INCREMENT,
    BranchID INT NOT NULL,
    UserID INT NOT NULL,
    Supplier VARCHAR(100) NOT NULL, -- KEPT and made mandatory
    CoffeeType ENUM('Arabica Beans', 'Robusta Beans', 'Organic Beans') NOT NULL,
    ProductType ENUM('Ground', 'Whole Bean', 'Instant') NOT NULL,
    ProductionQuantitiesOfCoffee_KG DECIMAL(10,2) CHECK (ProductionQuantitiesOfCoffee_KG > 0),
    Pr_CarbonEmissions_KG DECIMAL(10,2) GENERATED ALWAYS AS (
        ROUND(ProductionQuantitiesOfCoffee_KG * 6.4, 2)
    ) STORED,
    FOREIGN KEY (BranchID) REFERENCES Branch(BranchID),
    FOREIGN KEY (UserID) REFERENCES User(UserID)
);
INSERT INTO CoffeeProduction (BranchID, UserID, Supplier, CoffeeType, ProductType, ProductionQuantitiesOfCoffee_KG) VALUES
```

```
-- Cairo branches (highest production)
(1, 1, 'Egyptian Coffee Co.', 'Arabica Beans', 'Ground', 120.0),
(2, 2, 'Cairo Roasters', 'Robusta Beans', 'Whole Bean', 110.5),
(3, 3, 'Nile Valley Beans', 'Arabica Beans', 'Instant', 150.2),
(4, 4, 'Delta Coffee', 'Organic Beans', 'Ground', 130.7),

-- Aswan branches (medium-high production)
(5, 5, 'Aswan Farmers Co-op', 'Arabica Beans', 'Whole Bean', 95.0),
(6, 6, 'Upper Egypt Roasters', 'Robusta Beans', 'Ground', 88.4),
(7, 7, 'Nubian Gold', 'Organic Beans', 'Instant', 92.1),
(8, 8, 'Red Sea Coffee', 'Arabica Beans', 'Whole Bean', 90.8),

-- Alex branches (medium production)
(9, 9, 'Mediterranean Beans', 'Arabica Beans', 'Ground', 75.2),
(10, 10, 'Alexandria Roasters', 'Robusta Beans', 'Instant', 80.5),
(11, 11, 'Port City Coffee', 'Organic Beans', 'Whole Bean', 78.7),
(12, 12, 'Western Delta Co.', 'Arabica Beans', 'Ground', 82.3),

-- Suez branches (lowest production)
(13, 13, 'Suez Canal Coffee', 'Robusta Beans', 'Whole Bean', 52.4),
(14, 14, 'Gulf Roasters', 'Arabica Beans', 'Instant', 48.6),
(15, 15, 'East Wind Beans', 'Organic Beans', 'Ground', 50.2),
(16, 16, 'Sinai Coffee Co.', 'Arabica Beans', 'Whole Bean', 45.8);



CREATE TABLE ReductionStrategy (
    ReductionID INT PRIMARY KEY AUTO_INCREMENT,
    BranchID INT NOT NULL,
    UserID INT NOT NULL,
    ReductionStrategy TEXT NOT NULL,
    StatusID INT NOT NULL,
    ImplementationCosts DECIMAL(12,2) CHECK (ImplementationCosts >= 0),
    ProjectedAnnualProfits DECIMAL(12,2) GENERATED ALWAYS AS (
        ROUND(ImplementationCosts * 0.2, 2)
```

```
    ) STORED,
    FOREIGN KEY (BranchID) REFERENCES Branch(BranchID),
    FOREIGN KEY (UserID) REFERENCES User(UserID),
    FOREIGN KEY (StatusID) REFERENCES PlanStatus(StatusID)
);
INSERT INTO ReductionStrategy (BranchID, UserID, ReductionStrategy, StatusID, ImplementationCosts) VALUES
-- Cairo branches (most expensive/complex strategies)
(1, 1, 'Implement electric delivery vehicles and solar-powered packaging equipment', 3, 250000.00),
(2, 2, 'Install waste-to-energy system for coffee grounds and biodegradable packaging', 2, 180000.00),
(3, 3, 'Complete facility retrofit with energy-efficient roasting and LED lighting', 3, 300000.00),
(4, 4, 'Advanced water recycling system and carbon capture for roasting process', 1, 220000.00),

-- Aswan branches (medium-complexity strategies)
(5, 5, 'Switch to hybrid delivery vehicles and compostable packaging', 2, 120000.00),
(6, 6, 'Install solar panels for 50% of energy needs and improved insulation', 3, 95000.00),
(7, 7, 'Implement coffee chaff recycling program and efficient roasting', 2, 85000.00),
(8, 8, 'Rainwater harvesting system and local biomass packaging', 1, 75000.00),

-- Alex branches (moderate strategies)
(9, 9, 'Upgrade to energy-efficient roasting equipment', 3, 80000.00),
(10, 10, 'Switch to reusable transport containers and LED lighting', 2, 65000.00),
(11, 11, 'Implement waste segregation and recycling program', 1, 50000.00),
(12, 12, 'Install variable-speed drives on all motors', 2, 45000.00),

-- Suez branches (simplest/cheapest strategies)
(13, 13, 'Basic packaging optimization and staff training', 3, 30000.00),
(14, 14, 'Implement proper maintenance schedule for equipment', 2, 25000.00),
(15, 15, 'Switch to local suppliers to reduce transport emissions', 1, 35000.00),
(16, 16, 'Basic energy monitoring system installation', 2, 20000.00);

DELIMITER //
CREATE PROCEDURE GetUserByEmail(IN email VARCHAR(100))
BEGIN
    SELECT * FROM User WHERE UserEmail = email;
END //
```

```
CREATE PROCEDURE GetBranchMetrics(IN branchId INT)
BEGIN
   -- Returns all carbon metrics for a branch
   SELECT
      (SELECT SUM(V_CarbonEmissions_Kg) FROM CoffeeDistribution WHERE BranchID = branchId) AS distribution_emissions,
      (SELECT SUM(Pa_CarbonEmissions_KG) FROM CoffeePackaging WHERE BranchID = branchId) AS packaging_emissions,
      (SELECT SUM(Pr_CarbonEmissions_KG) FROM CoffeeProduction WHERE BranchID = branchId) AS production_emissions;
END //
DELIMITER ;
```