



- 1.1. Create a table `soil_sensors` if it does not exist with the following columns:
  - `sensor_id` (INT, Primary Key)
  - `moisture_level` (INT)
  - `recorded_time` (DATETIME)
- 1.2. Then Create a stored procedure `GetDryFields` that:
  - Accepts a moisture threshold as an input parameter
  - Displays all sensor records where `moisture_level` is below the given threshold.
  
- 2.1. Create a table `fields` if it does not exist with the following columns:
  - `field_id` (INT, Primary Key)
  - `area_sq_meters` (INT)
  - `crop_type` (VARCHAR(50))
- 2.2. Create a stored function `CalculateWaterRequirement` that:
  - Accepts `area_sq_meters` as input
  - Returns the total water required per irrigation cycle
  - Assume 5 liters of water per square meter
  
- 3.1. Create a table `irrigation_logs` if it does not exist with the following columns:
  - `log_id` (INT, Primary Key)
  - `field_id` (INT)
  - `water_used_liters` (DECIMAL(10,2))
  - `irrigation_date` (DATE)
- 3.2. Create a stored procedure `GetTotalWaterUsage` that:
  - Accepts a date as an input parameter
  - Returns the total water used on that date using an OUT parameter.
  
- 4.1. Create a table `field_status` if it does not exist with the following columns:
  - `field_id` (INT, Primary Key)
  - `moisture_level` (INT)
- 4.2. Create a stored procedure `CountCriticalFields` that:
  - Uses a cursor
  - Iterates through all records
  - Counts how many fields have a moisture level below 30
  - Displays the total count
  
- 5.1. Create a table `irrigation_schedule` if it does not exist with the following columns:
  - `schedule_id` (INT, Primary Key)
  - `field_id` (INT)
  - `irrigation_duration` (INT) (*in minutes*)
- 5.2. Create a stored procedure `AdjustIrrigationDuration` that:
  - Uses a cursor
  - Increases irrigation duration by 20%
  - Updates each row individually