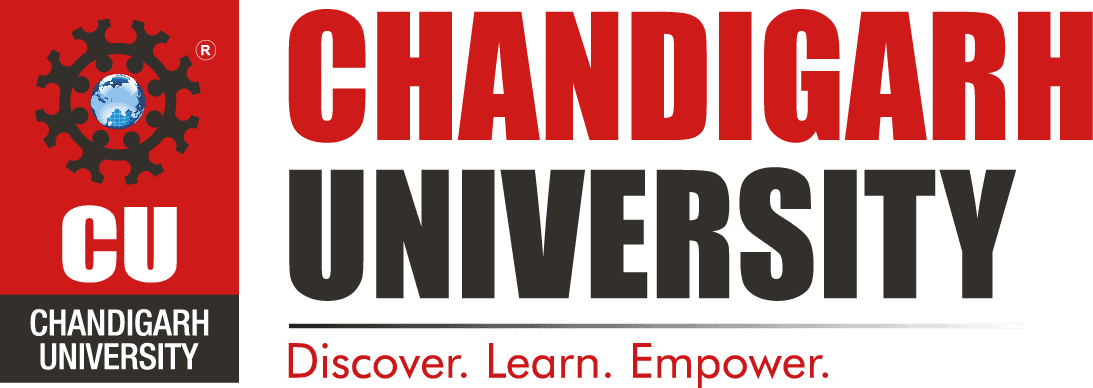
**CASE STUDY REPORT ON ARMY MANAGEMENT SYSTEM**

|  |  |
| --- | --- |
| **Student Name: Jashan Chouhan** | **UID: 23BCA10314** |
| **Branch: UIC/BCA** | **Section/Group: BCA-4/b** |
| **Semester: 4th**  **Submitted to: Mr. Arvinder Singh Code: 23CAP-252** | **Subject Name: DBMS** |

**CODE:**

create database huuu;

use huuu;

-- Creating the Units table

-- Creating the Units table

CREATE TABLE Units (

UnitID INT PRIMARY KEY,

UnitName VARCHAR(50) NOT NULL,

Location VARCHAR(50) NOT NULL

);

-- Creating the Soldiers table

CREATE TABLE Soldiers (

SoldierID INT PRIMARY KEY,

Name VARCHAR(50) NOT NULL,

UnitID INT NOT NULL,

ContactDetails VARCHAR(50),

FOREIGN KEY (UnitID) REFERENCES Units(UnitID) ON DELETE CASCADE

);

-- Creating the Weapons table

CREATE TABLE Weapons (

WeaponID INT PRIMARY KEY,

WeaponName VARCHAR(50) NOT NULL,

Type VARCHAR(30) NOT NULL,

UnitID INT NOT NULL,

FOREIGN KEY (UnitID) REFERENCES Units(UnitID) ON DELETE CASCADE

);

-- Creating the Missions table

CREATE TABLE Missions (

MissionID INT PRIMARY KEY,

MissionName VARCHAR(50) NOT NULL,

StartDate DATE NOT NULL,

EndDate DATE NOT NULL,

Status VARCHAR(20) CHECK (Status IN ('Planned', 'Ongoing', 'Completed'))

);

-- Creating the Assignments table

CREATE TABLE Assignments (

AssignmentID INT PRIMARY KEY,

SoldierID INT NOT NULL,

MissionID INT NOT NULL,

Role VARCHAR(50) NOT NULL,

FOREIGN KEY (SoldierID) REFERENCES Soldiers(SoldierID),

FOREIGN KEY (MissionID) REFERENCES Missions(MissionID)

);

-- Creating the Inventory table

CREATE TABLE Inventory (

ItemID INT PRIMARY KEY,

ItemName VARCHAR(50) NOT NULL,

Quantity INT NOT NULL,

UnitID INT NOT NULL,

FOREIGN KEY (UnitID) REFERENCES Units(UnitID)

);

-- Creating the Training table

CREATE TABLE Training (

TrainingID INT PRIMARY KEY,

TrainingType VARCHAR(50) NOT NULL,

TrainerID INT NOT NULL,

Duration INT NOT NULL

);

-- Creating the Attendance table

CREATE TABLE Attendance (

AttendanceID INT PRIMARY KEY,

SoldierID INT NOT NULL,

TrainingID INT NOT NULL,

Date DATE NOT NULL,

Status VARCHAR(10) NOT NULL,

FOREIGN KEY (SoldierID) REFERENCES Soldiers(SoldierID),

FOREIGN KEY (TrainingID) REFERENCES Training(TrainingID)

);

-- Populating Units table

INSERT INTO Units (UnitID, UnitName, Location) VALUES

(1, 'Alpha', 'Base A'),

(2, 'Bravo', 'Base B'),

(3, 'Charlie', 'Base C'),

(4, 'Delta', 'Base D'),

(5, 'Echo', 'Base E');

-- Populating Soldiers table

INSERT INTO Soldiers (SoldierID, Name, UnitID, ContactDetails) VALUES

(1, 'Raj Kumar', 1, 'raj.kumar@example.com'),

(2, 'Amit Singh', 2, 'amit.singh@example.com'),

(3, 'Priya Sharma', 3, 'priya.sharma@example.com'),

(4, 'Vikram Rao', 4, 'vikram.rao@example.com'),

(5, 'Neha Verma', 5, 'neha.verma@example.com');

-- Populating Weapons table

INSERT INTO Weapons (WeaponID, WeaponName, Type, UnitID) VALUES

(1, 'AK-47', 'Rifle', 1),

(2, 'M4 Carbine', 'Rifle', 2),

(3, 'Grenade', 'Explosive', 3),

(4, 'Sniper', 'Rifle', 4),

(5, 'Rocket Launcher', 'Heavy', 5);

-- Populating Missions table

INSERT INTO Missions (MissionID, MissionName, StartDate, EndDate, Status) VALUES

(1, 'Operation Thunder', '2025-01-05', '2025-01-10', 'Completed'),

(2, 'Desert Storm', '2025-02-15', '2025-02-20', 'Planned'),

(3, 'Sea Shield', '2025-03-10', '2025-03-15', 'Ongoing'),

(4, 'Mountain Strike', '2025-04-05', '2025-04-10', 'Completed'),

(5, 'Urban Defender', '2025-05-12', '2025-05-20', 'Planned');

-- Populating Assignments table

INSERT INTO Assignments (AssignmentID, SoldierID, MissionID, Role) VALUES

(1, 1, 1, 'Sniper'),

(2, 2, 2, 'Scout'),

(3, 3, 3, 'Communications'),

(4, 4, 4, 'Commander'),

(5, 5, 5, 'Medic');

-- Populating Inventory table

INSERT INTO Inventory (ItemID, ItemName, Quantity, UnitID) VALUES

(1, 'Ammo Boxes', 50, 1),

(2, 'First Aid Kits', 20, 2),

(3, 'Rations', 100, 3),

(4, 'Radios', 15, 4),

(5, 'Camouflage Gear', 30, 5);

-- Populating Training table

INSERT INTO Training (TrainingID, TrainingType, TrainerID, Duration) VALUES

(1, 'Sniper Training', 1, 14),

(2, 'Urban Warfare', 2, 10),

(3, 'Communication Skills', 3, 7),

(4, 'Medical Aid', 4, 5),

(5, 'Tactical Maneuvers', 5, 12);

-- Populating Attendance table

INSERT INTO Attendance (AttendanceID, SoldierID, TrainingID, Date, Status) VALUES

(1, 1, 1, '2025-01-15', 'Present'),

(2, 2, 2, '2025-02-20', 'Absent'),

(3, 3, 3, '2025-03-25', 'Present'),

(4, 4, 4, '2025-04-30', 'Present'),

(5, 5, 5, '2025-05-18', 'Present');

SELECT \* FROM Missions;

-- Insert a new soldier

INSERT INTO Soldiers (SoldierID, Name, UnitID, ContactDetails) VALUES

(6, 'Arjun Singh', 3, 'arjun.singh@example.com');

UPDATE Missions SET Status = 'Completed' WHERE MissionID = 1;

SELECT Soldiers.Name, Assignments.Role, Missions.MissionName

FROM Soldiers

JOIN Assignments ON Soldiers.SoldierID = Assignments.SoldierID

JOIN Missions ON Assignments.MissionID = Missions.MissionID;

SELECT Units.UnitName, Weapons.WeaponName, Weapons.Type

FROM Weapons

JOIN Units ON Weapons.UnitID = Units.UnitID;

SELECT Status, COUNT(\*) AS MissionCount

FROM Missions

GROUP BY Status;

SELECT Units.UnitName, COUNT(Soldiers.SoldierID) AS TotalSoldiers

FROM Units

JOIN Soldiers ON Units.UnitID = Soldiers.UnitID

GROUP BY Units.UnitName;

SELECT Units.UnitName, SUM(Inventory.Quantity) AS TotalSupplies

FROM Units

JOIN Inventory ON Units.UnitID = Inventory.UnitID

GROUP BY Units.UnitName;

SELECT TrainingType, AVG(Duration) AS AvgDuration

FROM Training

GROUP BY TrainingType;

SELECT Units.UnitName, COUNT(Soldiers.SoldierID) AS TotalSoldiers

FROM Units

JOIN Soldiers ON Units.UnitID = Soldiers.UnitID

GROUP BY Units.UnitName

HAVING COUNT(Soldiers.SoldierID) > 5;

-- Get soldiers and their respective unit names

SELECT Soldiers.Name, Units.UnitName

FROM Soldiers

INNER JOIN Units ON Soldiers.UnitID = Units.UnitID;

SELECT Soldiers.Name, Missions.MissionName

FROM Soldiers

LEFT JOIN Assignments ON Soldiers.SoldierID = Assignments.SoldierID

LEFT JOIN Missions ON Assignments.MissionID = Missions.MissionID;

SELECT Missions.MissionName, Soldiers.Name

FROM Missions

RIGHT JOIN Assignments ON Missions.MissionID = Assignments.MissionID

RIGHT JOIN Soldiers ON Assignments.SoldierID = Soldiers.SoldierID;