Mini Project Report

Database Management System

"MILITARY PERSONNEL MANAGEMENT"

COURSE NAME: DATABASE MANAGEMENT SYSTEM

COURSE ID: 2CP01

FACULTY: DR. HEMANT D. VASAVA

BRANCH: COMPUTER ENGINEERING

BATCH: B03

DEVELOPED BY:

RUSHABH A. SHAH -23CP031

VIRAL V. BHOI -23CP029

Birla Vishvakarma Mahavidyalaya

Engineering College, Vallabh Vidyanagar

[An Autonomous Institution]

Managed by Charutar Vidyamandal

1.FUNCTIONAL REQUIREMENTS:

1. Rank Management

Add, update, delete, and view ranks.

2. Unit Management

Add, update, delete, and view military units and their hierarchy.

3. Personnel Management

Add, update, delete personnel records. Assign/change rank and unit, and search personnel by various filters.

4. Training Programs

Manage training programs, assign personnel, and track training progress.

5. Deployment Management

Manage unit deployments, assign personnel, and view deployments by unit or location.

6. Leave Management

Record, approve/deny, and track personnel leave.

7. Equipment Management

Manage equipment inventory, assign to personnel, and track usage.

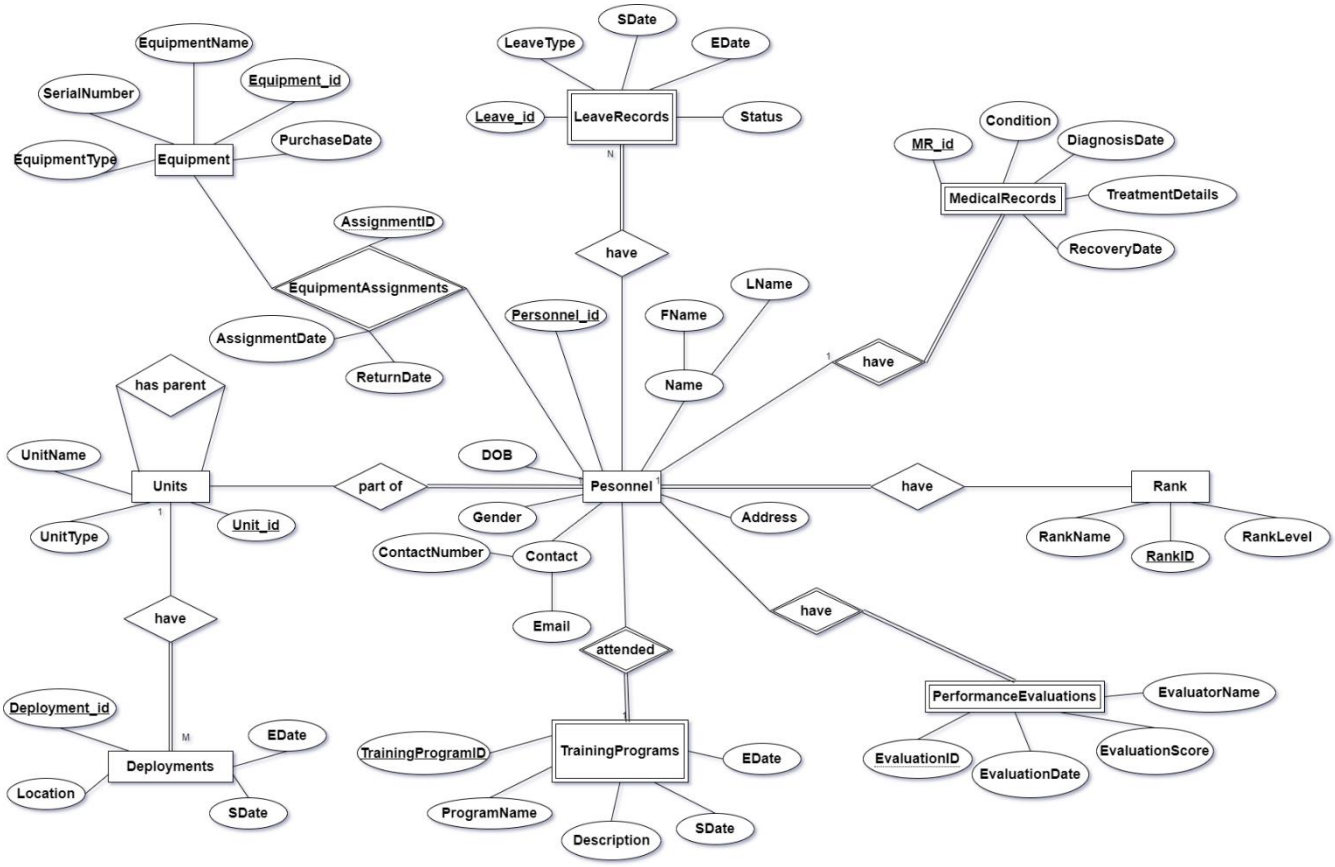
8. Medical Records

Create, update, and view personnel medical records.

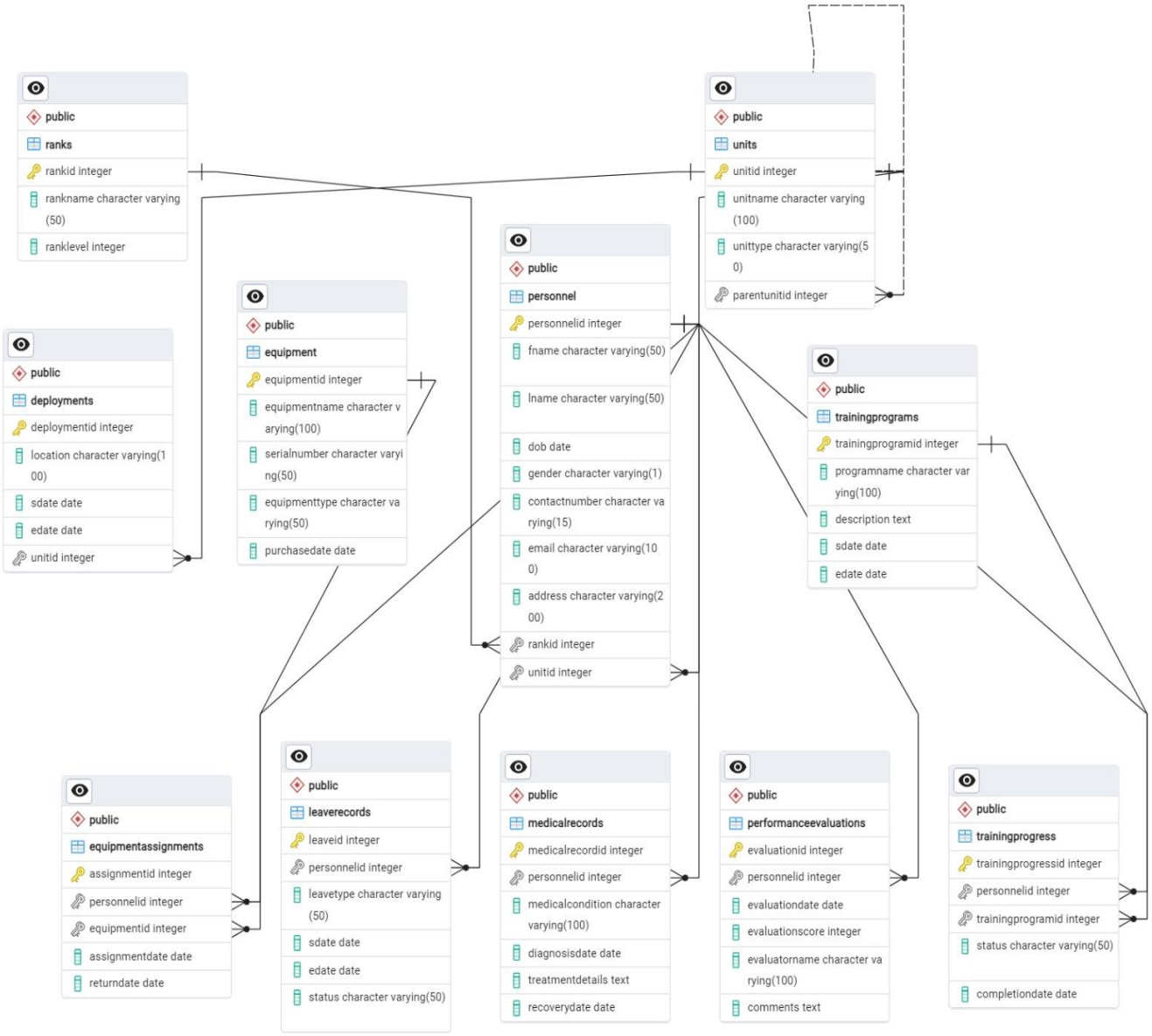
9. Performance Evaluations

Record and view personnel performance evaluations.

2.ER DIAGRAM:


EquipmentName
SerialNumber
Equipment_id
EquipmentType
Equipment
PurchaseDate
AssignmentID
EquipmentAssignments
AssignmentDate
ReturnDate
SDate
LeaveType
EDate
Leave_id
LeaveRecords
Status
N
have
Condition
MR_id
DiagnosisDate
MedicalRecords
TreatmentDetails
RecoveryDate
LName
Personnel_id
FName
Name
1
have
has parent
UnitName
Units
part of
DOB
Pesonnel
have
Rank
1
UnitType
Unit_id
Gender
Address
RankName
RankID
RankLevel
ContactNumber
Contact
have
Email
attended
Deployment_id
M
EDate
Deployments
Location
SDate
TrainingProgramID
TrainingPrograms
EDate
ProgramName
Description
SDate
PerformanceEvaluations
EvaluatorName
EvaluationID
EvaluationDate
EvaluationScore

3.RELATIONAL DIAGRAM:



4.FUNCTIONAL DEPENDENCIES AND NORMALIZATION:

1.Personnel Table

- **Functional Dependencies (FDs):**
  - PersonnelID → {FirstName, LastName, DateOfBirth, Gender, ContactNumber, Email, Address, RankID, UnitID, ServiceStartDate, ServiceEndDate}
- **Normalization:**
  - 3NF: No partial or transitive dependencies. All non-key attributes are fully functionally dependent on the primary key (PersonnelID). Therefore, it is in 3NF.

2. Units Table

- **Functional Dependencies (FDs):**
  - UnitID → {UnitName, UnitType, ParentUnitID}
- **Normalization:**

  - BCNF: This table is in BCNF since the primary key UnitID fully determines all other attributes, and there are no partial or transitive dependencies.

3. Ranks Table

- **Functional Dependencies (FDs):**

  - RankID → {RankName, RankLevel}

- **Normalization:**

  - BCNF: This table is in BCNF, as all attributes are fully dependent on the primary key.

4. TrainingPrograms Table

- **Functional Dependencies (FDs):**

  - TrainingProgramID → {ProgramName, Description, StartDate, EndDate}

- **Normalization:**

  - BCNF: This table is fully normalized since all attributes depend on the primary key.

5. Deployments Table

- **Functional Dependencies (FDs):**

  - DeploymentID → {DeploymentLocation, StartDate, EndDate, UnitID}

- **Normalization:**

  - BCNF: All attributes are fully functionally dependent on the primary key, making this table in BCNF.

6. TrainingProgress Table

- **Functional Dependencies (FDs):**

  - TrainingProgressID → {PersonnelID, TrainingProgramID, ProgressStatus, CompletionDate}

- **Normalization:**

  - BCNF: All attributes depend on the primary key, and there are no transitive dependencies. The table is in BCNF.

7. LeaveRecords Table

- **Functional Dependencies (FDs):**

  - LeaveID → {PersonnelID, LeaveType, StartDate, EndDate, ApprovalStatus}

- **Normalization:**

  - BCNF: The table is in BCNF because all non-key attributes are fully dependent on the primary key, LeaveID.

8. Equipment Table

- **Functional Dependencies (FDs):**

  - EquipmentID → {EquipmentName, SerialNumber, EquipmentType, PurchaseDate}

- **Normalization:**

  - BCNF: The table is in BCNF since there are no partial or transitive dependencies.

9. EquipmentAssignments Table

- **Functional Dependencies (FDs):**

  - AssignmentID → {PersonnelID, EquipmentID, AssignmentDate, ReturnDate}

- **Normalization:**

  - BCNF: All attributes depend on the primary key. There are no transitive dependencies, so this table is in BCNF.

10. MedicalRecords Table

- **Functional Dependencies (FDs):**

  - MedicalRecordID → {PersonnelID, MedicalCondition, DiagnosisDate, TreatmentDetails, RecoveryDate}

- **Normalization:**

  - BCNF: The table is in BCNF because all attributes depend on the primary key, and there are no transitive dependencies.

11. PerformanceEvaluations Table

- **Functional Dependencies (FDs):**

  - EvaluationID → {PersonnelID, EvaluationDate, EvaluationScore, EvaluatorName, Comments}

- **Normalization:**

  - BCNF: The table is in BCNF since all attributes are fully dependent on the primary key.

5.RELATION SCHEMA:

1. Ranks (RankID, RankName, RankLevel)
2. Units (UnitID, UnitName, UnitType, ParentUnitID)
3. Deployments (DeploymentID, Location, SDate, EDate, UnitID)
4. Personnel (PersonnelID, FName, LName, DOB, Gender, ContactNumber, Email, Address, RankID, UnitID)
5. TrainingPrograms (TrainingProgramID, ProgramName, Description, SDate, EDate)
6. TrainingProgress (TrainingProgressID, PersonnelID, TrainingProgramID, Status, CompletionDate)
7. LeaveRecords (LeaveID, PersonnelID, LeaveType, SDate, EDate, Status)
8. Equipment (EquipmentID, EquipmentName, SerialNumber, EquipmentType, PurchaseDate)
9. EquipmentAssignments (AssignmentID, PersonnelID, EquipmentID, AssignmentDate, ReturnDate)
10. MedicalRecords (MedicalRecordID, PersonnelID, MedicalCondition, DiagnosisDate, TreatmentDetails, RecoveryDate)
11. PerformanceEvaluations (EvaluationID, PersonnelID, EvaluationDate, EvaluationScore, EvaluatorName, Comments)

6.DATA DEFINITION LANGUAGE QUERY:

1.Ranks:

```
CREATE TABLE Ranks(
    RankID INT PRIMARY KEY,
    RankName VARCHAR(50),
    RankLevel INT);
```

2.Units:

```
CREATE TABLE Units(
    UnitID INT PRIMARY KEY,
    UnitName VARCHAR(100),
    UnitType VARCHAR(50),
    ParentUnitID INT,
    FOREIGN KEY (ParentUnitID) REFERENCES
Units(UnitID));
```

3.Deployments:

```
CREATE TABLE Deployments (
    DeploymentID INT PRIMARY KEY,
    Location VARCHAR(100),
    SDate DATE,EDate DATE,
    UnitID INT NOT NULL ,
     FOREIGN KEY (UnitID) REFERENCES
    Units(UnitID));
```

4.Personnel:

```
CREATE TABLE Personnel(
    PersonnelID INT PRIMARY KEY,
    FName VARCHAR(50),
    LName VARCHAR(50),
    DOB DATE,
    Gender VARCHAR(1),
    ContactNumber VARCHAR (15),
    Email VARCHAR(100),
    Address VARCHAR(200),
    RankID INT,
    UnitID INT,
    FOREIGN KEY (RankID) REFERENCES Ranks(RankID),
    FOREIGN KEY (UnitID) REFERENCES Units(UnitID)
);
```

5.TrainingPrograms:

```
CREATE TABLE TrainingPrograms (
    TrainingProgramID INT PRIMARY KEY,
    ProgramName VARCHAR(100),
    Description TEXT,
    SDate DATE,
    EDate DATE
);
```

6.TrainingProgress:

```
CREATE TABLE TrainingProgress (
    TrainingProgressID INT PRIMARY KEY,
    PersonnelID INT NOT NULL,
    TrainingProgramID INT NOT NULL,
    Status VARCHAR(50),
    CompletionDate DATE,
    FOREIGN KEY (PersonnelID) REFERENCES Personnel(PersonnelID),
    FOREIGN KEY (TrainingProgramID) REFERENCES TrainingPrograms(TrainingProgramID)
);
```

7.LeaveRecords:

```
CREATE TABLE LeaveRecords (
    LeaveID INT PRIMARY KEY,
    PersonnelID INT NOT NULL,
    LeaveType VARCHAR(50),
    SDate DATE,
    EDate DATE,
    Status VARCHAR(50),
    FOREIGN KEY (PersonnelID) REFERENCES Personnel(PersonnelID));
```

8.Equipment:

```
CREATE TABLE Equipment (
    EquipmentID INT PRIMARY KEY,
    EquipmentName VARCHAR(100),
    SerialNumber VARCHAR(50),
    EquipmentType VARCHAR(50),
    PurchaseDate DATE
);
```

9.EquipmentAssignments:

```
CREATE TABLE EquipmentAssignments (
    AssignmentID INT PRIMARY KEY,
    PersonnelID INT NOT NULL,
    EquipmentID INT NOT NULL,
    AssignmentDate DATE,
    ReturnDate DATE,
    FOREIGN KEY (PersonnelID) REFERENCES
Personnel(PersonnelID),
    FOREIGN KEY (EquipmentID) REFERENCES
Equipment(EquipmentID)
);
```

10.MedicalRecords:

```
CREATE TABLE MedicalRecords (
    MedicalRecordID INT PRIMARY KEY,
    PersonnelID INT NOT NULL,
    MedicalCondition VARCHAR(100),
    DiagnosisDate DATE,
    TreatmentDetails TEXT,
    RecoveryDate DATE,
    FOREIGN KEY (PersonnelID) REFERENCES Personnel(PersonnelID)
);
```

11.PerformanceEvaluations:

```
CREATE TABLE PerformanceEvaluations (
    EvaluationID INT PRIMARY KEY,
    PersonnelID INT NOT NULL,
    EvaluationDate DATE,
    EvaluationScore INT,
    EvaluatorName VARCHAR(100),
    Comments TEXT,
    FOREIGN KEY (PersonnelID) REFERENCES Personnel(PersonnelID)
);
```

7. QUERIES USING BASIC DBMS CONCEPTS, JOIN & SUBQUERIES:

1.List all Personnel along with their Ranks and Units:

```
SELECT P.PersonnelID, P.FName, P.LName, R.RankName,
U.UnitName

FROM Personnel P

JOIN Ranks R ON P.RankID = R.RankID

JOIN Units U ON P.UnitID = U.UnitID;
```

	personnelid integer	fname character varying (50)	lname character varying (50)	rankname character varying (50)	unitname character varying (100)
1	1	Arjun	Singh	Lieutenant	Eastern Command
2	2	Neha	Rao	Captain	Assam Rifles
3	3	Mahesh	Rawat	Major	Special Frontier Force
4	4	Sunil	Thakur	Colonel	Northern Command
5	5	Rajesh	Rana	Brigadier	Garhwal Rifles
6	6	Ankita	Singh	General	Rajput Regiment
7	7	Sakshi	Verma	Commander	Southern Command
8	8	Ajay	Kumar	Adjutant	Madras Regiment
9	9	Naveen	Yadav	Sergeant	Mahar Regiment
10	10	Priya	Deshmukh	Private	Parachute Regiment

2.Get Personnel details who have been deployed to a specific location:

```
SELECT P.PersonnelID, P.FName, P.LName

FROM Personnel P

WHERE P.UnitID = (SELECT UnitId FROM Deployments D

WHERE D.Location = 'Nagaland')
```

	personnelid [PK] integer	fname character varying (50)	lname character varying (50)
1	9	Naveen	Yadav

3. Show all Personnel currently on leave:

```
SELECT P.PersonnelID, P.FName, P.LName, L.LeaveType,
L.SDate, L.EDate

FROM Personnel P

JOIN LeaveRecords L ON P.PersonnelID = L.PersonnelID

WHERE L.Status = 'Approved'

  AND L.SDate >= '01-06-2023'

  AND L.EDate <= '31-12-2023';
```

	personnelid integer	fname character varying (50)	lname character varying (50)	leavetype character varying (50)	sdate date	edate date
1	1	Arjun	Singh	Sick Leave	2023-06-10	2023-06-20
2	2	Neha	Rao	Annual Leave	2023-07-01	2023-07-15
3	3	Mahesh	Rawat	Training Leave	2023-08-01	2023-08-15
4	4	Sunil	Thakur	Emergency Leave	2023-09-01	2023-09-10
5	5	Rajesh	Rana	Annual Leave	2023-10-01	2023-10-15

4.Find Personnel who have completed a specific training program:

```
SELECT P.PersonnelID, P.FName, P.LName,
T.ProgramName

FROM Personnel P

JOIN TrainingProgress TP ON P.PersonnelID =
TP.PersonnelID

JOIN TrainingPrograms T ON TP.TrainingProgramID =
T.TrainingProgramID

WHERE T.ProgramName = 'Basic Training' AND TP.Status
= 'Completed';
```

	personnelid integer	fname character varying (50)	lname character varying (50)	programname character varying (100)
1	1	Arjun	Singh	Basic Training

5. Retrieve all Equipment assigned to a specific Personnel:

```
SELECT P.PersonnelID, P.FName, P.LName,
E.EquipmentName, E.SerialNumber, EA.AssignmentDate

FROM Personnel P

JOIN EquipmentAssignments EA ON P.PersonnelID =
EA.PersonnelID

JOIN Equipment E ON EA.EquipmentID = E.EquipmentID

WHERE P.PersonnelID =5;
```

	personnelid integer	fname character varying (50)	lname character varying (50)	equipmentname character varying (100)	serialnumber character varying (50)	assignmentdate date
1	5	Rajesh	Rana	Night Vision Goggles	SN11122	2023-05-01

6. Get Personnel who have had MedicalRecords for a specific condition:

```
SELECT P.PersonnelID, P.FName, P.LName,
M.MedicalCondition, M.TreatmentDetails

FROM Personnel P

JOIN MedicalRecords M ON P.PersonnelID =
M.PersonnelID

WHERE M.MedicalCondition = 'Fracture';
```

	personnelid integer	fname character varying (50)	lname character varying (50)	medicalcondition character varying (100)	treatmentdetails text
1	1	Arjun	Singh	Fracture	Casted and rest for 6 weeks
2	5	Rajesh	Rana	Fracture	Casted and rest for 6 weeks
3	9	Naveen	Yadav	Fracture	Casted and rest for 8 weeks

7.Retrieve the Performance Evaluations of Personnel for a specific date:

```
SELECT P.PersonnelID, P.FName, P.LName,
PE.EvaluationScore, PE.Comments

FROM Personnel P

JOIN PerformanceEvaluations PE ON P.PersonnelID =
PE.PersonnelID

WHERE PE.EvaluationDate = '30-09-2023';
```

	personnelid integer	fname character varying (50)	lname character varying (50)	evaluationscore integer	comments text
1	4	Sunil	Thakur	75	Needs improvement in communication

8. Get Top 3 Personnel details who have received the highest evaluation score:

```
SELECT P.PersonnelID, P.FName, P.LName,
PE.EvaluationScore

FROM Personnel P

JOIN PerformanceEvaluations PE ON P.PersonnelID =
PE.PersonnelID

ORDER BY PE.EvaluationScore DESC

LIMIT 3;
```

	personnelid integer	fname character varying (50)	lname character varying (50)	evaluationscore integer
1	7	Sakshi	Verma	92
2	3	Mahesh	Rawat	90
3	10	Priya	Deshmukh	89

9.Retrieve Personnel details who are on Annual Leave:

```
SELECT P.PersonnelID, P.FName, P.LName,
LR.LeaveType, LR.SDate, LR.EDate

FROM Personnel P

JOIN LeaveRecords LR ON P.PersonnelID =
LR.PersonnelID

WHERE LR.LeaveType = 'Annual Leave';
```

	personnelid integer	fname character varying (50)	lname character varying (50)	leavetype character varying (50)	sdate date	edate date
1	2	Neha	Rao	Annual Leave	2023-07-01	2023-07-15
2	5	Rajesh	Rana	Annual Leave	2023-10-01	2023-10-15
3	7	Sakshi	Verma	Annual Leave	2023-12-01	2023-12-15
4	10	Priya	Deshmukh	Annual Leave	2024-03-01	2024-03-10

10.Retrieve all Training Programs and the number of Personnel enrolled in each:

```
SELECT T.ProgramName, COUNT(TP.PersonnelID) AS
NumberOfPersonnel

FROM TrainingPrograms T

LEFT JOIN TrainingProgress TP ON T.TrainingProgramID
= TP.TrainingProgramID

GROUP BY T.ProgramName;
```

	programname character varying (100)	numberofpersonnel bigint
1	Officer Leadership Course	1
2	Signal Corps Training	1
3	Basic Training	1
4	Advanced Infantry Training	1
5	Medical Officer Training	1
6	Parachute Regiment Training	1
7	Special Forces Training	1
8	Jungle Warfare School	1
9	Artillery Training	1
10	Armored Corps Training	1

8.PL/PGSQL:

1.TRIGGER FOR INSERTION IN PERSONNEL TABLE:

- **TRIGGER FUNCTION:**

```
CREATE OR REPLACE FUNCTION
notify_new_personnel()

RETURNS TRIGGER AS $$

BEGIN

   RAISE NOTICE 'New Personnel Added: % %',
NEW.FName, NEW.LName;

   RETURN NEW;

END;

$$ LANGUAGE plpgsql;
```

- **TRIGGER:**

```
CREATE TRIGGER personnel_insert_trigger

AFTER INSERT ON Personnel

FOR EACH ROW EXECUTE FUNCTION
notify_new_personnel();
```

```
NOTICE:  New Personnel Added: Priyal Devgan
INSERT 0 1
```

2.TRIGGER FOR DELETION IN PERSONNEL TABLE:

- **TRIGGER FUNCTION:**

```
CREATE OR REPLACE FUNCTION notify_personnel_deletion()

RETURNS TRIGGER AS $$

BEGIN

   RAISE NOTICE 'Personnel Deleted: % %', OLD.FName, OLD.LName;

   RETURN OLD;

END;

$$ LANGUAGE plpgsql;
```

- **TRIGGER:**

```
CREATE TRIGGER personnel_delete_trigger

AFTER DELETE ON Personnel

FOR EACH ROW EXECUTE FUNCTION notify_personnel_deletion();
```

```
NOTICE:  Personnel Deleted: Priyal Devgan
DELETE 1
```