



IT - 214

DATABASE MANAGEMENT SYSTEMS

Title : - Military Database

Queries

Group Details

Name	ID
Chirag Chaudhari	202201219
Hemal Ravrani	202201235
Harsh Bosamiya	202201243
Marmik Vasava	202201252
Saumya Vasa	202201254

1. List all the personnel with their pension.

We add 3% more to the existing pension of a personnel if he is martyred and has two dependents, 3.5% more if he has 3 dependents and 4% more if he has 4 or more dependents

```
SELECT
    p.ID,
    p.Firstname || ' ' || p.Middlename || ' ' || p.Lastname AS
Fullname,
    p.Gender,
    p.DateofBirth,
    p.Rank,
    p.ServiceStatus,
    p.Martyred,
    w.Salary,
    COUNT(d.DependentName) AS NumDependents,
    CASE
        WHEN p.Martyred = 'yes' THEN
            CASE
                WHEN COUNT(d.DependentName) = 2 THEN (w.Salary) *
(w.rate + 3)/100
                WHEN COUNT(d.DependentName) = 3 THEN (w.Salary) *
(w.rate + 3.5)/100
                WHEN COUNT(d.DependentName) >= 4 THEN (w.Salary)
* (w.rate + 4)/100
                ELSE w.Salary * w.rate/100
            END
        ELSE
            w.Salary * w.rate/100

    END AS Pension
FROM
    Personnel p
JOIN
    Work w ON p.ServiceName = w.ServiceName
LEFT JOIN
    Dependents d ON p.ID = d.PersonnelID
```

Joins the result of the previous join with the 'Dependents' table based on the 'ID' column, ensuring that all personnel are included in the result, even if they don't have dependents.

GROUP BY

p.ID, p.Firstname, p.Middlename, p.Lastname, p.Gender,
p.DateofBirth, p.Rank, p.ServiceStatus, p.Martyred,
w.Salary,w.rate;

2. Retrieve all equipment stored in locations with a specific status:

```
SELECT Equipment.Name, Equipment_Storage.Quantity
FROM Equipment
JOIN Equipment_Storage
  ON Equipment.EquipID = Equipment_Storage.EquipID
JOIN Storage
  ON Equipment_Storage.Location = Storage.Location
WHERE Storage.Status = 'Full';
```

3. List all strategic locations in a specific war and compute its status i.e. whether they are engaged or lost.

```
SELECT
  SL.Location,
  W.Region AS War_Region,
  CASE
    WHEN SUM(CASE WHEN P.Status = 'engaged' THEN 1 ELSE 0
END) > 0 THEN 'engaged'
    WHEN COUNT(*) = SUM(CASE WHEN P.Status IN ('missing',
'POW', 'martyred') THEN 1 ELSE 0 END) THEN 'lost'
    ELSE 'unknown'
  END AS Status
FROM
  StrategicLocation SL
JOIN
  pers_is_inawar P ON SL.Location = P.Location
```

```

JOIN
    War W ON SL.Region = W.Region
WHERE
    W.Region = 'Jammu and Kashmir'
GROUP BY
    SL.Location, W.Region;

```

4. Find all equipment manufactured after a certain delivery date:

```

SELECT * FROM EquipmentManufactured WHERE DeliveryDate > '2024-
05-12';

```

5. List all vehicles used in logistics on a specific route and date:

```

SELECT LogisticsVehicles.Name, LogisticsVehicles.Type
FROM LogisticsVehicles
JOIN VehiclesusedinLogistics ON LogisticsVehicles.VehicleNo =
VehiclesusedinLogistics.VehicleNo
WHERE VehiclesusedinLogistics.Route = 'Fazilka to Doklam to
Petrapole' AND VehiclesusedinLogistics.Date = '2022-04-25';

```

6. We want 67 M4 Carbine", 78 "M9 Pistol", 10 " M249 SAW", 89 "MQ-9 Reaper Drone", 56 "PP-19 Bizon", 140 "RIM-174 Standard Missile", 36 "MP40" for a mission. Check how many are actually available and how many are less

```

WITH RequiredEquipment AS (
    SELECT *
    FROM (VALUES
        ('M4 Carbine', 67),
        ('M9 Pistol', 78),
        ('M249 SAW', 10),
        ('MQ-9 Reaper Drone', 89),
        ('PP-19 Bizon', 56),
        ('RIM-174 Standard Missile', 140),
        ('MP40', 36)
    ) AS t(Name, RequiredQuantity)

```

```

)

SELECT
    Equipment.Name,
    Equipment.Quantity AS AvailableQuantity,
    COALESCE(SUM(CASE WHEN Maintenance.EndDate > CURRENT_DATE
THEN Maintenance.Quantity ELSE 0 END), 0) AS
UnderMaintenanceQuantity,
    (Equipment.Quantity - COALESCE(SUM(CASE WHEN
Maintenance.EndDate > CURRENT_DATE THEN Maintenance.Quantity ELSE
0 END), 0)) AS RemainingQuantity,
    RequiredEquipment.RequiredQuantity AS RequiredQuantity,
    CASE
        WHEN (Equipment.Quantity - COALESCE(SUM(CASE WHEN
Maintenance.EndDate > CURRENT_DATE THEN Maintenance.Quantity ELSE
0 END), 0)) >= RequiredEquipment.RequiredQuantity THEN
'Available'
        ELSE 'Not Available'
    END AS Availability
FROM
    Equipment
JOIN
    RequiredEquipment ON Equipment.Name = RequiredEquipment.Name
LEFT JOIN
    Maintenance ON Equipment.EquipID = Maintenance.EquipID
GROUP BY
    Equipment.Name, Equipment.Quantity,
    RequiredEquipment.RequiredQuantity;

```

- 7. List the soldiers who have served in a mission and returned safely and also served in war and are engaged**

```

SELECT p.*
FROM Personnel p
JOIN PersonAssociatedwithMission pam ON p.ID = pam.PersonID
JOIN Mission m ON pam.MissionCode = m.MissionCode
JOIN pers_is_inawar piw ON p.ID = piw.ID
JOIN War w ON piw.Location = w.Region
WHERE pam.FinalStatusforpers = 'SafeReturn'
AND piw.Status = 'Safe';

```

8. Retrieve all missions led by a specific personnel:

```
SELECT * FROM Mission WHERE LeaderID = 35;
```

9. Retrieve all personnel who are currently available:

```
SELECT * FROM Personnel WHERE ServiceStatus = 'Available';
```

10. List all equipment currently stored in a specific location having a specific category and having quantity greater than a certain limit.

```
SELECT Equipment.Name, Equipment.Category,  
Equipment_Storage.Quantity  
FROM Equipment  
JOIN Equipment_Storage ON Equipment.EquipID =  
Equipment_Storage.EquipID  
WHERE Equipment_Storage.Location = 'Pathankot'  
AND equipment_storage.quantity >= 40  
AND category = 'Firearms'
```

11. Retrieve all personnel who are currently available:

```
SELECT * FROM Personnel WHERE ServiceStatus = 'Available';
```

12. We want to check if 67 "M4 Carbine", 78 "M9 Pistol", 10 "M249 SAW", 89 "MQ-9 Reaper Drone", 56 "PP-19 Bizon", 140 "RIM-174 Standard Missile", 36 "MP40" are available or not as they might be in maintenance and also we want to check if 3 "Boeing C-17", 2 "Ashok Leyland FAT 6x6" and 2 "INS Vikramaditya" are available for the logistics of the above equipments

```
WITH EquipmentAvailability AS (
```

```
SELECT
```

```
    e.Name AS Equipment_Name,
```

```
    e.Quantity AS Total_Quantity,
```

```

        SUM(m.Quantity) AS Quantity_In_Maintenance
FROM
    Equipment AS e
LEFT JOIN
    Maintenance AS m ON e.EquipID = m.EquipID

```

LEFT JOIN is used here to ensure that all rows from the 'Equipment' table (e) are included in the result set, even if there are no matching rows in the 'Maintenance' table (m).

```

WHERE
    e.Name IN ('M4 Carbine', 'M9 Pistol', 'M249 SAW', 'MQ-9 Reaper
Drone', 'PP-19 Bizon', 'RIM-174 Standard Missile', 'MP40')

```

```

    GROUP BY
        e.Name, e.Quantity
),
LogisticsVehiclesAvailability AS (

```

```

    SELECT
        Name AS Vehicle_Name,
        SUM(CASE WHEN Availability = 'Available' THEN 1 ELSE 0 END) AS
Total_Available

```

```

    FROM
        LogisticsVehicles

```

```

    WHERE
        Name IN ('Boeing C-17', 'Ashok Leyland FAT 6x6', 'INS
Vikramaditya')

```

```

    GROUP BY
        Name

```

```

)
SELECT

```

```

Equipment_Name,
CASE
    WHEN (Total_Quantity - Quantity_In_Maintenance) >=
        CASE
            WHEN Equipment_Name = 'M4 Carbine' THEN 67
            WHEN Equipment_Name = 'M9 Pistol' THEN 78
            WHEN Equipment_Name = 'M249 SAW' THEN 10
            WHEN Equipment_Name = 'MQ-9 Reaper Drone' THEN 89
            WHEN Equipment_Name = 'PP-19 Bizon' THEN 56
            WHEN Equipment_Name = 'RIM-174 Standard Missile'
THEN 140
            WHEN Equipment_Name = 'MP40' THEN 36
            ELSE 0
        END THEN 'Available'
    ELSE 'Not Available'
END AS Equipment_Availability,
NULL AS Vehicle_Name,
NULL AS Vehicle_Availability
FROM
    EquipmentAvailability

UNION ALL

SELECT
    NULL AS Equipment_Name,
    NULL AS Equipment_Availability,
    Vehicle_Name,
    CASE

```



```

        WHEN (CASE
            WHEN Vehicle_Name = 'Boeing C-17' THEN 3
            WHEN Vehicle_Name = 'Ashok Leyland FAT 6×6' THEN 2
            WHEN Vehicle_Name = 'INS Vikramaditya' THEN 2
            ELSE 0
        END - Total_Available) >= 0 THEN 'Available'
        ELSE 'Not Available'
    END AS Vehicle_Availability
FROM
    LogisticsVehiclesAvailability;

```

- 13. List all logistics routes transferred by a specific unit on a particular date:**

```

SELECT Route, ResourcesTransferred
FROM Logistics
WHERE UnitName = 'Navy' AND Date = '2022-04-11';

```

- 14. List all units and their respective commanding officers:**

```

SELECT Unit.UnitName, Personnel.Firstname, Personnel.Lastname
FROM Unit
JOIN Personnel ON Unit.CommandingOfficerID = Personnel.ID;

```

- 15. Find all personnel who received awards in a specific year:**

```

SELECT Personnel.Firstname, Personnel.Lastname, Awarded.AwardName
FROM Personnel
JOIN Awarded
    ON Personnel.ID = Awarded.PersonnelID
WHERE Awarded.Year = 2023;

```

- 16. List all maintenance records for a particular equipment:**

```

SELECT * FROM Maintenance WHERE EquipID = 7;

```

- 17. Retrieve all personnel associated with a specific mission:**

```
SELECT Personnel.Firstname, Personnel.Lastname
FROM Personnel
JOIN
PersonAssociatedwithMission
ON Personnel.ID = PersonAssociatedwithMission.PersonID
WHERE PersonAssociatedwithMission.MissionCode = 7;
```

- 18. Find the total quantity of a specific equipment used by all units and also include how much quantity of equipment each unit uses**

```
SELECT
    UnitName,
    SUM(Quantity) AS TotalQuantityUsed
FROM
    UnitusesEquipment
WHERE
    EquipID = 31
GROUP BY
    UnitName;
```

- 19. List all strategic locations in a particular war region:**

```
SELECT * FROM StrategicLocation
WHERE Region = 'Jammu and Kashmir';
```

- 20. List all strategic locations in a specific war and compute its status i.e. whether they are engaged or lost.**

```
SELECT
    SL.Location,
    W.Region AS War_Region,
    CASE
        WHEN SUM(CASE WHEN P.Status = 'engaged' THEN 1 ELSE 0
END) > 0 THEN 'engaged'
        WHEN COUNT(*) = SUM(CASE WHEN P.Status IN ('missing',
'POW', 'martyred') THEN 1 ELSE 0 END) THEN 'lost'
        ELSE 'unknown'
```

```

        END AS Status
FROM
    StrategicLocation SL
JOIN
    pers_is_inawar P ON SL.Location = P.Location
JOIN
    War W ON SL.Region = W.Region
WHERE
    W.Region = 'Jammu and Kashmir'
GROUP BY
    SL.Location, W.Region;

```

21. Retrieve all personnel who were were martyred in war:

```

SELECT P.*
FROM Personnel P
JOIN pers_is_inawar PIW ON P.ID = PIW.ID
WHERE PIW.Status = 'martyred';

```

22. Find out the budget used and divide into the budget used for giving salary pension and used for equipments and then give remaining budget

```

WITH SalaryPerPerson AS (
    SELECT
        p.UnitName,
        SUM(w.Salary) AS TotalSalaryPerPerson
    FROM
        Personnel p
    JOIN
        Work w ON p.ServiceName = w.ServiceName
    GROUP BY
        p.UnitName, p.ID
),
TotalSalaryPerUnit AS (
    SELECT
        UnitName,
        SUM(TotalSalaryPerPerson) AS TotalSalary
    FROM
        SalaryPerPerson
    GROUP BY
        UnitName

```

```

),
EquipmentCostPerUnit AS (
    SELECT
        u.UnitName,
        SUM(e.PricePerUnit * ue.Quantity) AS TotalEquipmentCost
    FROM
        Unit u
    JOIN
        UnitusesEquipment ue ON u.UnitName = ue.UnitName
    JOIN
        Equipment e ON ue.EquipID = e.EquipID
    GROUP BY
        u.UnitName
)
SELECT
    u.UnitName,
    u.BudgetAllocated AS TotalBudget,
    ts.TotalSalary AS TotalSalary,
    ec.TotalEquipmentCost AS TotalEquipmentCost,
    (u.BudgetAllocated - ts.TotalSalary - ec.TotalEquipmentCost)
AS RemainingBudget
FROM
    Unit u
JOIN
    TotalSalaryPerUnit ts ON u.UnitName = ts.UnitName
JOIN
    EquipmentCostPerUnit ec ON u.UnitName = ec.UnitName;

```

23. Retrieve the commanding officer of a specific unit:

```

SELECT Personnel.*
FROM Personnel
JOIN Unit ON Personnel.ID = Unit.CommandingOfficerID
WHERE Unit.UnitName = 'Army';

```

24. Give total number of martyred, missing, pow and engaged soldiers for each strategic location for each war and hence give the total number of remaining alive soldiers also for that location

```

SELECT
    SL.Location,
    W.Region AS War_Region,
    SUM(CASE WHEN P.Status = 'martyred' THEN 1 ELSE 0 END) AS
Martyred,
    SUM(CASE WHEN P.Status = 'missing' THEN 1 ELSE 0 END) AS
Missing,
    SUM(CASE WHEN P.Status = 'pow' THEN 1 ELSE 0 END) AS POW,
    SUM(CASE WHEN P.Status = 'engaged' THEN 1 ELSE 0 END) AS
Engaged,
    SUM(CASE WHEN P.Status NOT IN ('martyred', 'missing', 'pow')
THEN 1 ELSE 0 END) AS Remaining_Alive
FROM
    StrategicLocation SL
LEFT JOIN
    pers_is_inawar P ON SL.Location = P.Location
LEFT JOIN
    War W ON SL.Region = W.Region
GROUP BY
    SL.Location, W.Region;

```

- 25. Retrieve all intelligence records used in a specific mission:**

```

SELECT IntelligenceRecords.IntelRecordName
FROM IntelligenceRecords
JOIN MissionUsesIntelligenceRecords ON
IntelligenceRecords.IntelRecordName =
MissionUsesIntelligenceRecords.Name
WHERE MissionUsesIntelligenceRecords.MissionCode = 3;

```

- 26. Find the total number of personnel dependents for each personnel:**

```

SELECT Personnel.ID, COUNT(Dependents.DependentName) AS
DependentCount
FROM Personnel
LEFT JOIN Dependents ON Personnel.ID = Dependents.PersonnelID
GROUP BY Personnel.ID;

```

- 27. Give unitwise allocation of budget, also include the total budget and the percentage of total budget each unit gets**

```
WITH TotalBudget AS (  
    SELECT SUM(BudgetAllocated) AS TotalBudget  
    FROM Unit  
)  
  
SELECT  
    u.UnitName,  
    u.BudgetAllocated,  
    tb.TotalBudget,  
    ROUND((u.BudgetAllocated * 100.0 / tb.TotalBudget), 2) AS  
    PercentageOfTotalBudget  
FROM  
    Unit u  
CROSS JOIN  
    TotalBudget tb;
```

- 28. Retrieve all personnel who participated in training camps with special abilities found:**

```
SELECT Personnel.Firstname, Personnel.Lastname  
FROM Personnel  
JOIN TrainingRecords  
ON Personnel.ID = TrainingRecords.PersonnelID  
WHERE TrainingRecords.SpecialAbilityFound IS NOT NULL;
```

- 29. List all equipment ordered by a particular unit:**

```
SELECT EquipmentManufactured.EquipName  
FROM EquipmentManufactured  
JOIN UnitOrderedEquipment ON EquipmentManufactured.EquipName =  
UnitOrderedEquipment.EquipName  
WHERE UnitOrderedEquipment.UnitName = 'Army';
```