Operation Archive:

Developing a Comprehensive Military Database for Efficient Data Management and Analysis

– –

By: Caden William Coutz

Table of Contents

Overview

The United States Military is a complex and intricate system that is responsible for protecting and defending our country. Composed of multiple branches, including the Army, Navy, Air Force, Marines, and Coast Guard, the Military is responsible for maintaining the safety and security of the United States both at home and abroad.

The men and women who serve in the Military work tirelessly to ensure that our nation remains safe from harm. From combat operations in overseas theaters of war to humanitarian missions in times of crisis, the Military is always ready to respond to any situation that may arise.

However, with such a vast and complex system, it can be challenging to keep track of every aspect of the Military. This is where the power of a comprehensive database comes in. By using a database, Military leaders and personnel can manage information about troops, missions, equipment, and more in a centralized location.

A database can also help to streamline communication and collaboration between different branches of the Military, ensuring that everyone has access to the information they need to make informed decisions. Additionally, it can help leaders to identify potential gaps or areas for improvement, allowing them to make necessary changes to enhance the efficiency and effectiveness of the Military.

Overall, the use of a database in the Military is crucial to ensure that our nation's defenders have the tools and information they need to protect our freedoms and maintain our security. By leveraging the power of technology, we can enhance the capabilities of the Military and ensure that it remains a reliable and effective force for years to come.

Entity Set Descriptions

Branches

This part of the database provides a comprehensive overview of the five branches of the United States military, including the Army, Navy, Air Force, Marine Corps, and Coast Guard. For each branch, the table lists important information such as the official name, year established, headquarters location, motto, approximate number of active duty personnel, and primary responsibilities. The descriptions of each attribute provide insight into the history, purpose, and mission of each branch, highlighting the unique contributions that they make to national security and defense. Overall, this table serves as a useful reference tool for anyone seeking to learn more about the various branches of the United States military and their role in safeguarding the country.

Military Bases

Another big part of this Military database holds records of all bases within each branch. Each base will also have a unique id number because some bases share the same name. One base can serve as the home to multiple branches of the Armed Forces. Each branch will have a recorded location. No base can ever have more than one branch assigned to it. There will also be records of the number of people who both work and/or live on base. There will also be a record of when the base was established and where it is located. The personnel section of this database will reference this section when assigning personnel to various military bases.

Deployment

Deployment is an essential part of the armed forces. This database will also track any deployment assigned to any personnel. A deployment will be recorded with a start date and an end date and must have an exact start date, but the end date can be not as exact if the end of the deployment is not known yet. Each deployment must also have a location, but it can have more than one location if needed. Each deployment will reference at least one person, if not more than one deployment will be referenced and assigned in the personnel section of this database.

Assignments

This database will also be made up of information about every assignment created and given to any military personnel. Each assignment will have at least one assigned location or base but may include several different bases, depending on the complexity of the assignment. Each assignment will be assigned at least one unit but can have multiple if needed. This database will also hold information in terms of the length of time each assignment takes. The locations of the assignments will be a reference to military bases within the database. Each assignment will also indicate what type of assignment it is, and there are about 7 to choose from.

Personnel Information

This database would accompany all information referring to personal information about each person who is serving in the Armed Forces. Each person can only serve in a single branch of the military at one time. Each person will only be assigned one rank at any particular time. Within this database, we will also be able to identify the needed contact information of each person enlisted in the Military. There will be a record of what base each person is assigned to and will reference another part of the database. Each person will be assigned to various deployments and these will be another reference to a different part of the database. A person can only be assigned to one deployment at a time. Each training that a person performs will also be recorded

Medical Records

The military database will also hold medical information about each one of the service men and women. This database will keep track of any sickness or injury that any person may have encountered when they were enlisted into the armed forces. Information about treatments performed will also be recorded within this database. Each medical record will be assigned to only one person, referenced by the personnel section of the database. Each medical record can only be linked to one person. There will also be records of ongoing conditions that will be documented as well in the database. This database will also hold information as to different allergies a person may have and what medication they are on if any. A record of each person's blood type will be contained in this section of the database as well.

Expenses

Another portion of this database is used to keep track of expenses incurred by military service members in various categories. It helps the military maintain financial records and ensure that expenses are made for legitimate purposes and in compliance with military regulations. The table may be used to identify areas where costs are higher than expected or where cost-saving measures may be implemented. It may also be used to track expenses across different military units and to ensure that expenses are allocated fairly across the military.

Equipment

Equipment will also be recorded and kept within this database. Each piece of equipment will have its unique serial number to keep track of it and no two pieces of equipment can have the same serial number. The database will record whoever has used the equipment and if no one is in current use of a piece of the equipment it needs to be recorded as such and need to update as soon as it is given or returned. There must be a date of when it is issued if the status of any equipment is “in use”, this cannot be left blank. Each piece of equipment had a record of who had the equipment

This will be split up into two smaller sections one to inventory all of the equipment and then there will be a different section to record all the checks out each piece of equipment has gone through. This will reference the personnel section to track who has any piece of equipment at any given point.

Training

This database will also include information relating to training that is performed by the military in every branch and it will specify what kind of training it was that was performed. It will also identify who has done any of the training, and how long it took to do each training. This database will also hold information as to where and when the training was conducted, as well.

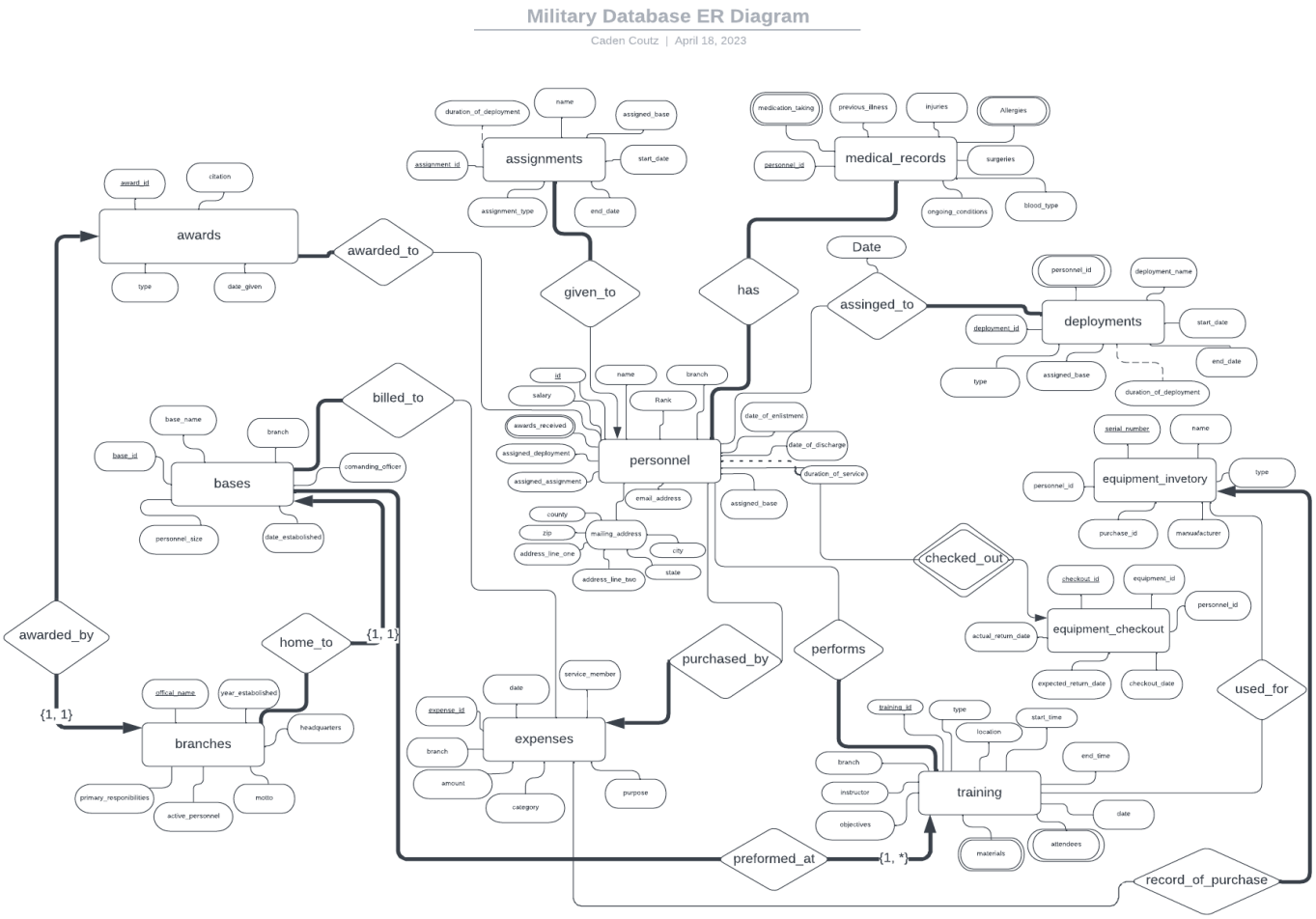
Equipment that is used during training will also be recorded and will reference another part of the database. Attendance is taken during the pieces of training and will also be recorded with reference personnel from a different part of the database. The database will also have a record of the instructor or instructors for each of the pieces of training and the instructors will be a reference to the personnel database. Instructors will have to be a certain rank to lead a training, referenced also by the personnel section of the database of training, and will also have a branch of the military assigned to it, based on who is conducting the training.

Awards and Honors

This Database will also contain information about all possible awards and honors that exist and that can be given to personnel in the military. Each award will be also described in the database as well. This database will also have a record of when the award was given or presented. Personnel can have one award or multiple awards. Not every person in the Armed Forces has to have an award or honor either. This will reference awards noted in the personnel section of the database.

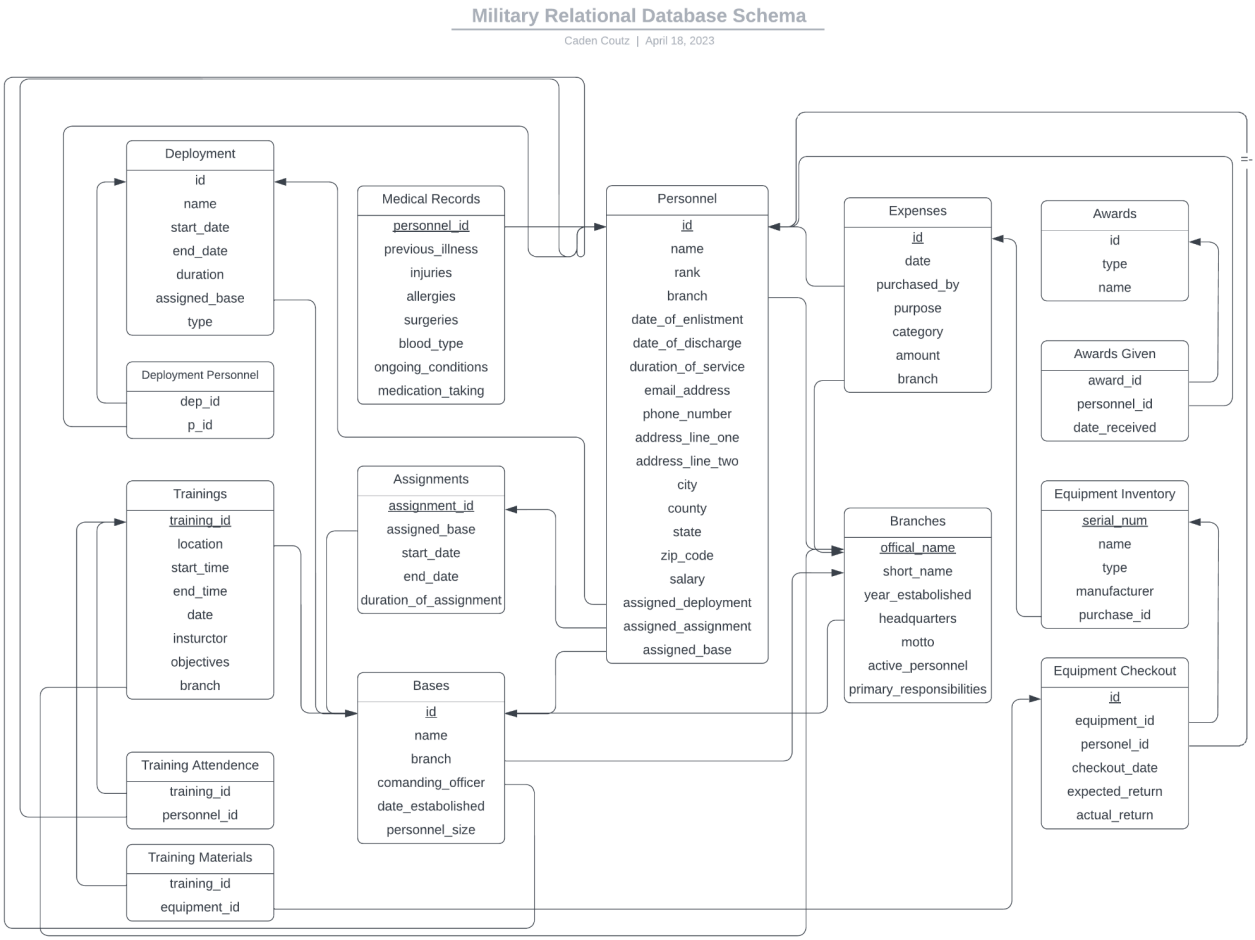
The ER Diagram

Below is the representation of the intended military database in the form of an ER diagram.



The Relational Diagram

Down below is a relational diagram that also represents the intended database. This is a more comprehensive diagram of the database, as there is less going on. Each box represents a table in the database and the lines and arrows represent attributes that reference other attributes in other tables.



Creating the Database in SQL

Implementation of The Database Tables

```sql
CREATE TABLE branch (
    official_name            varchar(30),
    short_name               varchar(12),
    date_established         varchar(10),
    headquarters             varchar(8)   DEFAULT 'KY-45692',
    motto                    varchar(45)  DEFAULT 'To Serve and Protect',
    active_personnel         int,
    primary_responsibilities varchar(50),
    CONSTRAINT branch_pkey   PRIMARY KEY (short_name)
);
```

```sql
CREATE TABLE base (
    id                 varchar(9),  -- 'KY-45692'
    name               varchar(40),
    branch             varchar(12),
    commanding_officer varchar(6),
    date_established   varchar(10),
    personnel_size     int,
    CONSTRAINT base_pkey  PRIMARY KEY (id),
    CONSTRAINT base_fkey1 FOREIGN KEY (branch) REFERENCES branch(short_name)
                                ON DELETE SET DEFAULT
                                ON UPDATE CASCADE
);
```

```sql
CREATE TABLE deployment (
    id             varchar(6),  -- such as 'AR-123'
    name           varchar(30),
    start_date     varchar(10),
    end_date       varchar(10),
    assigned_base  varchar(9)  DEFAULT 'KY-92362',
    CONSTRAINT deployment_pkey PRIMARY KEY (id),
    CONSTRAINT personnel_fkey4 FOREIGN KEY (assigned_base) REFERENCES base(id)
                            ON DELETE SET DEFAULT
                            ON UPDATE CASCADE
);
```

```
CREATE TABLE assignment (
    assignment_id  varchar(6),
    name           varchar(30),
    assigned_base  varchar(9),
    start_date     varchar(10),
    end_date       varchar(10),
    CONSTRAINT assignment_pkey PRIMARY KEY (assignment_id),
    CONSTRAINT assignment_fkey FOREIGN KEY (assigned_base) REFERENCES base(id)
                               ON DELETE SET NULL
                               ON UPDATE CASCADE
);
```

```
CREATE TABLE personnel (
    id                     varchar(6),
    DOB                    varchar(10),
    name                   varchar(30)     NOT NULL,
    rank                   varchar(30)     DEFAULT 'Private',
    branch                 varchar(12)     DEFAULT 'Army',
    date_of_enlistment     varchar(10)     NOT NULL,
    date_of_discharge      varchar(18)     DEFAULT 'Current',
    duration               varchar(18)     DEFAULT 'Current',
    address_line_1         varchar(60),
    address_line_2         varchar(60),
    city                   varchar(50),
    state                  varchar(2),
    zip code                varchar(10),
    country                varchar(30),
    assigned_deployment    varchar(40)     DEFAULT 'AR-123',
    assigned_assignment    varchar(40)     DEFAULT 'A00001',
    assigned_base          varchar(30)     DEFAULT 'KY-92362',
    CONSTRAINT personnel_pkey PRIMARY KEY (id),
    CONSTRAINT personnel_fkey1 FOREIGN KEY (branch) REFERENCES branch(short_name)
                               ON DELETE SET DEFAULT
                               ON UPDATE CASCADE,
    CONSTRAINT personnel_fkey2 FOREIGN KEY (assigned_deployment) REFERENCES deployment(id)
                           ON DELETE SET DEFAULT
                           ON UPDATE CASCADE,
    CONSTRAINT personnel_fkey3 FOREIGN KEY (assigned_assignment) REFERENCES assignment(assignment_id)
                          ON DELETE SET DEFAULT
                          ON UPDATE CASCADE,
    CONSTRAINT personnel_fkey4 FOREIGN KEY (assigned_base) REFERENCES base(id)
                               ON DELETE SET DEFAULT
                               ON UPDATE CASCADE
);
```

```
CREATE TABLE medical_record (
    id                  varchar(6),
    previous_illness    varchar(60),
    injuries            varchar(60),
    allergies           varchar(60),
    surgeries           varchar(60),
    blood_type          varchar(8),
    ongoing_conditions  varchar(60),
    medication_taking   varchar(60),
    CONSTRAINT medical_record_pkey PRIMARY KEY (id),
    CONSTRAINT medical_record_fkey1 FOREIGN KEY (id) REFERENCES personnel(id)
                                    ON DELETE CASCADE
                                    ON UPDATE CASCADE
);
```

```
CREATE TABLE expense (
  id           varchar(6),
  date         varchar(10),
  purchased_by varchar(6),
  purpose      varchar(40),
  category     varchar(15) CHECK (category IN ('Technology', 'Weapons', 'Uniforms',
                                               'Maintenance', 'Food', 'Survival',
                                               'Other')),
  amount       numeric(8,2),
  branch       varchar(12) DEFAULT 'Army',
  CONSTRAINT expense_pkey PRIMARY KEY (id)
  CONSTRAINT expense_pkey PRIMARY KEY (id),
  CONSTRAINT expense_fkey1 FOREIGN KEY (purchased_by) REFERENCES personnel(id)
                           ON DELETE CASCADE
                           ON UPDATE CASCADE,
  CONSTRAINT expense_fkey2 FOREIGN KEY (branch) REFERENCES branch(short_name)
                           ON DELETE SET DEFAULT
                           ON UPDATE CASCADE
);
```

```
CREATE TABLE equipment (
    serial_num   varchar(6),
    name         varchar(15),
    type         varchar(15) CHECK (category IN('Technology', 'Weapons',
                                                'Uniforms', 'Maintenance',
                                                'Food', 'Survival', 'Other')),

    manufacturer varchar(20),
    purchase_id  varchar(6),
    CONSTRAINT equipment_pkey PRIMARY KEY (serial_num),
    CONSTRAINT equipment_fkey1 FOREIGN KEY (purchase_id)
        REFERENCES expense(id) ON DELETE SET NULL ON UPDATE CASCADE
);
```

```
CREATE TABLE equipment_checkout (
    id              varchar(6),
    equipment_id    varchar(6),
    personnel_id    varchar(6),
    checkout_date   varchar(10),
    expected_return varchar(10),
    actual_return   varchar(10),
    CONSTRAINT equipment_checkout_pkey PRIMARY KEY (id),
    CONSTRAINT equipment_checkout_fk1 FOREIGN KEY (equipment_id) REFERENCES equipment(serial_num)
                                ON DELETE CASCADE
                                ON UPDATE CASCADE,
    CONSTRAINT equipment_checkout_fk2 FOREIGN KEY (personnel_id) REFERENCES personnel(id)
                                   ON DELETE CASCADE
                                   ON UPDATE CASCADE
);
```

```
CREATE TABLE training (
    id           varchar(6),
    location     varchar(9),
    start_date   varchar(10),
    end_date     varchar(12) DEFAULT 'In Progress',
    instructor   varchar(6),
    objectives   varchar(50),
    branch       varchar(12) DEFAULT 'Army',
    CONSTRAINT training_pkey PRIMARY KEY (id),
```

```
    CONSTRAINT training_fk1 FOREIGN KEY (branch) REFERENCES branch(short_name)
                                ON DELETE SET DEFAULT
                                ON UPDATE CASCADE,
    CONSTRAINT training_fk2 FOREIGN KEY (instructor) REFERENCES personnel(id)
                                ON DELETE SET DEFAULT
                                ON UPDATE CASCADE
);
```

```
CREATE TABLE training_attendance (
    training_id  varchar(6),
    personnel_id varchar(6),
    CONSTRAINT training_attendance_pkey PRIMARY KEY (training_id, personnel_id),
    CONSTRAINT training_attendance_fk1 FOREIGN KEY (training_id) REFERENCES training(id)
                                    ON DELETE CASCADE
                                    ON UPDATE CASCADE,
    CONSTRAINT training_attendance_fk2 FOREIGN KEY (personnel_id) REFERENCES personnel(id)
                                    ON DELETE CASCADE
                                    ON UPDATE CASCADE
);
```

```
CREATE TABLE training_materials (
    training_id  varchar(6),
    equipment_id varchar(6),
    CONSTRAINT training_materials_pkey PRIMARY KEY (training_id, equipment_id),
    CONSTRAINT training_materials_fk1 FOREIG N KEY (training_id) REFERENCES training(id)
                                    ON DELETE CASCADE
                                    ON UPDATE CASCADE,
    CONSTRAINT training_materials_fk2 FOREIGN KEY (equipment_id) REFERENCES equipment(serial_num)
                                    ON DELETE CASCADE
                                    ON UPDATE CASCADE
);
```

```
CREATE TABLE award (
    id   varchar(6),
    type varchar(10),
    name varchar(40),
    CONSTRAINT award_pkey PRIMARY KEY (id)
);
```

```
CREATE TABLE awards_given (
    award_id       varchar(6),
    personnel_id   varchar(6),
    date_received varchar(10),
    CONSTRAINT awards_given_pkey PRIMARY KEY (award_id, personnel_id),
    CONSTRAINT awards_given_fk1 FOREIGN KEY (award_id) REFERENCES award(id)
                                ON DELETE SET NULL
                                ON UPDATE CASCADE,
    CONSTRAINT awards_given_fk2 FOREIGN KEY (personnel_id) REFERENCES personnel(id)
                                ON DELETE SET NULL
                                ON UPDATE CASCADE
);
```

Alter Table Statements

Adding Additional Foreign Keys

When creating the tables, I had to leave out a couple of foreign keys in order for everything to be created. Below is the code to add in the remaining foreign keys.

```
ALTER TABLE branch
ADD CONSTRAINT branch_fkey FOREIGN KEY (headquarters) REFERENCES base(id)
                           ON DELETE SET DEFAULT
                           ON UPDATE SET DEFAULT;
```

```
ALTER TABLE base
ADD CONSTRAINT base_fkey2 FOREIGN KEY (commanding_officer) REFERENCES personnel(id)
                        ON DELETE SET NULL
                        ON UPDATE CASCADE;
```

Altering Data Types

These have been fixed in the code above, but originally I had the wrong data type assigned to a few attributes and then the alter statement to fix them to be the correct data types.

```
ALTER TABLE expense
ALTER COLUMN branch TYPE varchar(12);
```

```
ALTER TABLE expense
ALTER COLUMN amount TYPE numeric(8,2) USING amount::numeric(8,2);
```

```
ALTER TABLE personnel
ALTER COLUMN dob TYPE date;
```

Insert Statements

```
INSERT INTO branch (official_name, short_name, date_established, active_personnel, primary_responsibilities)
VALUES ('United States Army', 'Army', '1775-06-14', 480000, 'Land warfare');

INSERT INTO branch (official_name, short_name, date_established, active_personnel, primary_responsibilities)
VALUES('United States Navy', 'Navy', '1775-05-13', 332000, 'Naval warfare');

INSERT INTO branch (official_name, short_name, date_established, active_personnel, primary_responsibilities)
VALUES('United States Air Force', 'Air Force', '1947-09-18', 333000, 'Aerial warfare');

INSERT INTO branch (official_name, short_name, date_established, active_personnel, primary_responsibilities)
VALUES('United States Coast Guard', 'Coast Guard', '1790-08-14', 42000, 'Maritime law enforcement');

INSERT INTO branch (official_name, short_name, date_established, active_personnel, primary_responsibilities)
VALUES('United States Marine Corps', 'Marines', '1775-11-10', 186000, 'Amphibious warfare');
```

```
-- United States Army
INSERT INTO base (id, name, branch, commanding_officer, date_established, personnel_size)
VALUES ('KY-92362', 'Fort Knox', 'Army', '567890', '1942-01-16', 32000);

-- United States Air Force
INSERT INTO base (id, name, branch, commanding_officer, date_established, personnel_size)
VALUES ('VA-49261', 'Naval Station Norfolk', 'Air Force', '234567', '1947-09-18', 15000);

-- Insert a record for a military base in the United States Marine Corps
INSERT INTO base (id, name, branch, commanding_officer, date_established, personnel_size)
VALUES ('VA-43875', 'Henderson Hall', 'Marines', '456789', '1941-02-01', 47000);

-- Insert a record for a military base in the United States Coast Guard
INSERT INTO base (id, name, branch, commanding_officer, date_established, personnel_size)
VALUES ('OH-36945', 'Coast Guard Sector Ohio Valley', 'Coast Guard', '678901', '1790-08-04', 100);

INSERT INTO base (id, name, branch, commanding_officer, date_established, personnel_size)
       -- new
VALUES ('NY-53786', 'Coast Guard Sector New York', 'Coast Guard', '93531', '2005-07-22', 14000);

-- Navy base
INSERT INTO base (id, name, branch, commanding_officer, date_established, personnel_size)
VALUES ('WA-98312', 'Naval Base Kitsap', 'Navy', '678901','1917-11-14', 17000);
```

```sql
-- Navy Base
INSERT INTO base (id, name, branch, commanding_officer, date_established, personnel_size)
VALUES ('HI-56392', 'Pearl Harbor', 'Navy', '456789', '2010-05-01', 54000);
-- Army Base
INSERT INTO base (id, name, branch, commanding_officer, date_established, personnel_size)
VALUES ('NC-56392', 'Base Bragg', 'Army', '567890', '1963-05-01', 35000);
```

```sql
INSERT INTO deployment (id, name, start_date, end_date,  assigned_base)
VALUES ('AR-123', 'Operation XYZ', '2023-01-01', '2023-06-30', 'KY-92362');

INSERT INTO deployment (id, name, start_date, end_date)
VALUES ('NV-456', 'Training Exercise ABC', '2023-04-15', '2023-04-30');

INSERT INTO deployment (id, name, start_date, end_date, assigned_base)
VALUES ('CG-789', 'Logistics Support', '2023-03-01', '2023-09-01', 'OH-36945' );

INSERT INTO deployment (id, name, start_date, end_date, assigned_base)
VALUES ('MC-012', 'Surveillance Mission', '2023-06-01', '2023-06-30', 'VA-43875');

INSERT INTO deployment (id, name, start_date, end_date, assigned_base)
VALUES ('AF-345', 'Engineering Project', '2023-02-15', '2023-12-15', 'VA-49261');

INSERT INTO deployment (id, name, start_date, end_date, assigned_base)
VALUES ('AF-012', 'Operation Phoenix', '2023-06-01', '2023-09-30', 'VA-49261');

INSERT INTO deployment (id, name, start_date, end_date, assigned_base)
VALUES ('MC-345', 'Operation Cyclone', '2024-01-01', '2024-03-31', 'VA-43875');

INSERT INTO deployment (id, name, start_date, end_date, assigned_base)
VALUES ('NV-678', 'Operation Avalanche', '2024-04-15', '2024-06-30', 'HI-56392');

INSERT INTO deployment (id, name, start_date, end_date)
VALUES ('AR-901', 'Operation Typhoon', '2024-07-01', '2024-09-30');

INSERT INTO deployment (id, name, start_date, end_date, assigned_base)
VALUES ('NV-234', 'Operation Hurricane', '2024-10-01', '2024-12-31', 'WA-98312');
```

```sql
INSERT INTO assignment (assignment_id, name, assigned_base, start_date, end_date)
VALUES ('A00001', 'Naval Officer', 'KY-92362', '2022-01-01', '2022-06-30');

INSERT INTO assignment (assignment_id, name, assigned_base, start_date)
VALUES ('A00002', 'Artilleryman', 'VA-49261', '2022-05-15');
```

```
INSERT INTO assignment (assignment_id, name, assigned_base, start_date, end_date)
VALUES ('A00003', 'Logistics Officer', 'VA-43875', '2023-01-01', '2023-12-31');

INSERT INTO assignment (assignment_id, name, assigned_base, start_date, end_date)
VALUES ('A00004', 'Cyber Warfare Specialist', 'OH-36945', '2022-07-01', '2022-12-31');

INSERT INTO assignment (assignment_id, name, assigned_base, start_date)
VALUES ('A00005', 'Chaplin', 'WA-98312', '2022-08-15');

INSERT INTO assignment (assignment_id, name, assigned_base, start_date)
VALUES ('A00006', 'Intelligence Analyst', 'HI-56392', '2022-09-01');

INSERT INTO assignment (assignment_id, name, assigned_base, start_date, end_date)
VALUES ('A00007', 'Helicopter Pilot', 'NC-56392', '2022-10-01', '2023-03-31');
```

```
-- Insert a record for a person who enlisted in the United States Marine Corps
INSERT INTO personnel (id, DOB, name, rank, branch, date_of_enlistment, date_of_discharge, duration,
address_line_1, city, state, zipcode, country,  assigned_deployment, assigned_assignment, assigned_base)
VALUES ('456789', '1997-11-22', 'Emily Brown', 'Sergeant', 'Marines', '2010-08-12', '2022-12-31', '12
years','987 Oak St', 'Sometown', 'FL', '45678', 'USA', 'MC-012', 'A00001', 'VA-43875');

-- Insert a record for a person with a null value for date_of_discharge
INSERT INTO personnel (id, DOB, name, rank, branch, date_of_enlistment, address_line_1, city, state,
zipcode, country, assigned_deployment, assigned_assignment, assigned_base)
VALUES ('567890', '1998-04-10', 'Michael Lee', 'Private', 'Army', '2016-09-01', '345 Cedar St', 'Reading',
'CA', '12345', 'USA', 'AR-123', 'A00004', 'KY-92362');

-- Insert a record for a person with a null value for assigned_deployment
INSERT INTO personnel (id, DOB, name, rank, branch, date_of_enlistment, address_line_1, city, state,
zipcode, country, assigned_deployment, assigned_assignment, assigned_base)
VALUES ('678901', '1993-09-19', 'Ashley Kim', 'Comander', 'Coast Guard', '2012-02-20', '567 Maple St',
'Sometown', 'WA', '45678', 'USA', 'CG-789', 'A00006', 'NY-53786'); -- add the last two things

-- Insert a record for a person who enlisted in the United States Air Force
INSERT INTO personnel (id, DOB, name, rank, branch, date_of_enlistment, address_line_1, city, state,
zipcode, country, assigned_deployment, assigned_assignment, assigned_base)
VALUES ('345677', '1995-06-30', 'David Lee', 'Lieutenant Colonel', 'Air Force', '2017-09-01', '789 Maple
Ave', 'Anycity', 'TX', '45678', 'USA', 'AF-012', 'A00004','VA-49261'); -- add last two things

-- Insert a record for a person who enlisted in the United States Marine Corps
INSERT INTO personnel (id, DOB, name, rank, branch, date_of_enlistment, date_of_discharge, duration,
address_line_1, city, state, zipcode, country, assigned_deployment, assigned_assignment, assigned_base)
VALUES ('43576', '1993-11-12', 'Amy Chen', 'Gunnery Sergeant', 'Marines', '2012-07-01', '2020-07-01', '8
years', '321 Oak St', 'Somewhere', 'CA', '12345', 'USA', 'MC-345', 'A00003','VA-43875');

-- Insert a record for a person who enlisted in the United States Coast Guard
INSERT INTO personnel (id, DOB, name, rank, branch, date_of_enlistment, address_line_1, city, state,
zipcode, country, assigned_deployment, assigned_assignment, assigned_base)
VALUES ('58632', '1998-08-25', 'Mark Johnson', 'Petty Officer Second Class', 'Coast Guard', '2016-06-15',
'456 Pine St', 'Anothercity', 'FL', '67890', 'USA', 'CG-789', 'A00003', 'OH-36945');
```

```
-- Insert a record for a person who enlisted in the British Army
INSERT INTO personnel (id, DOB, name, rank, branch, date_of_enlistment, address_line_1, city, state,
zipcode, country, assigned_assignment, assigned_base)
VALUES ('93531',' 1992-03-17', 'William Davies', 'Major', 'Army', '2010-09-01', '789 Chestnut St',
'Savanah', 'GA', '35629', 'USA', 'A00005','KY-92362');

-- Insert a record for a person who enlisted in the Royal Navy
INSERT INTO personnel (id, DOB, name, rank, branch, date_of_enlistment, date_of_discharge, duration,
address_line_1, city, state, zipcode, country, assigned_deployment, assigned_assignment, assigned_base)
VALUES ('789012', '1997-12-24', 'Sophie Brown', 'Leading Seaman', 'Navy', '2016-01-15', '2022-07-21', '8
years', '234 Oak Rd', 'Nashville', 'TN', '13457', 'USA', 'NV-234', 'A00002', 'HI-56392');
```

```
INSERT INTO medical_record(id, previous_illness, injuries, allergies, surgeries, blood_type,
ongoing_conditions, medication_taking)
VALUES ('456789', 'Pneumonia', 'None', 'Penicillin', 'Appendectomy', 'AB+', 'Asthma', 'Albuterol');

INSERT INTO medical_record(id, previous_illness, injuries, allergies, surgeries, blood_type,
ongoing_conditions, medication_taking)
VALUES ('567890', 'Migraine', 'Sprained ankle', 'None', 'None', 'O-', 'Depression',
'Sertraline');

INSERT INTO medical_record(id, previous_illness, injuries, allergies, surgeries, blood_type,
ongoing_conditions, medication_taking)
VALUES ('678901', 'Bronchitis', 'Broken arm', 'Dust', 'Tonsillectomy', 'B+', 'High blood
pressure', 'Lisinopril');

INSERT INTO medical_record(id, previous_illness, injuries, allergies, surgeries, blood_type,
ongoing_conditions, medication_taking)
VALUES ('345677', 'Sinusitis', 'Concussion', 'Latex', 'None', 'A-', 'Diabetes', 'Metformin');

INSERT INTO medical_record(id, previous_illness, injuries, allergies, surgeries, blood_type,
ongoing_conditions, medication_taking)
VALUES ('43576', 'Strep throat', 'Fractured wrist', 'Peanuts', 'None', 'O+', 'Arthritis',
'Ibuprofen');

INSERT INTO medical_record(id, previous_illness, injuries, allergies, surgeries, blood_type,
ongoing_conditions, medication_taking)
VALUES ('58632', 'UTI', 'Sprained knee', 'Shellfish', 'Hernia repair', 'AB-', 'None', 'None');

INSERT INTO medical_record(id, previous_illness, injuries, allergies, surgeries, blood_type,
ongoing_conditions, medication_taking)
VALUES ('93531', 'Food poisoning', 'None', 'Eggs', 'None', 'B-', 'Anxiety', 'Ativan');

INSERT INTO medical_record(id, previous_illness, injuries, allergies, surgeries, blood_type,
ongoing_conditions, medication_taking)
VALUES ('789012', 'Flu', 'Dislocated shoulder', 'None', 'ACL repair', 'O-', 'None', 'None');
```

```
INSERT INTO expense(id, date, purchased_by, purpose, category, amount, branch)
VALUES ('Exp001', '2022-01-02', '456789', 'Laptop', 'Technology', 1500.00, 'Navy');

INSERT INTO expense(id, date, purchased_by, purpose, category, amount, branch)
VALUES ('Exp002', '2022-02-05', '567890', 'Rifle', 'Weapons', 1200.00, 'Marines');

INSERT INTO expense(id, date, purchased_by, purpose, category, amount, branch)
VALUES ('Exp003', '2022-02-12', '93531', 'Uniforms', 'Uniforms', 2000.00, 'Army');

INSERT INTO expense(id, date, purchased_by, purpose, category, amount, branch)
VALUES ('Exp004', '2022-03-22', '456789', 'Office Supplies', 'Other', 500.00, 'Air Force');

INSERT INTO expense(id, date, purchased_by, purpose, category, amount, branch)
VALUES ('Exp005', '2022-04-01', '43576', 'Vehicle Maintenance', 'Maintenance', 800.00, 'Army');

INSERT INTO expense(id, date, purchased_by, purpose, category, amount, branch)
VALUES ('Exp006', '2022-05-10', '58632', 'Food for Base', 'Food', 1500.00, 'Navy');

INSERT INTO expense(id, date, purchased_by, purpose, category, amount, branch)
VALUES ('Exp007', '2022-06-14', '93531', 'Miscellaneous', 'Other', 400.00, 'Marines');

INSERT INTO expense(id, date, purchased_by, purpose, category, amount, branch)
VALUES ('Exp008', '2022-07-20', '789012', 'Medical Supplies', 'Other', 1200.00, 'Air Force');

INSERT INTO expense(id, date, purchased_by, purpose, category, amount, branch)
VALUES ('Exp009', '2022-08-01', '456789', 'Headset', 'Technology', 200.00, 'Navy');

INSERT INTO expense(id, date, purchased_by, purpose, category, amount, branch)
VALUES ('Exp010', '2022-08-08', '567890', 'Ammunition', 'Weapons', 800.00, 'Marines');

INSERT INTO expense(id, date, purchased_by, purpose, category, amount, branch)
VALUES ('Exp011', '2022-08-20', '456789', 'Tablet', 'Technology', 1000.00, 'Navy');

INSERT INTO expense(id, date, purchased_by, purpose, category, amount)
VALUES ('Exp012', '2022-08-30', '456789', 'Boots', 'Uniforms', 300.00);

INSERT INTO expense(id, date, purchased_by, purpose, category, amount)
VALUES ('Exp013', '2021-09-16', '58632', 'Camping', 'Survival', 900.00);

INSERT INTO expense(id, date, purchased_by, purpose, category, amount)
VALUES ('Exp014', '2022-02-14', '93531', 'Water Jugs', 'Survival', 100.00);
```

```
INSERT INTO equipment (serial_num, name, type, manufacturer, purchase_id)
VALUES ('346712', 'AR-15', 'Arms', 'Armalite', 'Exp002');

INSERT INTO equipment (serial_num, name, type, manufacturer, purchase_id)
VALUES ('42916', 'Rope', 'Other', 'Rope Co.', 'Exp007');

INSERT INTO equipment (serial_num, name, type, manufacturer, purchase_id)
VALUES ('485291', 'tent1', 'Survival', 'Coleman', 'Exp007' );
```

```
INSERT INTO equipment (serial_num, name, type, manufacturer, purchase_id)
VALUES ('485292', 'tent2', 'Survival', 'Coleman', 'Exp013' );

INSERT INTO equipment (serial_num, name, type, manufacturer, purchase_id)
VALUES ('485293', 'tent3', 'Survival', 'Coleman', 'Exp013' );
```

SELECT * FROM equipment_checkout;

```
INSERT INTO equipment_checkout (id, equipment_id, personnel_id, checkout_date, expected_return,
actual_return)
VALUES ('00001', '346712', '43576', '2022-02-03', '2022-02-10', '2022-02-11');

INSERT INTO equipment_checkout (id, equipment_id, personnel_id, checkout_date, expected_return)
VALUES ('00002', '42916', '789012', '2022-08-16', '2022-08-23');

INSERT INTO equipment_checkout (id, equipment_id, personnel_id, checkout_date, expected_return,
actual_return)
VALUES  ('00003', '485291', '58632', '2022-05-06', '2022-05-12', '2022-05-15');

INSERT INTO equipment_checkout (id, equipment_id, personnel_id, checkout_date, expected_return,
actual_return)
VALUES ('00004', '485292', '93531', '2022-07-19', '2022-07-26', '2022-07-25');

INSERT INTO equipment_checkout (id, equipment_id, personnel_id, checkout_date, expected_return,
actual_return)
VALUES ('00005', '42916', '789012', '2022-01-01', '2023-01-08', '2023-01-09');
```

```
INSERT INTO training (id, location, start_date, instructor, objectives, branch)
VALUES ('TRN001', 'KY-92362', '2022-05-01', '93531', 'Basic training for new recruits', 'Army');

INSERT INTO training (id, location, start_date, instructor, objectives, branch)
VALUES ('TRN002', 'VA-49261', '2023-01-01', '345677', 'Advanced weapons training for special forces', 'Air Force');

INSERT INTO training (id, location, start_date, instructor, objectives, branch)
VALUES ('TRN003', 'VA-43875', '2023-06-01', '789012', 'Medical training for Navy corpsmen', 'Navy');

INSERT INTO training (id, location, start_date, instructor, objectives)
VALUES ('TRN004', 'OH-36945', '2024-01-01', '58632', 'Leadership training for senior officers', 'Marines');

INSERT INTO training (id, location, start_date, instructor, objectives)
VALUES ('TRN005', 'WA-98312', '2024-06-01', '456789', 'Basic training for new sailors', 'Coast Guard' );
```

```
INSERT INTO training_attendance (training_id, personnel_id)
VALUES ('TRN001', '456789');
```

```
INSERT INTO training_attendance (training_id, personnel_id)
VALUES ('TRN001', '567890');

INSERT INTO training_attendance (training_id, personnel_id)
VALUES ('TRN001', '678901');

INSERT INTO training_attendance (training_id, personnel_id)
VALUES ('TRN002', '345677');

INSERT INTO training_attendance (training_id, personnel_id)
VALUES ('TRN002', '43576');

INSERT INTO training_attendance (training_id, personnel_id)
VALUES ('TRN002', '58632');

INSERT INTO training_attendance (training_id, personnel_id)
VALUES ('TRN003', '93531');

INSERT INTO training_attendance (training_id, personnel_id)
VALUES ('TRN003', '789012');
```

```
INSERT INTO training_materials (training_id, equipment_id)
VALUES ('TRN005','485291');

INSERT INTO training_materials (training_id, equipment_id)
VALUES ('TRN005','485292');

INSERT INTO training_materials (training_id, equipment_id)
VALUES ('TRN005','485293');

INSERT INTO training_materials (training_id, equipment_id)
VALUES ('TRN001', '42916');

INSERT INTO training_materials (training_id, equipment_id)
VALUES ('TRN002', '346712');
```

```
INSERT INTO award (id, type, name)
VALUES ('AWD001', 'Medal', 'Purple Heart');

INSERT INTO award (id, type, name)
VALUES ('AWD002', 'Medal', 'Bronze Star');
```

```
INSERT INTO award (id, type, name)
VALUES ('AWD003', 'Medal', 'Silver Star');

INSERT INTO award (id, type, name)
VALUES ('AWD004', 'Medal', 'Distinguished Service Cross');

INSERT INTO award (id, type, name)
VALUES ('AWD005', 'Medal', 'Medal of Honor');

INSERT INTO award (id, type, name)
VALUES ('AWD006', 'Badge', 'Parachutist Badge');

INSERT INTO award (id, type, name)
VALUES ('AWD007', 'Badge', 'Combat Infantryman Badge');

INSERT INTO award (id, type, name)
VALUES ('AWD008', 'Badge', 'Air Assault Badge');

INSERT INTO award (id, type, name)
VALUES ('AWD009', 'Ribbon', 'Army Service Ribbon');

INSERT INTO award (id, type, name)
VALUES ('AWD010', 'Ribbon', 'National Defense Service Medal');
```

```
INSERT INTO awards_given (award_id, personnel_id, date_recieved)
VALUES ('AWD010', '58632', '2019-03-28');

INSERT INTO awards_given (award_id, personnel_id, date_recieved)
VALUES ('AWD002', '93531', '2014-5-30');

INSERT INTO awards_given (award_id, personnel_id, date_recieved)
VALUES ('AWD004', '789012', '2020-10-20');

INSERT INTO awards_given (award_id, personnel_id, date_recieved)
VALUES ('AWD001', '456789', '2022-12-31');
```

Queries

This query returns the name and checkout date that has ever checked out a tent.

```
-- everyone who has checkout a tent
SELECT DISTINCT p.name, branch, e.name, checkout_date,
expected_return, actual_return
FROM  personnel AS p, equipment AS e, equipment_checkout AS ec
WHERE e.name LIKE 'tent%' AND serial_num = equipment_id;
```

Thie Query returns the number of soldiers that have earned a Purple Heart Award.

```
-- Everyone that has received a Purple Heart Award
SELECT COUNT(*) as count
FROM award, awards_given
WHERE name = 'Purple Heart' AND award.id = award_id;
```

This query returns the number of people that are currently assigned to the Fort Knox Base

```
-- Counting Everyone that is stationed at fort knox
SELECT COUNT(*) as count
FROM personnel
WHERE assigned_base = 'KY-92362';
```

This query returns the number of personnel associated with each branch of the Military.

```
-- people of each branch
SELECT branch, COUNT(*) as personnel
FROM personnel
GROUP BY branch;
```

This Query calculates how many completions have been done of each type of training that is offered.

```
SELECT objectives AS training_name, COUNT(*) as count
FROM training_attendance, training
WHERE training.id =  training_id
GROUP BY objectives;
```

This Query also calculates the spending of each branch, but is only worried about small purchases that are under $500.0 and totals up all small expenses of each branch as well.

```
SELECT branch, SUM(amount) AS total_spent, COUNT(*) AS num_purchases
FROM expense
WHERE amount < 500
GROUP BY branch;
```

All personnel of the military are meant to be recorded in this database, and they stay in the database even if they retire. This query returns every veteran, or personnel that is out of the Armed Forces.

```
SELECT *
FROM personnel
WHERE date_of_discharge != 'Current';
```

This query calculates the number of wards that have been awarded based on the type of award it is.

```
SELECT type, COUNT(*) AS num_given
FROM award, awards_given
WHERE award.id = awards_given.award_id
GROUP BY type;
```

Views

Equipment Checked Out

Within the military service men and women need various equipment in order to perform trains or other tasks associated with their duties. They can get equipment by checking it out. The view below gives a person in charge of equipment who can now have access to see all equipment that has been checked out.

```
CREATE VIEW equipment_checked_out AS
SELECT p.name AS personnel_name, e.name AS equipment_name, e.type,
       ec.checkout_date, ec.expected_return, ec.actual_return
FROM personnel AS p
JOIN equipment_checkout ec ON p.id = ec.personnel_id
JOIN equipment e ON ec.equipment_id = e.serial_num;
```

This is the application code for the defined view above and will give all the logs of equipment checkout performed by the Navy

```
SELECT *
FROM equipment_checked_out
WHERE branch = 'Navy'
```

Training History

While serving in the military, people need to be trained in a wide range of things in order to serve and protect the country. It might be hard to figure out what training has been completed by different personnel. The view below makes it easy for someone to look up the training history and can be used in various queries to be able to pinpoint what training someone has done and what equipment may have been used if any equipment was used.

```
CREATE VIEW training_history AS
SELECT t.id as training_id, t.location, t.start_date, t.end_date,
       t.objectives, ta.personnel_id, p.name as personnel_name,
       p.rank as personnel_rank, p.branch as personnel_branch,
       tm.equipment_id,  e.namev as
       equipment_name, e.type as equipment_type
FROM training AS t
LEFT JOIN training_attendance AS ta ON t.id = ta.training_id
LEFT JOIN personnel AS p ON ta.personnel_id = p.id
LEFT JOIN training_materials tm ON t.id = tm.training_id
LEFT JOIN equipment AS e ON tm.equipment_id = e.serial_num
WHERE personnel_id IS NOT NULL;
```

This is the application code of the view to see the training history of everyone that is serving or has served in the Army

```
SELECT *
FROM training_history
WHERE branch = 'Army'
```

Functions

Expenses of Each Branch

Different branches of the military have different missions and operations to perform. In order to perform these tasks the branches of the military have to buy certain equipment and other niceties. The function below takes in a branch of the military and calculates how much they have spent.

```
CREATE OR REPLACE FUNCTION budget (branch varchar(12))
      RETURNS INTEGER
      LANGUAGE plpgsql
      AS
      $$
            DECLARE
                  expenses int;
            BEGIN
                  SELECT SUM(amount) INTO expenses
                  FROM expense
                  WHERE budget.branch = expense.branch;

                  RETURN expenses;
            END;
      $$
```

This application code for the defined function above will output a total of what the Army has spent that has been recorded in the database.

```
SELECT budget('Army');
```

Getting the Age of Personnel

This database does not store the age of the personnel because it changes every year, as age does. Whenever someone might need to get the age of anyone in the database, they can access it by calling the function defined below.

```
CREATE OR REPLACE FUNCTION get_soldier_age(soldier_id varchar(6))
    RETURNS INTEGER
    LANGUAGE plpgsql
      AS
      $$
            DECLARE
                  soldier_dob DATE;
                  soldier_age INT;
            BEGIN
                  SELECT dob INTO soldier_dob
                  FROM personnel
                  WHERE id = soldier_id;
                  -- Calculate age using the EXTRACT function
                  SELECT EXTRACT(year FROM age(CURRENT_DATE, soldier_dob)) INTO soldier_age;

                  RETURN soldier_age;
            END;
      $$;
```

This is some example of the function being applied to a query. This will return the id, name, and age of each person that is in the database.

```
SELECT id, name, get_soldier_age(id) AS age
FROM personnel;
```

Updating Personnel Count Trigger

Within both the branch and base tables, there is an attribute to store the number of people assigned to each branch and branch, respectively. These attributes need to be updated each time a new person is imported into the personnel table. This will become very annoying to update the personnel count of a branch or base, so to fix this, we can use a trigger. The following function can help avoid the manual updating of these attributes and make the process more efficient. The trigger function would increment the designated personnel count in both the branch and base tables based on the newest input in the personnel table.

```
CREATE OR REPLACE FUNCTION update_personnel_count()
      RETURNS TRIGGER
      LANGUAGE PLPGSQL
      AS
      $$
      BEGIN
            UPDATE branch
            SET active_personnel = active_personnel + 1
            WHERE short_name = NEW.branch;

            UPDATE base
            SET personnel_size = personnel_size + 1
            WHERE id = NEW.assigned_base;

            RETURN NEW;
      END;
      $$;
```

When a new record is added to the personnel table, the trigger will be triggered and will call the function to increment the personnel size of the corresponding branch and base. The trigger will identify the branch and base associated with the new personnel record and pass that information to the function, which will then increment the personnel count for that branch and base in their respective tables. By doing so, the trigger will automatically update the personnel count in both the branch and base tables whenever a new person is added to the personnel table.

```
CREATE OR REPLACE TRIGGER update_person_count_trigger
      AFTER INSERT ON personnel
      FOR EACH STATEMENT
      EXECUTE FUNCTION update_personnel_count();
```

Promotion Procedure

Within any organization, there is always room to move up the working ladder to hire positions in a company. The Military is not apart from this Idea at all. After serving a certain number of years, completing various pieces of training, or completing deployments or assignments, anyone can be subject to a promotion. This procedure is a very rudimentary procedure to give someone a promotion in their rank. This database covers all branches of the military, but not all branches of the military use the same ranks. This procedure is just a mock-up of certain promotions within the Army. To make this procedure work for any person, we have to probably create a procedure for each branch to make rank promotion easier and more simple to call.

```
CREATE OR REPLACE PROCEDURE promotion(id varchar(6))
	LANGUAGE plpgsql
	AS $$
	BEGIN
		UPDATE personnel
		SET rank =
			CASE rank
				WHEN 'Private' THEN 'Corporal'
				WHEN 'Corporal' THEN 'Sergeant'
				WHEN 'Sergeant' THEN 'Lieutenant'
				ELSE rank
			END
		WHERE personnel_id = id;
	END;
	$$;
```