

9th Communications Bravo Company

By: Matthew Lake



Table of Contents

Table of Contents.....	2
Executive Summary.....	3
ER Diagram.....	4
People Table.....	5
States Table.....	7
DeploymentStatusCode Table.....	8
Marines Table.....	9
Staff Table.....	11
Positions Table.....	12
JobCode Table.....	13
Records Table.....	15
WeaponDetails Table.....	16
CIFgear Table.....	17
CIFgearDetails.....	18

Views

MarineInfo.....	20
MarineCIF.....	21

Reports

Marines under 25.....	23
Old Staff.....	24

Stored Procedure

Valid Weapon Check.....	25
-------------------------	----

Trigger

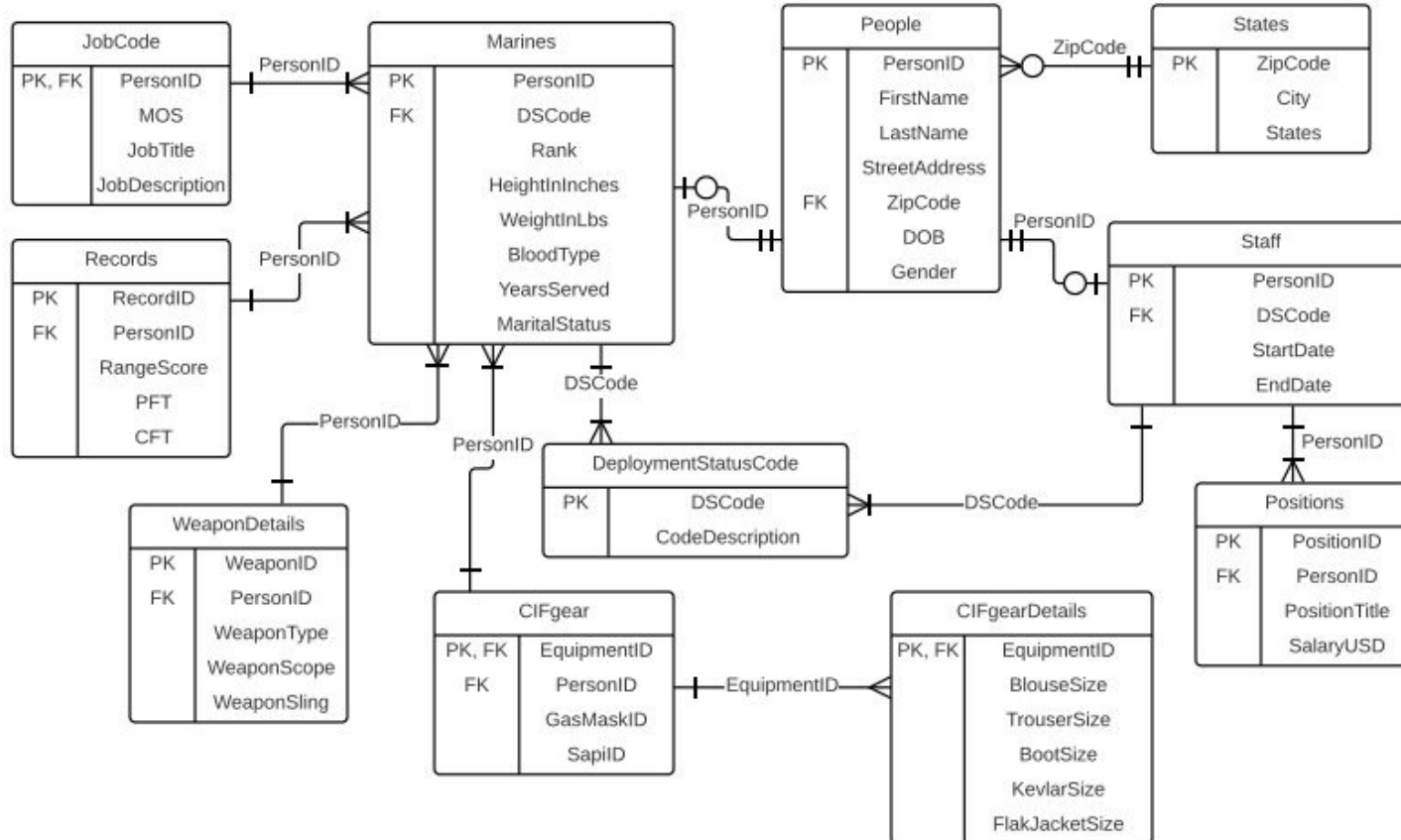
Valid Weapon Check.....	26
Security.....	27
Implementation.....	30
Known Problems.....	30
Future Enhancements.....	30

Executive Summary

9th Communications Bravo Company is a United States Marine Corps Company located on Camp Pendleton. They are responsible for providing communications to the 1st Marine Expeditionary Force (1 MEF). Bravo Company supports operations overseas in support of 1 MEF. Every Marine, staff, and piece of equipment within Bravo Company plays a vital role in making sure 1 MEF's communications are up and running.

This document outlines the design of a database for 9th Communications Bravo Company. This database will hold pertinent information of the Marines, staff, and equipment that are apart of 9th Communications Bravo Company. This database will be used to support 1 MEF operations in forward deployed zones all over the world. The data must be accurate and displayed in an organized manner. Information such as an individual's deployment status, equipment quantity, and weapon details will be used in the strategic planning of 1 MEF operations.

ER Diagram



People Table

The people table contains all individuals in 9th Comm Bn and their background information.

```
CREATE TABLE people (  
    PersonID          CHAR(8)          NOT NULL UNIQUE,  
    FirstName         TEXT             NOT NULL,  
    LastName          TEXT             NOT NULL,  
    StreetAddress     TEXT             NOT NULL,  
    ZipCode           INTEGER          NOT NULL,  
    DOB              DATE             NOT NULL,  
    Gender            CHAR(1)          NOT NULL,  
    CONSTRAINT check_Gender CHECK (Gender = 'M' OR Gender = 'F'),  
    primary key(PersonID),  
    foreign key(ZipCode) references states(ZipCode)  
);
```

Functional Dependencies

PersonID → FirstName, LastName, StreetAddress, ZipCode, DOB, Gender

Check_Gender → Checks Gender input to be either 'M' or 'F' (Representing Male or Female)

SAMPLE DATA ON FOLLOWING PAGE

personid character	firstname text	lastname text	streetaddress text	zipcode integer	dob date	gender character
p1	Matthew	Lake	24 Goose Rd	12603	1992-08-19	M
p2	Alan	Labouseur	24 Marist Dr	12603	1902-02-21	M
p3	Rocky	Balboa	35 Phili Dr	34213	1956-02-12	M
p4	John	Bones	657 Raptor St	26744	1984-01-18	M
p5	Jennifer	Lopez	242 Noxon Rd	73123	1956-02-12	F
p6	Brandon	Stellato	43 Cranberry Dr	12540	1992-06-12	M
p7	Mike	Lockwood	64 Huckleberry Dr	12540	1992-05-12	M
p8	Steph	Curry	12 MVP Ln	43276	1987-10-23	M
p9	Chris	Paul	32 Hollywood Dr	85463	1983-07-04	M
p10	Sean	Connery	18 Dolly Dr	12453	1946-01-12	M
p11	Dean	Winchester	14 Bighorn Dr	85463	1974-02-17	M
p12	Sam	Winchester	14 Bighorn Dr	85463	1971-11-25	M
p13	Larry	Bird	9 Garden St	21443	1959-10-21	M
p14	Ben	Wallace	821 City Dr	34213	1981-07-19	M
p15	Selena	Gomez	124 Princess Ln	12453	1989-06-27	F
p16	Marissa	McGrath	643 Hollywood Dr	85463	1983-04-06	F
p17	Will	Smith	Journey St	26744	1973-01-20	M
p18	Brad	Pitt	Stringham Rd	12603	1969-12-24	M
p19	Sue	Lemon	Fruit Dr	34213	1987-05-16	F

States Table

The states table lists all states and cities associated with zip codes.

```
CREATE TABLE states (  
    ZipCode      INTEGER      NOT NULL,  
    City         TEXT        NOT NULL,  
    State        TEXT        NOT NULL,  
    primary key(ZipCode)  
);
```

Functional Dependencies
ZipCode → City, State

Sample data:

zipcode integer	city text	state text
12603	Poughkeepsie	NY
34213	Philidelphia	PA
26744	Augusta	MI
73123	Little Rock	AK
12540	Hopewell Jct	NY
43276	Naples	FL
85463	Los Angeles	CA
12453	Venice	CA
21443	Tusan	AZ
44513	Boston	MA

DeploymentStatusCode Table

This table lists the deployment status of each marine and staff member of the company.

```
CREATE TABLE deploymentstatuscode (  
    DSCode                INTEGER        NOT NULL,  
    CodeDescription        TEXT          NOT NULL,  
    primary key(DSCode)  
);
```

Functional Dependencies

DeploymentStatusCode → CodeDescription

Sample data:

dscode integer	codedescription text
1001	Not Deployable
1002	Deployable no limitations
1003	Deployable maximum 6 month tour
1004	Deployable maximum 8 month tour

Marines Table

This table lists all marines in Bravo Company and information pertaining to their Marine Corps careers.

```
CREATE TABLE marines (
  PersonID      CHAR(8)      NOT NULL UNIQUE REFERENCES people(PersonID),
  DSCode        INTEGER      NOT NULL REFERENCES deploymentstatuscode(DSCode),
  Rank          TEXT         NOT NULL,
  HeightInInches INTEGER      NOT NULL,
  WeightInLbs   INTEGER      NOT NULL,
  BloodType     CHAR(3)      NOT NULL,
  YearsServed   INTEGER      NOT NULL,
  MaritalStatus CHAR(1)      NOT NULL,
  CONSTRAINT check_MaritalStatus CHECK (MaritalStatus = 'S' OR MaritalStatus = 'M'),
  primary key(PersonID)
);
```

Functional Dependencies

PersonID → DSCode, Rank, HeightInInches, WeightInLbs, BloodType, YearsServed, Marital Status

Check_MaritalStatus → Checks MaritalStatus input to be either 'S' or 'M' (Representing Single or Married)

SAMPLE DATA ON FOLLOWING PAGE

personid character	dscode integer	rank text	heightinches integer	weightinlbs integer	bloodtype character	yearserved integer	maritalstatus character
p1	1002	Sergeant	67	175	A+	4	S
p3	1002	Master Sergeant	73	210	A-	13	M
p4	1002	Lance Corporal	71	186	AB+	2	S
p6	1003	Sergeant	69	215	B-	6	S
p7	1001	Private First Class	70	174	O-	1	S
p9	1002	Gunnery Sergeant	75	225	AB-	9	M
p11	1004	Captain	77	200	O+	8	M
p13	1003	Corporal	71	160	B-	3	S
p14	1002	Staff Sergeant	73	195	A+	11	M
p16	1001	Private	66	155	O-	1	S
p17	1004	Private First Class	72	175	AB-	2	S
p20	1002	Lance Corporal	69	166	B+	2	M
p21	1001	Private First Class	75	197	A+	1	S

Staff Table

This table lists all staff that works in Bravo Company and information about their careers.

```
CREATE TABLE staff (
    PersonID      CHAR(8)      NOT NULL UNIQUE REFERENCES people(PersonID),
    DSCode        INTEGER      NOT NULL REFERENCES deploymentstatuscode(DSCode),
    StartDate     DATE         NOT NULL,
    EndDate       DATE ,
    primary key(PersonID)
);
```

Functional Dependencies

PersonID → DSCode, StartDate, EndDate

Sample data:

personid character	dscode integer	startdate date	enddate date
p2	1002	2016-03-23	
p5	1002	2015-01-14	
p8	1004	2017-07-23	
p10	1003	2014-11-13	2016-06-04
p12	1004	2013-09-12	
p15	1001	2015-05-18	
p18	1002	2012-02-08	
p19	1002	2010-03-04	

Positions Table

This table contains all the positions held by the staff in Bravo Company.

```
CREATE TABLE positions (
    PositionID          CHAR(4)          NOT NULL UNIQUE,
    PersonID            CHAR(8)          NOT NULL UNIQUE REFERENCES staff(PersonID),
    PositionTitle       TEXT ,
    SalaryUSD           NUMERIC ,
    primary key(PositionID)
);
```

Sample Data:

positionid character	personid character	positiontitle text	salaryusd numeric
1	p2	King Ruler	1000000000
2	p5	Operations Chief	130000
3	p8	Security Chief	125000
4	p10	Network Architect	115000
5	p12	Network Technician	95000
6	p15	Satellite Engineer	80000
7	p18	Radio Technician	65000
8	p19	Wire Technician	50000

Functional Dependencies

PositionID → PersonID, PositionTitle, SalaryUSD

JobCode Table

This table contains all Military Occupational Specialties of the marines in Bravo Company.

```
CREATE TABLE jobcode (  
    PersonID          CHAR(8)          NOT NULL UNIQUE REFERENCES marines(PersonID),  
    MOS               CHAR(4)          NOT NULL,  
    JobTitle          TEXT             NOT NULL,  
    JobDescription     TEXT             NOT NULL,  
    primary key(PersonID)  
);
```

Functional Dependencies

PersonID → MOS, JobTitle, JobDescription

SAMPLE DATA ON FOLLOWING PAGE

personid character	mos character	jobtitle text	jobdescription text
p1	0627	Satellite Operator	Works on antennas
p3	0621	Radio Operator	Works on radios
p4	0651	Network Engineer	Works on routers
p6	0611	Wire Technician	Works on cables
p7	0627	Satellite Operator	Works on antennas
p9	0621	Radio Operator	Works on radios
p11	0651	Network Engineer	Works on antennas
p13	0611	Wire Technician	Works on cables
p14	0621	Radio Operator	Works on radios
p16	0621	Radio Operator	Works on radios
p17	0651	Network Engineer	Works on routers
p20	0651	Network Engineer	Works on routers
p21	0651	Network Engineer	Works on routers

Records Table

This table contains the annual physical fitness scores and rifle range scores of each marine in Bravo Company.

CREATE TABLE records (

RecordID CHAR(8)
 PersonID CHAR(8)
 RangeScore INTEGER
 PFT INTEGER
 CFT INTEGER

primary key(RecordID)
);

Functional Dependencies

RecordID → PersonID, RangeScore, PFT, CFT

NOT NULL UNIQUE REFERENCES marines(PersonID),
 NOT NULL REFERENCES marines(PersonID),
 NOT NULL,
 NOT NULL,
 NOT NULL,

recordid character	personid character	rangescore integer	pft integer	cft integer
p1	p1	235	300	300
p3	p3	285	300	300
p4	p4	214	256	268
p6	p6	198	243	276
p7	p7	257	276	289
p9	p9	212	234	254
p11	p11	246	289	300
p13	p13	247	300	300
p14	p14	187	234	256
p16	p16	210	268	283
p17	p17	215	274	292
p20	p20	196	276	254
p21	p21	226	294	300

Sample Data:

WeaponDetails Table

Matthew Lake

This table contains all weapons and associated weapon gear of each marine in Bravo Company.

```
CREATE TABLE weapondetails (
```

```
    WeaponID          INTEGER
```

```
    PersonID          CHAR(8)
```

```
    WeaponType        TEXT
```

```
    WeaponScope       TEXT ,
```

```
    WeaponSling       TEXT ,
```

```
    primary key(WeaponID)
```

```
);
```

Functional Dependencies

WeaponID → PersonID, WeaponType, WeaponScope,
WeaponSling

NOT NULL UNIQUE,

NOT NULL UNIQUE REFERENCES marines(PersonID),

NOT NULL,

weaponid integer	personid character	weapon... text	weapons... text	weaponsl... text
100111	p1	M16A4	ACOG	3 Point
100200	p3	M16A4	Red Dot	3 Point
100132	p4	M4	ACOG	3 Point
100234	p6	M16A4	Red Dot	3 Point
100343	p7	M4	ACOG	1 Point
100457	p9	M9	Iron Sights	Holster
100567	p11	RPG		
100444	p13	M16A4	Red Dot	3 Point
100673	p14	M4	ACOG	1 Point
100662	p16	M4	Red Dot	1 Point
100772	p17	M16A4	ACOG	3 Point
100884	p20	M4	Red Dot	1 Point
100992	p21	M16A4	ACOG	3 Point

Sample Data:

CIFgear Table

Matthew Lake

This table contains the CIF ID that is given to every marine once they receive their gear. This table also contains the ID numbers of each marines gask mask and sapi plates.

```
CREATE TABLE cifgear (  
    EquipmentID      CHAR(8)      NOT NULL UNIQUE,  
    PersonID         CHAR(8)      NOT NULL UNIQUE REFERENCES marines(PersonID),  
    GasMaskID        INTEGER      NOT NULL,  
    SapiID           INTEGER      NOT NULL,  
    primary key(EquipmentID)  
);
```

Functional Dependencies

EquipmentID → PersonID, GasMaskID, SapiID

Sample Data:

equipme... character	personid character	gasmaskid integer	sapid integer
c1	p1	1001	2001
c2	p3	1002	2002
c3	p4	1003	2003
c4	p6	1004	2004
c5	p7	1005	2005
c6	p9	1006	2006
c7	p11	1007	2007
c8	p13	1008	2008
c9	p14	1009	2009
c10	p16	1010	2010
c11	p17	1011	2011
c12	p20	1012	2012
c13	p21	1013	2013

CIFgearDetails Table

This table contains specific sizes of each marines issued CIF gear.

```
CREATE TABLE cifgeardetails (  
    EquipmentID          CHAR(8)          NOT NULL UNIQUE REFERENCES cifgear(EquipmentID),  
    BlouseSize           TEXT              NOT NULL,  
    TrouserSize          TEXT              NOT NULL,  
    BootSize             NUMERIC           NOT NULL,  
    KevlarSize           TEXT              NOT NULL,  
    FlakJacketSize       TEXT              NOT NULL,  
    primary key (EquipmentID)  
);
```

Functional Dependencies

EquipmentID → BlouseSize, TrouserSize, BootSize, KevlarSize, FlakJacketSize

SAMPLE DATA ON FOLLOWING PAGE

equipme... character	blousesize text	trousersize text	bootsize numeric	kevlarsize text	flakjacket... text
c1	Medium	Small	9.5	Medium	Medium
c2	Medium	Medium	11	Medium	Large
c3	Small	Small	8.5	Extra Large	Medium
c4	Small	Small	12	Medium	Small
c5	Large	Medium	12	Extra Large	Large
c6	Small	Small	9	Small	Medium
c7	Large	Large	13	Medium	Large
c8	Medium	Medium	10	Large	Medium
c9	Large	Medium	11.5	Medium	Large
c10	Medium	Medium	10	Medium	Medium
c11	Medium	Small	11	Medium	Medium
c12	Medium	Small	10.5	Medium	Medium
c13	Small	Small	8	Small	Medium

VIEW MarineInfo

Matthew Lake

This view lists all the marines and the information that would be listed on a kill card (list held by a high ranking official to keep track of marine attendance and gear accountability)

```
CREATE VIEW MarineInfo AS
SELECT FirstName, LastName, Rank, MOS, WeaponID
FROM people
INNER JOIN marines
ON people.PersonID = marines.PersonID
INNER JOIN jobcode
ON marines.PersonID = jobcode.PersonID
INNER JOIN weapondetails
ON marines.PersonID = weapondetails.PersonID;
```

firstname text	lastname text	rank text	mos character	weaponid integer
Matthew	Lake	Sergeant	0627	100111
Rocky	Balboa	Master Se...	0621	100200
John	Bones	Lance Cor...	0651	100132
Brandon	Stellato	Sergeant	0611	100234
Mike	Lockwood	Private Fi...	0627	100343
Chris	Paul	Gunnery ...	0621	100457
Dean	Winchester	Captain	0651	100567
Larry	Bird	Corporal	0611	100444
Ben	Wallace	Staff Serg...	0621	100673
Marissa	McGrath	Private	0621	100662
Will	Smith	Private Fi...	0651	100772
Rob	Dyrdek	Lance Cor...	0651	100884
Brittany	Spears	Private Fi...	0651	100992

View MarineCIF

This view lists all the marines cif gear. This view would be used during a CIF inspection to verify marines have the correct gear that was assigned to them.

```
CREATE VIEW MarineCIF AS
SELECT LastName, Rank, GasMaskID, SapilD, BlouseSize, TrouserSize,
BootSize, KevlarSize, FlakJacketSize
FROM people
INNER JOIN marines
ON people.PersonID = marines.PersonID
INNER JOIN cifgear
ON marines.PersonID = cifgear.PersonID
INNER JOIN cifgeardetails
ON cifgear.EquipmentID = cifgeardetails.EquipmentID
ORDER By LastName;
```

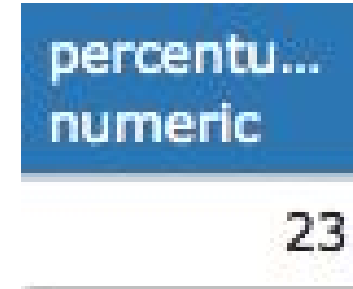
SAMPLE OF VIEW ON FOLLOWING PAGE

lastname text	rank text	gasmaskid integer	saplid integer	blousesize text	trousersize text	bootsize numeric	kevlarsize text	flakjacket... text
Balboa	Master Se...	1002	2002	Medium	Medium	11	Medium	Large
Bird	Corporal	1008	2008	Medium	Medium	10	Large	Medium
Bones	Lance Cor...	1003	2003	Small	Small	8.5	Extra Large	Medium
Dyrdek	Lance Cor...	1012	2012	Medium	Small	10.5	Medium	Medium
Lake	Sergeant	1001	2001	Medium	Small	9.5	Medium	Medium
Lockwood	Private Fi...	1005	2005	Large	Medium	12	Extra Large	Large
McGrath	Private	1010	2010	Medium	Medium	10	Medium	Medium
Paul	Gunnery ...	1006	2006	Small	Small	9	Small	Medium
Smith	Private Fi...	1011	2011	Medium	Small	11	Medium	Medium
Spears	Private Fi...	1013	2013	Small	Small	8	Small	Medium
Stellato	Sergeant	1004	2004	Small	Small	12	Medium	Small
Wallace	Staff Serg...	1009	2009	Large	Medium	11.5	Medium	Large
Winchester	Captain	1007	2007	Large	Large	13	Medium	Large

Report: Marines under the age of 25

This report allows high ranking leadership to check the percentage of marines that are under the age of 25 in Bravo Company. This statistic is important because the number of marines that commit a DUI are at a higher likelihood between the ages of 18 to 25. The leadership of Bravo Company can take into account the percentage of marines under 25 and plan accordingly with classes to limit DUI accidents.

```
SELECT TRUNC(  
    CAST(  
(SELECT COUNT (people.PersonID) AS under25  
FROM people  
RIGHT JOIN marines  
ON people.PersonID = marines.PersonID  
WHERE date_part('year',age(DOB)) < 25  
) as decimal (5,2)  
)  
/  
(SELECT COUNT (PersonID) AS allmarines  
FROM marines  
)  
*100  
) as percentUnder25;
```



Report: **Staff that no longer works at Bravo Company**

This report allows the leaders of Bravo Company to see past employers that once worked for them. If a situation comes up and an individual is no longer working for you, this report will give you their information in a quick and easy manner.

```
SELECT FirstName, LastName, PositionTitle, StartDate, EndDate
FROM people
INNER JOIN staff
ON people.PersonID = staff.PersonID
INNER JOIN positions
ON staff.PersonID = positions.PersonID
WHERE EndDate is not null;
```

firstname text	lastname text	positionti... text	startdate date	enddate date
Sean	Connery	Network ...	2014-11-...	2016-06-...

Stored Procedure: **Valid Weapon Check**

This stored procedure checks to make sure that a new entry in the WeaponDetails table has inputs for a WeaponID and a WeaponType. A weapon cannot belong to a marine unless it has a WeaponID assigned. A weapon cannot belong to a marine unless it has a Weapon Type assigned.

```
CREATE OR REPLACE FUNCTION ValidWeapon()  
RETURNS TRIGGER AS $$  
BEGIN  
    IF new.WeaponID IS null then RAISE EXCEPTION 'WeaponID cannot be null';  
    END IF;  
    IF new.WeaponType IS null then RAISE EXCEPTION 'WeaponType cannot be null';  
    END IF;  
    RETURN NEW;  
END;  
  
$$ LANGUAGE plpgsql;
```

Trigger: **Valid Weapon Check**

This trigger will run the Valid Weapon Check stored procedure whenever there is a new insert or update to the WeaponDetails table. This trigger will prevent inaccurate data of marines' weapons in Bravo Company.

```
CREATE trigger ValidWeapon  
before insert or update ON weapondetails  
for each row  
execute procedure ValidWeapon();
```

```
***** Error *****
```

```
ERROR: WeaponID cannot be null
```

```
ERROR: WeaponType cannot be null
```

```
CONTEXT: PL/pgSQL function validweapon() line 5 at  
RAISE
```

Security

Admin- The Database Administrator has full control over the database.

```
CREATE ROLE ADMIN;  
GRANT ALL ON ALL TABLES IN SCHEMA PUBLIC TO ADMIN;
```

Company Commander- In charge of Bravo Company. Given the ability to select from all tables since the Company Commander is in charge of everyone that is within this database. The only information they are unable to insert or update is any weapon or CIF gear data. The Company Armorer and Company Supply are responsible for that data only.

```
CREATE ROLE COMPANYCOMMANDER;  
REVOKE ALL ON ALL TABLES IN SCHEMA PUBLIC FROM COMPANYCOMMANDER;  
GRANT SELECT ON ALL TABLES IN SCHEMA PUBLIC TO COMPANYCOMMANDER;  
GRANT INSERT ON People, Marines, Staff, Positions, Records TO COMPANYCOMMANDER;  
GRANT UPDATE ON People, Marines, Staff, Positions, Records TO COMPANYCOMMANDER;
```

Security continued

Company Clerk- Is essentially the Company Commander's assistant/secretary. They will be be tasked with assignments from the Company Commander directly which is why it is important they have the ability to complete these tasks within the database. The only thing the Company Clerk is unable to do is access data pertaining to the civilian staff in Bravo Company. Besides that they have all the same capabilities as the Company Commander.

```
CREATE ROLE COMPANYCLERK;  
REVOKE ALL ON ALL TABLES IN SCHEMA PUBLIC FROM COMPANYCLERK;  
GRANT SELECT ON ALL TABLES IN SCHEMA PUBLIC TO COMPANYCLERK;  
GRANT INSERT ON People, Marines, Records TO COMPANYCLERK;  
GRANT UPDATE ON People, Marines, Records TO COMPANYCLERK;
```

Staff Officer- In charge of all civilian staff within Bravo Company. This is a high ranking position so they are granted access to select on all tables in the database. They are only allowed to insert or update the People, Staff, and Positions table.

```
CREATE ROLE STAFFOFFICER;  
REVOKE ALL ON ALL TABLES IN SCHEMA PUBLIC FROM STAFFOFFICER;  
GRANT SELECT ON ALL TABLES IN SCHEMA PUBLIC TO STAFFOFFICER;  
GRANT INSERT ON People, Staff, Positions TO STAFFOFFICER;  
GRANT UPDATE ON People, Staff, Positions TO STAFFOFFICER;
```

Security continued

Company Armorer- In charge of all the weapons in Bravo Company. Their job is to keep full weapons accountability for Bravo Company. In order to complete this task they are able to insert and update the WeaponDetails table in the database. They also are granted select access to the Marines table in order to get information about a marine if an issue occurs with his/her rifle.

```
CREATE ROLE COMPANYARMORER;  
REVOKE ALL ON ALL TABLES IN SCHEMA PUBLIC FROM COMPANYARMORER;  
GRANT SELECT ON Marines, WeaponDetails TO COMPANYARMORER;  
GRANT INSERT ON WeaponDetails TO COMPANYARMORER;  
GRANT UPDATE ON WeaponDetails TO COMPANYARMORER;
```

Company Supply- In charge of all the CIF gear in Bravo Company. Their job is to keep full CIF gear accountability for Bravo Company. In order to complete this task they are able to insert and update the CIFgear and CIFgearDetails tables. They are also granted select access to the Marines table in order to get information about a marine if an issue occurs with his/her gear.

```
CREATE ROLE COMPANYSUPPLY;  
REVOKE ALL ON ALL TABLES IN SCHEMA PUBLIC FROM COMPANYSUPPLY;  
GRANT SELECT ON Marines, CIFgear, CIFgearDetails TO COMPANYSUPPLY;  
GRANT INSERT ON CIFgear, CIFgearDetails TO COMPANYSUPPLY;  
GRANT UPDATE ON CIFgear, CIFgearDetails TO COMPANYSUPPLY;
```

Implementation Notes, Known Problems, Future Enhancements

Implementation- The purpose of this database is to aid in the planning and logistical process in regards to supporting 1 MEF as best as possible. This implementation is also important to holding an individual at fault. If poor planning is initiated, there is no excuse because this database provides a clear cut availability or your resources in Bravo Company.

Known Problems- There is no way of tracking marines that have left the company. This is a hard task to fix because of the vast number of marines that continued to get cycled through Bravo Company. Although each weapon ID is unique once a marine leaves the company his weapon can be transferred to a new marine that joins. This could create duplicate primary keys if Bravo Company was to run out of weapons and have to start reassigning old weapon ID numbers. This same issue of duplicate primary keys can also arise with CIF gear. Once a marine leaves the unit his CIF gear can also be given out to a new marine. They would hold the same ID numbers and create errors in the database.

Future Enhancements- Implement a way to remove a weapon ID number once a marine leaves the company without getting rid of all their personal information. Also implement a way to remove all CIF ID numbers once a marine is no longer apart of the company.