



Vlka Fenryka

By: Piradon (Tien) Liengtiraphan



Table of Contents

Table of contents.....	2
Executive Summary.....	3
Entity Relation Diagram.....	4
Create Table Statements.....	5
<i>Vlka Fenryka</i> table.....	5
<i>successorChapter</i> table.....	6
<i>geneseedBank</i> table.....	7
<i>greatCompany</i> table.....	8
<i>astartes</i> table.....	9
<i>dreadnought</i> table.....	10
<i>armaments</i> table.....	11
Create View Statements.....	12
<i>spacemarineInfo</i> view.....	12
<i>dreadnoughtInfo</i> view.....	14
<i>spacemarineEquipment</i> view.....	15
<i>spacemarineEquipment</i> breakdown.....	17
<i>greatCompanyIssued</i> view.....	18
<i>greatCompanyIssued</i> breakdown.....	20

Stored Procedures.....	21
<i>getSPInfoByName</i>	21
<i>getSPEquipByName</i>	23
<i>getDNInfoByName</i>	26
Report.....	27
<i>getGCNumByID</i>	27
Trigger.....	28
<i>getGCEquip</i>	28
Security.....	29
<i>adeptus_administatum</i> role.....	29
<i>chapter_master</i> role.....	29
<i>chapter_scribe</i> role.....	30
<i>space_marine</i> role.....	30
Implementation Notes.....	31
Known Issues.....	32
Future Enhancements.....	33



Executive Summary

"It is the 41st Millennium. For more than a hundred centuries the Emperor of Mankind has sat immobile on the Golden Throne of Earth. He is the master of mankind by the will of the gods and master of a million worlds by the might of his inexhaustible armies. He is a rotting carcass writhing invisibly with power from the Dark Age of Technology. He is the Carrion Lord of the vast Imperium of Man for whom a thousand souls are sacrificed every day so that he may never truly die. Yet even in his deathless state, the Emperor continues his eternal vigilance. Mighty battlefleets cross the daemon-infested miasma of the Warp, the only route between distant stars, their way lit by the Astronomican, the psychic manifestation of the Emperor's will. Vast armies give battle in His name on uncounted worlds. Greatest amongst his soldiers are the Adeptus Astartes, the Space Marines, bio-engineered super-warriors."

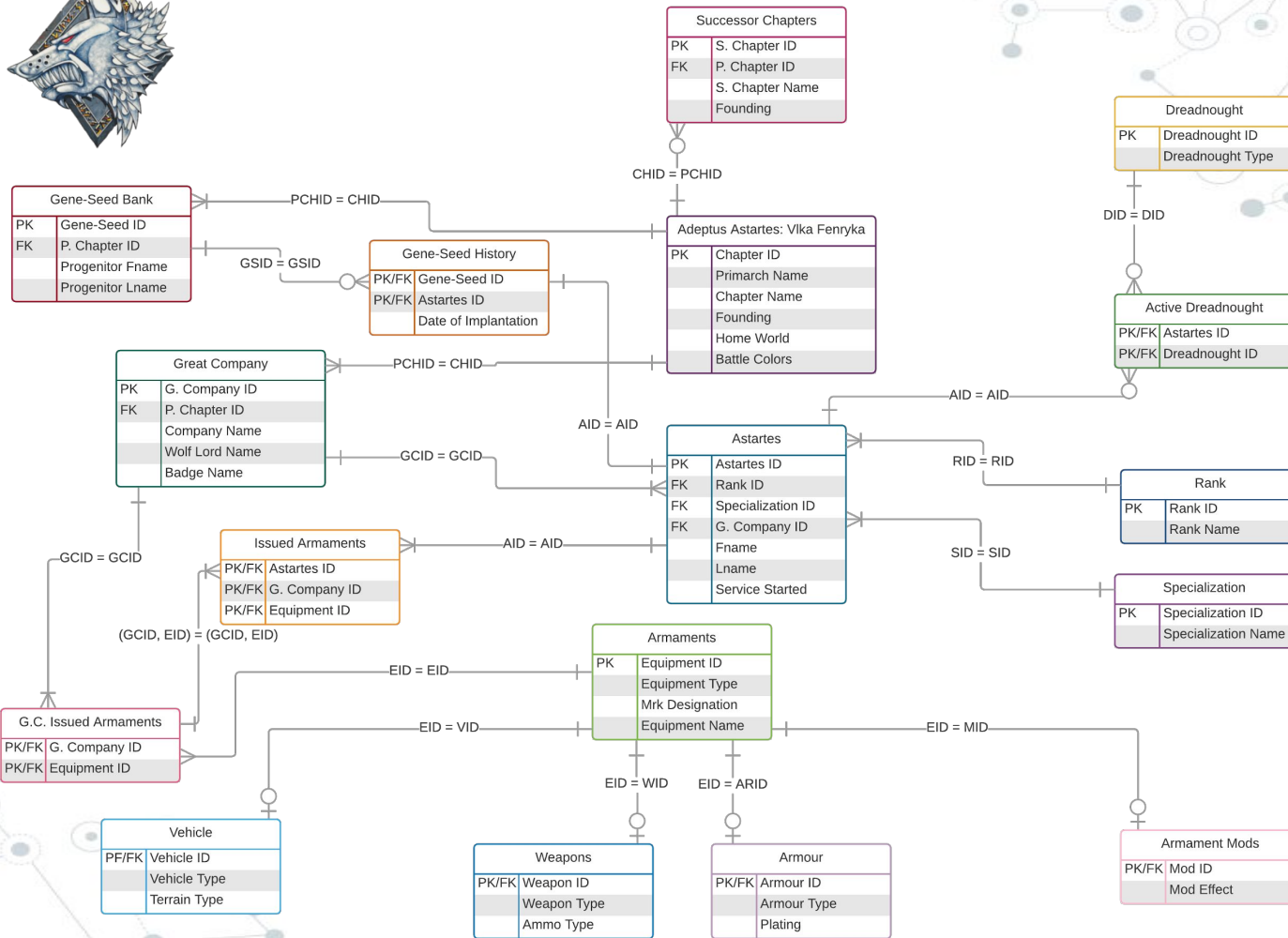
The design of this database serves to illustrate how all the gaps within the existing databases of the 41st Millennium could be solved via 3rd Millennium technologies and concepts: Referential Integrity. It includes: Chapters and their progenitor records, Gene-Seed records and assignments, Armaments assigned to each of the Great Companies, and Records for each and every Spacemarine. An overview of the database will be illustrated, followed by detailed queries and comments about the creation and purpose of the main tables and views. Lastly the different roles within the Chapter shall be detailed along with the rights they have within the system.

Further possible improvements to the database shall be listed at the end of the presentation.





Wolves of Fenris ER Diagram



Create Table Statements

***Vlka Fenryka* Table**

The most important table linking the Vlka Fenryka Space Marine Chapter to the greater Imperium Database is the table containing the over-all information about the chapter: ID, Primarch, Chapter Name, Founding, Homeworld, and Colors

```
CREATE TABLE vlkaFenryka (  
    chid          SERIAL UNIQUE NOT NULL,  
    primarchName  TEXT NOT NULL,  
    chapterName   TEXT NOT NULL,  
    founding      INT NOT NULL,  
    homeWorld     TEXT NOT NULL,  
    battleColors  TEXT NOT NULL,  
    primary key(chid)  
);
```

Functional Dependencies

chid → primarchName, chaptername, founding, homeWorld, battleColors

Sample Data

chid integer	primarchname text	chaptername text	founding integer	homeworld text	battlecolors text
1	Leman Russ	Vlka Fenryka	1	Fenris	Blue-Grey



Create Table Statements

successorChapter Table

Any successful Chapter in Warhammer 40K has successor Chapters. Successor Chapters are autonomous Chapters that originate from the source gene-seed of their progenitor. It is important keep track of these chapters in case the progenitor is in dire need for reinforcements. The Vlka Fenryka as proud as they are only have one successor chapter.

```
CREATE TABLE successorChapter (  
    schid          SERIAL UNIQUE NOT NULL,  
    pchid          INT NOT NULL REFERENCES vlkaFenryka(chid),  
    sChapterName   TEXT NOT NULL,  
    founding       INT NOT NULL,  
    primary key(schid)  
);
```

Functional Dependencies

schid → pchid, sChapterName, founding

Sampl

schid integer	pchid integer	schaptername text	founding integer
1	1	Wolf Brothers	2



Create Table Statements

geneseedBank Table

Nothing makes a chapter more than its astartes. But an astartes is nothing without his Gene-Seed. The gene-seed is the germ cells and viral machines that have been genetically-engineered to develop into the various organs that are implanted into a normal human adolescent male to transform him into a Space Marine, passed down from the mightiest warriors of Fenris.

```
CREATE TABLE geneseedBank (  
    gsid          SERIAL UNIQUE NOT NULL,  
    pchid         INT NOT NULL REFERENCES vlkaFenryka(chid),  
    progenitorFname TEXT NOT NULL,  
    progenitorLname TEXT NOT NULL,  
    primary key(gsid)  
);
```

Functional Dependencies

$gsid \rightarrow pchid, progenitorFname, progenitorLname$

Sample Data

gsid integer	pchid integer	progenitorfname text	progenitorlname text
1	1	Itard	Aenar
2	1	Raddin	Hulgar
3	1	Bjadmund	Brandr
4	1	Valund	Brandr
5	1	Kunthjol	Haaki
6	1	Fjolfrin	Snadmu



Create Table Statements

greatCompany Table

The Vlka Fenryka consist of 13 Great Companies of varying strengths. Each Great Company is based in The Fang, the Space Wolves Fortress-monastery and is led by a Wolf Lord, who answers only to the Great Wolf. Each Great Company is a fully self-contained army, with all the troops, vehicles and equipment necessary to make war, as well as the spacecraft necessary to transport itself.

```
CREATE TABLE greatCompany (  
    gcid          SERIAL UNIQUE NOT NULL,  
    pchid         INT NOT NULL REFERENCES vlkaFenryka(chid),  
    companyName   TEXT NOT NULL,  
    wolfLordName  TEXT NOT NULL,  
    badgeName     TEXT NOT NULL,  
    primary key(gcid)  
);
```

Functional Dependencies

gcid → pchid, companyName, wolfLordName, badgeName

Sample Data

gcid integer	pchid integer	companyname text	wolfordname text	badgename text
1	1	Company of the Great Wolf	Logan Grimnar	Wolf that Stalks the Stars
2	1	Bloodmaws	Bran Redmaw	Bloodied Hunter
3	1	Seawolves	Engir Krakendoom	Sea Wolf
4	1	Sons of Morkai	Erik Morkai	Morkai
5	1	Red Moons	Gunnar Red Moon	Wolf of the Red Moon
6	1	Deathwolves	Harald Deathwolf	Great Devourer
7	1	Stormwolves	Bjorn Stormwolf	Thunderwolf



Create Table Statements

astartes Table

The Space Marines or Adeptus Astartes are foremost amongst the defenders of humanity, the greatest of the Emperor of Mankind's warriors. Within each of them is a Gene-Seed, which grants them all the abilities of their progenitors and records their own for future generations

```
CREATE TABLE astartes (  
    aid          SERIAL UNIQUE NOT NULL,  
    rid          INT NOT NULL REFERENCES rank(rid),  
    sid          INT NOT NULL REFERENCES specialization(sid),  
    gcid         INT NOT NULL REFERENCES greatCompany(gcid),  
    fName       TEXT NOT NULL,  
    lName       TEXT NOT NULL,  
    serviceStart TEXT NOT NULL,  
    primary key(aid)  
);
```

Functional Dependencies

aid → rid, sid, gcid, fName, lName, serviceStart

Sample Data

aid integer	rid integer	sid integer	gcid integer	fName text	lName text	servicestart text
1	1	2	1	Logan	Grimnar	766 781.M35
2	2	1	2	Bran	Redmaw	099 158.M33
3	2	2	3	Engir	Krakendoom	325 969.M40
4	2	3	4	Erik	Morkai	707 242.M35
5	2	4	5	Gunnar	Red Moon	234 195.M38
6	2	5	6	Harald	Deathwolf	481 760.M39
7	2	6	7	Bjorn	Stormwolf	727 484.M32



Create Table Statements

dreadnought Table

When the greatest of the Astartes are crippled in combat the Battle-Brother's body will be repaired and transferred into an armoured cybernetic sarcophagus outfitted with extensive life support systems. This sarcophagus is then interred within the heart of the armoured body of a Dreadnought if there is even a spark of life left. These legends are kept alive to serve the Imperium once more. “Even in death I still serve”.

```
CREATE TABLE dreadnought (  
    did          SERIAL UNIQUE NOT NULL,  
    dreadType   TEXT NOT NULL,  
    primary key(did)  
);
```

Functional Dependencies

did → dreadType

Sample Data

did integer	dreadtype text
1	Venerable
2	Siege
3	Ironclad
4	Hellfire
5	Contemptor
6	Deredeo
7	Librarian
8	Chaplain
9	Leviathan



Create Table Statements

armaments Table

Space Marines are mankind's greatest weapons, but to these superhumans even deadlier weapons must be issued for them to be able to complete their mission. The best armour must be given so that they may stand in the face of overwhelming odds. Vehicles must be available to them so that they may go where they are needed most.

```
CREATE TABLE armaments (  
    eid          SERIAL UNIQUE NOT NULL,  
    type         TEXT NOT NULL,  
    mrkDesignation Char(12) NOT NULL,  
    eName        TEXT NOT NULL,  
    primary key(eid)  
);
```

Functional Dependencies

eid → type, mrkDesignation, eName

Sample Data

eid integer	type text	mrkdesignation character(12)	ename text
1	Weapon	VB	Godwyn Pattern Bolter
2	Weapon	IV	Ultra Pattern Bolter
3	Weapon	I	Lighting Claws
4	Weapon	I	Force Sword
5	Weapon	I	Doom Hammer
6	Weapon	Xf	Krakentooth Pattern Chainsword
7	Armour	I	Thunder Pattern Power Armour
8	Armour	II	Crusade Pattern Power Armour
9	Armour	III	Iron Pattern Power Armour



Create View Statements

spacemarineInfo View

Retrieves all relevant information about a Space Marine and displays them in one easy-to-understand table. (7 tables involved)

```
CREATE OR REPLACE VIEW spacemarineInfo AS
SELECT      a.aid,
            a.fname,
            a.lname,
            vf.chaptername,
            vf.primarchname,
            a.servicestart,
            gsb.gsid,
            gsh.dateofimplant,
            gc.companyname,
            s.specialization,
            vf.battlecolors
FROM astartes a
INNER JOIN geneseedHistory gsh
      ON a.aid = gsh.aid
INNER JOIN geneseedBank gsb
      ON gsh.gsid = gsb.gsid
INNER JOIN greatCompany gc
      ON gc.gcid = a.gcid
INNER JOIN rank r
      ON a.rid = r.rid
INNER JOIN specialization s
      ON a.sid = s.sid
INNER JOIN vlkaFenryka vf
      ON gsb.pchid = vf.chid;
```



Create View Statements

spacemarineInfo View - Results

aid integer	fname text	lname text	chaptername text	primarchname text	servicestart text	gsid integer	dateofimplant text	companyname text	specialization text	battlecolors text
1	Logan	Grimnar	Vlka Fenryka	Leman Russ	766 781.M35	30	643 830.M33	Company of the Great Wolf	Assault	Blue-Grey
2	Bran	Redmaw	Vlka Fenryka	Leman Russ	099 158.M33	1	318 077.M32	Bloodmaws	Tactical	Blue-Grey
3	Engir	Krakendoom	Vlka Fenryka	Leman Russ	325 969.M40	2	911 219.M39	Seawolves	Assault	Blue-Grey
4	Erik	Morkai	Vlka Fenryka	Leman Russ	707 242.M35	3	541 971.M34	Sons of Morkai	Terminator	Blue-Grey
5	Gunnar	Red Moon	Vlka Fenryka	Leman Russ	234 195.M38	4	012 578.M37	Red Moons	Seige Warfare	Blue-Grey
6	Harald	Deathwolf	Vlka Fenryka	Leman Russ	481 760.M39	5	569 896.M38	Deathwolves	Demolisher	Blue-Grey
7	Bjorn	Stormwolf	Vlka Fenryka	Leman Russ	727 484.M32	6	616 102.M31	Stormwolves	Calvary	Blue-Grey
8	Egil	Iron Wolf	Vlka Fenryka	Leman Russ	803 029.M40	7	161 999.M36	Ironwolves	Tactical	Blue-Grey
9	Krom	Dragongaze	Vlka Fenryka	Leman Russ	758 253.M35	8	210 645.M34	Drakeslayers	Assault	Blue-Grey
10	Ragnar	Blackmane	Vlka Fenryka	Leman Russ	838 401.M37	9	474 651.M36	Blackmanes	Assault	Blue-Grey
11	Sven	Bloodhowl	Vlka Fenryka	Leman Russ	822 786.M35	10	480 604.M34	Firehowlers	Terminator	Blue-Grey
12	Kjarl	Grimblood	Vlka Fenryka	Leman Russ	602 962.M39	11	452 065.M38	Grimbloods	Seige Warfare	Blue-Grey
13	Bulveye	Axeman of Russ	Vlka Fenryka	Leman Russ	313 169.M30	12	672 532.M29	Wulfen	Demolisher	Blue-Grey
14	Bjorn	Fell-Handed	Vlka Fenryka	Leman Russ	359 417.M31	13	519 441.M30	Company of the Great Wolf	Assault	Blue-Grey
15	Kverlaf	Murderfang	Vlka Fenryka	Leman Russ	273 678.M34	14	966 203.M33	Company of the Great Wolf	Assault	Blue-Grey



Create View Statements

***dreadnoughtInfo* View**

Retrieves all relevant information about a Dreadnought and displays them in one easy-to-understand table. (4 tables involved)

```
CREATE OR REPLACE VIEW dreadnoughtInfo AS
SELECT      a.aid,
            a.fname,
            a.lname,
            d.dreadtype,
            a.servicestart,
            ad.internmentdate,
            gc.companyname
FROM astartes a
INNER JOIN activeDreadnought ad
      ON a.aid = ad.aid
INNER JOIN dreadnought d
      ON ad.did = d.did
INNER JOIN greatCompany gc
      ON gc.gcid = a.gcid;
```

aid integer	fname text	lname text	dreadtype text	servicestart text	internmentdate text	companyname text
14	Bjorn	Fell-Handed	Venerable	359 417.M31	130 620.M33	Company of the Great Wolf
15	Kverlaf	Murderfang	Ironclad	273 678.M34	839 269.M35	Company of the Great Wolf



Create View Statements

spacemarineEquipment View

Filters through armaments and its subtypes, astartes, gclssuedArmaments, and issuedArmaments to retrieve all equipment assigned to a Space Marine. (Links together 3 other views to hide complexity)

```
CREATE OR REPLACE VIEW spacemarineEquipment AS
SELECT      aid,
            fname,
            lname,
            eid,
            mrkdesignatation,
            ename
FROM spacemarineWeapons
UNION
SELECT      aid,
            fname,
            lname,
            eid,
            mrkdesignatation,
            ename
FROM spacemarineArmour
UNION
SELECT      aid,
            fname,
            lname,
            eid,
            mrkdesignatation,
            ename
FROM spacemarineMods;
```



Create View Statements

spacemarineEquipment View - Results

aid integer	fname text	lname text	eid integer	mrkdesignatn character(12)	ename text
13	Bulveye	Axeman of Russ	30	X	Iron Halo
1	Logan	Grimnar	33	III	Teleporter
8	Egil	Iron Wolf	30	X	Iron Halo
10	Ragnar	Blackmane	4	I	Force Sword
10	Ragnar	Blackmane	18	IV	Tartaros Pattern Tactical Dreadnought Armour
11	Sven	Bloodhowl	14	VIII	Errant Pattern Power Armour
4	Erik	Morkai	31	VI	Machine Spirit Core
5	Gunnar	Red Moon	4	I	Force Sword
7	Bjorn	Stormwolf	1	VB	Godwyn Pattern Bolter
5	Gunnar	Red Moon	9	III	Iron Pattern Power Armour
11	Sven	Bloodhowl	1	VB	Godwyn Pattern Bolter
7	Bjorn	Stormwolf	3	I	Lighting Claws
12	Kjarl	Grimblood	3	I	Lighting Claws
2	Bran	Redmaw	10	IV	Maximus Pattern Power Armour
3	Engir	Krakendoom	2	IV	Ultra Pattern Bolter
1	Logan	Grimnar	19	V	Aegis Terminator Armour
4	Erik	Morkai	28	I	Night Vision
8	Egil	Iron Wolf	5	I	Doom Hammer
2	Bran	Redmaw	30	X	Iron Halo
4	Erik	Morkai	30	X	Iron Halo
7	Bjorn	Stormwolf	19	V	Aegis Terminator Armour
4	Erik	Morkai	16	II	Indomitus Pattern Tactical Dreadnought Armour



Create View Statements - Three Views for spacemarineEquipment

```
CREATE OR REPLACE VIEW
spacemarineMods AS
SELECT      a.aid,
            a.fname,
            a.lname,
            mi.eid,
            mi.mrkdesignation,
            mi.ename,
            mi.meffect
FROM astartes a
INNER JOIN issuedArmaments
ia
            ON a.aid = ia.aid
INNER JOIN modInfo mi
            ON ia.eid = mi.eid;
```

```
CREATE OR REPLACE VIEW
spacemarineArmour AS
SELECT      a.aid,
            a.fname,
            a.lname,
            ar.eid,
            ar.mrkdesignation,
            ar.ename,
            ar.atype,
            ar.plating
FROM astartes a
INNER JOIN issuedArmaments
ia
            ON a.aid = ia.aid
INNER JOIN armourInfo ar
            ON ia.eid = ar.eid;
```

```
CREATE OR REPLACE VIEW
spacemarineWeapons AS
SELECT      a.aid,
            a.fname,
            a.lname,
            wi.eid,
            wi.mrkdesignation,
            wi.ename,
            wi.wtype,
            wi.ammo
FROM astartes a
INNER JOIN issuedArmaments
ia
            ON a.aid = ia.aid
INNER JOIN weaponInfo wi
            ON ia.eid = wi.eid;
```



Create View Statements

greatCompanyIssued View

Filters through armaments and its subtypes, and gclassIssuedArmaments, to retrieve all equipment assigned to a Great Company.

```
CREATE OR REPLACE VIEW greatCompanyIssued AS
SELECT      gcid,
            eid,
            mrkdesignation,
            ename
FROM greatCompanyIssuedArmour
UNION
SELECT      gcid,
            eid,
            mrkdesignation,
            ename
FROM greatCompanyIssuedWeapons
UNION
SELECT      gcid,
            eid,
            mrkdesignation,
            ename
FROM greatCompanyIssuedMods;
```



Create View Statements

greatCompanyIssued View - Results

gcid integer	eid integer	mrkdesignatn character(12)	ename text
6	33	III	Teleporter
11	30	X	Iron Halo
1	32	IX	Stealth Drive
5	2	IV	Ultra Pattern Bolter
10	32	IX	Stealth Drive
11	13	VII	Aquila Pattern Power Armour
2	33	III	Teleporter
8	29	V	Stormwind Pattern Jetpack
8	13	VII	Aquila Pattern Power Armour
2	2	IV	Ultra Pattern Bolter
9	19	V	Aegis Terminator Armour
3	4	I	Force Sword
9	11	V	Heresy Pattern Power Armour
10	31	VI	Machine Spirit Core
3	14	VIII	Errant Pattern Power Armour
13	11	V	Heresy Pattern Power Armour
1	31	VI	Machine Spirit Core
12	28	I	Night Vision
8	30	X	Iron Halo
5	33	III	Teleporter
4	12	VI	Corcus Pattern Power Armour
1	28	I	Night Vision



Create View Statements - Three Views for greatCompanyIssued

```
CREATE OR REPLACE VIEW
greatCompanyIssuedWeapons
AS
SELECT      gca.gcid,
            gca.eid,
            wi.mrkdesignation,
            wi.ename,
            wi.wtype,
            wi.ammo
FROM gcArmaments gca
INNER JOIN weaponInfo wi
      ON gca.eid = wi.eid;
```

```
CREATE OR REPLACE VIEW
greatCompanyIssuedArmour
AS
SELECT      gca.gcid,
            gca.eid,
            ar.mrkdesignation,
            ar.ename,
            ar.atype,
            ar.plating
FROM gcArmaments gca
INNER JOIN armourInfo ar
      ON gca.eid = ar.eid;
```

```
CREATE OR REPLACE VIEW
greatCompanyIssuedMods AS
SELECT      gca.gcid,
            gca.eid,
            mi.mrkdesignation,
            mi.ename,
            mi.meffect
FROM gcArmaments gca
INNER JOIN modInfo mi
      ON gca.eid = mi.eid;
```

***Each of These Statements Break
into smaller views.**



Stored Procedures - Get Space Marine Information by Name

getSPInfoByName Procedure

There will be times when the Chapter Master wishes to look up a Space Marine's records but doesn't know his Astartes ID, the Chapter Master or Scribe could then use this function to retrieve a Marine's Records via the name he's called by. With this function the Chapter Master or Scribe need not write a query but simply pass values that they know into this function

```
CREATE OR REPLACE FUNCTION getSPInfoByName(TEXT, TEXT, REFCURSOR) RETURNS refcursor AS
$$
DECLARE
    spFname          TEXT          := $1;
    spLname          TEXT          := $2;
    resultset        REFCURSOR     := $3;
BEGIN
    OPEN resultset FOR
        SELECT      *
        FROM        spacemarineInfo
        WHERE       fname LIKE spFname
        AND         lname LIKE spLname;
    return resultset;
end;
$$
LANGUAGE plpgsql;
```



Stored Procedures - Get Space Marine Information by Name

getSPInfoByName Procedure - Results

```
SELECT getSPInfoByName('%', 'Death%', 'ref');  
FETCH ALL FROM ref;
```

aid integer	fname text	lname text	chaptername text	primarchname text	servicestart text	gsid integer	dateofimplant text	companyname text	rankname text	specialization text	battlecolors text
6	Harald	Deathwolf	Vlka Fenryka	Leman Russ	481 760.M39	5	569 896.M38	Deathwolves	Wolf Lord	Demolisher	Blue-Grey

```
SELECT getSPInfoByName('E%', 'K%', 'ref');  
FETCH ALL FROM ref;
```

aid integer	fname text	lname text	chaptername text	primarchname text	servicestart text	gsid integer	dateofimplant text	companyname text	rankname text	specialization text	battlecolors text
3	Engir	Krakendoom	Vlka Fenryka	Leman Russ	325 969.M40	2	911 219.M39	Seawolves	Wolf Lord	Assault	Blue-Grey



Stored Procedures - Get Space Marine Information by Name

getSPEquipByName Procedure

Each marine starts with simply a Bolter and a set of Armour. At times of promotion they are presented with another piece of equipment. A Chapter Master or Scribe might wish to present the Space Marine specified in the above stored procedure a piece of equipment they don't currently have as a sign of promotion, this stored procedure will return all the equipment a Space Marine has currently been issued.

```
CREATE OR REPLACE FUNCTION getSPEquipByName(TEXT, TEXT, REFCURSOR) RETURNS refcursor AS
$$
DECLARE
    spFname          TEXT          := $1;
    spLname          TEXT          := $2;
    resultset        REFCURSOR     := $3;
BEGIN
    OPEN resultset FOR
        SELECT eid,
               mrkdesignatn,
               ename
        FROM     spacemarineEquipment
        WHERE    fname LIKE spFname
               AND lname LIKE spLname;
    return resultset;
end;
$$
LANGUAGE plpgsql;
```



Stored Procedures - Get Space Marine Equipment by Name

getSPEquipByName Procedure - Results

```
SELECT getSPEquipByName('K%', 'Dragon%', 'results');  
FETCH ALL FROM results;
```

eid integer	mrkdesignat ion character(12)	ename text
5	I	Doom Hammer
2	IV	Ultra Pattern Bolter
8	II	Crusade Pattern Power Armour

```
SELECT getSPEquipByName('E%', 'K%', 'ref');  
FETCH ALL FROM ref;
```

eid integer	mrkdesignat ion character(12)	ename text
3	I	Lighting Claws
30	X	Iron Halo
28	I	Night Vision
1	VB	Godwyn Pattern Bolter
19	V	Aegis Terminator Armour
31	VI	Machine Spirit Core



Stored Procedures - Get Dreadnought Information by Name

getDNInfoByName Procedure

Each Dreadnought is a former battle-brother who has been crippled and interned within the life-sustaining machinations of the Dreadnought armor. These fallen battle brothers are given a new date to their record, their “internment date” the day in which the man ceased to exist and the machine rose to serve the Imperium in his place. This procedure retrieves all relevant information about a Dreadnought via their name and allows for easy record keeping.

```
CREATE OR REPLACE FUNCTION getDNInfoByName(TEXT, TEXT, REFCURSOR) RETURNS refcursor AS
$$
DECLARE
    dnFname          TEXT          := $1;
    dnLname          TEXT          := $2;
    resultset        REFCURSOR     := $3;
BEGIN
    OPEN resultset FOR
    SELECT*
        FROM          dreadnoughtInfo
        WHERE          fname LIKE dnFname
                    ANDlname LIKE dnLname;
    return resultset;
end;
$$
LANGUAGE plpgsql;
```



Stored Procedures - Get Dreadnought Information by Name

getDNInfoByName Procedure - Results

```
SELECT getDNInfoByName('B%', 'F%', 'ref2');  
FETCH ALL FROM ref2;
```

aid integer	fname text	lname text	dreadtype text	servicestart text	internmentdate text	companyname text
14	Bjorn	Fell-Handed	Venerable	359 417.M31	130 620.M33	Company of the Great Wolf

```
SELECT getDNInfoByName('Kverlaf', 'Murder%', 'ref2');  
FETCH ALL FROM ref2;
```

aid integer	fname text	lname text	dreadtype text	servicestart text	internmentdate text	companyname text
15	Kverlaf	Murderfang	Ironclad	273 678.M34	839 269.M35	Company of the Great Wolf



Report - Get Great Company Men-at-Arms By Id

getGCNumById Procedure - Results

This function allows any administrative body and/or Chapter Master to access the current number of Men-at-Arms for any Great Company via their ID. This is useful for allocating reinforcements to said Great Company during times of war or in situations where their numbers have plummeted below acceptable levels.

```
CREATE OR REPLACE FUNCTION getGCNumById(INT, REFCURSOR) RETURNS refcursor AS
$$
DECLARE
    wgcid      INT          := $1;
    resultset  REFCURSOR   := $2;
BEGIN
    OPEN resultset FOR
        SELECT    COUNT(aid)
        FROM      astartes
        WHERE     gcid = wgcid;
    return resultset;
end;
$$
LANGUAGE plpgsql;
```

Results →

count
bigint
3



Trigger - Check Equipment Before Assignment

getGCEquip Trigger

Checks to see if the entered eid exists within the equipments assigned to the Great Company to which the selected Astartes belongs. Runs before inserts and updates.

```
CREATE OR REPLACE FUNCTION getGCEquip() RETURNS TRIGGER AS
$getGCEquip$
DECLARE
    wgcid          INT          := (SELECT eid FROM greatCompanyIssued WHERE gcid = wgcid);

BEGIN
    IF(weid in (SELECT eid FROM greatCompanyIssued WHERE gcid = wgcid))
        THEN UPDATE issuedArmaments SET invalid = 1;
        END IF;

end;
$getGCEquip$
LANGUAGE plpgsql;

CREATE TRIGGER getGCEquip
    BEFORE INSERT or UPDATE ON issuedArmaments
    FOR EACH ROW
    EXECUTE PROCEDURE getGCEquip();
```



Security - Adeptus Administratum & Chapter Master

***adeptus_administratum* Role**

The Adeptus Administratum is the administrative and bureaucratic division of the Adeptus Terra, the heart of the gigantic bureaucracy that controls the government of the Imperium of Man, consisting of untold billions of clerks, scribes and administrative staff constantly working to manage the Imperium at every level, from assembling war fleets to levying taxes. It is the largest of the departments comprising the Adeptus Terra. They have access to every record available to the Imperium and can even supplant the authority of a Chapter Master, with dire consequences.

```
CREATE ROLE adeptus_administratum;  
GRANT SELECT, INSERT, UPDATE, DELETE  
ON ALL TABLES IN SCHEMA PUBLIC  
TO adeptus_administratum;
```

***chapter_master* Role**

To be a Chapter Master of an Imperial Space Marines Chapter is to be a superhuman avatar of war amongst mere mortals. The Chapter Masters of the Adeptus Astartes are unmatched in personal combat prowess, possessing the body of a genetically superior killing machine and literally centuries of combat experience. He is a true scion of the Emperor of Mankind, heir to the strength and fortitude possessed by his genetic forebears, the Primarchs. Hence he has access to the entire record of his chapter's history. With the title of Chapter Master comes the ability to induct new members, promote current members, and strike people from the Chapter's History in extreme cases.

```
CREATE ROLE chapter_master;  
GRANT SELECT, INSERT, UPDATE, DELETE  
ON ALL TABLES IN SCHEMA PUBLIC  
TO chapter_master;
```



Security - Chapter Scribe

***chapter_scribe* Role**

Chapter Scribes or Librarians are the psykers of the Space Marines who survive the Adeptus Astartes Chapter's rigorous screening and training to bend the powers of the Warp to their will for the benefit of their fellow Battle-Brothers and in service to the Emperor of Mankind. Beyond their psychic duties, the Librarians of the Astartes are also expected to record the great deeds of their Chapter and maintain the Chapter's storehouse of ancient lore, the functions for which they are named.

```
CREATE ROLE chapter_scribe;
GRANT SELECT, INSERT, UPDATE
ON ALL TABLES IN SCHEMA PUBLIC
TO chapter_scribe;
```

***space_marine* Role**

Each Space Marine who joins a chapter is inducted into the brotherhood as a recruit (naming scheme varies depending on chapter) they are stripped of their previous life and live to serve the Greater Imperium and die in service to her existence. They have the ability to view their own and other Space Marine's Equipment and basic information.

```
CREATE ROLE space_marine;
GRANT SELECT
ON greatCompanyIssued
TO space_marine;
GRANT SELECT
ON spacemarineEquipment
TO space_marine;
GRANT SELECT
ON spacemarineInfo
TO space_marine;
```



Implementation Notes

- ◎ Purpose
 - The purpose of the database is to allow the adeptus administratum to operate as it has done in the past, albeit with less loss of data regarding the each of the newly founded Space Marine Chapters, relying on Referential Integrity and Checks to keep data from going missing, and to be able to hold officers in the Administratum responsible for any missing data.
- ◎ Test Data
 - Test Data originated, from officially supported documents of the Black Library and the active community of the Warhammer40K wiki and WH40K.Lexicanum. If dates were unavailable they were generated to fit the WH date format using the javascript file provided in the “Documents” Folder.
 - All information here does not accurately reflect the actual lore of Warhammer 40K and should in no way be used as a reference when making claims and statements.



Known Issues

- ⦿ There is currently no way to check to see if a Space Marine is dead or not unless you check the Gene-Seed History. However there is no such case as-of-now and the database may react in an unpredictable manner.
- ⦿ Trigger is not certain to work the way it was intended (will verify when possible). Issue for Trigger's creation currently solved via Referential Integrity.
- ⦿ More Views should be created to assist with other possible use cases.



Future Enhancements

- Implement way to check if a Space Marine has completed term of service (KIA or MIA)
- Implement more checks to ensure accurate data input.
- Create linking table to greater Imperium database.
- Extends database information to include different time periods of the chapter (Great Crusade, Horus Heresy, The Scouring, The Time of Rebuilding, The Time of Ending)
- Supplement database with Adeptus Mechanicus Support as well as Human population of worlds under direct control

