Vlka Fenryka

By: Piradon (Tien) Liengtiraphan
# Table of Contents

- Table of contents ..................................................2
- Executive Summary .................................................3
- Entity Relation Diagram ...........................................4
- Create Table Statements ...........................................5
  - *Vlka Fenryka* table ...........................................5
  - *successorChapter* table .....................................6
  - *geneseedBank* table ..........................................7
  - *greatCompany* table ..........................................8
  - *astartes* table ................................................9
  - *dreadnought* table .........................................10
  - *armaments* table ...........................................11
- Create View Statements ...........................................12
  - *spacemarineInfo* view ......................................12
  - *dreadnoughtInfo* view ......................................14
  - *spacemarineEquipment* view ...............................15
  - *spacemarineEquipment* breakdown ........................17
  - *greatCompanyIssued* view .................................18
  - *greatCompanyIssued* breakdown ..........................20
- Stored Procedures ................................................21
  - *getSPInfoByName* .............................................21
  - *getSPEquipByName* ...........................................23
  - *getDNInfoByName* ...........................................26
- Report ...............................................................27
  - *getGCNumByID* ................................................27
- Trigger .................................................................28
  - *getGCEquip* .....................................................28
- Security ...............................................................29
  - *adeptus_administatum* role ...............................29
  - *chapter_master* role .......................................30
  - *chapter_scribe* 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.
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**

<table>
<thead>
<tr>
<th>chid</th>
<th>primarchName</th>
<th>chapterName</th>
<th>founding</th>
<th>homeWorld</th>
<th>battleColors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Leman Russ</td>
<td>Vlka Fenryka</td>
<td>1</td>
<td>Fenris</td>
<td>Blue-Grey</td>
</tr>
</tbody>
</table>
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.

```sql
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**

- \( \text{schid} \rightarrow \text{pchid}, \text{sChapterName}, \text{founding} \)

**Sample Data**

<table>
<thead>
<tr>
<th>schid</th>
<th>pchid</th>
<th>sChapterName</th>
<th>founding</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Wolf Brothers</td>
<td>2</td>
</tr>
</tbody>
</table>
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 → pchid, progenitorFname, progenitorLname

**Sample Data**

<table>
<thead>
<tr>
<th>gsid</th>
<th>pchid</th>
<th>progenitorFname</th>
<th>progenitorLname</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Iterd</td>
<td>Aner</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Vegdrin</td>
<td>Vulgan</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Bjadmund</td>
<td>Erandr</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>Vakund</td>
<td>Erandr</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>Munthjiol</td>
<td>Maatur</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>Fjolfrin</td>
<td>Snadnu</td>
</tr>
</tbody>
</table>
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**

\[ \text{gcid} \rightarrow \text{pchid, companyName, wolfLordName, badgeName} \]

**Sample Data**

<table>
<thead>
<tr>
<th>gcid</th>
<th>pchid</th>
<th>companyName</th>
<th>wolfLordName</th>
<th>badgeName</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Company of the Great Wolf</td>
<td>Logan Grimnar</td>
<td>Wolf that Stalks the Stars</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>Bloodwolves</td>
<td>Bran Redmaw</td>
<td>Bloodied Hunter</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>Seawolves</td>
<td>Engrir Weakenoom</td>
<td>Sea Wolf</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>Sons of Horkai</td>
<td>Erik Horkai</td>
<td>Horkai</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>Red Moons</td>
<td>Gunnar Red Moon</td>
<td>Wolf of the Red Moon</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>Deathwolves</td>
<td>Harold Deathwolf</td>
<td>Great Devourer</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>Stormwolves</td>
<td>Bjorn Stormwolf</td>
<td>Thunderwolf</td>
</tr>
</tbody>
</table>
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 the 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 \rightarrow rid, sid, gcid, fName, lName, serviceStart

Sample Data

<table>
<thead>
<tr>
<th>aid</th>
<th>rid</th>
<th>sid</th>
<th>gcid</th>
<th>fName</th>
<th>lName</th>
<th>serviceStart</th>
<th>Name</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>Logan</td>
<td>McLean</td>
<td>566 781.340M</td>
<td>McLean</td>
<td>28</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>Brian</td>
<td>Sams</td>
<td>099 158.534M</td>
<td>Sams</td>
<td>28</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>Greg</td>
<td>Krakendom</td>
<td>325 969.340M</td>
<td>Krakendom</td>
<td>28</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>Erik</td>
<td>Holt</td>
<td>707 242.340M</td>
<td>Holt</td>
<td>28</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>4</td>
<td>5</td>
<td>Lassal</td>
<td>Red Moon</td>
<td>234 196.340M</td>
<td>Red Moon</td>
<td>28</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>5</td>
<td>6</td>
<td>Harold</td>
<td>Beastwolf</td>
<td>481 760.340M</td>
<td>Beastwolf</td>
<td>28</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>4</td>
<td>7</td>
<td>Bjorn</td>
<td>Stormwolf</td>
<td>727 484.340M</td>
<td>Stormwolf</td>
<td>28</td>
</tr>
</tbody>
</table>
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”.

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

**Functional Dependencies**

`did → dreadType`

**Sample Data**

<table>
<thead>
<tr>
<th>did</th>
<th>dreadtype</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Venerable</td>
</tr>
<tr>
<td>2</td>
<td>Siege</td>
</tr>
<tr>
<td>3</td>
<td>Ironclad</td>
</tr>
<tr>
<td>4</td>
<td>Hellfire</td>
</tr>
<tr>
<td>5</td>
<td>Contemptor</td>
</tr>
<tr>
<td>6</td>
<td>Decade</td>
</tr>
<tr>
<td>7</td>
<td>Librarian</td>
</tr>
<tr>
<td>8</td>
<td>Chaplain</td>
</tr>
<tr>
<td>9</td>
<td>Leviathan</td>
</tr>
</tbody>
</table>
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

<table>
<thead>
<tr>
<th>eid</th>
<th>type</th>
<th>mrkDesignation</th>
<th>eName</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Weapon VB</td>
<td></td>
<td>Godwyn Pattern Bolter</td>
</tr>
<tr>
<td>2</td>
<td>Weapon IV</td>
<td></td>
<td>Ultra Pattern Bolter</td>
</tr>
<tr>
<td>3</td>
<td>Weapon I</td>
<td></td>
<td>Lighting Claws</td>
</tr>
<tr>
<td>4</td>
<td>Weapon I</td>
<td></td>
<td>Force Sword</td>
</tr>
<tr>
<td>5</td>
<td>Weapon I</td>
<td></td>
<td>Doom Hammer</td>
</tr>
<tr>
<td>6</td>
<td>Weapon MF</td>
<td></td>
<td>Brokentooth Pattern Chainsword</td>
</tr>
<tr>
<td>7</td>
<td>Armour I</td>
<td></td>
<td>Thunder Pattern Power Armour</td>
</tr>
<tr>
<td>8</td>
<td>Armour II</td>
<td></td>
<td>Crusade Pattern Power Armour</td>
</tr>
<tr>
<td>9</td>
<td>Armour III</td>
<td></td>
<td>Iron Pattern Power Armour</td>
</tr>
</tbody>
</table>
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)

```sql
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**

<table>
<thead>
<tr>
<th>oid</th>
<th>integer</th>
<th>name text</th>
<th>name text</th>
<th>chaptername text</th>
<th>primarchname text</th>
<th>servicestart text</th>
<th>gsid integer</th>
<th>dateofimplant text</th>
<th>companyname text</th>
<th>specialization text</th>
<th>battlecolors text</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Bran</td>
<td>Redkay</td>
<td>Vlka Fenyka</td>
<td>Lenan Russ</td>
<td>099 158.M33</td>
<td>1 318 077.M32</td>
<td>Bloodmaws Tactical</td>
<td>Blue-Grey</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Engir</td>
<td>Krakendoon</td>
<td>Vlka Fenyka</td>
<td>Lenan Russ</td>
<td>325 969.M40</td>
<td>2 911 219.M39</td>
<td>Seawolves Assault</td>
<td>Blue-Grey</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Harald</td>
<td>Deathwolf</td>
<td>Vlka Fenyka</td>
<td>Lenan Russ</td>
<td>481 760.M39</td>
<td>5 569 896.M38</td>
<td>Deathwolves Demolisher</td>
<td>Blue-Grey</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Bjorn</td>
<td>Stormwolf</td>
<td>Vlka Fenyka</td>
<td>Lenan Russ</td>
<td>727 484.M32</td>
<td>6 616 102.M31</td>
<td>Stormwolves Calvary</td>
<td>Blue-Grey</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Krom</td>
<td>Dragongaze</td>
<td>Vlka Fenyka</td>
<td>Lenan Russ</td>
<td>758 253.M35</td>
<td>8 210 645.M34</td>
<td>Drakeslayers Assault</td>
<td>Blue-Grey</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Create View Statements

*dreadnoughtInfo View*

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

```sql
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;
```

<table>
<thead>
<tr>
<th>aid</th>
<th>integer</th>
<th>fname</th>
<th>dreadtype</th>
<th>servicestart</th>
<th>internmentdate</th>
<th>companyname</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>Kverlaf</td>
<td>Murderfang</td>
<td>Ironclad</td>
<td>273</td>
<td>678.M34</td>
<td>Company of the Great Wolf</td>
</tr>
</tbody>
</table>
Create View Statements

**spacemarineEquipment View**

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

```sql
CREATE OR REPLACE VIEW spacemarineEquipment AS
SELECT aid,
    fname,
    lname,
    eid,
    mrkdesignation,
    ename
FROM spacemarineWeapons
UNION
SELECT aid,
    fname,
    lname,
    eid,
    mrkdesignation,
    ename
FROM spacemarineArmour
UNION
SELECT aid,
    fname,
    lname,
    eid,
    mrkdesignation,
    ename
FROM spacemarineMods;
```
## Create View Statements

**spacemarineEquipment View - Results**

<table>
<thead>
<tr>
<th>aid</th>
<th>integer</th>
<th>fname text</th>
<th>name text</th>
<th>fid integer</th>
<th>mkdesignation character(12)</th>
<th>ename text</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>Bulwye</td>
<td>Axeman of Russ</td>
<td>50</td>
<td>X</td>
<td>Iron Halo</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Logan</td>
<td>Grimnar</td>
<td>53</td>
<td>III</td>
<td>Teleporter</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Egil</td>
<td>Iron Wolf</td>
<td>50</td>
<td>X</td>
<td>Iron Halo</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Ragnar</td>
<td>Blackmane</td>
<td>4</td>
<td>I</td>
<td>Force Sword</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Ragnar</td>
<td>Blackmane</td>
<td>18</td>
<td>IV</td>
<td>Tarratas Pattern Tactical Dreadnought Armour</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Sven</td>
<td>Bloodhowl</td>
<td>14</td>
<td>VIII</td>
<td>Errant Pattern Power Armour</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Erik</td>
<td>Morkai</td>
<td>51</td>
<td>VI</td>
<td>Machine Spirit Core</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Gunnar</td>
<td>Red Moon</td>
<td>4</td>
<td>I</td>
<td>Force Sword</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Bjorn</td>
<td>Stormwolf</td>
<td>1</td>
<td>VB</td>
<td>Godwyn Pattern Bolter</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Gunnar</td>
<td>Red Moon</td>
<td>9</td>
<td>III</td>
<td>Iron Pattern Power Armour</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Sven</td>
<td>Bloodhowl</td>
<td>1</td>
<td>VB</td>
<td>Godwyn Pattern Bolter</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Bjorn</td>
<td>Stormwolf</td>
<td>3</td>
<td>I</td>
<td>Lighting Claw</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Kjarl</td>
<td>Grimblood</td>
<td>3</td>
<td>I</td>
<td>Lighting Claw</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Bran</td>
<td>Rednaw</td>
<td>10</td>
<td>IV</td>
<td>Maximus Pattern Power Armour</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Bran</td>
<td>Rednaw</td>
<td>2</td>
<td>IV</td>
<td>Ultra Pattern Bolter</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Logan</td>
<td>Grimnar</td>
<td>19</td>
<td>V</td>
<td>Angis Terminator Armour</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Erik</td>
<td>Morkai</td>
<td>28</td>
<td>I</td>
<td>Night Vision</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Krill</td>
<td>Iron Wolf</td>
<td>5</td>
<td>I</td>
<td>Doom Harmer</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Bran</td>
<td>Rednaw</td>
<td>30</td>
<td>X</td>
<td>Iron Halo</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Erik</td>
<td>Morkai</td>
<td>30</td>
<td>X</td>
<td>Iron Halo</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Bjorn</td>
<td>Stormwolf</td>
<td>19</td>
<td>V</td>
<td>Angis Terminator Armour</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Erik</td>
<td>Morkai</td>
<td>16</td>
<td>II</td>
<td>Indomitus Pattern Tactical Dreadnought Armour</td>
<td></td>
</tr>
</tbody>
</table>
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 gcIssuedArmaments, to retrieve all equipment assigned to a Great Company.

```sql
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

<table>
<thead>
<tr>
<th>gcid</th>
<th>eid</th>
<th>mkdesignation</th>
<th>ename text</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>33</td>
<td>III</td>
<td>Teleporter</td>
</tr>
<tr>
<td>11</td>
<td>30</td>
<td>X</td>
<td>Iron Halo</td>
</tr>
<tr>
<td>1</td>
<td>32</td>
<td>IX</td>
<td>Stealth Drive</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>IV</td>
<td>Ultra Pattern Bolter</td>
</tr>
<tr>
<td>10</td>
<td>32</td>
<td>IX</td>
<td>Stealth Drive</td>
</tr>
<tr>
<td>11</td>
<td>13</td>
<td>VII</td>
<td>Aquila Pattern Power Armour</td>
</tr>
<tr>
<td>2</td>
<td>33</td>
<td>III</td>
<td>Teleporter</td>
</tr>
<tr>
<td>8</td>
<td>29</td>
<td>V</td>
<td>Stormwind Pattern Jetpack</td>
</tr>
<tr>
<td>8</td>
<td>13</td>
<td>VII</td>
<td>Aquila Pattern Power Armour</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>IV</td>
<td>Ultra Pattern Bolter</td>
</tr>
<tr>
<td>9</td>
<td>19</td>
<td>V</td>
<td>Aegis Terminator Armour</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>I</td>
<td>Force Sword</td>
</tr>
<tr>
<td>9</td>
<td>11</td>
<td>V</td>
<td>Heresy Pattern Power Armour</td>
</tr>
<tr>
<td>10</td>
<td>31</td>
<td>VI</td>
<td>Machine Spirit Core</td>
</tr>
<tr>
<td>3</td>
<td>14</td>
<td>VIII</td>
<td>Errant Pattern Power Armour</td>
</tr>
<tr>
<td>13</td>
<td>11</td>
<td>V</td>
<td>Heresy Pattern Power Armour</td>
</tr>
<tr>
<td>1</td>
<td>31</td>
<td>VI</td>
<td>Machine Spirit Core</td>
</tr>
<tr>
<td>12</td>
<td>28</td>
<td>I</td>
<td>Night Vision</td>
</tr>
<tr>
<td>8</td>
<td>30</td>
<td>X</td>
<td>Iron Halo</td>
</tr>
<tr>
<td>5</td>
<td>33</td>
<td>III</td>
<td>Teleporter</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>VI</td>
<td>Corcus Pattern Power Armour</td>
</tr>
<tr>
<td>1</td>
<td>28</td>
<td>I</td>
<td>Night Vision</td>
</tr>
</tbody>
</table>
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;

<table>
<thead>
<tr>
<th>aid</th>
<th>fname</th>
<th>lname</th>
<th>chaptername</th>
<th>primarchname</th>
<th>servicestart</th>
<th>gsid</th>
<th>dateofimplant</th>
<th>companyname</th>
<th>rankname</th>
<th>specialization</th>
<th>battlecolors</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Harald Deathwolf</td>
<td>Vlka Fenryka</td>
<td>Leman Russ</td>
<td>481 760.M39</td>
<td>5 569 896.M38</td>
<td>Deathwolves</td>
<td>Wolf Lord</td>
<td>Demolisher</td>
<td>Blue-Grey</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>aid</th>
<th>fname</th>
<th>lname</th>
<th>chaptername</th>
<th>primarchname</th>
<th>servicestart</th>
<th>gsid</th>
<th>dateofimplant</th>
<th>companyname</th>
<th>rankname</th>
<th>specialization</th>
<th>battlecolors</th>
</tr>
</thead>
</table>
Stored Procedures - Get Space Marine Information by Name

**getSPEquipByNome 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.

```sql
CREATE OR REPLACE FUNCTION getSPEquipByNome(TEXT, TEXT, REFCURSOR) RETURNS refcursor AS $$
DECLARE
  spFname TEXT := $1;
  spLname TEXT := $2;
  resultset REFCURSOR := $3;
BEGIN
  OPEN resultset FOR
  SELECT eid, mrkdesignation, ename
  FROM spacemarineEquipment
  WHERE fname LIKE spFname
  AND lname LIKE spLname;
  return resultset;
end;
$$
LANGUAGE plpgsql;
```
Stored Procedures - Get Space Marine Equipment by Name

getSPEquipByName Procedure - Results

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

<table>
<thead>
<tr>
<th>cid</th>
<th>mrkdesignation character(12)</th>
<th>ename</th>
<th>text</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 I</td>
<td>Doom Hammer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 IV</td>
<td>Ultra Pattern Bolter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5 II</td>
<td>Crusade Pattern Power Armour</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>cid</th>
<th>mrkdesignation character(12)</th>
<th>ename</th>
<th>text</th>
</tr>
</thead>
<tbody>
<tr>
<td>3 I</td>
<td>Lighting Claws</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30 X</td>
<td>Iron Halo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>21 I</td>
<td>Night Vision</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 VB</td>
<td>Godwyn Pattern Bolter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>19 V</td>
<td>Aegis Terminator Armour</td>
<td></td>
<td></td>
</tr>
<tr>
<td>31 VI</td>
<td>Machine Spirit Core</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 $$
DECLARE
    dnFname TEXT := $1;
    dnLname TEXT := $2;
    resultset REFCURSOR := $3;
BEGIN
    OPEN resultset FOR
        SELECT *
        FROM dreadnoughtInfo
        WHERE fname LIKE dnFname
        AND lname LIKE dnLname;
    return resultset;
end;
$$
LANGUAGE plpgsql;
```
Stored Procedures - Get Dreadnought Information by Name

`getDNInfoByName` Procedure - Results

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

<table>
<thead>
<tr>
<th>aid</th>
<th>fname</th>
<th>lname</th>
<th>dreadtype</th>
<th>servicestar</th>
<th>internmentdate</th>
<th>companyname</th>
</tr>
</thead>
</table>

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

<table>
<thead>
<tr>
<th>aid</th>
<th>fname</th>
<th>lname</th>
<th>dreadtype</th>
<th>servicestar</th>
<th>internmentdate</th>
<th>companyname</th>
</tr>
</thead>
</table>
**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 → 27
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.

```sql
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.

```sql
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.

```sql
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.

```sql
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