Super Star Destroyer (SSD) *Executor* of the Galactic Empire

Database proposal

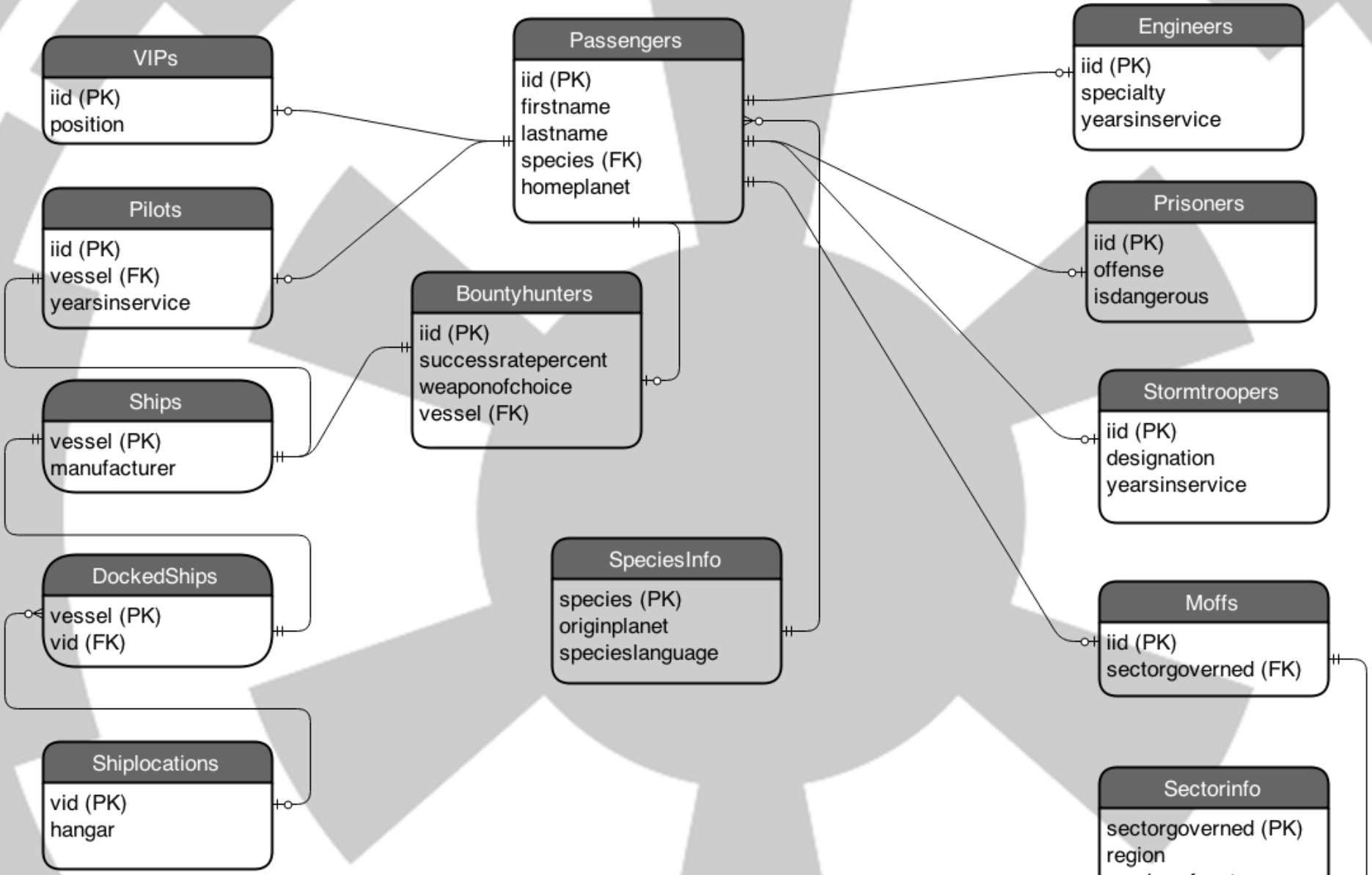
By Nicholas Marengo

The *Executor*

- The Super Star Destroyer *Executor* is the personal flagship of Lord Darth Vader, the premier command ship of the Imperial Navy, and the first of the *Executor-class* Star Destroyer line.
- At 19,000 meters in length, it is the largest traditional warship the galaxy has ever seen.
- The *Executor* boasts over 5,000 weapon placements as well as the most capable military compliment of the Empire.
- The primary hangar bay is large enough to dock an entire Imperial Star Destroyer, as well as countless other starfighters and freighters.

- This document outlines the structure of a system needed to create and implement a database to suit the needs of this Super Star Destroyer. By logging all of the personnel onboard, this database can keep track of their roles and attributes as they strive to maintain the vessel and make sure that it runs at maximum efficiency, for the glory of the Empire. Administrators and other authorized Imperial personnel will have the ability to generate necessary information from queries and such that will allow them to keep the *Executor* functioning smoothly.

  This database design allows administrators to track all Imperial and non-Imperial personnel on this vessel: engineers, bounty hunters, those serving in the Stormtrooper Corps, etc, all have an ID issued by the Empire to allow for better tracking and documentation. Ships, whether Imperial or non-Imperial have an Imperial issued ID as well. All in all, it is in Lord Vader’s best interest to implement this fully functional and useful database system.



Entity-Relational Diagram

Passengers – lists all personnel onboard and their basic attributes

```
CREATE TABLE passengers
(
    iid text NOT NULL,
    firstname text,
    lastname text,
    species text,
    homeplanet text,
    CONSTRAINT passengers_pkey PRIMARY KEY (iid),
    CONSTRAINT passengers_species_fkey FOREIGN KEY (species)
        REFERENCES speciesinfo (species) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

Functional Dependencies:

iid→firstname, lastname, species, homeplanet

Sample data can be found on the next page

Passengers –

Sample data

	iid [PK] text	firstname text	lastname text	species text	homeplanet text
1	B07	Tak	Bazierre	Human	Coruscant
2	B16	Shira	Brie	Human	Coruscant
3	B27	Cad	Bane	Duros	Duro
4	D14	Biggs	Darklighter	Human	Tatooine
5	D26	Bossh	Dosh	Trandoshan	Trandoshia
6	E15	Juno	Eclipse	Human	Corulag
7	F12	Davin	Felth	Human	Carida
8	F13	Soontir	Fel	Human	Corellia
9	F23	Boba	Fett	Mandalorian	Kamino
10	G24	IG	88	Assassin Droid	Halowan
11	H21	Han	Solo	Human	Corellia
12	J06	Tiaan	Jerjerrod	Human	Tinnel IV
13	K08	Kir	Kanos	Human	Coruscant
14	K25	Kenix	Kil	Human	Coruscant
15	L18	Umak	Leth	Human	Coruscant
16	M19	Dusque	Mistflier	Human	Talus
17	O22	Leia	Organa	Human	Alderaan
18	P17	Turr	Phennir	Human	Valahari
19	P20	Elo	Panil	Kaminoan	Kamino
20	Q28	Alessi	Quon	Sluissi	Sluis Van
21	S09	Nova	Stihl	Human	Dantooine
22	T02	Wilhuff	Tarkin	Human	Eriadu
23	T04	Derran	Takkar	Human	Coruscant
24	T05	Leonia	Tavira	Human	Eiattu 6
25	T10	Brenn	Tantor	Human	Garos IV
26	T11	Hume	Tarl	Human	Corellia
27	V01	Darth	Vader	Human	Tatooine
28	V03	Trachta	Ventor	Human	Coruscant
29	V31	Elon	Vedij	Human	Coruscant
30	W30	Lira	Wessex	Human	Kuat
31	X29	Qwi	Xux	Omwati	Omwat

VIPs and Moffs

```
CREATE TABLE vips
(
    iid text NOT NULL,
    "position" text,
    CONSTRAINT vips_pkey PRIMARY KEY (iid)
)
```

	iid [PK] text	position text
1	V01	Supreme Commander of Imperial Forces

Functional Dependencies:

iid → position

```
CREATE TABLE moffs
(
    iid text NOT NULL,
    sectorgoverned text,
    CONSTRAINT moffs_pkey PRIMARY KEY (iid),
    CONSTRAINT moffs_sectorgoverned_fkey FOREIGN KEY (sectorgoverned)
        REFERENCES sectorinfo (sectorgoverned) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

	iid [PK] text	sectorgoverned text
1	J06	Quanta Sector
2	T02	Seswenna Sector
3	T04	Tapani Sector
4	T05	Ado Sector
5	V03	Imperial Center Oversector

Functional Dependencies:

iid→ sectorgoverned

Pilots and Engineers

```
CREATE TABLE pilots
(
    iid text NOT NULL,
    vessel text,
    yearsinservice integer,
    CONSTRAINT pilots_pkey PRIMARY KEY (iid),
    CONSTRAINT pilots_vessel_fkey FOREIGN KEY (vessel)
        REFERENCES ships (vessel) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

	iid [PK] text	vessel text	yearsinservice integer
1	B16	TIE Fighter DS-38	15
2	D14	TIE Fighter DS-91	2
3	E15	TIE Interceptor DS-51	6
4	F13	TIE Fighter DS-06	11
5	P17	TIE Bomber DS-09	11

Functional Dependencies:
iid → vessel, yearsinservice

```
CREATE TABLE engineers
(
    iid text NOT NULL,
    specialty text,
    yearsinservice text,
    CONSTRAINT engineers_pkey PRIMARY KEY (iid)
)
```

	iid [PK] text	specialty text	yearsinservice text
1	L18	weaponry	12
2	M19	bioengineering	4
3	P20	weaponry	1

Functional Dependencies:
iid → specialty, yearsinservice

Stormtroopers and Prisoners

```
CREATE TABLE stormtroopers
(
    iid text NOT NULL,
    designation text,
    yearsinservice integer,
    CONSTRAINT stormtroopers_pkey PRIMARY KEY (iid)
)
```

	iid [PK] text	designation text	yearsinservice integer
1	B07	Assault Trooper	10
2	F12	EVO Trooper	6
3	K08	Elite Guard	15
4	S09	Scout Trooper	7
5	T10	Imperial Marine	4
6	T11	Shock Trooper	11

Functional Dependencies:
iid → designation, yearsinservice

```
CREATE TABLE prisoners
(
    iid text NOT NULL,
    offense text,
    isdangerous boolean,
    CONSTRAINT prisoners_pkey PRIMARY KEY (iid)
)
```

	iid [PK] text	offense text	isdangerous boolean
1	H21	treason	TRUE
2	O22	treason	FALSE

Functional Dependencies:
iid → offense, isdangerous

Bounty Hunters

```
CREATE TABLE bountyhunters
(
    iid text NOT NULL,
    successratepercent integer,
    weaponofchoice text,
    vessel text,
    CONSTRAINT bountyhunters_pkey PRIMARY KEY (iid),
    CONSTRAINT bountyhunters_vessel_fkey FOREIGN KEY (vessel)
        REFERENCES ships (vessel) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

Functional Dependencies:

iid → successratepercent, weaponofchoice, vessel

	iid [PK] text	successratepercent integer	weaponofchoice text	vessel text
1	B27	90	LL-30 Blaster Pistols	Sleight of Hand
2	D26	85	Relby-v10 Microgrenade Launcher	Hounds Tooth
3	F23	95	EE-3 Carbine Rifle	Slave 1
4	G24	75	DLT-20A Blaster Rifle	IG-2000
5	K25	70	Force Pike	A-Wing

Ships & Docked Ships

```
CREATE TABLE ships
(
    vessel text NOT NULL,
    manufacturer text,
    CONSTRAINT ships_pkey PRIMARY KEY (vessel)
)
```

Functional Dependencies:
vessel → manufacturer

	vessel [PK] text	manufacturer text
1	A-Wing	Incom Corporation
2	Hounds Tooth	Corellian Engineering Corporation
3	IG-2000	Mechis III
4	Slave 1	Kuat Systems Engineering
5	Sleight of Hand	Telgorn Corporation
6	TIE Bomber DS-09	Sienar Fleet Systems
7	TIE Fighter DS-06	Sienar Fleet Systems
8	TIE Fighter DS-38	Sienar Fleet Systems
9	TIE Fighter DS-91	Sienar Fleet Systems
10	TIE Interceptor DS-51	Sienar Fleet Systems

```
CREATE TABLE dockedships
(
    vessel text NOT NULL,
    vid text,
    CONSTRAINT dockedships_pkey PRIMARY KEY (vessel),
    CONSTRAINT dockedships_vid_fkey FOREIGN KEY (vid)
        REFERENCES shiplocations (vid) MATCH SIMPLE
        ON UPDATE NO ACTION ON DELETE NO ACTION
)
```

Functional Dependencies:
vessel → vid

	vessel [PK] text	vid text
1	A-Wing	AW03
2	Hounds Tooth	HT4
3	IG-2000	IG2000
4	Slave 1	SL1
5	Sleight of Hand	SLH2
6	TIE Bomber DS-09	TF09
7	TIE Fighter DS-06	TF06
8	TIE Fighter DS-38	TF38
9	TIE Fighter DS-91	TF91
10	TIE Interceptor DS-51	TF51

Ship Locations

```
CREATE TABLE shiplocations
(
    vid text NOT NULL,
    hangar integer,
    CONSTRAINT shiplocations_pkey PRIMARY KEY (vid)
)
```

Functional Dependencies:

vid → hangar

	vid [PK] text	hangar integer
1	AW03	19
2	HT4	17
3	IG2000	13
4	SL1	17
5	SLH2	15
6	TF06	8
7	TF09	12
8	TF38	9
9	TF51	6
10	TF91	12

Species Info

```
CREATE TABLE speciesinfo
(
    species text NOT NULL,
    originplanet text,
    specieslanguage text,
    CONSTRAINT speciesinfo_pkey PRIMARY KEY (species)
)
```

Functional Dependencies:

species → originplanet, specieslanguage

	species [PK] text	originplanet text	specieslanguage text
1	Assassin Droid	Halowan	Basic
2	Bothan	Bothawui	Bothese
3	Duros	Duro	Durese
4	Gand	Gand	Gand
5	Human	Coruscant	Basic
6	Kaminoan	Kamino	Kaminoan
7	Mandalorian	Mandalore	Mando'a
8	Omwati	Omwat	Omwatese
9	Quarren	Mon Calamri	Quarrenese
10	Rodian	Rodia	Rodese
11	Sluissi	Sluis Van	Sluissese
12	Sullustan	Sullust	Sullustese
13	Trandoshan	Trandoshia	Saurin
14	Wookie	Kashyyyk	Shyriiwook

SectorInfo

```
CREATE TABLE sectorinfo
(
    sectorgoverned text NOT NULL,
    region text,
    numberofsystems integer,
    numberoftraderoutes integer,
    sectorcapital text,
    CONSTRAINT sectorinfo_pkey PRIMARY KEY (sectorgoverned)
)
```

Functional Dependencies:

sectorgoverned → region, numberofsystems, numberoftraderoutes, sectorcapital

	sectorgoverned [PK] text	region text	numberofsystems integer	numberoftraderoutes integer	sectorcapital text
1	Ado Sector	Mid Rim Territories	11	3	Eiattu
2	Imperial Center Oversector	Deep Core	3	2	Anaxes
3	Quanta Sector	Core Worlds	1	1	Tinnel IV
4	Seswenna Sector	Outer Rim Territories	21	4	Eriadu
5	Tapani Sector	Colonies Territory	98	9	Procopia

Views

View that displays ships and their hangar locations onboard the *Executor.*

```
--Ships and their locations--
CREATE VIEW whereships AS
SELECT vessel, hangar
FROM shiplocations
INNER JOIN dockedships
ON dockedships.vid = shiplocations.vid
```

	vessel text	hangar integer
1	TIE Fighter DS-06	8
2	TIE Fighter DS-91	12
3	TIE Interceptor DS-51	6
4	TIE Fighter DS-38	9
5	TIE Bomber DS-09	12
6	Slave 1	17
7	IG-2000	13
8	A-Wing	19
9	Hounds Tooth	17
10	Sleight of Hand	15

Hangars 13 – 19 are reserved for non-imperial vessels.

View that displays the Bounty Hunters onboard in order of their success.

```
--Bounty Hunters in order of their successrate--
CREATE VIEW highestsuccessrate AS
SELECT firstname, lastname, successratepercent
FROM bountyhunters
INNER JOIN passengers
ON passengers.iid = bountyhunters.iid
ORDER BY successratepercent DESC
```

	firstname text	lastname text	successratepercent integer
1	Boba	Fett	95
2	Cad	Bane	90
3	Bossh	Dosh	85
4	IG	88	75
5	Kenix	Kil	70

View that displays Moffs in order of how big their governed sector is,
i.e., in order of their governing power

```
--Moffs in order of power (how big their governed sector is)--
CREATE VIEW moffspower AS
select firstname, lastname, moffs.sectorgoverned
from moffs
inner join passengers
on passengers.iid = moffs.iid
inner join sectorinfo
on moffs.sectorgoverned = sectorinfo.sectorgoverned
order by numberofsystems desc
```

	firstname text	lastname text	sectorgoverned text
1	Derran	Takkar	Tapani Sector
2	Wilhuff	Tarkin	Seswenna Sector
3	Leonia	Tavira	Ado Sector
4	Trachta	Ventor	Imperial Center Oversector
5	Tiaan	Jerjerrod	Quanta Sector

Let it be noted that even though Trachta is the Moff of the Imperial Center (the Coruscant area), he is far from the most powerful Moff. Grand Moffs Takkar and Tarkin seem to pull the most weight.

Reports

Information about personnel onboard the *Executor*

- Query to show the (rounded) percent of passengers who are nonhuman, so the Empire can keep an eye on them.

```
--Query to show percent of how many passengers are not human--
select round(cast((x.number * 100) as float) / cast((y.number * 100) AS float) * 100) as percentnonhuman
from
(
        select count(iid) as number
        from passengers
        where species <> 'Human'
) x
join
(
        select count(iid) as number
        from passengers

) y on 1=1
```

	percentnonhuman double precision
1	23

Reports

Information about personnel onboard the *Executor*

Query to show passengers onboard who are not part of the Galactic Empire
This was made by adding together the Bounty Hunter and Prisoner populations

```
--Query to show percent of passengers who are not of the Galactic Empire--
select round(cast((nonimperials * 100) as float) / cast((z.number * 100) as float) *100) as percentnonimperial
from
(select trunc(x.number + y.number) as nonimperials
from
(
        select count(iid) as number
        from bountyhunters
) x
join
(
        select count(iid) as number
        from prisoners
) y on 1=1) as notimperials
join
(
        select count(iid) as number
        from passengers
) z on 1=1
```

	percentnonimperial double precision
1	23

I would like to add that it happens to be plain coincidence that the percent of non imperials onboard as well as the percent of non humans happens to be the same number; as seen, the information came from different tables with different populations.
It is a small galaxy after all.

It is to be noted that even though the *Executor's* two current prisoners identify with the Rebel Alliance, one's imperial citizenship is revoked once imprisoned indefinitely. This means that if, lets say, a stormtrooper were to be imprisoned for whatever purpose, their imperial citizenship would be revoked and they would be counted on this list.

Reports

Information about personnel onboard the *Executor*

Displays percent of all personnel onboard who are combat ready to serve the Empire, whether for defense of this cruiser, space combat, or land invasion. This is taken from the amount of Stormtroopers and Imperial Pilots.

```
--combat ready personnel
select round(cast((combatpersonnel * 100) as float) / cast((z.number * 100) as float) *100) as percentcombatready
from
(select trunc(x.number + y.number) as combatpersonnel
from
(
        select count(iid) as number
        from stormtroopers
) x
join
(
        select count(iid) as number
        from pilots
) y on 1=1) as combatready
join
(
        select count(iid) as number
        from passengers
) z on 1=1
```

	percentcombatready double precision
1	35

It is to be noted that VIPs, such as Lord Vader, as well as Bounty Hunters are not obligated to serve on this vessel. Even though they are more than combat trained, they are not counted.

This also assumes that all Imperial Pilots are bred for combat – in the Empire, even transportation vessels are outfitted with some kind of weaponry for defense.

Stored Procedures

Functions that allow for easy information gathering

Given the first name of a Bounty Hunter, shows in which hangar their ship is docked.

```
--Given the firstname of a Bounty Hunter, shows where their ship is docked--
CREATE OR REPLACE FUNCTION wheredocked(text, refcursor) returns refcursor as
$$
declare
        bountyfirstname text :=$1;
        resultset refcursor :=$2;
begin
open resultset for

select hangar
from shiplocations
where vid = (

        select vid
        from dockedships
        where vessel = (

                select vessel
                from bountyhunters
                where iid = (

                        select iid
                        from bountyhunters
                        where iid = (

                                select iid
                                from passengers
                                where bountyfirstname = firstname
                                ))));
                                return resultset;
end;
$$
language plpgsql;
```

```
select wheredocked('Boba', 'results');
fetch all from results;
```

	hangar integer
1	17

Given a docked ship's VID, displays it's name.

```
--Given the VID of a vessel, shows the name of the vessel--
CREATE OR REPLACE FUNCTION vid(text, refcursor) returns refcursor as
$$
declare
        vesselid text:=$1;
        resultset refcursor :=$2;
begin
        open resultset for
        select vessel
        from dockedships
        where vesselid = vid
        limit 1;
        return resultset;
end;
$$
language plpgsql;
```

```
select vid('HT4', 'results');
fetch all from results;
```

	vessel text
1	Hounds Tooth

Stored Procedures

Functions that allow for easy information gathering

Given the capitol planet of a sector, this function displays which Moff rules it, making it easy to contact the proper administration when looking for certain authorization in a given world or sector.

```
--Given a sector capital, shows which Moff governs it--
create or replace function whichmoff(text, refcursor) returns refcursor as
$$
declare
        capitalplanet text :=$1;
        resultset refcursor :=$2;
begin
open resultset for
select firstname, lastname
from passengers
where iid = (

        select iid
        from moffs
        where sectorgoverned = (

                select sectorgoverned
                from sectorinfo
                where capitalplanet = sectorcapital));
                return resultset;
end;
$$
language plpgsql;
```

```
select whichmoff('Procopia', 'results');
fetch all from results;
```

	firstname text	lastname text
1	Derran	Takkar

Security

Identifies which roles personnel can have and the privileges of those roles

```
--Security--
--VIPs and Moffs--
create role admins;
grant all on all tables
in schema public
to admins;

--Imperial Administrators--
create role imperials;
grant select on bountyhunters, dockedships, engineers,
                moffs, passengers, pilots,
                prisoners, sectorinfo, shiplocations, ships,
                speciesinfo, stormtroopers, vips
to imperials;
grant insert on bountyhunters, dockedships, engineers,
                passengers, pilots, prisoners,
                sectorinfo, shiplocations, ships, stormtroopers
to imperials;
grant update on bountyhunters, dockedships, prisoners, sectorinfo,
                shiplocations, ships
to imperials;

--Non imperials--
create role nonimperials;
grant select on sectorinfo, shiplocations, ships, speciesinfo
to nonimperials;
grant insert on ships
to nonimperials;
```

Obviously, the VIPs and the Moffs onboard the *Executor* have direct authority to do as they please with the system.

Next on the hierarchy comes Imperial Personnel, who have many privileges but not as many as their higher ups.

Non Imperial personnel have nearly no privileges; they can only enjoy informational databases that do not reveal any of the passengers onboard.

Notes, Problems, Future Enhancements.

- Obviously, I could not include as much data as the "real" *Executor* Super Star Destroyer Dreadnaught.
- On average, there were over 280,000 personnel onboard the Executor, including 144 Starfighters, 200 various assault craft, and a compliment of 39,000 troopers. My database has 31 passengers and 10 ships.
- There are not many problems with this database, though I would like to address that there are so many other tables I could have possibly created that would have added depth, such as one for ground craft, non-military personnel, weapons placements, or maybe even a table that displayed where certain personnel were stationed onboard.
- Something else that could have improved this database but seemed extremely tedious would have been to link the home planet each passenger is from with a sector that a Moff governs. This would be difficult as there could exist home planets that a Moff does not rule (for example, if the planet was aligned with the Rebel Alliance). There are also so many sectors in the galaxy with so many systems that it is way too much to put into one database. I was surprised when I looked up the Tapani Sector to find that it had over 90 systems of planets, and that, after the Ruusan Reformations of 1000BBY (years before the battle of Yavin), the galaxy was reorganized into 1,024 regional sectors.
- I would have loved to add a table of bounties that the Bounty Hunters are currently hunting. In the format that this database is made in, this would be difficult to implement because all of the personnel are stored in one table, and that is the passengers table. I would assume that the bounties they are after are not onboard the same vessel that they are traveling in, and therefore they cannot be in this database.

Notes, Problems, Future Enhancements

- Inconsistencies: Stormtrooper Kir Kanos and Bounty Hunter Kenix Kil are the same person.
  Elo Panil, a Kaminoan engineer, would probably never be stationed onboard a Star Destroyer. (By this time, the Kaminoans' cloning and bioengineering technology were no longer of use to the Empire, and it is noted that the first alien to achieve high status among the Empire was Grand Admiral Thrawn who could not be in my database because he was promoted after Darth Vader and Palpatine's deaths).
  Shira Brie, later known as the Dark Lady of the Sith Lumiya, was also only an imperial pilot for a short time, in which she served on Coruscant before being promoted to the Emperor's exclusive circle, and probably never would have been stationed aboard the *Executor*.

All in all, I would like to thank Wookiepedia.org for all of the information needed to complete this database design proposal. All characters, ships, and other relevant information are from the Star Wars Expanded Universe, and none are made up by myself.