52-8305

39-3995

12-5638

96-7477

95-7221

75-2028

# DATABASE: THE NEXT GENERATION

## DESIGNED BY ALLIYAH TAYLOR

# TABLE OF CONTENTS

Alliyah Taylor

# EXECUTIVE SUMMARY

THIS DOCUMENT DESCRIBES THE DESIGN AND IMPLEMENTATION OF A DATABASE WHICH TRACKS ALL KNOWN SHIPS IN FEDERATION SPACE INCLUDING CREW, CARGO, AND SYSTEMS ABOARD THE SHIPS. FIRST THE ER DIAGRAM OF THE DATABASE WILL BE SHOWN, FOLLOWED BY A DESCRIPTION OF THE INDIVIDUAL TABLES, INCLUDING THEIR SQL CREATE STATEMENTS AND FUNCTIONAL DEPENDENCIES. SAMPLE DATED FOR EACH TABLE WILL BE PROVIDED. NEXT, THE VIEWS, REPORTS, AND STORED PROCEDURES FROM THE DATABASE WILL BE SHOWN ALONG WITH A SAMPLE OUTPUT. AFTER THIS, THERE WILL BE A DESCRIPTION OF SUGGESTED ROLES AND SECURITY PRIVILEGES FOR THE DATABASE. THE DOCUMENT WILL CLOSE WITH IMPLEMENTATION NOTES, KNOWN ISSUES, AND IDEAS FOR FUTURE IMPROVEMENT.

THIS DATABASE'S INTENDED USE IS FOR STARFLEET OFFICIALS TO TRACK SHIPS IN FEDERATION SPACE. THIS SHIP WILL ALLOW FOR BETTER ORGANIZATION OF INFORMATION REGARDING THE PERSONNEL, CARGO, SYSTEMS, AND LOCATION OF SHIPS IN FEDERATION TERRITORY. STARFLEET WILL BE ABLE TO USE QUERIES TO GAIN VALUABLE DATA ON THE STATUS OF SHIPS THAT CAN BE USED IN DEVELOPING TACTICS AND GIVING ORDERS. THE GOAL OF THIS DATABASE IMPLEMENTATION IS TO GIVE STARFLEET A FUNCTIONAL, NORMALIZED DATABASE THAT WILL STREAMLINE THE PROCESS OF KEEPING TRACK OF THE SHIPS IN FEDERATION TERRITORY.

Alliyah Taylor

E-R DIAGRAM

Alliyah Taylor

# TABLES

61554 710516 2479 472037 485 21 25962 7180055 73 33 44 220 69 701050054 2 13920157 8370 61320 58461 4924 5005 24399000 5759 9 89080553 96071953 631274690 7433407 313 1 701
29053 675071 9192 810633 984 50 33294 1127826 84 82 65 451 87 187045918 7 30131629 5292 20002 32841 1408 1367 35070814 9070 8 96011565 59031530 652106697 8114032 782 3 602
89782 7607614 42 61 97 847 98 078853892 6 73996511 9311 71423 59901 6920 5541 43120494 6803 4 13727316 52535981 411346137 5317388 954 0 263
79947 8737686 19 36 99 570 15 715686443 7 45658975 8653 54570 97397 3690 1884 52076545 2426 6 49435201 74135744 138263486 3835686 641 8 22794 1583 1976875 75076 983
24148 6107847 72 55 30 250 87 582165187 1 31592575 6269 37404 51438 7270 4868 68745300 3339 3 98345379 78754481 563945657 8064904 624 6 10829 6902 12252320 70440 224
19741 307454 5430 282037 720 86176 5152738 79 51 345 28 559921496 2 27328805 9448 01100 2304 7984 85089680 6442 5 92751037 76138823 521419162 6697335 213 5 67188 6937 60070702 39071 901

## BEINGS: THE BEINGS TABLE CONTAINS ALL SENTIENT BEINGS AND THEIR BASIC ATTRIBUTES, WHICH ARE SHARED WITH THE CREW, PASSENGERS, AND PRISONERS SUBTYPES.

05

```
CREATE TABLE BEINGS(

    BEINGID      INTEGER NOT NULL,
    FNAME        TEXT NOT NULL,
    LNAME        TEXT,
    SPECIES      TEXT NOT NULL,
    AGE          INTEGER,
    ORIGINPLANET TEXT,
    PRIMARY KEY (BEINGID)
);
```

FUNCTIONAL DEPENDENCIES:

BEINGID → FNAME, LNAME, SPECIES, AGE, ORIGINPLANET

Alliyah Taylor

| beingid integer | fname text | lname text | species text | age integer | originplanet text |
|---|---|---|---|---|---|
| 1 | Bok | | Ferengi | 63 | Ferenginar |
| 2 | Kazago | | Ferengi | 41 | Ferenginar |
| 3 | Rata | | Ferengi | 30 | Ferenginar |
| 4 | Lupak | | Ferengi | 52 | Ferenginar |
| 5 | Jean-Luc | Picard | Human | 59 | Earth |
| 6 | Data | Soong | Android | 26 | Omicron Theta |
| 7 | Deanna | Troi | Betazoid/... | 28 | Betazed |
| 8 | Q | | Q | | Q Continuum |
| 9 | Wesley | Crusher | Human | 16 | Earth |
| 10 | One | Zero | Bynar | | Bynaus |
| 11 | Zero | One | Bynar | | Bynaus |
| 12 | Thomas | Pucer | Human | 64 | Earth |
| 13 | Laren | Ro | Bajoran | 24 | Bajor |
| 14 | John | Connors | Human | 32 | Earth |
| 15 | Wyatt | Miller | Human | 30 | Haven |
| 16 | Wrenn | | Tarellian | 54 | Tarella |
| 17 | Ariana | | Tarellian | 28 | Tarella |
| 18 | Tebok | | Romulan | 40 | Romulus |
| 19 | Alan | Labouseur | Human | 49 | Earth |
| 20 | Ael | Mirok | Romulan | 35 | Romulus |

# CREW: SUBTYPE OF BEINGS, CONTAINS ALL PEOPLE THAT ARE CREW ON A SHIP AND INFORMATION ABOUT THEIR POSITION.

```
CREATE TABLE CREW(
        BEINGID        INTEGER NOT NULL REFERENCES BEINGS(BEINGID),
        SHIPRANK       TEXT,
        POSITION       TEXT NOT NULL,
        PRIMARY KEY (BEINGID)
);
```

## FUNCTIONAL DEPENDENCIES:

### BEINGID → SHIPRANK, POSITION

| beingid integer | shiprank text | position text |
|---|---|---|
| 1 | DaiMon | Helm |
| 2 | First Officer | Tactical |
| 3 | Second Officer | Engineering |
| 5 | Captain | Helm |
| 6 | Liutenant Comm... | Second Officer |
| 7 | Commander | Counselor |
| 12 | Captain | Helm |
| 13 | Ensign | Naviagation |
| 14 | Ensign | Tactical |
| 16 | Captain | Helm |
| 18 | Commander | Helm |
| 19 | Centurion | Engineering |
| 20 | Sub-commander | Tactical |

Alliyah Taylor

52-8305

39-3995

12-5638

96-7477

95-7221

75-2828

# PASSENGERS: SUBTYPE OF BEINGS. INCLUDES ALL NON-CREW PASSENGERS OF THE SHIP AND INFORMATION ABOUT THEIR PURPOSE ABOARD THE SHIP.

```
CREATE TABLE PASSENGERS(
        BEINGID                 INTEGER NOT NULL REFERENCES BEINGS(BEINGID),
        ORIGIN                  TEXT,
        DESTINATION             TEXT,
        PURPOSE                 TEXT NOT NULL,
        PRIMARY KEY (BEINGID)
);
```

| beingid integer | origin text | destination text | purpose text |
|---|---|---|---|
| 4 | Farenginar | Risa | Traveling to a vacation destination to make profit. |
| 8 | The Q Continuum | | To try the human race for their crimes. |
| 9 | Enterprise | Enterprise | Family Member of Crew Member Bevery Crusher. |
| 15 | Haven | Neutral Space | To find a cure to the Tarellian plague. |
| 17 | Tarella | Neutral Space | Seeking a peaceful place to live out the Tarellian plague. Family member of crew member |

## FUNCTIONAL DEPENDENCIES:

BEINGID → ORIGIN, DESTINATION, PURPOSE

Alliyah Taylor

## PRISONERS: SUBTYPE OF BEINGS, CONTAINS ALL PEOPLE THAT ARE HELD ON A SHIP AGAINST THEIR WILL AND WHY THEY ARE BEING HELD.

08

```
CREATE TABLE PRISONERS(
        BEINGID                 INTEGER NOT NULL REFERENCES BEINGS(BEINGID),
        ORIGIN                  TEXT NOT NULL,
        DESTINATION             TEXT NOT NULL,
        VALUEFEDCRED            INTEGER,
        REASONHELD              TEXT NOT NULL,
        PRIMARY KEY (BEINGID)
);
```

| beingid<br>integer | origin<br>text | destination<br>text | valuefed...<br>integer | reasonheld<br>text |
|---|---|---|---|---|
| 10 | Bynaus | Starbase 74 | 100000 | Awaiting trial by federation. |
| 11 | Bynaus | Starbase 74 | 100000 | Awaiting trial by federation. |

## FUNCTIONAL DEPENDENCIES:

## BEINGID → ORIGIN, DESTINATION, VALUEFEDCRED, REASONHELD

## SHIPS: THIS TABLE CONTAINS ALL KNOWN SHIPS IN FEDERATION SPACE AND THEIR BASIC INFORMATION.

```
CREATE TABLE SHIPS(
        SHIPID              INTEGER NOT NULL,
        NAME                TEXT NOT NULL,
        DATEENTEREDFED      NUMERIC(6,1) NOT NULL,
        STATUS              TEXT NOT NULL,
        PRIMARY KEY (SHIPID)
);
```

| shipid integer | name text | dateentere... numeric (6,1) | status text |
|---|---|---|---|
| 1 | FMS Chulebas | 41723.9 | Active |
| 2 | USS Enterprise-D | 41153.7 | Active |
| 3 | USS Wellington | 41365.3 | Active |
| 4 | Halsa | 41294.5 | Active |
| 5 | IRW Genorex | 41986 | Active |

## FUNCTIONAL DEPENDENCIES:

### SHIPID → NAME, DATEENTEREDFED, STATUS

Alliyah Taylor

# BEINGSON: THIS TABLE CONNECTS BEINGS TO THE SHIPS THEY ARE ABOARD.

```
CREATE TABLE BEINGSON(

        BEINGID       INTEGER NOT NULL REFERENCES BEINGS(BEINGID),

        SHIPID        INTEGER NOT NULL REFERENCES SHIPS(SHIPID),

        PRIMARY KEY (BEINGID, SHIPID)

);
```

## FUNCTIONAL DEPENDENCIES:

BEINGID, SHIPID→

| beingid integer | shipid integer |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |
| 4 | 1 |
| 5 | 2 |
| 6 | 2 |
| 7 | 2 |
| 8 | 2 |
| 9 | 2 |
| 10 | 2 |
| 11 | 2 |
| 12 | 3 |
| 13 | 3 |
| 14 | 3 |
| 15 | 4 |
| 16 | 4 |
| 17 | 4 |
| 18 | 5 |
| 19 | 5 |
| 20 | 5 |

Alliyah Taylor

# ALLIANCES: THIS TABLE CONTAINS ALL KNOWN ALLIANCES AND THEIR CURRENT RELATIONSHIP WITH THE FEDERATION.

```
CREATE TABLE ALLIANCES(
ALLIANCEID              INTEGER NOT NULL,
NAME                    TEXT NOT NULL,
RELATIONSWITHFED        TEXT,
PRIMARY KEY (ALLIANCEID)
);
```

| allianceid integer | name text | relationswithfed text |
|---|---|---|
| 0 | United Federation of Planets | |
| 1 | Ferengi Alliance | Neutral |
| 2 | Tarellian | Neutral |
| 3 | Romulan Empire | Enemies |
| 4 | The Q Continuum | Contemptuous |
| 5 | Bynaus | Friendly |

## FUNCTIONAL DEPENDENCIES:

### ALLIANCEID → NAME, RELASHIONSWITHFED

Alliyah Taylor

# BELONGSTO: THIS TABLE KEEPS TRACK OF WHAT ALLIANCE A SHIP BELONGS TO.

```
CREATE TABLE BELONGSTO(
        ALLIANCEID          INTEGER NOT NULL REFERENCES ALLIANCES(ALLIANCEID),
        SHIPID              INTEGER NOT NULL REFERENCES SHIPS(SHIPID),
        PRIMARY KEY (SHIPID, ALLIANCEID)
);
```

## FUNCTIONAL DEPENDENCIES:

### ALLIANCEID, SHIPID →

| allianceid integer | shipid integer |
|---|---|
| 2 | 1 |
| 0 | 2 |
| 0 | 3 |
| 2 | 4 |
| 3 | 5 |

Alliyah Taylor

# BELONGSTO: THIS TABLE KEEPS TRACK OF WHAT ALLIANCE A BEING IS AFFILIATED WITH.

```
CREATE TABLE AFFILIATEDWITH(
    ALLIANCEID          INTEGER NOT NULL REFERENCES ALLIANCES(ALLIANCEID),
    BEINGID             INTEGER NOT NULL REFERENCES BEINGS(BEINGID),
    PRIMARY KEY (ALLIANCEID, BEINGID)
);
```

## FUNCTIONAL DEPENDENCIES:

ALLIANCEID, BEINGID →

| allianceid integer | beingid integer |
|---|---|
| 1 | 1 |
| 1 | 2 |
| 1 | 3 |
| 1 | 4 |
| 0 | 5 |
| 0 | 6 |
| 0 | 7 |
| 4 | 8 |
| 0 | 9 |
| 5 | 10 |
| 5 | 11 |
| 0 | 12 |
| 0 | 13 |
| 0 | 14 |
| 2 | 15 |
| 2 | 16 |
| 2 | 17 |
| 3 | 18 |
| 0 | 19 |
| 3 | 20 |

Alliyah Taylor

# LOCATIONS: THIS TABLE CONTAINS ALL SYSTEMS AND QUADRANTS THAT COMPRISE FEDERATION TERRITORY.

```
CREATE TABLE LOCATIONS(

        LOCID       INTEGER NOT NULL,

        QUADRANT    TEXT NOT NULL,

        SYSTEM      TEXT NOT NULL,

    PRIMARY KEY (LOCID)

);
```

| locid integer | quadrant text | system text |
|---|---|---|
| 1 | Alpha | Tarsas |
| 2 | Alpha | Neutral Zone |
| 3 | Alpha | Xendi Sabu |
| 4 | Alpha | Beta Cassius |

## FUNCTIONAL DEPENDENCIES:

### LOCID → QUADRANT, SYSTEM

# LOCATIONOF: THIS TABLE CONNECTS SHIPS TO THE LOCATION THAT THEY ARE IN.

```
CREATE TABLE LOCATIONOF(
    LOCID              INTEGER NOT NULL REFERENCES LOCATIONS(LOCID),
    SHIPID             INTEGER NOT NULL REFERENCES SHIPS(SHIPID),
    PRIMARY KEY (LOCID, SHIPID)
);
```

| locid integer | shipid integer |
|---|---|
| 3 | 1 |
| 2 | 2 |
| 1 | 3 |
| 4 | 4 |
| 2 | 5 |

## FUNCTIONAL DEPENDENCIES:

### LOCID, SHIPID →

Alliyah Taylor

# CARGO: THIS TABLE CONTAINS ALL OF A SHIP'S CARGO AND BASIC INFORMATION ABOUT IT.

```
CREATE TABLE CARGO(
        CARGOID                 INTEGER NOT NULL,
        CARGOTYPE               TEXT NOT NULL,
        QTYINUNITS              INTEGER NOT NULL,
        ORIGIN                  TEXT NOT NULL,
        DESTINATION             TEXT NOT NULL,
        PERUNITCREDVAL          INTEGER,
    PRIMARY KEY (CARGOID)
);
```

| cargoid integer | cargotype text | qtyinunits integer | origin text | destination text | perunitcredval integer |
|---|---|---|---|---|---|
| 1 | Dilithium Ore | 40 | Dozaria | Ferenginar | 400 |
| 2 | Romulan Ale | 200 | Romulus | Coridan | 50 |

## FUNCTIONAL DEPENDENCIES:

CARGOID→ CARGOTYPE, QTYINUNITS, ORIGIN, DESTINATION, PERUNITCREDVAL

Alliyah Taylor

# CARGOON: THIS TABLE PAIRS CARGO WITH THE SHIP IT IS ON.

```
CREATE TABLE CARGOON(

        CARGOID                 INTEGER NOT NULL REFERENCES CARGO(CARGOID),

        SHIPID                  INTEGER NOT NULL REFERENCES SHIPS(SHIPID),

        PRIMARY KEY (CARGOID, SHIPID)

);
```

## FUNCTIONAL DEPENDENCIES:

### CARGOID, SHIPID →

| cargoid integer | shipid integer |
|---|---|
| 1 | 1 |
| 2 | 1 |

# SHIPCLASS: THIS TABLE CONTAINS ALL SHIP CLASSES AND THE TYPE OF SHIP THE CLASS IS USED FOR.

```
CREATE TABLE SHIPCLASS(
        CLASSID             INTEGER NOT NULL ,
        CLASSNAME           TEXT NOT NULL,
        CLASSTYPE           TEXT NOT NULL,
        PRIMARY KEY (CLASSID)
);
```

| classid integer | classname text | classtype text |
|---|---|---|
| 1 | Galaxy | Explorer |
| 2 | Niagara | Fast Cruiser |
| 3 | DKora | Attack Vessel |
| 4 | Dderidex | Warbird |
| 5 | Plague Ship | Transport |

## FUNCTIONAL DEPENDENCIES:

CLASSID→ CLASSNAME, CLASSTYPE

# SYSTEMS: THIS TABLE CONTAINS ALL SYSTEMS ABOARD A STARSHIP AND WHAT THEIR FUNCTIONS ARE.

```
CREATE TABLE SYSTEMS(
        SYSID            INTEGER NOT NULL,
        SYSNAME          TEXT NOT NULL,
        SYSFUNCTION      TEXT NOT NULL,
    PRIMARY KEY (SYSID)
);
```

| sysid integer | sysname text | sysfunction text |
|---|---|---|
| 1 | Warp Drive | Propulsion |
| 2 | Impulse Engines | Propulsion |
| 3 | Sensor Array | Observation and Research |
| 4 | Phaser Banks | Tactical |
| 5 | Torpedo Launchers | Tactical |
| 6 | Deflector Shields | Defense |
| 7 | Tractor Beam | Transport |
| 8 | Directed Energy Weapons | Tactical |
| 9 | Missile Launchers | Tactical |
| 10 | Electromagnetic Pulse Weapon | Tactical |
| 11 | Disruptor Array | Tactical |
| 12 | Cloaking Device | Defense |
| 13 | Holodeck | Leisure |
| 14 | Environmental Control System | Life Support |
| 15 | Transporter | Transport |

## FUNCTIONAL DEPENDENCIES:

SYSID → SYSNAME, SYSFUNCTION

Alliyah Taylor

# SYSTEMSON: THIS TABLE SHOWS WHICH SYSTEMS ARE USED ON WHICH SHIP CLASSES.

```
CREATE TABLE SYSTEMSON(
        SYSID       INTEGER NOT NULL REFERENCES SYSTEMS(SYSID),
        CLASSID     INTEGER NOT NULL REFERENCES SHIPCLASS(CLASSID),
        PRIMARY KEY (SYSID, CLASSID)
);
```

## FUNCTIONAL DEPENDENCIES:

### SYSID, CLASSID→

| sysid integer | classid integer |
|---|---|
| 1 | 1 |
| 2 | 1 |
| 3 | 1 |
| 4 | 1 |
| 5 | 1 |
| 6 | 1 |
| 7 | 1 |
| 13 | 1 |
| 14 | 1 |
| 15 | 1 |

Alliyah Taylor

# CLASSOF: THIS TABLE SHOWS WHICH SHIPS BELONG TO WHICH CLASSES.

```
CREATE TABLE CLASSOF(
        SHIPID        INTEGER NOT NULL REFERENCES SHIPS(SHIPID),
        CLASSID       INTEGER NOT NULL REFERENCES SHIPCLASS(CLASSID),
        PRIMARY KEY (SHIPID, CLASSID)
);
```

## FUNCTIONAL DEPENDENCIES:

### SHIPID, CLASSID→

| shipid integer | classid integer |
|---|---|
| 2 | 1 |
| 3 | 2 |
| 1 | 3 |
| 5 | 4 |
| 4 | 5 |

# SHIPCARGO: LISTS THE SHIP NAME, WHATEVER NON-SENTIENT CARGO IT IS CARRYING, AND THE CARGO'S PER UNIT VALUE IN CREDITS.

CREATE OR REPLACE VIEW SHIPCARGO

AS

SELECT DISTINCT SH.NAME, C.CARGOTYPE, C.QTYINUNITS, C.PERUNITCREDVAL

FROM SHIPS SH, CARGO C, CARGOON CGO

WHERE SH.SHIPID=CGO.SHIPID AND C.CARGOID=CGO.CARGOID

ORDER BY SH.NAME, C.CARGOTYPE, C.QTYINUNITS, C.PERUNITCREDVAL;

| name<br>text | cargotype<br>text | qtyinunits<br>integer | perunitcr...<br>integer |
|---|---|---|---|
| FMS Chulebas | Dilithium ... | 40 | 400 |
| FMS Chulebas | Romulan ... | 200 | 50 |

Alliyah Taylor

61554 710516 2479 472037 485 21 25962 7180055 73 33 44 220 69 701050055 2 13920157 8370 61320 58461 4924 5005 24399000 5759 9 89080553 96071953 831274890 7433407 313 1 701
29053 675071 9192 810633 984 50 33294 1127826 84 82 65 451 87 187045918 7 30131629 5292 20002 32841 1408 1367 35070814 9070 8 96011565 59031530 652186697 8114032 782 3 602
87503 500298 9442 388697 346 68 89782 7607614 42 61 97 847 98 878853892 6 73996511 9311 71423 59901 6920 5541 43120494 6803 4 13727316 52535981 411346137 5317388 954 9 283
65740 431430 8305 716231 317 66 79947 8737686 19 36 99 570 15 715686443 7 45658975 8653 54570 97397 3690 1884 52076545 2426 6 49435201 74135744 138263466 3835686 641 8 22794 1583 19767875 75076 983
10888 254605 6939 599036 705 74 24148 6107847 72 55 30 250 87 582165187 3 31592575 6269 37404 51438 7270 4868 68745300 3339 3 98345379 78754481 563945657 8064984 624 6 10829 6902 12252329 70440 224
19741 397454 5430 282037 726 80 36176 5152738 79 51 25 341 28 559921496 8 27328805 9448 79608 12810 2304 7984 85089680 6442 5 92751037 76138823 521419162 6697335 213 5 67188 6937 60078702 39871 391

25

## SHIPSYSTEMS: LISTS THE SHIP NAME AND EVERY SYSTEM ABOARD THAT SHIP.

```
CREATE OR REPLACE VIEW SHIPSYSTEMS
AS
SELECT DISTINCT SH.NAME, SYS.SYSNAME, SYS.SYSFUNCTION
FROM SHIPS SH, SYSTEMS SYS, CLASSOF CO, SHIPCLASS CL, SYSTEMSON SO
WHERE SH.SHIPID = CO.SHIPID AND CL.CLASSID = CO.CLASSID AND SO.SYSID =
SYS.SYSID AND CL.CLASSID = SO.CLASSID
ORDER BY SH.NAME, SYS.SYSNAME, SYS.SYSFUNCTION;
```

| name text | sysname text | sysfuncti... text |
|---|---|---|
| FMS Chulebas | Deflector Shields | Defense |
| FMS Chulebas | Directed Energy Weapons | Tactical |
| FMS Chulebas | Electromagnetic Pulse Weapon | Tactical |
| FMS Chulebas | Environmental Control System | Life Supp... |
| FMS Chulebas | Impulse Engines | Propulsion |
| FMS Chulebas | Missile Launchers | Tactical |
| FMS Chulebas | Sensor Array | Observat... |
| FMS Chulebas | Tractor Beam | Transport |
| FMS Chulebas | Transporter | Transport |
| FMS Chulebas | Warp Drive | Propulsion |
| Halsa | Deflector Shields | Defense |
| Halsa | Disruptor Array | Tactical |
| Halsa | Environmental Control System | Life Supp... |
| Halsa | Impulse Engines | Propulsion |
| Halsa | Sensor Array | Observat... |
| Halsa | Tractor Beam | Transport |
| Halsa | Transporter | Transport |
| Halsa | Warp Drive | Propulsion |

Alliyah Taylor

# REPORTS AND THEIR QUERIES

## QUERY TO VIEW THE TOTAL VALUE OF A SHIP'S CARGO

```
SELECT CREDVAL.NAME, SUM(TOTALCREDVAL) AS TOTALCARGOVAL

FROM (

SELECT NAME ,CARGOTYPE, QTYINUNITS*PERUNITCREDVAL AS TOTALCREDVAL

FROM SHIPCARGO

GROUP BY NAME, CARGOTYPE, QTYINUNITS,PERUNITCREDVAL) AS CREDVAL

WHERE CREDVAL.NAME =CREDVAL.NAME

GROUP BY CREDVAL.NAME
```

| name text | totalcarg... bigint |
|-----------|---------------------|
| FMS Chulebas | 26000 |

Alliyah Taylor

# STORED PROCEDURE

PROCEDURE TO SEE WHAT CLASSES A SYSTEM IS USED IN.

```sql
CREATE OR REPLACE FUNCTION ISINCLASS(INT, REFCURSOR) RETURNS REFCURSOR AS

$$
DECLARE
SYS_ID      INT           :=$1;
RESULTSET   REFCURSOR     := $2;
            BEGIN
            OPEN RESULTSET FOR
                    SELECT *
                    FROM SHIPCLASS
                    WHERE CLASSID IN(
                    SELECT SO.CLASSID
                    FROM SYSTEMSON SO
                    WHERE SO.SYSID = SYS_ID) ;
            RETURN RESULTSET;


                    END;
                    $$
            LANGUAGE PLPGSQL;
```

```sql
SELECT ISINCLASS(7, 'RESULTS');
FETCH ALL FROM RESULTS;
```

| classid integer | classname text | classtype text |
|---|---|---|
| 1 | Galaxy | Explorer |
| 2 | Niagara | Fast Cruiser |
| 3 | DKora | Attack Vessel |
| 4 | Dderidex | Warbird |
| 5 | Plague Ship | Transport |

Alliyah Taylor

ADMINISTRATOR ROLE: REPRESENTS THE ADMINISTRATOR WHO HAS FULL ACCESS.

CREATE ROLE ADMIN;

GRANT ALL ON ALL TABLES IN SCHEMA PUBLIC TO ADMIN;

ADMIRAL ROLE: REPRESENTS THE USERS WITH ADMIRAL-LEVEL CLEARANCE WHO NEED TO ACCESS AND CHANGE RECORDS WHEN RECEIVING UPDATED INFORMATION FROM STARSHIPS.

CREATE ROLE ADMIRAL;

REVOKE ALL ON ALL TABLES IN SCHEMA PUBLIC FROM ADMIRALS;

GRANT SELECT ON ALL TABLES IN SCHEMA PUBLIC TO ADMIRALS;

GRANT INSERT ON BEINGS, PRISONERS, CREW, PASSENGERS, CARGO, BEINGSON, CARGOON, AFFILIATEDWITH, LOCATIONOF
TO ADMIRAL;

CREATE ROLE CAPTAIN;

REVOKE ALL ON ALL TABLES IN SCHEMA PUBLIC FROM CAPTAIN;

GRANT SELECT ON BEINGS, PRISONERS, CREW, PASSENGERS, BEINGSON, ALLIANCES, AFFILIATEDWITH, BELONGSTO, CARGO,
CARGOON, LOCATIONS, LOCATION OF, SYSTEMS, SYSTEMSON, SHIPCLASS,CLASSOF
TO CAPTAIN;

Alliyah Taylor

# IMPLEMENTATION NOTES/KNOWN PROBLEMS

- POSGRES DOES NOT YET SUPPORT STARDATES. CURRENTLY, STAR DATES ARE STORED AS NUMERIC VALUES WITH A SINGLE DECIMAL POINT.

- THE DATABASE CURRENLY COUNTS SENTIENT CARGO AS PRISONERS. THIS IS NOT REPRESENTATIVE OF EVERY CASE AND MUST BE AMENDED IN THE FUTURE. (SOME SPECIES KEEP FARMED HOSTS AS CARGO, BUT THEY ARE NOT HELD AGAINST THEIR WILL AND THUS ARE NOT PRISONERS.)

- THE CURRENT IMPLEMENTATION DOES NOT KEEP TRACK OF WHEN SHIPS LEAVE FEDERATION SPACE, ONLY WHEN THEY ENTER IT. THUS, A FUTURE VERSION SHOULD IMPLEMENT A WAY OF KEEPING TRACK WHEN AND WHERE A SHIP LEFT FEDERATION SPACE.

- THERE ARE MANY MORE ATTRIBUTES THAT SHOULD BE ADDED TO A FUTURE IMPLEMENTATION OF THIS DATABASE SUCH AS A "STATUS" FIELD FOR BEINGS. THIS WOULD ALLOW STARFLEET TO BETTER UNDERSTAND WHAT TACTICS NEED TO BE IMPLEMENTED.

Alliyah Taylor