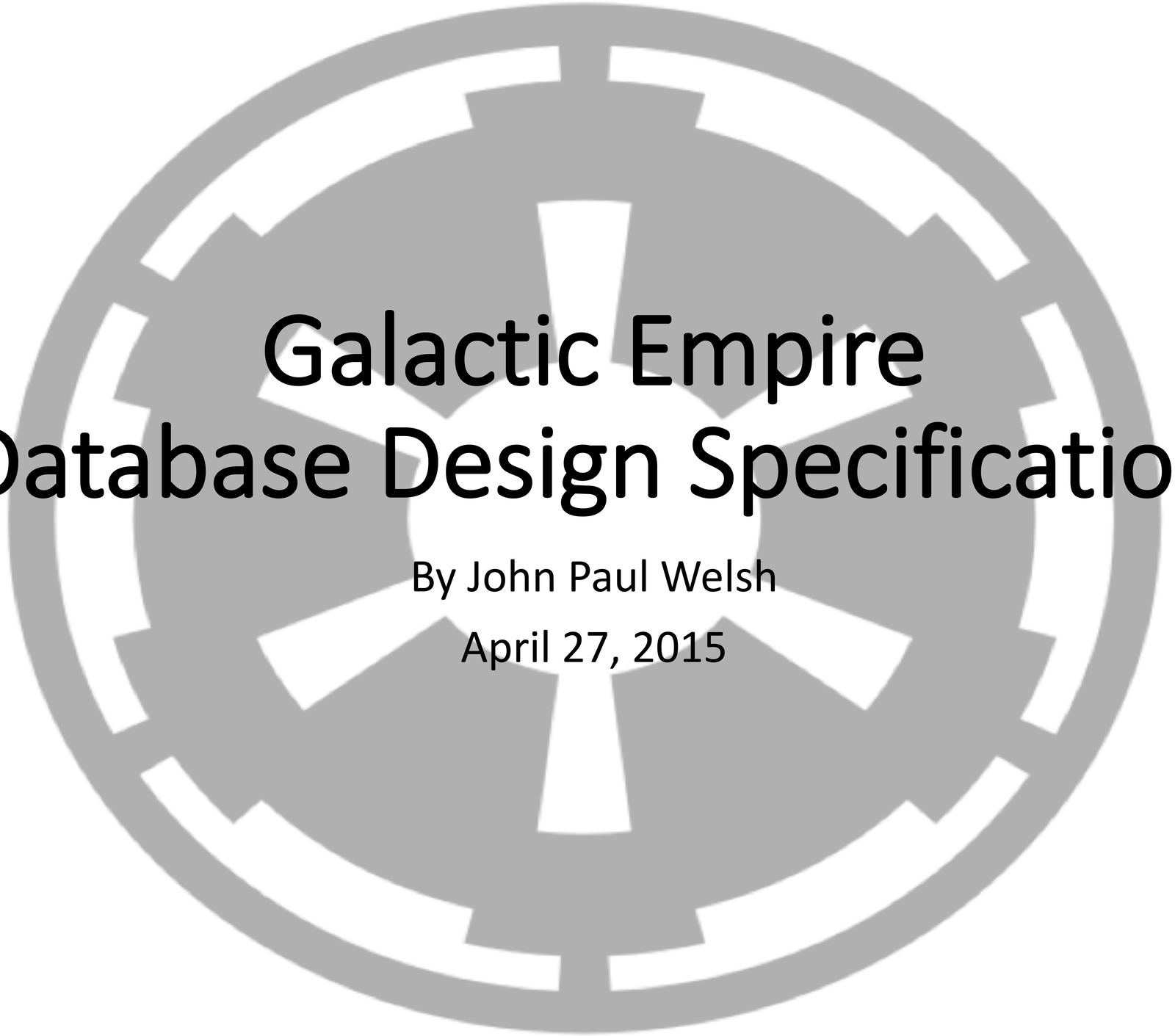
A long time ago in a galaxy far, far away....



Galactic Empire
Database Design Specification

By John Paul Welsh

April 27, 2015

Executive Summary

The Galactic Empire is the most powerful organization in the galaxy. Therefore, it deserves a well-designed database to keep track of the seemingly never-ending stream of data moving through the universe. Like all effective organizations, the Empire needs to keep careful tabs on every place, person, or group who crosses its path, for good or ill. Especially for ill.

This specification contains an overview of all the most important aspects of the database, and will demonstrate its excellence in terms of structure, security protection, and flexibility. This will all but ensure the Galactic Empire's position as the more formidable and efficient organization in the galaxy....

ER Diagram

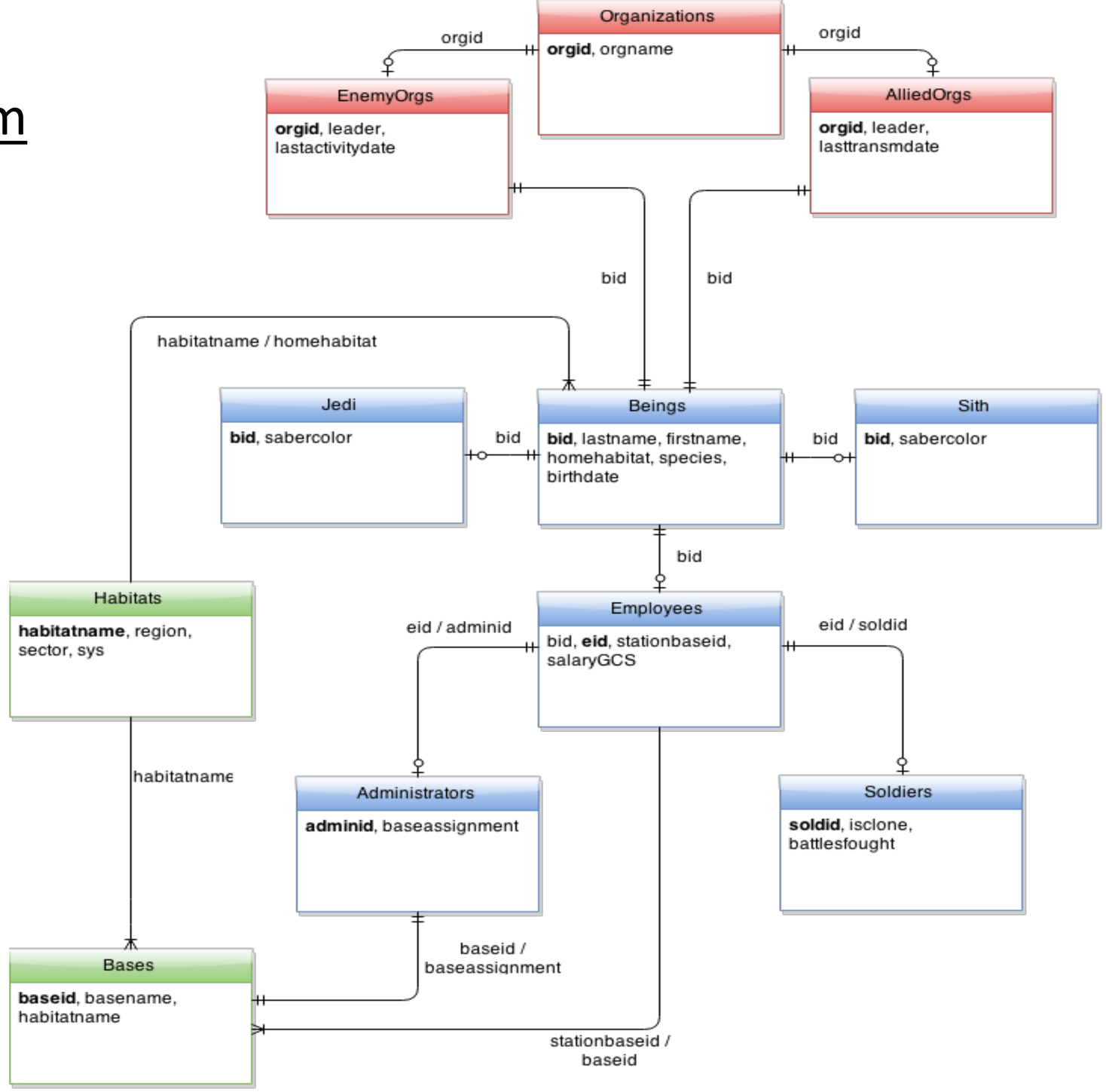

Organizations
orgid, orgname
orgid
orgid
EnemyOrgs
orgid, leader,
lastactivitydate
AlliedOrgs
orgid, leader,
lasttransmdate
bid
bid
habitatname / homehabitat
Jedi
bid, sabercolor
bid
Beings
bid, lastname, firstname,
homehabitat, species,
birthdate
bid
Sith
bid, sabercolor
bid
Habitats
habitatname, region,
sector, sys
Employees
bid, eid, stationbaseid,
salaryGCS
eid / adminid
eid / soldid
habitatname
Administrators
adminid, baseassignment
Soldiers
soldid, isclone,
battlesfought
baseid /
baseassignment
Bases
baseid, basename,
habitatname
stationbaseid /
baseid

Table: Habitats

Habitats include planets, moons, and any other domains that can sustain life.

```
CREATE TABLE habitats (
  habitatname text not null,
  region      text,
  sector      text,
  sys         text,
  primary key(habitatname)
);
```

Functional Dependencies

habitatname → region, sector, sys

	habitatname text	region text	sector text	sys text
1	Alderaan	Core Worlds	Alderaan	Alderaan
2	Tatooine	Outer Rim Territories	Arkanis	Tatoo
3	Yavin 4	Outer Rim Territories	Gordian Reach	Yavin
4	Kashyyyk	Mytaranor	Kashyyyk	Kashyyyk
5	Eriadu	Outer Rim Territories	Seswanna	Eriadu
6	Naboo	Mid Rim	Chommell	Naboo
7	Geonosis	Outer Rim Territories	Arkanis	Geonosis

Table: Beings

Beings include any sentient living form.

```
CREATE TABLE beings (
  bid          char(4) not null,
  lastname     text,
  firstname    text,
  homehabitat  text references
                 habitats(habitatname),
  species      text,
  birthdate    date,
  primary key(bid)
);
```

Functional Dependencies
bid → lastname, firstname, homehabitat, species, birthdate

	bid character(4)	lastname text	firstname text	homehabitat text	species text	birthdate date
1	x001	Skywalker	Luke	Tatooine	Human	1965-06-13
2	x002	Skywalker	Anakin	Tatooine	Human	1941-02-02
3	x003	<NULL>	Chewbacca	Kashyyyk	Wookie	1950-04-18
4	x004	Tarkin	Wilhuff	Eriadu	Human	1920-10-01
5	x005	Organa	Leia	Alderaan	Human	1965-06-13
6	x006	Palpatine	<NULL>	Naboo	Human	1919-12-01
7	x007	Jacksonista	Randy	Alderaan	Human	1938-11-04
8	x008	<NULL>	Jabba	Tatooine	Hutt	1968-08-08
9	y001	<NULL>	<NULL>	Geonosis	Human	1940-04-01
10	y002	<NULL>	<NULL>	Geonosis	Human	1940-04-01
11	y003	<NULL>	<NULL>	Geonosis	Human	1940-04-01
12	y004	Phosphor	Pilth	Alderaan	Human	1938-09-01

Table: Jedi

```
CREATE TABLE jedi (
  bid        char(4) not null references beings(bid),
  sabercolor text,
  primary key(bid)
);
```

Functional Dependencies

bid → sabercolor

	bid character(4)	sabercolor text
1	x001	green

Table: Sith

```
CREATE TABLE sith (
  bid        char(4) not null references beings(bid),
  sabercolor text,
  primary key(bid)
);
```

Functional Dependencies

bid → sabercolor

	bid character(4)	sabercolor text
1	x002	red
2	x006	red

Table: Bases

Bases are imperial military or administrative installations, that may or may not be part of an existing habitat.

```
CREATE TABLE bases (
  baseid      char(5) not null,
  basename    text,
  habitatname text,
  primary key(baseid)
);
```

Functional Dependencies

baseid → basename, habitatname

	baseid character(5)	basename text	habitatname text
1	b0001	Death Star	<NULL>
2	b0002	Fondor II	Eriadu
3	b0003	Tremor Base	Eriadu
4	b0004	Kashyyyk Imperial Base	Kashyyyk
5	b0005	Alpha Outpost	Tatooine

Table: Employees

An entity subtype of beings, so this table only contains information specific to employees of the Galactic Empire.

```
CREATE TABLE employees (
  bid           char(4) not null references beings(bid),
  eid           char(4),
  stationbaseid char(5) references bases(baseid),
  salaryGCS     numeric(10,2),
  primary key(eid)
);
```

Functional Dependencies

bid → eid, stationbaseid, salaryGCS

	bid character(4)	eid character(4)	stationbaseid character(5)	salarygcs numeric(10,2)
1	x002	e001	b0001	100000.00
2	x004	e002	b0001	90000.00
3	x006	e003	b0001	150000.00
4	y001	e004	b0004	5000.00
5	y002	e005	b0004	5000.00
6	y003	e006	b0005	5000.00
7	y004	e007	b0002	10000.00

Table: Administrators

Administrative employees of the Empire.

```
CREATE TABLE administrators (
  adminid        char(4) not null references employees(eid),
  baseassignment char(5) references bases(baseid),
  primary key(adminid)
);
```

Functional Dependencies

adminid → baseassignment

	adminid character(4)	baseassignment character(5)
1	e001	b0001
2	e002	b0003

Table: Soldiers

Soldiers who work for the Galactic Empire.

```
CREATE TABLE soldiers (
  soldid        char(4) not null references employees(eid),
  isclone       boolean,
  battlesfought int,
  primary key(soldid)
);
```

Functional Dependencies

soldid → isclone, battlesfought

	soldid character(4)	isclone boolean	battlesfought integer
1	e004	t	30
2	e005	t	15
3	e006	t	24
4	e007	f	8

Table: Organizations

Any kind of acknowledged group, such as a company, governmental body, or association.

```
CREATE TABLE organizations (
  orgid   char(3) not null,
  orgname text,
  primary key(orgid)
);
```

Functional Dependencies

orgid → orgname

	orgid character(3)	orgname text
1	o01	Raider Allegiance
2	o02	Senatorial Committee
3	o03	Fleet General Academy
4	o04	Rebel Alliance
5	o05	Hutt Gang

Table: AlliedOrgs

Organizations that hold allegiance with the Galactic Empire.

```
CREATE TABLE alliedOrgs (
  orgid          char(3) not null references organizations(orgid),
  leader         char(4) references beings(bid),
  lasttransmdate date,
  primary key(orgid)
);
```

Functional Dependencies

orgid → leader, lasttrasmdate

	orgid character(3)	leader character(4)	lasttransmdate date
1	o05	x008	2012-10-06
2	o03	x007	2015-01-01

Table: EnemyOrgs

Organizations that are enemies to the Galactic Empire.

```
CREATE TABLE enemyOrgs (
  orgid            char(3) not null references organizations(orgid),
  leader           char(4) references beings(bid),
  lastactivitydate date,
  primary key(orgid)
);
```

Functional Dependencies

orgid → leader, lastactivitydate

	orgid character(3)	leader character(4)	lastactivitydate date
1	o04	x001	2015-02-03
2	o02	x005	2013-12-21

View: CloneSoldiers

Empire soldiers who are clones of Jango Fett.

```
CREATE VIEW cloneSoldiers(soldid, battlesfought) AS
  select s.soldid,
         s.battlesfought
    from soldiers s
   where s.isclone = true;
```

	soldid character(4)	battlesfought integer
1	e004	30
2	e005	15
3	e006	24

View: NonEmployees

Beings who are not employees of the Galactic Empire.

```
CREATE VIEW nonEmployees(bid, lastname, firstname, homehabitat, birthdate) AS
  select b.bid,
         b.lastname,
         b.firstname,
         b.homehabitat,
         b.birthdate
    from beings b
   where b.bid not in (select e.bid
                         from employees e);
```

	bid character(4)	lastname text	firstname text	homehabitat text	birthdate date
1	x001	Skywalker	Luke	Tatooine	1965-06-13
2	x003	<NULL>	Chewbacca	Kashyyyk	1950-04-18
3	x005	Organa	Leia	Alderaan	1965-06-13
4	x007	Jacksonista	Randy	Alderaan	1938-11-04
5	x008	<NULL>	Jabba	Tatooine	1968-08-08

View: HumanBeings

Beings who are of the human species.

```
CREATE VIEW humanBeings(bid, lastname, firstname, homehabitat, birthdate) AS
  select b.bid,
         b.lastname,
         b.firstname,
         b.homehabitat,
         b.birthdate
    from beings b
   where b.species = 'Human';
```

	bid character(4)	lastname text	firstname text	homehabitat text	birthdate date
1	x001	Skywalker	Luke	Tatooine	1965-06-13
2	x002	Skywalker	Anakin	Tatooine	1941-02-02
3	x004	Tarkin	Wilhuff	Eriadu	1920-10-01
4	x005	Organa	Leia	Alderaan	1965-06-13
5	x006	Palpatine	<NULL>	Naboo	1919-12-01
6	x007	Jacksonista	Randy	Alderaan	1938-11-04
7	y001	<NULL>	<NULL>	Geonosis	1940-04-01
8	y002	<NULL>	<NULL>	Geonosis	1940-04-01
9	y003	<NULL>	<NULL>	Geonosis	1940-04-01
10	y004	Phosphor	Pilth	Alderaan	1938-09-01

View: NeutralOrgs

Organizations who are not allied with or enemies to the Empire.

```
CREATE VIEW neutralOrgs(orgid, orgname) AS
  select o.orgid,
         o.orgname
    from organizations o
   where o.orgid not in (select ao.orgid
                           from alliedOrgs ao)
     and o.orgid not in (select en.orgid
                           from enemyOrgs en);
```

	orgid character(3)	orgname text
1	o01	Raider Allegiance

Report: High-Paid Employees

Employees whose salary of at least 100,000 Galactic Standard Credits.

```
select *
  from employees e
 where e.salaryGCS >= 100000.00;
```

	bid character(4)	eid character(4)	stationbaseid character(5)	salarygcs numeric(10,2)
1	x002	e001	b0001	100000.00
2	x006	e003	b0001	150000.00

Report: Beings from Tatooine

All beings whose home habitat is Tatooine.

```
select *
  from beings b
 where b.homehabitat = 'Tatooine';
```

	bid character(4)	lastname text	firstname text	homehabitat text	species text	birthdate date
1	x001	Skywalker	Luke	Tatooine	Human	1965-06-13
2	x002	Skywalker	Anakin	Tatooine	Human	1941-02-02
3	x008	<NULL>	Jabba	Tatooine	Hutt	1968-08-08

Report: Admins and Names of Non-Habitat Bases

The administrator's id of, and the name of, all bases that do not belong to a habitat.

```
select a.adminid,
       b.basename
  from bases b
       left outer join
       administrators a
    on b.baseid = a.baseassignment
 where b.habitatname is null;
```

	adminid character(4)	basename text
1	e001	Death Star

Report: Non-Employees from Sith Planet

The names and ids of all beings who are not employees, and whose home habitat is shared with a Sith lord.

```
select b.bid,
       b.lastname,
       b.firstname
  from beings b,
       nonEmployees ne
 where b.bid = ne.bid
   and b.homehabitat in (select b2.homehabitat
                           from beings b2,
                                sith s
                          where b2.bid = s.bid);
```

	bid character(4)	lastname text	firstname text
1	x001	Skywalker	Luke
2	x008	<NULL>	Jabba

Stored Procedure: AllBeingsFromHabitat

Returns info on all beings whose home habitat matches the one given in the paremeter.

```
CREATE OR REPLACE FUNCTION AllBeingsFromHabitat(text, refcursor) RETURNS refcursor AS
$$
DECLARE
  searchHabitat text      := $1;
  results       refcursor := $2;
BEGIN
  OPEN results FOR
    select *
      from beings b
     where b.homehabitat = searchHabitat;
  RETURN results;
END;
$$ LANGUAGE plpgsql;

select AllBeingsFromHabitat('Alderaan', 'results');
fetch all from results;
```

	bid character(4)	lastname text	firstname text	homehabitat text	species text	birthdate date
1	x005	Organa	Leia	Alderaan	Human	1965-06-13
2	x007	Jacksonista	Randy	Alderaan	Human	1938-11-04
3	y004	Phosphor	Pilth	Alderaan	Human	1938-09-01

Stored Procedure: EmployeesOlderThan

Returns info on all employees who are older than the given date.

```
CREATE OR REPLACE FUNCTION EmployeesOlderThan(date, refcursor) RETURNS refcursor AS
$$
DECLARE
  searchDate date      := $1;
  results    refcursor := $2;
BEGIN
  OPEN results FOR
    select e.eid
      from beings b,
           employees e
     where b.bid = e.bid
       and b.birthdate < searchDate;
  RETURN results;
END;
$$ LANGUAGE plpgsql;

select EmployeesOlderThan('1940-01-01', 'results');
fetch all from results;
```

	eid character(4)
1	e002
2	e003
3	e007

Trigger: IncrNonCloneSalaryOn50Battles

This trigger will automatically increase the salary of any non-clone imperial soldiers who reach 50 completed battles. It happens on updating the “soldiers” table.

```
CREATE OR REPLACE FUNCTION incrSalary() RETURNS trigger AS
$$
BEGIN
  IF (NEW.isclone = false) THEN
    IF (NEW.battlesfought > 49) THEN
      UPDATE employees
         SET salaryGCS = salaryGCS + 3000.00
       WHERE employees.eid = NEW.soldid;
    END IF;
  END IF;
  RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER incrNonCloneSalaryOn50Battles
AFTER UPDATE ON soldiers FOR EACH ROW EXECUTE PROCEDURE incrSalary();
```

	bid character(4)	eid character(4)	stationbaseid character(5)	salarygcs numeric(10,2)
1	x002	e001	b0001	100000.00
2	x004	e002	b0001	90000.00
3	x006	e003	b0001	150000.00
4	y001	e004	b0004	5000.00
5	y002	e005	b0004	5000.00
6	y003	e006	b0005	5000.00
7	y004	e007	b0002	13000.00

Security

There are only a few user groups necessary in the current version of this database: full admin rights, soldier rights, and imperial admin rights. These three groups are created and given permissions below. Only the dbadmin can delete anything from the database, for safety precautions, and impadmins are explicitly not given update access to the “administrator” table.

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

CREATE ROLE soldiers;
GRANT SELECT ON habitats, bases, soldiers, jedi, sith
TO soldiers;

CREATE ROLE impadmin;
GRANT SELECT ON ALL TABLES IN SCHEMA PUBLICTO impadmin;
GRANT INSERT, UPDATE ON soldiers, employees, jedi, sith, bases,
                        habitats, organizations, alliedOrgs, enemyOrgsTO impadmin;
```

Implementation Notes – Known Problems – Future Enhancements

Although this specification is a bit shabby in terms of example data, it sufficiently demonstrates the ability that the Empire has to determine relationships between entities. The political and military benefits of a properly-populated database would make this design indispensable to the Empire.

There are a variety of flaws in the design of this database that would be ironed out with the passage of time. For instance, the database has no sense of history, so Alderaan is still considered a very much intact habitat. There is also a potentially BCNF-breaking construct between employee ids and being ids, since they both uniquely identify non-key attributes in the employees table. Finally, there are plenty of other kinds of data that have been left out of this simple mockup, including imperial spies, mechanics, and engineers, as well as resources such as weapons and raw materials.