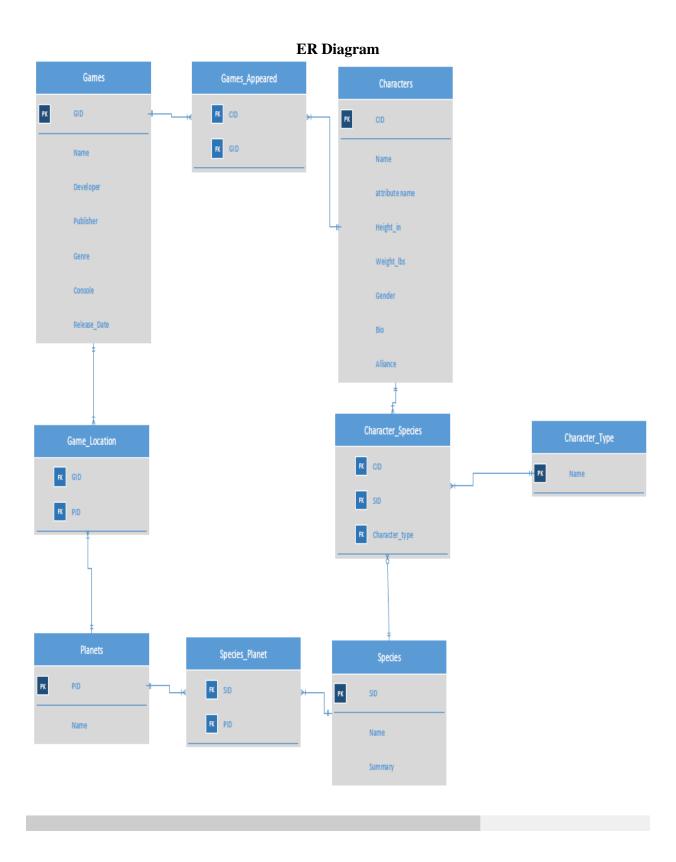


1 | Design Project Halo Database

Table of Contents

ER-Diagram	3
Summary	4
Overview	4
Entities	5
Characters	5
Games	6
Species	6
Planets	
Character Type	8
Game Location	8
Games Appeared	9
Species Planet	
Character Species	
Views	12
Spartans	12
Marines	
A.I	13
Stored Procedures	14
SP: All games with a given character	14
SP: All Characters within a given race	
Sample Reports	15
Query: What Species Is Master Chief?	15
Query: What Planet Are The Sangheili from?	15
Query: What are the games in which Master Chief has appeared in?	
Security	
Known Issues	16
Future Enhancements	17



Summary

This document serves the purpose of revealing the design and implementation of a database made for the Halo Universe. This will be a central repository for all information related to the Halo franchise. We'll start by showing an ER Diagram that will allow us to portray a brief description of our proposed solution for the Data Collection needs of Microsoft. We have also included is the purpose of each entity within the database as portrayed in the ER Diagram. The details will include the purpose of the table as well as the reasoning behind each entity and its design. Along with the detailed explanation we also provide the SOL statements that the Entities in the database were created with and alongside that sample results are also included. This will allow for a visual interpretation of the end result of our work and how it will provide a well-functioning database. In order to drive simplicity we also have included views and stored procedures that allow the user to implement complex queries in simple single line statements. These functions can be implemented by those in web design, application development, or project development within the Microsoft Corporation without having to do the difficult work of formulating complex queries. Just as we did with the Entities we also provide sample data for the user to visualize what exactly we are discussing. We'll then show a collection of sample queries and close with a small discussion on security, some of the tradeoffs of our design and some enhancements that could be made in the future.

Overview

Based on the requirements outlined by Microsoft Corporation we believe our solution is suitable to their needs. Our goals for designing the database were based on the following observations.

- Once the initial database is made operational and the base data has been inserted, the database will require minimal effort in upkeep and adding new information to it while it is operational.
- Additions to the database will most commonly be inserted at the release of a new game, every few years or so after a game comes out of development. The database will be read only for the most part since insertion into the database won't usually be a common occurrence. This allowed us to make our design as easy as possible to use being only used to retrieve data in a majority of its uses allows us to focus our time in creating stored procedures that the user can just call. Because of this we try to allow the user to insert data simply and without altering the structure of the database.
- Since the popularity of the Halo universe is monumental and is the face of the Xbox console we wanted to also allow access to fan sites and any one curious about the universe. The number of fans and fan sites increase with every new game. Our database solution provides a public API to allow fan sites access to game data while keeping Microsoft's systems secure. Our database provides some stored-procedures that allow those that need to the ability to query the data using simple function calls. We also utilize views to increase usability and security.

Based on these observations we are confident that our database will meet your business requirements by providing you with a system that is easy to use, accessible, and secure.

Entities

Characters

The characters entity stores data on any sentient or artificial life form the plays a part in the Halo Universe. This includes supporting and non-playable characters from each game. This also serves as a log of the attributes of character and helps to show the main attributes of characters. By logging this we are more easily able to decipher the differences between similar characters. Attributes like hair and eye color are left out due to a good percentage of the characters not being human and having identifiable characteristics. Also added some things that are abnormal like being able to select genderless and players choice as a gender were needed since the characters are a part of a game and are intractable. It was also thought to let the bio be a field that allows nulls due to some of the characters being secondary (Not important enough for a bio entry) but still important. The field alliance reveals the faction that the character is dedicated to this helps to grasp basic insight on the characteristics of the individual.

--This is the creation of the Characters table

```
create table characters (
       character id serial not null primary key,
       name
                     text not null,
                           not null default 0.0 check (height in > 0),
       height_in
                     int
                           not null default 0.0 check (weight_lbs > 0),
       weight lbs
                     int
                           not null check (gender = 'Male' or gender = 'Female' or gender =
       gender
                     text
       'Genderless' or gender = 'Players Choice'),
       bio
                     text.
                           not null check (alliance = 'UNSC' or alliance = 'Covenant' or alliance =
       alliance
                     text
       'Flood' or alliance = 'Forerunner') not null
);
```

Functional Dependencies: Character_id → Name, Height_in, Weight_lbs, Gender, Bio, Alliance

	character_id integer	name text	height_in integer	weight_lbs integer	gender text	bio text	alliance text
1	1	Master Chief John-117	84	1200	Male	A Short Bio About This Character	UNSC
2	2	Cortana	7	1	Female	A Short Bio About This Character	UNSC
3	3	Sergeant Major Avery Junior Johnson	74	185	Male	A Short Bio About This Character	UNSC
4	4	Captain Jacob Keyes	76	212	Male	A Short Bio About This Character	UNSC
5	5	Miranda Keyes	68	135	Female	A Short Bio About This Character	UNSC
6	6	Doctor Catherine Elizabeth Halsey	67	120	Female	A Short Bio About This Character	UNSC
7	7	Commander Carter-A259 (Noble One)	82	950	Male	A Short Bio About This Character	UNSC
8	8	Lieutenant Commander Catherine-B320 (Noble Two)	81	850	Female	A Short Bio About This Character	UNSC
9	9	Emile-A239 (Noble Three)	82	1000	Male	A Short Bio About This Character	UNSC
10	10	Chief Warrent Officer Jorge-052 (Noble Four)	96	1500	Male	A Short Bio About This Character	UNSC
11	11	Jun-A266 (Noble Five)	83	945	Male	A Short Bio About This Character	UNSC
12	12	Spartan-B312 (Noble Six)	81	1200	Players Choice	A Short Bio About This Character	UNSC
13	13	High Prophet of Truth	87	560	Male	A Short Bio About This Character	Covenant
14	14	High Prophet of Mercy	87	560	Male	A Short Bio About This Character	Covenant
15	15	High Prophet of Regret	86	560	Male	A Short Bio About This Character	Covenant
16	16	Arbiter (Thel Vadamee)	94	830	Male	A Short Bio About This Character	Covenant
17	17	Tartarus	108	1100	Male	A Short Bio About This Character	Covenant
18	18	343 Guilty Spark (The Monitor of Installation 04)	20	25	Male	A Short Bio About This Character	Forerunner
19	19	Didact (Ur-Didact or Shadow of Sundered Star)	136	795	Male	A Short Bio About This Character	Forerunner
20	20	Librarian (First Light Weaves Living Song)	114	441	Female	A Short Bio About This Character	Forerunner
21	21	Gravemind (Inferi Sententia)	3360	52000	Genderless	A Short Bio About This Character	Flood

Games

The games entity is responsible for storing all of the data about the games in the Halo universe. The data stored here shows all who were involved in the making of the game and allows us to get a feel for how the games came to be and why they turned out the way they did. Including fields like platform released and genre allow insight into where the games series is headed in terms of consoles. Same with the developer field to determine where the story was focused and where it might be headed.

```
--This is the creation of the Games table
```

```
create table games(
       game_id
                         serial not null primary key,
                          text not null,
       name
       developer
                          text not null,
       publisher
                          text not null,
       genre
                          text not null,
       platform_released text not null check (platform_released = 'Xbox' or platform_released =
       'Xbox 360' or platform_released = 'Windows Phone 8'),
                          date not null
       release_date
);
```

Functional Dependencies: Game_id → Name, Developer, Publisher, Genre, Platform_Released, Release_Date

	game_id integer	name text	developer text	publisher text		genre text	platform_released text	release_date date
1	1	Halo: Combat Evolved	Bungie Studios	Microsoft Game	Studio	First Person Shooter	Xbox	2001-11-15
2	2	Halo 2	Bungie Studios	Microsoft Game	Studio	First Person Shooter	Xbox	2004-11-09
3	3	Halo 3	Bungie Studios	Microsoft Game	Studio	First Person Shooter	Xbox 360	2007-09-25
4	4	Halo Wars	Ensemble Studios	Microsoft Game	Studio	Real Time Strategy	Xbox 360	2009-03-03
5	5	Halo 3: ODST	Bungie Studios	Microsoft Game	Studio	First Person Shooter	Xbox 360	2009-09-22
6	6	Halo: Reach	Bungie Studios	Microsoft Game	Studio	First Person Shooter	Xbox 360	2010-09-14
7	7	Halo: Combat Evolved An	niversary 343 Industries	Microsoft Game	Studio	First Person Shooter	Xbox 360	2011-11-15
8	8	Halo 4	343 Industries	Microsoft Game	Studio	First Person Shooter	Xbox 360	2012-11-06
9	9	Halo: Spartan Assault	343 Industries	Microsoft Game	Studio	Top Down Shooter	Windows Phone 8	2013-07-18

Species

The species entity in the table allows us to see all of the type of life involved in the creation of the story in game. It was decided that to keep things as simplistic as possible that all that would be inserted would be the name and a short summary on the race. A future enhancement here could be to allow a nickname field of some sort. This is because in the game none of the characters are referred to as the specie type because the humans have renamed them all and those are the names heard when referring to a race of sentient life.

Functional Dependencies: Species_id → Name, Summary

	species_id integer	name text		ımmary ext	•			
1	1	Human	A	Short	Summary	About	The	Species
2	2	Sangheili	A	Short	Summary	About	The	Species
3	3	Jiralhanae	A	Short	Summary	About	The	Species
4	4	Kig-Yar	A	Short	Summary	About	The	Species
5	5	Unggoy	A	Short	Summary	About	The	Species
6	6	Mgalekgolo	A	Short	Summary	About	The	Species
7	7	San Shyuum	A	Short	Summary	About	The	Species
8	8	Yanmee	A	Short	Summary	About	The	Species
9	9	Forerunner	A	Short	Summary	About	The	Species
10	10	Flood	A	Short	Summary	About	The	Species
11	11	Precursor	A	Short	Summary	About	The	Species
12	12	Huragok	A	Short	Summary	About	The	Species

Planets

The Planets entity shows only the major locations that the games have taken place in. This is its own entity because it is referenced in multiple other tables. Only a name is required for this table.

```
--This is the creation of the Planets table
create table planets
(
planet_id serial not null primary key,
name text not null
);
```

Functional Dependencies: $Planet_id \rightarrow Name$

	planet_id integer	name text
1	1	Installation 00
2	2	Installation 03
3	3	Installation 04
4	4	Installation 05
5	5	Requiem
6	6	Earth
7	7	Reach
8	8	Chuot
9	9	Decided Heart
10	10	Joyous Exultation
11	11	Metisette
12	12	UNSC Infinity

Character Type

Character Type serves the purpose of assigning subtypes to the alliances column. This serves a purpose in the case of specifying specifically what a person is responsible for within the alliance. This also makes it a little easier for the common user to navigate because the names used or the more common names for the races within the game.

```
--This is the creation of the Character Type table create table character_type (
name text not null
);
```

	name text
1	Spartan
2	Elite
3	Brute
4	Marine
5	Promethian
6	Flood
7	A.I.
8	High Prophet

Game Location

Game location serves as the connecting link between the Games table and the Locations table. This table allows us to see the various locations in which a game has taken place. Having this as a separate entity allows for the entry of multiple locations to be in a single game. This also allows the user to input referenced locations from the games too. This table is reliant on the games and planets table for references to data to reduce the risk of update errors. This table is useful for queries such as "Where did Halo: Combat Evolved take place?"

```
--This is the creation of the Game Location table
create table game_location
(
game_id serial not null references games(game_id),
planet_id serial not null references planets(planet_id)
);
```

	game_id integer	planet_id integer
1	1	3
2	2	4
3	3	1
4	4	6
5	5	6
6	6	7
7	7	3
8	8	5
9	9	12

Games Appeared

Games Appeared serves as a link between the Games and Characters tables. This is required to show what characters appear in what particular games. This table is very important because it allows the user to keep track of what games a character has appeared in. This table would come in handy if a query would ask "What are the games in which Master Chief has appeared in?" It's important to note that this table helps ensure that only valid Characters and Games are entered. You can only enter an id from the Games Table or the Characters Table.

```
--This is the creation of the Games Appeared table
create table games_appeared
(
character_id serial not null references characters(character_id),
game_id serial not null references games(game_id)
);
```

	character_id integer	game_id integer
1	1	1
2	1	2
3	1	3
4	1	6
5	1	7
6	1	8
7	2	1
8	2	2
9	2	3
10	2	4
11	2	7
12	3	1
13	3	2
14	3	3
15	3	7
16	4	1
17	4	6
18	4	7
19	5	2
20	5	3
21	6	6
22	7	6
23	8	6
24	9	6
25	10	6
26	11	6
27	12	6
28	13	2
29	14	2
30	15	2
31	16	2
32	16	3
33	17	2
34	18	1
35	18	7
36	19	8
37	20	8

Species Planet

The species planet entity is responsible for telling the user what species in the games belong to what planets. This serves as a link between the species and planets table. This table is responsible for keeping track of where the many species of the games originate. This table is useful for queries like "What Planet are the Sangheili from?"

```
--This is the creation of the Species Planet table
create table species_planet
(
species_id serial not null references species(species_id),
planet_id serial not null references planets(planet_id)
);
```

	species_id integer	planet_id integer
1	1	6
2	2	9
3	3	10
4	4	8
5	5	11
6	6	10
7	7	10
8	8	10
9	9	5
10	10	3
11	12	10

Character Species

The character species entity shows the species of all of the characters in the halo universe. This table allows the user to simply refer to this table whenever they want to find the specific race of any of the characters. This table also serves as a link between the character table and the species table. This is similar to the previous three in which it refers to two other tables in order to reduce update anomalies. We would refer to this table is someone was to ask "What species is Master Chief?"

```
--This is the creation of the Character Species table
create table character_species
(
character_id serial not null references characters(character_id),
species_id serial not null references species(species_id)
);
```

	character_id integer	species_id integer
1	1	1
2	2	1
3	3	1
4	4	1
5	5	1
6	6	1
7	7	1
8	8	1
9	9	1
10	10	1
11	11	1
12	12	1
13	13	7
14	14	7
15	15	7
16	16	2
17	17	3
18	18	9
19	19	9
20	20	9
21	21	10

Views

The following section demonstrates the creation and use of a few views. The view names are in bold font and are followed by a sample SQL statement used to query the view, sample data generated from the query, and the SQL statements used to define the view.

Spartans

Select * From spartan

	game_title text	character_name text		weight(lbs) integer	gender text	bio text	alliance text
1	Halo: Combat Evolved	Master Chief John-117	84	1200	Male	A Short Bio About This Character	UNSC
2	Halo 2	Master Chief John-117	84	1200	Male	A Short Bio About This Character	UNSC
3	Halo 3	Master Chief John-117	84	1200	Male	A Short Bio About This Character	UNSC
4	Halo: Reach	Master Chief John-117	84	1200	Male	A Short Bio About This Character	UNSC
5	Halo: Combat Evolved Anniversary	Master Chief John-117	84	1200	Male	A Short Bio About This Character	UNSC
6	Halo 4	Master Chief John-117	84	1200	Male	A Short Bio About This Character	UNSC
7	Halo: Reach	Commander Carter-A259 (Noble One)	82	950	Male	A Short Bio About This Character	UNSC
8	Halo: Reach	Lieutenant Commander Catherine-B320 (Noble Two)	81	850	Female	A Short Bio About This Character	UNSC
9	Halo: Reach	Emile-A239 (Noble Three)	82	1000	Male	A Short Bio About This Character	UNSC
10	Halo: Reach	Chief Warrent Officer Jorge-052 (Noble Four)	96	1500	Male	A Short Bio About This Character	UNSC
11	Halo: Reach	Jun-A266 (Noble Five)	83	945	Male	A Short Bio About This Character	UNSC
12	Halo: Reach	Spartan-B312 (Noble Six)	81	1200	Player	A Short Bio About This Character	UNSC

create view spartans as
select games.name as game_title,
characters.name as character_name,
characters.Height_in as "height(in)",
characters.Weight_lbs as "weight(lbs)",
characters.gender as gender,
characters.Bio as Bio,
characters.alliance as Alliance
from games, games_appeared, Characters, character_species
where games.game_id = games_appeared.game_id and
games_appeared.character_id = characters.character_id and
characters.character_id = character_species.character_id and
character_species.character_type = 'Spartan'

Marines

Select * **From** marine

	game_title text	character_name text		weight(lbs) integer		bio text	alliance text
1	Halo: Combat Evolved	Sergeant Major Avery Junior Johnson	74	185	Male	A Short Bio About This Character	UNSC
2	Halo 2	Sergeant Major Avery Junior Johnson	74	185	Male	A Short Bio About This Character	UNSC
3	Halo 3	Sergeant Major Avery Junior Johnson	74	185	Male	A Short Bio About This Character	UNSC
4	Halo: Combat Evolved Anniversary	Sergeant Major Avery Junior Johnson	74	185	Male	A Short Bio About This Character	UNSC
5	Halo: Combat Evolved	Captain Jacob Keyes	76	212	Male	A Short Bio About This Character	UNSC
6	Halo: Reach	Captain Jacob Keyes	76	212	Male	A Short Bio About This Character	UNSC
7	Halo: Combat Evolved Anniversary	Captain Jacob Keyes	76	212	Male	A Short Bio About This Character	UNSC
8	Halo 2	Miranda Keyes	68	135	Female	A Short Bio About This Character	UNSC
9	Halo 3	Miranda Keyes	68	135	Female	A Short Bio About This Character	UNSC
10	Halo: Reach	Doctor Catherine Elizabeth Halsey	67	120	Female	A Short Bio About This Character	UNSC

create view marines as
select games.name as game_title,
characters.name as character_name,
characters.Height_in as "height(in)",
characters.Weight_lbs as "weight(lbs)",
characters.gender as gender,
characters.Bio as Bio,
characters.alliance as Alliance
from games, games_appeared, Characters, character_species
where games.game_id = games_appeared.game_id and
games_appeared.character_id = characters.character_id and
characters.character_id = character_species.character_id and
character_species.character_type = 'Marine'

A.I.

Select * From AI

	game_title text	character_name text	height(in) integer	weight(lbs) integer	gender text	bio text	alliance text
1	Halo: Combat Evolved	Cortana	7	1	Female	A Short Bio About This Character	UNSC
2	Halo 2	Cortana	7	1	Female	A Short Bio About This Character	UNSC
3	Halo 3	Cortana	7	1	Female	A Short Bio About This Character	UNSC
4	Halo Wars	Cortana	7	1	Female	A Short Bio About This Character	UNSC
5	Halo: Combat Evolved Anniversary	Cortana	7	1	Female	A Short Bio About This Character	UNSC
6	Halo: Combat Evolved	343 Guilty Spark (The Monitor of Installation 04)	20	25	Male	A Short Bio About This Character	Forerunner
7	Halo: Combat Evolved Anniversary	343 Guilty Spark (The Monitor of Installation 04)	20	25	Male	A Short Bio About This Character	Forerunner

create view AI as
select games.name as game_title,
characters.name as character_name,
characters.Height_in as "height(in)",
characters.Weight_lbs as "weight(lbs)",
characters.gender as gender,
characters.Bio as Bio,
characters.alliance as Alliance
from games, games_appeared, Characters, character_species
where games.game_id = games_appeared.game_id and
games_appeared.character_id = characters.character_id and
characters.character_id = character_species.character_id and
character_species.character_type = 'A.I.'

Stored Procedures

The following section demonstrates the use of stored procedures to query the database for information. These procedures can be called through a web-interface or application program and will return the information specified by the user input

SP: All Games with a given Character

```
create function listgames(Halo_Character text)
returns table (Name text, publisher text) as $$
begin
```

Procedure Call: **Select** listgames('Cortana')

return query select games.name, game.publisher
from games, game_appeared, characters
where games.game_id = games_appeared.game_id and
games_appeared.character_id = characters.cid and
characters.name = Halo_Character;

end;

create function listnames(Race name text)

SP:All Characters within a given race

```
returns table (Name text, publisher text) as $$
begin

return query select characters.name, character.alliance

From characters, character_species, species

where characters.character_id = character_species.character_id and

character_species.species_id = species.species_id and

species.name = Race_name;
```

end:

\$\$ language plpgsql

Sample Reports

The following section demonstrates how the database can be queried to generate useful information.

Query: What species is Master Chief?

```
select species.name
from species, character_species, characters
where species.species_id = character_Species.Species_id and
character_species.character_id = characters.character_id and
characters.name = 'Master Chief John-117'
```

Result: Human

Query: What Planet are the Sangheili from?

```
select planets.name
from planets, species_planet, species
where planets.planet_id = species_planet.planet_id and
species_planet.species_id = species.species_id and
species.name = 'Sangheili'
```

Result: Decided Heart

Query: What are the games in which Master Chief has appeared in?"

```
select games.name
from games, games_appeared, characters
where games.game_id = games_appeared.game_id and
games_appeared.character_id = characters.character_id and
characters.name = 'Master Chief John-117'
```

	name text			
1	Halo: Combat Evolved			
2	Halo 2			
3	Halo 3			
4	Halo: Reach			
5	Halo: Combat Evolved Anniversary			
6	Halo 4			

Security

We made the decision to form two user-types with the following usage requirements.

DBA: The DBA performs general upkeep of the database. This may include updating game
info for games that are still in production, inserting data for new games or characters, and
correcting any exsiting error or conflicts in the database. These operations require full read
and write access to all of the views and the base tables. The following SQL statement
explains how to create a DBA role and enable access to the operations needed to perform
upkeep.

CREATE ROLE dba
GRANT SELECT, INSERT, UPDATE
ON ALL TABLES IN SCHEMA PUBLIC
TO dba

• Public User: The public user is restricted to performing only select operations on the views and to only a select few of the base tables. The public user could represent an application developer or a web-designer looking to create a fansite or just a normal person who wants access to the games data. This user should have minimal knowledge of the base tables. The data that we hold in our database is not for the most part sensitive, but a malicious user could still use the knowledge of our database structure to destroy the database of portray an image that we do not wish to represent. The following SQL statements demonstrate one method for creating a public user.

CREATE ROLE public_user
GRANT SELECT
ON bonds, villains, henchmen, movies
TO public_user

Additional security measures, such as secure authentication, will depend on the host system and on the chosen DBMS.

Known Issues

• The current design only allows for the characters to be allied with one faction, this is a problem going from Halo 2 to Halo 3. The Arbiter in between the games becomes an ally to the UNSC this isn't properly portrayed in the database because of this error.

• Currently the flood is listed as being genderless, when in actuality it can actually be Genderless, Male, or Female depending on its mutation. So there is technically an error because the flood consumes and absorbs the physical traits if the sentient life that it consumes

Future Enhancements

Some features to consider for future updates.

- Query the database for Recognizable skulls.
- Query the database by Chapter within the games.
- Query the Skulls within the game, what they do and in some cases where they are located
- Retrieve the Game awards
- Retrieve the Games soundtrack information for each game.