Conor Kincart – Database Systems Design Project

The Legend of Zelda: Ocarina of Time

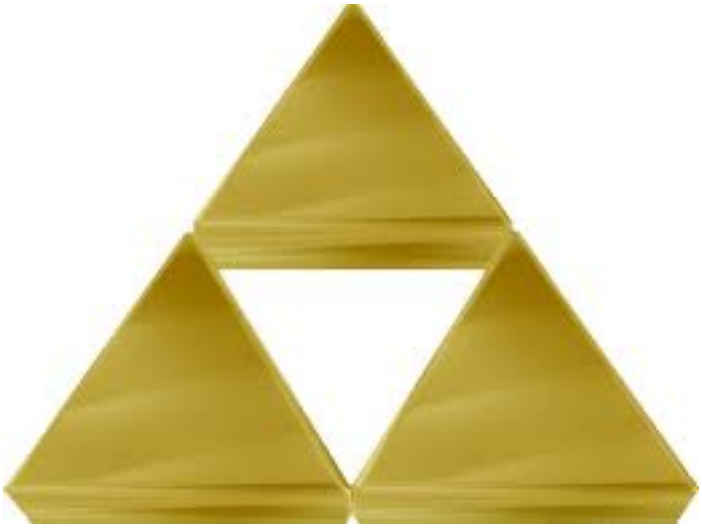
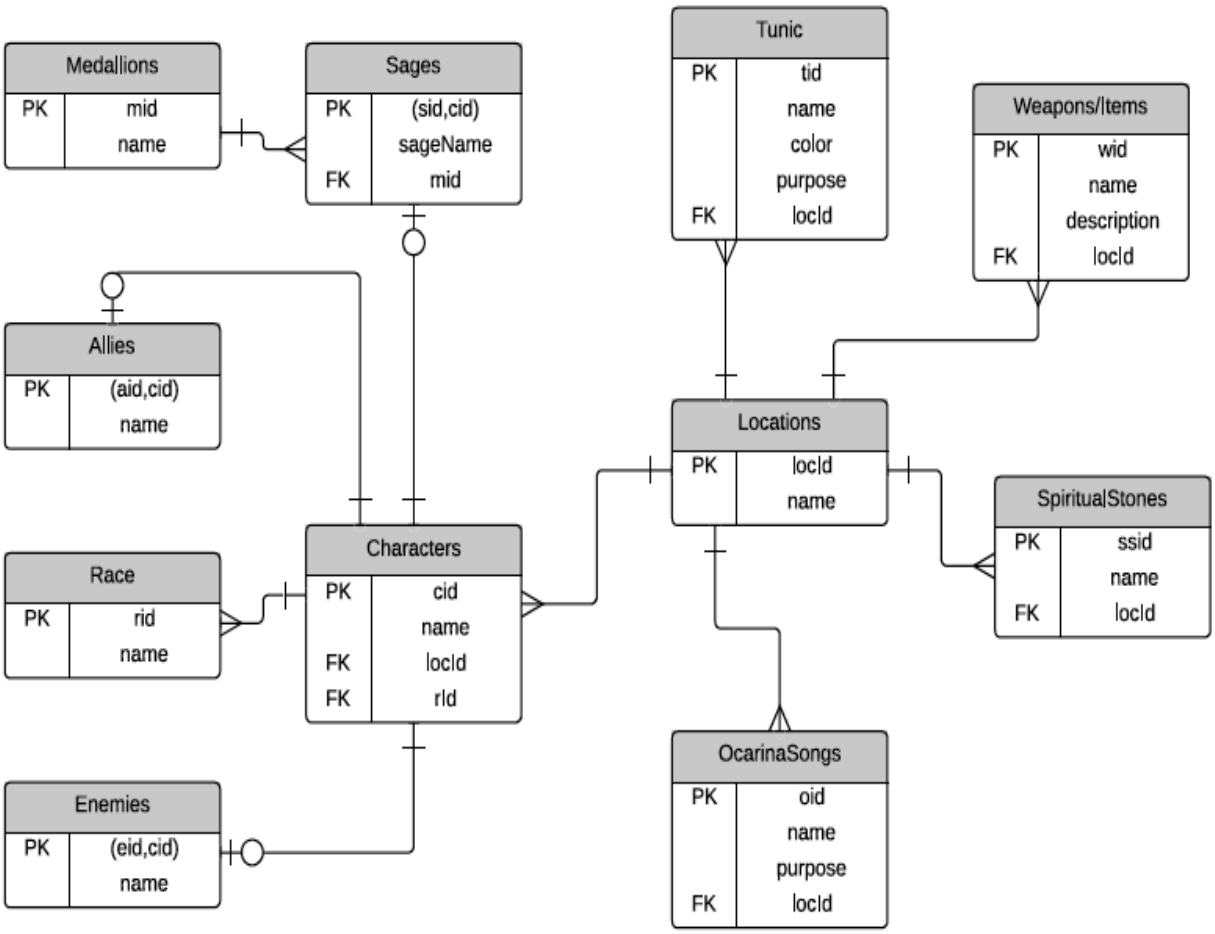


Table of Contents

Executive Summary

The following report gives an in depth look at the 1998 Nintendo 64 game, The Legend of Zelda: Ocarina of Time. It provides a reference for users that will allow insight to help them along their journey throughout the game. For people who are not playing the game, it gives them an introduction to get a feel for the game's concept. The report includes lists and descriptions of characters; both allies and enemies, locations, items, and various other concepts of the game. Views created in this document would be helpful for readers to find what they are looking for in the game quickly, without wasting hours of gameplay without resulting in any accomplishments.

Entity Relationship Diagram



Tables

(Create statements, functional dependencies, and sample data)

Characters Table

Purpose:

Display a list of characters that are present in the Ocarina of Time game.

Functional Dependencies:

cid ➔ name, locId, rid

Create Statement:

```
create table Characters (
    cid         char(3) not null,
    name            text,
    locId           char(5) not null,
    rid         char(3) not null,
    Primary Key (cid) );
```

Sample Data:

	cid character(3)	name text	locid character(5)	rid character(3)
1	c01	Link	loc01	r01
2	c02	Zelda	loc09	r01
3	c03	Ganondorf	loc14	r03
4	c04	Sheik	loc04	r01
5	c05	Rauru	loc13	r01
6	c06	Saria	loc01	r02
7	c07	Darunia	loc06	r04
8	c08	Princess_Ruto	loc05	r05
9	c09	Impa	loc04	r01
10	c10	Nabooru	loc14	r03
11	c11	Malon	loc03	r01
12	c12	Talon	loc03	r01
13	c13	Biggoron	loc06	r04
14	c14	King_Zora	loc05	r05
15	c15	Queen_Gohma	loc08	r06
16	c16	King_Dodongo	loc06	r06
17	c17	Barinade	loc07	r06
18	c18	Volvagia	loc11	r06
19	c19	Bongo_Bongo	loc15	r06

Race Table

Purpose:

Describe the different races of characters throughout the game.

Functional Dependencies:

rid ➔ name

Create Statement:

```
create table Race(
    rid             char(3) not null,
    name            text,
    Primary Key (rid)
);
```

Sample Data:

	rid character(3)	name text
1	r01	Hyrulian
2	r02	Kokori
3	r03	Gerudian
4	r04	Kokori
5	r05	Zoran
6	r06	Shadow_Being

Allies Table

Purpose:

Allows user to examine a list of ally characters in the game to seek help from.

Functional Dependencies:

(aid,cid) ➔ name

Create Statement:

```
create table Allies (
        aid         char(3) not null,
        cid         char(3) not null,
        name        text,
        Primary Key (aid,cid)
);
```

Sample Data:

	cid character(3)	aid character(3)	name text
1	c02	a01	Zelda
2	c04	a02	Sheik
3	c05	a03	Rauru
4	c06	a04	Saria
5	c07	a05	Darunia
6	c08	a06	Princess Ruto
7	c09	a07	Impa
8	c10	a08	Nabooru
9	c11	a09	Malon
10	c12	a10	Talon
11	c13	a11	Biggoron
12	c14	a12	King_Zora

Enemies Table

Purpose:

Describes your in game enemies and charts out major bosses you will encounter during the game.

Functional Dependencies:

(eid, cid) ➔ name

Create Statement:

```
create table Enemies (
        eid         char(3) not null,
        cid         char(3) not null,
        name        text,
        Primary Key (eid,cid)
);
```

Sample Data:

	cid character(3)	eid character(3)	name text
1	c03	e01	Ganondorf
2	c15	e02	Queen_Gohma
3	c16	e03	King_Dodongo
4	c17	e04	Barinade
5	c18	e05	Volvagia
6	c19	e06	Bongo_Bongo

Sages Table

Purpose:

Displays the holy sages who will grant you power during the game.

Functional Dependencies:

(sid, cid) ➔ sageName

Create Statement:

```
create table Sages (
        sid         char(3) not null,
        cid         char(3) not null,
        name        text,
        Primary Key (sid,cid)
);
```

Sample Data:

	cid character(3)	sid character(3)	sagename text
1	c05	s01	Rauru
2	c06	s02	Saria
3	c07	s03	Darunia
4	c08	s04	Princess_Ruto
5	c09	s05	Impa
6	c10	s06	Nabooru

Medallions Table

Purpose:

Lists medallions that you will receive from different stages which track progress and reward completion of temples.

Functional Dependencies:

mid ➔ name

Create Statement:

```
create table Medallions (
    mid     char(3) not null,
    name    text,
    Primary Key (mid)
);
```

Sample Data:

	mid character(3)	name text
1	m01	Light Medallion
2	m02	Forest Medallion
3	m03	Fire Medallion
4	m04	Water Medallion
5	m05	Shadow Medallion
6	m06	Spirit Medallion

Locations Table

Purpose:

This table allows the player to take a look at what locations there are, and will later help show where to find certain weapons, items, and songs for the Ocarina.

Functional Dependencies:

locId➔ name

Create Statement:

```
create table Locations (
    locId       char(5) not null,
    name        text,
    Primary Key (locId)
);
```

Sample Data:

	locid character(5)	name text
1	loc01	Kokori Forest
2	loc02	Hyrule Field
3	loc03	Lon Lon Ranch
4	loc04	Kakariko Village
5	loc05	Zoras Domain
6	loc06	Goron City
7	loc07	Inside Jabu Jabus Belly
8	loc08	Inside the Deku Tree
9	loc09	Hyrule Castle
10	loc10	Forest Temple
11	loc11	Fire Temple
12	loc12	Ice Cavern
13	loc13	Temple of Time
14	loc14	Gerudo Valley
15	loc15	Shadow Temple

Weapons/Items Table

Purpose:

Shows which items and weapons are available to pick up throughout the game, including where to find these items and a brief description of them.

Functional Dependencies:

wid ➔ name, description, locId

Create Statement:

```
create table WeaponsItems (
    wid             char(3) not null,
    name            text,
    description     text,
    locId       char(5) not null,
    Primary Key (wid)
    );
```

Sample Data:

	wid character(3)	name text	description text	locid character(5)
1	w01	Kokori Sword	Your first blade in the game.	loc01
2	w02	Deku Shield	Bought in the Deku Shop	loc01
3	w03	Fairy Slingshot	First projectile weapon, use to shoot eyes.	loc08
4	w04	Bombs	Look for cracks in the wall.	loc06
5	w05	Fairy Bow	Finally a real bow and arrow.	loc10
6	w06	Megaton Hammer	Very powerful, bone-crushing hammer.	loc11
7	w07	Gorons Bracelet	You can now pickup bomb flowers.	loc06
8	w08	Iron Boots	Now stay underwater.	loc12
9	w09	Master Sword	You can slay Ganondorf with this ultimate sword.	loc13
10	w10	Hylian Shield	80 rupees at the market.	loc09

Spiritual Stones Table

Purpose:

Provides a table that shows the location of each of the three spiritual stones that you need to acquire.

Functional Dependencies:

ssid ➔ name, locId

Create Statement:

```
create table SpiritualStones (
        ssid            char(4) not null,
        name            text,
        locId           char(5) not null,
    Primary Key (ssid)
);
```

Sample Data:

	ssid character(4)	name text	locid character(5)
1	ss01	Kokori Emerald	loc08
2	ss02	Goron Ruby	loc06
3	ss03	Zora Sapphire	loc07

Ocarina Songs Table

Purpose:

Explains the effect of what specific songs will do when you play them, and where to first learn how to play them.

Functional Dependencies:

oid ➔ name, purpose (effect), locId

Create Statement:

```
create table OcarinaSongs (
        oid             char(3) not null,
        name            text,
        purpose         text,
        locId           char(5) not null,
        Primary Key (oid)
);
```

Sample Data:

	oid character(3)	name text	purpose text	locid character(5)
1	o01	Zeldas Lullaby	Various magical purposes.	loc09
2	o02	Sarias Song	Lets you always speak with Saria.	loc01
3	o03	Eponas Song	Calls your horse.	loc03
4	o04	Song of Time	Allows access to the Master Sword.	loc13
5	o05	Minuet of Forest	Teleports Link to the Forest Temple.	loc01
6	o06	Bolero of Fire	Teleports Link to the Fire Temple.	loc06
7	o07	Serenade of Water	Teleports Link to the Water Temple.	loc05
8	o08	Song of Storms	Creates stormy weather.	loc04

Tunic Table

Purpose:

This table gives you information on the three different tunics, their color, effect, and where to find them.

Functional Dependencies:

tid ➔ name, color, purpose, locId

Create Statement:

```
create table Tunic (
        tid         char(3) not null,
        name        text,
        color       text,
        purpose     text,
        locId       char(5) not null,
        Primary Key (tid)
);
```

Sample Data:

	tid character(3)	name text	color text	purpose text	locid character(5)
1	t01	Kokori Tunic	green	Standard tunic.	loc01
2	t02	Goron Tunic	red	Allows unlimited extreme heat.	loc06
3	t03	Zora Tunic	blue	Allows breathing underwater.	loc05

Views

This view shows the location of where each tunic can be found throughout the game. Tunics allow access to areas of certain temperature, or depth of water. Knowing the location to find the correct tunic will allow the player to gain access to these locations, allowing advance in the game.

```
Create view tunicLocator AS

select locations.name as Locations_Name,
        t.name

from Locations,
        Tunic t

where t.locid = Locations.locId
;
```

Reports / Queries

These two reports are useful to the player by informing them where a specific weapon or ocarina song can be found with a spiritual stone in the game.

Bomb Report:

```
select l.name, w.name, l.locID
from Locations l,
        WeaponsItems w
where l.locId = w.locID and w.name = 'Bombs'
;
```

Ocarina Songs:

```
select o.name, l.name, s.name
from OcarinaSongs o,
        Locations l,
        SpiritualStones s
where l.locId = o.locId
    and s.locId = l.locId
 ;
```

Stored Procedures

This stored procedure is beneficial to users in that it shows which sage provides which medallion when you complete a given temple. It uses the Medallions and Sages tables.

Medallions Stored Procedure:

```
create or replace function get_medallions_by_sage(int, REFCURSOR)
returns refcursor as
$$
declare
        medallion int       := $1;
        resultset   REFCURSOR := $2;
begin
        open resultset for
                select m.name, m.locId, s.name,
                from   Medallions m,
                        Sages s
                where  m.locId = s.locId;
       return resultset;
end;
$$
language plpgsql;
select get_medallions_by_sage(1, 'results');
Fetch all from results;
```

This store procedure will work with the following trigger. This procedure inputs a value for the name for a given character if the name attribute is left null. It will be replaced with the unknown character name.

Character Stored Procedure:

```
Create function Character_Name()

Return trigger as $$

Begin
	If (name == null) then
	Update Characters set name = “Unknown Character Name” where name = null;
	End if;

End

$$

Language plpgsql

;
```

Triggers

This trigger checks when new Character data is inputted into the Characters Table. The trigger calls the Stored Procedure 'Character' and runs the Stored Procedure to see if any null values were entered in the name field. If they were, an assigned value will be swapped with the null value.

```
Create Trigger movie_title_trigger

        After insert or update

        On Characters

        For each row

        Execute Procedure Character_Name()
;
```

Security

Security allows access for two user types.

First, you grant access for the Admin for the database:

```
Create role admin
Grant select, insert, update,
On all tables in schema public
To admin
;
```

Secondly, you want public users to have some access:

```
Create role user,
Grant select,
On all tables in schema public,
To public
;
```

Implementation Notes, Known Problems, and Future Enhancements

Implementation Notes:

This database provides knowledge for the user to access certain information on the details and locations of various objects of the game. The user will need a list of search statements to gain access, but should be fairly easy to come to the required query search.

Known Problems:

Weapons only show where they are originally found in the game, a specific column could have been dedicated to where it is originally found versus other areas.

Future Enhancements:

This database focuses on only certain aspects of this game. There are over ten other titles in the series, and the scope of this database could be expanded to cover other games on the Nintendo 64 along with other consoles as well.