Design-Your-Own Database Assignment

Timothy Zwart

CMPT\_308N\_112

Dr. Donald Schwartz

December 5, 2016

**Table of Contents**

Cover Page…………………….....……………………………………………………………….1

Table of Contents……………………………………………………………………………….....2

Model Description.…………….....……………………………………………………………….3

E/R Diagram………………….....……………………………………………………………….5

Create Tables...……………….....……………………………………………………………….6

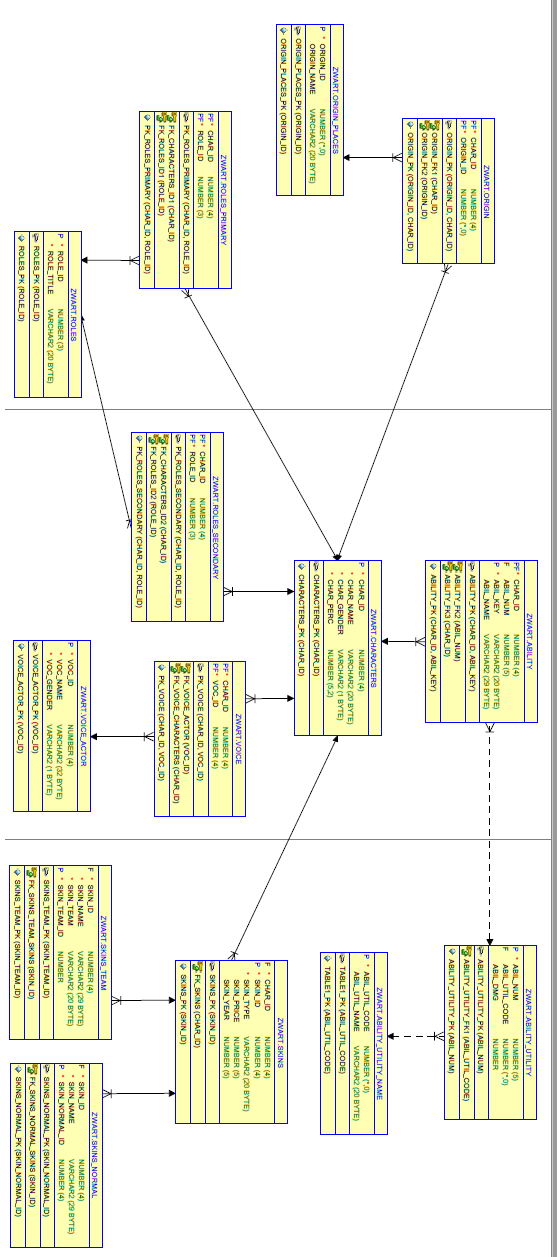
Justification..………………….....……………………………………………………………….21

Queries…......………………….....………………………………………………………………22

**Model Description**

Riot games wants to compile data into a database in order to compare characters play rates and what may be affecting their play rates. This is to ensure that when developing new characters they will be more inclined to be used. Since the database is a prototype there will need to only be the first 20 characters out of the over 100 character cast.

1. There each character must be assigned its own identification number which will be used to link the characters to other tables. The each character’s name must be identifiable in this main table. Each character must be assigned its gender within a column. Each character must also be assigned its play rate percentage.
2. Each Character has a primary role. Within the game there are 6 roles a character can have for primary roles. Every character is required to have one of these roles as this primary role.
3. Within the game the some characters are assigned secondary roles as well, but not all characters are assigned a secondary role. The roles for secondary are picked from the same six roles as the primary role.
4. Each character has 4 abilities. Each one of these abilities must receive its own identifier. Each ability is character specific, so no other character can have the same ability. Each Character has one and only one ability appearing on each of the following keyboard key: q,w,e,r. These each ability can be found by linking both its specific key with the characters identification number.
5. A majority of the abilities have their own effects referred to as the abilities utility. Although, a majority of abilities do some do not offer utility. Each utility comes from a list of possible utilities. Each utility should be given its own identifier. This identifier then should be used to link and ability to the utility it offers. Each utility must only have one utility.
6. Each ability does an amount of damage. Some abilities don’t do any damage but must be given the value zero, in case of adding up each character’s total damage. The range of values for damage for all the abilities is too wide to add an identifier to. This damage must be placed with the utility the character provides.
7. Each Character comes from a region within storylines behind the game. One or more characters may come from a region. More characters could be added. These new characters maybe from either regions that exist or that may not currently exist. Each region must have a unique identifier. This unique identifier must connect to the character identification number, in a separate table. ‘Unknown’ must be included as one of the places of origin. This is to be used for any character that does not have a currently know origin. ‘Null’ must be reserved for any characters in development that may not have an origin that has been decided yet.
8. Every Character is released or is plan to be released with at least one skin that is able to be purchased once a character is released to the public. Skins are cosmetics used to change a character’s appearance. Although, some skins are released with the characters, some are released later on possibly years after. Therefore, each skin needs to have its year of release indicated. The skins must have a unique identifier because some skins have similar names, and could cause problems if misspelled. Each skin is character specific, meaning only one specific character of the cast can wear the skin. Every skin has a price, but the pricing overall does not depend on the skin’s typing. Some skins do share the same price. Many Skins share the same typing. The price of the skin is valued based on a currency created by the company. There are also many different skin types.
   1. One type of Skin is called Team Skins. These skins require a partially different set of data, due to the circumstances they are created for. The name of the team that won. And the name of the skin. Each team skin requires a given identification number pertaining only to team skins
   2. All others skins can be grouped into a table referred to as normal skins. This table requires the name of the skins and their own unique normal skin identification number.
9. Every character has a voice actor. Although, one voice actor can voice multiple characters. The Voice actor’s gender must also be included, some characters have voice actors of the opposite gender. The voice actor must have their own number to be identified by. The voice actor name should be within one field because some voice actors may go by one word, instead of a full name. Do not include any voice actor’s middle name only use their one word or their First and Last name for this Column. Gender for Actors must use the same values as gender for Characters to indicate male and female.

****

**Create Table Statements**

CREATE TABLE "ABILITY"

( "CHAR\_ID" NUMBER(4,0) NOT NULL ENABLE,

"ABIL\_NUM" NUMBER(5,0),

"ABIL\_KEY" VARCHAR2(20 BYTE) NOT NULL ENABLE,

"ABIL\_NAME" VARCHAR2(29 BYTE),

CONSTRAINT "ABILITY\_PK" PRIMARY KEY ("CHAR\_ID", "ABIL\_KEY"),

CONSTRAINT "ABILITY\_FK2" FOREIGN KEY ("ABIL\_NUM")

REFERENCES "ABILITY\_UTILITY" ("ABIL\_NUM") ENABLE,

CONSTRAINT "ABILITY\_FK3" FOREIGN KEY ("CHAR\_ID")

REFERENCES "CHARACTERS" ("CHAR\_ID") ENABLE);

Creates the ability table. Linked to tables Ability Utility and Characters. Contains the characters identification. Contains the ability identification number. The abilities’ names. And the Key that the ability uses on the keyboard. Primary Key is a composite key of {Char\_ID, Abil\_Key}. The Foreign keys are Char\_ID and Abil\_Num.

CREATE TABLE "ABILITY\_UTILITY"

( "ABIL\_NUM" NUMBER(5,0) NOT NULL ENABLE,

"ABIL\_UTIL\_CODE" NUMBER(\*,0),

"ABIL\_DMG" NUMBER,

CONSTRAINT "ABILITY\_UTILITY\_PK" PRIMARY KEY ("ABIL\_NUM"),

CONSTRAINT "ABILITY\_UTILITY\_FK1" FOREIGN KEY ("ABIL\_UTIL\_CODE")

REFERENCES "ABILITY\_UTILITY\_NAME" ("ABIL\_UTIL\_CODE") ENABLE

);

Creates the Ability\_Utility Table. Linked to the Ability table and the Ability\_Utilitiy\_Name table. Contains Columns Abil\_Num, Abil\_Util\_ Abil\_Dmg. The Primary Key is Abil\_Num. The Foreign Key is Abil\_Util\_Code.

CREATE TABLE "ABILITY\_UTILITY\_NAME"

( "ABIL\_UTIL\_CODE" NUMBER(\*,0) NOT NULL ENABLE,

"ABIL\_UTIL\_NAME" VARCHAR2(20 BYTE),

CONSTRAINT "TABLE1\_PK" PRIMARY KEY ("ABIL\_UTIL\_CODE"));

Creates the Ability\_Utility\_Name table. Linked to the Ability\_Utility table. The Primary Key is Abil\_Util\_Code. There is no Foreign key.

CREATE TABLE "CHARACTERS"

( "CHAR\_ID" NUMBER(4,0) NOT NULL ENABLE,

"CHAR\_NAME" VARCHAR2(20 BYTE) NOT NULL ENABLE,

"CHAR\_GENDER" VARCHAR2(1 BYTE) NOT NULL ENABLE,

"CHAR\_PERC" NUMBER(5,2) NOT NULL ENABLE,

CONSTRAINT "CHARACTERS\_PK" PRIMARY KEY ("CHAR\_ID"));

Creates the Characters table. This table links all the tables on the different strands together within the database. Contains Char\_ID, Char\_Name, Char\_Gender, and Char\_Perc Char\_ID is the Primary Key. There is no Foreign Key. Directly connected to the tables Voice, Roles\_Primary, Roles\_Secondary, Skins, Ability, and Origin.

CREATE TABLE "ORIGIN"

( "CHAR\_ID" NUMBER(4,0) NOT NULL ENABLE,

"ORIGIN\_ID" NUMBER(\*,0) NOT NULL ENABLE,

CONSTRAINT "ORIGIN\_PK" PRIMARY KEY ("ORIGIN\_ID", "CHAR\_ID"),

CONSTRAINT "ORIGIN\_FK1" FOREIGN KEY ("CHAR\_ID")

REFERENCES "CHARACTERS" ("CHAR\_ID") ENABLE,

CONSTRAINT "ORIGIN\_FK2" FOREIGN KEY ("ORIGIN\_ID")

REFERENCES "ORIGIN\_PLACES" ("ORIGIN\_ID") ENABLE

);

Creates the Origin table. Includes Char\_ID and Origin\_ID. The Primary Key is the Composite key of {Char\_ID, Origin\_ID}. The Foreign Keys are CHAR\_ID and ORIGIN\_ID. This table is linked to Origin\_Places and Characters.

CREATE TABLE "ORIGIN\_PLACES"

( "ORIGIN\_ID" NUMBER(\*,0) NOT NULL ENABLE,

"ORIGIN\_NAME" VARCHAR2(20 BYTE),

CONSTRAINT "ORIGIN\_PLACES\_PK" PRIMARY KEY ("ORIGIN\_ID"));

Creates the Origin\_Places table. This table holds the columns Origin\_ID and Origin\_Name. The Primary Key is Origin\_ID. There is no Foreign Key. Linked to Origin table.

CREATE TABLE "ROLES"

( "ROLE\_ID" NUMBER(3,0) NOT NULL ENABLE,

"ROLE\_TITLE" VARCHAR2(20 BYTE) NOT NULL ENABLE,

CONSTRAINT "ROLES\_PK" PRIMARY KEY ("ROLE\_ID"));

Creates the Roles table. Contains columns Role\_ID and Role\_Title. This table is linked to both the Roles\_Primary table and the Roles\_Secondary table. The Primary Key for this table is Role\_ID.

CREATE TABLE "ROLES\_PRIMARY"

( "CHAR\_ID" NUMBER(4,0) NOT NULL ENABLE,

"ROLE\_ID" NUMBER(3,0) NOT NULL ENABLE,

CONSTRAINT "PK\_ROLES\_PRIMARY" PRIMARY KEY ("CHAR\_ID", "ROLE\_ID"),

CONSTRAINT "FK\_CHARACTERS\_ID1" FOREIGN KEY ("CHAR\_ID")

REFERENCES "CHARACTERS" ("CHAR\_ID") ENABLE,

CONSTRAINT "FK\_ROLES\_ID1" FOREIGN KEY ("ROLE\_ID")

REFERENCES "ROLES" ("ROLE\_ID") ENABLE

);

Creates the Roles\_Primary table. This table contains the columns Char\_ID and Role\_ID. The Primary Key for this table is the composite key of {Char\_ID,Role\_ID}. The Foreign Keys are Char\_ID and Role\_ID. This table is linked to the Characters table and the Roles table.

CREATE TABLE "ROLES\_SECONDARY"

( "CHAR\_ID" NUMBER(4,0) NOT NULL ENABLE,

"ROLE\_ID" NUMBER(3,0) NOT NULL ENABLE,

CONSTRAINT "PK\_ROLES\_SECONDARY" PRIMARY KEY ("CHAR\_ID", "ROLE\_ID"),

CONSTRAINT "FK\_CHARACTERS\_ID2" FOREIGN KEY ("CHAR\_ID")

REFERENCES "CHARACTERS" ("CHAR\_ID") ENABLE,

CONSTRAINT "FK\_ROLES\_ID2" FOREIGN KEY ("ROLE\_ID")

REFERENCES "ROLES" ("ROLE\_ID") ENABLE

);

Creates the Roles\_Secondary table. This table contains the columns Char\_ID and Role\_ID. The Primary Key for this table is the composite key of {Char\_ID,Role\_ID}. The Foreign Keys are Char\_ID and Role\_ID. This table is linked to the Characters table and the Roles table.

CREATE TABLE "SKINS"

( "CHAR\_ID" NUMBER(4,0) NOT NULL ENABLE,

"SKIN\_ID" NUMBER(4,0) NOT NULL ENABLE,

"SKIN\_TYPE" VARCHAR2(20 BYTE) NOT NULL ENABLE,

"SKIN\_PRICE" NUMBER(5,0),

"SKIN\_YEAR" NUMBER(5,0),

CONSTRAINT "SKINS\_PK" PRIMARY KEY ("SKIN\_ID"),

CONSTRAINT "FK\_SKINS" FOREIGN KEY ("CHAR\_ID")

REFERENCES "CHARACTERS" ("CHAR\_ID") ENABLE

);

Creates the Skins table. Contains the columns of Char\_ID, Skin\_ID, Skin\_Type, Skin\_Price, and Skin\_Year. This table is linked to the Characters table, the Skins\_Team table, and the Skins\_normal table. The Primary Key for this table is Skins\_ID. The Foreign Key for this table is Char\_ID.

CREATE TABLE "SKINS\_NORMAL"

( "SKIN\_ID" NUMBER(4,0) NOT NULL ENABLE,

"SKIN\_NAME" VARCHAR2(29 BYTE) NOT NULL ENABLE,

"SKIN\_NORMAL\_ID" NUMBER(4,0) NOT NULL ENABLE,

CONSTRAINT "SKINS\_NORMAL\_PK" PRIMARY KEY ("SKIN\_NORMAL\_ID"),

CONSTRAINT "FK\_SKINS\_NORMAL\_SKINS" FOREIGN KEY ("SKIN\_ID")

REFERENCES "SKINS" ("SKIN\_ID") ENABLE

);

Creates the Skins\_Normal table. Contains the columns Skin\_ID, Skin\_Name, and Skin\_Normal\_ID. Linked to Skins table. The Primary Key is Skins\_Normal\_ID. The Foreign Key is Skin\_ID.

CREATE TABLE "SKINS\_TEAM"

( "SKIN\_ID" NUMBER(4,0) NOT NULL ENABLE,

"SKIN\_NAME" VARCHAR2(29 BYTE) NOT NULL ENABLE,

"SKIN\_TEAM" VARCHAR2(20 BYTE) NOT NULL ENABLE,

"SKIN\_TEAM\_ID" NUMBER NOT NULL ENABLE,

CONSTRAINT "SKINS\_TEAM\_PK" PRIMARY KEY ("SKIN\_TEAM\_ID"),

CONSTRAINT "FK\_SKINS\_TEAM\_SKINS" FOREIGN KEY ("SKIN\_ID")

REFERENCES "SKINS" ("SKIN\_ID") ENABLE

);

Creates the Skins\_Team table. This table contains columns Skin\_ID, Skin\_Name, Skin\_Team, and Skin\_Team\_ID. This table’s Primary key is Skin\_Team\_ID. This table’s Foreign Key is Skin\_ID. This table is linked to the Skins table.

CREATE TABLE "VOICE"

( "CHAR\_ID" NUMBER(4,0) NOT NULL ENABLE,

"VOC\_ID" NUMBER(4,0) NOT NULL ENABLE,

CONSTRAINT "PK\_VOICE" PRIMARY KEY ("CHAR\_ID", "VOC\_ID"),

CONSTRAINT "FK\_VOICE\_ACTOR" FOREIGN KEY ("VOC\_ID")

REFERENCES "VOICE\_ACTOR" ("VOC\_ID") ENABLE,

CONSTRAINT "FK\_VOICE\_CHARACTERS" FOREIGN KEY ("CHAR\_ID")

REFERENCES "CHARACTERS" ("CHAR\_ID") ENABLE

);

Creates the voice table. Linked to both the Characters table and the Voice\_Actor table. Contains columns Char\_ID and Voc\_ID. The Primary Key for this table is the composite key of {Char\_ID, Voc\_ID}. The Foreign Keys are Char\_ID and Voc\_ID.

CREATE TABLE "VOICE\_ACTOR"

( "VOC\_ID" NUMBER(4,0) NOT NULL ENABLE,

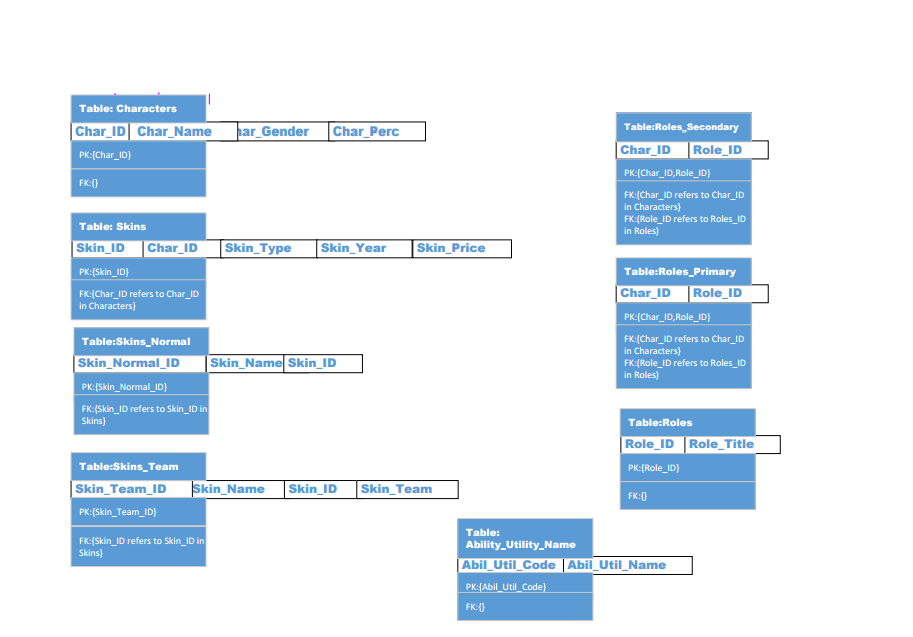
"VOC\_NAME" VARCHAR2(32 BYTE) NOT NULL ENABLE,

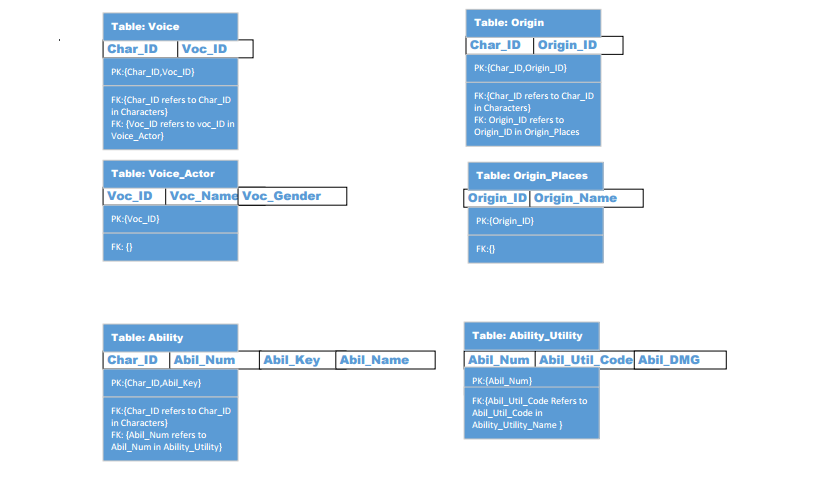
"VOC\_GENDER" VARCHAR2(1 BYTE) NOT NULL ENABLE,

CONSTRAINT "VOICE\_ACTOR\_PK" PRIMARY KEY ("VOC\_ID"));

Creates the voice\_actor table. This table contains the columns Voc\_ID, Voc\_Name, and Voc\_Gender. This table is linked to the Voice table. The Primary Key for this table is Voc\_ID.

There is no Foreign Key for this table.

****

****

**Queries**

All/Every

Name the Characters whose every ability does greater than 100 damage. Cardinality= 3.

SELECT C.char\_name

FROM Characters C

WHERE NOT EXISTS(

SELECT \*

FROM ABILITY AB

WHERE EXISTS(

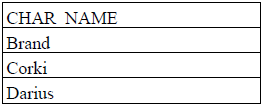
SELECT \*

FROM ABILITY\_UTILITY AU

WHERE AU.ABIL\_NUM = AB.ABIL\_NUM

AND AB.CHAR\_ID=C.CHAR\_ID

AND NOT AU.ABIL\_DMG > 100));



Only

Name the characters whose only skins are event skins. Cardinality 2.

SELECT DISTINCT characters.char\_name

From Characters

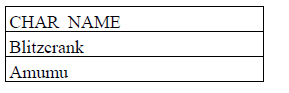
Where Characters.Char\_id Not IN(

SELECT Skins.CHAR\_ID

FROM Skins

Where skins.SKIN\_TYPE <> 'Event'

);



None

Name the characters whose Skins are none of the team skins. Cardinality=17.

SELECT Q.CHAR\_NAME

FROM Characters Q

WHERE NOT EXISTS(

SELECT \*

FROM Skins S

WHERE EXISTS(

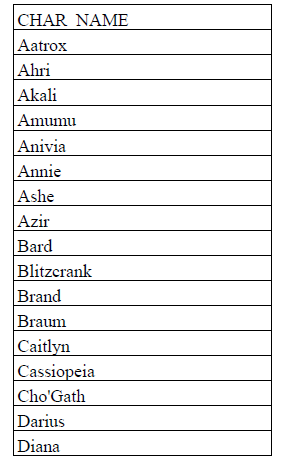
SELECT \*

FROM SKINS\_TEAM ST

WHERE S.CHAR\_ID = Q.CHAR\_ID

AND S.Skin\_ID= ST.Skin\_ID

));



Right Outer Join

Name all the Places of Origin a character can come from, and give any male characters associated with that origin. Cardinaity = 14

SELECT Distinct Origin\_Places.ORIGIN\_NAME,Characters.CHAR\_NAME

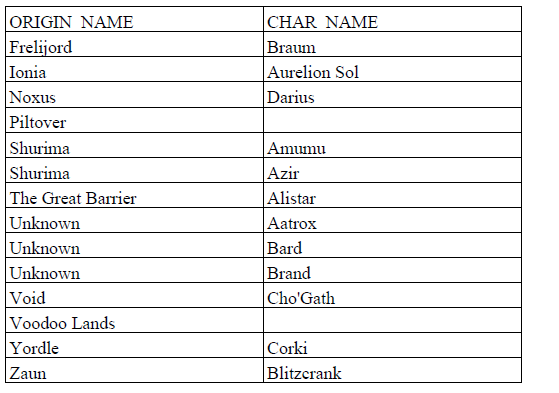
FROM Origin Right outer Join Characters on Origin.CHAR\_ID = Characters.CHAR\_ID

AND Characters.CHAR\_GENDER='m'

Right outer join ORIGIN\_PLACES on Origin\_Places.ORIGIN\_ID = Origin.ORIGIN\_ID

ORDER by ORIGIN\_PLACES.ORIGIN\_NAME

;



Left Outer Join

Name all female characters and provide each ones Primary\_role and Secondary\_Role. Cardinality=8.

SELECT Characters.Char\_name, R1.ROLE\_TITLE AS Role\_Primary, R2.ROLE\_TITLE AS Role\_Secondary

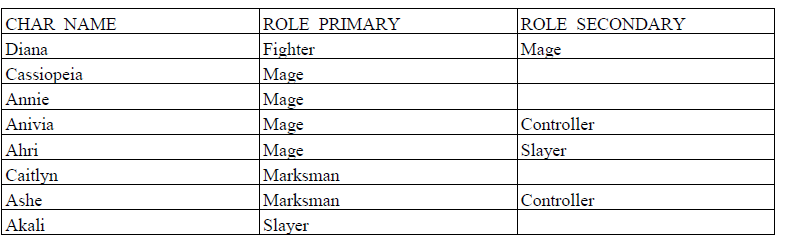
FROM Characters left Outer Join ROLES\_PRIMARY on Characters.char\_ID = Roles\_Primary.char\_ID

Left Outer Join Roles\_Secondary on Characters.Char\_ID = Roles\_Secondary.CHAR\_ID

Left Outer Join ROLES R1 on ROLES\_PRIMARY.ROLE\_ID= R1.ROLE\_ID

Left OUTER JOIN Roles R2 On Roles\_Secondary.Role\_ID= R2.ROLE\_ID

Where Characters.CHAR\_GENDER='f'



Total Outer Join

Name all female characters, also give all their abilities and whatever utility their ability provides for the characters that are picked more than 4% of the time. Cardinality=20.

SELECT DISTINCT Ability.Abil\_name, Characters.Char\_name, ABILITY\_UTILITY\_NAME.ABIL\_UTIL\_NAME

FROM ABILITY FULL join ABILITY\_UTILITY on Ability.ABIL\_NUM= ABILITY\_UTILITY.ABIL\_NUM

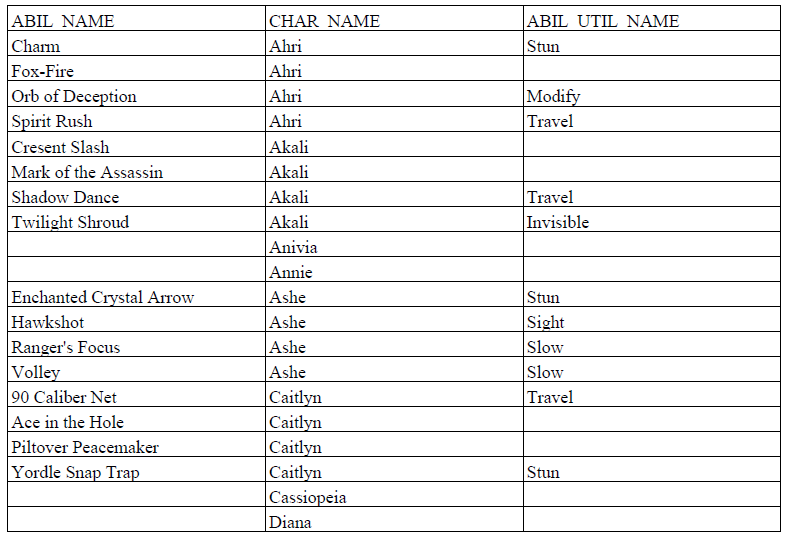
FULL join ABILITY\_UTILITY\_NAME on ABILITY\_UTILITY\_NAME.ABIL\_UTIL\_CODE= ABILITY\_UTILITY.ABIL\_UTIL\_CODE

Full Join Characters ON Characters.CHAR\_ID = Ability.CHAR\_ID

AND Characters.CHAR\_perc > 4

WHERE Characters.CHAR\_GENDER='f'

Order by Characters.Char\_name;



6 Tables

Give all character names who are voiced by the opposite sex, along with the characters origin place and the characters total number of skins. Cardinality=1

SELECT DISTINCT characters.Char\_Name, origin\_places.Origin\_name, Count(Skins.Char\_ID)AS Total\_Skins

FROM CHARACTERS, Origin, origin\_places, voice, voice\_actor,Skins

WHERE Characters.char\_Gender <> voice\_actor.VOC\_GENDER

AND characters.char\_id= Origin.char\_id

AND Origin.Origin\_id = origin\_places.ORIGIN\_ID

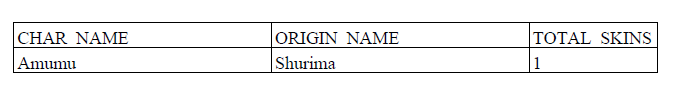
AND Characters.Char\_ID= Origin.CHAR\_ID

AND Characters.Char\_ID= Skins.CHAR\_ID

AND Characters.Char\_ID= voice.CHAR\_ID

AND Voice.Voc\_ID=voice\_actor.VOC\_ID

GROUP BY CHARACTERS.CHAR\_NAME, origin\_places.ORIGIN\_NAME;



1 Remaining

Get the name of all the Frelijord characters and the amount of skins each one has. Cardinality=3

SELECT DISTINCT characters.char\_name, Count(Skins.Char\_ID) AS Amount\_OF\_Skins

FROM CHARACTERS, SKINS, ORIGIN,ORIGIN\_PLACES

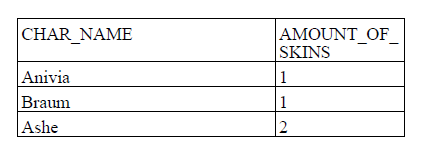
WHERE SKINS.CHAR\_ID = CHARACTERS.CHAR\_ID

AND ORIGIN.CHAR\_ID = CHARACTERS.CHAR\_ID

AND ORIGIN\_PLACES.ORIGIN\_ID = ORIGIN.ORIGIN\_ID

AND ORIGIN\_PLACES.ORIGIN\_NAME= 'Frelijord'

GROUP BY CHARACTERS.CHAR\_NAME;



2 Remaining

Name the characters whose primary roles has more characters than the primary role of marksman. Cardinality=11.

SELECT Characters.Char\_Name

FROM Characters, Roles\_Primary

WHERE

Characters.char\_ID=Roles\_Primary.CHAR\_ID

AND Roles\_Primary.ROLE\_ID

IN(

SELECT ROLES.Role\_ID

FROM Roles,Roles\_Primary

WHERE Roles.ROLE\_ID = Roles\_PRimary.ROLE\_ID

Group by Roles.ROLE\_ID

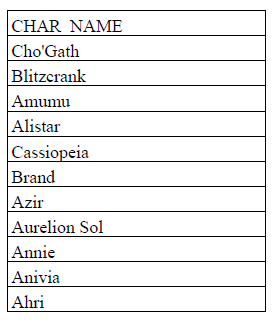
Having Count(Roles\_primary.Char\_ID)>(SELECT Count(Roles\_Primary.ROLE\_ID)

FROM ROLES\_PRIMARY, ROLES

WHERE ROLES.ROLE\_TITLE='Marksman'

AND ROLES.ROLE\_ID =ROLES\_PRIMARY.ROLE\_ID)

);



3 Remaining

Name the Voice actors who voice a character with a play rate that is higher than the average play rate of characters voiced by males. Cardinality=9

SELECT Distinct Voice\_Actor.Voc\_name

FROM VOICE, VOICE\_ACTOR, CHARACTERS

WHERE VOICE\_ACTOR.VOC\_ID = VOICE.VOC\_ID

AND CHARACTERS.CHAR\_ID=VOICE.CHAR\_ID

AND Characters.Char\_perc <

(SELECT AVG(Characters.Char\_perc)

FROM Characters,VOICE\_ACTOR, Voice

Where VOICE\_ACTOR.VOC\_ID = VOICE.VOC\_ID

AND CHARACTERS.CHAR\_ID=VOICE.CHAR\_ID

AND VOICE\_ACTOR.VOC\_GENDER = 'm');

