Irish Dance Database

Final Project

Jordan Murray

Donald Schwartz

CMPT 308N-113

November 29, 2018

Table of Contents

Database Description

The world of Irish dance has many different aspects that need to be organized. The database being modeled is a way to organize many aspects of this sport throughout the United States. Some main entities that are involved are Dancers, Dance Schools, Dance Teachers, Wig Companies, Shoe Companies, Dress Companies, Majors, and Judges.

A dancer has an ID that uniquely identifies it, a name, an age, and a level. There are many aspects that contribute to a dancer's look. Some parts of a dancer's costume are shoes, a dress, and a wig. Dancers acquire these things from shoe companies, dress companies, and wig companies. A dancer can only get their dress from one dress company. This is the same for their shoes and their wig. However, a shoe company, a dress company, and a wig company can provide their services to multiple dancers.

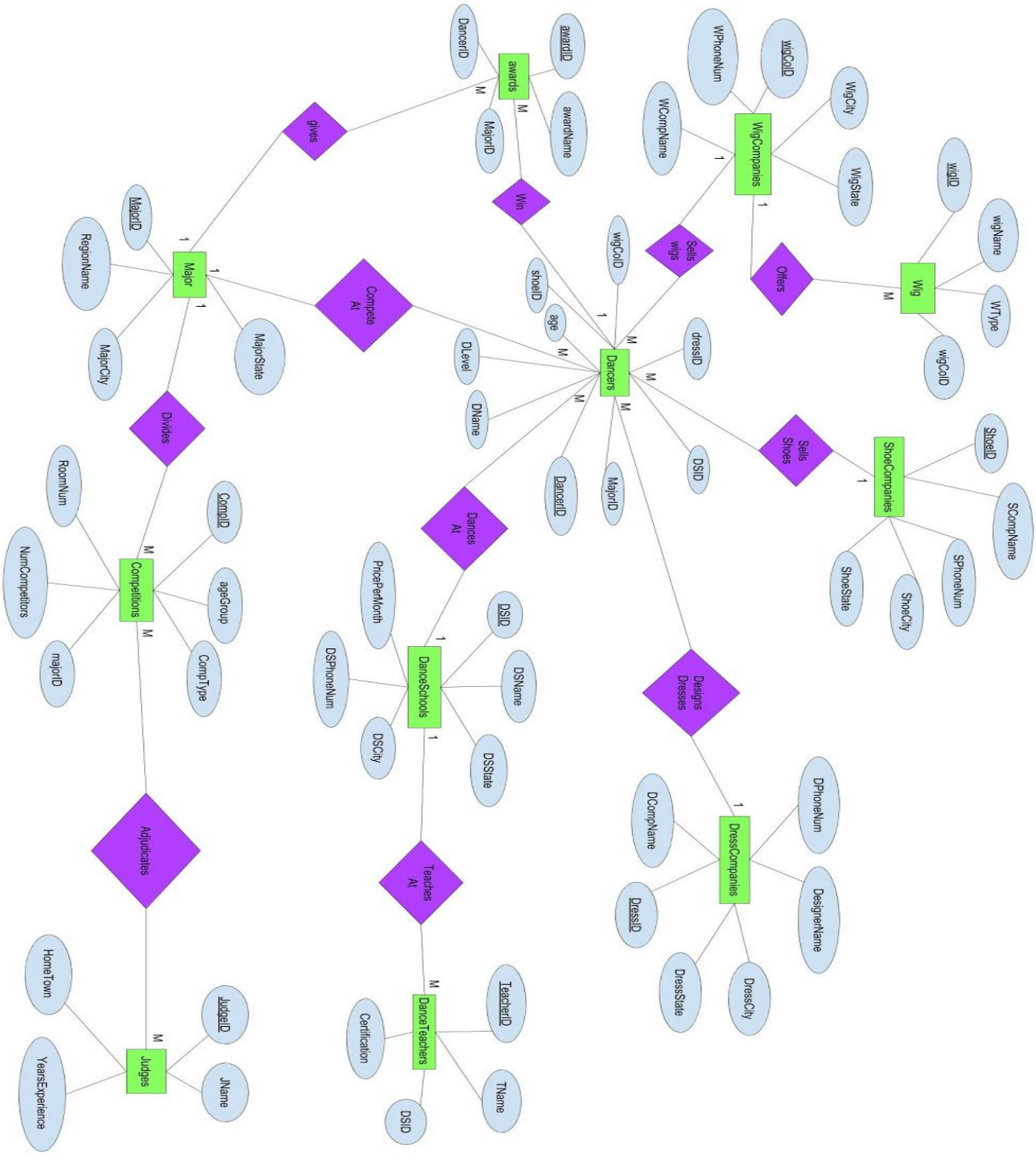
A shoe company has an ID number that uniquely identifies it, a company name, a phone number, a city and state. A dress company has an ID number to uniquely identify it, a company name, a designer name, a phone number, a city and state. A wig Company has the same attributes as a shoe company. There are many different kinds of wigs that a dancer has to choose from. A wig has a unique ID number, a name, a type, and the company it belongs to. A wig company can offer many different kinds of wigs, but a specific wig can only be offered by one company.

A dancer dances at one dance school. A dance school consists of many dancers and has a unique ID number, a name, a city, a state, a phone number, and a price per month. Similarly, a dance teacher teaches at a dance school which can have many different teachers. A dance teacher has an original ID number to identify them, a name, and a certification.

A dancer can win many awards, but an award can only be given to one dancer. An award has a unique ID and a name. Dancers can only compete at one major competitions(regionals) to win awards. So, a major gives many awards, but a specific award can only be given at one major. A major has a special ID number, a region name, a city, and a state. At a major competition, there are many different categories of competitions, but any individual competition can only be at one major. An individual competition has a unique ID, an age group, a room number, a competition type and the number of competitors. Competitions can be adjudicated by many judges and a judge can adjudicate many different competitions. A judge has a unique ID, a name, a hometown, and the number of years they have been judging.

These are some of aspects of the Irish dance that need to be organized. This description gives a brief overview of the database being modeled to show the relationships between these entities and their individual attributes.

ER Diagram


awards
awardID
awardName
DancerID
MajorID
gives
Win
Major
MajorID
RegionName
MajorCity
MajorState
Compete At
Divides
Competitions
CompID
RoomNum
NumCompetitors
ageGroup
majorID
CompType
Adjudicates
Judges
JudgeID
JName
HomeTown
YearsExperience
Dancers
shoelD
age
DLevel
DName
DancerID
MajorID
dressID
DSID
wigColD
Dances At
DanceSchools
DSID
DSName
DSCity
DSState
DSPhoneNum
PricePerMonth
Teaches At
DanceTeachers
TeacherID
TName
DSID
Certification
Sells wigs
WigCompanies
wigColD
WPhoneNum
WCompName
WigCity
WigState
Offers
Wig
wigID
wigName
WType
wigColD
Sells Shoes
ShoeCompanies
ShoeID
SCompName
SPhoneNum
ShoeCity
ShoeState
Designs Dresses
DressCompanies
DressID
DCompName
DPhoneNum
DesignerName
DressCity
DressState

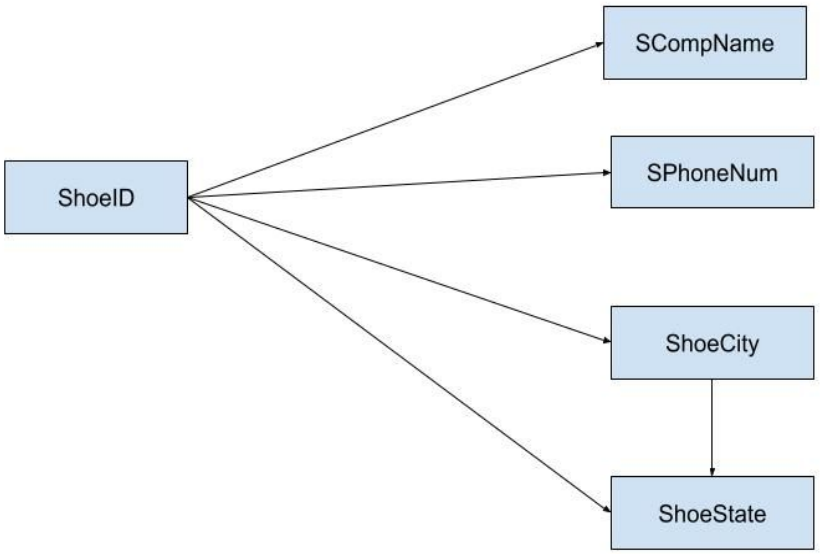
ShoeCompanies Table

CREATE TABLE Shoe Companies(

ShoeID	INT	NOT NULL,
SCompName	NVARCHAR2(20)	NOT NULL,
SPhoneNum	NVARCHAR(12)	NOT NULL,
ShoeCity	NVARCHAR2(15)	NOT NULL,
ShoeState	NVARCHAR2(2)	NOT NULL,

PRIMARY KEY (ShoeID));

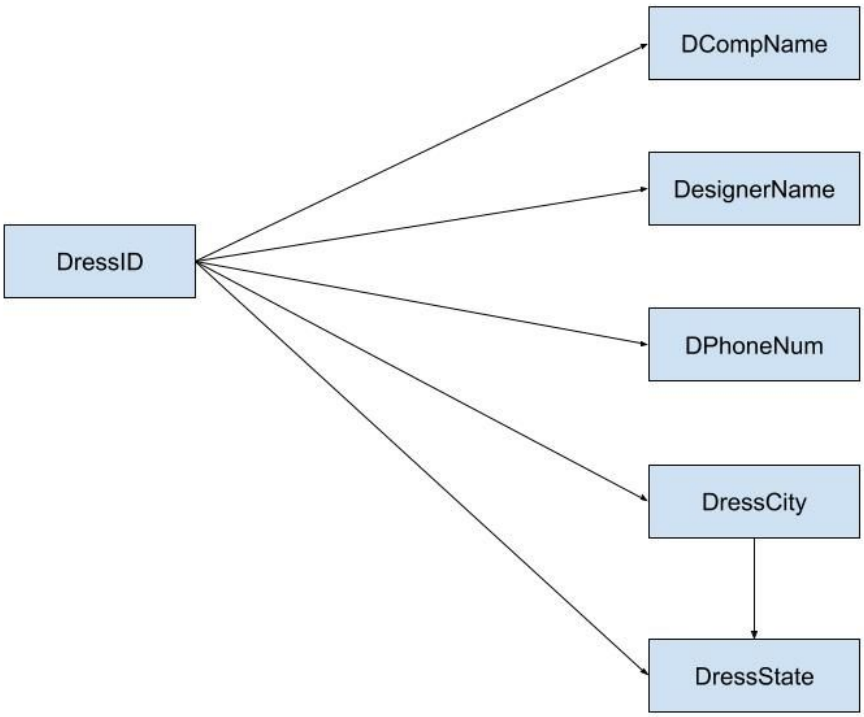
This table lists all information regarding the different kinds of shoe companies that are available to a dancer. It gives a Unique ShoeID to every company. This unique ID determines the SCompName, the SPhoneNum, ShoeCity, and ShoeState. This table is left in second normal form because although ShoeCity does determine ShoeState (transitive dependency), making a separate table for this would slow down the efficiency of the database considering each city and state is unique to each company.



DressCompanies Table

```
CREATE TABLE DressCompanies(
DressID          INT              NOT NULL,
DCompName        NVARCHAR2(20)    NOT NULL,
DesignerName     NVARCHAR2(20)    NOT NULL,
DPhoneNum        NVARCHAR(12)     NOT NULL,
DressCity        NVARCHAR2(15)    NOT NULL,
DressState       NVARCHAR2(2)     NOT NULL,
PRIMARY KEY (DressID));
```

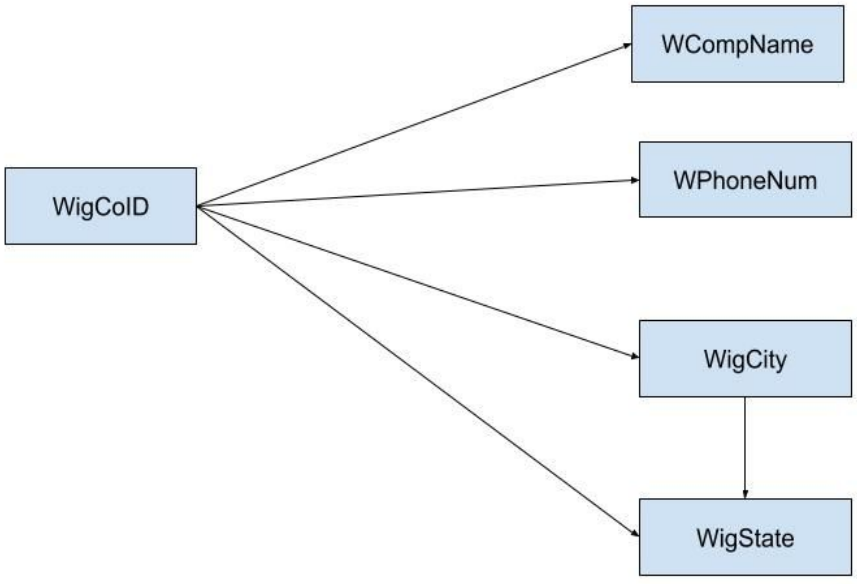
This table lists all information regarding the different kinds of Dress companies that are available to a dancer. It gives a Unique Dress to every company. This unique ID determines the DCompName, the DesigerName the DPhoneNum, DressCity, and Dress State. This table is left in second normal form because although DressCity does determine DressState (transitive dependency), making a separate table for this would slow down the efficiency of the database considering each city and state is unique to each company.



WigCompanies Table

```
CREATE TABLE WigCompanies(
WigCoID          INT              NOT NULL,
WCompName        NVARCHAR2(20)    NOT NULL,
WPhoneNum        NVARCHAR(12)     NOT NULL,
WigCity          NVARCHAR2(15)    NOT NULL,
WigState         NVARCHAR2(2)     NOT NULL,
PRIMARY KEY (WigCoID));
```

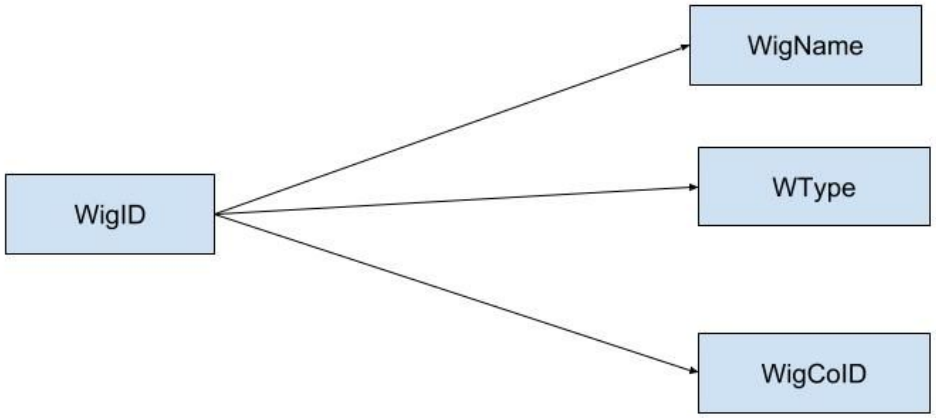
This table lists all information regarding the different kinds of Wig companies that are available to a dancer. It gives a Unique WigCoID to every company. This unique ID determines the WCompName, the WPhoneNum, WigCity, and WigState. This table is left in second normal form because although WigCity does determine WigState (transitive dependency), making a separate table for this would slow down the efficiency of the database considering each city and state is unique to each company.



Wig Table

```
CREATE TABLE Wig(
WigID           INT             NOT NULL,
WigName         NVARCHAR2(20)   NOT NULL,
WType           NVARCHAR2(10)   NOT NULL,
WigCoID         INT,
PRIMARY KEY (WigID),
FOREIGN KEY (WigCoID) REFERENCES WigCompanies(WigCoID));
```

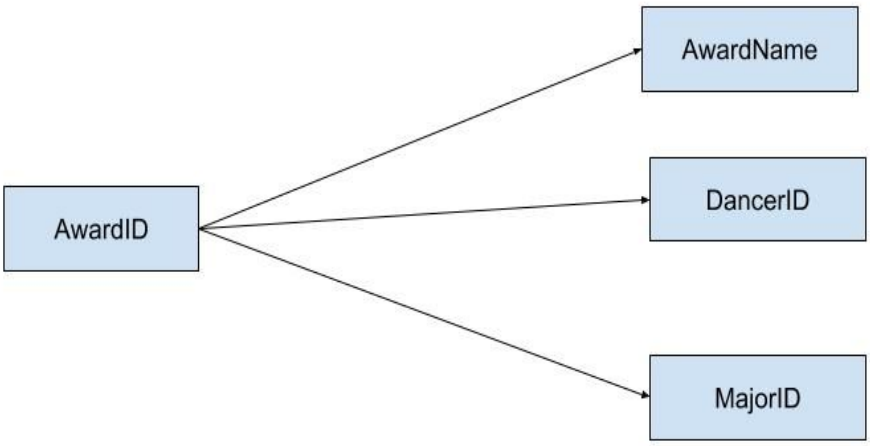
This table gives all details about the different kinds of wigs that a different dance company offers. There is a wigID which determines the WigName, WType, and the WigCoID. WigCoID is a foreign key that references wigCompanies. This table is in third normal form.



Awards Table

```
CREATE TABLE Awards(
AwardID          INT              NOT NULL,
AwardName        NVARCHAR2(20)    NOT NULL,
DancerID         INT,
MajorID          INT,
PRIMARY KEY (AwardID),
FOREIGN KEY (DancerID) REFERENCES Dancers(DancerID),
FOREIGN KEY (MajorID) REFERENCES Major(MajorID));
```

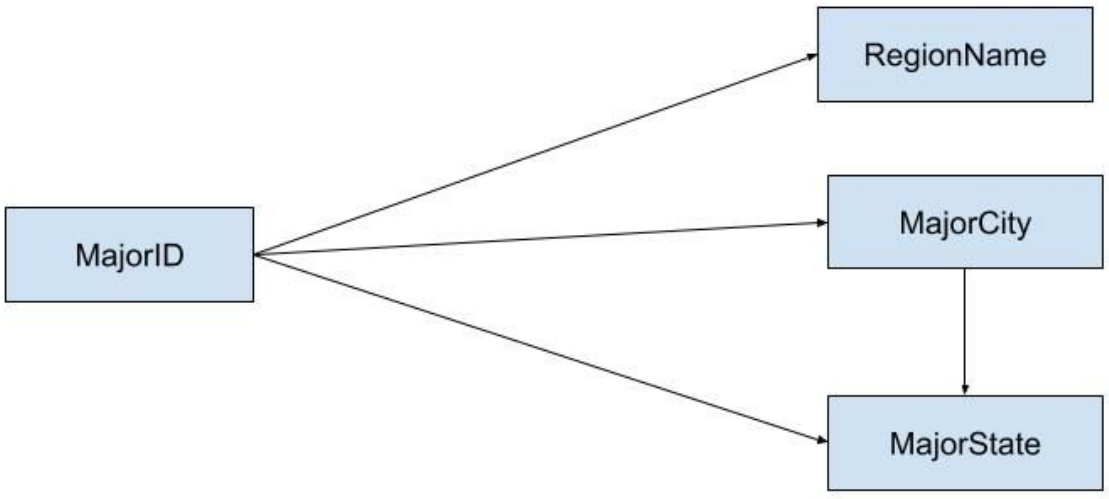
The content of this table describes the different kinds of awards that dancers can win. The AwardID determines the AwardName, the DancerID, and the MajorID. DancerID and MajorID are foreign keys that have been left in this table. DancerID references the Dancers table and MajorID references the Major table. Hence, this table is in third normal form.



Major Table

```
CREATE TABLE Major(
MajorID          INT              NOT NULL,
RegionName       NVARCHAR2(20)    NOT NULL,
MajorCity        NVARCHAR2(15)    NOT NULL,
MajorState       NVARCHAR2(2)     NOT NULL,
PRIMARY KEY (MajorID));
```

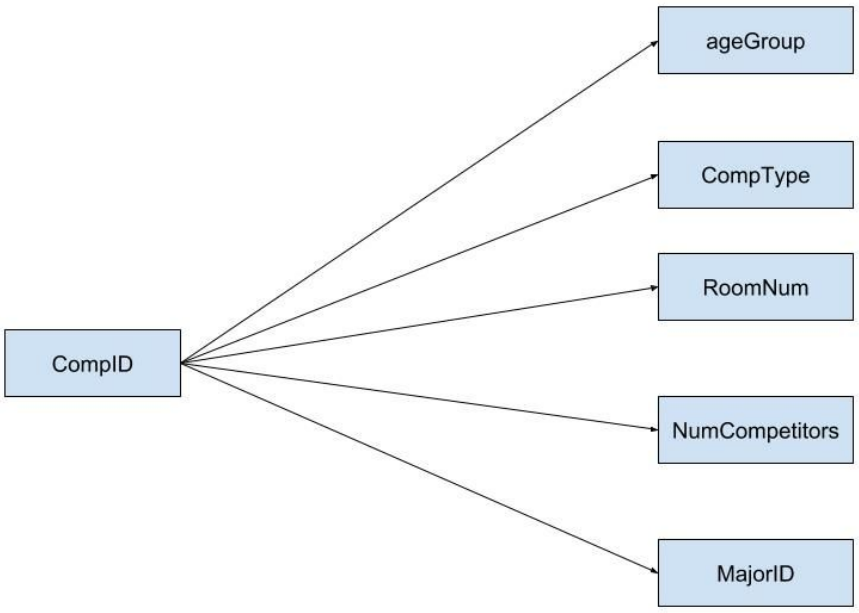
This table gives the different regional competitions that a dancer can dance at. The majorID determines the RegionName, the MajorCity, and the MajorState. This table is left in second normal form because although MajorCity does determine MajorState (transitive dependency), making a separate table for this would slow down the efficiency of the database considering each MajorCity and MajorState is unique to each Major.



Competitions Table

```
CREATE TABLE Competitions(
CompID          INT             NOT NULL,
ageGroup        NVARCHAR2(3)    NOT NULL,
CompType        NVARCHAR2(15)   NOT NULL,
RoomNum         INT             NOT NULL,
NumCompetitors  INT             NOT NULL,
MajorID         INT,
PRIMARY KEY (CompID),
FOREIGN KEY (MajorID) REFERENCES Major(MajorID));
```

The Competitions table explains the different categories of competitions that are at each major. The CompID determines the ageGroup, CompType, RoomNum, NumCompetitors, and MajorID. This table is in third normal form. It has one foreign key MajorID which references the Major table.



Adjudicated Table

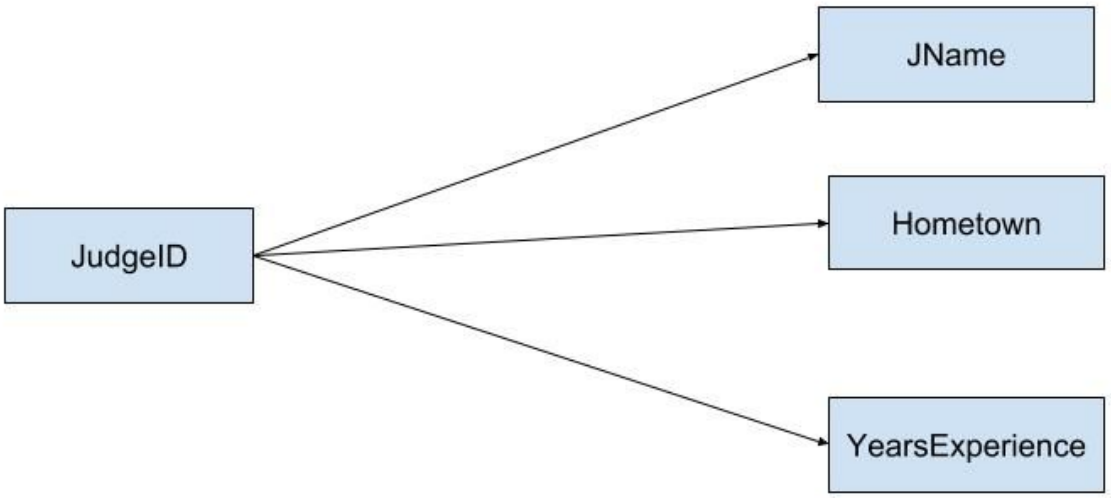
```
CREATE TABLE Adjudicated(
JudgeID          INT                    NOT NULL,
CompID           INT                    NOT NULL,
PRIMARY KEY (JudgeID, CompID),
FOREIGN KEY (JudgeID) REFERENCES Judges (JudgeID),
FOREIGN KEY (CompID) REFERENCES Competitions (CompID));
```

This table represents the relationship between a competition and a judge. A competition can be adjudicated by many different judges and a judge can adjudicate multiple competitions. This table is automatically in third normal form because it does not have any determinants. Together the JudgeID and the CompID make up the primary key. Each ID separately is a foreign key that references its corresponding table.

Judges Table

```
CREATE TABLE Judges(
JudgeID          INT              NOT NULL,
JName            NVARCHAR2(20)    NOT NULL,
Hometown         NVARCHAR2(15)    NOT NULL,
YearsExperience  INT              NOT NULL,
PRIMARY KEY (JudgeID));
```

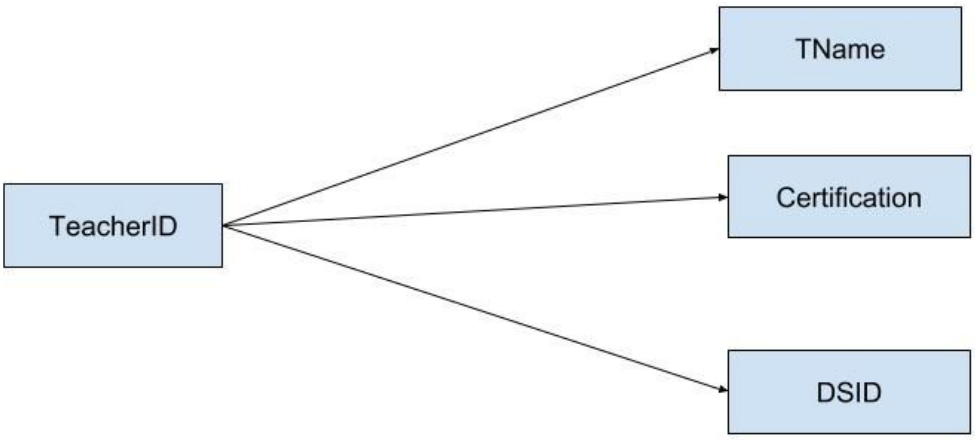
This table describes information about the judges who adjudicate the competitions. The unique JudgeID determines the JName, their Hometown, and their yearsExperience. Therefore this table is in third normal form.



DanceTeachers Table

```
CREATE TABLE DanceTeachers(
TeacherID        INT              NOT NULL,
TName            NVARCHAR2(30)    NOT NULL,
Certification    NVARCHAR2(6)     NOT NULL,
DSID             INT,
PRIMARY KEY (TeacherID),
FOREIGN KEY (DSID) REFERENCES DanceSchools(DSID));
```

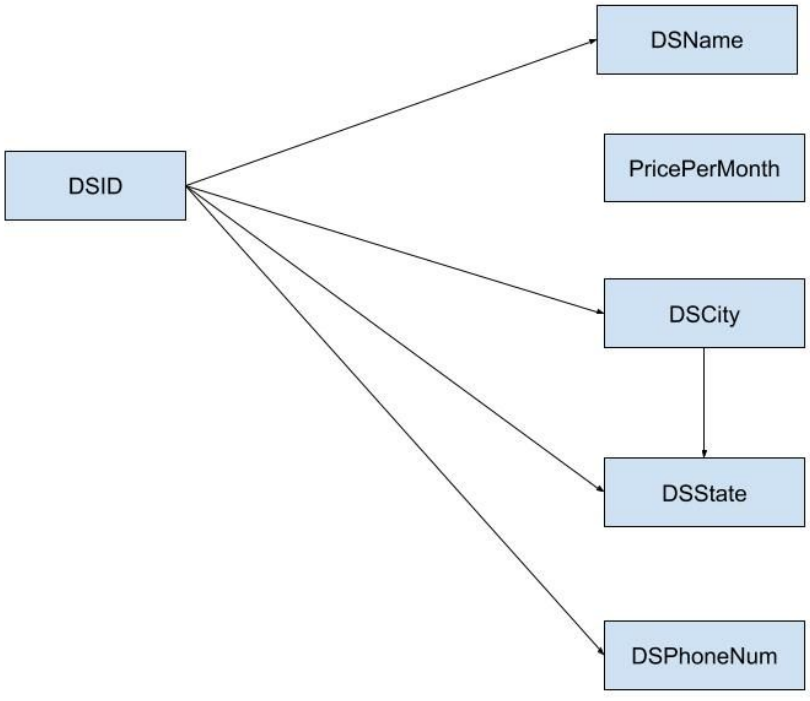
This table gives information about dance teachers. The TeacherID determines the TName, certification, and DSID. DSID is a foreign key that references the DanceSchools table. Hence, this table is in third normal form.



DanceSchools Table

```
CREATE TABLE DanceSchools(
DSID            INT             NOT NULL,
DSName          NVARCHAR2(20)   NOT NULL,
PricePerMonth   INT             NOT NULL,
DSCity          NVARCHAR2(15)   NOT NULL,
DSState         NVARCHAR2(2)    NOT NULL,
DSPhoneNum      NVARCHAR2(12)   NOT NULL,
PRIMARY KEY (DSID));
```

This table lists dance schools around the country and all of the information associated with it. The DSID determines the DSName, PricePerMonth, DSCity, DSState, and DSPhoneNum. This table is left in second normal form because although DSCity does determine DSState (transitive dependency), making a separate table for this would slow down the efficiency of the database considering each DSCity and DSState is unique to each Dance School.

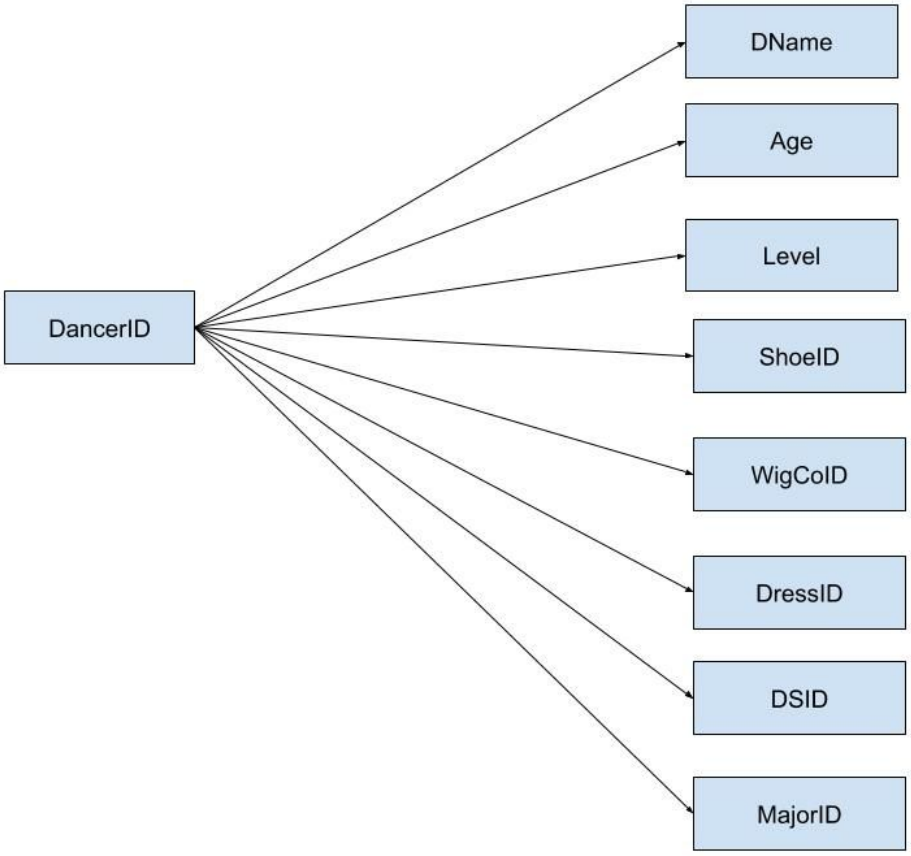


Dancers Table

```
CREATE TABLE Dancers(
DancerID        INT             NOT NULL,
DName           NVARCHAR2(30)   NOT NULL,
Age             INT             NOT NULL,
DLevel          NVARCHAR2(20)   NOT NULL,
ShoeID          INT,
WigCoID         INT,
DressID         INT,
DSID            INT,
MajorID         INT,
PRIMARY KEY (DancerID),
FOREIGN KEY (ShoeID) REFERENCES ShoeCompanies(ShoeID),
FOREIGN KEY (WigCoID) REFERENCES WigCompanies(WigCoID),
FOREIGN KEY (DressID) REFERENCES DressCompanies(DressID),
FOREIGN KEY (DSID) REFERENCES DanceSchools(DSID),
FOREIGN KEY (MajorID) REFERENCES Major(MajorID));
```

This table gives all information about a dancer. The dancerID determines the DName, Age, Level, ShoeID, wigCoID, DressID, DSID, and MajorID. This table has five foreign keys: ShoeID, wigCoID, DressID, DSID, and MajorID. ShoeID references the ShoeCompanies table, wigCoID references the wigCompanies table, DressID references the DressCompanies table, DSID references the DanceSchools table, and MajorID references the MajorID table. These foreign

keys have been left in this table and the transitive and partial dependencies have been removed. Hence, this table is in third normal form.



Universal Quantifier

Name the judges who judge all of the competitions

```
SELECT Judges.JName
FROM Judges
WHERE NOT EXISTS
  (SELECT *
   FROM Competitions
   WHERE NOT EXISTS
         (SELECT *
         FROM Adjudicated
         WHERE Judges.JudgeID = Adjudicated.JudgeID
         AND Competitions.CompID = Adjudicated.CompID));
```

JNAME
John O'Keefe

Cardinality: 1

Only

Name dancers, their age and their level who are only taught by Michael Farrell.

```
SELECT Dancers.DName, Dancers.Age, Dancers.DLevel
FROM Dancers
WHERE Dancers.DSID NOT IN
  (SELECT DanceSchools.DSID
   FROM DanceSchools
   WHERE DanceSchools.DSID NOT IN
         (SELECT DanceTeachers.DSID
         FROM DanceTeachers
         WHERE DanceTeachers.TName = 'Michael Farrell'));
```

DNAME	AGE	DLEVEL
Dale Kane	18	Prelim Champion
Maura Sabini	12	Prizewinner
Jenna Murray	15	Prizewinner

Cardinality: 3

None

Name the dress companies that designed a dress for a dancer who has won no awards.

```
SELECT DressCompanies.DCompName
FROM DressCompanies
WHERE DressCompanies.DressID NOT IN
  (SELECT Dancers.DressID
   FROM Dancers
   WHERE Dancers.DancerID NOT IN
         (SELECT awards.DancerID
         FROM awards));
```

DCOMPNAME
Eire Designs

Cardinality: 1

Right Join:

Name all dancers and their awards if they won any

SELECT Dancers.DName, awards.awardName

FROM awards RIGHT JOIN Dancers ON Dancers.DancerID = awards.DancerID;

DNAME	AWARDNAME
Julia O'Rourke	1st
Owen Lubers	1st
Owen Lubers	3rd
Simone Adele	2nd
Jenna Murray	
Grace Duncan	
Melanie Valdez	2nd
Meaghan Houlihan	1st
Ashley Harten	
Samantha Brewster	
Curtis Long	2nd
Olivia Murray	2nd
Gianna Cheeseman	
Maura Sabini	
Dale Kane	3rd
Fiona Dargan	1st
Fiona Dargan	3rd
Mitchell Hodges	1st

Mitchell Hodges	2nd
Mitchell Hodges	3rd
Julia Marino	
Brogan McCay	3rd
Cyra Taylor	
Jess Hindley	

Cardinality: 24

Left Join

Name all dance teachers, the school they teach at and their certification if they have one

SELECT DanceTeachers.Certification, DanceTeachers.TName, DanceSchools.DSName FROM DanceTeachers LEFT JOIN DanceSchools ON DanceSchools.DSID = DanceTeachers.DSID;

CERTIFICATION	TNAME	DSNAME
	Ashley Gilnack	Farrell School
TCRG	Michael Farrell	Farrell School
TCRG	Karen Petri	Doherty-Petri
TCRG	Lisa Petri	Doherty-Petri
	Caitrin O'Meara	Broesler
	Eileen Paulson	Broesler
TCRG	Kevin Broesler	Broesler
	Erin Collins	Lenihan
TCRG	Meghan Lenihan	Lenihan
TCRG	Patty Lenihan	Lenihan
TCRG	Geraldine Murray	Murray Academy
TCRG	Patrick Campbell	Brady Campbell
TCRG	Rebecca Brady-Campbell	Brady Campbell
	Natalie Findling	Burke Connolly
	Molly Gareau	Burke Connolly
TCRG	Emma Burke	Burke Connolly
TCRG	Erin Connelly	Burke Connolly

	Briley Mastis	Clark Academy
	Olivia Smugala	Clark Academy
ADCRG	Mar Jo Clark Cange	Clark Academy
ADCRG	Alisa Dosch	Clan Rince
ADCRG	Jeannie Thornton	Clan Rince
ADCRG	Deirdre O'Sullivan-Toolan	O'sullivan Academy
TCRG	Karen Petri	O'sullivan Academy
TCRG	Theresa O'Sullivan-Randall	O'sullivan Academy

Cardinality: 25

Full Join

Name all dancers, the major they competed at if they did and the award they won if they won an award.

SELECT Dancers.DName, Major.RegionName, awards.awardName

FROM Dancers FULL JOIN awards ON Dancers.DancerID = awards.DancerID

FULL JOIN Major ON Major.MajorID = Dancers.MajorID;

DNAME	REGIONNAME	AWARDNAME
Julia O'Rourke	Mid-Atlantic	1st
Owen Lubers	New England	3rd
Owen Lubers	New England	1st
Simone Adele	New England	2nd
Jenna Murray	Mid-Atlantic	
Grace Duncan		
Melanie Valdez	Mid-Atlantic	2nd
Meaghan Houlihan	Western	1st
Ashley Harten		
Samantha Brewster		
Curtis Long	Southern	2nd
Olivia Murray	Western	2nd
Gianna Cheeseman	Mid-Atlantic	
Maura Sabini	Mid-Atlantic	
Dale Kane	Mid-Atlantic	3rd
Fiona Dargan	Southern	3rd

Fiona Dargan	Southern	1st
Mitchell Hodges	Mid-America	3rd
Mitchell Hodges	Mid-America	2nd
Mitchell Hodges	Mid-America	1st
Julia Marino	Western	
Brogan McCay	Western	3rd
Cyra Taylor		
Jess Hindley		

Cardinality: 24

6 Tables

Name dancers if they have the following: the name of their shoe Company, dress company, wig company, dance school, the awards they have won, and the region they dance at for majors.

```
SELECT Dancers.DName, SC.SCompName, DC.DCompName, WC.WCompName,
DS.DSName, awards.awardName, Major.RegionName
FROM Dancers, ShoeCompanies SC, DressCompanies DC, WigCompanies WC,
DanceSchools DS, awards, Major
WHERE Dancers.ShoeID = SC.ShoeID
AND Dancers.DressID = DC.DressID
AND Dancers.WigCoID = WC.WigCoID
AND Dancers.DSID = DS.DSID
AND Dancers.DancerID = awards.DancerID
AND Dancers.MajorID = Major.MajorID;
```

DNAME	SCOMPNAME	DCOMPNAME	WCOMPNAME	DSNAME	AWARDNAME	REGIONNAME
Julia O'Rourke	Fays	PrimeDress Designs	Celtic Curls	Doherty-Petri	1st	Mid-Atlantic
Melanie Valdez	Feis Fayre	Elevation	Camelia Rose	Doherty-Petri	2nd	Mid-Atlantic
Dale Kane	Fays	Eire Designs	Celtic Curls	Farrell School	3rd	Mid-Atlantic
Simone Adele	Ryan & O'Donnell	Lewis	Camelia Rose	Broesler	2nd	New England
Meaghan Houlihan	Fays	Eire Designs	Emerald Key	Murray Academy	1st	Western
Olivia Murray	Fays	Rising Star	Celtic Curls	Clan Rince	2nd	Western

Brogan McCay	Fays	Rising Star	Celtic Curls	Clan Rince	3rd	Western
Fiona Dargan	Feis Fayre	PrimeDress Designs	Camelia Rose	Burke Connolly	1st	Southern
Fiona Dargan	Feis Fayre	PrimeDress Designs	Camelia Rose	Burke Connolly	3rd	Southern

Cardinality: 9

Query1

Get the judge ID numbers of judges who have a lower number years of experience than Ryan Carroll

```
SELECT Judges2.JudgeID

FROM Judges Judges1, Judges Judges2

WHERE Judges1.JName = 'Ryan Carroll'

AND Judges1.YearsExperience > Judges2.YearsExperience;
```

JUDGEID
901
902
904
906
907
908

Cardinality: 6

Query2

Select the dancer ID number for dancers who danced at a major that had a competition judged by a judge from Dublin

```
SELECT DISTINCT Dancers.DancerID

FROM Dancers, Major, Competitions, Judges, Adjudicated

WHERE Dancers.MajorID = Major.MajorID

AND Competitions.MajorID = Major.MajorID

AND Competitions.CompID = Adjudicated.CompID

AND Adjudicated.JudgeID = Judges.JudgeID

AND Judges.Hometown = 'Dublin';
```

DANCERID
110
114
115
113
104
117
118
112
101
106
107
111

Cardinality: 1

Query3

Get the Dancers Name, the Wig Name about a wig that belongs to Camelia Rose company that offers a bun wig to dancers who are the level Open Champion

```
SELECT Dancers.DName, Wig.WigName, Wig.Wtype, WigCompanies.WCompName
FROM Wig, WigCompanies, Dancers
WHERE WigCompanies.WigCoID = Wig.WigCoID
AND WigCompanies.WigCoID = Dancers.WigCoID
AND Dancers.DLevel = 'Open Champion'
AND Wig.WType = 'Bun'
AND WigCompanies.WCompName = 'Camelia Rose'
ORDER BY(Dancers.DName);
```

DNAME	WIGNAME	WTYPE	WCOMPNAME
Fiona Dargan	Double Lucy	Bun	Camelia Rose
Fiona Dargan	Alliyah	Bun	Camelia Rose
Gianna Cheeseman	Double Lucy	Bun	Camelia Rose
Gianna Cheeseman	Alliyah	Bun	Camelia Rose
Melanie Valdez	Alliyah	Bun	Camelia Rose
Melanie Valdez	Double Lucy	Bun	Camelia Rose

Cardinality: 6