

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light greenish-blue. They are positioned diagonally, with the blue one partially covering the green one.

The Feis Database

Declan Byrne



Table of Contents

Executive Summary...3

ER Diagram...4

Tables...5

Views...18

Queries...20

Stored Procedures...22

Security Roles...26

Known Problems/Future Enhancements...28



Executive Summary

A Feis (pronounced Fesh, plural Feisanna) is an Irish Arts festival. Most modern feisanna are focused heavily on Irish Step Dancing. Since it is such an old tradition, some of the technology is outdated. Many dancers compete at different levels of competition, which are called Beginner, Prizewinner, Preliminary, and Open. Also, there are different types of dances that you can compete in, which change depending on what level you are. We need to record different feisanna and their competitions, who judged the competitions, who played the music, the results, competitors and their school info.

ER Diagram

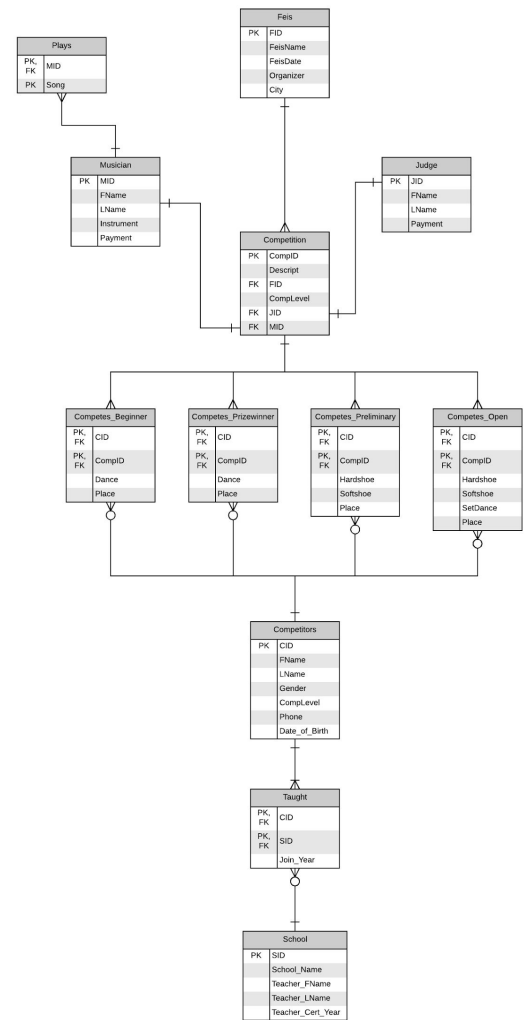




Table Create Statements

Feis Table

Data Output

Explain

Messages

History

	fid character varying(4)	feisname character(20)	feisdate date	organizer character(30)	city character(10)
1	f001	Inis Fada	2017-09-15	Donny Golden	Mineola
2	f002	Big Apple	2017-10-02	Irish American Society	Manhattan
3	f003	AOH Feis	2017-11-27	Ancient Order of Hibernians	Long Beach

Feis is the table that contains the data for the different feiseanna. Feiseanna have ids, and names, usually either named for the location they're held in (Inis Fada is the Gaelic term for Long Island, and the main feis in Manhattan is called the Big Apple Feis), or the organization that runs it (the AOH is short for the Ancient Order of Hibernians, a major Irish American group). They also have a group or person who organizes the event, and the city that the feis is held in.

```
CREATE TABLE Feis (
```

```
    FID          varchar(4),
    FeisName     char(20) not null,
    FeisDate     Date not null,
    Organizer    char(30) not null,
    City         char(10) not null,
```

```
    primary key(FID)
```

```
);
```

Dependencies:

FID → FeisName, FeisDate, Organizer, City

Competition Table

This is a table for the individual competitions that are in a feis. They have an id, and a description. There are different competitive levels. From lowest to highest, they are: Beginner, Prizewinner, Preliminary, and Open. The main differences in these levels are the way that competitions are split up within that level (Beginner and Prizewinner are split up by each dance type, whereas Preliminary and Open are split up by age), as well as the types of dances that the competitors are allowed/required to do. They are also identified by what feis they are a part of, who judges them, and who plays the music.

```
CREATE TABLE Competition (  
    CompID          integer,  
    Descript        varchar(25) not null,  
    FID             varchar(4) not null references Feis(FID),  
    JID             varchar(4) not null references Judge(JID),  
    MID             varchar(4) not null references Musician(MID),  
    primary key(CompID)  
);
```

	compid integer	descript character varying(25)	fid character varying(4)	jid character varying(4)	mid character varying(4)
1	100	Beginner Reel	f001	j001	m004
2	101	Prizewinner Treble Jig	f001	j002	m005
3	102	Prizewinner Hornpipe	f001	j001	m003
4	103	Prizewinner Slip Jig	f001	j003	m004
5	104	Preliminary Mens	f001	j005	m002
6	105	Preliminary Womens	f001	j003	m004
7	106	Open Mens	f001	j005	m004
8	107	Open Womens	f001	j002	m002
9	108	Beginner Light Jig	f002	j004	m001
10	109	Beginner Slip Jig	f002	j004	m005
11	110	Beginner Reel	f002	j006	m001
12	111	Preliminary Mens	f002	j001	m005
13	112	Preliminary Womens	f002	j002	m002
14	113	Prizewinnner Light Jig	f003	j006	m003
15	114	Prizewinner Reel	f003	j005	m001
16	115	Prizewinner Treble Jig	f003	j003	m002
17	116	Prizewinner Hornpipe	f003	j004	m005
18	117	Open Mens	f003	j004	m002
19	118	Open Womens	f003	j001	m004

Dependencies:
CompID → Descript, FID, JID, MID



Judge Table

These are the people who judge the competitions and decide who wins, and are certified by the World Irish Dance Association. Judges have an ID, a first and last name, and a payment that they receive for each feis that they participate in

	jid character varying(4)	fname character(12)	lname character varying(15)	payment money
1	j001	Kevin	Dobrowolski	\$500.00
2	j002	Liam	Ollive	\$350.00
3	j003	Nick	Lange	\$400.00
4	j004	Matt	Batory	\$400.00
5	j005	Liam	Cullen	\$550.00
6	j006	Ryan	Parbs	\$350.00

```
CREATE TABLE Judge (  
    JID          varchar(4),  
    FName        char(12) not null,  
    LName        varchar(15) not null,  
    Payment      Money not null,  
    primary key(JID)  
);
```

Dependencies:
JID → FName, LName, Payment



Musician Table

These are the people who play the music that the dancers dance to. Like Judges, they also have an id, first and last name, and how much they're paid for the feis

```
CREATE TABLE Musician (  
    MID          varchar(4),  
    FName        char(12) not null,  
    LName        varchar(15) not null,  
    Instrument    char(12) not null,  
    Payment       Money not null,  
    primary key(MID)  
);
```

	mid character varying(4)	fname character(12)	lname character varying(15)	instrument character(12)	payment money
1	m001	Daniel	Squicciarino	Tin Whistle	\$500.00
2	m002	Alex	Sarbin	Accordion	\$400.00
3	m003	Mike	Yelovitch	Fiddle	\$450.00
4	m004	Kenny	Marples	Keyboard	\$500.00
5	m005	Elena	Pileh	Keyboard	\$450.00

Dependencies:

MID → FName, LName, Instrument, Payment



Plays Table

This is a list of each musician and the songs that they can play, which is very important to know so that you can delegate who can play what competitions based on the dances in those competitions.

```
CREATE TABLE Plays (  
    MID          varchar(4) references Musician(MID),  
    Song         char(17) not null,  
    primary key(MID, Song)  
);
```

	mid character varying(4)	song character(17)
1	m001	Slip Jig
2	m001	Light Jig
3	m001	Reel
4	m001	Treble Jig
5	m002	Slip Jig
6	m002	Light Jig
7	m002	Reel
8	m002	Hornpipe
9	m002	Treble Jig
10	m003	Slip Jig
11	m003	Hornpipe
12	m004	Slip Jig
13	m004	Reel
14	m004	Hornpipe
15	m004	Treble Jig
16	m005	Treble Jig
17	m005	Reel

Competes_Beginner Table

This is the table for the beginner competitions and the competitors. In Beginner, the dance can be Light_Reel, Slip_Jig, and Reel. It also contains the place for each competitor in the competition. Competitions are broken up by the type of dance.

```
CREATE TABLE Competes_Beginner (  
    CID                varchar(4) references Competitor(CID),  
    CompID             integer references  
Competition(CompID),  
    Dance              char(10) not null CHECK( Dance IN ('Reel',  
'Light_Jig', 'Slip_Jig')),  
    Place              integer not null,  
    primary key(CID, CompID)  
);
```

	cid character varying(4)	compid integer	dance character(10)	place integer
1	c009	100	Reel	1
2	c007	100	Reel	2
3	c002	100	Reel	3
4	c014	108	Light Jig	1
5	c009	108	Light Jig	2
6	c002	108	Light Jig	3
7	c009	109	Slip Jig	1
8	c014	109	Slip Jig	2
9	c002	109	Slip Jig	3
10	c007	110	Reel	1
11	c009	110	Reel	2
12	c014	110	Reel	3
13	c002	110	Reel	4

Dependencies:
CompID → Dance
CID, CompID → Place

Competes_Prizewinner Table

This is the table for prizewinner competitions and their competitors. In this level, dance can be Slip_Jig, Reel, Light_Jig, Hornpipe, and Treble_Jig. It also contains the place each competitor receives, and competitions are broken up by the type of dance.

```
CREATE TABLE Competes_Beginner (  
    CID                varchar(4) references Competitor(CID),  
    CompID             integer references  
    Competition(CompID),  
    Dance              char(10) not null CHECK( Dance IN ('Reel',  
'Light_Jig', 'Slip_Jig', 'Treble_Jig', 'Hornpipe')),  
    Place              integer not null,  
    primary key(CID, CompID)  
);
```

	cid character varying(4)	compid integer	dance character(10)	place integer
1	c008	101	Treble Jig	1
2	c013	101	Treble Jig	2
3	c005	101	Treble Jig	3
4	c012	101	Treble Jig	4
5	c013	102	Hornpipe	1
6	c005	102	Hornpipe	2
7	c012	102	Hornpipe	3
8	c012	103	Slip Jig	1
9	c008	103	Slip Jig	2
10	c013	103	Slip Jig	3
11	c012	113	Light Jig	1
12	c008	113	Light Jig	2
13	c005	114	Reel	1
14	c012	114	Reel	2
15	c008	114	Reel	3
16	c013	114	Reel	4
17	c013	115	Treble Jig	1
18	c005	115	Treble Jig	2
19	c013	116	Hornpipe	1
20	c008	116	Hornpipe	2
21	c005	116	Hornpipe	3
22	c012	116	Hornpipe	4

Dependencies:
CompID → Dance
CID, CompID → Place

Competes_Preliminary Table

This is the table for Preliminary competitions and their competitors. In Preliminary, competitors are required to do a softshoe dance, which can be Slip_Jig or Reel (Light_Jig is a softshoe dance, but only at lower levels), as well as a hardshoe dance, which can be Treble_Jig or Hornpipe. It also contains the place that each competitor receives, and competitions are broken up by gender

	cid character varying(4)	compid integer	hardshoe character(10)	softshoe character(8)	place integer
1	c001	104	Hornpipe	Reel	1
2	c003	104	Treble Jig	Reel	2
3	c006	104	Hornpipe	Reel	3
4	c010	105	Hornpipe	Slip Jig	1
5	c017	105	Treble Jig	Reel	2
6	c006	111	Hornpipe	Reel	1
7	c001	111	Treble Jig	Reel	2
8	c016	112	Treble Jig	Slip Jig	1
9	c010	112	Hornpipe	Reel	2

```
CREATE TABLE Competes_Preliminary (  
    CID                varchar(4) references Competitor(CID),  
    CompID             integer references  
Competition(CompID),  
    Hardshoe           char(10) not null CHECK(Hardshoe IN  
( 'Treble_Jig', 'Hornpipe' )),  
    Softshoe           char(8) not null CHECK(Softshoe IN ('Slip_Jig',  
'Reel' )),  
    Place              integer not null,  
    primary key(CID, CompID)  
);
```

Dependencies:
CID, CompID → Hardshoe, Softshoe, Place

Competes_Open Table

	cid character varying(4)	compid integer	hardshoe character(10)	softshoe character(8)	setdance character(37)	place integer
1	c011	106	Treble Jig	Reel	Three Sea Captains	1
2	c018	107	Treble Jig	Slip Jig	The Wandering Musician	1
3	c004	107	Hornpipe	Reel	The Blackthorn Stick	2
4	c015	117	Hornpipe	Reel	Jockey to the Fair	1
5	c011	117	Treble Jig	Reel	Madame Bonaparte	2
6	c018	118	Hornpipe	Slip Jig	The Downfall of Paris	1
7	c004	118	Treble Jig	Slip Jig	King of the Fairies	2

This is the table for Open competitions and their competitors. Like Preliminary, competitors in Open are required to do a hardshoe and softshoe dance, as well as what is called a set dance. Sets are specific routines (usually either Treble_Jig or Hornpipe) that are the same across all schools. It also contains the place for each competitor, and competitions are broken up by gender.

```
CREATE TABLE Competes_Open (  
    CID                varchar(4) references Competitor(CID),  
    CompID             integer references  
Competition(CompID),  
    Hardshoe           char(10) not null CHECK(Hardshoe IN  
( 'Treble_Jig', 'Hornpipe' )),  
    Softshoe           char(8) not null CHECK(Softshoe IN ( 'Slip_Jig',  
'Reel' )),  
    SetDance           char(37) not null,  
    Place              integer not null,  
    primary key(CID, CompID)  
);
```

Dependencies:

CID, CompID → Hardshoe, Softshoe, SetDance, Place

Competitor Table

This is the table for all of those who compete in the feiseanna. Competitors have an id, first and last name, a gender, a competitive level, a phone number for contact, and a date of birth.

```
CREATE TABLE Competitor (  
    CID                varchar(4),  
    FName              char(12) not null,  
    LName              varchar(15) not null,  
    Gender             char(1) not null,  
    CompLevel          char(17) CHECK(CompLevel IN  
( 'Beginner', 'Prizewinner', 'Preliminary', 'Open')) not null,  
    Phone              varchar(10) not null,  
    Date_of_Birth      date not null,  
    primary key(CID)  
);
```

Dependencies:

CID → FName, LName, Gender, CompLevel, Phone, Date_of_Birth

	cid character varying(4)	fname character(12)	lname character varying(15)	gender character(1)	complevel character(17)	phone character varying(10)	date_of_birth date
1	c001	Declan	Byrne	M	Preliminary	6687325686	1997-11-19
2	c002	Alan	Labouseur	M	Beginner	2232832623	1968-03-17
3	c003	Peter	Hannon	M	Preliminary	6976666283	1999-08-14
4	c004	Brianna	D'Amico	F	Open	8434255929	1996-11-28
5	c005	Christina	Harrick	F	Prizewinner	8726733728	1997-11-06
6	c006	Dylan	O'Brien	M	Preliminary	8463243251	1998-12-13
7	c007	Tom	Vestuto	M	Beginner	8463243251	1999-06-22
8	c008	Katie	Kilcullen	F	Prizewinner	212735521	1996-09-03
9	c009	Anna	Wilson	F	Beginner	1373746261	1994-05-06
10	c010	Mary	Pulizzotto	F	Preliminary	1658945612	1997-05-03
11	c011	Kyle	Meit	M	Open	4868464866	1995-03-21
12	c012	Hayley	Petrizzi	F	Prizewinner	1686549732	1999-06-21
13	c013	Jean	Abecassis	M	Prizewinner	5948626573	1994-03-16
14	c014	Patrick	Lombardi	M	Beginner	6845257954	1994-02-19
15	c015	Charles	Beers	M	Open	6859432495	1996-03-12
16	c016	Jill	Craig	F	Preliminary	6584326879	1998-02-26
17	c017	Megan	Le	F	Preliminary	4625978164	1997-05-19
18	c018	Katarina	Mueller	F	Open	6435942801	1996-08-15



Taught Table

This is the table to show what Irish Step Dance School that the competitor has been a part of. Competitors can be independant, but more often than not competitors are part of a school. Also, competitors can change schools. This table contains a competitor and school id, and the year that the competitor joined that school.

```
CREATE TABLE Taught (  
    CID          varchar(4) references Competitor(CID),  
    SID          varchar(4) references School(SID),  
    Join_Year    int not null,  
    primary key(CID, SID)  
);
```

	cid character varying(4)	sid character varying(4)	join_year integer
1	c001	s01	2003
2	c003	s01	2006
3	c003	s04	2009
4	c004	s05	2005
5	c006	s02	2014
6	c006	s01	2016
7	c007	s03	2004
8	c008	s05	2000
9	c009	s02	2015
10	c009	s01	2007
11	c010	s05	2002
12	c011	s04	2011
13	c012	s01	2013
14	c013	s03	2016
15	c014	s05	2007
16	c014	s02	2014
17	c015	s04	2013
18	c016	s03	2008
19	c017	s05	2006
20	c018	s03	2005
21	c018	s04	2010

Dependencies:
CID, SID → Join_Year



School Table

Irish dance schools are the organizations that are led by a teacher certified by the World Irish Dance Association. They have a school id, a school name (which more often than not is named after the founder of the school), the name of the teacher, and the date that the teacher was certified.

	sid character varying(4)	school_name character(30)	teacher_fname character(12)	teacher_lname character varying(15)	teacher_cert_year integer
1	s01	Donny Golden	Donny	Golden	1963
2	s02	Mulvihill	Michelle	Byrne	2012
3	s03	Doherty Petri	Melanie	Deegan	2003
4	s04	Schade Academy	Mary Lou	Schade	1997
5	s05	Hagen	Claire	O'Brien	1982
6	s06	Shea Jennings	Caitlin	Ward	1978

```
CREATE TABLE School (  
    SID          varchar(4),  
    School_Name   char(30) not null,  
    Teacher_FName char(12) not null,  
    Teacher_LName varchar(15) not null,  
    Teacher_Cert_Year int not null,  
    primary key(SID)  
);
```

Dependencies:

SID → School_Name, Teacher_FName, Teacher_LName, Teacher_Cert_Year

Views: Results

To look at the results of all competitions across all levels:

```
CREATE VIEW Results AS
Select CompID, CID, Place
From Competes_Beginner
Union
Select CompID, CID, Place
From Competes_Prizewinner
Union
Select CompID, CID, Place
From Competes_Preliminary
Union
Select CompID, CID, Place
From Competes_Open
Order by CompID, Place ASC;
```

	compid integer	cid character varying(4)	place integer
1	100	c009	1
2	100	c007	2
3	100	c002	3
4	101	c008	1
5	101	c013	2
6	101	c005	3
7	101	c012	4
8	102	c013	1
9	102	c005	2
10	102	c012	3
11	103	c012	1
12	103	c008	2
13	103	c013	3
14	104	c001	1
15	104	c003	2
16	104	c006	3
17	105	c010	1
18	105	c017	2
19	106	c011	1
20	107	c018	1
21	107	c004	2
22	108	c014	1
23	108	c009	2
24	108	c002	3
25	109	c009	1
26	109	c014	2
27	109	c002	3
28	110	c007	1
29	110	c009	2
30	110	c014	3
31	110	c002	4
32	111	c006	1
33	111	c001	2
34	112	c016	1
35	112	c010	2

Views: Enrolled

To see the schools that the competitors are currently being taught at:

```
CREATE VIEW ENROLLED AS  
Select t.cid, t.sid, t.join_year  
From taught as t
```

Except

```
Select T1.cid, t1.sid, t1.join_year  
From taught as T1, taught as T2  
Where T1.join_year<T2.join_year  
and t1.cid=t2.cid
```

Order by cid asc;

	cid character varying(4)	sid character varying(4)	join_year integer
1	c001	s01	2003
2	c003	s04	2009
3	c004	s05	2005
4	c006	s01	2016
5	c007	s03	2004
6	c008	s05	2000
7	c009	s02	2015
8	c010	s05	2002
9	c011	s04	2011
10	c012	s01	2013
11	c013	s03	2016
12	c014	s02	2014
13	c015	s04	2013
14	c016	s03	2008
15	c017	s05	2006
16	c018	s04	2010

Query #1

Find a list of students who have changed schools, what school they changed to, and when:

```
Select T1.cid, S.School_Name, T1.join_year  
From taught as T1, taught as T2, School as S  
Where T1.cid=T2.cid  
And T1.sid=S.sid  
And T1.Join_Year>T2.Join_Year;
```

Alternatively:

```
Select Distinct t.cid  
From taught as t  
where t.cid in (  
    Select T1.cid  
    From taught as T1, taught as T2  
    Where T1.join_year>T2.join_year  
    and t1.cid=t2.cid  
);
```

	cid character varying(4)	school_name character(30)	join_year integer
1	c003	Schade Academy	2009
2	c006	Donny Golden	2016
3	c009	Mulvihill	2015
4	c014	Mulvihill	2014
5	c018	Schade Academy	2010

Query #2

Find the amount of total Last places of each competitor, from highest to lowest:

```
Select CID, Count(*)  
From Results  
Where (compid, place) in(  
    Select compid, max(place)  
    From Results  
    Group by (compid)  
    )  
Group By (CID)  
Order by count(*) DESC;
```

	cid character varying(4)	count bigint
1	c002	4
2	c012	3
3	c011	2
4	c004	2
5	c013	2
6	c010	1
7	c008	1
8	c017	1
9	c001	1
10	c006	1
11	c005	1



Stored Procedure #1

Given a musician, a judge, and a competitor, find competitions that were played by the musician and judged by the judge

```
CREATE OR REPLACE FUNCTION findConnection(TEXT, TEXT, REFCURSOR) RETURNS refcursor
AS
$$
    DECLARE
        Musician TEXT := $1;
        Judge TEXT := $2;
        resultSet REFCURSOR := $3;
BEGIN
    OPEN resultSet for
        Select c.mid, c.jid, c.comp_id
        From competition as c
        where c.mid = Musician
        and c.jid = Judge;
    RETURN resultSet;
END;
$$
LANGUAGE plpgsql;
```



Stored Procedure #1 (cont.)

Testing this procedure with:

```
Select findConnection('m001', 'j004', 'results');  
Fetch all from results;
```

Yields the following results:

	mid character varying(4)	jid character varying(4)	compid integer
1	m001	j004	108

Stored Procedure #2

Given a specific school, find how many first places competitors in that school earned.

```
CREATE OR REPLACE FUNCTION findSchoolFirsts(TEXT, REFCURSOR) RETURNS refcursor AS
$$
    DECLARE
        School TEXT := $1;
        resultset REFCURSOR := $2;
    BEGIN
        OPEN resultset for
            Select count(*) as FirstCount
            From Enrolled
            Where cid in(
                Select cid
                From Results
                where place = 1
            )
            and sid = School
            Group by (sid);
        RETURN resultset;
    END;
$$
LANGUAGE plpgsql;
```


Stored Procedure #2 (cont.)

Testing this procedure with the following:

```
select findSchoolFirsts('s01', 'results');  
fetch all from results;
```

Yields these results:

	firstcount bigint
1	3



Users and Security Roles

There are multiple roles: Organizer, Staff, and General

Organizer is the organizer of a feis, and they have full control of the database.

- create role Organizer;
- grant all on all tables in schema public to Organizer;

Staff are the musicians and judges, who can see what competitions they are involved in and how much they are paid (and for musicians, what songs they are expected to know)

- create role staff;
- grant SELECT on Judge to staff;
- grant SELECT on Musician to staff;
- grant SELECT on Competition to staff;
- grant SELECT on Plays to staff;



Users and Security Roles (cont.)

Then, there is General. This is any random person who wants to see what is happening in the feis, usually the parents of competitors (Dance moms are very invested in their child's performance, perhaps even enough to learn SQL)

- create role General;
- grant SELECT on Competition to General;
- grant SELECT on Competes_Beginner to General;
- grant SELECT on Competes_Prizewinner to General;
- grant SELECT on Competes_Open to General;
- grant SELECT on Competitor to General;
- grant SELECT on taught to General;
- grant SELECT on School to General;



Known Issues/Future Enhancements

- The 'Competes' tables are not fully normalized. Because of the nature of the way different levels have different possible Dances, I couldn't put a field called 'Dance' in the competition table, but instead had to put it in the competes prizewinner and beginner tables, creating a partial key dependency of CompID \rightarrow Dance
- Judges and musicians have a set salary that they charge for each feis, but I could implement a more complex way of having them charge a different amount for each feis depending on how much they judge/play