OLYMICS 2016

MADRID2016
CANDIDATE CITY

Database Management Design Project

1st December 2014

Ronald Dartey

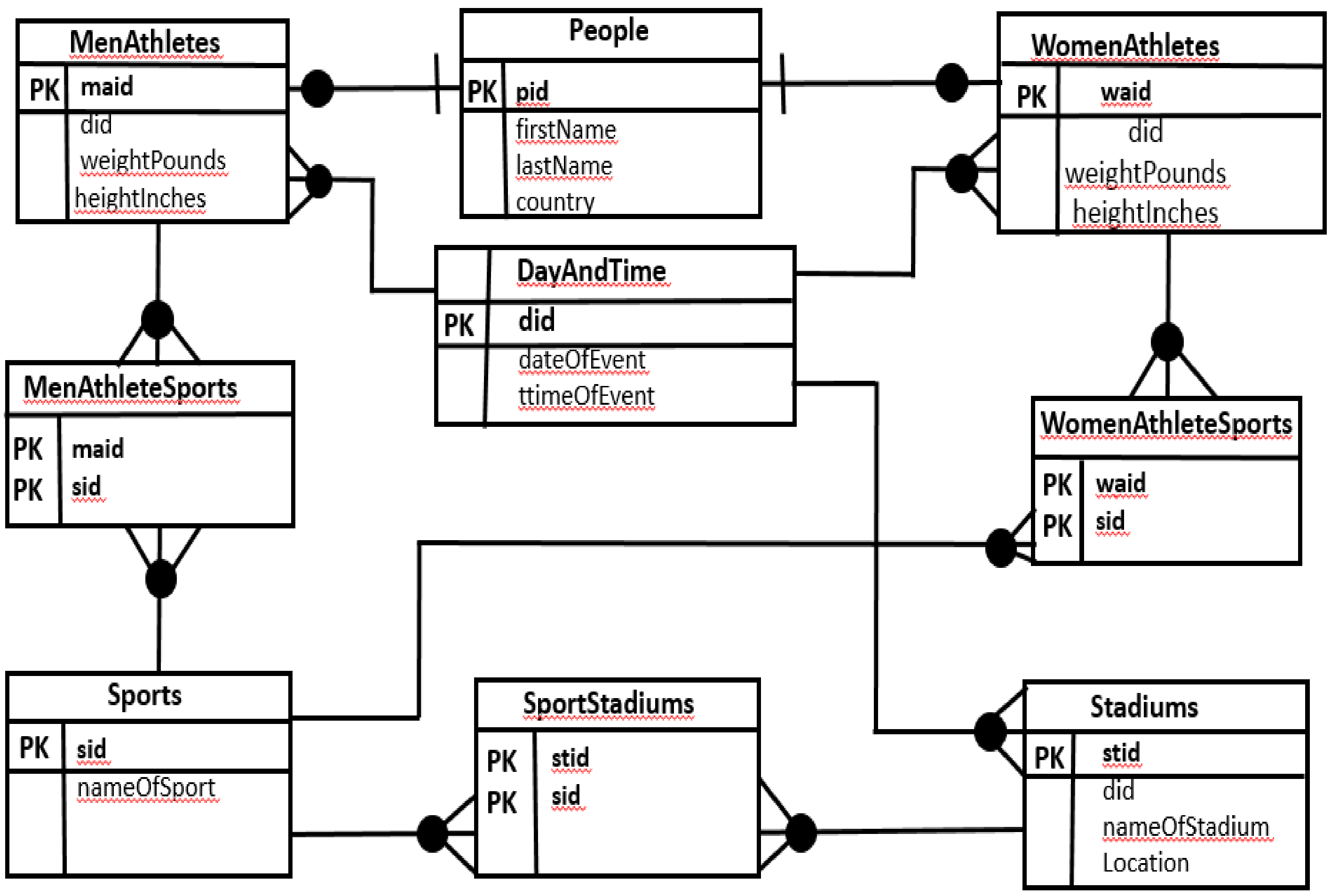
Table of Contents

Executive Summary

Spain is hosting Olympics in 2016 and the city for this event is Madrid. There will be different athletes from different countries to participate in this event. Athletes will be grouped into sexes in each sport (which can be swimming or handball). An athlete is allowed to play a sport with people in his or her gender group. There are days and times in which theses sports are played. There are also different stadiums to host this event.

This database is designed to keep track of the days and time each in which each athlete has a game. This will prevent athletes from different genders to play together in a sport. This database is also designed to prevent confliction, thus an event occurring at the same time in the same stadium.

Entity Relationship Diagram



Tables

1. **People**

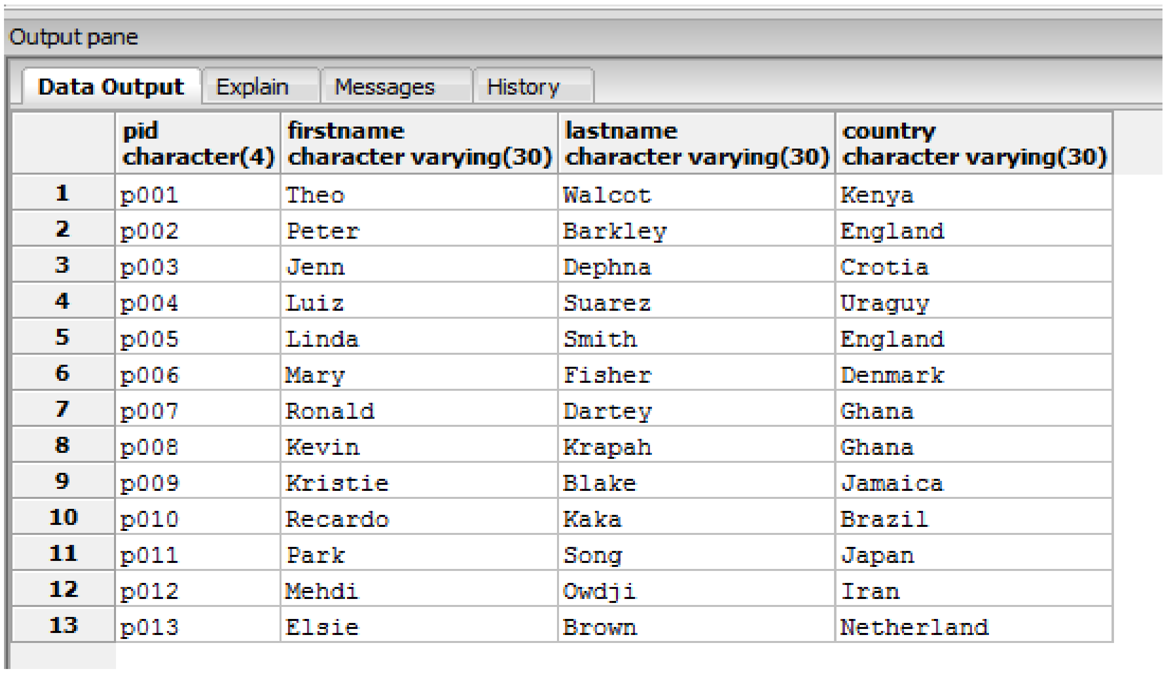
   Create Statement:

```
DROP TABLE IF EXISTS People;
-- People --
CREATE TABLE People(
pid char(4) not null,
firstName varchar(30) not null,
lastName varchar(30) not null,
country varchar(30),
primary key (pid)
);
```

   Insert Statements:

```
-- People --
INSERT INTO People( pid, firstName ,lastName, country)
  VALUES('p001', 'Theo', 'Walcot', 'Kenya');

INSERT INTO People( pid, firstName ,lastName, country)
  VALUES('p002', 'Peter', 'Barkley', 'England');

INSERT INTO People( pid, firstName ,lastName, country)
  VALUES('p003', 'Jenn', 'Dephna', 'Crotia');

INSERT INTO People( pid, firstName ,lastName, country)
  VALUES('p004', 'Luiz', 'Suarez', 'Uraguy');

INSERT INTO People( pid, firstName ,lastName, country)
  VALUES('p005', 'Linda', 'Smith', 'England');

INSERT INTO People( pid, firstName ,lastName, country)
  VALUES('p006', 'Mary', 'Fisher', 'Denmark');

INSERT INTO People( pid, firstName ,lastName, country)
  VALUES('p007', 'Ronald', 'Dartey', 'Ghana');

INSERT INTO People( pid, firstName ,lastName, country)
  VALUES('p008', 'Kevin', 'Krapah', 'Ghana');

INSERT INTO People( pid, firstName ,lastName, country)
  VALUES('p009', 'Kristie', 'Blake', 'Jamaica');

INSERT INTO People( pid, firstName ,lastName, country)
  VALUES('p010', 'Recardo', 'Kaka', 'Brazil');

INSERT INTO People( pid, firstName ,lastName, country)
  VALUES('p011', 'Park', 'Song', 'Japan');

INSERT INTO People( pid, firstName ,lastName, country)
  VALUES('p012', 'Mehdi', 'Owdji', 'Iran');

INSERT INTO People( pid, firstName ,lastName, country)
  VALUES('p013', 'Elsie', 'Brown', 'Netherland');
```

Sample data Output:


Output pane

Data Output | Explain | Messages | History

	pid character(4)	firstname character varying(30)	lastname character varying(30)	country character varying(30)
1	p001	Theo	Walcot	Kenya
2	p002	Peter	Barkley	England
3	p003	Jenn	Dephna	Crotia
4	p004	Luiz	Suarez	Uraguy
5	p005	Linda	Smith	England
6	p006	Mary	Fisher	Denmark
7	p007	Ronald	Dartey	Ghana
8	p008	Kevin	Krapah	Ghana
9	p009	Kristie	Blake	Jamaica
10	p010	Recardo	Kaka	Brazil
11	p011	Park	Song	Japan
12	p012	Mehdi	Owdji	Iran
13	p013	Elsie	Brown	Netherland

Functional Dependencies

Pid $\longrightarrow$ firstName, lastName, country

2. DayAndTime

Create Statement:

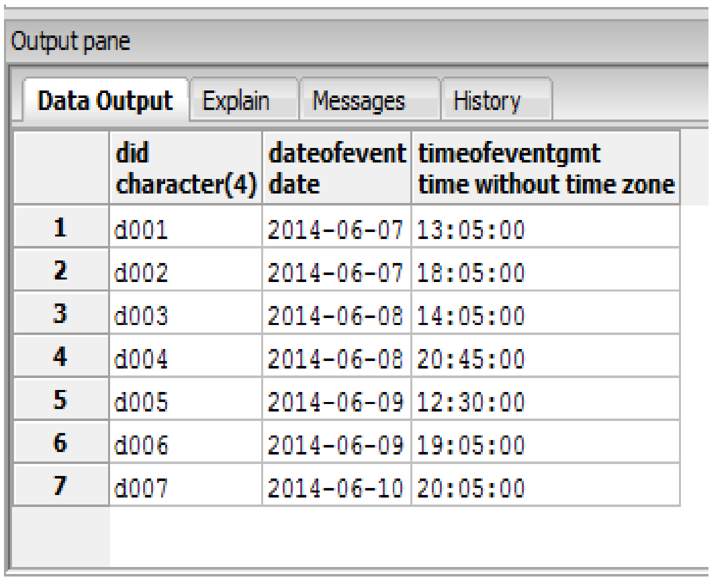
```
DROP TABLE IF EXISTS DayAndTime;
-- DayAndTime --
CREATE TABLE DayAndTime (
  did      char(4) not null,
  DateOfEvent   date,
  TimeOfEventGMT time,
 primary key(did)
);
```

Insert Statements:

```
-- DayAndTIme --
INSERT INTO DayAndTime( did, DateOfEvent, TimeOfEventGMT)
  VALUES('d001', '06/07/2014', '13:05');

INSERT INTO DayAndTime( did, DateOfEvent, TimeOfEventGMT)
  VALUES('d002', '06/07/2014', '18:05');

INSERT INTO DayAndTime( did, DateOfEvent, TimeOfEventGMT)
```

```
  VALUES('d003', '06/08/2014', '14:05');

INSERT INTO DayAndTime( did, DateOfEvent, TimeOfEventGMT)
  VALUES('d004', '06/08/2014', '20:45');

INSERT INTO DayAndTime( did, DateOfEvent, TimeOfEventGMT)
  VALUES('d005', '06/09/2014', '12:30');

INSERT INTO DayAndTime( did, DateOfEvent, TimeOfEventGMT)
  VALUES('d006', '06/09/2014', '19:05');

INSERT INTO DayAndTime( did, DateOfEvent, TimeOfEventGMT)
  VALUES('d007', '06/10/2014', '20:05');
```

Sample data output:



	did character(4)	dateofevent date	timeofeventgmt time without time zone
1	d001	2014-06-07	13:05:00
2	d002	2014-06-07	18:05:00
3	d003	2014-06-08	14:05:00
4	d004	2014-06-08	20:45:00
5	d005	2014-06-09	12:30:00
6	d006	2014-06-09	19:05:00
7	d007	2014-06-10	20:05:00

Functional Dependencies:

did $\longrightarrow$ dateOfEvent, timeOfEventGMT

3. MenAthletes

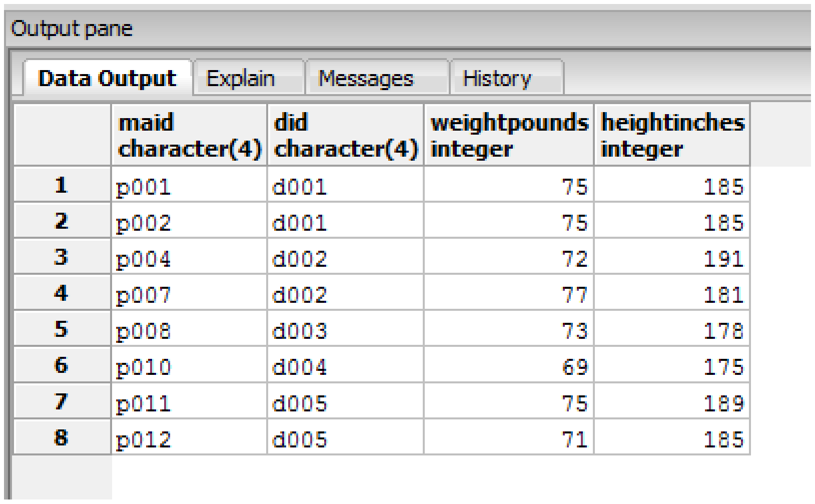
Create Statement:

```
DROP TABLE IF EXISTS MenAthletes;
-- MenAthletes --
CREATE TABLE MenAthletes (
  maid char(4) references People(pid),
  did     char(4) not null references DayAndTime(did),
  weightPounds int,
  heightInches int,
  primary key(maid),
  unique (maid)
);
```

Insert Statements:

```
-- MenAthletes --
INSERT INTO MenAthletes( maid, did, weightPounds, heightInches)
  VALUES('p001', 'd001', 75, 185);

INSERT INTO MenAthletes( maid, did, weightPounds, heightInches)
  VALUES('p002', 'd001', 75, 185);

INSERT INTO MenAthletes( maid, did, weightPounds, heightInches)
  VALUES('p004', 'd002', 72, 191);

INSERT INTO MenAthletes( maid, did, weightPounds, heightInches)
  VALUES('p007', 'd002', 77, 181);

INSERT INTO MenAthletes( maid, did, weightPounds, heightInches)
  VALUES('p008', 'd003', 73, 178);

INSERT INTO MenAthletes( maid, did, weightPounds, heightInches)
  VALUES('p010', 'd004', 69, 175);

INSERT INTO MenAthletes( maid, did, weightPounds, heightInches)
  VALUES('p011', 'd005', 75, 189);

INSERT INTO MenAthletes( maid, did, weightPounds, heightInches)
  VALUES('p012', 'd005', 71, 185);
```

Sample data output:



	maid character(4)	did character(4)	weightpounds integer	heightinches integer
1	p001	d001	75	185
2	p002	d001	75	185
3	p004	d002	72	191
4	p007	d002	77	181
5	p008	d003	73	178
6	p010	d004	69	175
7	p011	d005	75	189
8	p012	d005	71	185

Functional dependencies:

maid $\longrightarrow$ did, weightPounds, heightInches.

4. WomenAthletes

Create Statement:

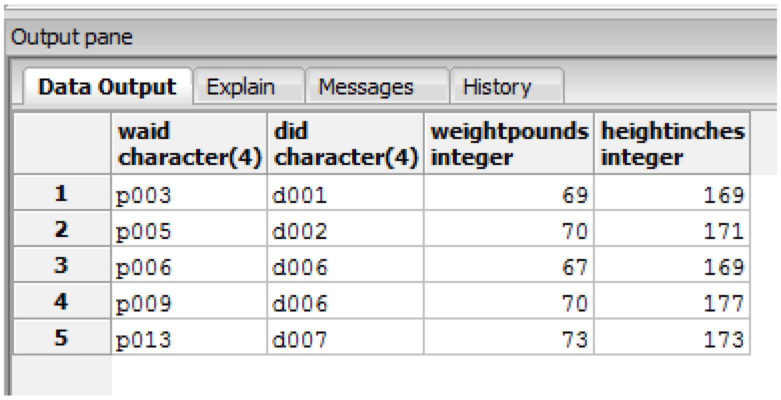
```
DROP TABLE IF EXISTS WomenAthletes;
-- WomenAthletes --
CREATE TABLE WomenAthletes (
  waid    char(4) references People(pid),
  did     char(4) not null references DayAndTime(did),
  weightPounds   int,
  heightInches   int,
  primary key(waid),
  unique (waid)
);
```

Insert Statements:

```
-- WomenAthletes --
INSERT INTO WomenAthletes( waid, did, weightPounds, heightInches)
  VALUES('p003', 'd001', 69, 169);

INSERT INTO WomenAthletes( waid, did, weightPounds, heightInches)
  VALUES('p005', 'd002', 70, 171);

INSERT INTO WomenAthletes( waid, did, weightPounds, heightInches)
  VALUES('p006', 'd006', 67, 169);
```

```
INSERT INTO WomenAthletes( waid, did, weightPounds, heightInches)
  VALUES('p009', 'd006', 70, 177);

INSERT INTO WomenAthletes( waid, did, weightPounds, heightInches)
  VALUES('p013', 'd007', 73, 173);
```

Sample data output:


Output pane

Data Output | Explain | Messages | History

	waid character(4)	did character(4)	weightpounds integer	heightinches integer
1	p003	d001	69	169
2	p005	d002	70	171
3	p006	d006	67	169
4	p009	d006	70	177
5	p013	d007	73	173

Functional dependencies:

waid $\longrightarrow$ did, weightPounds, heightInches.

5. Sports

Create Statement

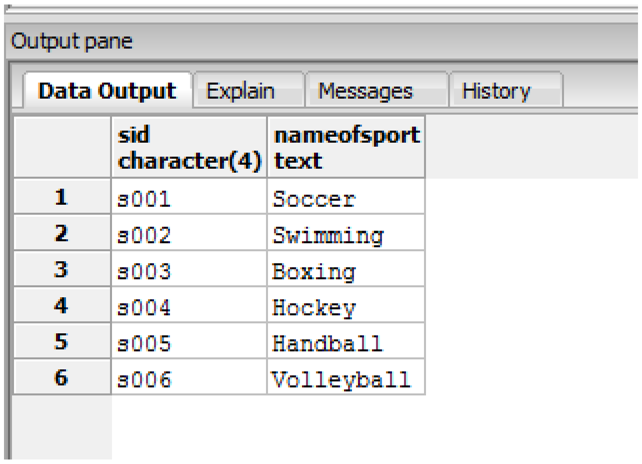
```
DROP TABLE IF EXISTS Sports;
-- Sports --
CREATE TABLE Sports (
  sid     char(4) not null,
  nameOfSport   text,
  primary key(sid)
);
```

Insert Statements:

```
-- Sports --
INSERT INTO Sports(sid, nameOfSport)
  VALUES('s001', 'Soccer');

INSERT INTO Sports(sid, nameOfSport)
  VALUES('s002', 'Swimming');

INSERT INTO Sports(sid, nameOfSport)
  VALUES('s003', 'Boxing');
```

```
INSERT INTO Sports(sid, nameOfSport)
  VALUES('s004', 'Hockey');

INSERT INTO Sports(sid, nameOfSport)
  VALUES('s005', 'Handball');

INSERT INTO Sports(sid, nameOfSport)
  VALUES('s006', 'Volleyball');
```

<u>Sample data output:</u>



	sid character(4)	nameofsport text
1	s001	Soccer
2	s002	Swimming
3	s003	Boxing
4	s004	Hockey
5	s005	Handball
6	s006	Volleyball

<u>Functional Dependencies:</u>

sid $\longrightarrow$ nameOfSport

6. MenAthleteSports

<u>Create Statement:</u>

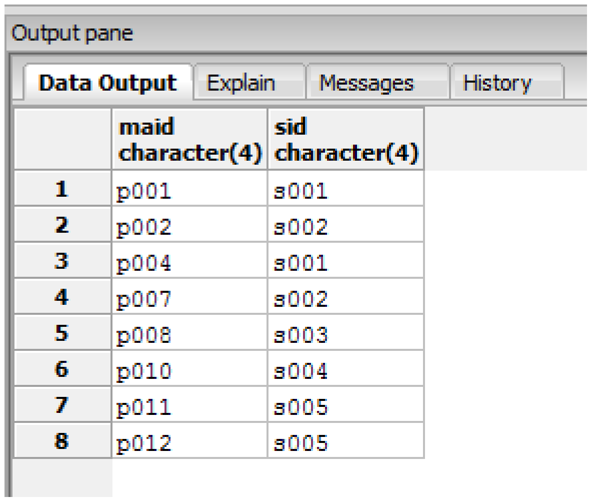
```
DROP TABLE IF EXISTS MenAthleteSports;
--MenAthleteSports--
CREATE TABLE MenAthleteSports (
  maid char(4) references MenAthletes(maid),
  sid char(4) references Sports(sid),
  primary key (maid, sid),
  unique (maid)
);
```

<u>Insert Statements:</u>

```
--MenAthleteSports--
```

```
INSERT INTO MenAthleteSports(maid, sid)
VALUES('p001', 's001');

INSERT INTO MenAthleteSports(maid, sid)
VALUES('p002', 's002');

INSERT INTO MenAthleteSports(maid, sid)
VALUES('p004', 's001');

INSERT INTO MenAthleteSports(maid, sid)
VALUES('p007', 's002');

INSERT INTO MenAthleteSports(maid, sid)
VALUES('p008', 's003');

INSERT INTO MenAthleteSports(maid, sid)
VALUES('p010', 's004');

INSERT INTO MenAthleteSports(maid, sid)
VALUES('p011', 's005');

INSERT INTO MenAthleteSports(maid, sid)
VALUES('p012', 's005');
```

Sample data output:



	maid character(4)	sid character(4)
1	p001	s001
2	p002	s002
3	p004	s001
4	p007	s002
5	p008	s003
6	p010	s004
7	p011	s005
8	p012	s005

Functional Dependencies:

maid and sid are the composite key for this table.

7. WomenAthleteSports

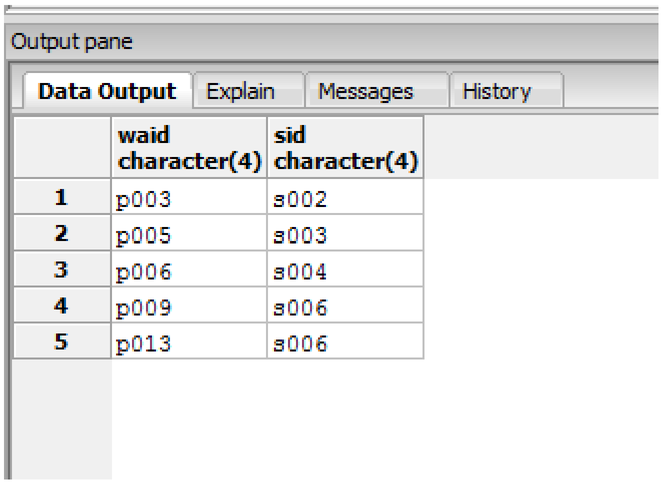
Create Statement:

```
DROP TABLE IF EXISTS WomenAthleteSports;
-- WomenAthleteSports --
CREATE TABLE WomenAthleteSports (
  waid char(4) references WomenAthletes(waid),
  sid char(4) references Sports(sid),
  primary key (waid, sid),
  unique (waid)
);
```

Insert Statements:

```
--WomenAthleteSports--
INSERT INTO WomenAthleteSports(waid, sid)
VALUES('p003', 's002');

INSERT INTO WomenAthleteSports(waid, sid)
VALUES('p005', 's003');

INSERT INTO WomenAthleteSports(waid, sid)
VALUES('p006', 's004');

INSERT INTO WomenAthleteSports(waid, sid)
VALUES('p009', 's006');

INSERT INTO WomenAthleteSports(waid, sid)
VALUES('p013', 's006');
```

Sample data output:



	waid character(4)	sid character(4)
1	p003	s002
2	p005	s003
3	p006	s004
4	p009	s006
5	p013	s006

Functional Dependencies:

waid and sid are the composite key for this table.

8. Stadiums

Create Statements :

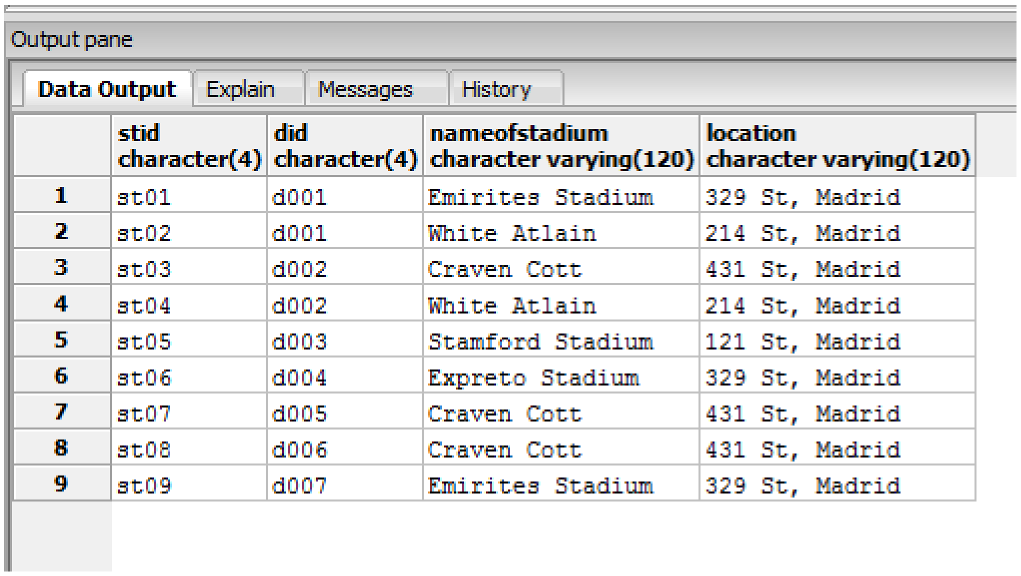
```
DROP TABLE IF EXISTS Stadiums;
-- Stadiums --
CREATE TABLE Stadiums (
  stid     char(4) not null,
  did     char(4) not null references DayAndTime(did),
  nameOfStadium   varchar(120),
  Location  varchar(120),
  primary key(stid),
  unique (stid)
);
```

Insert Statements:

```
-- Stadium --
INSERT INTO Stadiums( stid, did, nameOfStadium, Location)
  VALUES('st01', 'd001', 'Emirites Stadium', '329 St, London');

INSERT INTO Stadiums( stid, did, nameOfStadium, Location)
  VALUES('st02', 'd001', 'White Atlain', '214 St, London');

INSERT INTO Stadiums( stid, did, nameOfStadium, Location)
  VALUES('st03', 'd002', 'Craven Cott', '431 St, London');

INSERT INTO Stadiums( stid, did, nameOfStadium, Location)
  VALUES('st04', 'd002', 'White Atlain', '214 St, London');

INSERT INTO Stadiums( stid, did, nameOfStadium, Location)
  VALUES('st05', 'd003', 'Stamford Stadium', '121 St, London');

INSERT INTO Stadiums( stid, did, nameOfStadium, Location)
  VALUES('st06', 'd004', 'Emirites Stadium', '329 St, London');

INSERT INTO Stadiums( stid, did, nameOfStadium, Location)
  VALUES('st07', 'd005', 'Craven Cott', '431 St, London');

INSERT INTO Stadiums( stid, did, nameOfStadium, Location)
  VALUES('st08', 'd006', 'Craven Cott', '431 St, London');
```

INSERT INTO Stadiums(stid, did, nameOfStadium, Location)
  VALUES('st09', 'd007', 'Emirites Stadium', '329 St, London');

Sample data output:



	stid character(4)	did character(4)	nameofstadium character varying(120)	location character varying(120)
1	st01	d001	Emirites Stadium	329 St, Madrid
2	st02	d001	White Atlain	214 St, Madrid
3	st03	d002	Craven Cott	431 St, Madrid
4	st04	d002	White Atlain	214 St, Madrid
5	st05	d003	Stamford Stadium	121 St, Madrid
6	st06	d004	Expreto Stadium	329 St, Madrid
7	st07	d005	Craven Cott	431 St, Madrid
8	st08	d006	Craven Cott	431 St, Madrid
9	st09	d007	Emirites Stadium	329 St, Madrid

Functional Dependencies:

stid $\longrightarrow$ did, nameOfStadium, Location.

9. SportStadium

Create Statements:

DROP TABLE IF EXISTS SportStadium;
-- SportStadium --
CREATE TABLE SportStadium (
  stid char(4) references Stadiums(stid),
  sid char(4) references Sports(sid),
  primary key (stid, sid)
);

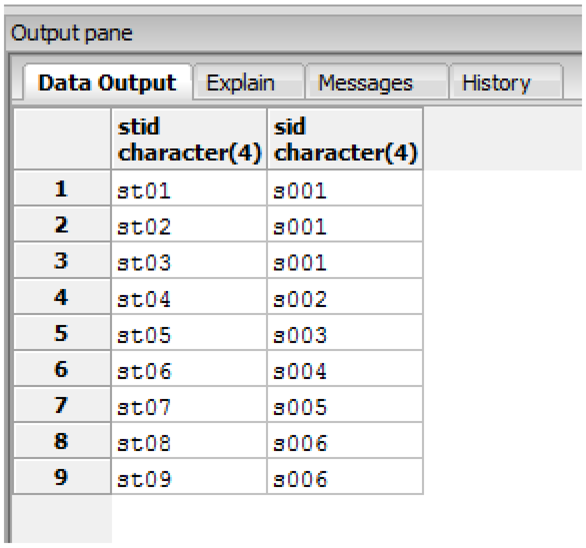
Insert Statements:

--SportStadium--
INSERT INTO SportStadium(stid, sid)
VALUES('st01', 's001');

INSERT INTO SportStadium(stid, sid)

```
VALUES('st02', 's001');

INSERT INTO SportStadium(stid, sid)
VALUES('st03', 's001');

INSERT INTO SportStadium(stid, sid)
VALUES('st04', 's002');

INSERT INTO SportStadium(stid, sid)
VALUES('st05', 's003');

INSERT INTO SportStadium(stid, sid)
VALUES('st06', 's004');

INSERT INTO SportStadium(stid, sid)
VALUES('st07', 's005');

INSERT INTO SportStadium(stid, sid)
VALUES('st08', 's006');

INSERT INTO SportStadium(stid, sid)
VALUES('st09', 's006');
```

Sample data output:



	stid character(4)	sid character(4)
1	st01	s001
2	st02	s001
3	st03	s001
4	st04	s002
5	st05	s003
6	st06	s004
7	st07	s005
8	st08	s006
9	st09	s006

Functional Dependencies:

stid and sid are the composite key for this table.

Views:

1. Athlete Names

This view gets the names of all the athletes present at the Event.

```
--view 1—
DROP VIEW IF EXISTS athleteName;


create view athleteName AS
	select distinct firstName, lastName
	from People p,
		MenAthletes ma,
		WomenAthletes wa
	where p.pid = ma.maid
	or p.pid = wa.waid


select *
from athleteName
```

2. SoccerTime

This view gets the date and time in which soccer is played.

```
--view 2—
DROP VIEW IF EXISTS SoccerTime;
create view SoccerTime AS
	select distinct d.dateOfEvent, d.timeOfEventGMT
	from DayAndTime d,
		Stadiums st,
		SportStadium sp,
		Sports s
	where d.did = st.did
	and st.stid = sp.stid
```

```
        and sp.sid = s.sid
        and s.nameOfSport = 'Soccer'


select *
from SoccerTime
```

Reports and queries.

This query gets the first and last name of athletes who are men and have a game on 2014-06-08.

```
    select p.firstName, p.lastName
    from People p
    where p.pid in ( select maid
                     from MenAthletes
                     where did in ( select did
                                    from DayAndTime
                                    where dateOfEvent = '2014-06-08')
                                    )
```

Stored Procedures:

This stored procedure has a function called athletesOfStadium. This function takes in a name of a stadium and returns the first and last names of athletes who have a game in that stadium.

```
--Stored Procedure--
create or replace function athletesOfStadium(varchar(120), REFCURSOR) returns refcursor as
$$
declare
  nameOfStadium varchar(120)   := $1;
  resultset   REFCURSOR := $2;
begin
  open resultset for
```

```
    select p.firstName, p.lastName
    from   People p,
                   MenAthletes ma
    where p.pid = ma.maid
    and ma.maid in (select mt.maid
                      from MenAthleteSports mt
                      where mt.sid in (select s.sid
                                         from Sports s
                                         where s.sid in (select sp.sid
                                                          from SportStadium sp
                                                          where sp.stid in (select st.stid
                                                                             from Stadiums st
                                                                             where st.nameOfStadium =
nameOfStadium))));


  return resultset;
end;
$$
language plpgsql;


select athletesOfStadium('Emirites Stadium', 'results');
Fetch all from results;
```

Triggers:

This trigger prevents user from inserting or updating data if the Olympic events are closed since the Olympics close on a specific date. It also helps to prevent feeding the database with wrong data.

```
DROP TRIGGER IF EXISTS dateTestTrigger on DayAndTime;
create or replace function dateTest()
```

```
  returns trigger as
$BODY$
declare
   account_type varchar;
begin
    IF (NEW.DateOfEvent > '2014-06-10') then
           raise NOTICE 'WARNING : There are no events on this day %' , NEW.DateOfEvent;
           end if;
   return null;
end;
$BODY$
  language plpgsql volatile
  cost 100;
alter function dateTest()
  owner to postgres;


create trigger dateTestTrigger
after insert or update
on DayAndTime
for each row
execute procedure dateTest();
```

Security:

Database administrators will have access to all the database. Women and men athletes will be allowed to enter the kind of sports they play. Women athletes will not be allowed to view the men athlete table and vice versa.

```
--Database Administrator--
grant all privileges on all tables in schema public to bdAdministrator;
--Women Athletes--
```

```
grant insert on WomenAthletes to Sports;

revoke WomenAthletes from MenAthletes


--MenAthletes--

grant insert on MenAthletes to Sports;

revoke MenAthletes from WomenAthletes
```

Implementation Notes:

An athlete will be allowed to view his table and will, therefore know the time he or she has a game. An athlete should also know the kind of sports he or she plays.

Known Problem:

Each athlete is allowed to play only one sport. Men athletes and women athletes can play the same kind of sports. Also a stadium can have more than one sport event but not the same time. Men and women athletes will not have a sport together at the same time and in the same stadium.

Future Enhancement:

People table could have weightPounds and heightInches to prevent the pain of entering the same column twice, thus in the men and women athlete's tables