A DATABASE DESIGN ON

GAMING EVENTS

Database Specifications

25 April 2014

Rakesh Reddy Nandigama

Table of Contents

INTRODUCTION

A game is structured playing, usually undertaken for enjoyment and sometimes used as an educational tool. The games plays an important role in the present day life. Now-a-days the amount spent on the games is increasing day-by-day because many people are attracting for the games whether the games may be online or outdoor or indoor games. Games become popular in all over the world. Many game events are designed in organizing the games.

This document gives the brief analysis, design and implementation of a database for the game events which include continent, country, players, events, game, ranking etc.,

Some of the possible cases in the database are

1. Continents: It gives the player a reference from which continent the players are from.

2. Country: It gives the player a reference from which country the players.

3. Players: It gives the information about the player from where he belongs to and maintains the track or the information about the player and to which the clan the player belongs to.

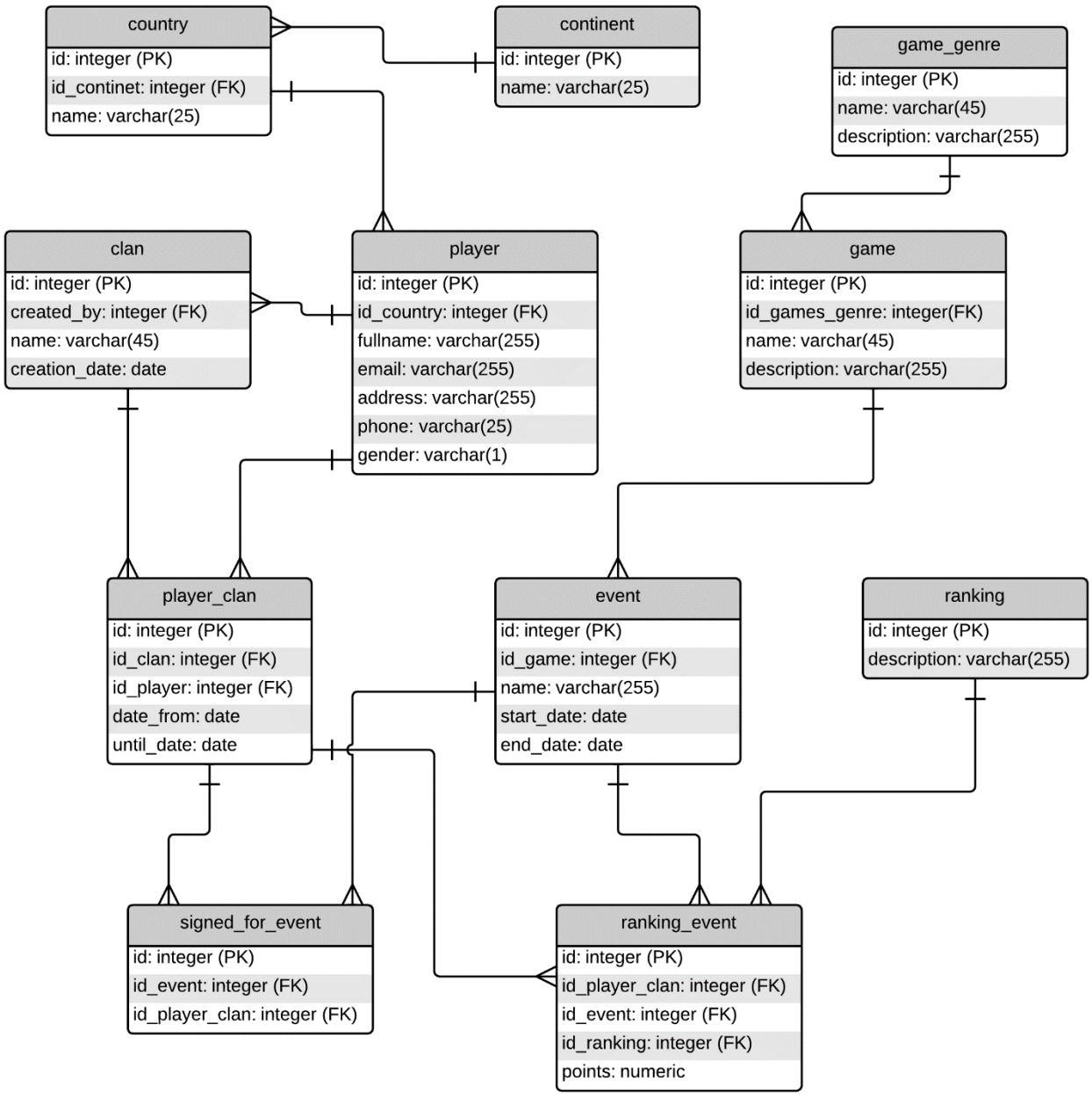
4. Clan: A clan is an organized group of players that play together by signing for an event. This clan is created on a particular and the clan start playing the game.

5. Game: Game is a table in the data base which gives a detailed information about all the games present and these games are classified according to game_genre.

6. Events: Event is a table which contain details of all the events that are happening and the games that are involved in the events. The table also gives information about, when the events are started and closed.

7. Ranking: Ranking table gives the information that how the clans are participating anh which clan have more points.

Entity Relationship Diagram



Tables

Country

Purpose

A table containing the group of countries and their id's.

Create Statement

```
CREATE TABLE country
(
  id integer NOT NULL,
  id_continent integer NOT NULL,
  name character varying(25) NOT NULL,
  CONSTRAINT "pkIdCountry" PRIMARY KEY (id),
  CONSTRAINT "fkIdContinent" FOREIGN KEY (id_continent)
      REFERENCES continent (id)
);
```

Functional Dependencies

Id → id_continent, name

Sample Data

	id integer	id_continent integer	name character varying(25)
1	1	1	venezuela
2	2	1	colombia
3	3	2	canada
4	4	2	usa
5	5	3	spain
6	6	3	italy
7	7	4	china
8	8	4	japan
9	9	4	south korea
10	10	5	south africa

Continent

Purpose :

A table containing the group of continents and their id's.

Create Statement

```
CREATE TABLE continent
(
  id integer NOT NULL,
  name character varying(25),
  CONSTRAINT "pkIdContinent" PRIMARY KEY (id)
);
```

Functional Dependencies

Id → name

Sample Data

	id integer	name character varying(25)
1	1	south america
2	2	north america
3	3	europe
4	4	asia
5	5	africa

Clan

Purpose

Clan is an organized group of players that play together. This table contains details related to clan regarding the name and the creation date of clan.

Create Statement

```
CREATE TABLE clan
(
  id integer NOT NULL,
  created_by integer NOT NULL,
  name character varying(45) NOT NULL,
  creation_date date NOT NULL,
  CONSTRAINT "pkIdClan" PRIMARY KEY (id),
  CONSTRAINT "fkIdPlayer1" FOREIGN KEY (created_by)
      REFERENCES player (id)
);
```

Functional Dependencies

Id → created_by, name, creation_date

Sample Data

	id integer	created_by integer	name character varying(45)	creation_date date
1	1	1	cyborg	2013-01-02
2	2	2	terminator	2013-06-07
3	3	3	crazy	2013-08-09
4	4	4	the random	2013-12-11
5	5	5	ninja criminal	2013-10-11
6	6	6	hiper skull	2013-09-08
7	7	7	bipolar mission	2013-12-12
8	8	10	liver code	2013-03-03
9	9	14	cat polse	2013-04-04
10	10	18	clan 080	2013-07-07
11	12	1	clan rest of world	2014-02-02
12	11	11	clan asia	2014-01-01

Event

Purpose

A table containing the group of event names and their starting and ending dates.

Create Statement

```
CREATE TABLE event
(
  id integer NOT NULL,
  id_game integer NOT NULL,
  name character varying(255) NOT NULL,
  start_date date NOT NULL,
  end_date date NOT NULL,
  CONSTRAINT "pkIdEvent" PRIMARY KEY (id),
  CONSTRAINT "fkIdGame" FOREIGN KEY (id_game)
      REFERENCES game (id)
);
```

Functional Dependencies

Id → id_game, name, start_date, end_date

Sample Data

	id integer	id_game integer	name character varying(255)	start_date date	end_date date
1	2	3	MMO event	2014-02-25	2014-03-04
2	1	1	FPS event	2014-03-03	2014-03-10
3	4	6	RPG event	2014-04-01	2014-04-07
4	5	7	TPS event	2014-03-20	2014-03-27
5	6	9	RACING event	2014-04-10	2014-04-17
6	3	5	MMORPG event	2014-03-15	2014-03-22

Game

Purpose

This table contains the information about the type of game and the game description.

Create Statement

```
CREATE TABLE game
(
  id integer NOT NULL,
  id_games_genre integer NOT NULL,
  name character varying(45) NOT NULL,
  description character varying(255),
  CONSTRAINT "pkIdGame" PRIMARY KEY (id),
  CONSTRAINT "fkIdGameGenre" FOREIGN KEY (id_games_genre)
      REFERENCES game_genre (id)
);
```

Functional Dependencies

Id → id_games_genre, name, description

Sample Data

	id integer	id_games_genre integer	name character varying(45)	description character varying(255)
1	1	1	Counter Strike	
2	2	1	call of duty	
3	3	2	Forge of empires	
4	5	3	dota 2	
5	4	3	LOL	league of legends
6	6	4	diablo 3	
7	7	5	gears of war 3	
8	9	6	need for speed	
9	8	6	mario kart	

Game Genre

Purpose

This table contains details about the game categories that are available and the id's of the game categories and its description.

Create Statement

```
CREATE TABLE game_genre
(
  id integer NOT NULL,
  name character varying(45) NOT NULL,
  description character varying(255) NOT NULL,
  CONSTRAINT "pkIdGameGenre" PRIMARY KEY (id)
);
```

Functional Dependencies

Id → name, description

Sample Data

	id integer	name character varying(45)	description character varying(255)
1	1	FPS	First-Person Shooter
2	2	MMO	Massively multiplayer online
3	3	MMORPG	massively multiplayer online role-playing game
4	5	TPS	third-person shooter
5	6	RACING	racing games
6	4	RPG	role-playing game

Player

Purpose

This table gives the entire information about the player like the player id, full name, address, phone number, gender and the players country.

Create Statement

```
CREATE TABLE player
(
    id integer NOT NULL,
    fullname character varying(255) NOT NULL,
    email character varying(255) NOT NULL,
    address character varying(255) NOT NULL,
    phone character varying(25) NOT NULL,
    gender "char" NOT NULL,
    id_country integer NOT NULL,
    CONSTRAINT "pkIdPlayer" PRIMARY KEY (id),
    CONSTRAINT "fkIdCountry" FOREIGN KEY (id_country)
        REFERENCES country (id)
);
```

Functional Dependencies

Id → fullname, email, address, phone, gender, id_country

Sample Data

	id integer	fullname character varying(255)	email character varying(255)	address character varying(255)	phone character varying(25)	gender "char"	id_country integer
1	3	Rosenfeld Tuly	t.rosenfeld@unsw.edu.au	av 1, house #2-7	425375796475	F	3
2	2	Friedlander Michael Leonard	z9000657@unsw.edu.au	av 10,house #8-2	437854903267	M	2
3	1	Andrews Peter Ian	i.andrews@UNSW.EDU.AU	av 3, house #15-2	653467633232	M	1
4	4	Bateman Hazel Jane	h.bateman@UNSW.EDU.AU	av 8, house #5-9	325467896745	F	4
5	5	Magnani Elisabetta	e.magnani@UNSW.EDU.AU	av 2, house #7-1	987645678976	F	5
6	6	Meagher Kieron John	k.meagher@UNSW.EDU.AU	av 7, house #5-7	213456789012	M	6
7	7	Zhao Yong	y.zhao@UNSW.EDU.AU	av 12, house #7-3	986745672331	M	7
8	8	Maher Lisa	L.Maher@unsw.edu.au	av 30, house #45-30	356453758665	F	7
9	9	Kaye Sharlene Susan	s.kaye@med.unsw.edu.au	av 20,house #3-1	898978677856	F	7
10	10	Munro Richard	r.munro@UNSW.EDU.AU	av 10, house #7-9	434556677889	M	8
11	11	Janssen David Adam	d.janssen@UNSW.EDU.AU	av 7,house #5-5	54545677876	M	8
12	12	Hoffman Mark John	mark.hoffman@UNSW.EDU.AU	av 9, house #6-7	436587897654	M	8
13	13	Murrell George Anthony Calvert	murrell.g@ori.org.au	av 2, house #8-4	098765767865	M	8
14	14	Douglas Kurt John	k.douglas@unsw.edu.au	av 22.house #2-2	656587767865	M	9
15	15	Sutherland Daniel David	d.sutherland@UNSW.EDU.AU	av 6 , house #7-4	213234456576	M	9
16	16	Tompkins Robyn	r.tompkins@UNSW.EDU.AU	av 8 , house #3-5	676567789887	M	9
17	17	Thompson Kevin Michael	z4510847@unsw.edu.au	av 9, house #6-6	566556566778	M	9
18	18	Crawford Philip John	p.crawford@UNSW.EDU.AU	av 7, house #3-33	343443344343	M	10

Player Clan

Purpose

This table contains information regarding the player id, clan id and the date when the clan has started.

Create Statement

```
CREATE TABLE player_clan
(
    id integer NOT NULL,
    id_clan integer NOT NULL,
    id_player integer NOT NULL,
    date_from date NOT NULL,
    until_date integer,
    CONSTRAINT "pkIdPlayerClan" PRIMARY KEY (id),
    CONSTRAINT "fkIdClan" FOREIGN KEY (id_clan)
        REFERENCES clan (id),
    CONSTRAINT "fkIdPlayer" FOREIGN KEY (id_player)
        REFERENCES player (id)
);
```

Functional Dependencies

Id → id_clan, id_player, date_from, until_date

Sample Data

	id integer	id_clan integer	id_player integer	date_from date	until_date integer
1	1	1	1	2013-01-02	
2	2	2	2	2013-06-07	
3	3	3	3	2013-08-09	
4	4	4	4	2013-12-11	
5	5	5	5	2013-10-11	
6	6	6	6	2013-09-08	
7	7	7	7	2013-12-12	
8	8	8	10	2013-03-03	
9	9	9	14	2013-04-04	
10	10	10	18	2013-07-07	
11	11	11	11	2014-01-01	
12	13	11	8	2014-01-01	
13	12	11	7	2014-01-01	
14	22	12	1	2014-02-02	

Ranking

Purpose

This table gives information about the rankings of the events corresponding to their id's.

Create Statement

```
CREATE TABLE ranking
(
  id integer NOT NULL,
  description character varying(255) NOT NULL,
  CONSTRAINT "pkIdRanking" PRIMARY KEY (id)
);
```

Functional Dependencies

Id → description

Sample Data

	id integer	description character varying(255)
1	1	ranking for FPS event
2	2	ranking for MMO event
3	3	ranking for MMORPG event
4	4	ranking for RPG event
5	5	ranking for TPS event
6	6	ranking for RACING event

Ranking Event

Purpose

This table gives the information regarding the player clan id who signed for the corresponding event, ranking and their points.

Create Statement

```
CREATE TABLE ranking_event
(
  id integer NOT NULL,
  id_player_clan integer NOT NULL,
  id_event integer NOT NULL,
  id_ranking integer NOT NULL,
  points numeric NOT NULL DEFAULT 0,
  CONSTRAINT "pkIdRankingEvent" PRIMARY KEY (id),
  CONSTRAINT "fkIdEvent" FOREIGN KEY (id_event)
      REFERENCES event (id),
  CONSTRAINT "fkIdPlayerClan" FOREIGN KEY (id_player_clan)
      REFERENCES player_clan (id),
  CONSTRAINT "fkIdRanking" FOREIGN KEY (id_ranking)
      REFERENCES ranking (id)
);
```

Functional Dependencies

Id → id_player_clan, id_event, id_ranking, points

Sample Data

	id integer	id_player_clan integer	id_event integer	id_ranking integer	points numeric
1	1	1	1	1	378
2	2	2	2	2	276
3	3	3	3	3	408
4	4	4	4	4	356
5	5	5	5	5	698
6	6	6	6	6	790
7	7	11	1	1	989
8	8	12	1	1	678
9	9	13	1	1	345
10	10	14	1	1	876
11	11	15	1	1	796
12	12	16	1	1	378
13	13	17	1	1	234
14	14	18	1	1	112
15	15	19	1	1	956
16	16	20	1	1	768

Signed for Event

Purpose

This table gives the information regarding the player clan id who signed for the corresponding event.

Create Statement

```
CREATE TABLE signed_for_event
(
  id integer NOT NULL,
  id_player_clan integer NOT NULL,
  id_event integer NOT NULL,
  CONSTRAINT "pkIdPlayerSigned" PRIMARY KEY (id),
  CONSTRAINT "fkIdEvent1" FOREIGN KEY (id_event)
      REFERENCES event (id),
  CONSTRAINT "fkIdPlayerClan" FOREIGN KEY (id_player_clan)
      REFERENCES player_clan (id)
);
```

Functional Dependencies

Id → id_player_clan, id_event

Sample Data

	id integer	id_player_clan integer	id_event integer
1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5
6	6	6	6
7	7	11	1
8	8	12	1
9	9	13	1
10	10	14	1
11	11	15	1
12	12	16	1
13	13	17	1
14	14	18	1
15	15	19	1
16	16	20	1

INTERESTING QUERIES

1.

```
  select * from ranking
where id in (  select c.id from player_clan c
               join  event ct
               on c.id=ct.id and ct.name='RACING event')
```

OUTPUT

Data Output | Explain | Messages | History

	id integer	description character varying(255)
1	6	ranking for RACING event

2.

```
select * from player
where id in (  select c.id from country c
               join continent ct
               on c.id=ct.id and ct.name='asia' )
```

OUTPUT

Data Output | Explain | Messages | History

	id integer	fullname character varying(255)	email character varying(255)	address character varying(255)	phone character varying(25)	gender "char"	id_country integer
1	4	Bateman Hazel Jane	h.bateman@UNSW.EDU..	av 8, house #5-9	325467896745	F	4

Triggers

1. Create Ranking

Purpose

A trigger is fired up when a row of a table is created, updated or deleted.

This trigger function is used in such a way that when the event is created then automatically the ranking is created.

Query

```
CREATE TRIGGER create_ranking
  AFTER INSERT
  ON event
  FOR EACH ROW
  EXECUTE PROCEDURE create_ranking();
```

```
CREATE OR REPLACE FUNCTION create_ranking()
  RETURNS trigger AS
$BODY$
    BEGIN
        insert into ranking values (new.id, 'ranking for '|| new.name);
        return new;
    END;
$BODY$
  LANGUAGE plpgsql VOLATILE
  COST 100;
```

2. Delete Ranking

Purpose

This trigger function is used in such a way that when the event is deleted then the automatically the ranking is deleted.

Query

```
CREATE TRIGGER delete_ranking
  BEFORE DELETE
  ON event
  FOR EACH ROW
  EXECUTE PROCEDURE delete_ranking();
```

```
CREATE OR REPLACE FUNCTION delete_ranking()
  RETURNS trigger AS
$BODY$
    BEGIN
        delete from ranking where old.id = id;
        return old;
    END;
$BODY$
  LANGUAGE plpgsql VOLATILE
  COST 100;
```

Stored Procedures

1. Number Player By Country

Purpose

This procedure is used to find the total players of all the countries.

Query

```
CREATE OR REPLACE FUNCTION number_player_by_country()
  RETURNS SETOF number_player_by_country AS
$BODY$
   select count(player.*) ||' from "' || country.name
   from player, country
   where id_country = country.id
   group by country.name
   $BODY$
  LANGUAGE sql VOLATILE
  COST 100
  ROWS 1000;
```

Output

```
select number_player_by_country()
```

Data Output | Explain | Messages | History

	number_player_by_country number_player_by_country
1	("1 from ""venezuela")
2	("1 from ""italy")
3	("1 from ""spain")
4	("1 from ""colombia")
5	("1 from ""canada")
6	("4 from ""japan")
7	("3 from ""china")
8	("1 from ""usa")
9	("4 from ""south korea"
10	("1 from ""south africa

2. Ranking By Event Id

Purpose

This procedure is used to see the ranking when the id of the event is given

Query

```
CREATE OR REPLACE FUNCTION ranking_by_event_id(x integer)
  RETURNS SETOF ranking_by_event_id AS
$BODY$
   select player.fullname||' from "' || clan.name ||'" with: '|| points
   from player, clan, player_clan, event, ranking, ranking_event
   where event.id = ranking.id and
   player.id = id_player and
   clan.id = id_clan and
   player_clan.id = id_player_clan and
   ranking.id = id_ranking and
   event.id = ranking_event.id_event and
   event.id = $1
   order by points desc
   $BODY$
  LANGUAGE sql VOLATILE
  COST 100
  ROWS 1000;
```

Output

```
select ranking_by_event_id(1)
```

Data Output | Explain | Messages | History

	ranking_by_event_id ranking_by_event_id
1	("Janssen David Adam from ""clan asia"" with: 989")
2	("Sutherland Daniel David from ""clan asia"" with: 956")
3	("Kaye Sharlene Susan from ""clan asia"" with: 876")
4	("Munro Richard from ""clan asia"" with: 796")
5	("Tompkins Robyn from ""clan asia"" with: 768")
6	("Zhao Yong from ""clan asia"" with: 678")
7	("Thompson Kevin Michael from ""clan asia"" with: 569")
8	("Hoffman Mark John from ""clan asia"" with: 378")
9	("Andrews Peter Ian from ""cyborg"" with: 378")
10	("Maher Lisa from ""clan asia"" with: 345")
11	("Murrell George Anthony Calvert from ""clan asia"" with: 234")
12	("Douglas Kurt John from ""clan asia"" with: 112")

SECURITY

The admin who can change, update, and maintain the database.

```
CREATE ROLE admin

GRANT SELECT, INSERT, UPDATE, ALTER

ON ALL TABLES IN SCHEMA PUBLIC

TO admin
```

KNOWN PROBLEMS

It is known that this database design possesses the following problems:

1. Reports about the number of players belong to particular continents, countries may or may not be desired.
2. In the real scenario it is difficult to collect the entire information regarding the events, players and clans.
3. The constraints between these relations were numerous and complicated some were easy to resolve ant other constraints were much more complicated.
4. Difficulties arises while the player clans signed for more than one event.

FUTURE ENHANCEMENTS

Some features and functionalities that might be desirable in the future:

1. Separation of security permissions between players, events and clans.
2. Organizing games effectively and creating more games to attract many players.
3. Should be more care full when the ne games are introduced in the events.