A Project Report On FIFA World Cup Management System

Developed By:

IT075- Nasit Nirj Hareshbhai

IT080- Parekh Bhavya Anilkumar

IT082- Parikh Aryan Paragkumar

Guided By

Internal Guide:

Prof. Ravindra A. Vyas

Department Of Information Technology

Faculty of Technology

DD University

Department of Information Technology Faculty of Technology,

Dharmsinh Desai University College Road,Nadiad-387001

March-2023

INDEX

I.Certificate.....I

II. Commendation II

1.SYSTEM OVERVIEW 4

a. Current system.....4

b. Advantages of the Proposed system (over current).....4

2.E-R Diagram.....5

3.Relaition Schema.....6

4.Data DICTIONARY.....7

5.DATABASE IMPLEMENTATION.....12

5.1 Create Schema 12

5.2 Insert Data values.....15

5.3 Queries.....20

5.4Queries (Based on Joins & Sub-Queries).....20

5.5Functions & Triggers.....26

5.6Cursors.....30

Certificate

This is certify that the project entitled “FIFA World Cup Management System” is a bonafide report of the work carried out by

1**) Nasit Nirj** Student Id:21ITUOS111

2**) Parekh Bhavya** Student Id:21ITUOS040

3) **Parikh Aryan** Student Id:21ITUOS025

Of Department of Infotmation Technology , Semester IV,under the guidance and supervision for the subject Database Management System.They were involved in Project training during the academic year 2022-2023.

Prof.Ravindra A. Vyas

Project Guide,Department Of Information Technology,

Faculty of Technology,

Dharmsinh Desai University,Nadiad

Date:13/03/2023

Prof.Vipul Dabhi

Head,Department Of Information Technology

1.SYSTEM OVERVIEW

Data is the code word of the computer industry. Data refers to a collection of factsusually collected as a result of observation and experiment or processes within acomputer system. This may consist of numbers, words or images or observationsof a set of variables. Data are often viewed as a lowest level of abstraction fromwhich information and knowledge are derived.

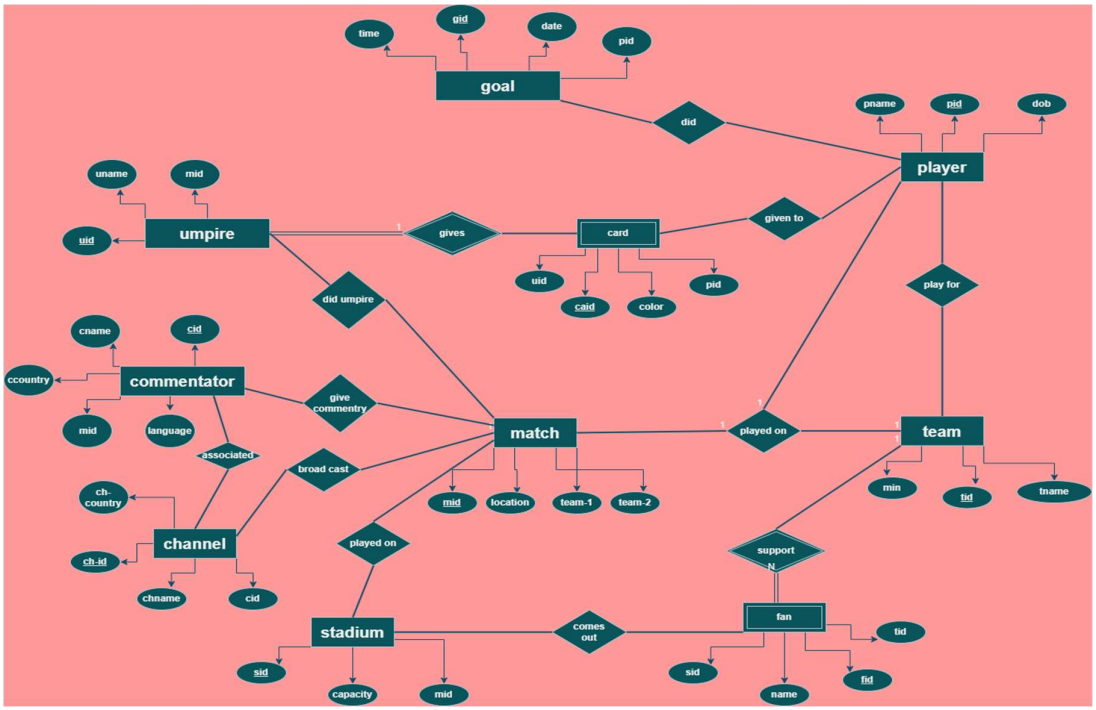
FIFA World Cup System is a system for football world cup that can provide all the information about the world cup management. With this system management and public can get the brief information about different entities like Cards(Yellow and Red) ,Goal(time and date),Player as well as umpire and etc.

1.1 ADVANTAGES OF THE PROPOSED SYSTEM

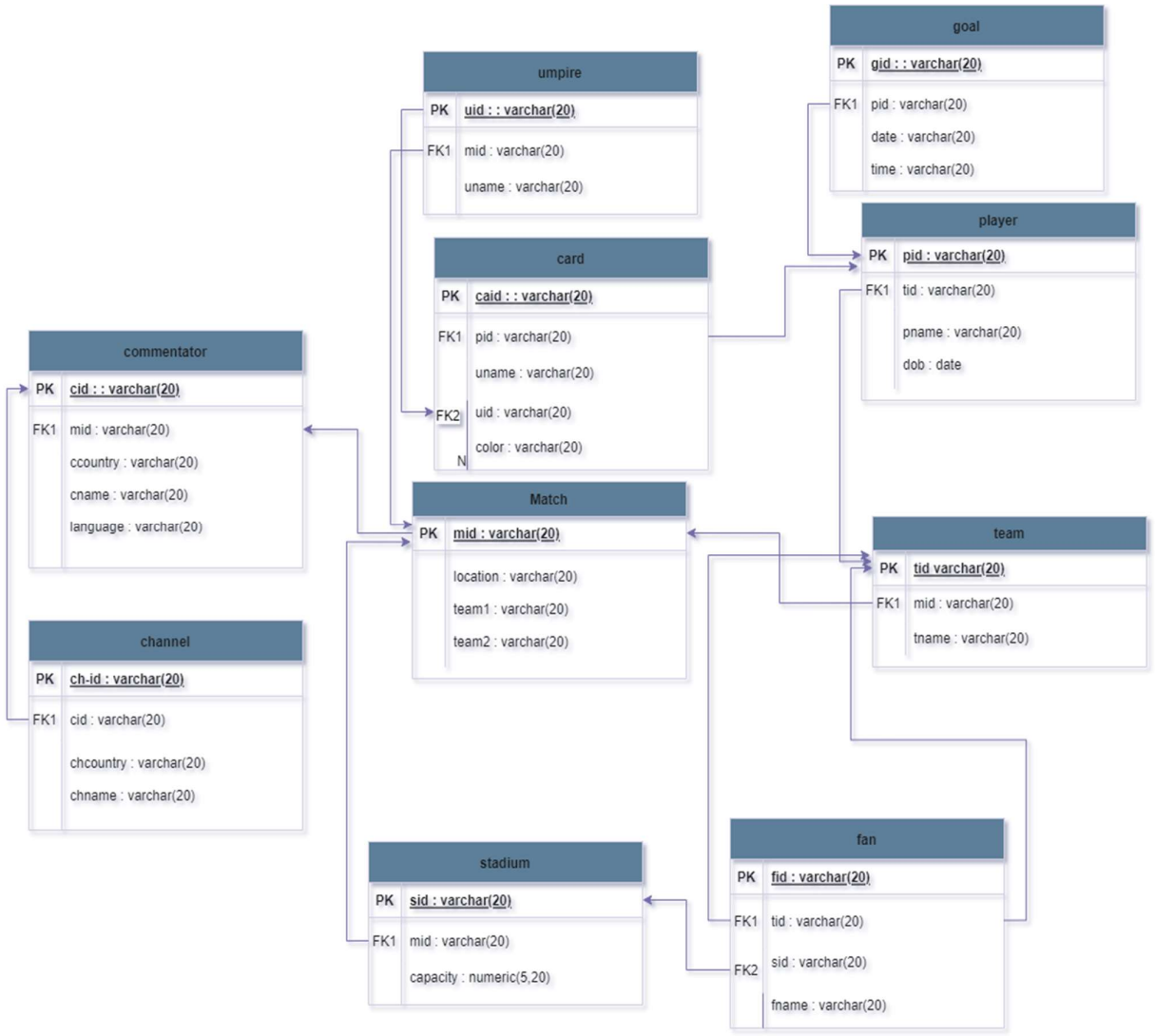
1.Infrastructure improvement can attract more people to the sport, as well as clubs can now offer better training facilities to their players. This becomes more vital especially in countries where local economy is not so strong to support huge investments in football.

3. he game is the chief form of entertainment in many countries across the world. Also, the game provides the host country with a lot of pride and publicity.

2. E-R DIAGRAM


goal
gid
time
date
pid
did
player
pname
pid
dob
umpire
uname
mid
uid
gives
card
given to
uid
caid
color
pid
did umpire
play for
commentator
cname
cid
ccountry
mid
language
give commentry
associated
broad cast
match
played on
team
mid
location
team-1
team-2
min
tid
tname
channel
ch-country
ch-id
chname
cid
played on
support
N
fan
tid
sid
name
fid
stadium
comes out
sid
capacity
mid

3.SCHEMA RELATION



4.DATA DICTIONARY

4.1 Umpire

```
postgres=# \d umpire;
                    Table "public.umpire"
 Column |         Type          | Collation | Nullable | Default
--------+-----------------------+-----------+----------+---------
 uid    | character varying(20) |           | not null |
 uname  | character varying(20) |           |          |
 min    | numeric(10,0)         |           |          |
Indexes:
    "umpire_pkey" PRIMARY KEY, btree (uid)
Foreign-key constraints:
    "umpire_min_fkey" FOREIGN KEY (min) REFERENCES match(min)
Referenced by:
    TABLE "card" CONSTRAINT "card_uid_fkey" FOREIGN KEY (uid) REFERENCES umpire(uid)
```

4.2 Card

```
postgres=# \d card;
                    Table "public.card"
 Column |         Type          | Collation | Nullable | Default
--------+-----------------------+-----------+----------+---------
 caid   | character varying(20) |           | not null |
 pid    | character varying(20) |           |          |
 uid    | character varying(20) |           |          |
 color  | character varying(20) |           |          |
Indexes:
    "card_pkey" PRIMARY KEY, btree (caid)
Foreign-key constraints:
    "card_pid_fkey" FOREIGN KEY (pid) REFERENCES player(pid)
    "card_uid_fkey" FOREIGN KEY (uid) REFERENCES umpire(uid)
```

4.3 Goal

```
postgres=# \d goal;
                     Table "public.goal"
 Column |         Type          | Collation | Nullable | Default
--------+-----------------------+-----------+----------+---------
 gid    | character varying(20) |           | not null |
 time   | character varying(20) |           |          |
 date   | date                  |           |          |
 pid    | character varying(20) |           |          |
Indexes:
    "goal_pkey" PRIMARY KEY, btree (gid)
Foreign-key constraints:
    "goal_pid_fkey" FOREIGN KEY (pid) REFERENCES player(pid)
```

4.4 Player

```
postgres=# \d player;
                    Table "public.player"
 Column |         Type          | Collation | Nullable | Default
--------+-----------------------+-----------+----------+---------
 plname | character varying(20) |           |          |
 pid    | character varying(20) |           | not null |
 dob    | date                  |           |          |
 tid    | character varying(20) |           |          |
Indexes:
    "player_pkey" PRIMARY KEY, btree (pid)
Foreign-key constraints:
    "player_tid_fkey" FOREIGN KEY (tid) REFERENCES team(tid)
Referenced by:
    TABLE "card" CONSTRAINT "card_pid_fkey" FOREIGN KEY (pid) REFERENCES player(pid)
    TABLE "goal" CONSTRAINT "goal_pid_fkey" FOREIGN KEY (pid) REFERENCES player(pid)
```

4.5 Team

```
postgres=# \d team;
                     Table "public.team"
 Column |         Type          | Collation | Nullable | Default
--------+-----------------------+-----------+----------+---------
 tid    | character varying(20) |           | not null |
 min    | numeric(10,0)         |           |          |
 tname  | character varying(20) |           |          |
Indexes:
    "team_pkey" PRIMARY KEY, btree (tid)
Foreign-key constraints:
    "team_min_fkey" FOREIGN KEY (min) REFERENCES match(min)
Referenced by:
    TABLE "fan" CONSTRAINT "fan_tid_fkey" FOREIGN KEY (tid) REFERENCES team(tid)
    TABLE "player" CONSTRAINT "player_tid_fkey" FOREIGN KEY (tid) REFERENCES team(tid)
```

4.6 Fan

```
postgres=# \d fan;
                      Table "public.fan"
 Column |         Type          | Collation | Nullable | Default
--------+-----------------------+-----------+----------+---------
 fname  | character varying(20) |           |          |
 fid    | character varying(20) |           | not null |
 sid    | character varying(20) |           |          |
 tid    | character varying(20) |           |          |
Indexes:
    "fan_pkey" PRIMARY KEY, btree (fid)
Foreign-key constraints:
    "fan_sid_fkey" FOREIGN KEY (sid) REFERENCES stadium(sid)
    "fan_tid_fkey" FOREIGN KEY (tid) REFERENCES team(tid)
```

4.7 Commentator

```
postgres=# \d commentrator;
                  Table "public.commentrator"
  Column  |         Type          | Collation | Nullable | Default
----------+-----------------------+-----------+----------+---------
 cid      | character varying(20) |           | not null |
 min      | numeric(10,0)         |           |          |
 country  | character varying(20) |           |          |
 cname    | character varying(20) |           |          |
 language | character varying(20) |           |          |
Indexes:
    "commentrator_pkey" PRIMARY KEY, btree (cid)
Foreign-key constraints:
    "commentrator_min_fkey" FOREIGN KEY (min) REFERENCES match(min)
Referenced by:
    TABLE "channel" CONSTRAINT "channel_cid_fkey" FOREIGN KEY (cid) REFERENCES commentrator(cid)
```

4.8 Channel

```
postgres=# \d channel;
                     Table "public.channel"
  Column   |         Type          | Collation | Nullable | Default
-----------+-----------------------+-----------+----------+---------
 chid      | character varying(20) |           | not null |
 chcountry | character varying(20) |           |          |
 cid       | character varying(20) |           |          |
 chname    | character varying(20) |           |          |
Indexes:
    "channel_pkey" PRIMARY KEY, btree (chid)
Foreign-key constraints:
    "channel_cid_fkey" FOREIGN KEY (cid) REFERENCES commentrator(cid)
```

4.9 Match

```
postgres=# \d match;
                    Table "public.match"
  Column  |         Type          | Collation | Nullable | Default
----------+-----------------------+-----------+----------+---------
 team1    | character varying(20) |           |          |
 team2    | character varying(20) |           |          |
 min      | numeric(10,0)         |           | not null |
 location | character varying(20) |           |          |
Indexes:
    "match_pkey" PRIMARY KEY, btree (min)
Referenced by:
    TABLE "commentrator" CONSTRAINT "commentrator_min_fkey" FOREIGN KEY (min) REFERENCES match(min)
    TABLE "stadium" CONSTRAINT "stadium_min_fkey" FOREIGN KEY (min) REFERENCES match(min)
    TABLE "team" CONSTRAINT "team_min_fkey" FOREIGN KEY (min) REFERENCES match(min)
    TABLE "umpire" CONSTRAINT "umpire_min_fkey" FOREIGN KEY (min) REFERENCES match(min)
```

4.10 Stadium

```
postgres=# \d stadium;
                      Table "public.stadium"
     Column     |          Type          | Collation | Nullable | Default
----------------+------------------------+-----------+----------+---------
 sid            | character varying(20) |           | not null |
 min            | numeric(10,0)         |           |          |
 capacityinlakh | numeric(15,0)         |           |          |
Indexes:
    "stadium_pkey" PRIMARY KEY, btree (sid)
Foreign-key constraints:
    "stadium_min_fkey" FOREIGN KEY (min) REFERENCES match(min)
Referenced by:
    TABLE "fan" CONSTRAINT "fan_sid_fkey" FOREIGN KEY (sid) REFERENCES stadium(sid)
```

5. DATA IMPLEMENTATION

A)SCHEMA

5.1.1 Match

create table match(team1 varchar(20) , team2 varchar(20) , min numeric(10) primary key , location varchar(20));

5.1.2 Stadium

create table stadium(sid numeric(10) primary key , min numeric(10) references match , capacityinlakh numeric(15));

5.1.3 Commentator

CREATE TABLE COMMENTRATOR(CID VARCHAR(20) PRIMARY KEY , MIN NUMERIC(10) REFERENCES MATCH , COUNTRY VARCHAR(20) , CNAME VARCHAR(20) , LANGUAGE VARCHAR(20));

5.1.4 Channel

CREATE TABLE CHANNEL(CHID VARCHAR(20) PRIMARY KEY , CHCOUNTRY VARCHAR(20) , CID VARCHAR(20) REFERENCES COMMENTRATOR , CHNAME VARCHAR(20));

5.1.5 Fan

CREATE TABLE FAN(FNAME VARCHAR(20),FID VARCHAR(20) PRIMARY KEY,SID VARCHAR(20) REFERENCES STADIUM,TID VARCHAR(20) REFERENCES TEAM);

5.1.6 Umpire

create table umpire(uid varchar(20) primary key , uname varchar(20) , min numeric(10) references match);

5.1.7 Player

create table player(plname varchar(20) , pid varchar(20) primary key , dob date);

5.1.8 Goal

create table goal(gid varchar(20) primary key,time varchar(20),date date,pid varchar(20) references player);

5.1.9 Card

create table card(caid varchar(20) primary key,pid varchar(20) references player,uid varchar(20) references umpire,color varchar(20));

B)DATA INSERTION

5.2.1 Match

```
insert into match values('india','china',1,'berlin');
insert into match values('argentina','brazil',2,'paris');
insert into match values('psg','barcelona',3,'lusain');
insert into match values('france','portugal',4,'al khor');
insert into match values('australia','england',5,'al rayyan');
insert into match values('landmark','soude arab',6,'rio');
insert into match values('maxico','netherland',7,'moscow');
insert into match values('poland','spain',8,'jonisberg');
insert into match values('tunisia','kruesia',9,'solma');
insert into match values('costa rica','italy',9,'rome');
insert into match values('costa rica','italy',10,'rome');
```

5.2.2 Stadium

```
insert into stadium values('S001',1 , 2);
insert into stadium values('S002',2 , 4);
insert into stadium values('S003',3 , 9);
insert into stadium values('S004',4 , 7);
insert into stadium values('S005',5 ,16);
insert into stadium values('S006',6 ,27);
insert into stadium values('S007',7 ,30);
```

```
insert into stadium values('S008',8 ,20);

insert into stadium values('S009',9 ,22);

insert into stadium values('S010',10 ,33);
```

5.2.3 Commentator

```
INSERT INTO COMMENTRATOR  VALUES
('C001',1,'INDIA','JAY','HINDI');

INSERT 0 1

INSERT INTO COMMENTRATOR  VALUES
('C002',2,'INDONESIA','LAURA','SPANISH');

INSERT 0 1

 INSERT INTO COMMENTRATOR  VALUES
('C003',3,'JAPAN','ZURUKU','JAPANESE');

INSERT 0 1

INSERT INTO COMMENTRATOR  VALUES
('C004',4,'RUSIA','SELENA','RUSSIAN');

INSERT 0 1

INSERT INTO COMMENTRATOR  VALUES
('C005',5,'USA','JOHN','ENGLISH');

INSERT 0 1

 INSERT INTO COMMENTRATOR  VALUES
('C006',6,'CANADA','JIM','CANADIAN');

INSERT 0 1
```

```
INSERT INTO COMMENTRATOR  VALUES
('C007',7,'FRENCH','SWET','FRENCH');

INSERT 0 1

 INSERT INTO COMMENTRATOR  VALUES
('C008',8,'ITALY','KINZER','ITALIAN');

INSERT 0 1

INSERT INTO COMMENTRATOR  VALUES
('C009',9,'MAXICO','ARTIK','MAXICAN');

INSERT 0 1

INSERT INTO COMMENTRATOR  VALUES
('C010',10,'AUSTRALIA','MERRY','AUSTRALIAN');

INSERT 0 1

 SELECT * FROM COMMENTRATOR;
```

5.2.4 Channel

```
INSERT INTO CHANNEL
VALUES('CH001','INDIA','C001','RUDRAX');

INSERT 0 1

INSERT INTO CHANNEL
VALUES('CH002','USA','C002','NIRAV');

INSERT 0 1

INSERT INTO CHANNEL
VALUES('CH003','QUTAR','C003','ARYAN');
```

INSERT 0 1

INSERT INTO CHANNEL
VALUES('CH004','CHINA','C004','BHAVYA');

INSERT 0 1

INSERT INTO CHANNEL
VALUES('CH005','KOREA','C005','NIRJ');

INSERT 0 1

INSERT INTO CHANNEL
VALUES('CH006','RUSSIA','C006','DEVEN');

INSERT 0 1

5.2.5 Fan

INSERT INTO FAN VALUEs('aryan','f001','s001','t001');

INSERT INTO FAN VALUEs('aryan','F001','S001','T001');

INSERT 0 1

INSERT INTO FAN VALUEs('BHAVYA','F002','S002','T002');

INSERT 0 1

INSERT INTO FAN VALUEs('NIRJ','F003','S003','T003');

INSERT 0 1

INSERT INTO FAN VALUEs('DEVEN','F004','S004','T004');

INSERT 0 1

INSERT INTO FAN VALUEs('DEV','F005','S005','T005);

INSERT INTO FAN VALUEs('DEV','F005','S005','T005');

LINE 2: INSERT INTO FAN VALUEs('DEV','F005','S005','T005');

INSERT INTO FAN VALUEs('DEV','F005','S005','T005');

INSERT 0 1

INSERT INTO FAN VALUEs('dax','F006','S006','T006');

INSERT 0 1

INSERT INTO FAN VALUEs('meet','F007','S007','T007');

INSERT 0 1

INSERT INTO FAN VALUEs('het','F008','S008','T008');

INSERT 0 1

INSERT INTO FAN VALUEs('mann','F009','S009','T009');

INSERT 0 1

INSERT INTO FAN VALUEs('om','F010','S010','T010');

INSERT 0 1

5.2.6 Umpire

insert into umpire values('u001','avi',1);

INSERT 0 1

insert into umpire values('u002','meet',2);

INSERT 0 1

insert into umpire values('u003','methali',3);

INSERT 0 1

insert into umpire values('u004','nir',4);

INSERT 0 1

insert into umpire values('u005','het',5);

INSERT 0 1

select * from umpire;

5.2.7 Player

insert into player values('messi','p001','30-7-1960','T001');

INSERT 0 1

insert into player values('m BAPPE','p002','4-8-1968','T002');

INSERT 0 1

insert into player values('RONALDO','p003','14-8-1988','T003');

INSERT 0 1

insert into player values('NEYMAR','p004','24-8-1998','T004');

INSERT 0 1

insert into player values('PELE','p005','1-1-1966','T005');

INSERT 0 1

insert into player values('KEVIN','p006','4-10-1968','T006');

INSERT 0 1

insert into player values('MOHAMMAD','p007','4-11-1968','T007');

INSERT 0 1

insert into player values('SEDIO','p008','22-8-1968','T008');

INSERT 0 1

insert into player values('SUNIL','p009','22-8-1968','T009');

INSERT 0 1

insert into player values('ALFERO','p010','21-12-1968','T010');

INSERT 0 1

SELECT * FROM PLAYER;

5.2.8 Goal

insert into goal values('g001','12:45pm','12-jan-2021','p001');

INSERT 0 1

insert into goal values('g002','2:00pm','20-jan-2021','p002');

INSERT 0 1

insert into goal values('g003','8:00am','20-feb-2021','p005');

INSERT 0 1

insert into goal values('g004','8:00pm','20-feb-2021','p006');

INSERT 0 1

insert into goal values('g005','8:36pm','18-march-2021','p004');

INSERT 0 1

insert into goal values('g006','9:36pm','6-march-2021','p002');

INSERT 0 1

insert into goal values('g007','10:36pm','6-may-2022','p009');

INSERT 0 1

insert into goal values('g008','10:00am','29-may-2022','p007');

INSERT 0 1

insert into goal values('g009','1:00am','29-may-2022','p008');

INSERT 0 1

insert into goal values('g0010','4:00pm','29-jun-2022','p003');

INSERT 0 1

5.2.9 Card

insert into card values('ca001','p001','u001','yellow');

INSERT 0 1

insert into card values('ca002','p006','u002','yellow');

INSERT 0 1

insert into card values('ca003','p009','u003','red');

INSERT 0 1

insert into card values('ca004','p010','u004','yellow');

INSERT 0 1

insert into card values('ca005','p005','u005','yellow');

INSERT 0 1

Insertion output:

5.3.1 Match:

```
postgres=# select * from match;
   team1    |   team2    | min | location
------------+------------+-----+-----------
 india      | china      |   1 | berlin
 argentina  | brazil     |   2 | paris
 psg        | barcelona  |   3 | lusain
 france     | portugal   |   4 | al khor
 australia  | england    |   5 | al rayyan
 landmark   | soude arab |   6 | rio
 maxico     | netherland |   7 | moscow
 poland     | spain      |   8 | jonisberg
 tunisia    | kruesia    |   9 | solma
 costa rica | italy      |  10 | rome
(10 rows)
```

5.3.2 Commentrator:

```
postgres=# select * from commentrator;
 cid  | min |  country  | cname  |  language
------+-----+-----------+--------+------------
 C001 |   1 | INDIA     | JAY    | HINDI
 C002 |   2 | INDONESIA | LAURA  | SPANISH
 C003 |   3 | JAPAN     | ZURUKU | JAPANESE
 C004 |   4 | RUSIA     | SELENA | RUSSIAN
 C005 |   5 | USA       | JOHN   | ENGLISH
 C006 |   6 | CANADA    | JIM    | CANADIAN
 C007 |   7 | FRENCH    | SWET   | FRENCH
 C008 |   8 | ITALY     | KINZER | ITALIAN
 C009 |   9 | MAXICO    | ARTIK  | MAXICAN
 C010 |  10 | AUSTRALIA | MERRY  | AUSTRALIAN
(10 rows)
```

5.3.3 Channel:

```
postgres=# select * from channel;
 chid  | chcountry | cid  | chname
-------+-----------+------+--------
 CH001 | INDIA     | C001 | RUDRAX
 CH002 | USA       | C002 | NIRAV
 CH003 | QUTAR     | C003 | ARYAN
 CH004 | CHINA     | C004 | BHAVYA
 CH005 | KOREA     | C005 | NIRJ
 CH006 | RUSSIA    | C006 | DEVEN
(6 rows)
```

5.3.4 Stadium:

```
postgres=# select * from stadium;
 sid  | min | capacityinlakh
------+-----+----------------
 S001 |   1 |              2
 S002 |   2 |              4
 S003 |   3 |              9
 S004 |   4 |              7
 S005 |   5 |             16
 S006 |   6 |             27
 S007 |   7 |             30
 S008 |   8 |             20
 S009 |   9 |             22
 S010 |  10 |             33
(10 rows)
```

5.3.5 Fan:

```
postgres=# select * from fan;
 fname  | fid  | sid  | tid
--------+------+------+------
 aryan  | F001 | S001 | T001
 BHAVYA | F002 | S002 | T002
 NIRJ   | F003 | S003 | T003
 DEVEN  | F004 | S004 | T004
 DEV    | F005 | S005 | T005
 dax    | F006 | S006 | T006
 meet   | F007 | S007 | T007
 het    | F008 | S008 | T008
 mann   | F009 | S009 | T009
 om     | F010 | S010 | T010
(10 rows)
```

5.3.6 Team:

```
postgres=# select * from team;
 tid  | min |   tname
------+-----+-----------
 T001 |   1 | RUSSIA
 T002 |   2 | INDIA\
 T003 |   3 | ITALY
 T004 |   4 | GERMANY
 T005 |   5 | AUSTRALIA
 T006 |   6 | USA
 T007 |   7 | LANDMARK
 T008 |   8 | SPAIN
 T009 |   9 | POLLAND
 T010 |  10 | SCOTLAND
(10 rows)
```

5.3.7 Player:

```
postgres=# select * from player;
  plname   | pid  |    dob     | tid
-----------+------+------------+------
 messi     | p001 | 1960-07-30 | T001
 m BAPPE   | p002 | 1968-08-04 | T002
 RONALDO   | p003 | 1988-08-14 | T003
 NEYMAR    | p004 | 1998-08-24 | T004
 PELE      | p005 | 1966-01-01 | T005
 KEVIN     | p006 | 1968-10-04 | T006
 MOHAMMAD  | p007 | 1968-11-04 | T007
 SEDIO     | p008 | 1968-08-22 | T008
 SUNIL     | p009 | 1968-08-22 | T009
 ALFERO    | p010 | 1968-12-21 | T010
(10 rows)
```

5.3.8 Goal:

```
postgres=# select * from goal;
  gid  |  time   |    date    | pid
-------+---------+------------+------
 g001  | 12:45pm | 2021-01-12 | p001
 g002  | 2:00pm  | 2021-01-20 | p002
 g003  | 8:00am  | 2021-02-20 | p005
 g004  | 8:00pm  | 2021-02-20 | p006
 g005  | 8:36pm  | 2021-03-18 | p004
 g006  | 9:36pm  | 2021-03-06 | p002
 g007  | 10:36pm | 2022-05-06 | p009
 g008  | 10:00am | 2022-05-29 | p007
 g009  | 1:00am  | 2022-05-29 | p008
 g0010 | 4:00pm  | 2022-06-29 | p003
(10 rows)
```

5.3.9 Umpire:

```
postgres=# select * from umpire;
 uid  |  uname   | min
------+---------+-----
 u001 | avi     |   1
 u002 | meet    |   2
 u003 | methali |   3
 u004 | nir     |   4
 u005 | het     |   5
(5 rows)
```

5.3.10 Card:

```
postgres=# select * from card;
 caid  | pid  | uid  | color
-------+------+------+--------
 ca001 | p001 | u001 | yellow
 ca002 | p006 | u002 | yellow
 ca003 | p009 | u003 | red
 ca004 | p010 | u004 | yellow
 ca005 | p005 | u005 | yellow
(5 rows)
```

5.4 QUERIES USING BASIC DBMS CONSTRUCTS

JOIN & SUBQUERIES:

5.4.1 give the detail of channel with commentator.

```
postgres=# select chid,chcountry,chname,cname from channel inner join commentrator on channel.ci
 chid  | chcountry | chname | cname
-------+-----------+--------+--------
 CH001 | INDIA     | RUDRAX | JAY
 CH002 | USA       | NIRAV  | LAURA
 CH003 | QUTAR     | ARYAN  | ZURUKU
 CH004 | CHINA     | BHAVYA | SELENA
 CH005 | KOREA     | NIRJ   | JOHN
 CH006 | RUSSIA    | DEVEN  | JIM
(6 rows)
```

5.4.2 give name of commentator who give comentry on match number 2.

```
postgres=# select * from commentrator where min=2;
 cid  | min |  country  | cname | language
------+-----+-----------+-------+----------
 C002 |   2 | INDONESIA | LAURA | SPANISH
(1 row)
```

5.4.3 give all the match location who has played on stadium which has stadium capacity more than 16 lack.

```
postgres=# select location from match inner join stadium on stadium.min=match.min;
 location
-----------
 berlin
 paris
 lusain
 al khor
 al rayyan
 rio
 moscow
 jonisberg
 solma
 rome
(10 rows)
```

5.4.4 give the both team name of match where commentrator give comentry whose country is Japan.

```
postgres=# select team1,team2 from match where min in (select min from commentrator where country='JAPAN')
 team1 |   team2
-------+-----------
 psg   | barcelona
(1 row)
```

5.4.5 give capacity of stadium whose team 1 is France and Portugal.

```
postgres=# select capacityinlakh from stadium where min in (select min from match where team1='france' and team2='portugal
 capacityinlakh
----------------
              7
(1 row)
```

5.4.6 give the time and date of goal which is done by the player who receive red card.

```
postgres=# select time,date from goal where pid in(select pid from card where color='red');
  time   |    date
---------+------------
 10:36pm | 2022-05-06
(1 row)
```

5.4.7 give the information on goal.

```
postgres=# select * from card ;
 caid  | pid  | uid  | color
-------+------+------+--------
 ca001 | p001 | u001 | yellow
 ca002 | p006 | u002 | yellow
 ca003 | p009 | u003 | red
 ca004 | p010 | u004 | yellow
 ca005 | p005 | u005 | yellow
(5 rows)
```

5.4.8 give the match number where this umpire take umpiring who give red card to player.

```
postgres=# select min from umpire where uid in (select uid from card where color='red');
 min
-----
   3
(1 row)
```

5.4.9 give the date of birth of player whose player id is T001.

```
postgres=# select dob from player where tid in(select tid from team where tid='T001');
    dob
------------
 1960-07-30
(1 row)
```

5.4.10 give the name of player whose date of birth id 30 jan-1960.

```
postgres=# select plname from player where dob='30-07-1960';
 plname
--------
 messi
(1 row)
```

5.5 FUNCTION && TRIGGER

5.5.1 create a trigger to get new stadium capacity input which not be less than 60 in lakh

Function:

```
create function check_cap() returns trigger as $$
BEGIN
if NEW.capacityinlakh<60 then
raise exception 'capacity Must be greater than 60';
end if;
return NEW;
END;
$$
LANGUAGE plpgsql;
```

```
postgres=# create function check_cap() returns trigger as $$
postgres$# BEGIN
postgres$# if NEW.capacityinlakh<60 then
postgres$# raise exception 'capacity Must be greater than 60';
postgres$# end if;
postgres$# return NEW;
postgres$# END;
postgres$# $$
postgres-# LANGUAGE plpgsql;
CREATE FUNCTION
```

Trigger:

```
create trigger cap_check
BEFORE INSERT OR UPDATE
ON stadium
FOR EACH ROW
EXECUTE PROCEDURE check_cap()
;
```

```
postgres=# create trigger cap_check
postgres-# BEFORE INSERT OR UPDATE
postgres-# ON stadium
postgres-# FOR EACH ROW
postgres-# EXECUTE PROCEDURE check_cap()
postgres-# ;
CREATE TRIGGER
```

Check: insert into stadium values('s011',1,20);

```
postgres=# insert into stadium values('s011',1,20);
ERROR:  capacity Must be greater than 60
CONTEXT:  PL/pgSQL function check_cap() line 4 at RAISE
```

5.5.2 create function to count the total stadium whose capacity id given between specific value given by user.

Function:

```
CREATE function stadium_caprange(cap_i int, cap_f int)
returns int
language plpgsql
as
$$
declare
stadiumcount integer;
begin
select count(*)
into stadiumcount
from stadium
where capacityinlakh between cap_i and cap_f;
return stadiumcount;
end;
$$;
```

```
postgres=# CREATE function stadium_caprange(cap_i int, cap_f int)
postgres-# returns int
postgres-# language plpgsql
postgres-# as
postgres-# $$
postgres$# declare
postgres$# stadiumcount integer;
postgres$# begin
postgres$# select count(*)
postgres$# into stadiumcount
postgres$# from stadium
postgres$# where capacityinlakh between cap_i and cap_f;
postgres$# return stadiumcount;
postgres$# end;
postgres$# $$;
CREATE FUNCTION
```

Check: select stadium_caprange(2,20);

```
postgres=# select stadium_caprange(2,20);
 stadium_caprange
------------------
                6
(1 row)
```

5.5.3 create trigger to insert new commentator into the database which should not belong to the Japan.

Function:

```
CREATE function check_comm_country() returns trigger as $$
BEGIN
if(NEW.country='japan') then
raise exception 'commentrator must not belong to japan!';
end if;
return NEW;
end;
$$
LANGUAGE plpgsql;
```

```
postgres=# CREATE function check_comm_country() returns trigger as $$
postgres$# BEGIN
postgres$# if(NEW.country='japan') then
postgres$# raise exception 'commentrator must not belong to japan!';
postgres$# end if;
postgres$# return NEW;
postgres$# end;
postgres$# $$
postgres-# LANGUAGE plpgsql;
CREATE FUNCTION
```

Trigger:

create trigger country_check

BEFORE INSERT OR UPDATE

ON commentrator

FOR EACH ROW

EXECUTE PROCEDURE check_comm_country();

```
postgres=# CREATE function check_comm_country() returns trigger as $$
postgres$# BEGIN
postgres$# if(NEW.country='japan') then
postgres$# raise exception 'commentrator must not belong to japan!';
postgres$# end if;
postgres$# return NEW;
postgres$# end;
postgres$# $$
postgres-# LANGUAGE plpgsql;
CREATE FUNCTION
```

Check: insert into commentrator(cid,min,country,cname,language) values ('c011',11,'japan','scarlet','japanese');

```
postgres=# insert into commentrator(cid,min,country,cname,language) values
postgres-# ('c011',11,'japan','scarlet','japanese');
ERROR:  commentrator must not belong to japan!
CONTEXT:  PL/pgSQL function check_comm_country() line 4 at RAISE
```

5.6 CURSOR:

Create a cursor to select match id sratring from 4 ans fetch the next match id.

Cursor syntax:

begin;

BEGIN

declare my_match_id cursor for

select min from match where min > 4;

fetch next from my_match_id;

close my_match_id;

cursor;

```
postgres=# begin;
BEGIN
postgres=*# declare my_match_id cursor for
postgres-*# select min from match where min > 4;
DECLARE CURSOR
postgres=*# fetch next from my_match_id;
 min
-----
   5
(1 row)


postgres=*# close my_match_id;
CLOSE CURSOR
postgres=*#
```