ENES BAŞER - 2020400141

YASİN ATLI - 2020400246

Report - Part 2

Part-2 was relatively easier to implement than Part-1. Firstly, we went over generic SQL commands and features for more flexibility to handle edge cases or validations provided in homework description. Especially usage of PK , FK , CHECK constraints have been handy through PART-2 . As a group, we actually handled some parts which we cannot in ER diagram as much as possible.Let me break down logical db design part into branches. :

Let’s see tables :

```
CREATE TABLE user(
name varchar(30) PRIMARY KEY,
surname varchar(30) ,
password varchar(12),
username varchar(30)
);
CREATE TABLE jury(
name varchar(30) PRIMARY KEY,
nationality varchar(4) ,
UNIQUE (name,nationality),
FOREIGN KEY (name) REFERENCES user(name)
);
```

```
create table player(
name varchar(30) PRIMARY KEY,
height int ,
weight int,
date_of_birth date,
FOREIGN KEY (name) REFERENCES user(name)
);
```

```
CREATE table coach(
name varchar(30) PRIMARY KEY,
nationality varchar(4) ,
UNIQUE (name,nationality),
FOREIGN KEY (name) REFERENCES user(name));
```

We handled ISA hierarchy defining for each suentitiy with PK which is name of the user. MYSQL DBMS would throw an error if we defines an player,coach or jury without defining user as it should do. These cases also tested and verified.

```
CREATE TABLE positions(
id int PRIMARY KEY,
name varchar(50)
);
```

```
CREATE TABLE player_position (
player_name VARCHAR(30),
position_id INT,
PRIMARY KEY (player_name, position_id),
FOREIGN KEY (player_name) REFERENCES player(name),
FOREIGN KEY (position_id) REFERENCES positions(id)
);
```

Above table, we defined PK as pairs of name and position_id so that one player can have multiple roles such as (“ENES”,”A”),(“ENES”,”B”). If we would define PK as just player_name, one player couldn’t have multiple roles (capable of roles).

```
create table team(
id int PRIMARY KEY,
name varchar(50)
);
create table coach_directs_team(
coach_name varchar(50) Not null ,
start date,
finish date,
team_id int,
FOREIGN KEY (coach_name) references coach(name),
FOREIGN KEY (team_id) references team(id),
primary key(team_id,coach_name));
```

For team and coach relation (ER diagram handled this relation) we defined mapping table such that one team have to be directed ***exactly by one coach.*** That’s why PK declared as a pair of team_id and coach_name also coach_name cannot be null.

```
create table channel(
id int PRIMARY KEY,
name varchar(50),
UNIQUE (name,id)
);
create table team_contracted_channel(
team_id int PRIMARY KEY,
channel_id int NOT NULL,
start date,
finish date,
FOREIGN KEY(team_id) REFERENCES team(id),
FOREIGN KEY(channel_id) REFERENCES channel(id));
```

Again reference to ER diagram and description requirements there should be some mechanism for channel and team contract constraints. As indicated in description, channels are defined as unique pair of name and id. For relation between entities of channel and team , we defined team_id PK and channel_id “not null “ so for team_id there should be at least one channel contracted with. And we use references with FK

defined above such that one cannot add data to table of team_contracted_channel without define channel and team instances in corresponding tables.

```
CREATE TABLE stadium (
stadium_id INT PRIMARY KEY,
stadium_country VARCHAR(50),
stadium_name VARCHAR(50),
UNIQUE (stadium_id, stadium_country, stadium_name)
);
```

```
create table match_session(
session_id int primary key,
match_session_date date ,
rating double not null,
assigned_jury varchar(30) not null,
UNIQUE (session_id,assigned_jury),
time_slot ENUM('1', '2', '3', '4') not null,
match_stadium_id int not null,
UNIQUE(match_session_date,time_slot,match_stadium_id),
FOREIGN KEY (assigned_jury) REFERENCES jury(name),
FOREIGN KEY(match_stadium_id) REFERENCES stadium(stadium_id)
);
```

These two tables were the most time-consuming tables for us. Let's start with stadium.As expected stadium_id is PK. As identified in description stadium_id depends on country and name of it so we defined these fields as triplet . For other two tables, it would be helpful to focus on parts which are constraints. For example , in match_session table there is a field named rating not null meaning for every match session there should be rating. And along with rating there should be assigned_jury for rating process (reference to jury table as explained previously) which is also not null. And we defined UNIQUE (session_id,assigned_jury) so that assigned_jury can rate matches only once also match cannot be rated by more than 1 jury. And for time slots we defined enum as identified in description but actually, we couldn't cover the constraint which is the mechanism of slots like [2:3] etc.

Lastly, UNIQUE(match_session_date,time_slot,match_stadium_id) this specification handles the constraint of at the same time , matches cannot overlap.

```
create table player_match_matchsession(
player_name VARCHAR(30),
team_id int,
match_session_id int,
position_id INT,
PRIMARY KEY (player_name, match_session_id),
FOREIGN KEY (player_name) REFERENCES player(name),
FOREIGN key(team_id) REFERENCES team(id),
FOREIGN KEY(match_session_id) REFERENCES match_session(session_id),
FOREIGN KEY (player_name, position_id) REFERENCES
player_position(player_name, position_id)
);
```

For the last table of our database design(out of 13 table) about the constraint of player cannot play different roles ***during the match session.***(Not capable of). So we defined PK as PRIMARY KEY (player_name, match_session_id) so that player can play just one role among he/she is capable of. It would be more clear through an example, assume that there is data in this table like name:"test" match_session_id : 2 position : 3 and if one would try to add data again like "test", 2 , 4(which is position) which is not allowed because name and match_session_id declared as PK. And also we inject positions played by specific user through player_position table as pair so we are sure that player can play position which he/she is capable of. **Initially we thought that we can use CHECK constraint here with subquery to ensure that player can play position which he/she is capable of. However, MYSQL does not support subqueries in CHECK constraint.**
🔗*https://stackoverflow.com/a/9723670/22454238*

PARTS WE CANNOT COVER

- The duration of the match is closely related to the time slots. The time slot attribute determines the starting time of the match, and the end time is determined by the duration. Each match has a duration of 2 time slots. (For example, if a match starts at time slot 2 and has a duration of 2)
- Assigned jury for match session cannot edit/change match's rating. We thought that we can use CHECK constraint here. However, in order to use CHECK constraint one have to use know initial value of rating and embed it into table like CHECK rating = initial_rating (smth. like this.)

SOURCES AND LINKS :

For full code and diagrams related files.

🔗 **https://github.com/EnesBaserr/ER-Diagram-Homework**