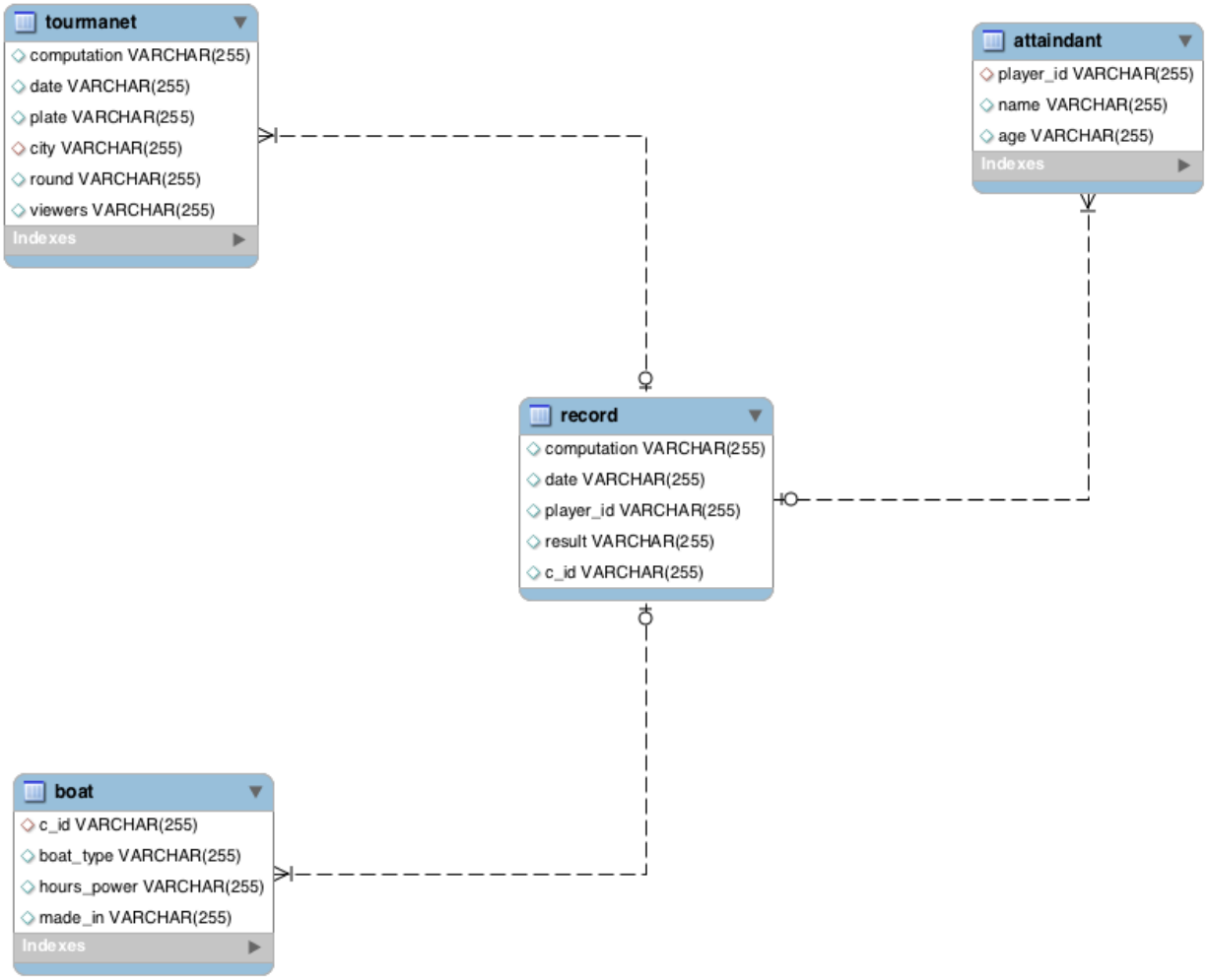
Programming Assignment 2 Report

Student(s):	Yohnnes Gebreyohannes Tesfagiorgis - yt222bn@student.lnu.se Earmyas Measho Bebre – eg223di@student.lnu.se

1. Project Idea

The main idea of this project is a boat race competition that records tournaments, records of computation, boats and attendants. We did not use realistic data from the internet. rather we inserted dummy data ourselves using the excel data generator. Our tool enables a user to track any data in the tables using queries and visualization of the statistical data

2. Schema Design



Tournament	Computation,	date	plate	city,	round,	viewers
Record	Computation	date	player_id	result,	c_id	
Boat	c_id,	boat_type	hours_power,	made_in		
Attendants	Player_id,	name,	age			

The relationship between the tables is as follow

a record has many tournaments

a Record has many boat

a Record has many attendants

As can be seen above, we have four tables. Namly tournament, record, boat and attendants. tournament with column names “computation(type of computation),date(date of computation),plate(id number of the tournament),city(city of competition),round(number of rounds),viewers(number of viewers)”

record has column names “computation(type of computation),date,player_id,result,c_id(unique boat id) “

The data from the boats table has column names ” c_id(unique boat id),boat_type,hours_power,made_in”.

Primary key for tournament is computation

Primary key for record is player_id, c_id, computation

Primary key for attendants is playes_id

Primary key for boats is c_id

3. SQL Queries

We create 5 different queries which queries in between at list two tables each. And utilize Join, aggregation, group by and more sql functions. We also use view in the 5 query.

1. Who has participated most:

   This sql query use two tables record (t1) and attindant using (t2) and make a subquery table by using INNER JOIN between the two tables t1 and t2 in such a way by counting the number of player_id (aggregation) repeated or group by player_id. Finally, get the maximum player_id repeated in the subquery.

```
SELECT t2.name, COUNT(t1.player_id) AS totalnumber
```

```
FROM record t1 INNER JOIN attaindant t2 ON t1.player_id = t2.player_id

GROUP BY t1.player_id ORDER BY totalnumber DESC LIMIT 1;
```

2. The youngest person that participate in the sailing race:

   The following query also uses two tables, record (t1) and attendant by inner join using their key/forign key player_id simultaneously. Then having the computation value ‘sailling’ and ordering by the minimum of age values, we can get the youngest person/ the lost age value.

```
SELECT t1. computation, t2.name FROM record t1 INNER JOIN attaindant t2 ON t1.player_id = t2.player_id HAVING t1.computation = 'sailing' ORDER BY min(t2.age);
```

3. Who won most from all races:

   Here we intend to get a person (name and number of participation) that participated in most of the boat racing. So We create a quarry that involves once again two tables, record t1 and attendant t2 and join them and evaluate with a condition where record result=1 while counting the number of presents and get the max (using DESC LIMIT).

```
SELECT t2.name, COUNT(t1.player_id) AS Total FROM record t1 INNER JOIN attendant t2 ON t2.player_id = t1.player_id WHERE result=1 GROUP BY t2.name ORDER BY Total DESC LIMIT 1;
```

4. Who has won the most amount of competition?

   In this query we used once again two different tables and joined them to get one subquery table in such a way counting the player participation in the race and sorting it in ascending order and getting first place.

```
SELECT t2.made_in, COUNT(t2.c_id) AS numberOfboats FROM record t1 LEFT JOIN boat t2 ON t1.c_id = t2.c_id GROUP BY t2.made_in ORDER BY numberOfboats DESC LIMIT 1;
```

5. Which country produced a boat that has 300 hp or more and achieved a good result ?

   In this query we use the view that expelled in section 6 and record table by inner joining them using as below

```
SELECT t1.result, t2.made_in, t2.hours_power FROM record t1 INNER JOIN
more_than_300HP_boats t2 On t1.c_id = t2.c_id GROUP BY t1.result ORDER BY
t1.result limit 1;
```

6. Create View:

This query creates a view (all boats having horsepower of 300 or more) or replace if it already exists. We use this view as helping to query on the 5th problem.

```
CREATE OR REPLACE VIEW more_than_300HP_boats AS select c_id, made_in
FROM boat where hours_power > 300;
```

4. Discussion and Resources

We have been trying with a lot of ideas but it has not been easy to land on one idea. We had issues with organizing the keys and getting a meaningful query. And connecting with the mysql with the mysql connector has been a big challenge. Not to mention writing the code. We used an object oriented approach to make our code more organized. Generally we dumme the assignment as a very challenging one. But finally here we are.

https://www.youtube.com/channel/UCYTg1Qz_xco1dNY2l32D38A

https://gitlab.com/yt222bn/1dv503_database_ass_3

Changelog

person	task	Date
Earmyas Measho Gebre	Setting-up server environment and Git repository	**2020-03-30**
Yohannes Gebreyohannes Tesfagiorgis	Implemented module for loading the data	**2020-04-30**
Earmyas measho gebre	Designing and implementing the home-page	**2020-04-13**
Earmyas Measho Gebre	Implementing authentication for youtube	**2020-04-15**
Yohannes Gebreyohannes Tesfagiorgis	Documented my changes/contributions in the assignment report	**2020-05-10**
Yohannes Gebreyohannes Tesfagiorgis	Implementing the code	**2020-05-21**