# Programming Assignment 2 Report

|  |  |
| --- | --- |
| Student(s): | Johan Gustafsson – jg223fp@student.lnu.se  Anas Hallak Mohamadanas hm222ua@student.lnu.se |

## Project Idea

We have developed a database for disc-golfers. It contains data about different discs, courses, players, and results. Some of the data is made up, for example competitions, players, and results. We have collected some data about different discs from a company named Latitude64 which produce a variety of different discs. The data about the courses is taken from a homepage called Udisc.

Our app lets you see what discs different players have in their possession. It can also tell you which discs would be good if you want to throw further than a specific distance depending on you level. Of course, it can also calculate the winner of competitions. Another nice feature is that it can tell which is the by players most owned disc. This could be of great in market research.

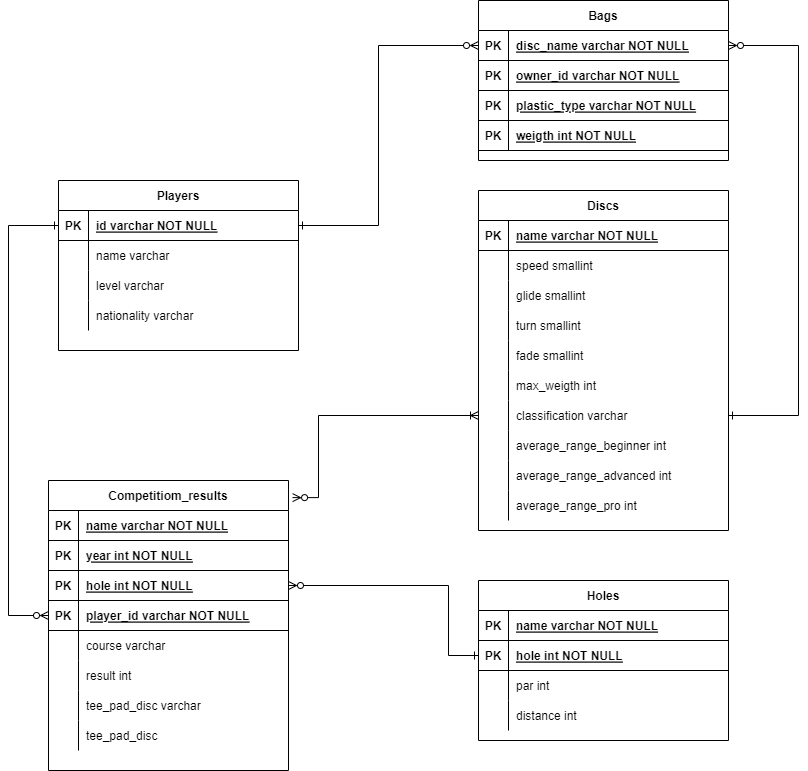
Data sources:

Latitude64: https://www.latitude64.se/disc-golf-products/golf-discs/

Teleborgs disc golf course: <https://udisc.com/courses/teleborgs-discgolfbana-i7Em>

Älmhults disc golf course: <https://udisc.com/courses/aelmhults-discgolfbana-eNDA>

## Schema Design



*Picture 1: Schema design for the disc golf database*

Tables explanations

Players

This table represents a disc golf player. The have a player id, a name, and a nationality. They are also ranked in different levels. The id is the primary key to identify a player.

Discs

This table contains facts about different disc models. They have different flying attributes which describes how they behave in the air. Those are speed, glide, turn, and fade. The discs are also classified for their intended usage in driver, fairway driver, midrange and putts. The table also provides data of the average throwing distance for each disc for each different player level. The primary key is the name of the disc model

Holes

A record in the holes table is a hole on a course. A course usually has 9 or 18 holes. Each hole has a distance and also a par result. A par is the result you are aiming to beat when playing the hole. To identify a specific hole the primary keys are the name of the course and the hole number.

Bags

The table “Bags” are actually discs that is owned by someone and are put in bags. The name can be a bit confusing but the table consist of discs with different owner id’s and some specific attributes which identifies the disc. Those attribute is plastic type and weigth. Discs are sold in different materials and can have different weigth depending on if you are strong or weak.

Competition\_results

The records in competition\_results are actually the result for each player for each hole in a competition. In each record we can also se what discs where used at throw from tee pad and the disc which was used when finishing the hole. The primary keys are name of competition, year, hole number, and player id. The course is not needed for primary key as a competition can be played on different courses.

Relations explanations

* A bagged disc has one owner id and one player can own zero or many discs.
* A competition result has one player and a player can have zero or many competition result.
* A competition result have one hole. A hole can have zero or many competition results.
* A competition result has one or many discs (disc used on tee pad and disc used to finish the hole) and a disc can be used in zero or many results.
* A disc in a bag is of one specific disc model and a disc model can exist in zero or many players bags.

## SQL Queries

Here you present and discuss the most interesting queries. Make sure you have 5 of them at least and check the specification in the assignment sheet. One example is found below:

Q: **List the name, last name and job title of the employee from a given city.**

The following query is a multirelation query and uses *JOIN.* We pass the argument of the city name (marked with ? in the query) and the query should give us all the employees of the corresponding shop. We join table *Employees* on table *Shops* by matching the *Shops.ID* to the foreign key *Employees.shopID*

SELECT firstname, lastname, jobtitle

FROM Employees

JOIN Shops ON Employees.shopID = Shops.ID

WHERE city=?;

Q: **Who is the winner of a given competition?**

The following query is multirelational and uses JOIN. The program will first send a query to list all competitions. Then it lets the user select a competition and pass the name (marked with ? in the query) and year (marked with X in the query) to it on the the following query. The query sums the result of the given competition and picks the one with the lowest result. Then it joins the result with the player table to get facts about the winner and return togheter with the result. Primary keys and foreign key

SELECT name,id,nationality,total FROM disc\_golf.players

JOIN

(SELECT

player\_id, SUM(result) AS total

FROM

disc\_golf.competition\_results

WHERE

year = X

AND competition\_results.name = ?

GROUP BY player\_id

ORDER BY total

LIMIT 1

) as winner ON players.id = winner.player\_id;

## Discussion and Resources

Here you can write anything you might think it is important and provide the link to the required resources. For example:

We had issues with the missing and inconsistent data. We decided to remove/insert NULLS in the missing/corrupted attributes/tuples…...

The project uses xyz library, please check readme.txt for installation details.

Source code: [github/... link]

Video demonstration: [youtube/vimeo/... link]

# Changelog

|  |  |  |
| --- | --- | --- |
| Person | Task | Date |
| Ilir | Setting-up server environment and Git repository | 2018-08-20 |
| Ilir | Implemented module for loading the data | 2018-08-20 |
| Maria | Designing and implementing the home-page | 2018-08-20 |
| Maria | Implementing Ouath authentication for Twitter | 2018-08-21 |
| Maria | Documented my changes/contributions in the assignment report | 2018-08-21 |
| Ilir | Documented my changes/contributions in the assignment report | 2018-08-21 |
|  |  |  |