|  |  |
| --- | --- |
|  | COS221 Practical 5 |
|  |  |
|  | Swing Steezy  June 2022 |

Contents

[Task 1: Research 2](#_Toc105514193)

[Overview 2](#_Toc105514194)

[Rules 2](#_Toc105514195)

[Terminology 2](#_Toc105514196)

[Game Structure 2](#_Toc105514197)

[Task 2: (E)ER Diagram 2](#_Toc105514198)

[Overview and Descriptions 2](#_Toc105514199)

[Version 1 5](#_Toc105514200)

[Final Version 5](#_Toc105514201)

[Task 3: (E)ER Diagram to Relational Mapping 6](#_Toc105514202)

[Task 4: Relational Exclusion 6](#_Toc105514203)

[SQL Statements 6](#_Toc105514204)

[Visual Database 10](#_Toc105514205)

[Task 6: Sample Data 10](#_Toc105514206)

[Task 7: Analyse and Optimise 10](#_Toc105514207)

[Query 1 10](#_Toc105514208)

[Query 2 10](#_Toc105514209)

[Query 3 11](#_Toc105514210)

# Task 1: Research

### Overview

“Golf, a cross-country game in which a player strikes a small ball with various clubs from a series of starting points (teeing grounds) into a series of holes on a course. The player who holes his ball in the fewest strokes wins. The origins of the game are difficult to ascertain, although evidence now suggests that early forms of golf were played in the Netherlands first and then in Scotland.” – Britannica.com/sports

### Rules

Golf is a single sport played on a set course with, usually, either 9 or 18 holes. A player hits the golf ball from a teeing area to start the game. From there, every shot is counted until the player hits the ball into the designated hole. Every hole on a course has a par number, or the number of shots it is expected the player should have used to hit the ball into the hole. For every shot that the player hits over par, their score goes up. The score goes down by the number of shots the player used to hit the ball into the hole below par. The player with the lowest score when the end of the course is reached wins.

### Terminology

#### Birdie

A player makes a birdie if they have used one fewer shots than the par for the hole.

#### Eagle

A player makes an eagle if they have a score of two under par for the hole.

#### Bogey

A player makes a bogey if they have used on more shots than the par for the hole.

#### Drive

A drive is a long-distance shot played from the tee box. When a player hits a drive, it is said that they have “teed off”.

### Game Structure

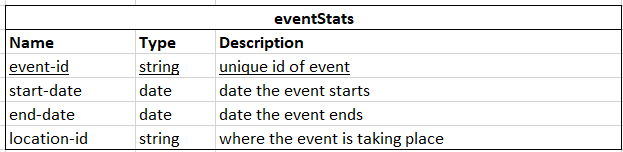
Games are played on a single course at one location. Each location has a different par number and layout. A collection of games played at different locations over a period of time is called a *tournament.* Each game has individual winners, and the overall winner of the tournament is the player with the lowest score after the collection of games has been played.

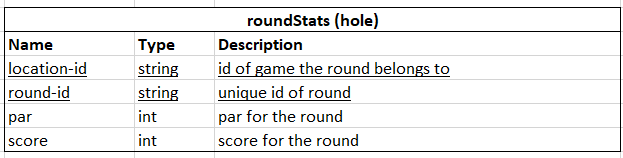
# Task 2: (E)ER Diagram

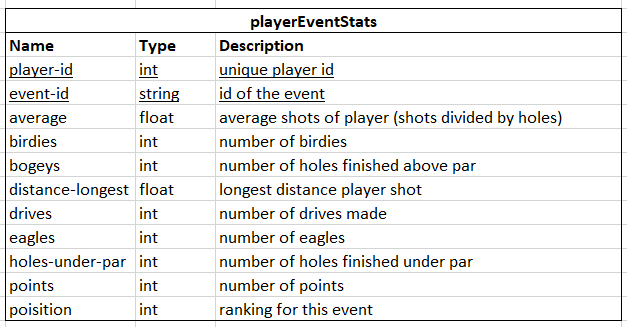
### Overview and Descriptions

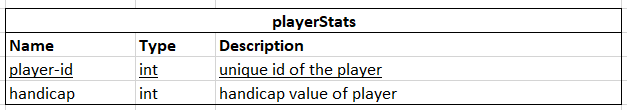
These tables contain the original ideas for what would be needed, in order to provide a sense of direction before beginning the final ER diagram. They were continuously updated as new requirements were discovered and used as the base for the ER diagram.

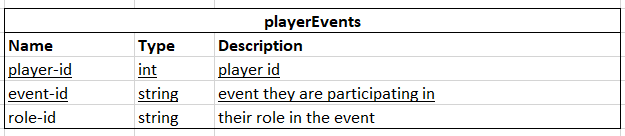


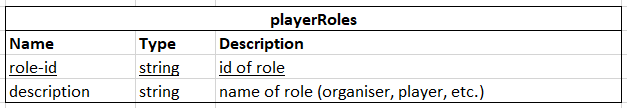


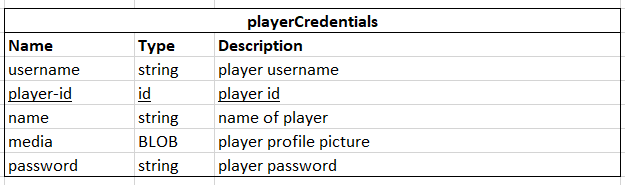


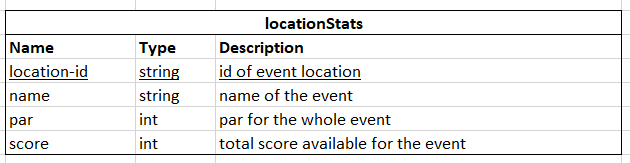






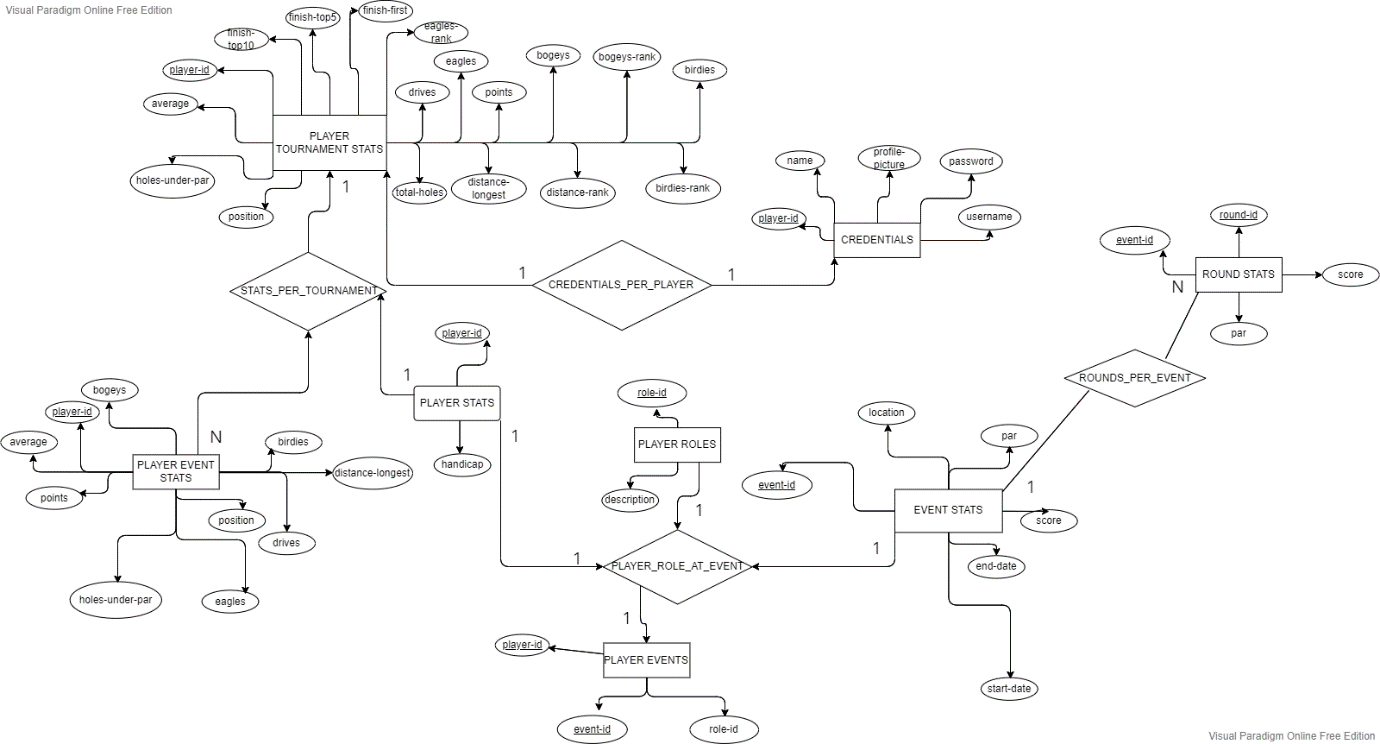






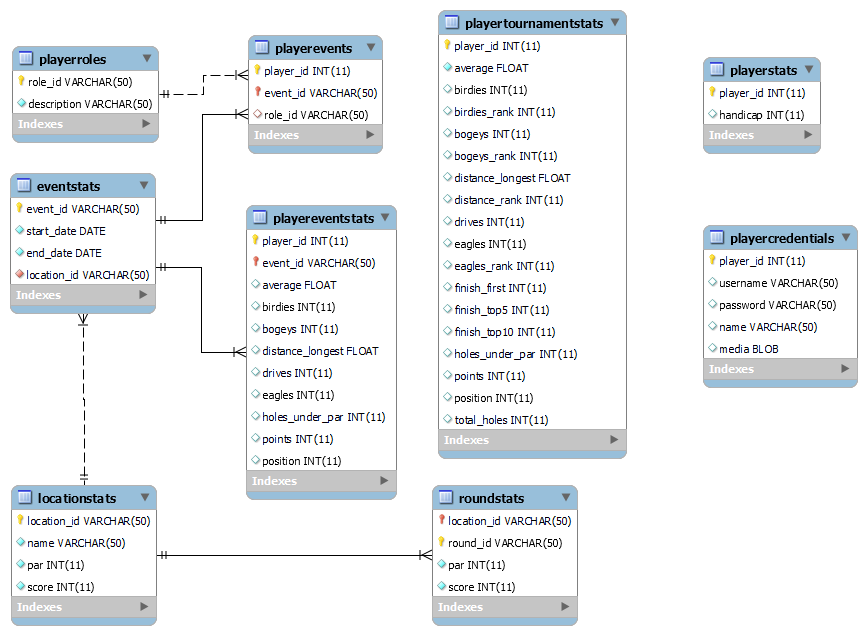
### Version 1

Below is the first draft of the ER diagram, drawn based on the above tables and later expanded upon.



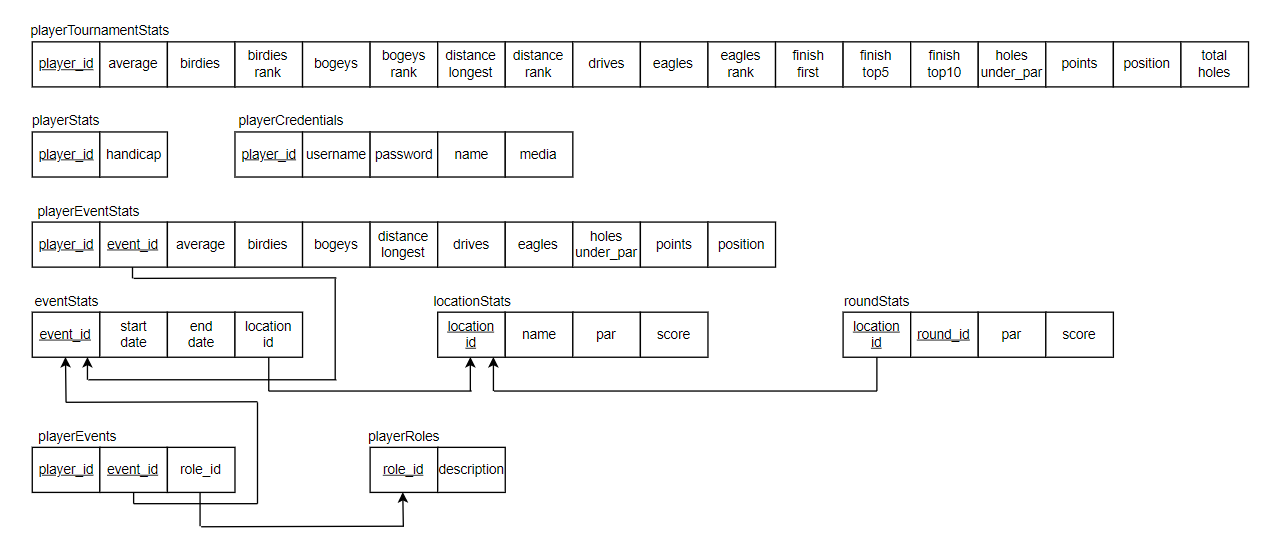
### Final Version

The final version of the ER diagram was generated using MySQL Workbench after the database had been created and contains all elements and relationships we created for the database extension.



* player\_id is linked to the persons(id) field as a foreign key.

# Task 3: (E)ER Diagram to Relational Mapping



# 

# Task 4: Relational Exclusion

### 

### SQL Statements for Extension Creation

create table locationStats

(location\_id varchar(50) not null primary key,

name varchar(50) not null,

par int not null,

score int not null);

create table playerRoles

(role\_id varchar(50) not null primary key,

description varchar(50) not null);

create table eventStats

(event\_id varchar(50) not null primary key,

start\_date date not null,

end\_date date not null,

location\_id varchar(50) not null,

foreign key (location\_id) references locationStats(location\_id));

create table roundStats

(location\_id varchar(50) not null,

round\_id varchar(50) not null, par int not null,

score int not null, primary key (location\_id, round\_id),

foreign key (location\_id) references locationStats(location\_id));

CREATE TABLE `sportsdb`.`playerstats`

(`player\_id` INT NOT NULL,

`handicap` INT NOT NULL,

PRIMARY KEY (`player\_id`),

CONSTRAINT `playerStats\_id` FOREIGN KEY (`player\_id`) REFERENCES `sportsdb`.`persons` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE=MyISAM;

create table playertournamentstats

(player\_id int not null,

average float not null,

birdies int,

birdies\_rank int,

bogeys int,

bogeys\_rank int,

distance\_longest float,

distance\_rank int,

drives int,

eagles int,

eagles\_rank int,

finish\_first int,

finish\_top5 int,

finish\_top10 int,

holes\_under\_par int,

points int,

position int,

total\_holes int,

primary key (player\_id),

foreign key (player\_id) references persons(id))

engine=MyISAM;

create table playereventstats

(player\_id int,

event\_id varchar(50),

average float,

birdies int,

bogeys int,

distance\_longest float,

drives int,

eagles int,

holes\_under\_par int,

points int,

position int,

primary key (player\_id, event\_id),

foreign key (player\_id) references persons(id),

foreign key (event\_id) references eventstats(event\_id))

engine=MyISAM;

create table playerevents

(player\_id int,

event\_id varchar(50),

role\_id varchar(50),

primary key (player\_id, event\_id),

foreign key (player\_id) references persons(id),

foreign key (event\_id) references eventstats(event\_id),

foreign key (role\_id) references playerroles(role\_id))

engine=MyISAM;

create table playercredentials

(player\_id int,

username varchar(50),

password varchar(50),

name varchar(50),

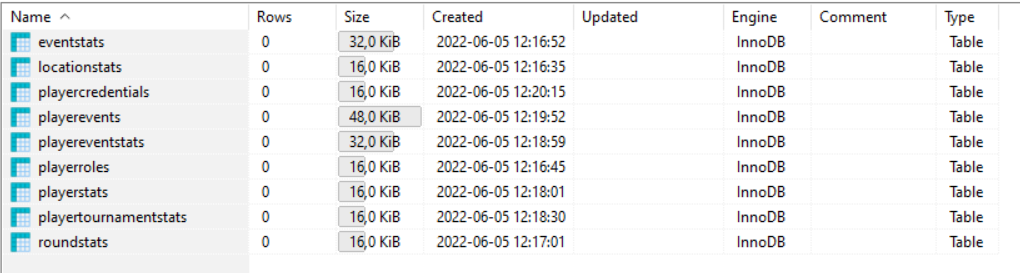
media blob,

primary key (player\_id),

foreign key (player\_id) references persons(id))

engine=MyISAM;

### Visual Database Extension



# Task 6: Sample Data

The sample data was inputted by hand to ensure the accuracy of the entered data, and to familiarise the group with the layout of the overall database. This was extremely helpful when testing, as we had a solid knowledge of the data we were working with.

# Task 7: Analyse and Optimise

### Query 1

The original query execution plan is made up of two statements:

"select role\_id from playerevents where player\_id = " . $playerID . " and event\_id = " . $eventID . ";"

and

"select description from playerroles where role\_id = " . $result . ";"

When executed in the database, the processing time for query 1 is 0

NEED MORE DATA

### Query 2

This statement is a collection of insert statements made to populate the database:

**INSERT** **INTO** eventStats (event\_id, start\_date, end\_date, location\_id) **VALUES** ('EVE001', '2022-06-06', '2022-06-06', 'LOC001');

**INSERT** **INTO** eventStats (event\_id, start\_date, end\_date, location\_id) **VALUES** ('EVE002', '2022-06-09', '2022-06-09', 'LOC002');

**INSERT** **INTO** eventStats (event\_id, start\_date, end\_date, location\_id) **VALUES** ('EVE003', '2022-06-13', '2022-06-13', 'LOC003');

**INSERT** **INTO** eventStats (event\_id, start\_date, end\_date, location\_id) **VALUES** ('EVE004', '2022-06-16', '2022-06-16', 'LOC004');

**INSERT** **INTO** eventStats (event\_id, start\_date, end\_date, location\_id) **VALUES** ('EVE005', '2022-06-20', '2022-06-20', 'LOC001');

These statements take 0.031 seconds to execute.

Grouping this collection into a single SQL statement time is much more efficient, so it was changed to the following:

**INSERT** **INTO** eventStats **VALUES**

('EVE001', '2022-06-06', '2022-06-06', 'LOC001'),

('EVE002', '2022-06-09', '2022-06-09', 'LOC002'),

('EVE003', '2022-06-13', '2022-06-13', 'LOC003'),

('EVE004', '2022-06-16', '2022-06-16', 'LOC004'),

('EVE005', '2022-06-20', '2022-06-20', 'LOC001');

This query executed exceptionally quickly, with a duration of 0.000 seconds according to HeidiSQL. This is very likely because only one statement is being executed in the population process instead of 5 separate statements.

### Query 3