# Practical 5 Documentation

Swing Steezy

June 2022

## Contents

## 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.

| playerTournamentStats | | |
|---|---|---|
| **Name** | **Type** | **Description** |
| player-id | int | unique player id |
| average | float | average shots of player (shots divided by holes) |
| birdies | int | number of birdies |
| birdies-rank | int | how player ranks in tournament regarding birdies |
| bogeys | int | number of holes finished above par |
| bogeys-rank | int | how player ranks in tournament regarding bogeys |
| distance-longest | float | longest distance player shot |
| distance-rank | int | rank when comparing longest distances |
| drives | int | number of drives made |
| eagles | int | number of eagles |
| eagles-rank | int | rank when comparing number of eagles |
| finish-first | int | number of times player finished first |
| finish-top5 | int | number of times player finished in the top 5 |
| finish-top10 | int | number of times player finished in the top 10 |
| holes-under-par | int | number of holes finished under par |
| points | int | number of points |
| poisition | int | overall tournament ranking |
| total-holes | int | number of holes played |

| eventStats | | |
|---|---|---|
| **Name** | **Type** | **Description** |
| event-id | string | unique id of event |
| start-date | date | date the event starts |
| end-date | date | date the event ends |
| location-id | string | where the event is taking place |

| roundStats (hole) | | |
|---|---|---|
| **Name** | **Type** | **Description** |
| location-id | string | id of game the round belongs to |
| round-id | string | unique id of round |
| par | int | par for the round |
| score | int | score for the round |

| playerEventStats | | |
|---|---|---|
| **Name** | **Type** | **Description** |
| player-id | int | unique player id |
| event-id | string | id of the event |
| average | float | average shots of player (shots divided by holes) |
| birdies | int | number of birdies |
| bogeys | int | number of holes finished above par |
| distance-longest | float | longest distance player shot |
| drives | int | number of drives made |
| eagles | int | number of eagles |
| holes-under-par | int | number of holes finished under par |
| points | int | number of points |
| poisition | int | ranking for this event |

| playerStats | | |
|---|---|---|
| **Name** | **Type** | **Description** |
| player-id | int | unique id of the player |
| handicap | int | handicap value of player |

| playerEvents | | |
|---|---|---|
| **Name** | **Type** | **Description** |
| player-id | int | player id |
| event-id | string | event they are participating in |
| role-id | string | their role in the event |

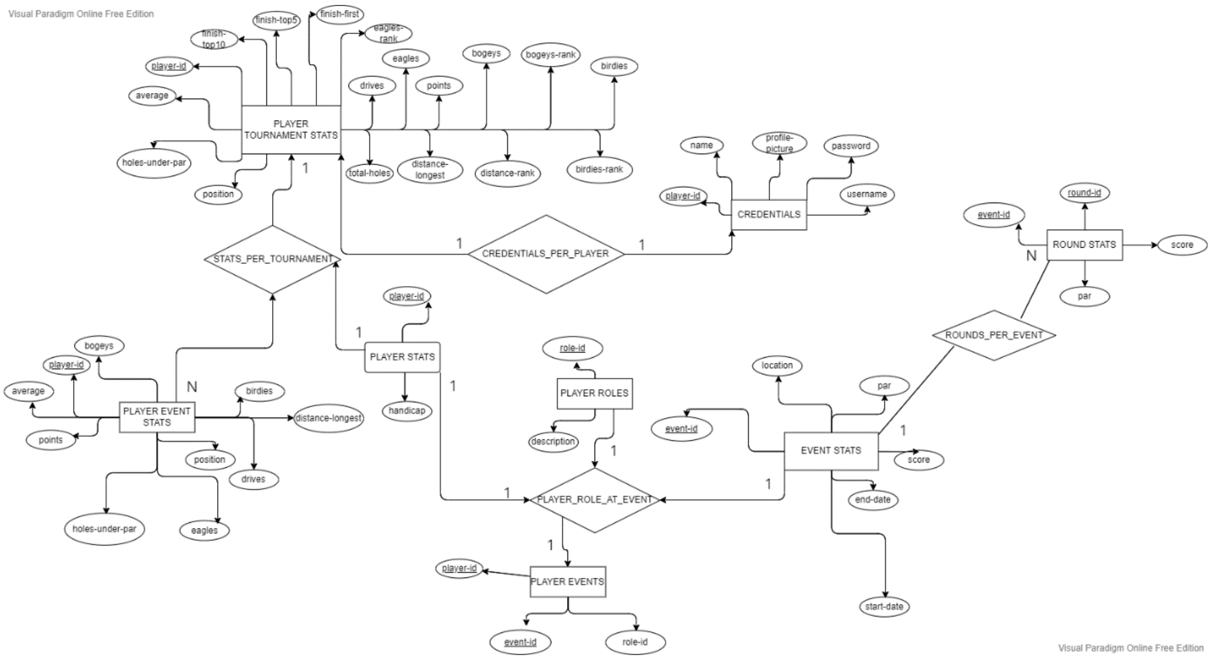
| playerRoles | | |
|---|---|---|
| **Name** | **Type** | **Description** |
| role-id | string | id of role |
| description | string | name of role (organiser, player, etc.) |

| playerCredentials | | |
|---|---|---|
| **Name** | **Type** | **Description** |
| username | string | player username |
| player-id | id | player id |
| name | string | name of player |
| media | BLOB | player profile picture |
| password | string | player password |

| locationStats | | |
|---|---|---|
| **Name** | **Type** | **Description** |
| location-id | string | id of event location |
| name | string | name of the event |
| par | int | par for the whole event |
| score | int | total score available for the event |

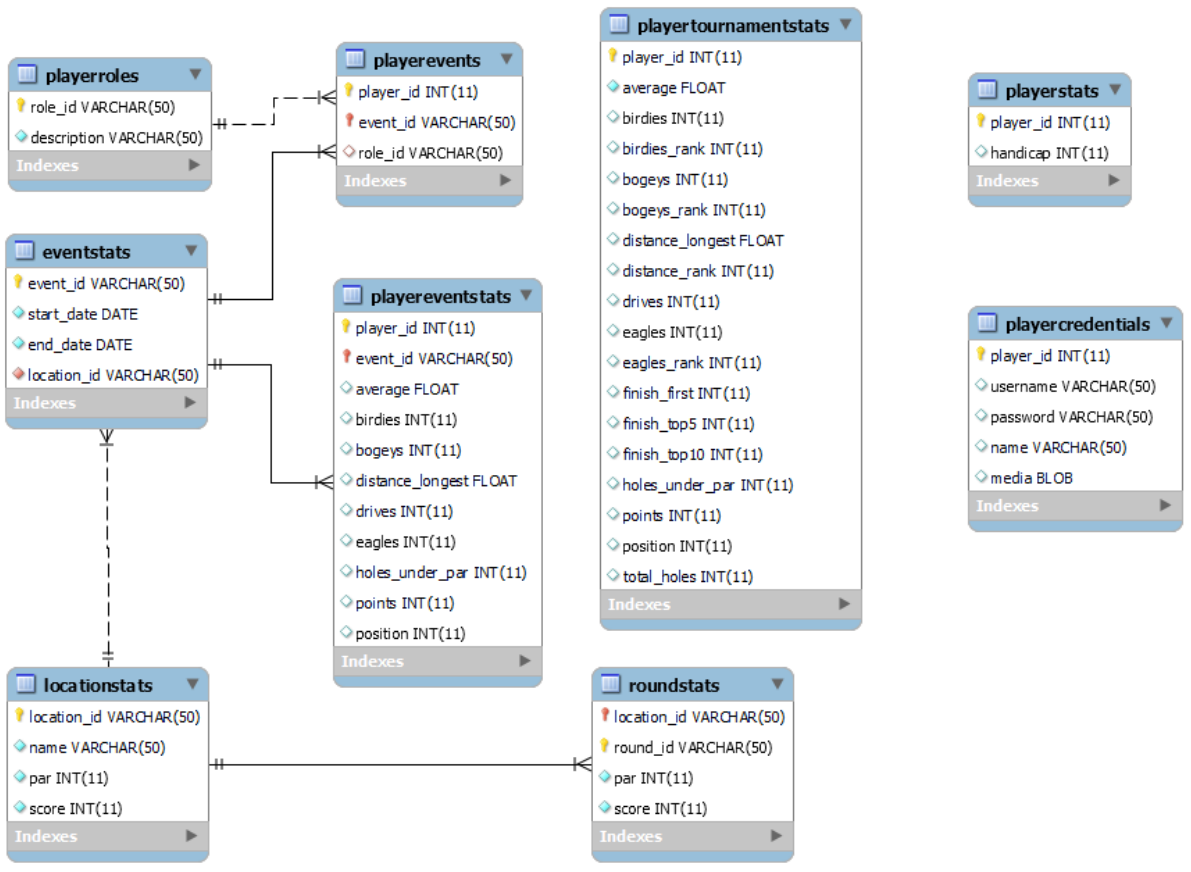
## Version 1

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



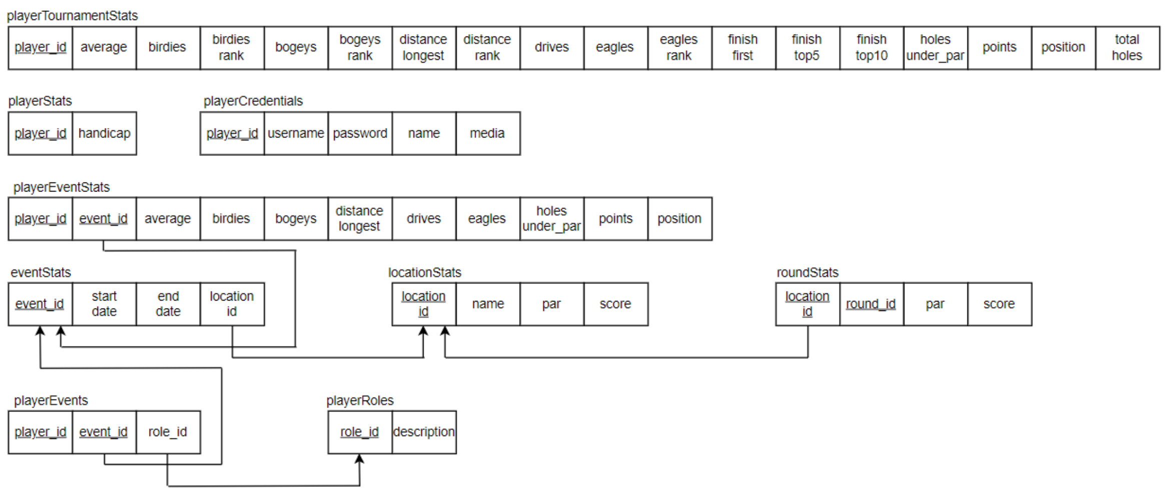
## Final Version

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

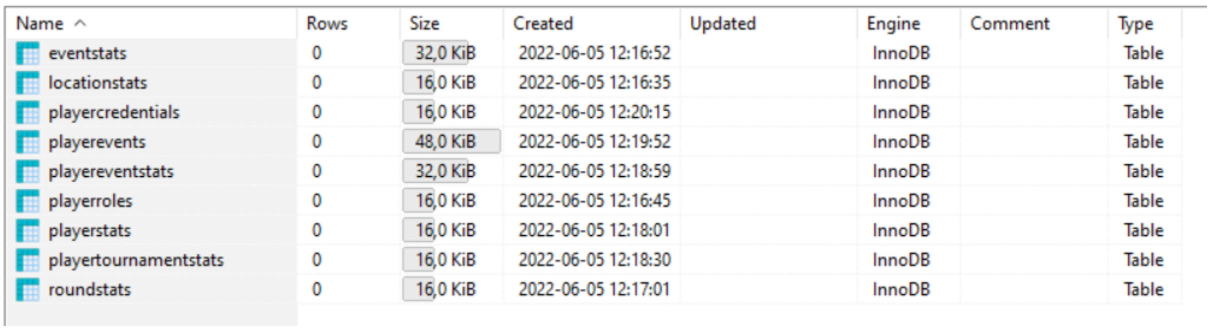
```
create table locationStats
(location_id varchar(50) not null primary key,
name varchar(50) not null,
par int not null,
score int not null);

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



| Name | Rows | Size | Created | Updated | Engine | Comment | Type |
|---|---|---|---|---|---|---|---|
| eventstats | 0 | 32,0 KiB | 2022-06-05 12:16:52 | | InnoDB | | Table |
| locationstats | 0 | 16,0 KiB | 2022-06-05 12:16:35 | | InnoDB | | Table |
| playercredentials | 0 | 16,0 KiB | 2022-06-05 12:20:15 | | InnoDB | | Table |
| playerevents | 0 | 48,0 KiB | 2022-06-05 12:19:52 | | InnoDB | | Table |
| playereventstats | 0 | 32,0 KiB | 2022-06-05 12:18:59 | | InnoDB | | Table |
| playerroles | 0 | 16,0 KiB | 2022-06-05 12:16:45 | | InnoDB | | Table |
| playerstats | 0 | 16,0 KiB | 2022-06-05 12:18:01 | | InnoDB | | Table |
| playertournamentstats | 0 | 16,0 KiB | 2022-06-05 12:18:30 | | InnoDB | | Table |
| roundstats | 0 | 16,0 KiB | 2022-06-05 12:17:01 | | InnoDB | | Table |

## 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 0.000 seconds, and the second is 0.002 seconds, for a combines execution time of 0.002 seconds.

To optimise this query set, an inner join was used to reduce to combine them into a single SQL statement:

```sql
SELECT description
FROM playerroles
INNER JOIN playerevents ON playerroles.role_id = playerevents.role_id
WHERE player_id = 3949 AND playerevents.event_id = "EVE001";
```

The resulting processing time is 0.015 seconds. This is a noticeable increase in execution time when compared to the first query set. This is likely due to each of the two original queries only having to search through one database, speeding up the search time. The inner join added time to connect the two tables, then to search both, which caused it to take longer.

### Query 2

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

```sql
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:

```sql
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

The final original query is as follows:

```sql
SELECT p.player_id, c.name, c.pic_is_set FROM playerEvents AS p NATURAL
JOIN playercredentials AS c WHERE p.event_id = "EVE001";
```

When executed, it takes 0.015 seconds to produce results. The query was slightly modified:

```sql
SELECT playerEvents.player_id, playercredentials.name,
playercredentials.pic_is_set FROM playerEvents NATURAL JOIN
playercredentials WHERE playerEvents.event_id = "EVE001";
```

Already, the execution time is down to 0.000 seconds. This is likely because of the time it takes to rename the tables within the statement. Having the select statement before the natural join in the query means that the statement is already correct in terms of heuristic optimisation, leading to faster execution time due to the reduced size of the files involved in the join.