

DATABASE DELIVERABLES

EGGS AND BACON BAY ARCHERY CLUB DATABASE

FINAL REPORT BY BANH PATE CHAUD
ASSOCIATES

Submitted 26/06/2024

Team Members:

- Joshua Burns
- Anh Khoa Nguyen
- Phuong Bao Minh Nguyen
- Sukeyna Ali

Home Page	3
User Introductions/Manuals	4
Joshua Burns	5
Melvin	6
Anh Khoa Nguyen (Ryan) user manual	7
Sukeyna Ali	8
Work Agreement	9
Roles and responsibilities	12
Project plan	14
Risk Assessment Matrix	17
Recorder Persona	19
Archer Persona	21
Empathy Map	23
Product Requirements Unique	25
Technical Details	29
ERD Diagram Draft (Out of date)	30
Final ERD for E&BBAC Database	33
Database Queries	35
CREATE DATABASE Individual Drafts	36
Melvin and Khoa Draft	37
Sukeyna Individual Draft	39
Josh Individual Draft	44
CREATE DATABASE Code	48
Performance Indexes Implementation	52
Database Interaction Queries	54
SQL commands: Sukeyna	55
SQL Commands by Melvin	57
SQL command by Anh Khoa Nguyen	61
SQL Commands: Joshua	63
Major Specific Work	70
Melvin's Major Specific Technology (Data Science)	71
Sukeyna's Major Specific Work (Data Science)	78
Anh Khoa Nguyen's Major Specific Work	81
Joshua's Major Specific Work	86
Meeting Notes	88
2024-02-28 Workshop Notes	90
2024-03-05 Meeting notes	91
2024-03-12 Meeting notes	92
2024-03-19 Meeting notes	93
2024-03-26 Meeting notes	94
2024-04-09 Meeting notes	95
2024-04-16 Meeting notes	96
2024-04-23 Meeting notes	98
2024-04-30 Meeting notes	99
2024-05-07 Meeting notes	100

2024-05-14 Meeting notes.....	101
2024-05-21 Meeting notes.....	102
Team Health Monitor.....	103
4Ls Retrospective.....	106

Home Page

Welcome to Banh Paté Chaud Associates

Banh Paté Chaud's goal in this project is to assist the Eggs and Bacon Archery club to streamline their scoring process with a custom-made database to help their club.

About Banh Paté Chaud Associates

Our team is named after a Vietnamese pie made of puff pastry and hot meats. This is because we aim to make any process we take on with our customers a comfortable enjoyable experience.

We are a small team comprised of 4 members each having a specialised set of skills to ensure a quality product is produced, have a look at our [User Introductions/Manuals](#) to meet the team and read through everything in our plan to see our strategy for tackling your specific problem.

Meet the team

 JB

@Joshua Burns
Team Leader

 SA

@SUKEYNA ALI
Technical Writer

 AN

@Anh Khoa Nguyen
Stress Tester

 BP

@Bảo Minh Nguyễn Phương
Database Developer

Resources

Restrict search to this space's space key.

Our calendar

Submit Project Proposal

Mar 24, 2024

By the 24th of March we will have submitted a finished Project Proposal to Eggs and Bacon Bay Archery Club

Submit Progress Plan

Apr 14, 2024

By the 14th of April we will have created a draft of the database. Filled it with data and will be beginning to draft our Queries. At this time we will submit a Progress Plan in order to keep our stakeholders updated and in the loop on our progress plan. At this point we may discuss extensions and any budget concerns that have arisen.

Submit Finished Project

May 19, 2024

On May 19th we aim to present and submit our finished database to the Eggs and Bacon Bay Archery Club. Our agreed on delivery date is May 26th however we believe by submitting a week early we allow for a number of factors including workflow problems, budget issues etc. It will also provide positive engagement with the customer.

User Introductions/Manuals

See each of our introductions to learn a little about who you are working with.

[✉ Joshua Burns](#)

[✉ Sukeyna Ali](#)

[✉ Melvin](#)

[✉ Anh Khoa Nguyen \(Ryan\) user manual](#)

Joshua Burns

Environments I like to work in	<ul style="list-style-type: none">• Communicative environments• Fun environments• Stressful environments
Preferred working hours	I prefer night rather than early morning
Communication preferences	Discord or text
Preferred ways to receive feedback	Direct feedback, telling me what the issue is and offering advice if you have any on how to resolve it
Things I need	Open communication, including how you are doing on tasks, if you need any assistance, if you are struggling with a specific task.
How I learn best	Rapid paced environments, with the ability to ask questions.
Things I struggle with	Focusing for extended periods of time, rigidly remaining on task
Things I love	Good food and the outdoors.
If I were an animated gif/meme/animal/song, I would be...	Tulio from The Road To El Dorado
My favorite saying	"Go with the flow, you're going that direction anyway"
Other things I want you to know about me	I have my Advanced Scuba Diving Certificates

 For a facilitation guide and more info on running this play with your team, visit <https://www.atlassian.com/team-playbook/plays/my-user-manual>

Melvin

Environments I like to work in	<ul style="list-style-type: none">Peaceful, airconditioned, home office environment
Preferred working hours	<ul style="list-style-type: none">From Thursday to Sunday, could be day or night, maximum 5-6 hours a week
Communication preferences	<ul style="list-style-type: none">Discord
Preferred ways to receive feedback	<ul style="list-style-type: none">Directly in person, discord or Jira
Things I need	<ul style="list-style-type: none">An idea that has been well brainstormed by the whole team, objectives was broken down into agile steps as clear and practical as possible.
How I learn best	<ul style="list-style-type: none">Given Instruction, explanation and time to execute
Things I struggle with	<ul style="list-style-type: none">Being a leader, solely break the project down into agile pieces.
Things I love	<ul style="list-style-type: none">A well-managed team in terms of workload, time and instructions.
If I were an animated gif/meme/animal/song, I would be...	<ul style="list-style-type: none">Maybe a funny lion meme
My favorite saying	<ul style="list-style-type: none">What are we doing guys? Is everything on track?
Other things I want you to know about me (out of project's scope)	<ul style="list-style-type: none">I am addicted to playing pool

 For a facilitation guide and more info on running this play with your team, visit <https://www.atlassian.com/team-playbook/plays/my-user-manual>

Anh Khoa Nguyen (Ryan) user manual

Environments I like to work in	any environment
Preferred working hours	5-6hrs a day
Communication preferences	Online (discord)
Preferred ways to receive feedback	Straightforward feedback with some solutions included
Things I need	Nothing rn
How I learn best	Self-study
Things I struggle with	Communication
Things I love	Football, games, music
If I were an animated gif/meme/animal/song, I would be...	I would be a corgi
My favorite saying	I am > I were
Other things I want you to know about me	Nothing

-  For a facilitation guide and more info on running this play with your team, visit <https://www.atlassian.com/team-playbook/plays/my-user-manual>

Sukeyna Ali

Environments I like to work in	<ul style="list-style-type: none">• Collaborative• Flexible• Constructive• Fun
Preferred working hours	<ul style="list-style-type: none">• No Thursdays and Fridays
Communication preferences	<ul style="list-style-type: none">• Discord, Email
Preferred ways to receive feedback	<ul style="list-style-type: none">• In a constructive way• Offer other suggestions or advise
Things I need	<ul style="list-style-type: none">• To not fail the class
How I learn best	<ul style="list-style-type: none">• Through examples
Things I struggle with	<ul style="list-style-type: none">• Remembering details
Things I love	<ul style="list-style-type: none">• Movies, Sudoku
My favourite saying	<ul style="list-style-type: none">• “I guess”
Other things I want you to know about me	<ul style="list-style-type: none">• My name is pronounced “Sue- kay- na”

 For a facilitation guide and more info on running this play with your team, visit <https://www.atlassian.com/team-playbook/plays/my-user-manual>

Work Agreement

👤 Team Preferences

Team Member	Melvin Nguyen	Joshua Burns (Manager, I guess)	Sukeyna Ali	Anh Khoa Nguyen (Ryan)
Working location and timezone	Online or on campus Thursday - Sunday	Close to campus	Online On campus (Tuesday, Wednesday or Friday)	Close to campus or Online
Working hours and commitments	All day on my available day	Work schedule changes regularly. Free Mon-Wednesday	Free on Tuesday, Wednesday, Friday	Free everyday except weekend
Working environment and preferences	Home office or meet up somewhere	WFH or on campus	Home office or on campus	Campus or home office
How I like receiving feedback	Directly into my face	Directly, tell me what the issue is and what your advice is if you have any	Directly, in person	Directly
Context about me	Second semester of my Computer Science (Data Science) Bachelors	Second semester of my Computer Science (Cyber Security) Bachelors	First year second semester of my Bachelors of Computer Science (Data Science)	Second year of Bachelor of computer science majoring soft dev

💬 Communication Channels

Channel	Purpose	Audience	Standards
Confluence	Official Documentation and Note Taking.	All	Remain formal, use best practices.
Discord	General Communications	All	Informal communications, non-vital issues
Jira	Lodge issues, assign tasks, track task progress	All	Use best practices and remain formal.

Meetings

Objective	Finalize and discuss Project Proposal.	Discuss Normalisation and Database Design	Draft Queries	Finalise Database
Outcomes	Arrive at a final point with the project brief. Discussing and agreeing amongst the group what the Project Brief should look like	Arrive at a list of data the team believes will be entered into the database as well as a draft of how we believe the database will interact	Using test-data, draft queries to navigate the database and request data that the end user wants.	After stress-testing, complete final changes and edits to the database for the final submission.
Format	Whole Team, Scrum, In Person,	Whole Team, Scrum, Demo, In Person,	Whole Team, Scrum, Hybrid,	Whole Team, Scrum, In Person, Demo,
Who	Whole Team	Whole Team	Whole Team	Whole Team
Resources	Laptops	draw.io, FeeNIX Maria.	Laptop/Note Taking devices,	Laptop,
How will we show up?	Collaboration, respect, engaging with the topic.	Collaboration, creativity, respect, efficiency.	Respect, equity, accountability, authenticity.	Respect, open-minded, critical (of the project, not of others), collaboration
How will we manage follow up?	Record meeting notes, maintain consistent communication, Follow agreed upon processes	Record data-points on confluence as well as any queries that have been created on Git/Jira	Save queries to Git/Jira, Continue to edit queries as the Database progresses.	Compile all queries/necessary information for the database in a centralized location, ready for submission.

Escalation Process

Decider	How	Transparency
Joshua	Discuss with the group about issue, attempt to resolve issue with team.	Summarize the issue and discussion after the fact on Confluence.
Tutor	Issues that cannot be dealt with amongst the group will be raised to the tutor for adjudication and assistance.	Emails will be sent to the tutor.

Continuous Improvement

Purpose	How	Standards
Discuss Progress Weekly	TBD, either in person or discord call	All feedback will be constructive criticism so that individuals can benefit and to ensure that moral remains high.
Ensure Progress is on Track	Consistently refer to the timeline	All tasks will be completed according to the timeline decided on, if a team member gets stuck, reach out for assistance.

Roles and responsibilities

📋 Overview

Identify and discuss team responsibilities by following the instructions for the [Roles and Responsibilities Play](#).

Team	Banh Paté Chaud Associates
Team members	@SUKEYNA ALI @Bảo Minh Nguyễn Phương @Anh Khoa Nguyen @Joshua Burns
Date	12/03/24
Team mission	To provide the archery club with a database that can store archery scores and other set requirements.

📘 Roles and responsibilities

Roles	Responsibilities (what others think)	Responsibilities (what I think)	Comments
Team Leader	<ul style="list-style-type: none">• Lead workshop discussions• Delegating tasks• Resolving team issues	<ul style="list-style-type: none">• Ensure forward team movement on tasks• Ensure team feels supported• Regularly check-in with team on progress/moral	@Joshua Burns
Note Taker	<ul style="list-style-type: none">• Summarising team discussions• Be attentive during team discussions• Summarise important communications	<ul style="list-style-type: none">• Update meeting notes when necessary• Include input from all team members	Rotative: Week 1: @Joshua Burns Week 2: @SUKEYNA ALI Week 3: @Anh Khoa Nguyen Week 4: @Bảo Minh Nguyễn Phương
Quality control	<ul style="list-style-type: none">• Ensure tasks are submitted to a high standard• Spell-check/Plagiarism	<ul style="list-style-type: none">• Providing feedback on improvements and changes that need to be made	@Joshua Burns, @SUKEYNA ALI, @Bảo Minh Nguyễn

	Check work		Phương , @Anh Khoa Nguyen
Database developer	<ul style="list-style-type: none"> • Creating the blueprint of the data base • Ensuring information is easily accessible and understandable 	<ul style="list-style-type: none"> • Ensuring that the data base is regularly backed up • Noting down the way the system functions 	@Bảo Minh Nguyễn Phương
Requirement Gatherer	<ul style="list-style-type: none"> • Understanding the needs of the project • Explaining the expectations that need to be met 	<ul style="list-style-type: none"> • Informing the team on any updates during the project • Documenting the specific functions that the data base need to include • Communicating with the data base developer and stakeholders 	@Joshua Burns
Technical Writer	<ul style="list-style-type: none"> • Documenting for developers on how the database works • Creating guides for potential users of the data base 	<ul style="list-style-type: none"> • Creating troubleshooting guides • Producing release notes 	@SUKEYNA ALI
Stress Tester	<ul style="list-style-type: none"> • Analysing the performance of the data base • Finding weak points in the data base • Ensuring the data base can be recovered in case of failures 	<ul style="list-style-type: none"> • Communicating the performance of the data base and what changed may need to occur • Enhancing the performance and efficiency of the data base • Stress testing the data base 	@Anh Khoa Nguyen

Project plan

Driver	@Joshua Burns
Approver	Lecturer
Contributors	@Anh Khoa Nguyen @SUKEYNA ALI @Bảo Minh Nguyễn Phương
Informed	Eggs and Bacon Bay Archery Club.
Objective	Create a database to store archery scores according to requirements provided by the archery club.
Due date	May 19, 2024
Key outcomes	Database is able to store the required data and provide user interactable data access according to specifications provided by the archery club.
Status	IN PROGRESS

🤔 Problem Statement

The local club wishes to be able to easily input, edit and access their scores. We think that by having a number of tables within the database it will make the database simpler and more versatile to understand and access.

🎯 Scope

Archer

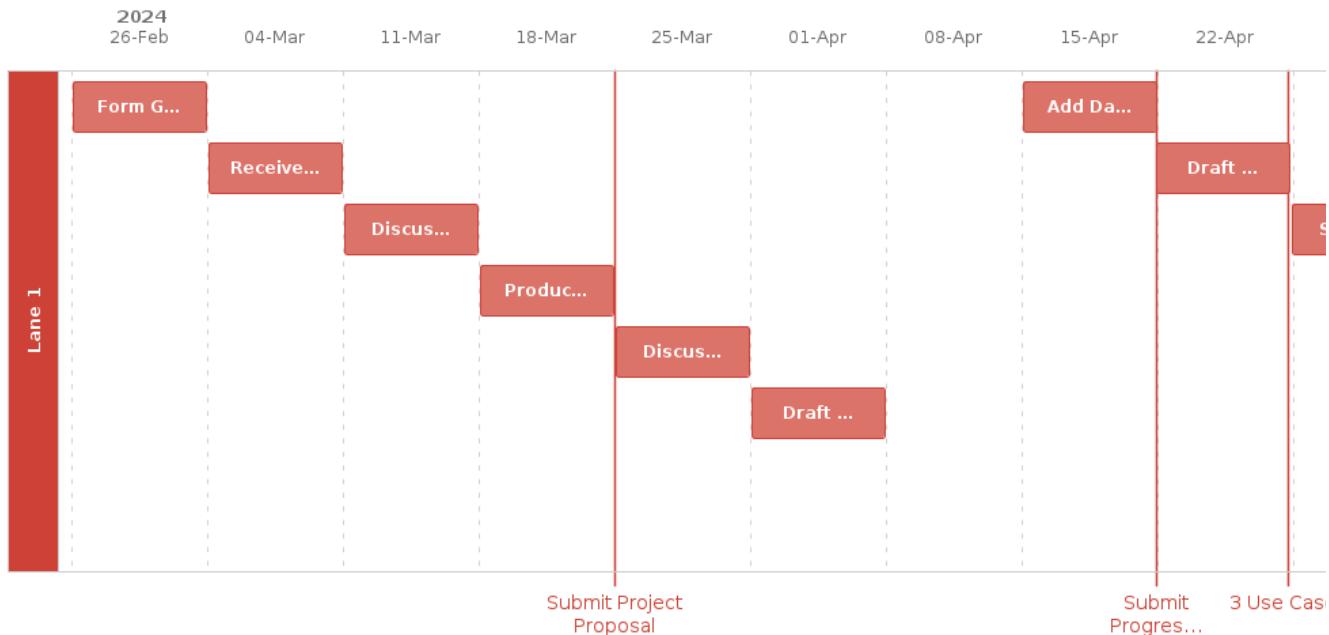
Must have:	<ul style="list-style-type: none">Ability to lookup scores (Scores should be linked to date and round type)Ability to record Scores (Able to add a Date and Time, Round, Equipment (1 of 6),Ability to look up what a round is (Distances, no. of ends, target faces, number of ends total)Ability to search competitions (Name, Score, rounds equipment)Add filterable by related to club championshipArchers best score for a round.Clubs best score and who shot it.
Nice to have:	<ul style="list-style-type: none">Add club championship TableAllow only the user and recorders to be able to access a users history.
Not in scope:	<ul style="list-style-type: none">Does not need to be able to add new, Equipment types, Rounds, Divisions, Competitions or Championships.

Recorder

Must have:	<ul style="list-style-type: none">Add new Archers
-------------------	---

	<ul style="list-style-type: none"> • Add new Round • Add new Competition • Approve Archer Scores • Scores have to be arrow by arrow (Sorted highest to lowest in ends), and must be grouped by ends. [Array?] • Able to link to competition, able to link comp to champ • Division (age, gender) • Default Equipment • Time dependent rounds. (Valid Invalid rounds)
Nice to have:	<ul style="list-style-type: none"> • Archer details (Join date etc.) • Autofill Round Specifications • Link Competition to date • When round becomes invalid, shift to historic. • Separate DB for Comps and Championships.
Not in scope:	<ul style="list-style-type: none"> • Does not need to be able to enter brand new scores • Don't need to be able to create new: Divisions, Equipment or Competitions

📅 Timeline



🚩 Milestones and deadlines

Milestone	Deadline	Status
Submit Project Proposal	Mar 24, 2024	In Progress
Begin Draft of DB Design	Apr 7, 2024	To Do
Implement Draft	Apr 14, 2024	To Do

Fill Table with data	Apr 21, 2024	To Do
Draft Queries	Apr 28, 2024	To Do
Stress Test and Fix Errors	May 12, 2024	To Do
Submit Finished DB	May 19, 2024	To Do

🔗 Reference materials

Risk Assessment Matrix

 This template is brought to you by Stiltsoft, an Atlassian platinum partner.

Background

The eggs and bacon bay archery club has reached out to have our team design a database for them in order to fulfil the list of specifications requested.

Risks management

Identify and prioritize risks based on their probability and severity. Then define what further actions you need to take to control the risks, and who needs to carry out these actions.

Risk rating

LOW	MEDIUM	HIGH	EXTREME
<ul style="list-style-type: none">AcceptableOk to proceed	<ul style="list-style-type: none">As low as reasonably practicableTake mitigation efforts	<ul style="list-style-type: none">Generally unacceptableSeek support	<ul style="list-style-type: none">IntolerablePlace event on hold

LIKELIHOOD	SEVERITY			
	ACCEPTABLE <i>Little to no effect on event</i>	TOLERABLE <i>Effects are felt, but not critical to outcome</i>	UNDESIRABLE <i>Serious impact to the course of action and outcome</i>	INTOLERABLE <i>Could result in disaster</i>
IMPROBABLE <i>Risk is unlikely to occur</i>		Group member drops out.	Database progress lost. Database becomes inaccessible. Overrun given budget. Budget reduced.	Loss of all work. Don't have deliverable product by due date.
POSSIBLE <i>Risk will likely occur</i>		Roadblock of Database Understanding.	Group member not completing assigned weekly tasks.	

		Misunderstand specifications/instructions.		
PROBABLE <i>Risk will occur</i>	Showing up late to meetings.	Scheduling Issues.	Customer requests additional features.	

Budget, Timeline, Customer not happy with product

✓ Action items

- Regular Communication between team members.
-
- Regular backups of database commands and queries to shared GitHub
- Utilize the agreed-upon escalation process when necessary
- Communicate availabilities with team.
- Regularly check task progress
- Weekly check-ins with supervisor
- Allow a 1 week buffer to ensure quality product is delivered on time.
- Regularly check budget

Recorder Persona

Persona name	Jeremy
Persona role	Recorder
Job description	Must observe and authenticate scores shot and entered by archers

🏢 Company

Company name	Egg and Bacon Bay Archery Club
Company size	80 (approx)
Industry	Sporting Club

👤 Demographic information

Age	30
Gender	Male
Income	Volunteer
Education level	N/A
Residential environment	Country Town

✍ Personal quote

"It's just a hassle when we have to jump through so many hoops to get it done"

📜 Biography

Professional goals	Motivators
<ul style="list-style-type: none">• Make archery easier and more fun for everyone• One day be a recorder for a competition	<ul style="list-style-type: none">• Has been an archer for most of his life and wants to contribute some time to the club to help everything run smoothly
Challenges	Sources of information
<ul style="list-style-type: none">• Not the most technologically savvy• Doesn't want to spend most of his time recording when he is at the range, would prefer to be shooting himself.	<ul style="list-style-type: none">• Years of experience• Club communications.

Archer Persona

Persona name	Amy
Persona role	Archer
Job description	Engaging in competitions involving the use of a bow and arrow while also recording details of personal scoring.

🏢 Company

Company name	Egg and Bacon Bay Archery Club
Company size	80 (approx)
Industry	Sporting Club

👤 Demographic information

Age	18
Gender	Female
Income	N/A
Education level	Student
Residential environment	Country Town

✍ Personal quote

"I'm spending more time entering and finding scores than shooting arrows"

📜 Biography

Professional goals	Motivators
<ul style="list-style-type: none">To become a professional archerTo enhance her archery skills by focusing on practice and consistency	<ul style="list-style-type: none">To have her own competition details easily availableWould like for the scoring entry process to be more time efficient
Challenges	Sources of information

<ul style="list-style-type: none">• Finds it time consuming to note down scores during competitions• Personal scores and club scores cannot be accessed with ease• Not skilled in using technology	<ul style="list-style-type: none">• Communication with peers in the club• The Egg and Bacon Bay Archery Club website
--	---

Empathy Map

Empathy Map Relating To Amy the Archer Persona	
Think and Feel	Say and Do
Likes to focus on the archery Wants to spend most of their time shooting Feel good when they find the right equipment for them It's fun to be in competitions Wants a simple and streamlines data entry process Wants to easily access their scores Feels good to be on the scoreboard Feel a sense of belonging when they see their scores displayed Worried their data will be lost Worried their scores will be incorrect Thinking recorder should be entering their data	"I'm spending more time entering scores than shooting arrows" "This system is just so old and clunky" "I've had my scores lost" They go shooting every week They focus more on shooting when they don't have to enter scores Get confused about the order when entering their scores Have to enter scores by hand Gets frustrated when shooting and they have to stop and enter their scores Enjoy the shooting more when they focus on entering scores less Look for their scores after they shoot to see where they place Rush the score entry so they can shoot more Get mad when they shoot bad
See Enthusiastic people around them passionate about the same sport Lots of different people shooting different types of equipment and rounds I see lots of people of different ages Data entry can be hard with bad weather The scores are very human dependent Not many safeguards That people are sick of the current system	Hear "That's a cool sport" "I've seen that at the Olympics" "That seems so complicated" "What kind of equipment do you use" "How many things have you won" "Are there archery competitions?" "Entry by paper is just old fashioned and out of date" "Anyone could change the scores when you hand it in"
Pain Doing anything that isn't shooting Entering the data Data being lost Having to look up round specifications every shoot	

Incorrect data entered by human error

Scared that other people will see their bad scores

Keeping ends in order, keeping track of distances and arrows for rounds

Gain

Easy to enter data

More time shooting, less time entering

Easy to look up definitions

More time shooting arrows

Easier to display good scores

Comparing Scores/Performances

Trying different equipment

Product Requirements Unique

Target release	20/05/2024
Epic	Type /Jira to add Jira epics and issues
Document status	DRAFT
Document owners	@Bảo Minh Nguyễn Phương @Anh Khoa Nguyen @SUKEYNA ALI @Joshua Burns
Designers	@Bảo Minh Nguyễn Phương @Anh Khoa Nguyen @SUKEYNA ALI @Joshua Burns
Tech lead	@Joshua Burns
QA	@Bảo Minh Nguyễn Phương @Anh Khoa Nguyen @SUKEYNA ALI @Joshua Burns

🎯 Objective

This Product Requirements Document aims to define the product objectives, scope, and requirements for the Eggs and Bacon Bay Archery Club project.

The initiative of the Eggs and Bacon Bay Archery Club is to create an application that will allow archers to search and store information about scores, rounds, competitions, and club activities. Record keepers can manage archers, rounds, tournaments, and scoring approvals. Arrow each arrow, and scores are recorded and grouped according to ends. Division and default equipment management are provided, along with rounds that depend on time.

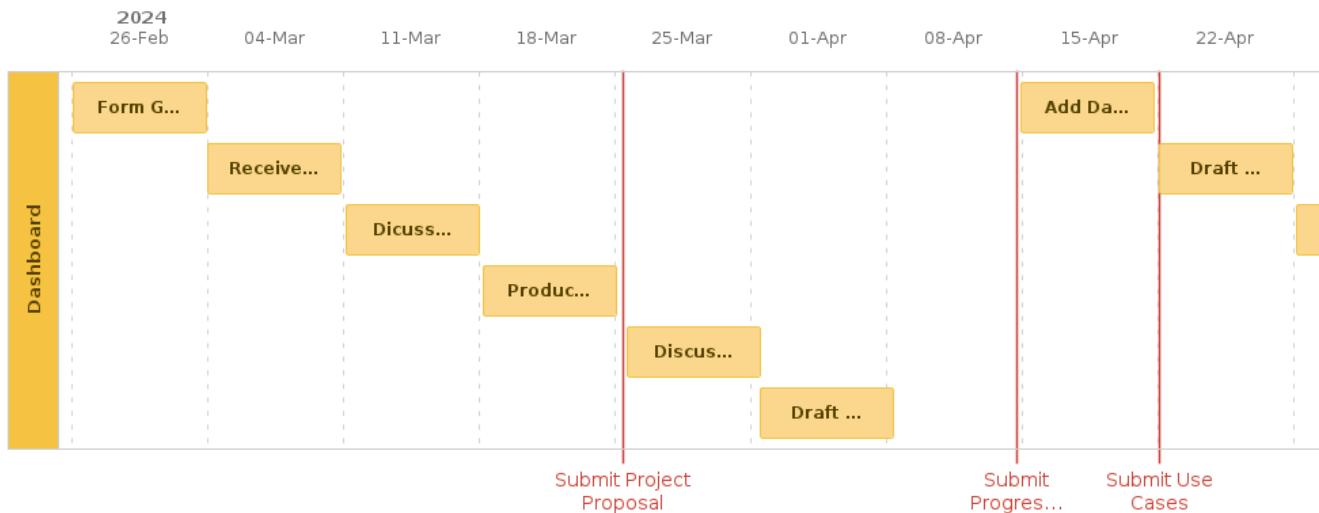
📊 Success metrics

Goal	Metric
Increasing users engagement	The frequency of archers accessing to Eggs and Bacon Bay Archery Club to look up scores, rounds and competitions skyrockets
Increasing users satisfaction	From feedback to improve ease of use and effectiveness to increase users satisfaction
Increasing Club Participation	Tracking numbers of archers that are active on club participation and compared to the numbers of archers with the application launch.

🤔 Assumptions

The local club wishes to be able to input, edit, and access their scores quickly. We think having several tables within the database will make it simpler and more versatile to understand and access.

🌟 Milestones



📋 Requirements

Requirement	User Story	Importance	Page	Type & Author
Archers should be able to lookup scores linked to date and round type.	As an archer, I want to be able to search for my scores in various rounds so I can tell if I have progressed.	HIGH		SELECT Khoa
Archers should be able to record scores with date, time, round, and equipment details	As an archer, I want to be able to save my scores with the date, time round and equipment to keep track of my record perfomance.	HIGH		INSERT Josh
Archers should be able to access information about different rounds including distances, number of ends, target faces, and total number of ends.	As an archer, I want to be able to look up for information in different rounds such as distances, number of ends, target faces to better prepare for the next competitions.	HIGH		SELECT Sukeyna
Archers should be able to search competitions by name, score, rounds, and equipments	As an archer, I want to be able to search competitions by name, score, rounds and equipment to relevant competitions that I want to join.	HIGH		SELECT Melvin
Archers should be able to filter competitions related to club championships	As an archer, I want to be able to filter competitions to my desired club championships so I can focus on events that	HIGH		SELECT Khoa

	increase my club's performance.			
Archers should be able to view their best scores for each round.	As an archer, I want to be able to view my best scores for each round to find strength and area needed to improve.	HIGH		SELECT Melvin
Archers should be able to view the club's best score and the archer who achieved it	As an archer, I want to be able to view the clubs' best scores and the archer who do it so I can approximate the club's overall performance.	HIGH		SELECT Sukeyna
Recorders should be able to add new archers.	As a recorder, I want to be able to add new archers to the database so I can ensure the club's roster information stay up-to-date.	HIGH		Insert Khoa
Recorders should be able to add new rounds	As a recorder, I want to be able to add new rounds' details such as distances and target faces so the competition has accurate information.	HIGH		Insert Sukeyna
Recorders should be able to add new competitions	As a recorder, I want to be able to add new competitions to the database with details such as name and date to improve event management.	HIGH		Insert Melvin
Recorders should be able to approve archer scores	As a recorder, I want to be able to approve archer scores to ensure archer's data stay up-to-date.	HIGH		SELECT, Insert and a Delete Josh
Recorders should be able to record scores arrow by arrow, grouped by ends.	As a recorder, I want to be able to record scores arrow by arrow, grouped by ends to keep a detailed record of archers' achievement.	HIGH		Insert Josh
Recorders should be able to link scores to competitions and championships	As a recorder, I want to be able to link scores to competitions and champions to keep track performance in particular events.	HIGH		UPDATE/IN SERT Sukeyna
Recorders should be able to manage default	As a recorder, I want to be able to manage default	HIGH		UPDATE Melvin

equipment.	equipment to streamline the scoring process.			
Recorders should be able to manage time-dependent rounds.	As a recorder, I want to manage time-dependent rounds to keep only valid scores recorded.	HIGH		UPDATE Khoa
Archers should be able to view equivalent rounds.	As a recorder, I want to be able to see what round are equivalent for me personally at a competition.	HIGH		SELECT Sukeyna

⚠️ Out of Scope

Archer:

- Nice to have:
 - Add club championship Table
 - Allow only the user and recorders to access a user's history.
- Not in scope:
 - Add new Equipment types, rounds, divisions, competitions or championships

Recorder

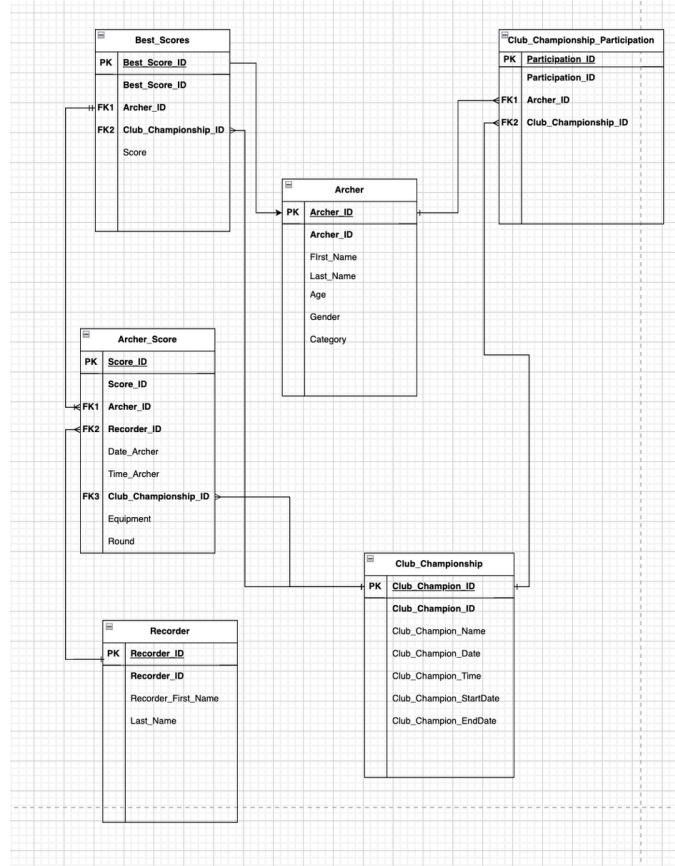
- Nice to have:
 - Archer details (Join date etc.)
 - Autofill Round Specifications
 - Link Competition to date
 - When the round becomes invalid, shift to historic.
 - Separate DB for Competitions and Championships
- Not in scope:
 - Enter brand-new scores
 - Create new elements such as Divisions, Equipment, and Competitions.

Technical Details

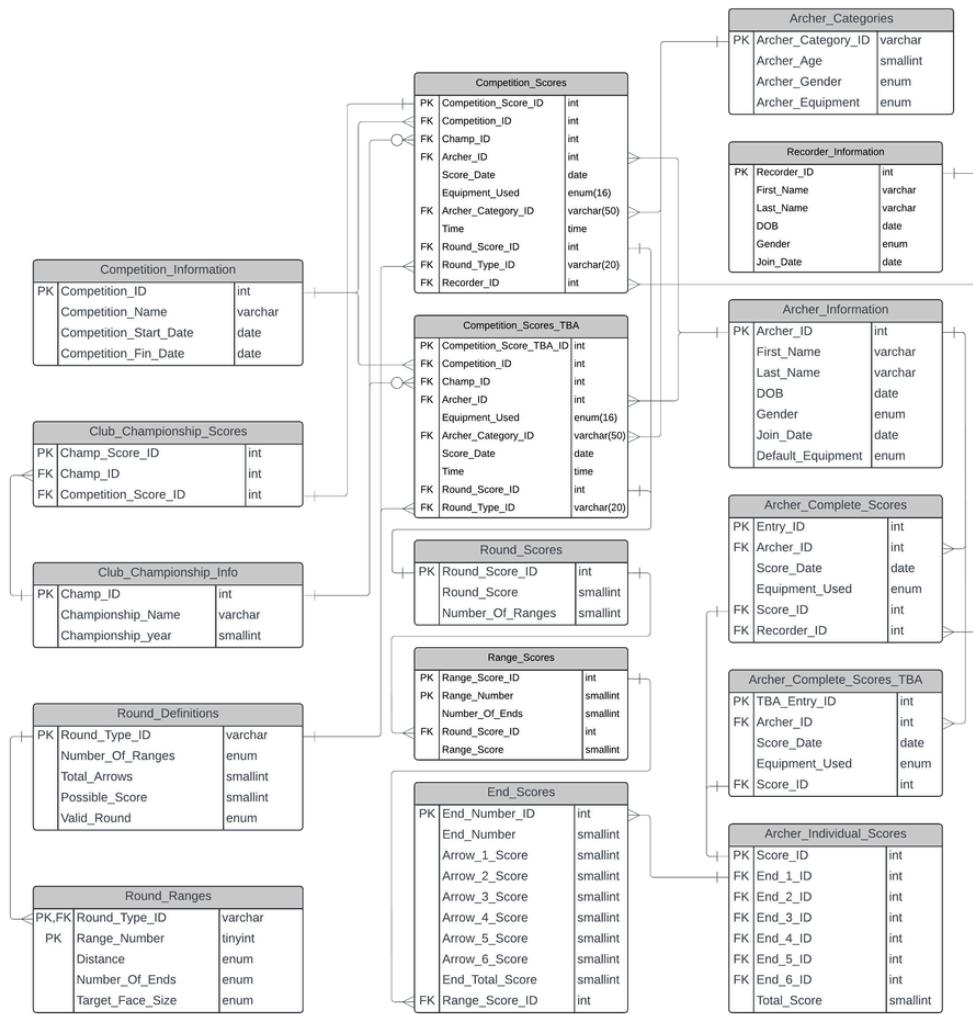
ERD Diagram Draft (Out of date)

See [Final ERD for E&BBAC Database](#) for the final ERD

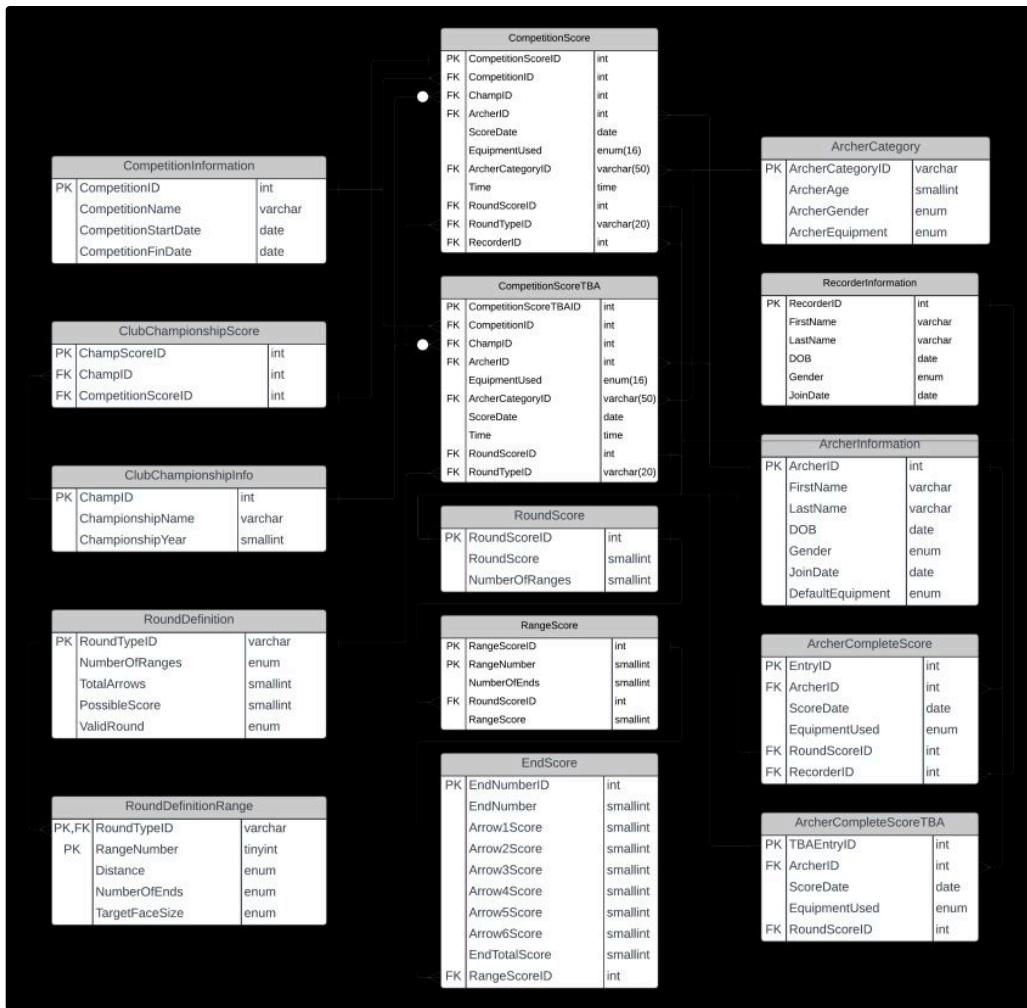
Our first ERD had a very limited understanding of the requirements and while it did help provide a skeleton and an idea of how the database interacts on a more granular level, it was iterated on multiple times.



This iteration of the Database ERD met most of the requirements however did not follow correct naming scheme, it contained both underscores and plurals. It was also overcomplicated and included the unnecessary tables 'Archer_Individual_Scores' and 'Archer_Complete_Scores_TBA'



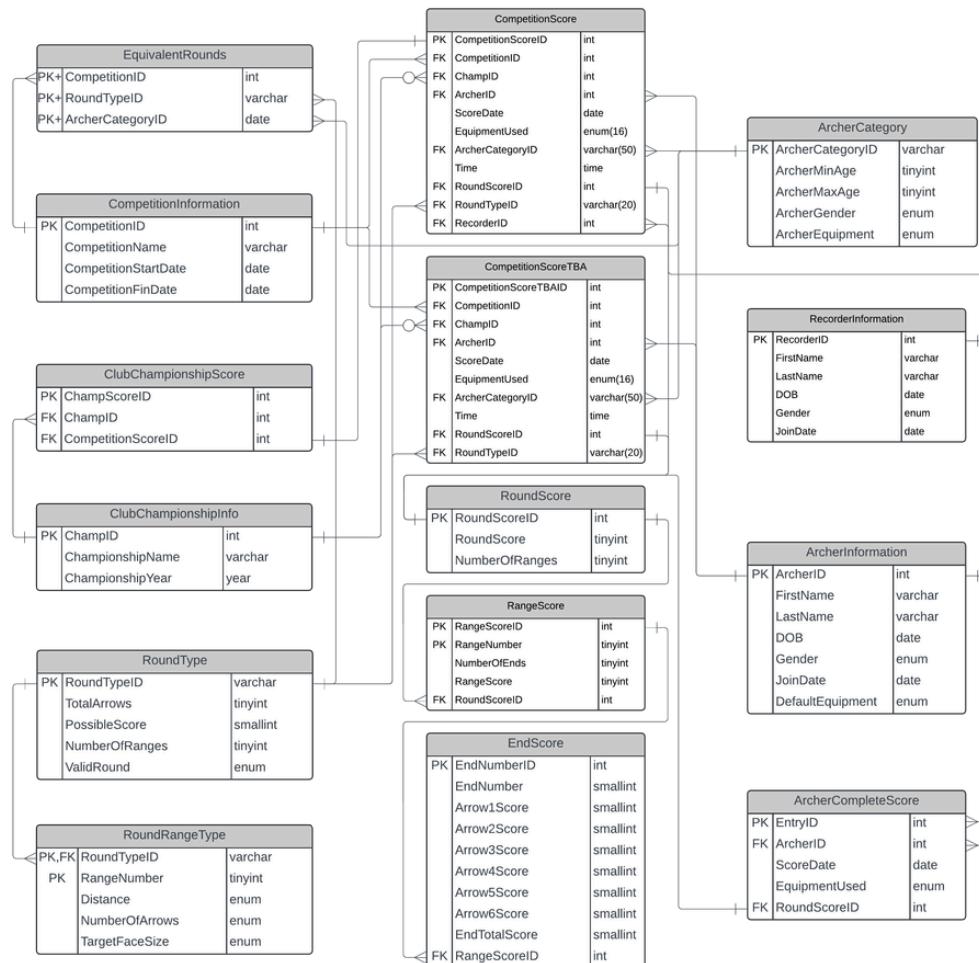
This iteration removes the underscores and the 'ArcherIndividualScore' table however is still missing the Equivalent rounds table.



Final ERD for E&BBAC Database

Attached is a copy of the finalized ERD for the Eggs and Bacon Bay Archery Clubs Database. It is attached in both SVG and PNG form, as well as a PDF attached.

Changes from the final Draft include the adding of 'EquivalentRounds' Table and removal of the ArcherCompleteScoreTBA.





ERD FINAL.pdf

Database Queries

CREATE DATABASE Individual Drafts

Each Team member created their own Database creation queries. This acted as a form of both brain-storming and error prevention. The Databases were then compared to one another to analyze their strengths, weaknesses and differences to ensure that the final Database creation commands meet the requirements.

- [Melvin and Khoa Draft](#)
- [Sukeyna Individual Draft](#)
- [Josh Individual Draft](#)
- [CREATE DATABASE Code](#)
- [Performance Indexes Implementation](#)

Some observed errors in our initial Drafts were:

- Not meeting certain requirements, some of our drafts did not include tables to account for arrow-by-arrow scores.
- Missing required information within tables, some tables did not contain fields that were required, for example, the Default Equipment field for the Archer Information table.
- Invalid syntax for MariaDB 5.5.68, which is the Database Management System being utilized on Fenix, because of this we had to reevaluate fields, commands and triggers as triggers are not supported.

Melvin and Khoa Draft

Unfortunately there was a small miscommunication and Melvin and Khoa created a draft between themselves.

```
1 CREATE TABLE `Archer` (
2   `Archer_ID` int(11) NOT NULL,
3   `First_Name` varchar(50) DEFAULT NULL,
4   `Last_Name` varchar(50) DEFAULT NULL,
5   `Age` int(11) DEFAULT NULL,
6   `Gender` varchar(10) DEFAULT NULL,
7   `Category` varchar(50) DEFAULT NULL
8 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
9
10 CREATE TABLE `Archer_Score` (
11   `Score_ID` int(11) NOT NULL,
12   `Archer_ID` int(11) DEFAULT NULL,
13   `Recorder_ID` int(11) DEFAULT NULL,
14   `Date_Archer` date DEFAULT NULL,
15   `Time_Archer` time DEFAULT NULL,
16   `Club_Championship_ID` int(11) DEFAULT NULL,
17   `Equipment` varchar(100) DEFAULT NULL,
18   `Round` varchar(50) DEFAULT NULL
19 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
20
21 CREATE TABLE `Best_Score` (
22   `Best_Score_ID` int(11) NOT NULL,
23   `Archer_ID` int(11) DEFAULT NULL,
24   `Club_Championship_ID` int(11) DEFAULT NULL,
25   `Score` decimal(10,2) DEFAULT NULL
26 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
27
28 CREATE TABLE `Club_Champion` (
29   `Club_Champion_ID` int(11) NOT NULL,
30   `Club_Champion_Name` varchar(50) DEFAULT NULL,
31   `Club_Champion_Date` date DEFAULT NULL,
32   `Club_Champion_Time` time DEFAULT NULL,
33   `Club_Champion_StartDate` date DEFAULT NULL,
34   `Club_Champion_EndDate` date DEFAULT NULL
35 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
36
37 CREATE TABLE `Club_Champion_Participation` (
38   `Participation_ID` int(11) NOT NULL,
39   `Archer_ID` int(11) DEFAULT NULL,
40   `Club_Championship_ID` int(11) DEFAULT NULL
41 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
42
43 CREATE TABLE `Recorder` (
44   `Recorder_ID` int(11) NOT NULL,
45   `Recorder_First_Name` varchar(50) DEFAULT NULL,
46   `Recorder_Last_Name` varchar(50) DEFAULT NULL
47 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
48
49 ALTER TABLE `Archer`
50   ADD PRIMARY KEY (`Archer_ID`);
51
```

```

52 ALTER TABLE `Archer_Score`  

53   ADD PRIMARY KEY (`Score_ID`),  

54   ADD KEY `Archer_ID` (`Archer_ID`),  

55   ADD KEY `Recorder_ID` (`Recorder_ID`),  

56   ADD KEY `Club_Championship_ID` (`Club_Championship_ID`);  

57  

58 ALTER TABLE `Best_Score`  

59   ADD PRIMARY KEY (`Best_Score_ID`),  

60   ADD KEY `Archer_ID` (`Archer_ID`),  

61   ADD KEY `Club_Championship_ID` (`Club_Championship_ID`);  

62  

63 ALTER TABLE `Club_Champion`  

64   ADD PRIMARY KEY (`Club_Champion_ID`);  

65  

66 ALTER TABLE `Club_Champion_Participation`  

67   ADD PRIMARY KEY (`Participation_ID`),  

68   ADD KEY `Archer_ID` (`Archer_ID`),  

69   ADD KEY `Club_Championship_ID` (`Club_Championship_ID`);  

70  

71 ALTER TABLE `Recorder`  

72   ADD PRIMARY KEY (`Recorder_ID`);  

73  

74 ALTER TABLE `Archer_Score`  

75   ADD CONSTRAINT `Archer_Score_ibfk_1` FOREIGN KEY (`Archer_ID`) REFERENCES `Archer` (`Archer_ID`),  

76   ADD CONSTRAINT `Archer_Score_ibfk_2` FOREIGN KEY (`Recorder_ID`) REFERENCES `Recorder` (`Recorder_ID`),  

77   ADD CONSTRAINT `Archer_Score_ibfk_3` FOREIGN KEY (`Club_Championship_ID`) REFERENCES `Club_Championship` (`Club_Championship_ID`);  

78  

79 ALTER TABLE `Best_Score`  

80   ADD CONSTRAINT `Best_Score_ibfk_1` FOREIGN KEY (`Archer_ID`) REFERENCES `Archer` (`Archer_ID`),  

81   ADD CONSTRAINT `Best_Score_ibfk_2` FOREIGN KEY (`Club_Championship_ID`) REFERENCES `Club_Championship` (`Club_Championship_ID`);  

82  

83 ALTER TABLE `Club_Champion_Participation`  

84   ADD CONSTRAINT `Club_Champion_Participation_ibfk_1` FOREIGN KEY (`Archer_ID`) REFERENCES `Archer` (`Archer_ID`),  

85   ADD CONSTRAINT `Club_Champion_Participation_ibfk_2` FOREIGN KEY (`Club_Championship_ID`) REFERENCES `Club_Championship` (`Club_Championship_ID`);  

86

```

Sukeyna Individual Draft

```
1  CREATE TABLE Archer (
2      Archer_ID INT PRIMARY KEY,
3      Archer_First_Name VARCHAR(255),
4      Archer_Last_Name VARCHAR(255),
5      Archer_DOB DATE,
6      Archer_Gender CHAR(1),
7      Archer_Join_Date DATE,
8      Archer_Default_Equipment VARCHAR(255)
9  );
10
11 CREATE TABLE Recorder (
12     Recorder_ID INT PRIMARY KEY,
13     Recorder_First_Name VARCHAR(255),
14     Recorder_Last_Name VARCHAR(255),
15     Recorder_DOB DATE,
16     Recorder_Gender CHAR(1),
17     Recorder_Join_Date DATE
18 );
19
20 CREATE TABLE Competition_Score (
21     Competition_Score_ID INT PRIMARY KEY,
22     Competition_ID INT,
23     Championship_ID INT,
24     Archer_ID INT,
25     Score_Date DATE,
26     Equipment_Used VARCHAR(255),
27     Archer_Category_ID INT,
28     Time_Of_Day TIME,
29     Recorded_Round INT,
30     Round_Score INT,
31     Round_Type_ID INT,
32     Possible_Score INT,
33     Recorder_ID INT,
34     FOREIGN KEY (Competition_ID),
35     REFERENCES Competition(Competition_ID),
36     FOREIGN KEY (Championship_ID),
37     REFERENCES Club_Championship(Championship_ID),
38     FOREIGN KEY (Archer_ID),
39     REFERENCES Archer(Archer_ID),
40     FOREIGN KEY (Archer_Category_ID),
41     REFERENCES Archer_Category(Archer_Category_ID),
42     FOREIGN KEY (Recorded_Round),
43     REFERENCES Recorded_Rounds(Recorded_Round),
44     FOREIGN KEY (Round_Type_ID),
45     REFERENCES Round_Types(Round_Type_ID),
46     FOREIGN KEY (Recorder_ID),
47     REFERENCES Recorder(Recorder_ID)
48 );
49
50 CREATE TABLE Competition_Score_to_be_Approved (
51     Pending_Score_ID INT PRIMARY KEY,
52     Archer_ID INT,
53     Score_Date DATE,
```

```

54 Equipment_Used VARCHAR(225),
55 Time_Of_Day TIME,
56 Recorded_Round INT,
57 Round_Score INT,
58 Round_Type_ID INT,
59 Possible_Score INT,
60 FOREIGN KEY (Archer_ID)
61 REFERENCES Archer(Archer_ID),
62 FOREIGN KEY (Recorded_Round)
63 REFERENCES Recorded_Rounds(Recorded_Round),
64 FOREIGN KEY (Round_Type_ID)
65 REFERENCES Round_Types(Round_Type_ID)
66 );
67
68 CREATE TABLE Recorded_Rounds (
69     Recorded_Round INT,
70     Round_Score INT,
71     Round_Type_ID INT,
72     Possible_Score INT,
73     Number_of_Ranges INT,
74     Range_Score_ID INT,
75     PRIMARY KEY (Recorded_Round, Round_Score),
76     FOREIGN KEY (Range_Score_ID)
77 REFERENCES Range_Score(Range_Score_ID)
78 );
79
80 CREATE TABLE Range_Score (
81     Range_Score_ID INT PRIMARY KEY,
82     Number_Of_Ends INT,
83     End_Number_ID INT,
84     Range_Score INT,
85     Range_Number INT,
86     FOREIGN KEY (End_Number_ID)
87 REFERENCES End_Score(End_Number_ID)
88 );
89
90 CREATE TABLE Archer_All_Scores (
91     Entry_ID INT PRIMARY KEY,
92     Archer_ID INT,
93     Score_Date DATE,
94     Equipment_Used VARCHAR(225),
95     Score_ID INT,
96     Total_Score INT,
97     Possible_Score INT,
98     Recorder_ID INT,
99     FOREIGN KEY (Archer_ID)
100 REFERENCES Archer(Archer_ID),
101 FOREIGN KEY (Score_ID)
102 REFERENCES Archer_Individual_Score(Score_ID),
103 FOREIGN KEY (Recorder_ID)
104 REFERENCES Recorder(Recorder_ID)
105 );
106
107 CREATE TABLE Archer_Individual_Scores_to_be_Approved (
108     Entry_To_Be_Approved_ID INT
109 PRIMARY KEY,
110     Archer_ID INT,
111     Score_Date DATE,

```

```

112     Equipment_Used VARCHAR(225),
113     Score_ID INT,
114     FOREIGN KEY (Archer_ID)
115 REFERENCES Archer(Archer_ID),
116     FOREIGN KEY (Score_ID)
117 REFERENCES Archer_Individual_Score(Score_ID)
118 );
119
120 CREATE TABLE Archer_Individual_Score (
121     Score_ID INT PRIMARY KEY,
122     End_Number_ID INT,
123     Total_Score INT,
124     Possible_Score INT,
125     FOREIGN KEY (End_Number_ID)
126 REFERENCES End_Score(End_Number_ID)
127 );
128
129 CREATE TABLE Club_Championship (
130     Championship_ID INT PRIMARY KEY,
131     Championship_Name VARCHAR(225),
132     Championship_Year YEAR,
133     Archer_ID INT,
134     Score_ID INT,
135     Competition_ID INT,
136     FOREIGN KEY (Archer_ID)
137 REFERENCES Archer(Archer_ID),
138     FOREIGN KEY (Score_ID)
139 REFERENCES Archer_Individual_Score(Score_ID),
140     FOREIGN KEY (Competition_ID)
141 REFERENCES Competition(Competition_ID)
142 );
143
144 CREATE TABLE Competition (
145     Competition_ID INT PRIMARY KEY,
146     Competition_Name VARCHAR(225),
147     Competition_Start_Date DATE,
148     Competition_Finish_Date DATE
149 );
150
151 CREATE TABLE Round_Types (
152     Round_Type_ID INT,
153     Total_Arrows INT,
154     Possible_Score INT,
155     Number_Of_Arrows INT,
156     Total_Score INT,
157     Distance INT,
158     Target_Face_Size INT,
159     Valid_Round BOOLEAN,
160     PRIMARY KEY (Round_Type_ID, Total_Arrows, Possible_Score)
161 );
162
163 CREATE TABLE Archer_Category (
164     Archer_Category_ID INT PRIMARY KEY,
165     Archer_Age INT,
166     Archer_Gender CHAR(1),
167     Archer_Equipment VARCHAR(225)
168 );
169

```

```

170 CREATE TABLE End_Score (
171     End_Number_ID INT PRIMARY KEY,
172     Arrow_Score_ID_1 INT,
173     Arrow_Score_ID_2 INT,
174     Arrow_Score_ID_3 INT,
175     Arrow_Score_ID_4 INT,
176     Arrow_Score_ID_5 INT,
177     Arrow_Score_ID_6 INT,
178     End_Number INT,
179     End_Score INT,
180     FOREIGN KEY (Arrow_Score_ID_1)
181     REFERENCES Arrow_Score(Arrow_Score_ID),
182     FOREIGN KEY (Arrow_Score_ID_2)
183     REFERENCES Arrow_Score(Arrow_Score_ID),
184     FOREIGN KEY (Arrow_Score_ID_3)
185     REFERENCES Arrow_Score(Arrow_Score_ID),
186     FOREIGN KEY (Arrow_Score_ID_4)
187     REFERENCES Arrow_Score(Arrow_Score_ID),
188     FOREIGN KEY (Arrow_Score_ID_5)
189     REFERENCES Arrow_Score(Arrow_Score_ID),
190     FOREIGN KEY (Arrow_Score_ID_6)
191     REFERENCES Arrow_Score(Arrow_Score_ID)
192 );
193
194 CREATE TABLE Arrow_Score (
195     Arrow_Score_ID INT,
196     End_Number_ID INT,
197     Arrow_Score INT,
198     PRIMARY KEY (Arrow_Score_ID,End_Number_ID)
199 );
200
201 ALTER TABLE Competition_Score
202 ADD
203 CONSTRAINT fk_competition_id
204     FOREIGN KEY (Competition_ID)
205     REFERENCES Competition(Competition_ID),
206 ADD
207 CONSTRAINT fk_championship_id
208     FOREIGN KEY (Championship_ID)
209     REFERENCES Club_Championship(Championship_ID),
210 ADD
211 CONSTRAINT fk_archer_id
212     FOREIGN KEY (Archer_ID)
213     REFERENCES Archer(Archer_ID),
214 ADD
215 CONSTRAINT fk_archer_category_id
216     FOREIGN KEY (Archer_Category_ID)
217     REFERENCES Archer_Category(Archer_Category_ID),
218 ADD
219 CONSTRAINT fk_recorded_round
220     FOREIGN KEY (Recorded_Round)
221     REFERENCES Recorded_Rounds(Recorded_Round),
222 ADD
223 CONSTRAINT fk_round_type_id
224     FOREIGN KEY (Round_Type_ID)
225     REFERENCES Round_Types(Round_Type_ID),
226 ADD
227 CONSTRAINT fk_recorder_id

```

```
228 FOREIGN KEY (Recorder_ID)
229 REFERENCES Recorder(Recorder_ID);
230
231 ALTER TABLE Competition_Score_to_be_Approved
232 ADD
233 CONSTRAINT fk_archer_id2
234 FOREIGN KEY (Archer_ID)
235 REFERENCES Archer(Archer_ID),
236 ADD
237 CONSTRAINT fk_recorded_round2
238 FOREIGN KEY (Recorded_Round)
239 REFERENCES Recorded_Rounds(Recorded_Round),
240 ADD
241 CONSTRAINT fk_round_type_id2
242 FOREIGN KEY (Round_Type_ID)
243 REFERENCES Round
```

Josh Individual Draft

```
1 CREATE TABLE `Competition_Score` (
2     `Comp_Score_ID` INT AUTO_INCREMENT UNIQUE,
3     `Comp_ID` INT,
4     `Champ_ID` INT,
5     `Archer_ID` INT,
6     `Score_Date` DATETIME,
7     `Equipment_Used` ENUM() DEFAULT 'Archer_Default_Equipment',
8     `Archer_Category_ID` VARCHAR(255),
9     `Time` TIME,
10    `Recorded_Round` INT,
11    `Round_Type_ID` VARCHAR(255),
12    `Recorder_ID` INT,
13    PRIMARY KEY(`Comp_Score_ID`)
14 );
15
16 CREATE TABLE `Competition_Score_TBA` (
17     `Competition_Score_TBA_ID` INT AUTO_INCREMENT UNIQUE,
18     `Comp_ID` INT,
19     `Champ_ID` INT,
20     `Archer_ID` INT,
21     `Score_Date` DATE,
22     `Equipment_Used` ENUM() DEFAULT 'Archer_Default_Equipment',
23     `Archer_Category_ID` VARCHAR(255),
24     `Time` TIME,
25     `Recorded_Round` INT,
26     `Round_Type_ID` VARCHAR(255),
27     PRIMARY KEY(`Competition_Score_TBA_ID`)
28 );
29
30 CREATE TABLE `Recorded_Rounds` (
31     `Recorded_Round` INT AUTO_INCREMENT UNIQUE,
32     `Round_Score` SMALLINT,
33     `Possible_Score` SMALLINT,
34     `Number_Of_Ranges` SMALLINT,
35     `Range_Score_ID` INT,
36     PRIMARY KEY(`Recorded_Round`)
37 );
38
39 CREATE TABLE `Range_Score` (
40     `Range_Score_ID` INT AUTO_INCREMENT UNIQUE,
41     `Number_Of_Ends` SMALLINT,
42     `End_Number_ID` INT,
43     `Range_Score` SMALLINT,
44     `Range_Number` SMALLINT,
45     PRIMARY KEY(`Range_Score_ID`, `Number_Of_Ends`)
46 );
47
48 CREATE TABLE `End_Scores` (
49     `End_Number_ID` INT NOT NULL AUTO_INCREMENT UNIQUE,
50     `Arrow_1_Score` SMALLINT,
51     `Arrow_2_Score` SMALLINT,
52     `Arrow_3_Score` SMALLINT,
53     `Arrow_4_Score` SMALLINT,
```

```

54     `Arrow_5_Score` SMALLINT,
55     `Arrow_6_Score` SMALLINT,
56     `End_Number` SMALLINT,
57     `End_Total_Score` SMALLINT,
58     PRIMARY KEY(`End_Number_ID`)
59 );
60
61 CREATE TABLE `Archer_All_Scores` (
62     `Entry_ID` INT AUTO_INCREMENT UNIQUE,
63     `Archer_ID` INT,
64     `Score_Date` DATE,
65     `Equipment_Used` ENUM(),
66     `Individual_Entry_ID` INT,
67     `Total_Score` SMALLINT,
68     `Possible_Score` SMALLINT,
69     `Recorder_ID` INT,
70     PRIMARY KEY(`Entry_ID`, `Archer_ID`)
71 );
72
73 CREATE TABLE `Archer_Individual_Scores_TBA` (
74     `Individual_Entry_ID` INT AUTO_INCREMENT UNIQUE,
75     `Archer_ID` INT,
76     `Score_Date` DATE,
77     `Equipment_Used` ENUM(),
78     `Score_ID` INT,
79     `Total_Score` SMALLINT,
80     `Possible_Score` SMALLINT,
81     PRIMARY KEY(`Individual_Entry_ID`)
82 );
83
84 CREATE TABLE `Archer_Individual_Score` (
85     `Score_ID` INT NOT NULL AUTO_INCREMENT UNIQUE,
86     `End_Number_ID` INT,
87     `Total_Score` SMALLINT,
88     `Possible_Score` SMALLINT,
89     PRIMARY KEY(`Score_ID`, `Total_Score`)
90 );
91
92 CREATE TABLE `Club_Championship` (
93     `Champ_ID` INT NOT NULL AUTO_INCREMENT UNIQUE,
94     `Championship_Name` VARCHAR(255),
95     `Championship_year` SMALLINT,
96     `Archer_ID` INT,
97     `Recorded_Round_ID` INT,
98     `Competition_ID` INT,
99     `Archer_Category_ID` VARCHAR(255),
100    PRIMARY KEY(`Champ_ID`)
101 );
102
103 CREATE TABLE `Competitions` (
104     `Comp_ID` INT NOT NULL AUTO_INCREMENT UNIQUE,
105     `Comp_Name` VARCHAR(255),
106     `Comp_Start_Date` DATE,
107     `Comp_Fin_Date` DATE,
108     PRIMARY KEY(`Comp_ID`)
109 );
110
111 CREATE TABLE `Round_Types` (

```

```

112   `Round_Type_ID` VARCHAR(255) NOT NULL UNIQUE,
113   `Possible_Score` SMALLINT,
114   `Total_Arrows` SMALLINT,
115   `Distance` ENUM(),
116   `Number_Of_Arrows` ENUM(),
117   `Target_Face_Size` ENUM(),
118   `Valid_Round` BOOLEAN DEFAULT True,
119   PRIMARY KEY(`Round_Type_ID`)
120 );
121
122 CREATE TABLE `Recorders` (
123   `Recorder_ID` INT NOT NULL AUTO_INCREMENT UNIQUE,
124   `First_Name` VARCHAR(255),
125   `Last_Name` VARCHAR(255),
126   `DOB` DATE,
127   `Gender` ENUM(),
128   `Join_Date` DATE,
129   PRIMARY KEY(`Recorder_ID`)
130 );
131
132 CREATE TABLE `Archers` (
133   `Archer_ID` INT NOT NULL AUTO_INCREMENT UNIQUE,
134   `First_Name` VARCHAR(255),
135   `Last_Name` VARCHAR(255),
136   `DOB` DATE,
137   `Gender` ENUM(),
138   `Join_Date` DATE,
139   `Default_Equipment` ENUM(),
140   PRIMARY KEY(`Archer_ID`)
141 );
142
143 CREATE TABLE `Archer_Categories` (
144   `Archer_Category_ID` VARCHAR(255) NOT NULL UNIQUE,
145   `Archer_Age` SMALLINT,
146   `Archer_Gender` ENUM(),
147   `Archer_Equipment` ENUM(),
148   PRIMARY KEY(`Archer_Category_ID`)
149 );
150
151 ALTER TABLE `Competition_Score`  

152 ADD FOREIGN KEY(`Comp_ID`) REFERENCES `Competitions`(`Comp_ID`)  

153 ON UPDATE NO ACTION ON DELETE NO ACTION;  

154 ALTER TABLE `Competition_Score`  

155 ADD FOREIGN KEY(`Champ_ID`) REFERENCES `Club_Championship`(`Champ_ID`)  

156 ON UPDATE NO ACTION ON DELETE NO ACTION;  

157 ALTER TABLE `Competition_Score`  

158 ADD FOREIGN KEY(`Archer_ID`) REFERENCES `Archers`(`Archer_ID`)  

159 ON UPDATE NO ACTION ON DELETE NO ACTION;  

160 ALTER TABLE `Competition_Score`  

161 ADD FOREIGN KEY(`Recorded_Round`) REFERENCES `Recorded_Rounds`(`Recorded_Round`)  

162 ON UPDATE NO ACTION ON DELETE NO ACTION;  

163 ALTER TABLE `Competition_Score`  

164 ADD FOREIGN KEY(`Round_Type_ID`) REFERENCES `Round_Types`(`Round_Type_ID`)  

165 ON UPDATE NO ACTION ON DELETE NO ACTION;  

166 ALTER TABLE `Competition_Score`  

167 ADD FOREIGN KEY(`Recorder_ID`) REFERENCES `Recorders`(`Recorder_ID`)  

168 ON UPDATE NO ACTION ON DELETE NO ACTION;  

169 ALTER TABLE `Competition_Score`  


```

```

170 ADD FOREIGN KEY(`Archer_Category_ID`) REFERENCES `Archer_Categories`(`Archer_Category_ID`)
171 ON UPDATE NO ACTION ON DELETE NO ACTION;
172 ALTER TABLE `Competition_Score_TBA`
173 ADD FOREIGN KEY(`Comp_ID`) REFERENCES `Competitions`(`Comp_ID`)
174 ON UPDATE NO ACTION ON DELETE NO ACTION;
175 ALTER TABLE `Club_Championship`
176 ADD FOREIGN KEY(`Champ_ID`) REFERENCES `Competition_Score_TBA`(`Champ_ID`)
177 ON UPDATE NO ACTION ON DELETE NO ACTION;
178 ALTER TABLE `Archers`
179 ADD FOREIGN KEY(`Archer_ID`) REFERENCES `Competition_Score_TBA`(`Archer_ID`)
180 ON UPDATE NO ACTION ON DELETE NO ACTION;
181 ALTER TABLE `Archer_Categories`
182 ADD FOREIGN KEY(`Archer_Category_ID`) REFERENCES `Competition_Score_TBA`(`Archer_Category_ID`)
183 ON UPDATE NO ACTION ON DELETE NO ACTION;
184 ALTER TABLE `Recorded_Rounds`
185 ADD FOREIGN KEY(`Recorded_Round`) REFERENCES `Competition_Score_TBA`(`Recorded_Round`)
186 ON UPDATE NO ACTION ON DELETE NO ACTION;
187 ALTER TABLE `Round_Types`
188 ADD FOREIGN KEY(`Round_Type_ID`) REFERENCES `Competition_Score_TBA`(`Round_Type_ID`)
189 ON UPDATE NO ACTION ON DELETE NO ACTION;
190 ALTER TABLE `End_Scores`
191 ADD FOREIGN KEY(`End_Number_ID`) REFERENCES `Range_Score`(`End_Number_ID`)
192 ON UPDATE NO ACTION ON DELETE NO ACTION;
193 ALTER TABLE `Range_Score`
194 ADD FOREIGN KEY(`Range_Score_ID`) REFERENCES `Recorded_Rounds`(`Range_Score_ID`)
195 ON UPDATE NO ACTION ON DELETE NO ACTION;
196 ALTER TABLE `End_Scores`
197 ADD FOREIGN KEY(`End_Number_ID`) REFERENCES `Archer_Individual_Score`(`End_Number_ID`)
198 ON UPDATE NO ACTION ON DELETE NO ACTION;
199 ALTER TABLE `Archer_Individual_Score`
200 ADD FOREIGN KEY(`Score_ID`) REFERENCES `Archer_Individual_Scores_TBA`(`Score_ID`)
201 ON UPDATE NO ACTION ON DELETE NO ACTION;
202 ALTER TABLE `Archers`
203 ADD FOREIGN KEY(`Archer_ID`) REFERENCES `Archer_Individual_Scores_TBA`(`Archer_ID`)
204 ON UPDATE NO ACTION ON DELETE NO ACTION;
205 ALTER TABLE `Archers`
206 ADD FOREIGN KEY(`Archer_ID`) REFERENCES `Archer_All_Scores`(`Archer_ID`)
207 ON UPDATE NO ACTION ON DELETE NO ACTION;
208 ALTER TABLE `Archer_Individual_Scores_TBA`
209 ADD FOREIGN KEY(`Individual_Entry_ID`) REFERENCES `Archer_All_Scores`(`Individual_Entry_ID`)
210 ON UPDATE NO ACTION ON DELETE NO ACTION;
211 ALTER TABLE `Recorders`
212 ADD FOREIGN KEY(`Recorder_ID`) REFERENCES `Archer_All_Scores`(`Recorder_ID`)
213 ON UPDATE NO ACTION ON DELETE NO ACTION;
214 ALTER TABLE `Archers`
215 ADD FOREIGN KEY(`Archer_ID`) REFERENCES `Club_Championship`(`Archer_ID`)
216 ON UPDATE NO ACTION ON DELETE NO ACTION;
217 ALTER TABLE `Recorded_Rounds`
218 ADD FOREIGN KEY(`Recorded_Round`) REFERENCES `Club_Championship`(`Recorded_Round_ID`)
219 ON UPDATE NO ACTION ON DELETE NO ACTION;
220 ALTER TABLE `Competitions`
221 ADD FOREIGN KEY(`Comp_ID`) REFERENCES `Club_Championship`(`Competition_ID`)
222 ON UPDATE NO ACTION ON DELETE NO ACTION;
223 ALTER TABLE `Archer_Categories`
224 ADD FOREIGN KEY(`Archer_Category_ID`) REFERENCES `Club_Championship`(`Archer_Category_ID`)
225 ON UPDATE NO ACTION ON DELETE NO ACTION;

```

CREATE DATABASE Code

Below is the final code used to create the final database. It fulfills all requirements set out in the Project Brief and is compatible with MariaDB5.5.68

```
1 CREATE TABLE `ArcherInformation` (
2     `ArcherID` INT (7) NOT NULL AUTO_INCREMENT,
3     `FirstName` VARCHAR(20) NOT NULL,
4     `LastName` VARCHAR(20) NOT NULL,
5     `DOB` DATE NOT NULL,
6     `Gender` ENUM('Female','Male') NOT NULL,
7     `JoinDate` DATE NOT NULL,
8     `DefaultEquipment` ENUM('Recurve', 'Compound', 'Recurve Barebow', 'Compound Barebow', 'Longbow') NOT NULL,
9     PRIMARY KEY(`ArcherID`)
10 );
11 CREATE TABLE `RecorderInformation` (
12     `RecorderID` INT(7) NOT NULL AUTO_INCREMENT,
13     `FirstName` VARCHAR(20) NOT NULL,
14     `LastName` VARCHAR(20) NOT NULL,
15     `DOB` DATE NOT NULL,
16     `Gender` ENUM('Male','Female') NOT NULL,
17     `JoinDate` DATE NOT NULL COMMENT 'The date that the recorder became a recorder. For example if the recorder
18 PRIMARY KEY(`RecorderID`)
19 );
20 CREATE TABLE `ArcherCategory` (
21     `ArcherCategoryID` VARCHAR(40) NOT NULL,
22     `ArcherMinAge` INT(2) NOT NULL,
23     `ArcherMaxAge` INT(2) NOT NULL,
24     `ArcherGender` ENUM('Male','Female') NOT NULL,
25     `ArcherEquipment` ENUM('Recurve', 'Compound', 'Recurve Barebow', 'Compound Barebow', 'Longbow') NOT NULL,
26     PRIMARY KEY(`ArcherCategoryID`)
27 );
28 CREATE TABLE `CompetitionInformation` (
29     `CompetitionID` INT(7) NOT NULL AUTO_INCREMENT,
30     `CompetitionName` VARCHAR(50) NOT NULL,
31     `CompetitionStartDate` DATE NOT NULL,
32     `CompetitionFinDate` DATE NOT NULL,
33     PRIMARY KEY (`CompetitionID`)
34 );
35 CREATE TABLE `RoundType` (
36     `RoundTypeID` VARCHAR(40) NOT NULL,
37     `TotalArrows` INT(3) NOT NULL,
38     `PossibleScore` INT(4) NOT NULL,
39     `NumberOfRanges` INT(1) NOT NULL,
40     `ValidRound` ENUM('Yes', 'No') DEFAULT 'Yes' NOT NULL,
41     PRIMARY KEY (`RoundTypeID`)
42 );
43 CREATE TABLE `RoundTypeRange` (
44     `RoundTypeID` VARCHAR(40) NOT NULL,
45     `RangeNumber` INT(1) NOT NULL,
46     `Distance` ENUM('20m','30m','40m','50m','60m','70m','90m') NOT NULL,
47     `NumberOfArrows` ENUM('30','36') NOT NULL,
48     `TargetFaceSize` ENUM('80cm','122cm') NOT NULL,
49     PRIMARY KEY (`RoundTypeID`, `RangeNumber`),
50     FOREIGN KEY (`RoundTypeID`) REFERENCES `RoundType`(`RoundTypeID`)
```

```

51 );
52 CREATE TABLE `ClubChampionshipInfo` (
53     `ChampID` INT(7) NOT NULL AUTO_INCREMENT,
54     `ChampionshipName` VARCHAR(50) NOT NULL,
55     `ChampionshipYear` YEAR NOT NULL,
56     PRIMARY KEY(`ChampID`)
57 );
58 CREATE TABLE `ClubChampionshipScore` (
59     `ChampScoreID` INT(7) NOT NULL AUTO_INCREMENT,
60     `ChampID` INT(7) NOT NULL, /*FK*/
61     `CompetitionScoreID` INT(7) NOT NULL, /*FK*/
62     PRIMARY KEY(`ChampScoreID`)
63 );
64 CREATE TABLE `CompetitionScore` (
65     `CompetitionScoreID` INT(7) NOT NULL AUTO_INCREMENT,
66     `CompetitionID` INT(7) NOT NULL, /*FK*/
67     `ChampID` INT(7), /*FK*/
68     `ArcherID` INT(7) NOT NULL, /*FK*/
69     `ScoreDate` DATE NOT NULL,
70     `EquipmentUsed` ENUM('Recurve', 'Compound', 'Recurve Barebow', 'Compound Barebow', 'Longbow'),
71     `ArcherCategoryID` VARCHAR(40) NOT NULL, /*Using age and gender from ArcherID and Equipment Used*/
72     `Time` TIME NOT NULL,
73     `RoundScoreID` INT(7) NOT NULL, /*FK*/
74     `RoundTypeID` VARCHAR(40) NOT NULL, /*FK*/
75     `RecorderID` INT(7) NOT NULL, /*FK*/
76     PRIMARY KEY(`CompetitionScoreID`)
77 );
78 CREATE TABLE `CompetitionScoreTBA` (
79     `CompetitionScoreTBAID` INT(7) NOT NULL AUTO_INCREMENT,
80     `CompetitionID` INT(7) NOT NULL, /*FK*/
81     `ChampID` INT(7), /*FK*/
82     `ArcherID` INT(7) NOT NULL, /*FK*/
83     `ScoreDate` DATE NOT NULL,
84     `EquipmentUsed` ENUM('Recurve', 'Compound', 'Recurve Barebow', 'Compound Barebow', 'Longbow'),
85     `ArcherCategoryID` VARCHAR(40) NOT NULL, /*FK*/
86     `Time` TIME NOT NULL,
87     `RoundScoreID` INT(7) NOT NULL, /*FK*/
88     `RoundTypeID` VARCHAR(40) NOT NULL, /*FK*/
89     PRIMARY KEY(`CompetitionScoreTBAID`)
90 );
91 CREATE TABLE `RoundScore` (
92     `RoundScoreID` INT(7) NOT NULL AUTO_INCREMENT,
93     `RoundScore` INT(4) NOT NULL, /*Is equal to the Total Score of all ranges associated with the RecordedRound*/
94     `NumberOfRanges` INT(1) NOT NULL,
95     PRIMARY KEY (`RoundScoreID`)
96 );
97 CREATE TABLE `RangeScore` (
98     `RangeScoreID` INT(7) NOT NULL AUTO_INCREMENT,
99     `RangeNumber` INT(1) NOT NULL,
100    `NumberOfEnds` INT(1) NOT NULL,
101    `RangeScore` INT(3) NOT NULL, /*Is equal to the Total Score of all ends associated with the range scoreID*/
102    `RoundScoreID` INT(7) NOT NULL, /*FK*/
103    PRIMARY KEY(`RangeScoreID`)
104 );
105 CREATE TABLE `EndScore` (
106     `EndNumberID` INT(7) NOT NULL AUTO_INCREMENT,
107     `EndNumber` INT(1) NOT NULL,
108     `Arrow1Score` INT(2) NOT NULL,

```

```

109   `Arrow2Score` INT(2) NOT NULL,
110   `Arrow3Score` INT(2) NOT NULL,
111   `Arrow4Score` INT(2) NOT NULL,
112   `Arrow5Score` INT(2) NOT NULL,
113   `Arrow6Score` INT(2) NOT NULL,
114   `EndTotalScore` INT(2) NOT NULL,
115   `RangeScoreID` INT(7) NOT NULL, /*FK*/
116   PRIMARY KEY (`EndNumberID`)
117 );
118 CREATE TABLE `ArcherCompleteScore` (
119   `EntryID` INT(7) NOT NULL AUTO_INCREMENT,
120   `ArcherID` INT(7) NOT NULL, /*FK*/
121   `ScoreDate` DATE NOT NULL,
122   `EquipmentUsed` ENUM('Recurve', 'Compound', 'Recurve Barebow', 'Compound Barebow', 'Longbow'),
123   `RoundScoreID` INT(7) NOT NULL, /*FK*/
124   PRIMARY KEY (`EntryID`)
125 );
126 CREATE TABLE `EquivalentRounds` (
127   `CompetitionID` INT(7) NOT NULL,
128   `RoundTypeID` VARCHAR(40) NOT NULL,
129   `ArcherCategoryID` VARCHAR(40) NOT NULL,
130   PRIMARY KEY (`CompetitionID`, `RoundTypeID`, `ArcherCategoryID`)
131 );
132
133 ALTER TABLE `ArcherCompleteScore`
134   ADD FOREIGN KEY (`ArcherID`) REFERENCES `ArcherInformation`(`ArcherID`),
135   ADD FOREIGN KEY (`RoundScoreID`) REFERENCES `RoundScore`(`RoundScoreID`);
136
137 ALTER TABLE `EndScore`
138   ADD FOREIGN KEY (`RangeScoreID`) REFERENCES `RangeScore`(`RangeScoreID`);
139
140 ALTER TABLE `RangeScore`
141   ADD FOREIGN KEY (`RoundScoreID`) REFERENCES `RoundScore`(`RoundScoreID`);
142
143 ALTER TABLE `CompetitionScoreTBA`
144   ADD FOREIGN KEY (`CompetitionID`) REFERENCES `CompetitionInformation`(`CompetitionID`),
145   ADD FOREIGN KEY (`ChampID`) REFERENCES `ClubChampionshipInfo`(`ChampID`),
146   ADD FOREIGN KEY (`ArcherID`) REFERENCES `ArcherInformation`(`ArcherID`),
147   ADD FOREIGN KEY (`ArcherCategoryID`) REFERENCES `ArcherCategory`(`ArcherCategoryID`),
148   ADD FOREIGN KEY (`RoundScoreID`) REFERENCES `RoundScore`(`RoundScoreID`),
149   ADD FOREIGN KEY (`RoundTypeID`) REFERENCES `RoundType`(`RoundTypeID`);
150
151 ALTER TABLE `CompetitionScore`
152   ADD FOREIGN KEY (`CompetitionID`) REFERENCES `CompetitionInformation`(`CompetitionID`),
153   ADD FOREIGN KEY (`ChampID`) REFERENCES `ClubChampionshipInfo`(`ChampID`),
154   ADD FOREIGN KEY (`ArcherID`) REFERENCES `ArcherInformation`(`ArcherID`),
155   ADD FOREIGN KEY (`ArcherCategoryID`) REFERENCES `ArcherCategory`(`ArcherCategoryID`),
156   ADD FOREIGN KEY (`RoundScoreID`) REFERENCES `RoundScore`(`RoundScoreID`),
157   ADD FOREIGN KEY (`RoundTypeID`) REFERENCES `RoundType`(`RoundTypeID`),
158   ADD FOREIGN KEY (`RecorderID`) REFERENCES `RecorderInformation`(`RecorderID`);
159
160 ALTER TABLE `ClubChampionshipScore`
161   ADD FOREIGN KEY (`ChampID`) REFERENCES `ClubChampionshipInfo`(`ChampID`),
162   ADD FOREIGN KEY (`CompetitionScoreID`) REFERENCES `CompetitionScore`(`CompetitionScoreID`);

```



Performance Indexes Implementation

- **Composite Index:** (Championshipyear, ChampionshipName) ClubChampionshipInfo Table

```
CREATE INDEX idxClubChampionshipYear ON ClubChampionshipInfo (Championshipyear);
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
ClubChampionshipInfo	0	PRIMARY	1	ChampID	A	1	NULL NULL	BTREE				
ClubChampionshipInfo	1	idxClubChampionshipYearName	1	Championshipyear	A	1	NULL NULL	BTREE				
ClubChampionshipInfo	1	idxClubChampionshipYearName	2	ChampionshipName	A	1	NULL NULL	BTREE				
ClubChampionshipInfo	1	idxClubChampionshipYear	1	Championshipyear	A	1	NULL NULL	BTREE				

- **Index:** (LastName, FirstName) RecorderInformation Table

```
CREATE INDEX idxRecordersName ON RecorderInformation (LastName, FirstName);
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
RecorderInformation	0	PRIMARY	1	RecorderID	A	1091	NULL NULL	BTREE				
RecorderInformation	1	idxRecordersName	1	LastName	A	218	NULL NULL	BTREE				
RecorderInformation	1	idxRecordersName	2	FirstName	A	218	NULL NULL	BTREE				

- **Index:** (LastName, FirstName) ArcherInformation Table

```
CREATE INDEX idxArchersName ON ArcherInformation (LastName, FirstName);
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
ArcherInformation	0	PRIMARY	1	ArcherID	A	939	NULL NULL	BTREE				
ArcherInformation	1	idxArchersName	1	LastName	A	187	NULL NULL	BTREE				
ArcherInformation	1	idxArchersName	2	FirstName	A	187	NULL NULL	BTREE				

- **Index:** (CompetitionStartDate) CompetitionInformation Table

```
CREATE INDEX idxCompetitionStartDate ON CompetitionInformation (CompetitionStartDate);
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
CompetitionInformation	0	PRIMARY	1	CompetitionID	A	2	NULL NULL	BTREE				
CompetitionInformation	1	idxCompetitionDate	1	CompetitionStartDate	A	2	NULL NULL	BTREE				
CompetitionInformation	1	idxCompetitionDate	2	CompetitionFinDate	A	2	NULL NULL	BTREE				
CompetitionInformation	1	idxCompetitionStartDate	1	CompetitionStartDate	A	2	NULL NULL	BTREE				
CompetitionInformation	1	idxCompetitionFinDate	1	CompetitionFinDate	A	2	NULL NULL	BTREE				

- **Index:** (CompetitionFinDate) CompetitionInformation Table

```
CREATE INDEX idxCompetitionFinDate ON CompetitionInformation (CompetitionFinDate);
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
CompetitionInformation	0	PRIMARY	1	CompetitionID	A	2	NULL NULL	BTREE				
CompetitionInformation	1	idxCompetitionDate	1	CompetitionStartDate	A	2	NULL NULL	BTREE				
CompetitionInformation	1	idxCompetitionDate	2	CompetitionFinDate	A	2	NULL NULL	BTREE				
CompetitionInformation	1	idxCompetitionStartDate	1	CompetitionStartDate	A	2	NULL NULL	BTREE				
CompetitionInformation	1	idxCompetitionFinDate	1	CompetitionFinDate	A	2	NULL NULL	BTREE				

- **Index:** (ArcherMinAge, ArcherMaxAge) ArcherCategory Table

```
CREATE INDEX idxArcherAgeRange ON ArcherCategory (ArcherMinAge, ArcherMaxAge);
```

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment
ArcherCategory	0	PRIMARY	1	ArcherCategoryID	A	1	NULL NULL			BTREE		
ArcherCategory	1	idxArcherCategories	1	ArcherMinAge	A	1	NULL NULL			BTREE		
ArcherCategory	1	idxArcherCategories	2	ArcherMaxAge	A	1	NULL NULL			BTREE		
ArcherCategory	1	idxArcherCategories	3	ArcherGender	A	1	NULL NULL			BTREE		
ArcherCategory	1	idxArcherCategories	4	ArcherEquipment	A	1	NULL NULL			BTREE		
ArcherCategory	1	idxArcherAgeRange	1	ArcherMinAge	A	1	NULL NULL			BTREE		
ArcherCategory	1	idxArcherAgeRange	2	ArcherMaxAge	A	1	NULL NULL			BTREE		

- **Index:** (ScoreDate) CompetitionScore Table

```
CREATE INDEX idxCompetitionScoreScoreDate ON CompetitionScore (ScoreDate);
```

CompetitionScore	1	idxCompetitionScoreScoreDate	1	ScoreDate	A	1	NULL NULL	BTREE
------------------	---	------------------------------	---	-----------	---	---	-----------	-------

- **Index:** (ScoreDate) CompetitionScoreTBA Table

```
CREATE INDEX idxCompetitionScoreTbaScoreDate ON CompetitionScoreTBA (ScoreDate);
```

CompetitionScoreTBA	1	idxCompetitionScoreTbaScoreDate	1	ScoreDate	A	1	NULL NULL	BTREE
---------------------	---	---------------------------------	---	-----------	---	---	-----------	-------

Database Interaction Queries

SQL commands: Sukeyna

Requirement: Archers should be able to access information about different rounds including distances, number of ends, target faces, and total number of ends.

```
1 SELECT t.RoundTypeID, t.TotalArrows, t.PossibleScore, t.NumberOfRanges, r.RangeNumber, r.Distance, r.NumberOfArrows
2 FROM RoundType AS t JOIN RoundTypeRange AS r ON t.RoundTypeID = r.RoundTypeID
3 WHERE t.RoundTypeID = 'Long Sydney';
```

The screenshot shows a MySQL query results page. At the top, it says "Showing rows 0 - 3 (4 total, Query took 0.0021 seconds.)". Below that is the SQL query: "SELECT t.RoundTypeID, t.TotalArrows, t.PossibleScore, t.NumberOfRanges, r.RangeNumber, r.Distance, r.NumberOfArrows, r.TargetFaceSize FROM RoundType AS t JOIN RoundTypeRange AS r ON t.RoundTypeID = r.RoundTypeID WHERE t.RoundTypeID = 'Long Sydney';". Underneath the query is a toolbar with options: Profiling, Edit inline, Edit, Explain SQL, Create PHP code, Refresh. Below the toolbar are two sets of filters: "Show all" and "Number of rows: 25", and "Filter rows: Search this table". A "Extra options" button is also present. The main content area displays a table with the following data:

RoundTypeID	TotalArrows	PossibleScore	NumberOfRanges	RangeNumber	Distance	NumberOfArrows	TargetFaceSize
Long Sydney	120	1200	4	1	90m	30	122cm
Long Sydney	120	1200	4	2	70m	30	122cm
Long Sydney	120	1200	4	3	60m	30	122cm
Long Sydney	120	1200	4	4	50m	30	122cm

Below the table are two sets of filters: "Show all" and "Number of rows: 25", and "Filter rows: Search this table".

Requirement: Archers should be able to view the club's best score and the archer who achieved it.

```
1 SELECT ai.FirstName, ai.LastName, rt.RoundTypeID, rt.PossibleScore AS MaxScore, rs.RoundScoreID, rs.RoundScore AS TotalScore
2 FROM ArcherCompleteScore AS acs
3 JOIN RoundScore AS rs ON acs.RoundScoreID = rs.RoundScoreID
4 JOIN CompetitionScore AS cs ON acs.ArcherID = cs.ArcherID
5 JOIN ArcherInformation AS ai ON acs.ArcherID = ai.ArcherID
6 JOIN RoundType rt ON cs.RoundTypeID = rt.RoundTypeID
7 WHERE cs.CompetitionID = 2002
8 ORDER BY TotalScore DESC;
```

The screenshot shows a MySQL query results page. At the top, it says "Showing rows 0 - 0 (1 total, Query took 0.0027 seconds.)". Below that is the SQL query: "SELECT ai.FirstName, ai.LastName, rt.RoundTypeID, rt.PossibleScore AS MaxScore, rs.RoundScoreID, rs.RoundScore AS TotalScore FROM ArcherCompleteScore AS acs JOIN RoundScore AS rs ON acs.RoundScoreID = rs.RoundScoreID JOIN CompetitionScore AS cs ON acs.ArcherID = cs.ArcherID JOIN ArcherInformation AS ai ON acs.ArcherID = ai.ArcherID JOIN RoundType rt ON cs.RoundTypeID = rt.RoundTypeID WHERE cs.CompetitionID = 2002 ORDER BY TotalScore DESC;". Underneath the query is a toolbar with options: Profiling, Edit inline, Edit, Explain SQL, Create PHP code, Refresh. Below the toolbar are two sets of filters: "Show all" and "Number of rows: 25", and "Filter rows: Search this table". A "Extra options" button is also present. The main content area displays a table with the following data:

FirstName	LastName	RoundTypeID	MaxScore	RoundScoreID	TotalScore
Jimmy	John	Long Sydney	1200	4000	310

Below the table are two sets of filters: "Show all" and "Number of rows: 25", and "Filter rows: Search this table".

Requirement: Recorders should be able to add new rounds.

```
1 INSERT INTO RoundType (RoundTypeID, TotalArrows, PossibleScore, NumberOfRanges, ValidRound)
2 VALUES ('Long Melbourne', 120, 1200, 4, 'Yes');
```

Edit Copy Delete Long Melbourne 120 1200 4 Yes

```

1 INSERT INTO RoundTypeIDRange (RoundTypeID, RangeNumber, Distance, NumberOfArrows, TargetFaceSize)
2 VALUES ('Long Melbourne', 1, '90m', 30, '122cm');

```

Edit Copy Delete Long Melbourne 1 90m 30 122cm

Requirement: Recorders should be able to link scores to competitions and championships

```

1 UPDATE CompetitionScore AS cs
2 JOIN CompetitionInformation AS ci ON cs.CompetitionID = ci.CompetitionID
3 JOIN ClubChampionshipInfo AS cci ON cs.ChampID = cci.ChampID
4 SET cs.CompetitionID = ci.CompetitionID,
5 cs.ChampID = cci.ChampID
6 WHERE cs.RoundScoreID = 6001;

```

✓ 0 rows affected. (Query took 0.0019 seconds.)
`UPDATE CompetitionScore AS cs JOIN CompetitionInformation AS ci ON cs.CompetitionID = ci.CompetitionID JOIN ClubChampionshipInfo AS cci ON cs.ChampID = cci.ChampID SET cs.CompetitionID = ci.CompetitionID, cs.ChampID = cci.ChampID WHERE cs.RoundScoreID = 6001;`
[\[Edit Inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Requirement: Archers should be able to view equivalent rounds.

```

1 SELECT er.CompetitionID, er.RoundTypeID, er.ArcherCategoryID
2 FROM EquivalentRounds AS er
3 JOIN CompetitionScore AS cs ON er.ArcherCategoryID = cs.ArcherCategoryID
4 WHERE cs.ArcherID = 2000;

```

✓ Showing rows 0 - 0 (1 total, Query took 0.0021 seconds.)
`SELECT er.CompetitionID, er.RoundTypeID, er.ArcherCategoryID FROM EquivalentRounds AS er JOIN CompetitionScore AS cs ON er.ArcherCategoryID = cs.ArcherCategoryID WHERE cs.ArcherID = 2000;`
 Profiling [\[Edit Inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)
 Show all | Number of rows: 25

CompetitionID	RoundTypeID	ArcherCategoryID
2002	Adelaide	Male Under 21 Recurve

SQL Commands by Melvin

Melvin Queries:

1. Archers should be able to search competitions by name, round score, round type, and equipment.

Search competitions by archer name

```
1  SELECT cs.*, ci.CompetitionName, rs.RoundScore
2  FROM CompetitionScore cs
3  JOIN ArcherInformation ai ON cs.ArcherID = ai.ArcherID
4  JOIN CompetitionInformation ci ON cs.CompetitionID = ci.CompetitionID
5  JOIN RoundScoreTable rs ON cs.RoundScoreID = rs.RoundScoreID
6  WHERE ai.FirstName = 'Jimmy'
7  AND ai.LastName = 'John'
```

The screenshot shows a MySQL query results page. The query is:

```
SELECT cs.*, ci.CompetitionName FROM CompetitionScore cs JOIN ArcherInformation ai ON cs.ArcherID = ai.ArcherID JOIN CompetitionInformation ci ON cs.CompetitionID = ci.CompetitionID WHERE ai.FirstName = 'Jimmy' OR ai.LastName = 'John';
```

The results table has the following columns: CompetitionScoreID, CompetitionID, ChampID, ArcherID, ScoreDate, EquipmentUsed, ArcherCategoryID, Time, RoundScoreID, RoundTypeID, RecorderID, CompetitionName. The data for the single row is: 6000, 2002, 2001, 2000, 2022-09-11, Recurve, Male Under 21 Recurve, 11:27:00, 6001, Long Sydney, 2003, Spring Archery Competition.

Search competitions by round score

```
1  SELECT cs.*, ci.CompetitionName, rs.RoundScore
2  FROM CompetitionScore cs
3  JOIN ArcherInformation ai ON cs.ArcherID = ai.ArcherID
4  JOIN CompetitionInformation ci ON cs.CompetitionID = ci.CompetitionID
5  JOIN RoundScoreTable rs ON cs.RoundScoreID = rs.RoundScoreID
6  WHERE rs.RoundScoreID = 6001;
```

The screenshot shows a MySQL query results page. The query is:

```
SELECT CompetitionScore.*, CompetitionInformation.CompetitionName FROM CompetitionScore JOIN CompetitionInformation ON CompetitionScore.CompetitionID = CompetitionInformation.CompetitionID WHERE CompetitionScore.RoundScoreID = 6001;
```

The results table has the same columns as the previous screenshot. The data for the single row is: 6000, 2002, 2001, 2000, 2022-09-11, Recurve, Male Under 21 Recurve, 11:27:00, 6001, Long Sydney, 2003, Spring Archery Competition.

Search competitions by round type

```
1  SELECT cs.*, ci.CompetitionName, rs.RoundScore
2  FROM CompetitionScore cs
3  JOIN ArcherInformation ai ON cs.ArcherID = ai.ArcherID
4  JOIN CompetitionInformation ci ON cs.CompetitionID = ci.CompetitionID
5  JOIN RoundScoreTable rs ON cs.RoundScoreID = rs.RoundScoreID
6  WHERE cs.RoundType = 'Long Sydney';
```

Showing rows 0 - 0 (1 total, Query took 0.0025 seconds.)

```
SELECT CompetitionScore.*, CompetitionInformation.CompetitionName FROM CompetitionScore JOIN CompetitionInformation ON CompetitionScore.CompetitionID = CompetitionInformation.CompetitionID WHERE CompetitionScore.RoundTypeID = 'Long Sydney';
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

CompetitionScoreID	CompetitionID	ChampID	ArcherID	ScoreDate	EquipmentUsed	ArcherCategoryID	Time	RoundScoreID	RoundTypeID	RecorderID	CompetitionName
6000	2002	2001	2000	2022-09-11	Recurve	Male Under 21 Recurve	11:27:00	6001	Long Sydney	2003	Spring Archery Competition

Search competitions by equipment

```

1 SELECT cs.* , ci.CompetitionName , rs.RoundScore
2 FROM CompetitionScore cs
3 JOIN ArcherInformation ai ON cs.ArcherID = ai.ArcherID
4 JOIN CompetitionInformation ci ON cs.CompetitionID = ci.CompetitionID
5 JOIN RoundScoreTable rs ON cs.RoundScoreID = rs.RoundScoreID
6 WHERE cs.Equipment = 'Recurve';

```

Showing rows 0 - 0 (1 total, Query took 0.0025 seconds.)

```
SELECT CompetitionScore.* , CompetitionInformation.CompetitionName FROM CompetitionScore JOIN CompetitionInformation ON CompetitionScore.CompetitionID = CompetitionInformation.CompetitionID WHERE CompetitionScore.EquipmentUsed = 'Recurve';
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

CompetitionScoreID	CompetitionID	ChampID	ArcherID	ScoreDate	EquipmentUsed	ArcherCategoryID	Time	RoundScoreID	RoundTypeID	RecorderID	CompetitionName
6000	2002	2001	2000	2022-09-11	Recurve	Male Under 21 Recurve	11:27:00	6001	Long Sydney	2003	Spring Archery Competition

Search competitions by combinations

```

1 SELECT CompetitionScore.* , ArcherInformation.FirstName , ArcherInformation.LastName , CompetitionInformation.Compe
2 FROM CompetitionScore
3 JOIN ArcherInformation
4 ON CompetitionScore.ArcherID = ArcherInformation.ArcherID
5 JOIN CompetitionInformation
6 ON CompetitionScore.CompetitionID = CompetitionInformation.CompetitionID
7 WHERE (ArcherInformation.FirstName = 'Jimmy'
8 AND ArcherInformation.LastName = 'John')
9 AND CompetitionScore.RoundScoreID = '6001'
10 AND CompetitionScore.RoundTypeID = 'Long Sydney'
11 AND CompetitionScore.EquipmentUsed = 'Recurve';

```

Showing rows 0 - 0 (1 total, Query took 0.0021 seconds.)

```
SELECT CompetitionScore.* , ArcherInformation.FirstName , ArcherInformation.LastName , CompetitionInformation.CompetitionName FROM CompetitionScore JOIN ArcherInformation ON CompetitionScore.ArcherID = ArcherInformation.ArcherID JOIN CompetitionInformation ON CompetitionScore.CompetitionID = CompetitionInformation.CompetitionID WHERE (ArcherInformation.FirstName = 'Jimmy' OR ArcherInformation.LastName = 'John') AND CompetitionScore.RoundScoreID = '6001' AND CompetitionScore.RoundTypeID = 'Long Sydney' AND CompetitionScore.EquipmentUsed = 'Recurve';
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 Filter rows: Search this table

Extra options

CompetitionScoreID	CompetitionID	ChampID	ArcherID	ScoreDate	EquipmentUsed	ArcherCategoryID	Time	RoundScoreID	RoundTypeID	RecorderID	FirstName	LastName	CompetitionName
6000	2002	2001	2000	2022-09-11	Recurve	Male Under 21 Recurve	11:27:00	6001	Long Sydney	2003	Jimmy	John	Spring Archery Competition

2. Archers should be able to view their best scores for each round.

Search the best score for each round for a specific archer.

```

1  SELECT CompetitionScore.*  

2  FROM CompetitionScore  

3  WHERE ArcherID = '2000'  

4  AND RoundTypeID = 'Long Sydney'  

5  ORDER BY CompetitionScoreID DESC LIMIT 10;

```



Showing rows 0 - 0 (1 total). Query took 0.0020 seconds. [CompetitionScoreID: 6000... - 6000...]

SELECT CompetitionScore.* FROM CompetitionScore WHERE ArcherID = '2000' AND RoundTypeID = 'Long Sydney' ORDER BY CompetitionScoreID DESC LIMIT 10;

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Extra options

CompetitionScoreID	CompetitionID	ChampID	ArcherID	ScoreDate	EquipmentUsed	ArcherCategoryID	Time	RoundScoreID	RoundTypeID	RecorderID
6000	2002	2001	2000	2022-09-11	Recurve	Male Under 21 Recurve	11:27:00	6001	Long Sydney	2003

Edit Copy Delete

Select the best score for each round a specific archer using group by

```

1  SELECT cs.CompetitionScoreID,  

2      ci.CompetitionName,  

3      cs.ScoreDate,  

4      cs.EquipmentUsed,  

5      cs.Time,  

6      rs.RoundScore,  

7      rt.TotalArrows,  

8      rt.PossibleScore  

9  FROM CompetitionScore cs  

10 JOIN CompetitionInformation ci ON cs.CompetitionID = ci.CompetitionID  

11 JOIN RoundScore rs ON cs.RoundScoreID = rs.RoundScoreID  

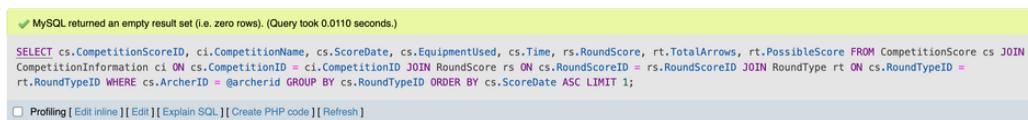
12 JOIN RoundType rt ON cs.RoundTypeID = rt.RoundTypeID  

13 WHERE cs.ArcherID = @archerid  

14 GROUP BY cs.RoundTypeID  

15 ORDER BY cs.ScoreDate ASC;

```



MySQL returned an empty result set (i.e. zero rows). (Query took 0.0110 seconds.)

SELECT cs.CompetitionScoreID, ci.CompetitionName, cs.ScoreDate, cs.EquipmentUsed, cs.Time, rs.RoundScore, rt.TotalArrows, rt.PossibleScore FROM CompetitionScore cs JOIN CompetitionInformation ci ON cs.CompetitionID = ci.CompetitionID JOIN RoundScore rs ON cs.RoundScoreID = rs.RoundScoreID JOIN RoundType rt ON cs.RoundTypeID = rt.RoundTypeID WHERE cs.ArcherID = @archerid GROUP BY cs.RoundTypeID ORDER BY cs.ScoreDate ASC LIMIT 1;

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

3. Recorders should be able to add new competitions

```

1  INSERT INTO CompetitionInformation (CompetitionID, CompetitionName, CompetitionStartDate, CompetitionFinDate)  

2  VALUES ('2001', 'Melvin Archery Competition', '2001-08-01', '2001-11-01');

```



1 row inserted. (Query took 0.0034 seconds.)

INSERT INTO CompetitionInformation (CompetitionID, CompetitionName, CompetitionStartDate, CompetitionFinDate) VALUES ('2001', 'Melvin Archery Competition', '2001-08-01', '2001-11-01');

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code]

CompetitionID	CompetitionName	CompetitionStartDate	CompetitionFinDate
2001	Melvin Archery Competition	2001-08-01	2001-11-01

Edit Copy Delete

4. Recorders should be able to manage default equipment.

```
1 UPDATE CompetitionScoreTBA
2 SET EquipmentUsed =
3     SELECT DefaultEquipment
4     FROM ArcherInformation
5     WHERE ArcherID = '2001'
6 )
7 WHERE ArcherID = '2001'
8     AND EquipmentUsed IS NULL
```

✓ 0 rows affected. (Query took 0.0022 seconds.)
`UPDATE CompetitionScoreTBA SET EquipmentUsed = (SELECT DefaultEquipment FROM ArcherInformation WHERE ArcherID = '2001') WHERE ArcherID = '2001' AND EquipmentUsed IS NULL;`
[Edit inline] [Edit] [Create PHP code]

SQL command by Anh Khoa Nguyen

Archers should be able to lookup scores linked to date and round type:

```
1 SELECT
2     ai.ArcherID AS 'Archer ID',
3     acs.ScoreDate AS 'Date',
4     cs.CompetitionID AS 'Competition ID',
5     rs.RoundScore AS 'Score',
6     rd.RoundTypeID AS 'Round Type'
7 FROM
8     ArcherCompleteScore AS acs
9 JOIN
10    RoundScore AS rs ON acs.RoundScoreID = rs.RoundScoreID
11 JOIN
12    ArcherInformation AS ai ON acs.ArcherID = ai.ArcherID
13 JOIN
14    CompetitionScore AS cs ON ai.ArcherID = cs.ArcherID
15 JOIN
16    RoundType AS rd ON cs.RoundTypeID = rd.RoundTypeID
17 WHERE
18     ai.ArcherID = '4026' AND
19     acs.ScoreDate = '2024-08-20' AND
20     rd.RoundTypeID = 'CCE4'
21 ORDER BY
22     acs.ScoreDate;
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0021 seconds.)
SELECT ai.ArcherID AS 'Archer ID', acs.ScoreDate AS 'Date', cs.CompetitionID AS 'Competition ID', rs.RoundScore AS 'Score', rd.RoundTypeID AS 'Round Type' FROM ArcherCompleteScore AS acs JOIN RoundScore AS rs ON acs.RoundScoreID = rs.RoundScoreID JOIN ArcherInformation AS ai ON acs.ArcherID = ai.ArcherID JOIN CompetitionScore AS cs ON ai.ArcherID = cs.ArcherID JOIN RoundType AS rd ON cs.RoundTypeID = rd.RoundTypeID WHERE ai.ArcherID = '4026' AND acs.ScoreDate = '2024-08-20' AND rd.RoundTypeID = 'CCE4' ORDER BY acs.ScoreDate;
 Profiling | Edit inline | Edit | Explain SQL | Create PHP code | Refresh |

Archers should be able to filter competitions related to club championships:

```
1 SELECT
2     cs.CompetitionScoreID AS 'Competition Score ID',
3     cs.ChampID AS 'Championship ID',
4     cci.ChampionshipName AS 'Championship Name',
5     cs.ArcherID AS 'Archer ID',
6     cs.ScoreDate AS 'Score Date',
7     cs.EquipmentUsed AS 'Equipment Used',
8     cs.ArcherCategoryID AS 'Archer Category ID',
9     cs.Time AS 'Time',
10    cs.RoundScoreID AS 'Round Score ID',
11    cs.RoundTypeID AS 'Round Type ID',
12    cs.RecorderID AS 'Recorder ID',
13    ci.CompetitionName AS 'Competition Name',
14    cci.ChampionshipYear AS 'Championship Year'
15 FROM
16    CompetitionScore AS cs
17 JOIN
18    CompetitionInformation AS ci ON cs.CompetitionID = ci.CompetitionID
19 JOIN
20    ClubChampionshipInfo AS cci ON cs.ChampID = cci.ChampID
21 WHERE
```

```
22 cs.ChampID = '9464';
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0021 seconds.)

```
SELECT cs.CompetitionScoreID AS 'Competition Score ID', cs.ChampID AS 'Championship ID', cci.ChampionshipName AS 'Championship Name', cs.ArcherID AS 'Archer ID', cs.ScoreDate AS 'Score Date', cs.EquipmentUsed AS 'Equipment Used', cs.ArcherCategoryID AS 'Archer Category ID', cs.Time AS 'Time', cs.RoundScoreID AS 'Round Score ID', cs.RoundTypeID AS 'Round Type ID', cs.RecorderID AS 'Recorder ID', ci.CompetitionName AS 'Competition Name', cci.ChampionshipYear AS 'Championship Year' FROM CompetitionScore AS cs JOIN CompetitionInformation AS ci ON cs.CompetitionID = ci.CompetitionID JOIN ClubChampionshipInfo AS cci ON cs.ChampID = cci.ChampID WHERE cs.ChampID = '9464';
```

Profiling [Edit] [Explain SQL] [Create PHP code] [Refresh]

Competition Score ID | Championship ID | Championship Name | Archer ID | Score Date | Equipment Used | Archer Category ID | Time | Round Score ID | Round Type ID | Recorder ID | Competition Name | Championship Year

Recorders should be able to add new archers:

```
1 INSERT INTO ArcherInformation (FirstName, LastName, DOB, Gender, JoinDate, DefaultEquipment)
2 VALUES ('Anh Khoa', 'Nguyen/', '2004-05-01', 'Male', '2024-05-08', 'Recurve');
```

1 row inserted.
Inserted row id: 1001 (Query took 0.0021 seconds.)

```
INSERT INTO ArcherInformation (FirstName, LastName, DOB, Gender, JoinDate, DefaultEquipment) VALUES ('Anh Khoa', 'Nguyen/', '2004-05-01', 'Male', '2024-05-08', 'Recurve');
```

[Edit inline] [Edit] [Create PHP code]

Recorders should be able to manage time-dependent rounds:

```
1
2 UPDATE RoundType
3 SET ValidRound = 'Yes'
4 WHERE RoundTypeID = '1';
5
6
```

1 row affected. (Query took 0.0020 seconds.)

```
UPDATE RoundType SET ValidRound = 'Yes' WHERE RoundTypeID = '1';
```

[Edit inline] [Edit] [Create PHP code]

SQL Commands: Joshua

Requirement: "Archers should be able to lookup scores linked to date and round type."

```
1 START TRANSACTION;
2
3 INSERT INTO RoundScore (RoundScoreID, RoundScore, NumberOfRanges) VALUES (NULL, '0', '1');
4 SET @lastRoundScoreID = LAST_INSERT_ID();
5
6 INSERT INTO RangeScore (RangeScoreID, RangeNumber, NumberOfEnds, RoundScoreID, RangeScore) VALUES (NULL, '1'
7 SET @lastRangeScoreID = LAST_INSERT_ID();
8
9     INSERT INTO EndScore (EndNumberID, EndNumber, Arrow1Score, Arrow2Score, Arrow3Score, Arrow4Score, Arrow5
10    VALUES (NULL, '1', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
11
12    INSERT INTO EndScore (EndNumberID, EndNumber, Arrow1Score, Arrow2Score, Arrow3Score, Arrow4Score, Arrow5
13    VALUES (NULL, '2', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
14
15    INSERT INTO EndScore (EndNumberID, EndNumber, Arrow1Score, Arrow2Score, Arrow3Score, Arrow4Score, Arrow5
16    VALUES (NULL, '3', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
17
18    INSERT INTO EndScore (EndNumberID, EndNumber, Arrow1Score, Arrow2Score, Arrow3Score, Arrow4Score, Arrow5
19    VALUES (NULL, '4', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
20
21    INSERT INTO EndScore (EndNumberID, EndNumber, Arrow1Score, Arrow2Score, Arrow3Score, Arrow4Score, Arrow5
22    VALUES (NULL, '5', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
23
24    INSERT INTO EndScore (EndNumberID, EndNumber, Arrow1Score, Arrow2Score, Arrow3Score, Arrow4Score, Arrow5
25    VALUES (NULL, '6', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
26
27 UPDATE RangeScore
28 SET RangeScore = (SELECT SUM(EndTotalScore) FROM EndScore WHERE RangeScoreID = @lastRangeScoreID)
29 WHERE RangeScoreID = @lastRangeScoreID;
30
31 UPDATE RoundScore
32 SET RoundScore = (SELECT SUM(RangeScore) FROM RangeScore WHERE RoundScoreID = @lastRoundScoreID)
33 WHERE RoundScoreID = @lastRoundScoreID;
34
35 SET @equipment = (SELECT DefaultEquipment FROM ArcherInformation WHERE ArcherID = '2000');
36
37 INSERT INTO ArcherCompleteScore (EntryID, ArcherID, ScoreDate, EquipmentUsed, RoundScoreID)
38 VALUES (NULL, '2000', '2022.09.10', @equipment, @lastRoundScoreID);
39
40 COMMIT;
```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0019 seconds.)

```
START TRANSACTION;
```

[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted.

Inserted row id: 6002 (Query took 0.0022 seconds.)

```
INSERT INTO RoundScore VALUES (NULL, '0', '1');
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0019 seconds.)

```
SET @lastRoundScoreID = LAST_INSERT_ID();
```

[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted.

Inserted row id: 60015 (Query took 0.0104 seconds.)

```
INSERT INTO RangeScore VALUES (NULL, '1', '6', '0', @lastRoundScoreID);
```

[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0019 seconds.)

```
SET @lastRangeScoreID = LAST_INSERT_ID();
```

[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted.

Inserted row id: 600146 (Query took 0.0028 seconds.)

```
INSERT INTO EndScore VALUES (NULL, '1', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
```

[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted.

Inserted row id: 600147 (Query took 0.0018 seconds.)

```
INSERT INTO EndScore VALUES (NULL, '2', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
```

[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted.

Inserted row id: 600148 (Query took 0.0021 seconds.)

```
INSERT INTO EndScore VALUES (NULL, '3', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
```

[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted.

Inserted row id: 600149 (Query took 0.0022 seconds.)

```
INSERT INTO EndScore VALUES (NULL, '4', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
```

[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted. Inserted row id: 600149 (Query took 0.0022 seconds.)
<code>INSERT INTO EndScore VALUES (NULL, '4', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);</code>
[Edit inline] [Edit] [Create PHP code]
✓ 1 row inserted. Inserted row id: 600150 (Query took 0.0018 seconds.)
<code>INSERT INTO EndScore VALUES (NULL, '5', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);</code>
[Edit inline] [Edit] [Create PHP code]
✓ 1 row inserted. Inserted row id: 600151 (Query took 0.0019 seconds.)
<code>INSERT INTO EndScore VALUES (NULL, '6', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);</code>
[Edit inline] [Edit] [Create PHP code]
✓ 1 row affected. (Query took 0.0020 seconds.)
<code>UPDATE RangeScore SET RangeScore = (SELECT SUM(EndTotalScore) FROM EndScore WHERE RangeScoreID = @lastRangeScoreID) WHERE RangeScoreID = @lastRangeScoreID;</code>
[Edit inline] [Edit] [Create PHP code]
✓ 1 row affected. (Query took 0.0024 seconds.)
<code>UPDATE RoundScore SET RoundScore = (SELECT SUM(RangeScore) FROM RangeScore WHERE RoundScoreID = @lastRoundScoreID) WHERE RoundScoreID = @lastRoundScoreID;</code>
[Edit inline] [Edit] [Create PHP code]
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0024 seconds.)
<code>SET @equipment = (SELECT DefaultEquipment FROM ArcherInformation WHERE ArcherID = '2000');</code>
[Edit inline] [Edit] [Create PHP code]
✓ 1 row inserted. Inserted row id: 3001 (Query took 0.0031 seconds.)
<code>INSERT INTO ArcherCompleteScore VALUES (NULL, '2000', '2022.09.10', @equipment, @lastRoundScoreID);</code>
[Edit inline] [Edit] [Create PHP code]
✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0045 seconds.)
<code>COMMIT;</code>
[Edit inline] [Edit] [Create PHP code]

Requirement: "Recorders should be able to record scores arrow by arrow, grouped by ends."

```

1  START TRANSACTION;
2
3  INSERT INTO RoundScore VALUES (NULL, '0', '1');
4  SET @lastRoundScoreID = LAST_INSERT_ID();
5
6  INSERT INTO RangeScore VALUES (NULL, '1', '6', '0', @lastRoundScoreID);
7  SET @lastRangeScoreID = LAST_INSERT_ID();
8
9    INSERT INTO EndScore VALUES (NULL, '1', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
10
11   INSERT INTO EndScore VALUES (NULL, '2', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
12
13   INSERT INTO EndScore VALUES (NULL, '3', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
14
15   INSERT INTO EndScore VALUES (NULL, '4', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
16
17   INSERT INTO EndScore VALUES (NULL, '5', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
18
19   INSERT INTO EndScore VALUES (NULL, '6', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
20
21 UPDATE RangeScore
22 SET RangeScore = (SELECT SUM(EndTotalScore)
23   FROM EndScore
24   WHERE RangeScoreID = @lastRangeScoreID)
25 WHERE RangeScoreID = @lastRangeScoreID;
26
27 INSERT INTO RangeScore VALUES (NULL, '2', '6', '0', @lastRoundScoreID);
28 SET @lastRangeScoreID = LAST_INSERT_ID();
29
30   INSERT INTO EndScore VALUES (NULL, '1', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);

```

```

32      INSERT INTO EndScore VALUES (NULL, '2', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
33
34      INSERT INTO EndScore VALUES (NULL, '3', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
35
36      INSERT INTO EndScore VALUES (NULL, '4', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
37
38      INSERT INTO EndScore VALUES (NULL, '5', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
39
40      INSERT INTO EndScore VALUES (NULL, '6', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
41 UPDATE RangeScore
42 SET RangeScore = (SELECT SUM(EndTotalScore)
43   FROM EndScore
44   WHERE RangeScoreID = @lastRangeScoreID)
45 WHERE RangeScoreID = @lastRangeScoreID;
46
47      INSERT INTO RangeScore VALUES (NULL, '3', '6', '0', @lastRoundScoreID);
48      SET @lastRangeScoreID = LAST_INSERT_ID();
49
50      INSERT INTO EndScore VALUES (NULL, '1', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
51
52      INSERT INTO EndScore VALUES (NULL, '2', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
53
54      INSERT INTO EndScore VALUES (NULL, '3', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
55
56      INSERT INTO EndScore VALUES (NULL, '4', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
57
58      INSERT INTO EndScore VALUES (NULL, '5', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
59
60      INSERT INTO EndScore VALUES (NULL, '6', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
61 UPDATE RangeScore
62 SET RangeScore = (SELECT SUM(EndTotalScore)
63   FROM EndScore
64   WHERE RangeScoreID = @lastRangeScoreID)
65 WHERE RangeScoreID = @lastRangeScoreID;
66
67      INSERT INTO RangeScore VALUES (NULL, '4', '6', '0', @lastRoundScoreID);
68      SET @lastRangeScoreID = LAST_INSERT_ID();
69
70      INSERT INTO EndScore VALUES (NULL, '1', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
71
72      INSERT INTO EndScore VALUES (NULL, '2', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
73
74      INSERT INTO EndScore VALUES (NULL, '3', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
75
76      INSERT INTO EndScore VALUES (NULL, '4', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
77
78      INSERT INTO EndScore VALUES (NULL, '5', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
79
80      INSERT INTO EndScore VALUES (NULL, '6', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
81
82 UPDATE RangeScore
83 SET RangeScore = (SELECT SUM(EndTotalScore)
84   FROM EndScore
85   WHERE RangeScoreID = @lastRangeScoreID)
86 WHERE RangeScoreID = @lastRangeScoreID;
87
88 UPDATE RoundScore
89 SET RoundScore = (SELECT SUM(RangeScore)

```

```
90     FROM RangeScore
91     WHERE RoundScoreID = @lastRoundScoreID)
92 WHERE RoundScoreID = @lastRoundScoreID;
93
94 INSERT INTO CompetitionScore VALUES (NULL, '2002', '2001', '20000', '2022.09.10', 'recurve', 'Male Under 21 Recu
95
96 COMMIT;
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0052 seconds.)

```
START TRANSACTION;
```

[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted.

Inserted row id: 6011 (Query took 0.0022 seconds.)

```
INSERT INTO RoundScore VALUES (NULL, '0', '1');
```

[Edit inline] [Edit] [Create PHP code]

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0021 seconds.)

```
SET @lastRoundScoreID = LAST_INSERT_ID();
```

[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted.

Inserted row id: 60045 (Query took 0.0031 seconds.)

```
INSERT INTO RangeScore VALUES (NULL, '1', '6', '0', @lastRoundScoreID);
```

[Edit inline] [Edit] [Create PHP code]

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0022 seconds.)

```
SET @lastRangeScoreID = LAST_INSERT_ID();
```

[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted.

Inserted row id: 600320 (Query took 0.0024 seconds.)

```
INSERT INTO EndScore VALUES (NULL, '1', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
```

[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted.

Inserted row id: 600321 (Query took 0.0032 seconds.)

```
INSERT INTO EndScore VALUES (NULL, '2', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
```

[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted.

Inserted row id: 600322 (Query took 0.0022 seconds.)

```
INSERT INTO EndScore VALUES (NULL, '3', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
```

[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted.

Inserted row id: 600323 (Query took 0.0022 seconds.)

```
INSERT INTO EndScore VALUES (NULL, '4', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
```

[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted.

Inserted row id: 600324 (Query took 0.0026 seconds.)

```
INSERT INTO EndScore VALUES (NULL, '5', '8', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
```

[Edit inline] [Edit] [Create PHP code]

```

✓ 1 row inserted.
Inserted row id: 600340 (Query took 0.0029 seconds)

INSERT INTO EndScore VALUES (NULL, '3', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted.
Inserted row id: 600341 (Query took 0.0024 seconds)

INSERT INTO EndScore VALUES (NULL, '4', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted.
Inserted row id: 600342 (Query took 0.0020 seconds)

INSERT INTO EndScore VALUES (NULL, '5', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted.
Inserted row id: 600343 (Query took 0.0021 seconds)

INSERT INTO EndScore VALUES (NULL, '6', '8', '8', '8', '8', '10', '50', @lastRangeScoreID);
[Edit inline] [Edit] [Create PHP code]

✓ 1 row affected. (Query took 0.0022 seconds)

UPDATE RangeScore SET RangeScore = (SELECT SUM(EndTotalScore) FROM EndScore WHERE RangeScoreID = @lastRangeScoreID) WHERE RangeScoreID = @lastRangeScoreID;
[Edit inline] [Edit] [Create PHP code]

✓ 1 row affected. (Query took 0.0020 seconds)

UPDATE RoundScore SET RoundScore = (SELECT SUM(RangeScore) FROM RangeScore WHERE RoundScoreID = @lastRoundScoreID) WHERE RoundScoreID = @lastRoundScoreID;
[Edit inline] [Edit] [Create PHP code]

✓ 1 row inserted.
Inserted row id: 6002 (Query took 0.0022 seconds)

INSERT INTO CompetitionScore VALUES (NULL, '2002', '2001', '20000', '2022.09.10', 'recurve', 'Male Under 21 Recurve', '11:27', @lastRoundScoreID, 'Long Sydney', '2003');
[Edit inline] [Edit] [Create PHP code]

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0076 seconds)

COMMIT;
[Edit inline] [Edit] [Create PHP code]

```

Requirement: "Recorders should be able to approve archer scores"

If the recorder needed to edit an entry before approving it, they would use the code below

```

1 UPDATE CompetitionScoreTBA
2 Set column = value,
3 WHERE CompetitionScoreTBAID = '2000';

```

To approve the score it would be the code below

```

1 INSERT INTO CompetitionScore (CompetitionScoreID, CompetitionID, ChampID, ArcherID, ScoreDate, EquipmentUsed, Arc
2         SELECT CompetitionScoreTBAID, CompetitionID, ChampID, ArcherID, ScoreDate, EquipmentUsed, ArcherCateg
3             FROM CompetitionScoreTBA
4             WHERE CompetitionScoreTBAID = '2000';

```

```

✓ 0 rows inserted. (Query took 0.0020 seconds)

INSERT INTO CompetitionScore (CompetitionScoreID, CompetitionID, ChampID, ArcherID, ScoreDate, EquipmentUsed, ArcherCategoryID, Time, RoundScoreID, RoundTypeID, RecorderID) SELECT CompetitionScoreTBAID, CompetitionID, ChampID, ArcherID, ScoreDate, EquipmentUsed, ArcherCategoryID, 'Time', RoundScoreID, RoundTypeID, RecorderID FROM CompetitionScoreTBA WHERE CompetitionScoreTBAID = '2000';
[Edit inline] [Edit] [Create PHP code]

```

Major Specific Work

Melvin's Major Specific Technology (Data Science)

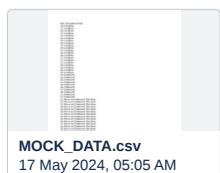
Step 1: Collect the dataset from the database

SQL Query

```
WITH ArcherWithAge AS (
    SELECT
        TIMESTAMPDIFF(YEAR, ai.DOB, CURDATE()) AS Age,
        acs.EquipmentUsed,
        ROW_NUMBER() OVER (PARTITION BY
            CASE
                WHEN TIMESTAMPDIFF(YEAR, ai.DOB, CURDATE()) <= 18 THEN 'Section 1: <= 18'
                WHEN TIMESTAMPDIFF(YEAR, ai.DOB, CURDATE()) <= 21 THEN 'Section 2: 19 to 21'
                WHEN TIMESTAMPDIFF(YEAR, ai.DOB, CURDATE()) <= 49 THEN 'Section 3: 22 to 49'
                WHEN TIMESTAMPDIFF(YEAR, ai.DOB, CURDATE()) >= 50 THEN 'Section 4: 50 and above'
                ELSE 'Section 5: 60 and above'
            END
        ORDER BY ai.DOB) AS RowNum
    FROM
        ArcherInformation ai
    JOIN
        ArcherCompleteScore acs ON ai.ArcherID = acs.ArcherID
)
SELECT
    Age,
    EquipmentUsed
FROM
    ArcherWithAge
WHERE
    RowNum <= 100
ORDER BY
    Age;
```

Step 2: Extract the collected dataset into a CSV file

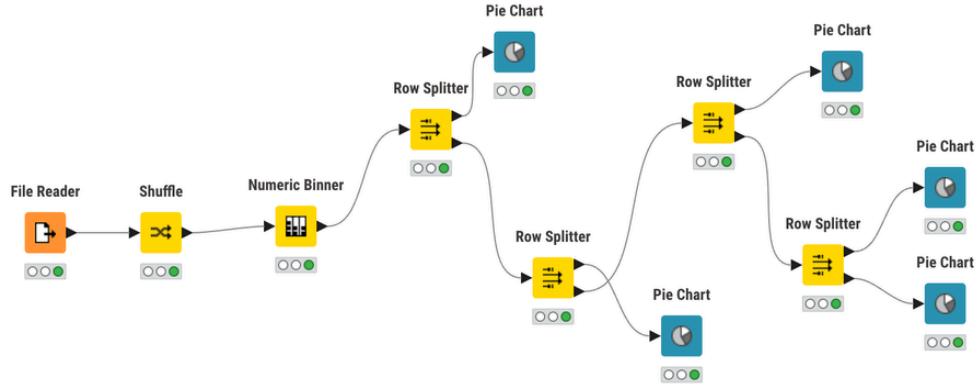
Export the dataset from MariaDB into a CSV file and save it locally.



Step 3: Load the dataset into Knime software and manipulate the data

Here is the process of data manipulation.





Step 4: After numeric binner by age (Under 18 and 18; Under 21 and 21; Open; 50+; 60,70+), split the data based on age_binned.

Numeric Binner Node outcome:

#	RowID	age	equipment	age_binned
1	Row... 48		Compound	Open
2	Row... 69		Compound	60,70+
3	Row... 61		Recurve	60,70+
4	Row... 57		Recurve/Compound Barebow	50+
5	Row... 53		Recurve/Compound Barebow	50+
6	Row... 60		Recurve	60,70+
7	Row... 58		Longbow	50+
8	Row... 27		Compound	Open
9	Row... 50		Recurve/Compound Barebow	50+
10	Row53 16		Recurve/Compound Barebow	Under 18 and 18
11	Row... 51		Recurve/Compound Barebow	50+
12	Row... 44		Longbow	Open
13	Row... 19		Longbow	Under 21 and 21
14	Row... 65		Compound	60,70+
15	Row... 63		Recurve	60,70+
16	Row... 29		Compound	Open
17	Row... 20		Compound	Under 21 and 21
18	Row... 50		Recurve	50+
19	Row... 19		Recurve/Compound Barebow	Under 21 and 21
20	Row... 36		Recurve	Open
21	Row... 25		Longbow	Open

1st Row Splitter Node:

#	RowID	age	equipment	age_binned
1	Row53 16		Recurve/Compound Barebow	Under 18 and 18
2	Row92 17		Recurve	Under 18 and 18
3	Row28 17		Compound	Under 18 and 18
4	Row85 17		Recurve	Under 18 and 18
5	Row73 18		Recurve	Under 18 and 18
6	Row46 17		Recurve/Compound Barebow	Under 18 and 18
7	Row83 16		Recurve	Under 18 and 18
8	Row68 17		Recurve	Under 18 and 18
9	Row17 16		Longbow	Under 18 and 18
10	Row40 17		Recurve/Compound Barebow	Under 18 and 18
11	Row0 18		Longbow	Under 18 and 18
12	Row8 17		Longbow	Under 18 and 18
13	Row63 18		Recurve	Under 18 and 18
14	Row4 17		Longbow	Under 18 and 18
15	Row3 17		Longbow	Under 18 and 18
16	Row11 16		Longbow	Under 18 and 18
17	Row90 16		Recurve	Under 18 and 18
18	Row23 17		Compound	Under 18 and 18
19	Row9 18		Longbow	Under 18 and 18
20	Row39 16		Recurve/Compound Barebow	Under 18 and 18
21	Row78 18		Recurve	Under 18 and 18

2nd Row Splitter Node:

Rows: 100 | Columns: 3

#	RowID	age Number (Integer)	equipment String	age_binned String
1	Row...	19	Longbow	Under 21 and 21
2	Row...	20	Compound	Under 21 and 21
3	Row...	19	Recurve/Compound Barebow	Under 21 and 21
4	Row...	19	Compound	Under 21 and 21
5	Row...	21	Recurve	Under 21 and 21
6	Row...	21	Compound	Under 21 and 21
7	Row...	21	Recurve/Compound Barebow	Under 21 and 21
8	Row...	21	Longbow	Under 21 and 21
9	Row...	19	Recurve/Compound Barebow	Under 21 and 21
10	Row...	20	Longbow	Under 21 and 21
11	Row...	20	Recurve/Compound Barebow	Under 21 and 21
12	Row...	21	Recurve	Under 21 and 21
13	Row...	19	Longbow	Under 21 and 21
14	Row...	20	Recurve	Under 21 and 21
15	Row...	19	Longbow	Under 21 and 21
16	Row...	20	Compound	Under 21 and 21
17	Row...	20	Recurve	Under 21 and 21
18	Row...	20	Recurve	Under 21 and 21
19	Row...	20	Longbow	Under 21 and 21
20	Row...	21	Compound	Under 21 and 21
21	Row...	20	Longbow	Under 21 and 21

3rd Row Splitter Node:

Rows: 100 | Columns: 3

#	RowID	age Number (Integer)	equipment String	age_binned String
1	Row...	48	Compound	Open
2	Row...	27	Compound	Open
3	Row...	44	Longbow	Open
4	Row...	29	Compound	Open
5	Row...	36	Recurve	Open
6	Row...	25	Longbow	Open
7	Row...	31	Recurve/Compound Barebow	Open
8	Row...	44	Recurve	Open
9	Row...	27	Recurve/Compound Barebow	Open
10	Row...	30	Recurve	Open
11	Row...	33	Recurve/Compound Barebow	Open
12	Row...	27	Longbow	Open
13	Row...	24	Longbow	Open
14	Row...	45	Recurve	Open
15	Row...	49	Recurve/Compound Barebow	Open
16	Row...	24	Longbow	Open
17	Row...	37	Compound	Open
18	Row...	38	Recurve/Compound Barebow	Open
19	Row...	43	Longbow	Open
20	Row...	40	Recurve/Compound Barebow	Open
21	Row...	23	Compound	Open

4th Row Spiller Node:

► 1: Filtered ► 2: Filtered Out ⚡ Flow Variables

Rows: 99 | Columns: 3

Table Statistics

#	RowID	age Number (integer)	equipment String	age_binned String
1	Row...	57	Recurve/Compound Barebow	50+
2	Row...	53	Recurve/Compound Barebow	50+
3	Row...	58	Longbow	50+
4	Row...	50	Recurve/Compound Barebow	50+
5	Row...	51	Recurve/Compound Barebow	50+
6	Row...	50	Recurve	50+
7	Row...	57	Recurve	50+
8	Row...	55	Recurve	50+
9	Row...	53	Recurve/Compound Barebow	50+
10	Row...	50	Compound	50+
11	Row...	51	Longbow	50+
12	Row...	51	Recurve	50+
13	Row...	50	Recurve/Compound Barebow	50+
14	Row...	53	Longbow	50+
15	Row...	52	Recurve	50+
16	Row...	54	Compound	50+
17	Row...	52	Recurve	50+
18	Row...	50	Recurve	50+
19	Row...	57	Recurve/Compound Barebow	50+
20	Row...	58	Recurve	50+
21	Row...	53	Longbow	50+

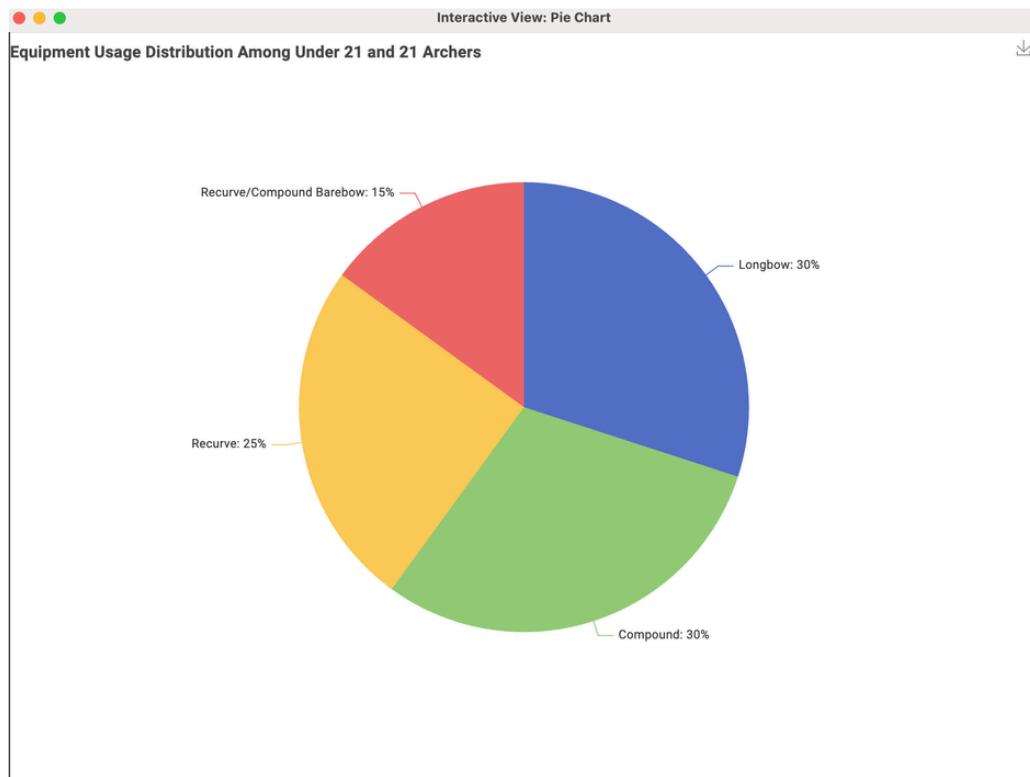
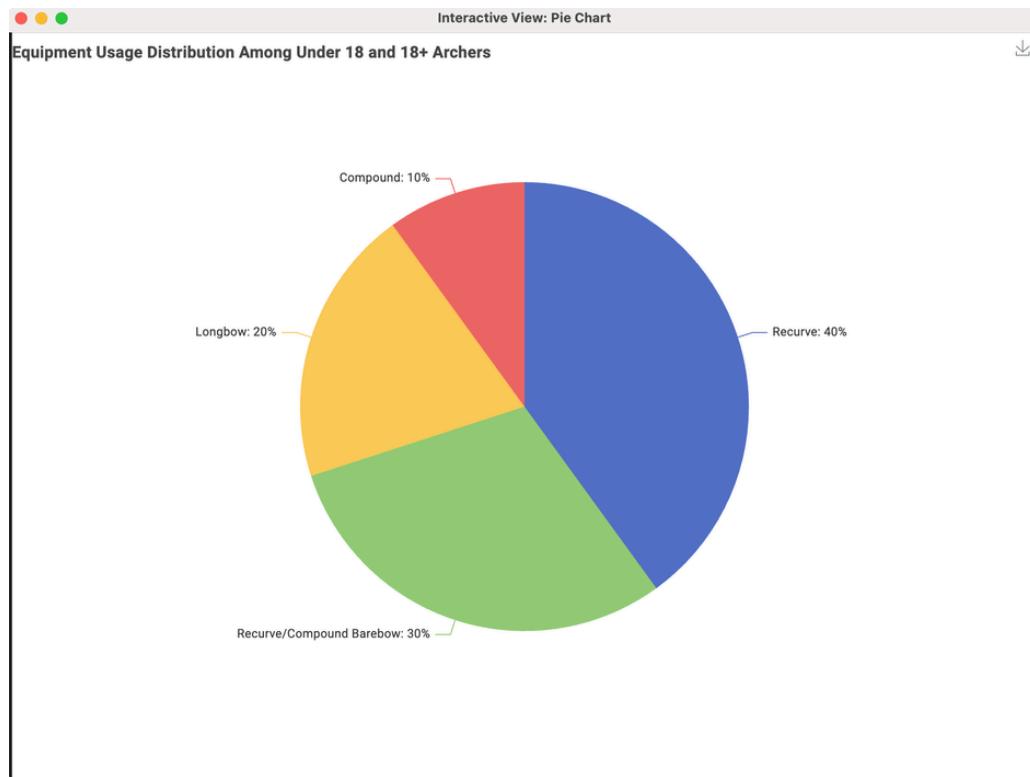
► 1: Filtered ► 2: Filtered Out ⚡ Flow Variables

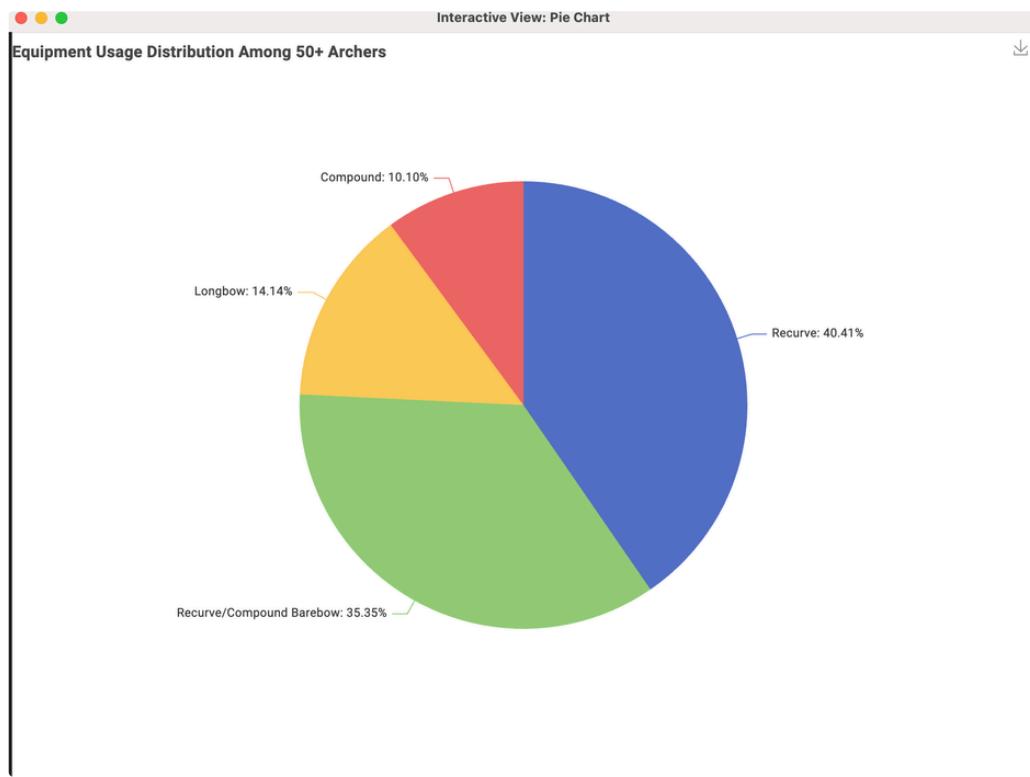
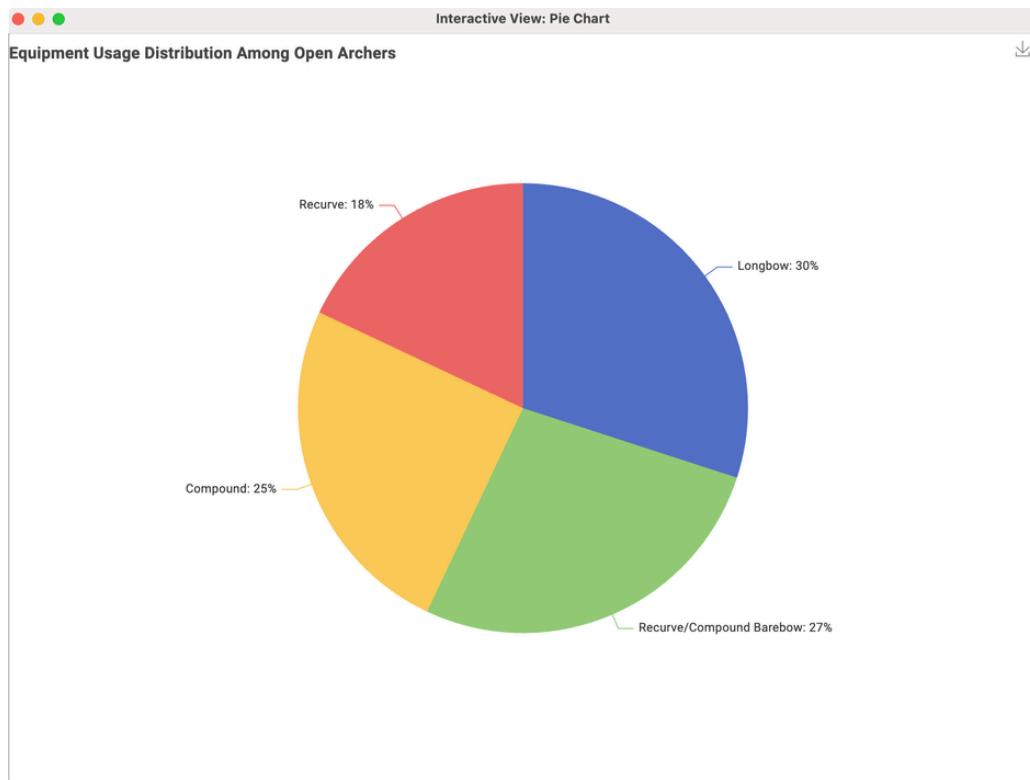
Rows: 101 | Columns: 3

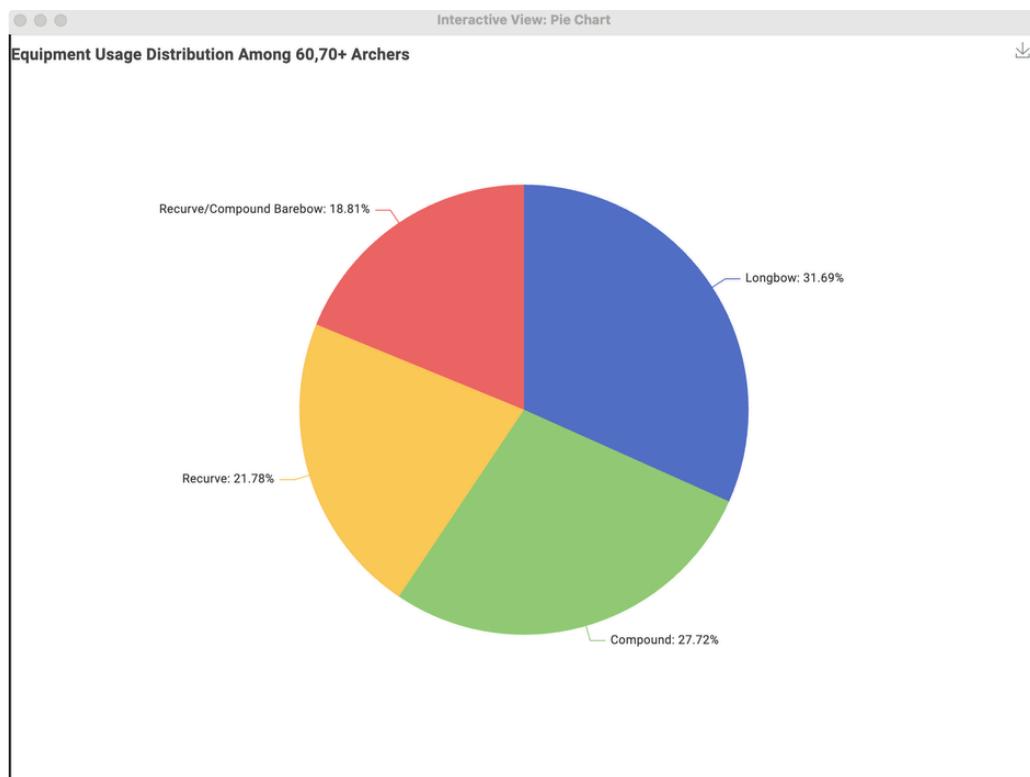
Table Statistics

#	RowID	age Number (integer)	equipment String	age_binned String
1	Row...	69	Compound	60,70+
2	Row...	61	Recurve	60,70+
3	Row...	60	Recurve	60,70+
4	Row...	65	Compound	60,70+
5	Row...	63	Recurve	60,70+
6	Row...	70	Compound	60,70+
7	Row...	68	Longbow	60,70+
8	Row...	65	Recurve	60,70+
9	Row...	70	Recurve	60,70+
10	Row...	64	Compound	60,70+
11	Row...	66	Longbow	60,70+
12	Row...	66	Recurve	60,70+
13	Row...	64	Recurve/Compound Barebow	60,70+
14	Row...	63	Recurve/Compound Barebow	60,70+
15	Row...	67	Longbow	60,70+
16	Row...	63	Recurve	60,70+
17	Row...	69	Compound	60,70+
18	Row...	64	Recurve	60,70+
19	Row...	64	Recurve/Compound Barebow	60,70+
20	Row...	68	Recurve/Compound Barebow	60,70+
21	Row...	69	Recurve/Compound Barebow	60,70+

Step 5: Using Pie Chart Node to visualize the data







Sukeyna's Major Specific Work (Data Science)

Analysis Goal: Equipment Score Performance Comparison

1. Generate mockup data using Mockaroo

(The data generated using Mockaroo is artificial and differs significantly from real-life data as it lacks authenticity.)

- Columns (Equipment, Scores)

2. Upload Mockup Data (in CSV form) into Kinetica Workbench



3. Apply SQL command to find the average score of each type of equipment:

```
1 SELECT Equipment, AVG(Scores) AS Average_Score
2 FROM archery_perfromance
3 GROUP BY Equipment
4 ORDER BY Average_Score DESC;
```

A screenshot of the Kinetica Workbench interface. At the top, there is a SQL editor window with the following code:

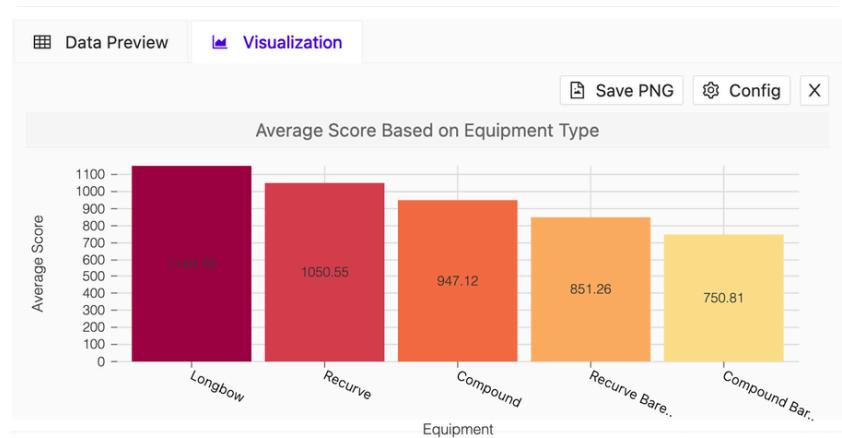
```
1 SELECT Equipment, AVG(Scores) AS Average_Score
2 FROM archery_perfromance
3 GROUP BY Equipment
4 ORDER BY Average_Score DESC;
```

Below the editor, a message box displays: "Number of Records: 5, Completed in 0.02361 seconds". Underneath, there are two tabs: "Data" and "Visualization". The "Data" tab is selected, showing a table with the following data:

Equipment	Average_Score
Longbow	1149.79
Recurve	1050.55
Compound	947.12
Recurve Barebow	851.26
Compound Barebow	750.81

At the bottom of the interface, there are buttons for "Copy", "Export CSV", and "Clear".

4. Visualise finding as a Bar Chart:

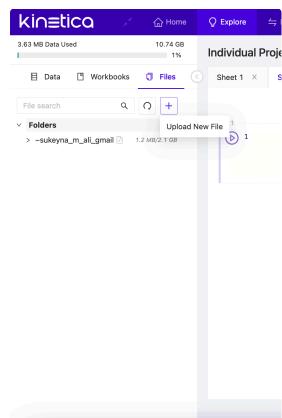


Procedure that would be compatible with the MariaDB database

1. Export relevant tables into CVS form:

- Archer Complete Score
- Round Score

2. Import Tables into Kinetica Workbench:



3. Implement SQL Command for Analysis:

```

1  SELECT c.EquipmentUsed, AVG(r.RoundScore) AS AverageScore
2  FROM ArcherCompleteScore as c JOIN RoundScore as r ON c.RoundScoreID = r.RoundScoreID
3  GROUP BY c.EquipmentUsed
4  ORDER BY AverageScore DESC;

```

```

1  SELECT c.EquipmentUsed, AVG(r.RoundScore) AS AverageScore
2  FROM ArcherCompleteScore as c JOIN RoundScore as r ON c.RoundScoreID = r.RoundSc
3  GROUP BY c.EquipmentUsed
4  ORDER BY AverageScore DESC;

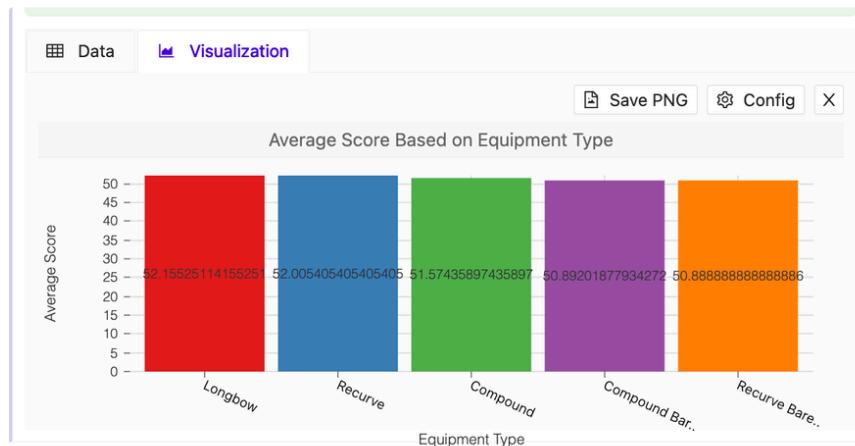
```

Number of Records: 5, Completed in 0.04976 seconds

EquipmentUsed	AverageScore
Longbow	52.15525114155251
Recurve	52.005405405405405
Compound	51.57435897435897
Compound Barebow	50.89201877934272
Recurve Barebow	50.888888888888886

4. Visualise Findings using a Bar Chart:

(These findings are heavily flawed because the dummy data used is excessively artificial for proper analysis.)



Anh Khoa Nguyen's Major Specific Work

Use case: The entry of a single score

The round is chosen from a list and the archer enter their full name and choose from the default equipment list provided in the UI.

Choose round

Round

Round 2 - 50m

Round 1 - 50m

Round 2 - 50m

Round 3 - 90m

Round 4 - 60m

Round 5 - 70m

Firstname:

Anh Khoa

Lastname:

Nguyen

Equipment

Recurve

Target Distance (m):

122

Go to Score Entry

Recurve

Recurve

Longbow

Compound

Recurve barebow

Compound barebow

After the setup, archers are prompted to type in the correct score

Enter Scores

Enter your scores:

	9
	8
	7
	6
	5
	1

And calculate the total score and submit the scores:

Total Score:

36

Submit Scores

For the implementation of the submit scores form, I used four different languages and tools to operate the program which are HTML, CSS, PHP and JavaScript. These tools help UI look more professional and fancier but also maintain the user-friendly process. It has two primary sections: one for selecting the round and inputting archer information, and one for entering and calculating scores. The "apply_content" div has a form where users can select a round, input their first and last names, choose their equipment, and set the desired distance. After pressing the "Go to Score Entry" button, the "apply_content" div disappears and the "score-entry" div emerges. This new part allows users to enter their scores, which are checked to ensure they are acceptable values (such as 'X', 'M', or integers 1-10). The scores are then utilised to generate and display the overall score in real time. The script routines are responsible for displaying and hiding the essential sections, ensuring the validity of score inputs, and calculating the final score based on user inputs.

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <title>Archer UI</title>
7   <style>
8     body {
9       font-family: Arial, sans-serif;
10      background-color: #f0f0f0;
11      margin: 0;
12      padding: 0;
13    }
14    .background-image {
15      background-image: url('background.jpg');
16      background-size: cover;
17      position: absolute;
18      width: 100%;
19      height: 100%;
20      z-index: -1;
21      opacity: 0.5;
22    }
23    .apply_content, .score-entry {
24      background-color: white;
25      padding: 20px;
26      margin: 50px auto;
27      border-radius: 10px;
28      box-shadow: 0 0 10px rgba(0,0,0,0.1);
29      max-width: 600px;
```

```

30     }
31     .apply_content h1, .score-entry h1 {
32         color: #333;
33     }
34     .apply_content label, .score-entry label {
35         display: block;
36         margin: 10px 0 5px;
37     }
38     .apply_content select, .apply_content input, .score-entry input {
39         width: 100%;
40         padding: 8px;
41         margin-bottom: 10px;
42         border: 1px solid #ccc;
43         border-radius: 5px;
44     }
45     .apply_content button, .score-entry button {
46         background-color: #4CAF50;
47         color: white;
48         padding: 10px 20px;
49         border: none;
50         border-radius: 5px;
51         cursor: pointer;
52     }
53     .apply_content button:hover, .score-entry button:hover {
54         background-color: #45a049;
55     }
56     .score-input {
57         width: 40px;
58         text-align: center;
59     }
60 </style>
61 </head>
62
63 <body>
64 <?php
65     require './submit_scores.php';
66
67     // Create connection
68     $conn = @mysqli_connect($servername, $username, $password, $dbname);
69
70     // Check connection
71     if (!$conn) {
72         die("Connection failed: " . mysqli_connect_error());
73     }
74     ?>
75     <div class="background-image"></div>
76
77     <div class="apply_content">
78         <h1>Choose round</h1>
79         <form id="round-form" method="post" action="submit_scores.php">
80             <label for="round">Round</label><br>
81             <select name="round" id="round">
82                 <option value="1">Round 1 - 50m</option>
83                 <option value="2">Round 2 - 50m</option>
84                 <option value="3">Round 3 - 90m</option>
85                 <option value="4">Round 4 - 60m</option>
86                 <option value="5">Round 5 - 70m</option>
87             </select>
88             <br><br>
89             <label for="FirstName">Firstname:</label>
90             <input type="text" name="FirstName" id="FirstName" maxlength="20" required /><br>
91             <br>
92             <label for="Lastname">Lastname:</label>
93             <input type="text" name="Lastname" id="Lastname" maxlength="20" required /><br>
94             <br>
95             <label for="equipment">Equipment</label><br>
96             <select name="equipment" id="equipment">
97                 <option value="Recurve">Recurve</option>
98                 <option value="Longbow">Longbow</option>
99                 <option value="Compound">Compound</option>
100                <option value="Recurve barebow">Recurve barebow</option>
101                <option value="Compound barebow">Compound barebow</option>

```

```

102         </select>
103         <br><br>
104         <label for="distance">Target Distance (m):</label>
105         <input type="text" id="distance" name="distance" value="122" required>
106         <br><br>
107         <button type="button" onclick="showScoreEntry()">Go to Score Entry</button>
108     </form>
109 </div>
110
111 <div class="score-entry" id="score-entry" style="display:none;">
112     <h1>Enter Scores</h1>
113     <form id="score-form" method="post" action="submit_scores.php">
114         <div>
115             <label for="scores">Enter your scores:</label><br>
116             <input type="text" class="score-input" name="scores[]" maxlength="1" oninput="enforceOrder(this); calculateTotalScore(); return true;" />
117             <input type="text" class="score-input" name="scores[]" maxlength="1" oninput="enforceOrder(this); calculateTotalScore(); return true;" />
118             <input type="text" class="score-input" name="scores[]" maxlength="1" oninput="enforceOrder(this); calculateTotalScore(); return true;" />
119             <input type="text" class="score-input" name="scores[]" maxlength="1" oninput="enforceOrder(this); calculateTotalScore(); return true;" />
120             <input type="text" class="score-input" name="scores[]" maxlength="1" oninput="enforceOrder(this); calculateTotalScore(); return true;" />
121             <input type="text" class="score-input" name="scores[]" maxlength="1" oninput="enforceOrder(this); calculateTotalScore(); return true;" />
122         </div>
123         <br>
124         <div>
125             <label for="totalScore">Total Score:</label><br>
126             <input type="text" id="totalScore" name="totalScore" readonly />
127         </div>
128         <br>
129         <button type="submit">Submit Scores</button>
130     </form>
131 </div>
132
133 <script>
134     function showScoreEntry() {
135         document.querySelector('.apply_content').style.display = 'none';
136         document.querySelector('.score-entry').style.display = 'block';
137     }
138
139     function enforceOrder(input) {
140         const validScores = ['X', '10', '9', '8', '7', '6', '5', '4', '3', '2', '1', 'M'];
141         let value = input.value.toUpperCase();
142         if (!validScores.includes(value)) {
143             input.value = '';
144         } else {
145             input.value = value;
146         }
147     }
148
149     function calculateTotalScore() {
150         const scoreInputs = document.querySelectorAll('.score-input');
151         let totalScore = 0;
152
153         scoreInputs.forEach(input => {
154             let score = input.value.toUpperCase();
155             if (score === 'X') {
156                 score = 10;
157             } else if (score === 'M') {
158                 score = 0;
159             } else {
160                 score = parseInt(score);
161             }
162             totalScore += score;
163         });
164
165         document.getElementById('totalScore').value = totalScore;
166     }
167 </script>
168 </body>
169
170 </html>
171

```


Joshua's Major Specific Work

As a Cyber-Security major, it is my responsibility to control security threats to the database.

Data

The simplest thing that I have done in the database is ensuring that each field in the database can contain only the smallest type of data necessary, for example, there will never be more than 9 ranges in a round, so the integer value is limited to 1 character. This is present throughout the database, see the [CREATE DATABASE Code](#) for examples.

The next thing that I have implemented on the database is limiting permissions and only granting specific permissions to specific users. (I would have implemented these, however I don't have permissions to on feenix)

Archer -

The archer role would have the least access. They would be able to view a number of tables (see below), they would be unable to access the: RecorderInformation; this is so that archers must request to know which recorder approved their score so there is no chance of conflict between archers and recorders, CompetitionScoreTBA; This is so that archers are not able to access un-authenticated scores.

```
1 GRANT SELECT
2 ON TABLE ArcherCategory, ArcherCompleteScore, ArcherInformation, ClubChampionshipInfo, ClubChampionshipScore, Com
3 TO archer;
```

They would also be able to Insert values into the End, Range and RoundScore Tables, as well as the ArcherCompleteScore and CompetitionScoreTBA tables. This allows archers to insert their individual scores into the table and upload scores to be approved for competitions. (see below)

```
1 GRANT INSERT
2 ON TABLE EndScore, RangeScore, RoundScore, ArcherCompleteScore, CompetitionScoreTBA
3 TO archer;
```

Recorder -

The recorder role would have more permissions than the archer role would, however they still would not have full permissions. Recorders would be able to view all tables (see below)

```
1 GRANT SELECT
2 ON TABLE ArcherCategory, ArcherCompleteScore, ArcherInformation, ClubChampionshipInfo, ClubChampionshipScore, Com
3 TO recorder;
```

The recorder role would have the ability to insert value into a large number of tables, the tables that they would not be able to insert values into would be the RecorderInformation table, ArcherCompleteScore table; this is because recorders have no reason to add or edit entries in this table, ClubChampionshipInformation. (see below)

```
1 GRANT INSERT
2 ON TABLE ArcherCategory, ArcherInformation, ClubChampionshipScore, CompetitionScore, CompetitionInformation, Equi
3 TO recorder;
```

Recorders will also be able to UPDATE the ArcherInformation, CompetitionScoreTBA and RoundType Tables. Updating ArcherInformation to change DefaultEquipment is a requirement, Should a recorder need to edit an entry in the CompetitionScoreTBA table before inserting it, and editing the ValidRound Field in the RoundType Table is neccesary.

```
1 GRANT UPDATE
```

```
2 ON TABLE ArcherInformation, CompetitionScoreTBA, RoundType
3 TO recorder;
```

Recorders will only be able to DELETE from the CompetitionScoreTBA Table in order to ensure that data remains safe

```
1 GRANT DELETE
2 ON TABLE CompetitionScoreTBA
3 TO recorder;
```

Admin -

The Admin role is permitted all permissions, the difference between a recorder and Admin is that the Admin role can Insert Recorder information and is in charge of creating all of the competition and championship information tables, including Equivalent Rounds. It also grants them the ability to update and delete entries from the CompetitionScores table, this is to ensure that data uploaded into this table is secure.

```
1 GRANT ALL PRIVILEGES
2 ON TABLE ArcherCategory, ArcherCompleteScore, ArcherInformation, ClubChampionshipInfo, ClubChampionshipScore, Com
3 TO admin;
```

To prevent SQL insertion, all of the Used Cases Queries should be encased in Procedures, to do this, we would change all of the created Queries to something like this -

```
1 DELIMITER //
2 CREATE PROCEDURE GetArcherInformation(IN archerID INT)
3 BEGIN
4     SELECT * FROM ArcherInformation WHERE ArcherID = archerID;
5 END //
6 DELIMITER ;
```

Meeting Notes

Create meeting note

To take notes and document discussions in workshops and classes

Incomplete tasks from meetings

Task report

Looking good, no incomplete tasks.

Decisions from meetings

Page Title	Decisions
2024-03-05 Meeting notes	<ul style="list-style-type: none">↳ Finish project plan, roles and responsibilities and risk assessment matrix in the workshop↳ Everyone make a rough draft of the database should be designed by next week
2024-03-12 Meeting notes	<ul style="list-style-type: none">↳ Everyone discusses about individual draft database.↳ Everyone agrees on the project proposal.
2024-03-19 Meeting notes	<ul style="list-style-type: none">↳ Everyone commits to completing the Risk assessment matrix and role Responsibility before our workshop on Wednesday.↳ Everyone commits to completing the Empathy by the end of 21/4/2024.
2024-03-26 Meeting notes	<ul style="list-style-type: none">↳ Our Database should be minimum Second Normal Form, aiming for Third.
2024-04-09 Meeting notes	<ul style="list-style-type: none">↳ Individually complete a draft SQL database
2024-04-16 Meeting notes	<ul style="list-style-type: none">↳ Complete project team health monitor in the workshop↳ Submit progress report
2024-04-23 Meeting notes	<ul style="list-style-type: none">↳ Complete SQL commands
2024-05-07 Meeting notes	<ul style="list-style-type: none">↳ Execute SQL commands on the database↳ Discuss individual major contributions next week

2024-05-14 Meeting notes	 Update project documentation to include new individual contributions
2024-05-21 Meeting notes	 Complete and submit finalised project  Discuss and record reflection on the project as a team

All meeting notes

Title	Creator	Modified
2024-05-07 Meeting notes	SUKEYNA ALI	yesterday at 7:55 PM
2024-05-14 Meeting notes	SUKEYNA ALI	yesterday at 7:53 PM
2024-05-21 Meeting notes	SUKEYNA ALI	yesterday at 7:51 PM
2024-04-30 Meeting notes	SUKEYNA ALI	yesterday at 3:42 PM
2024-04-23 Meeting notes	SUKEYNA ALI	May 15, 2024
2024-04-16 Meeting notes	Bảo Minh Nguyễn Phương	May 04, 2024
2024-03-26 Meeting notes	Joshua Burns	May 04, 2024
2024-04-09 Meeting notes	Bảo Minh Nguyễn Phương	Apr 17, 2024
2024-03-12 Meeting notes	Joshua Burns	Mar 24, 2024
2024-03-19 Meeting notes	Joshua Burns	Mar 24, 2024
2024-03-05 Meeting notes	SUKEYNA ALI	Mar 06, 2024
2024-02-28 Workshop Notes	Joshua Burns	Mar 04, 2024

2024-02-28 Workshop Notes

Date

Feb 28, 2024

Participants

- [@Joshua Burns](#)
- [@Bảo Minh Nguyễn Phương](#)
- [@SUKEYNA ALI](#)
- [@Anh Khoa Nguyen](#)

Goals

- Set up Jira and Confluence and familiarize the team with the interface
- Establish a general guideline for how to accomplish our goals on the project

Discussion topics

Time	Item	Presenter	Notes
N/A	Establishing Work Agreement	Josh	<ul style="list-style-type: none">• Discussed work agreement, established a work agreement on confluence that the group agrees on• Discussed how to use the different channels of communication listed in the group agreement
N/A	Introductions	Individual	<ul style="list-style-type: none">• Each individual filled out an introduction and uploaded to Confluence.

Action items

- Discuss and establish work agreement
- Write out introductions
 - Josh
 - Melvin
 - Ryan
 - Sukeyna

Decisions



2024-03-05 Meeting notes

Date

Mar 5, 2024

Participants

- Everyone

Goals

- Go over the project plan
- Mock up a timeline
- Set milestones and deadlines (have provisions in case deadlines are not met)

Discussion topics

Time	Item	Presenter	Notes
05/03/24 (Tutorial)	Project Plan		<ul style="list-style-type: none">• Scopes (recorders)
06/03/24 (Workshop)	Project Plan		<ul style="list-style-type: none">• Scopes (archer)• Timeline• Milestones and Deadlines

Action items

- Complete Project Plan

Decisions

 Finish project plan, roles and responsibilities and risk assessment matrix in the workshop

 Everyone make a rough draft of the database should be designed by next week

2024-03-12 Meeting notes

Date

Mar 12, 2024

Participants

- @Joshua Burns
- @SUKEYNA ALI
- @Anh Khoa Nguyen
- @Bảo Minh Nguyễn Phương

Goals

- Finish project plan.
- Discuss databases draft.

Discussion topics

Time	Item	Presenter	Notes
12/03/2024	Roles and Responsible		<ul style="list-style-type: none">• Divide roles for each member.
13/03/2024	Draft databases Risk Assessment		<ul style="list-style-type: none">• Discuss possibly risks and actions.• Discuss and decide the database format.

Action items

- Complete Responsibility and risk assessment.

Decisions

 Everyone discusses about individual draft database.

 Everyone agrees on the project proposal.

2024-03-19 Meeting notes

Date

Mar 12, 2024

Participants

- @Joshua Burns
- @Bảo Minh Nguyễn Phương
- @Anh Khoa Nguyen
- @SUKEYNA ALI

Goals

- The team attempts to finish the project prosal and consult the supervisor about the documents that are not complete yet (Risk Assessment, Personal map) before the end of week 4.

Discussion topics

Time	Item	Presenter	Notes
19/03/2024	Risk assessment matrix, Role Responsibility		<ul style="list-style-type: none">• Dicuss with supervisors to make some changes to make our documents more towards professional.
20/03/2024	Empathy map		<ul style="list-style-type: none">• Dicuss and complete empathy map and submit this Sunday.• We should add the empathy as an image, like the one we see on the internet (advised by the supervisor). Our content is considered to be good so far.

Action items

- Complete fixing Risk assessment matrix and role Responsibility.
- Asking for supervisor's insight.
- Complete the Empathy map.

Decisions

 Everyone commits to completing the Risk assessment matrix and role Responsibility before our workshop on Wednesday.

 Everyone commits to completing the Empathy by the end of 21/4/2024.

2024-03-26 Meeting notes

Date

Mar 12, 2024

Participants

- @Joshua Burns
- @Bảo Minh Nguyễn Phương
- @Anh Khoa Nguyen
- @SUKEYNA ALI

Goals

- Complete the individual task (Week 4 and Week 5)

Discussion topics

Time	Item	Presenter	Notes
26/03/2024	Normal form	Tutor	<ul style="list-style-type: none">• Discuss 1NF, 2NF, 3NF, why they matter?
26/03/2024	Normalization of the team	Team members	<ul style="list-style-type: none">• Minimum 2NF for our database and aim for 3NF.• Create additional table if necessary.

Action items

- Team members has completed the individual assessment.
- Team members designed ERD attempting to fulfill 3NF.

Decisions

 Our Database should be minimum Second Normal Form, aiming for Third.



2024-04-09 Meeting notes

Date

Apr 9, 2024

Participants

- @Bảo Minh Nguyễn Phương
- @Anh Khoa Nguyen
- @SUKEYNA ALI
- @Joshua Burns

Goals

- Please watch all the videos in the independent study.
- Please participate in all the tasks in Jira, comment on them, and give feedback on tasks assigned to you.

Discussion topics

Time	Item	Presenter	Notes
09/04/2024	MySQL Learning	Tutor	Watch the independent study which links to LinkedIn Learning about MySQL Essential Training.
09/04/2024	Create a draft of MySQL database in our Mercury server.	Everyone	This will be checked back on April 10th

Action items

- Watch the Independent Study Week 6.
- Complete a draft of a MySQL database of your own based on our initial ERD (you can adjust it if you feel it to be better).

Decisions

 Individually complete a draft SQL database

2024-04-16 Meeting notes

Date

Apr 16, 2024

Participants

- [@Bảo Minh Nguyễn Phương](#)
- [@Joshua Burns](#)
- [@Anh Khoa Nguyen](#)
- [@SUKEYNA ALI](#)

Goals

-

Discussion topics

Time	Item	Presenter	Notes
16/04/24	Adding and Manipulating Data	Tutor	<ul style="list-style-type: none">• Watch independent study to learn new SQL commands• Go through Tutorial 7 (Loading the Data)
16/04/24	Project Report	Everyone	<ul style="list-style-type: none">• Upload Confluence• Upload Jira Work Management• Complete project team health monitor• Discuss changes made to the project during the workshop
17/04/24	Project Team Health Monitor	Joshua	<ul style="list-style-type: none">• Complete task as a team
17/04/24	Draft SQL database	Everyone	<ul style="list-style-type: none">• Upload draft databases onto Confluence

Action items

- Watch week 7 independent study videos
- Complete Project team health monitor
- Upload draft databases
- Complete Progress Report

Decisions

-  Complete project team health monitor in the workshop

 Submit progress report

2024-04-23 Meeting notes

Date

Apr 23, 2024

Participants

- [@SUKEYNA ALI](#)
- [@Joshua Burns](#)
- [@Bảo Minh Nguyễn Phương](#)
- [@Anh Khoa Nguyen](#)

Goals

- Complete independent study
- Start implementing SQL commands

Discussion topics

Time	Item	Presenter	Notes
23/04/24	SQL Queries	Everyone	<ul style="list-style-type: none">• Watch independent study materials• Complete tutorial work
24/04/24	SQL Statements	Everyone	<ul style="list-style-type: none">• Delegate SQL commands for user requirements

Action items

- Complete independent contributions
- Complete two SQL commands

Decisions

 Complete SQL commands

2024-04-30 Meeting notes

Date

May 1, 2024

Participants

- [@SUKEYNA ALI](#)
- [@Joshua Burns](#)
- [@Anh Khoa Nguyen](#)
- [@Bảo Minh Nguyễn Phương](#)

Goals

- Complete individual study
- Complete individual assignment
- Finalise SQL commands
- Apply indexes to the database

Discussion topics

Time	Item	Presenter	Notes
30/04/24	Index	Everyone	<ul style="list-style-type: none">• Watch individual study on Indexes• Complete individual tutorial assignment
01/05/24	SQL Commands	Everyone	<ul style="list-style-type: none">• Go through completed SQL commands• Complete additional SQL commands

Action items

- Complete Individual Assignment
- Complete SQL Commands
- Discuss completed SQL Commands
- Discuss Indexing in relation to Archery Club Database

Decisions



2024-05-07 Meeting notes

Date

May 7, 2024

Participants

- [@SUKEYNA ALI](#)
- [@Bảo Minh Nguyễn Phương](#)
- [@Anh Khoa Nguyen](#)
- [@Joshua Burns](#)

Goals

- Complete all SQL commands
- Specialised enhancements of database solutions

Discussion topics

Time	Item	Presenter	Notes
07/05/24	Major-specific contributions	Everyone	<ul style="list-style-type: none">• Work on individual major specific contributions to the project
08/05/24	SQL commands		<ul style="list-style-type: none">• Complete individual SQL commands

Action items

- Discuss completed SQL commands

Decisions

 Execute SQL commands on the database

 Discuss individual major contributions next week

2024-05-14 Meeting notes

Date

May 14, 2024

Participants

- [@SUKEYNA ALI](#)
- [@Bảo Minh Nguyễn Phương](#)
- [@Joshua Burns](#)
- [@Anh Khoa Nguyen](#)

Goals

- Add Major specific to project repository

Discussion topics

Time	Item	Presenter	Notes
14/05/24	Major specific project	Everyone	<ul style="list-style-type: none">• Finalise major specific contributions
15/05/24	Major specific project	Everyone	<ul style="list-style-type: none">• Discuss and explain each major specific contribution• Add all major specific projects into Confluence

Action items

- Discuss individual additions to the project
- Add major specific contributions to Confluence

Decisions

-  Update project documentation to include new individual contributions

2024-05-21 Meeting notes

Date

May 21, 2024

Participants

- [@SUKEYNA ALI](#)
- [@Anh Khoa Nguyen](#)
- [@Joshua Burns](#)
- [@Bảo Minh Nguyễn Phương](#)

Goals

- Finalise project
- Team reflection

Discussion topics

Time	Item	Presenter	Notes
21/05/24	Finalising the project	Everyone	<ul style="list-style-type: none">• Discuss final adjustments to the project• Finalise testing• Prepare project presentation for stakeholders
22/05/24	Project finalisation	Everyone	<ul style="list-style-type: none">• Finalising project documentation and presentation video
22/05/24	Retrospective reflection	Everyone	<ul style="list-style-type: none">• Team retrospective review of the project

Action items

- Finalise testing on the database
- Update project documentation
- Team reflection discussion
- Finalise project presentation

Decisions

 Complete and submit finalised project

 Discuss and record reflection on the project as a team

Team Health Monitor

- i** Use the project team health monitor template to keep track of your team's health. Keep this template in your team space and if there are any areas that you're not confident are green, dive into the plays to get back on track. For detailed facilitation instructions go to [health monitor for project teams](#).

Team name	Banh Pate Chaud
Sponsor	Raid Shadow Legends
Health monitor cadence	Weekly

👩‍💻 Team health assessment

With your team, read the definition of each attribute of healthy, high-performing teams out loud. On the count of three have each person rate how they feel the team is doing compared to each definition (thumbs-up/green, thumbs-sideways/yellow, thumbs-down/red). Record the results of each attribute rating in the table. Highlight each cell using this color code: **HEALTHY** = "We're strong here", **BIT SICK** = "We're ok... but a little shaky", **SICK** = "We're not healthy".

Area	Date	Scheduled
<p>🚩 Full-time owner</p> <p>There is one lead who is accountable for the result of this project. This needs to be someone whose time is at least 80% dedicated to it, and who can champion the mission inside and outside of the team.</p>	Apr 16, 2024	Scheduled Apr 30th
<p>⚖️ Balanced team</p> <p>Roles and responsibilities are clear and agreed upon. The project has people with the right blend of skill set. Acknowledge that team members can change by stage.</p>	Each team member engages in the topics they are proficient in and the team manager knows how to allocate team members based on tasks.	
<p>❤️ Shared understanding</p> <p>The team has a common understanding of why they're here, the problem/need, are convinced about the idea, confident they have what they need, and trust each other.</p>	The task assigned is not within our teams usual skillset (No one is very familiar with Archery) However by acknowledging our short comings and researching the topic we are understanding our customers more efficiently.	

 Value and metrics	We have a clear understanding of the end goal however, due to the above limitation we are not able to identify as well how well our solution solves the problem fully.	
 Proof of concept	We have a draft/V1 of our DB design however have not been able to implement/test it presently.	
 One-pager	We do not have a 1 pager as we had not realised it would be useful/a requirement.	
 Managed dependencies	As we have a complex and clear project plan we are able to identify easily our timeline, the project requirements and their complexity and remain on track, reaching out to support networks if necessary.	
 Velocity	As we are creating a database as the product, most of our time has been theoretical and written work and as such we have not been shipping much concrete work. However we appear to be on-target for our timeline.	

Focus areas

Ask your team to collectively come up with one attribute you want to focus on. Then, call out ways to move the **SICK** or **BIT SICK** toward **HEALTHY**. Make sure they are actionable, specific, and measurable.

Date	Focus areas and action items
16/04/2024	<ul style="list-style-type: none"> <input type="checkbox"/> Create a One-pager <input type="checkbox"/> Achieve concrete progress, implement a physical iteration of the Database. <input type="checkbox"/> Explain/Ask questions about how archery works to better understand the product requirements. <p>Check back on actions as of Apr 30th</p>

4Ls Retrospective

📋 Overview

Reflect back on what you and your team learned and what motivates the group to succeed by following the instructions for the 4Ls Retrospective Play.

Team	Banh Paté Chaud Associates
Team members	@Anh Khoa Nguyen @Joshua Burns @Bảo Minh Nguyễn Phương @SUKEYNA ALI
Date	22/05/2024
Retrospective period	

💡 4Ls retrospective

Milestones	Loved	Longed for	Loathed	Learned
Submit Project Proposal <ul style="list-style-type: none">Discussing requirementsMeeting with stakeholders	<ul style="list-style-type: none">We all equally contributed to the proposal	<ul style="list-style-type: none">Better understanding of the context surrounding the project.	<ul style="list-style-type: none">Lack of contact with stakeholders after the initial Project Brief was arrived at.	<ul style="list-style-type: none">Fundamentals of managed teamwork using the Agile Principles.
Create Database <ul style="list-style-type: none">NormalisationERD DesignIndexing	<ul style="list-style-type: none">Exploration of ideasAllowed for individual approaches	<ul style="list-style-type: none">Inflexibility due to lack of contact with stakeholders.More up-to-date RDMS (relational database management system)	<ul style="list-style-type: none">It felt like we were placed slightly out of our depthTechnical difficulties lead to some members' inability to access the database.	<ul style="list-style-type: none">How to implement a logical database design to a physical database.Impacts of Normalization and how to implement.Impacts of Indexing.
Database Finalisation <ul style="list-style-type: none">Data CreationStress TestingIntegrity TestingIndividual Improvements	<ul style="list-style-type: none">Seeing our work pay offThe ability to utilize our individual skills to improve the final product	<ul style="list-style-type: none">More resources for individual improvements.More freedom to pursue Individual improvements of our choosing	<ul style="list-style-type: none">Lack of notice for Individual improvementsNot being provided with meaningful data from the club to populate the DB.	<ul style="list-style-type: none">Different data-types in a DatabaseEnsuring Data StabilityQueries to interact both with the database directly and

				from a front-end perspective.
Submitting Product Deliverables	<ul style="list-style-type: none"> • Communicating results to stakeholders • Presenting our work to the stakeholders 	<ul style="list-style-type: none"> • Constructive feedback on our product • Ongoing relationship with the Club 	<ul style="list-style-type: none"> • Not being able to present meaningful data to demonstrate the product. 	<ul style="list-style-type: none"> • Interpersonal skills. • Communicating technical specifications in a meaningful, non-jargonistic way.

⚡ Action plan

Action	Due date	Action items
Contact Stakeholders	The day of the Project Proposal	<ul style="list-style-type: none"> • Have team members partake in archery to understand how the system works and what the current system is
Better Technical Support		<ul style="list-style-type: none"> • Utilize a more up to date version of Maria DB • Ensure each team member has multiple ways of accessing a database creation tool.
Be more assertive		<ul style="list-style-type: none"> • Provide the stakeholders with a list of database improvements based on team capabilities, rather than having them instruct us on the improvements we can implement. • Request meaningful data from the club so that we may assert that the database works correctly with pre-existing data.