

Cos20031 Database Design	2
Our Team!	3
Working Agreement Template	4
User Manual - Aiden Large	7
User Manual - Reeve Kariyawasam	8
User Manual - John Jaswanth Carmel Madanu	9
User Manual - Matthew Xu	10
User Manual - Donovan Quilty	11
Meeting notes in space	12
2025-03-04 Meeting notes	14
2025-03-11 Meeting notes	16
2025-03-18 Meeting notes	18
2025-03-25 Meeting notes	19
2025-03-28 Meeting notes	21
2025-04-01 Meeting notes	22
2025-04-08 Meeting notes	23
2025-04-15 Meeting notes	24
2025-04-29 Meeting notes	25
2025-05-06 Meeting notes	26
2025-05-13 Meeting notes	27
2025-05-20 Meeting notes	28
2025-05-27 Meeting notes	29
Project Plan	31
Roles and Responsibilities	34
Risk Assessment Matrix	36
Product Requirements	38
Draft 1 Entity Relationship Diagram	41
Draft 2 Entity Relationship Diagram	42
Final Entity Relationship Diagram	43
Draft 1 Create Table Commands	44
Draft 2 Create Table Commands	50
Final Create Table Commands	55
Dummy Data	57
Indexing	59
Business Logic and Assumptions	61
User Needs	65
Team Health Monitor	70
Software Development Tasks	74
Software Development Outcomes	75
Cybersecurity Major Task - John	81
4Ls Retrospective	84
Product Video Division of Labor	86

Cos20031 Database Design

Description

The purpose of this space is to collaborate in designing a database which will be used for tracking archery statistics.

Recently updated content

-  [4Ls Retrospective](#)
a minute ago • contributed by [DONOVAN QUILTY](#)
-  [Software Development Outcomes](#)
about 4 hours ago • contributed by [DONOVAN QUILTY](#)
-  [Cybersecurity Major Task - John](#)
yesterday at 2:54 PM • contributed by [DONOVAN QUILTY](#)
-  [Project Plan](#)
Jun 01, 2025 • contributed by [DONOVAN QUILTY](#)
-  [2025-05-27 Meeting notes](#)
Jun 01, 2025 • contributed by [Aiden Large](#)
-  [2025-05-20 Meeting notes](#)
Jun 01, 2025 • contributed by [Aiden Large](#)
-  [2025-05-13 Meeting notes](#)
Jun 01, 2025 • contributed by [Aiden Large](#)
-  [2025-05-06 Meeting notes](#)
Jun 01, 2025 • contributed by [Aiden Large](#)
-  [2025-04-29 Meeting notes](#)
Jun 01, 2025 • contributed by [Aiden Large](#)
-  [User Needs](#)
Jun 01, 2025 • contributed by [DONOVAN QUILTY](#)

Contributors

[Aiden Large](#), [DONOVAN QUILTY](#), [Reeve Kariyawasam](#), [John Jaswanth Carmel Madanu](#), [Matthew Xu](#)

Our Team!

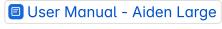
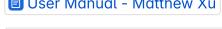
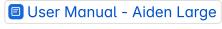
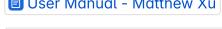
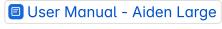
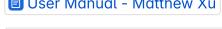
 Welcome to MAJRD 

A Database Design Project team.

 Team metrics 
Deliver Project Proposal
Updated Mar 28, 2025
Target Date Mar 30, 2025
 COMPLETED
Assess Database Performance
Updated May 30, 2025
 COMPLETED
Deploy and Deliver Final Product
Updated Jun 1, 2025
Target Date Jun 1, 2025
 COMPLETED

 About MAJRD 
We are a team of tech enthusiasts focused on building a database-driven score-tracking application for target archery. Our goal is to provide competing archers with a simple and streamlined way to record their scores and see how they measure up against the competition.

 Meet the team 												
<table border="1"><tr><td> @Aiden Large</td><td> @DONOVAN QUILTY</td></tr><tr><td>Team Member</td><td>Team Member</td></tr><tr><td> @Matthew Xu</td><td> @Reeve Kariyawasam</td></tr><tr><td>Team Member</td><td>Team Member</td></tr><tr><td> @John Jaswanth Carmel Madanu</td><td></td></tr><tr><td>Team Member</td><td></td></tr></table>	 @Aiden Large	 @DONOVAN QUILTY	Team Member	Team Member	 @Matthew Xu	 @Reeve Kariyawasam	Team Member	Team Member	 @John Jaswanth Carmel Madanu		Team Member	
 @Aiden Large	 @DONOVAN QUILTY											
Team Member	Team Member											
 @Matthew Xu	 @Reeve Kariyawasam											
Team Member	Team Member											
 @John Jaswanth Carmel Madanu												
Team Member												

 About the team 					
<table border="1"><tr><td> User Manual - Aiden Large</td></tr><tr><td> User Manual - Matthew Xu</td></tr><tr><td> User Manual - John Jaswanth Carmel Madanu</td></tr><tr><td> User Manual - Donovan Quilty</td></tr><tr><td> User Manual - Reeve Kariyawasam</td></tr></table>	 User Manual - Aiden Large	 User Manual - Matthew Xu	 User Manual - John Jaswanth Carmel Madanu	 User Manual - Donovan Quilty	 User Manual - Reeve Kariyawasam
 User Manual - Aiden Large					
 User Manual - Matthew Xu					
 User Manual - John Jaswanth Carmel Madanu					
 User Manual - Donovan Quilty					
 User Manual - Reeve Kariyawasam					

 **Resources** 

Restrict search to this space's space key.

 Latest updates 
2025-03-28 Meeting notes
2025-03-25 Meeting notes
2025-03-18 Meeting notes
2025-03-11 Meeting notes
2025-03-04 Meeting notes

 Where to find us 
 Discord - Group Chat That's All Fun & Games
 For individual emails, take a look the relevant member's user manual.

 Featured resources 
 Project Plan
 Product Requirements
 Draft 1 Entity Relationship Diagram

Working Agreement Template 
This working agreement outlines how our team collaborates effectively on our project. It covers our working hours, preferred communication channels, meeting structures, and feedback preferences.
 Working Agreement Template

Pre-Planning 
The project plan outline the requirements, scope, and objectives whereas the risk assessment matrix identifies any potential issues and/or complications that may occur during the development of this project.
 Project Plan
 Risk Assessment Matrix

3

Working Agreement Template

👥 Team Preferences 💬

Team Member	@DONOVAN QUILTY	@Reeve Kariyawasam	@Matthew Xu	@Aiden Large	@John Jaswanth Carmel Madanu
Working location and timezone	Melbourne AEDT	Melbourne AEDT	Melbourne AEDT	Melbourne AEDT	Melbourne AEDT
Working hours and commitments	I got all the time in the world...	I have some time	Everyday but Thursdays	Hours change based on work roster	Anytime in morning and afternoon
Working environment and preferences	I mostly work at home unless I'm in a class	I work from home & the university library.	Mostly work from home. At campus on Wednesdays & Thursdays.	Mostly working from home, but I am on campus Tuesdays and Thursdays	Mostly working from home. I will be at campus on Tuesdays and Thursdays
How I like receiving feedback	Directly, just give it to me straight.	F2F, tell me what my mistakes are and how I can improve.	Just message or talk to me about it anytime.	Message me about any criticisms or improvements I could make.	Talk to me about it.
Context about me	I have a large cat and a larger dog, I like to play music + tennis and write code.	I'm into running & general fitness.	I like to have plenty of balance between work and hobbies.	I am in my fifth year of studying engineering and computer science.	I am currently studying my second year. I love playing sports and gaming.

💬 Communication Channels 💬

Channel	Purpose	Audience	Standards
Discord	Regular communication, updates, bits and pieces, 'informal'	Everyone	Respond when convenient, check regularly enough that you won't miss anything.
Email	More formal, evidence for when	Everyone	Just casual stuff, be clear. Make sure the

	issues arise.		email has a clear subject.
--	---------------	--	----------------------------

📅 Meetings 🎯

Objective	<ul style="list-style-type: none"> Share progress with group / stakeholders. 	<ul style="list-style-type: none"> Work collaboratively on weekly project tasks. 	<ul style="list-style-type: none"> Distribute work for the coming week between group members.
Outcomes	<ul style="list-style-type: none"> Group members / stakeholders have been made aware of and shared their thoughts on current progress. 	<ul style="list-style-type: none"> Group has started work on current week project tasks. 	<ul style="list-style-type: none"> Each group member is aware of their project responsibilities for the coming week.
Format	<ul style="list-style-type: none"> Meeting part 1: Each group member is given a chance to share their progress and respond to progress of other members. 	<ul style="list-style-type: none"> Meeting part 2: Group members divide responsibilities for the coming week. 	<ul style="list-style-type: none"> Meeting part 3: Group members start working on their assigned tasks.
Who	<p>Group members:</p> <ul style="list-style-type: none"> @DONOVAN QUILTY @Reeve Kariyawasam @Matthew Xu @Aiden Large @John Jaswanth Carmel Madanu 	<p>Tutor (during workshops):</p> <ul style="list-style-type: none"> 	
Resources	<p>Discord:</p> <ul style="list-style-type: none"> Sharing resources Updates Online meetings Informal messages 	<p>Confluence:</p> <ul style="list-style-type: none"> Collaborative project workspace 	<p>Email:</p> <ul style="list-style-type: none"> Messaging tutor Formal messages
How will we show up?	<ul style="list-style-type: none"> Respectful Ready to share updates Having read through weekly project material and tasks 		
How will we manage follow	Discord:	Meetings:	

up?	<ul style="list-style-type: none"> Share updates or questions about assigned tasks. 	<ul style="list-style-type: none"> There will be a section of each meeting dedicated to following up on work assigned in previous meeting (see “Format” section above). 	
------------	--	--	--

⬆ Escalation Process ⬇

Decider	How	Transparency	Feedback Loop
Group	A message including details about the issue will be sent to the group chat (Discord group), and each member will share thoughts about potential solutions before a collective decision is agreed upon.	All group members will be made aware of the problem and be given a chance to share or vote on potential solutions.	The final decision will be explicitly stated in the group chat.
To our tutor	We will decide as a group to email/speak with them.	Any communication being done will be shared with the team in advance	We will share the response from the tutor with our team.

💡 Continuous Improvement 💡

Purpose	How	Standards
Feedback: To help each other improve their solutions and work.	F2F, Discord	Feedback to be provided objectively with empathy and understanding.
Checking in on others' progress	F2F, Discord	Tell others what you're up to, then ask about everyone else's progress.

User Manual - Aiden Large

Environments I like to work in	<ul style="list-style-type: none">• Quiet spaces.• Cooperative team members.
Preferred working hours	Preferably in the afternoon.
Communication preferences	<ul style="list-style-type: none">• Discord messages/calls• Email
Preferred ways to receive feedback	Feel free to message me with any criticism you have about my work. Though I would appreciate specific examples of what needs improving rather than vague suggestions (such as by providing screenshots of problematic lines).
Things I need	Plenty of coffee.
How I learn best	<ul style="list-style-type: none">• In quiet spaces.• Examples of how something should be done.
Things I struggle with	I often struggle with report writing and prefer more practical tasks.
Things I love	<ul style="list-style-type: none">• Coffee.• Cold weather (but not too cold).
If I were an animated gif/meme/animal/song, I would be...	Probably a very cool dog.
My favorite saying	“Let it be.”
Other things I want you to know about me	I recently adopted a dog. His name is Percy.

User Manual - Reeve Kariyawasam

Environments I like to work in	<ul style="list-style-type: none">Wide open spaces with fresh air and floodlights.
Preferred working hours	<ul style="list-style-type: none">Start early in the morning and finish the day early. 6am - 2pm.
Communication preferences	<ul style="list-style-type: none">Face to face or live audio.
Preferred ways to receive feedback	<ul style="list-style-type: none">Show me my mistakes and show me where and how I can improve.F2F or text.
Things I need	Internet, good keyboard, good weather.
How I learn best	Watching videos, reading manuals and experimenting.
Things I struggle with	Being productive around others.
Things I love	Green curry.
If I were an animated gif/meme/animal/song, I would be...	Animal - Tortoise.
My favorite saying	You only live once.
Other things I want you to know about me	I enjoy computer science! 😊

User Manual - John Jaswanth Carmel Madanu

Environments I like to work in	Quiet and Peaceful environments		
Preferred working hours	Anytime in the morning and afternoon		
Communication preferences	Mail, Teams, Discord		
Preferred ways to receive feedback	one on one		
Things I need	A structured plan to stay on track with goals		
How I learn best	I learn best by learning doing it.		
Things I struggle with	Juggling multiple tasks at once sometimes I need to prioritise better.		
Things I love	Sports, Gaming, Music		
If I were an animated gif/meme/animal/song, I would be...	A loading symbol—because I'm always processing and improving.		
My favourite saying	Work smart, not harder		
Other things I want you to know about me	I enjoy solving technical challenges and figuring out how things work		

User Manual - Matthew Xu

Environments I like to work in	Home
Preferred working hours	8am - 5pm
Communication preferences	Discord, Messenger, Email
Preferred ways to receive feedback	One on one
Things I need	Employment
How I learn best	Early mornings in a low stress environment
Things I struggle with	Cramming, reading non-fiction
Things I love	Climbing, reading, playing bass
If I were an animated gif/meme/animal/song, I would be...	
My favorite saying	
Other things I want you to know about me	I like cats

User Manual - Donovan Quilty

Environments I like to work in	<ul style="list-style-type: none">• Quiet, low stimulation with good chairs so I can sit comfortably
Preferred working hours	1pm-9pm
Communication preferences	Discord, but anything is fine, really.
Preferred ways to receive feedback	Just give it to me straight
Things I need	Cooperation, communication and a fun attitude
How I learn best	By doing, practicing different things with hands on experience
Things I struggle with	Giving criticism, letting people do their thing.
Things I love	Coding, playing music, working hard, my pets.
If I were an animated gif/meme/animal/song, I would be...	the guy with the really low res sounding sweating while he slams on his keyboard playing a game
My favorite saying	When the going gets tough the tough get going
Other things I want you to know about me	I'm really impressed that you've read all the way down here.

Meeting notes in space

[Create meeting note](#)

Incomplete tasks from meetings

Description	Due date	Assignee	Task appears on
<input type="checkbox"/> Determining the necessity for transaction for each use case.			2025-04-29 Meeting notes

Decisions from meetings

Title	Decisions
2025-03-25 Meeting notes	 Work on remaining Confluence pages was divided between project members.
2025-03-28 Meeting notes	 Deadline was set for 12:00PM on 30/03, after which the project proposal Confluence pages will be submitted as they are.
2025-05-13 Meeting notes	 The webpages were divided as follows: <ul style="list-style-type: none">- Add scores (Aiden)- View scores (Matt)- View rounds / add rounds (Donovan)- View / add / edit archers (Reeve)- View / add / edit competitions (Reeve)
2025-05-27 Meeting notes	 Product video sections were distributed between group members. (Can be seen in the “Product Video Division of Labor” page)

All meeting notes

Title	Creator	Modified
2025-05-27 Meeting notes	Aiden Large	Jun 01, 2025
2025-05-20 Meeting notes	Aiden Large	Jun 01, 2025
2025-05-13 Meeting notes	Aiden Large	Jun 01, 2025
2025-05-06 Meeting notes	Aiden Large	Jun 01, 2025
2025-04-29 Meeting notes	Aiden Large	Jun 01, 2025
2025-04-08 Meeting notes	Aiden Large	Jun 01, 2025
2025-03-25 Meeting notes	Aiden Large	Apr 29, 2025

2025-04-01 Meeting notes	Aiden Large	Apr 27, 2025
2025-04-15 Meeting notes	Aiden Large	Apr 27, 2025
2025-03-28 Meeting notes	Aiden Large	Mar 30, 2025
2025-03-11 Meeting notes	Aiden Large	Mar 28, 2025
2025-03-04 Meeting notes	Aiden Large	Mar 28, 2025
2025-03-18 Meeting notes	Aiden Large	Mar 28, 2025

2025-03-04 Meeting notes

📅 Date

Mar 4, 2025

👥 Participants

- [@Aiden Large](#)
- [@DONOVAN QUILTY](#)
- [@Reeve Kariyawasam](#)
- [@Matthew Xu](#)
- [@John Jaswanth Carmel Madanu](#)

📋 Goals

- Form a group.
- Create a Confluence space for archery project.
- Create team homepage.
- Create user manuals for each group member.
- Create working agreement.

🗣 Discussion topics

Time	Item	Presenter	Notes
2:30PM	Forming groups	Whole team	<ul style="list-style-type: none">• Members chose to form a group and made introductions.
2:45PM	Communication method	Whole team	<ul style="list-style-type: none">• A Discord group chat was created and will be used for communication.
3:00PM	Confluence space	Whole team	<ul style="list-style-type: none">• A Confluence space was created for collaborative work on the archery database design project.• Each member started work on their individual user manuals.• A team homepage was created for the purpose of introducing the team, as well as listing project goals.• Designed a working agreement which will be used for organizing meetings, setting standards and resolving possible conflicts in the future.

✓ Action items

- User Manual
- Working Agreement

 Decisions 

RY

2025-03-11 Meeting notes

📅 Date

Mar 11, 2025

👥 Participants

- [@Aiden Large](#)
- [@DONOVAN QUILTY](#)
- [@John Jaswanth Carmel Madanu](#)
- [@Reeve Kariyawasam](#)

📋 Goals

Create the following:

- Project plan
- Roles and responsibilities
- Risk assessment matrix

🗣 Discussion topics

Time	Item	Presenter	Notes
2:45	Project plan	Whole team	<ul style="list-style-type: none">• A project plan was created in collaboration with the team.• Each member contributed their individual ideas to the plan.• Aiden worked on the scope of the project.
3:30	Roles and Responsibilities	Whole team	<ul style="list-style-type: none">• The roles and responsibilities page was created with the team.• We brainstormed to discuss possible responsibilities that may come up, and how they should be delegated.• We created 5 different roles to assign to each group member.
3:45	Risk Assessment Matrix	Whole Team	<ul style="list-style-type: none">• The Risk Assessment Matrix was created by the team.• We brainstormed to discuss possible scenarios and how they might impact the project.• Each member wrote their individual ideas into the matrix.

✓ Action items

- Project plan
- Roles and responsibilities

Risk assessment matrix

Decisions

Y

2025-03-18 Meeting notes

📅 Date [🔗](#)

Mar 18, 2025

👥 Participants [🔗](#)

- [@Aiden Large](#)
- [@DONOVAN QUILTY](#)
- [@Reeve Kariyawasam](#)
- [@Matthew Xu](#)
- [@John Jaswanth Carmel Madanu](#)

📋 Goals [🔗](#)

- Design an initial database ER diagram.

🗣 Discussion topics [🔗](#)

Time	Item	Presenter	Notes
2:30	Reviewing work done in the previous week.	Whole team	<ul style="list-style-type: none">• Scope section was updated to reflect a greater understanding of the project requirements.
2:45	Beginning work on initial ER diagram.	Whole team	<ul style="list-style-type: none">• Whole group discussed possible entities to include within the database, linking them practically via primary keys and foreign keys.• Discussing entity fields illuminated many knowledge gaps in regards to common archery terms, which were promptly cleared up with the tutor.

✓ Action items [🔗](#)

- ✓ Design initial ER diagram.

⌚ Decisions [🔗](#)



2025-03-25 Meeting notes

📅 Date

Mar 25, 2025

👥 Participants

- [@Aiden Large](#)
- [@DONOVAN QUILTY](#)
- [@Reeve Kariyawasam](#)
- [@Matthew Xu](#)
- [@John Jaswanth Carmel Madanu](#)

📋 Goals

- Review current work done towards project proposal.
- Allocate remaining workload between team members.

🗣 Discussion topics

Time	Item	Presenter	Notes
4:00PM	Confluence Space Review	Whole team	<ul style="list-style-type: none">• Entire group looked through each Confluence space and identified sections where work could still be done.<ul style="list-style-type: none">◦ Primary points of focus were the home page, the product requirements page and the ER diagram.• Aiden updated the meeting notes.
4:15PM	Wrap up	Whole team	<ul style="list-style-type: none">• Remaining project proposal work was divided among members:<ul style="list-style-type: none">◦ Aiden: Meeting notes◦ Donovan: Risk assessment matrix◦ Reeve: ER diagram◦ John: Product requirements◦ Matthew: Home page

✓ Action items

- ✓ [@Aiden Large](#) Meeting notes
- ✓ [@DONOVAN QUILTY](#) Risk assessment matrix
- ✓ [@Reeve Kariyawasam](#) ER diagram
- ✓ [@John Jaswanth Carmel Madanu](#) Product requirements
- ✓ [@Matthew Xu](#) Home page

⌚ Decisions

 Work on remaining Confluence pages was divided between project members.

2025-03-28 Meeting notes

📅 Date

Mar 28, 2025

👥 Participants

- [@Aiden Large](#)
- [@DONOVAN QUILTY](#)
- [@Matthew Xu](#)

📋 Goals

- Review project proposal progress.
- Set a deadline for submission.

🗣 Discussion topics

Time	Item	Presenter	Notes
10:00AM	Project proposal review	Whole team	<ul style="list-style-type: none">• Matthew has added many subsections to the home page, and it is now mostly complete.• Donovan still has to work on the risk assessment matrix, but it is mostly complete.• Aiden has updated the meeting notes to be more thorough.• Reeve has made some changes to the ER diagram and has added entries into the requirements table of the Product Requirements page• A deadline has been chosen for submission (12:00PM on 30/03) and shared with the rest of the group via Discord.

✓ Action items

- ✓ Complete assigned Confluence pages before the chosen deadline.

↻ Decisions

👉 Deadline was set for 12:00PM on 30/03, after which the project proposal Confluence pages will be submitted as they are.

2025-04-01 Meeting notes

📅 Date

Apr 1, 2025

👥 Participants

- [@Aiden Large](#)
- [@DONOVAN QUILTY](#)
- [@Reeve Kariyawasam](#)
- [@Matthew Xu](#)
- [@John Jaswanth Carmel Madanu](#)

📋 Goals

- Normalize current ER diagram.

🗣 Discussion topics

Time	Item	Presenter	Notes
3:30	Normalization		<ul style="list-style-type: none">• The current ER diagram was analyzed to see how it could be normalized. This process highlighted some significant problems with the current design which will need to be further looked into and improved.

✓ Action items



⌚ Decisions



2025-04-08 Meeting notes

📅 Date [🔗](#)

Apr 8, 2025

👥 Participants [🔗](#)

- [@Aiden Large](#)
- [@DONOVAN QUILTY](#)
- [@Reeve Kariyawasam](#)
- [@Matthew Xu](#)
- [@John Jaswanth Carmel Madanu](#)

📋 Goals [🔗](#)

- Write and implement “CREATE TABLE” statements for each of the archery project tables outlined in the ER diagram.

🗣 Discussion topics [🔗](#)

Time	Item	Presenter	Notes
3:30	Create table commands		<ul style="list-style-type: none">• Team collaborated to design “CREATE TABLE” commands for each of the tables drawn in the ER diagram.• Work was split up with each member designated 2 or 3 tables to write the command for.• This activity revealed some key issues with our current database design, particularly in the design of the championship and competition tables which will need to be revised after gaining a clearer idea of what they should include.

✓ Action items [🔗](#)

- Write “CREATE TABLE” statements.
- Use the designed statements to implement the tables into a physical database.

⤳ Decisions [🔗](#)



2025-04-15 Meeting notes

📅 Date

Apr 15, 2025

👥 Participants

- @Aiden Large
- @DONOVAN QUILTY
- @Reeve Kariyawasam
- @Matthew Xu
- @John Jaswanth Carmel Madanu

📋 Goals

- Complete the team health monitor task.

🗣 Discussion topics

Time	Item	Presenter	Notes
3:45	Team Health Monitor		<ul style="list-style-type: none">• Each member of the team voted on how they thought the team was faring in terms of various aspects. The general response was positive.

✓ Action items

↻ Decisions

↻

2025-04-29 Meeting notes

📅 Date [🔗](#)

Apr 29, 2025

👥 Participants [🔗](#)

- [@Aiden Large](#)
- [@DONOVAN QUILTY](#)
- [@Reeve Kariyawasam](#)
- [@Matthew Xu](#)
- [@John Jaswanth Carmel Madanu](#)

📋 Goals [🔗](#)

- Create a list of user needs (e.g., view personal scores for a round, add an archer, etc.).
- Write functionality for these needs (SELECT, INSERT, UPDATE, DELETE statements).
- Decide whether there is a need for transactions, and implement if necessary.

🗣 Discussion topics [🔗](#)

Time	Item	Presenter	Notes
3:30			<ul style="list-style-type: none">• Began to design a list of user needs, along with MySQL statements necessary to implement them.

✓ Action items [🔗](#)

- Design a list of user needs.
- For each user need, write the necessary functionality.

⌚ Decisions [🔗](#)



Tasks To Be Done [🔗](#)

- Determining the necessity for transaction for each use case.

2025-05-06 Meeting notes

📅 Date [🔗](#)

May 6, 2025

👥 Participants [🔗](#)

- [@Aiden Large](#)
- [@DONOVAN QUILTY](#)
- [@Reeve Kariyawasam](#)
- [@Matthew Xu](#)
- [@John Jaswanth Carmel Madanu](#)

📋 Goals [🔗](#)

- Decide how enumerators should be used in the database.
-

🗣 Discussion topics [🔗](#)

Time	Item	Presenter	Notes
3:00	Enumerators		<ul style="list-style-type: none">• Members were confused about how enumerators should be used in the archery database.• After talking with the tutor, it was decided that gender, division, shooting distance and arrow score would be represented by enumerators.
3:30	Arrow score		<ul style="list-style-type: none">• Members discussed how individual arrow scores would be stored within the archery database, taking into account the possibilities of scoring M or X.

✓ Action items [🔗](#)



↻ Decisions [🔗](#)



2025-05-13 Meeting notes

📅 Date [🔗](#)

May 13, 2025

👥 Participants [🔗](#)

- [@Aiden Large](#)
- [@DONOVAN QUILTY](#)
- [@Reeve Kariyawasam](#)
- [@Matthew Xu](#)
- [@John Jaswanth Carmel Madanu](#)

📋 Goals [🔗](#)

- Make a list of pages to design for the software development major specific work / split this work up.

🗣 Discussion topics [🔗](#)

Time	Item	Presenter	Notes
2:30	Webpages		<ul style="list-style-type: none">• Group members listed potential webpages to add during development of the website (in the “Software Development Tasks” page), as well as what each of these pages will need to accomplish.

✓ Action items [🔗](#)

- Add “base round” and “date changed” to “RoundCategories” table. (Potentially also change name to “EquivalentRounds”).
- Add arrows as columns under the “ShotEnds” table.

⌚ Decisions [🔗](#)

👉 The webpages were divided as follows:

- Add scores (Aiden)
- View scores (Matt)
- View rounds / add rounds (Donovan)
- View / add / edit archers (Reeve)
- View / add / edit competitions (Reeve)

2025-05-20 Meeting notes

📅 Date [🔗](#)

May 20, 2025

👥 Participants [🔗](#)

- [@Aiden Large](#)
- [@DONOVAN QUILTY](#)
- [@Matthew Xu](#)
- [@Reeve Kariyawasam](#)

📋 Goals [🔗](#)

- Find out where everyone is at with development of the website
- Figure out a timeline for the final week of semester. To set due dates for everything.

🗣 Discussion topics [🔗](#)

Time	Item	Presenter	Notes
14:38	Website Skeleton (SD major task)	Aiden & Donovan	<ul style="list-style-type: none">• Discussed order of input for scores in scores page, initial repo setup.
14:58	Website - design	Whole team	<ul style="list-style-type: none">• Discussed timeline for completing the major specific task (website)
15:16	Website - file structure	Whole team	<ul style="list-style-type: none">• Discussed appropriate file structure, repo setup on github.
15:25	Indexing - DB structure	Donovan	<ul style="list-style-type: none">• Discussed potential areas we could use indexing in the ER diagram.
15:29	Website - Security	Donovan	<ul style="list-style-type: none">• Do we hash database credentials?
15:44	Website - timeline	Whole team	<ul style="list-style-type: none">• Considered time required to create product video and when we need the website completed by.

✓ Action items [🔗](#)

- Figure out if we would hash database credentials: [if we have time to spare]
- Set due dates for remaining tasks: [website: 2 days before submission, video: remaining 2 days]

➡ Decisions [🔗](#)



2025-05-27 Meeting notes

📅 Date [🔗](#)

May 27, 2025

👥 Participants [🔗](#)

- [@Aiden Large](#)
- [@DONOVAN QUILTY](#)
- [@Reeve Kariyawasam](#)
- [@Matthew Xu](#)
- [@John Jaswanth Carmel Madanu](#)

📋 Goals [🔗](#)

- Review major specific work and necessary changes.
- Complete 4Ls Confluence activity.
- Divide work for product video.

🗣 Discussion topics [🔗](#)

Time	Item	Presenter	Notes
3:00	Review		<ul style="list-style-type: none">• Met with the tutor to discuss indexing and what is expected. It was explained that the amount of indexes included in the final product can vary, so long as the choices of indexes can be explained.• Reviewed major specific work, and discussed necessary changes to the database which were identified as a result of this work. This included adding first name and last name fields to the archer table.
3:45	4Ls Activity		<ul style="list-style-type: none">• The team used the 4Ls Confluence template to reflect on what was loved, longed for, loathed and learned throughout the project timeline.• Three primary milestones were identified (ER diagram design, MYSQL implementation and major specific work).• Each member was free to contribute to the 4Ls table.

✓ Action items [🔗](#)

- Complete product video (each member assigned a section)

⬆️ Decisions [🔗](#)

▶ Product video sections were distributed between group members. (Can be seen in the “Product Video Division of Labor” page)

Project Plan

Contributors	@DONOVAN QUILTY @Aiden Large @Reeve Kariyawasam @Matthew Xu @John Jaswanth Carmel Madanu
Informed	<ul style="list-style-type: none"> Irene Moser, client. Archers (Want to view their scores, scores of others and competition/championship information) Recorders (Need to be able to enter and verify round data new club members, competitions and championships)
Objective	To create a working, relational database to track archery scores. It should be fully functional and adhere to all client requirements.
Due date	Jun 1, 2025
Key outcomes	<p>Working physical database created</p> <p>Document on data creation and null values</p> <p>Indexes on performance</p> <p>ER Diagram</p>
Status	NOT STARTED / IN PROGRESS / COMPLETE

💡 Problem Statement 🧐

We must have a fully working relational database able to track archers scores, and have recorders able enter the scores of ends shot by archers through a simple app. We'll know we've succeeded if Records can approve scores entered by archers or enter scores themselves, and be able to add new archers, and if Archers can easily display and filter through their scores, as well as look at the top scores of the year and the results of any competitions that take place.

🎯 Scope 🎯

Must have:	<ul style="list-style-type: none"> Design a relational database which contains the following information stored in tables, including appropriate column names and data types: <ul style="list-style-type: none"> Club members Round types Rounds (played) <ul style="list-style-type: none"> Ends Arrows Competitions Championships Write documentation regarding SQL statements required for clients to interact with the database, including statements which would allow users to:
-------------------	--

	<ul style="list-style-type: none"> ◦ Fetch all rounds played, restricting the search by club member, round type and date. ◦ Add and update rounds, ends and arrows played by archers. ◦ Add club members. ◦ Add and fetch competitions and championships. <ul style="list-style-type: none"> • The Database must have all information needed to identify archer's class • The past records must remain valid even if the rules change.
Nice to have:	<ul style="list-style-type: none"> • Simple app which allows users to interact with the database, differing in functionality depending on if the user is an archer or a recorder. • Archers must be able to: <ul style="list-style-type: none"> ◦ look up information about rounds played by club members, restricting their searches by date and round type. ◦ look up their own personal best score for each round type. ◦ look up personal best scores for a round type across the entire club, ordered by highest score and differentiated by equipment used. ◦ record a new round into a staging table, including information such as date and time, round type, and equipment. ◦ look up round definitions, including equivalent rounds. ◦ view club competition information such as eligible round types, member placements and scores. • Round information must include: <ul style="list-style-type: none"> ◦ Shooter. ◦ Round type. ◦ Individual arrow scores. ◦ Total score. ◦ Date. ◦ Validation status. • Club recorders must be able to: <ul style="list-style-type: none"> ◦ enter new archers into the system, including their Archery Victoria ID, information which makes up their class (birth year and gender), as well as their default equipment. ◦ enter new round types into the system, including the round name and ranges. ◦ enter new club competitions including information such as the base round, the venue and date of the competition and whether it is a part of a championship. ◦ enter new championships, including the participating round types and the duration of the competition ◦ verify rounds stored in the staging table. • Archers may want to create custom groups of competition with their friends and be able to search for results within that specific group • Archers may want to search or organise their results by more than just date and score
Not in scope:	<ul style="list-style-type: none"> • There does not need to be an easy way for recorders to edit scores once they have been submitted into the database. • There does not need to be a way to remove the information of any archers/any archer IDs once they have been entered into the database

📅 Timeline



▶ Milestones and deadlines

Milestone	Owner	Deadline	Status
Complete Project Proposal		Mar 30, 2025	Completed On Time
Complete Database Design		May 1, 2025	Completed On Time
SQL Implementation Done		May 8, 2025	Completed On Time
Complete Progress Report		Apr 27, 2025	Completed On Time
Complete Process and Product Video		Jun 1, 2025	Completed On Time
Complete Final Report and Product Deliverable		Jun 1, 2025	Completed On Time

🔗 Reference materials

Database for Archery Score Recording.pdf

Roles and Responsibilities

📋 Overview ↗

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

Team	MAJRD
Team members	@DONOVAN QUILTY @Aiden Large @Matthew Xu @Reeve Kariyawasam @John Jaswanth Carmel Madanu
Date	11/03/2025
Team mission	To create fully working database able to track archers scores, and enable recorders to enter new members, competitions, and scores of ends shot by archers through an user friendly app.

📘 Roles and responsibilities ↗

Roles	Assigned Team Member	Responsibilities
Team Lead/Project Manager	@DONOVAN QUILTY	<ul style="list-style-type: none">Ensures all the pieces are coming togetherImplements the connections between the functionalityResponsible for minutes and confluence notes
Database Designer	@Aiden Large	<ul style="list-style-type: none">Ensures that the database is set up to function smoothly.Creates the relational database model.Optimizes for performance.
Database Implementer	@Reeve Kariyawasam	<ul style="list-style-type: none">Implements the actual databaseCreates the SQL commandsTests to ensure they work as expected
Front End Designer	@Matthew Xu	<ul style="list-style-type: none">Creates any queries for data that are neededCreates the SQL commands that allow for data to be inputted into the databaseTests to ensure data is added correctly
Report Producer	@John Jaswanth Carmel Madanu	<ul style="list-style-type: none">Creates the report to deliver to shareholders

		<ul style="list-style-type: none">• Ensures that the final product meets the brief• Writes documentation for the front-end and the database.
--	--	---

Unassigned responsibilities

- Create product video
- Create presentation

Risk Assessment Matrix

📋 Background ↗

Our goal is to create a relational database tracking individual and competition scores in Target Archery. The two major stakeholders are Archers and Recorders. Archers should have little to no issue accessing their scores and information, and information should be protected to make sure it is not wiped. Recorders should be able to easily add information into the database and make larger changes such as setting the club championships or adding a new archer.

 [Project Plan](#) See our project plan for more information.

💡 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>	Unappealing front-end UI	User input errors	UI struggles to interact with the database/input time is slow.	Database is not backed up and therefore lost during production.
POSSIBLE <i>Risk will likely occur</i>	Database performance is unoptimized/the re are bottlenecks.	Wrong entry of scores	Incorrect data entry Data is not shaped properly	Security issues/database is vulnerable to attacks.

PROBABLE <i>Risk will occur</i>	Archer entering incomplete scores	Misclassification of archers class, gender, and division	Duplicate Records	Failure to track score history properly
---	-----------------------------------	--	-------------------	---

✓ Action items ☰

- Ensure any important data is encrypted
- Ensure data is sanitized prior to input to prevent SQL injections
- Ensure database is improved and iterated upon after being created and tested with real data
- Back up database regularly and use a version control system such as Git whenever code is implemented

Product Requirements

Target release	Jun 1, 2025
Document status	DRAFT
Document owner	@DONOVAN QUILTY @Reeve Kariyawasam @Matthew Xu @Aiden Large @John Jaswanth Carmel Madanu

🎯 Objective 🔗

The goal is to create a structured database for Archery Score Recording System that allows archery club recorders to efficiently record, manage, and retrieve individual and competition scores. The system will provide functionalities for archers to track their performance, club recorders to maintain accurate records.

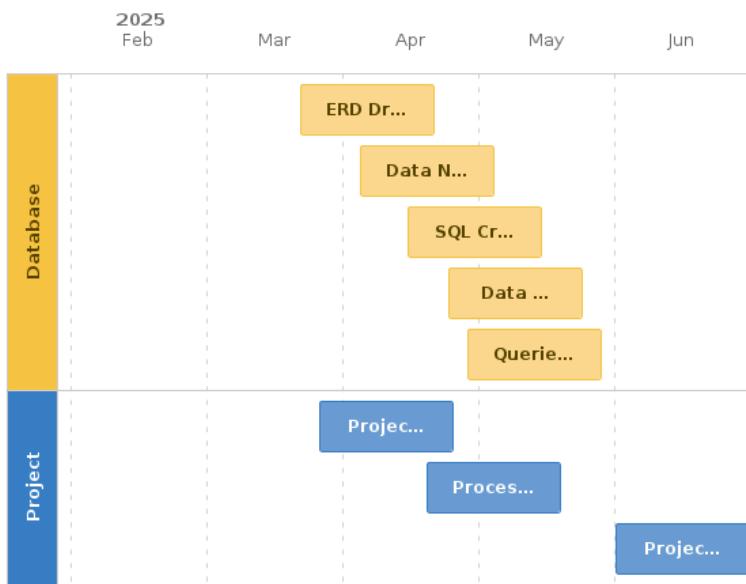
📊 Success metrics

Goal	Metric
Lookup of Scores by Archer	Archer can lookup their
Allow users to track scores	Measure total score in each round
Record competition results	Measure individual scores against other participants

👤 Assumptions 🔗

- An Archer may use different equipment on different days
- scores must be recorded with the specific equipment that archer uses.
- Some competitions contribute to Club championship, and not all other scores are considered for championship
- A round exists of specific number of ends and distances.

⭐ Milestones 🔗



📋 Requirements 🔗

Requirement	Description	Importance
Archer Score Lookup	Archer should be able to look up their score over time	HIGH
Archer Score Entry	Archer should have the access to enter their scores via hand-held device	HIGH
Archer Entry (by recorder)	The recorder should be able to add new archers to the database with details such as birth-date, gender, archery-vic-number, and their preferred default-equipment	HIGH
New Round (by recorder)	The recorder should be able to enter in new rounds that archers will shoot.	HIGH
Check if round in competition	The recorder should be able to check if a round counts towards a competition.	HIGH
Equipment check per round (by recorder)	The recorder should be able to add the equipment used by an archer in a round.	HIGH
New competition (by recorder)	Recorder to have the capability to add new competitions that rounds can be a part of.	HIGH
Round & Equivalent round lookup	Archers should be able to look up round definitions and find equivalent rounds, considering time-dependent changes by Archery Australia.	HIGH
Personal and Club best records Lookup	Archers must be able to view their personal best scores and club records, sorted by	HIGH

	category, date, and round type.	
Filter by club competition	Capability to filter scores by club competition	HIGH
Look up definitions of a range	Find details of a range that was shot, or a range to be shot. This includes distance, number of ends per distance, target face size etc.	HIGH
Look up definitions of rounds	Archers will need to get a 'template' round to fill in before they shoot a round. It should include input fields for all details to be recorded.	HIGH
Straightforward SQL & efficiency	The database should be designed in such a way that SQL commands to retrieve and store information are not prone to error, and are straightforward. The database should be also be able to handle multiple archers and recorders retrieving and updating records simultaneously.	MEDIUM

?

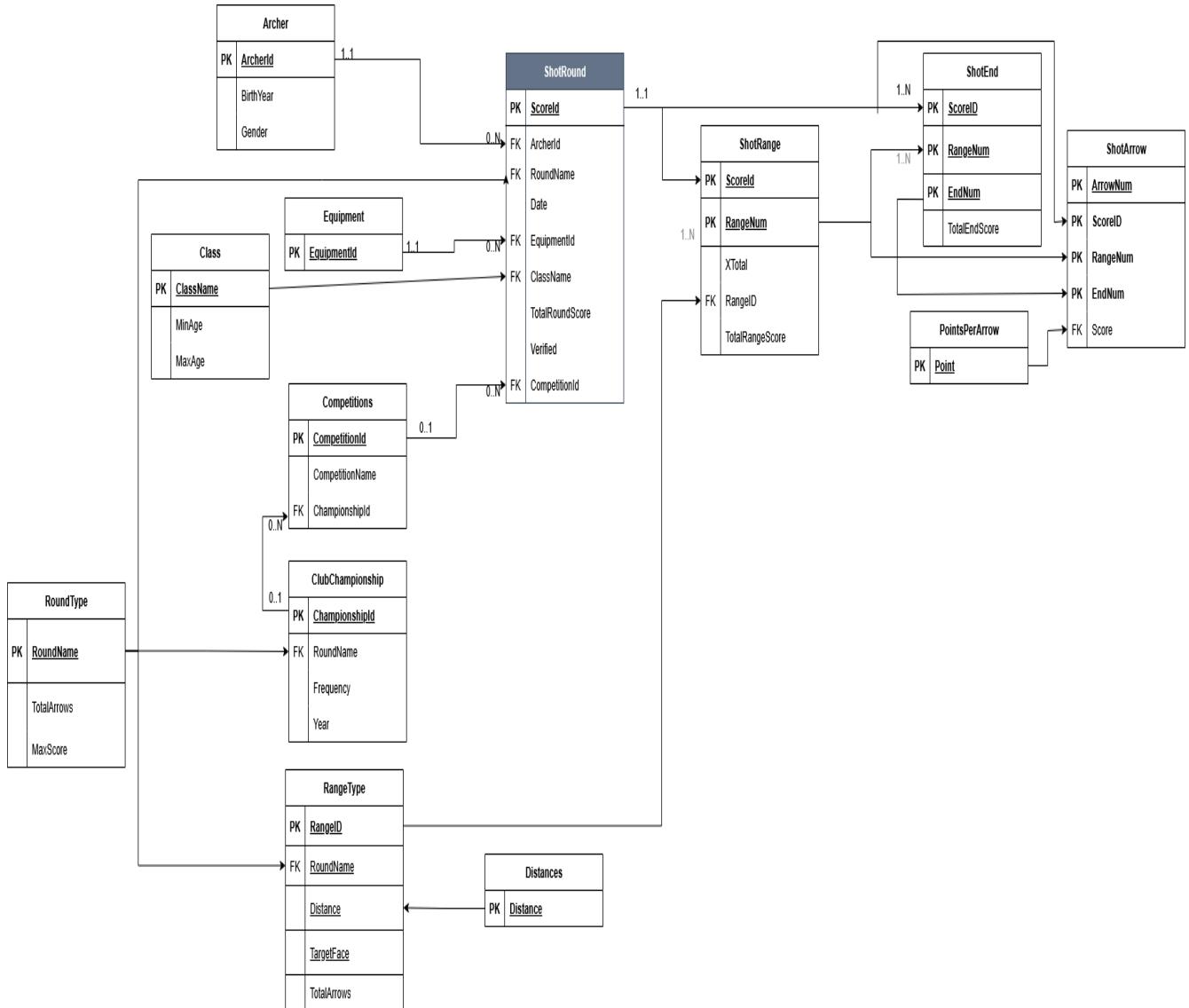
Open Questions [🔗](#)

Question	Answer
How will we ensure that data is secure?	We will only ask for non-sensitive data from users. As well as this, we will sanitize any data being inputted to prevent SQL injections.
How will Archers and Recorders get used to the new way of inputting scores?	We will make any frontend applications easy to use and understand, as well as making inputting scoring as similar to traditional paper sheets as is practical.
How will Archers ensure that their scores are inputted correctly?	Recorders will have a chance to review scores before they are confirmed in the database. This will ensure that there are no discrepancies in the scoring.

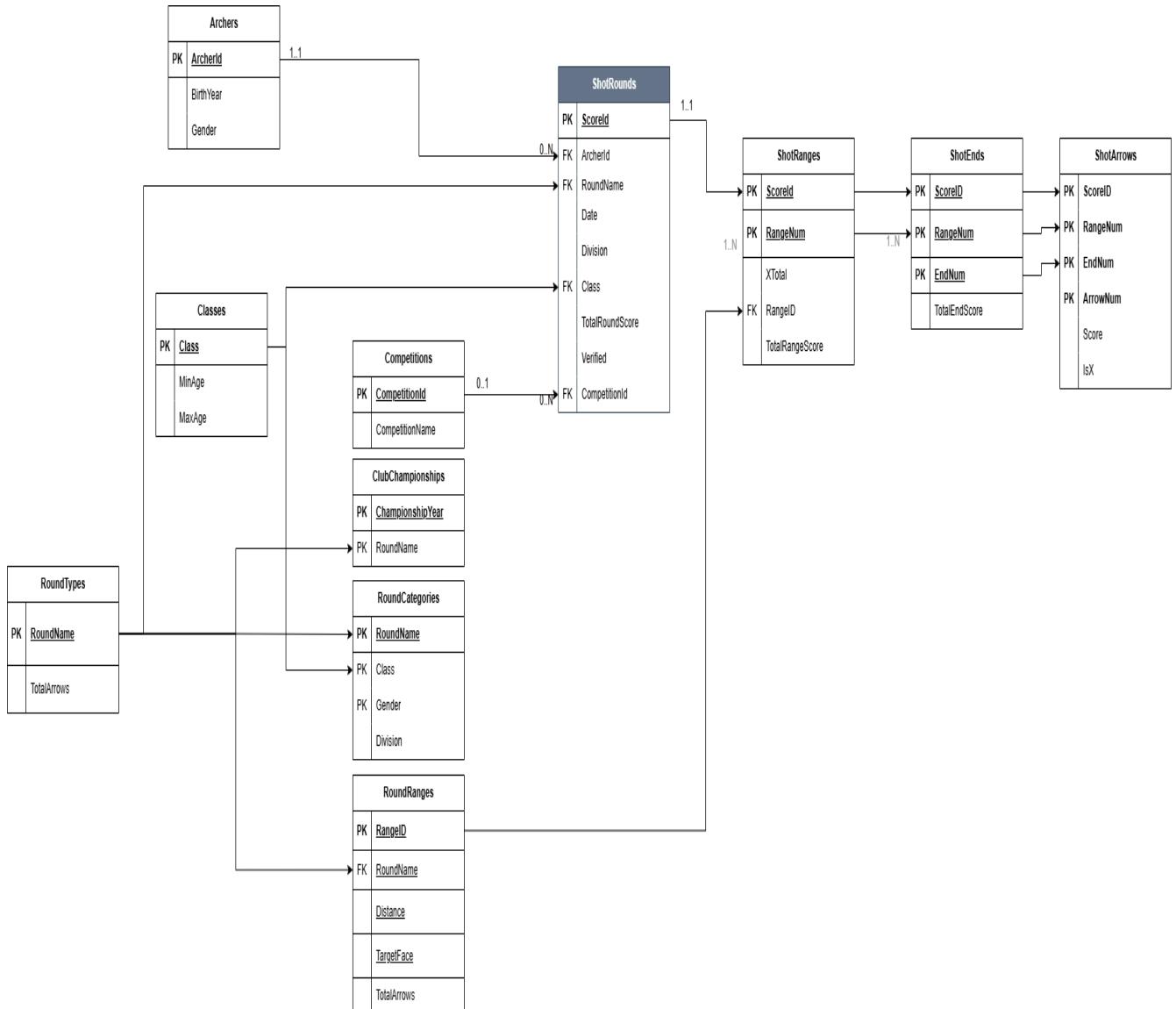
⚠ Out of Scope [🔗](#)

- Editing scores once they have been added to the database
- Removal of Archer IDs or scores from the database

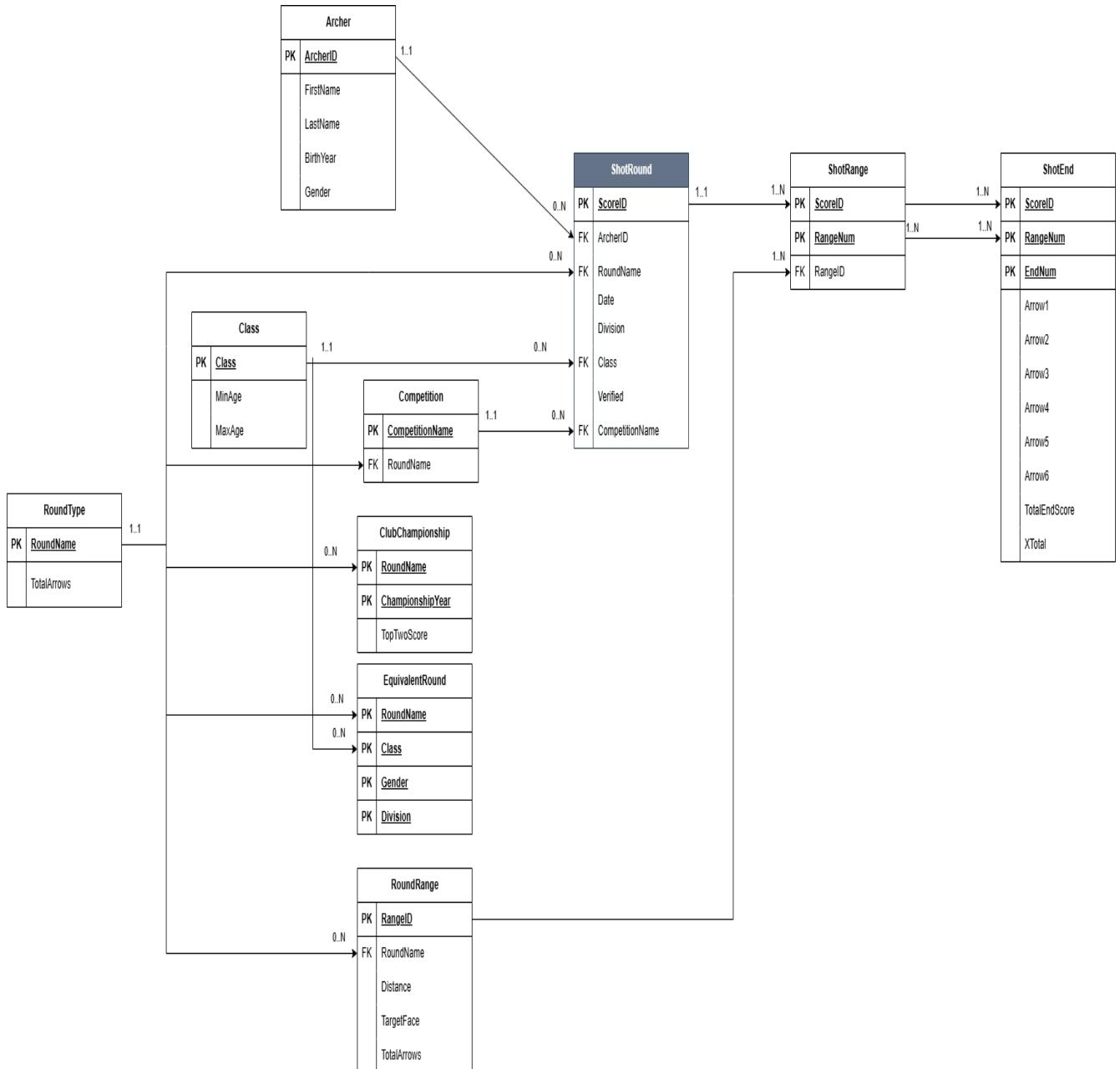
Draft 1 Entity Relationship Diagram



Draft 2 Entity Relationship Diagram



Final Entity Relationship Diagram



Draft 1 Create Table Commands

ARCHER (MATT)

```
CREATE table Archer (
    ArcherID VARCHAR(50) not null,
    BirthYear YEAR not null,
    Gender ENUM('Male', 'Female', 'Other') not null
);
```

CLASS (MATT)

```
CREATE table Class (
    ClassName VARCHAR(255) not null,
    MinAge INT not null,
    MaxAge INT not null
);
```

EQUIPMENT (MATT)

```
CREATE table Equipment (
    EquipmentID VARCHAR(50) not null,
    PRIMARY KEY (EquipmentID)
);
```

ROUNDTYPE (AIDEN)

```
CREATE TABLE RoundType(
    RoundName VARCHAR(255) not null,
    TotalArrows SMALLINT not null,
    MaxScore SMALLINT not null,
    PRIMARY KEY (RoundName)
);
```

COMPETITIONS (AIDEN)

```
CREATE TABLE Competitions(
    CompetitionID SMALLINT not null,
    CompetitionName VARCHAR(255) not null,
    PRIMARY KEY (CompetitionID)
);
```

CLUBCHAMPIONSHIP (AIDEN)

```
CREATE TABLE ClubChampionships(
    ChampionshipYear YEAR not null,
    RoundName VARCHAR(255) not null,
    PRIMARY KEY (ChampionshipYear, RoundName),
    FOREIGN KEY (RoundName) REFERENCES RoundType(RoundName)
);
```

RANGETYPE (JOHN)

```
CREATE TABLE RangeType (
    RangeTypeID SMALLINT not null,
    TargetFace SMALLINT not null,
    TotalArrows SMALLINT not null,
    DistanceID TINYINT not null,
    PRIMARY KEY (RangeTypeID),
    FOREIGN KEY (RoundName) REFERENCES RoundType(RoundName),
    FOREIGN KEY (DistanceID) REFERENCES Distances(DistanceID)
);
```

DISTANCES (JOHN)

```
CREATE TABLE Distances (
    DistanceID TINYINT not null,
    PRIMARY KEY(DistanceID)
);
```

SHOTROUND (DON)

```
CREATE TABLE ShotRound (
    ScoreID VARCHAR(50) not null,
    ArcherID VARCHAR(50) not null,
    RoundName VARCHAR(255) not null,
    Date DATE not null,
    Time TIME not null,
    EquipmentID VARCHAR(50) not null,
    ClassName VARCHAR(255) not null,
    TotalRoundScore INT not null,
    Verified BOOL not null,
    CompetitionID SMALLINT,
    PRIMARY KEY (ScoreID),
    FOREIGN KEY (ArcherID) REFERENCES Archer(ArcherID),
    FOREIGN KEY (RoundName) REFERENCES RoundType(RoundName),
    FOREIGN KEY (EquipmentID) REFERENCES Equipment(EquipmentID),
    FOREIGN KEY (ClassName) REFERENCES Class(ClassName),
    FOREIGN KEY (CompetitionID) References Competitions(CompetitionID)
);
```

SHOTRANGE (DON)

```

CREATE TABLE ShotRange (
    ScoreID VARCHAR(50) not null,
    RangeNum TINYINT not null,
    XTotal TINYINT not null,
    RangeTypeID SMALLINT not null,
    TotalRangeScore SMALLINT not null,
    PRIMARY KEY (ScoreID, RangeNum),
    FOREIGN KEY (ScoreID) REFERENCES ShotRound(ScoreID),
    FOREIGN KEY (RangeTypeID) REFERENCES RangeType(RangeTypeID)
);

```

SHOTEND (PANDU)

```

CREATE TABLE ShotEnd (
    ScoreID VARCHAR(50) not null,
    RangeNum TINYINT not null,
    EndNum TINYINT not null,
    TotalEndScore TINYINT not null,
    PRIMARY KEY (ScoreID, RangeNum, EndNum),
    FOREIGN KEY (RangeNum, ScoreID) REFERENCES ShotRange(RangeNum, ScoreID)
);

```

SHOTARROW (PANDU)

```

CREATE TABLE ShotArrow (
    ArrowNum TINYINT not null,
    ScoreID VARCHAR(50) not null,
    RangeNum TINYINT not null,
    EndNum TINYINT not null,
    Score CHAR(2) not null,
    PRIMARY KEY (ArrowNum, ScoreID, RangeNum, EndNum),
    FOREIGN KEY (EndNum, RangeNum, ScoreID) REFERENCES ShotEnd(EndNum, RangeNum, ScoreID),
    FOREIGN KEY (Score) REFERENCES PointsPerArrow(Score),
);

```

POINTSPERARROW (PANDU)

```

CREATE TABLE PointsPerArrow (
    Point CHARACTER(2) not null,
    PRIMARY KEY (Point)
);

```

Command ready-to-go (doesnt work yet):

```
1 /* Added primary keys to Archer and Class tables */
2 CREATE table Archer (
3     ArcherID VARCHAR(50) not null,
4     BirthYear YEAR not null,
5     Gender ENUM('Male', 'Female', 'Other') not null,
6     PRIMARY KEY (ArcherID)
7 );
8
9 CREATE table Class (
10    ClassName VARCHAR(255) not null,
11    MinAge INT not null,
12    MaxAge INT not null,
13    PRIMARY KEY (ClassName)
14 );
15
16 CREATE table Equipment (
17    EquipmentID VARCHAR(50) not null,
18    PRIMARY KEY (EquipmentID)
19 );
20
21 CREATE TABLE RoundType(
22     RoundName VARCHAR(255) not null,
23     TotalArrows SMALLINT not null,
24     MaxScore SMALLINT not null,
25     PRIMARY KEY (RoundName)
26 );
27
28 CREATE TABLE Distances (
29     DistanceID TINYINT not null,
30     PRIMARY KEY(DistanceID)
31 );
32
33 CREATE TABLE PointsPerArrow (
34     Point CHAR(2) not null,
35     PRIMARY KEY (Point)
36 );
37
38 CREATE TABLE RangeType (
39     RangeTypeID SMALLINT not null,
40     RoundName VARCHAR(255) not null,
41     TargetFace SMALLINT not null,
42     TotalArrows SMALLINT not null,
43     DistanceID TINYINT not null,
44     PRIMARY KEY (RangeTypeID),
45     FOREIGN KEY (RoundName) REFERENCES RoundType(RoundName),
46     FOREIGN KEY (DistanceID) REFERENCES Distances(DistanceID)
47 );
48
49 CREATE TABLE ClubChampionships(
50     ChampionshipYear YEAR not null,
51     RoundName VARCHAR(255) not null,
52     PRIMARY KEY (ChampionshipYear),
53     FOREIGN KEY (RoundName) REFERENCES RoundType(RoundName) -- changed
```

```

54 );
55
56 /* changed this table */
57 CREATE TABLE Competitions(
58     CompetitionID SMALLINT not null,
59     CompetitionName VARCHAR(255) not null,
60     PRIMARY KEY (CompetitionID)
61 );
62
63 CREATE TABLE ShotRound (
64     ScoreID VARCHAR(50) not null,
65     ArcherID VARCHAR(50) not null,
66     RoundName VARCHAR(255) not null,
67     Date DATE not null,
68     Time TIME not null,
69     EquipmentID VARCHAR(50) not null,
70     ClassName VARCHAR(255) not null,
71     TotalRoundScore INT not null,
72     Verified BOOL not null,
73     CompetitionID SMALLINT not null,
74     PRIMARY KEY (ScoreID),
75     FOREIGN KEY (ArcherID) REFERENCES Archer(ArcherID),
76     FOREIGN KEY (RoundName) REFERENCES RoundType(RoundName),
77     FOREIGN KEY (EquipmentID) REFERENCES Equipment(EquipmentID),
78     FOREIGN KEY (ClassName) REFERENCES Class(ClassName),
79     FOREIGN KEY (CompetitionID) REFERENCES Competitions(CompetitionID)
80 );
81
82 CREATE TABLE ShotRange (
83     ScoreID VARCHAR(50) not null,
84     RangeNum TINYINT not null,
85     XTotal TINYINT not null,
86     RangeTypeID SMALLINT not null,
87     TotalRangeScore SMALLINT not null,
88     PRIMARY KEY (ScoreID, RangeNum),
89     FOREIGN KEY (ScoreID) REFERENCES ShotRound(ScoreID),
90     FOREIGN KEY (RangeTypeID) REFERENCES RangeType(RangeTypeID)
91 );
92
93 /* idk why this aint working*/
94 CREATE TABLE ShotEnd (
95     ScoreID VARCHAR(50) not null,
96     RangeNum TINYINT not null,
97     EndNum TINYINT not null,
98     TotalEndScore TINYINT not null,
99     PRIMARY KEY (ScoreID, RangeNum, EndNum),
100    FOREIGN KEY (RangeNum, ScoreID) REFERENCES ShotRange(RangeNum, ScoreID)
101 );
102
103 /* ths should work once ShotEnd works */
104 CREATE TABLE ShotArrow (
105     ArrowNum TINYINT not null,
106     ScoreID VARCHAR(50) not null,
107     RangeNum TINYINT not null,
108     EndNum TINYINT not null,
109     Score CHAR(2) not null,
110     PRIMARY KEY (ArrowNum, ScoreID, RangeNum, EndNum),
111     FOREIGN KEY (EndNum, RangeNum, ScoreID) REFERENCES ShotEnd(EndNum, RangeNum, ScoreID),

```

```
112     FOREIGN KEY (Score) REFERENCES PointsPerArrow(Score)
113 );
114
115 INSERT INTO PointsPerArrow(`Point`) VALUES ('M');
116 INSERT INTO PointsPerArrow(`Point`) VALUES ('1');
117 INSERT INTO PointsPerArrow(`Point`) VALUES ('2');
118 INSERT INTO PointsPerArrow(`Point`) VALUES ('3');
119 INSERT INTO PointsPerArrow(`Point`) VALUES ('4');
120 INSERT INTO PointsPerArrow(`Point`) VALUES ('5');
121 INSERT INTO PointsPerArrow(`Point`) VALUES ('6');
122 INSERT INTO PointsPerArrow(`Point`) VALUES ('7');
123 INSERT INTO PointsPerArrow(`Point`) VALUES ('8');
124 INSERT INTO PointsPerArrow(`Point`) VALUES ('9');
125 INSERT INTO PointsPerArrow(`Point`) VALUES ('10');
126 INSERT INTO PointsPerArrow(`Point`) VALUES ('X');
127
128 INSERT INTO Equipment(`EquipmentID`) VALUES ('Recurve');
129 INSERT INTO Equipment(`EquipmentID`) VALUES ('Compound');
130 INSERT INTO Equipment(`EquipmentID`) VALUES ('Recurve Barebow');
131 INSERT INTO Equipment(`EquipmentID`) VALUES ('Compound Barebow');
132 INSERT INTO Equipment(`EquipmentID`) VALUES ('LongBow');
133
134 INSERT INTO Distance('Distance') VALUES ('10');
135 INSERT INTO Distance('Distance') VALUES ('20');
136 INSERT INTO Distance('Distance') VALUES ('30');
137 INSERT INTO Distance('Distance') VALUES ('40');
138 INSERT INTO Distance('Distance') VALUES ('50');
139 INSERT INTO Distance('Distance') VALUES ('60');
140 INSERT INTO Distance('Distance') VALUES ('70');
141 INSERT INTO Distance('Distance') VALUES ('90');
```

Draft 2 Create Table Commands

```
1  /* To drop existing tables */
2  DROP TABLE IF EXISTS
3      ShotArrows,
4      Scores,
5      ShotEnds,
6      ShotRanges,
7      ShotRounds,
8      Competitions,
9      ClubChampionships,
10     RoundRanges,
11     Distances,
12     RoundCategories,
13     RoundTypes,
14     Classes,
15     Archers
16 ;
17
18 CREATE table Archers (
19     ArcherID VARCHAR(11) not null,
20     BirthYear YEAR not null,
21     Gender ENUM('Male', 'Female') not null,
22     PRIMARY KEY (ArcherID)
23 );
24
25 CREATE table Classes (
26     Class VARCHAR(255) not null,
27     MinAge TINYINT,
28     MaxAge TINYINT,
29     PRIMARY KEY (Class)
30 );
31
32 CREATE TABLE RoundTypes (
33     RoundName VARCHAR(255) not null,
34     TotalArrows SMALLINT not null,
35     MaxScore SMALLINT not null,
36     PRIMARY KEY (RoundName)
37 );
38
39 CREATE TABLE RoundCategories (
40     RoundName VARCHAR(255) not null,
41     Class VARCHAR(255) not null,
42     Gender ENUM('Male', 'Female') not null,
43     Division ENUM('Recurve', 'Compound', 'Recurve Barebow', 'Compound Barebow', 'Longbow') not null,
44     PRIMARY KEY(RoundName, Class, Gender),
45     FOREIGN KEY (RoundName) REFERENCES RoundTypes(RoundName),
46     FOREIGN KEY (Class) REFERENCES Classes(Class)
47 );
48
49 CREATE TABLE RoundRanges (
50     RangeID INT not null AUTO_INCREMENT,
51     RoundName VARCHAR(255) not null,
52     TargetFace ENUM("80", "120") not null,
53     TotalArrows SMALLINT not null,
```

```

54     Distance ENUM('10', '20', '30', '40', '50', '60', '70', '90') not null,
55     PRIMARY KEY (RangeID),
56     FOREIGN KEY (RoundName) REFERENCES RoundTypes(RoundName),
57 );
58
59 CREATE TABLE ClubChampionships (
60     ChampionshipYear YEAR not null,
61     RoundName VARCHAR(255) not null,
62     PRIMARY KEY (ChampionshipYear, RoundName),
63     FOREIGN KEY (RoundName) REFERENCES RoundTypes(RoundName)
64 );
65
66 CREATE TABLE Competitions (
67     CompetitionID INT not null AUTO_INCREMENT,
68     CompetitionName VARCHAR(255) not null,
69     PRIMARY KEY (CompetitionID)
70 );
71
72 CREATE TABLE ShotRounds (
73     ScoreID INT not null AUTO_INCREMENT,
74     ArcherID VARCHAR(11) not null,
75     RoundName VARCHAR(255) not null,
76     Date DATE not null,
77     Division ENUM('Recurve', 'Compound', 'Recurve Barebow', 'Compound Barebow', 'Longbow') not null,
78     Class VARCHAR(255) not null,
79     TotalRoundScore SMALLINT not null,
80     Verified BOOLEAN not null,
81     CompetitionID INT,
82     PRIMARY KEY (ScoreID),
83     FOREIGN KEY (ArcherID) REFERENCES Archers(ArcherID),
84     FOREIGN KEY (RoundName) REFERENCES RoundTypes(RoundName),
85     FOREIGN KEY (Class) REFERENCES Classes(Class),
86     FOREIGN KEY (CompetitionID) REFERENCES Competitions(CompetitionID)
87 );
88
89 CREATE TABLE ShotRanges (
90     ScoreID INT not null,
91     RangeNum TINYINT not null,
92     XTotal TINYINT not null,
93     RangeID INT not null,
94     TotalRangeScore SMALLINT not null,
95     PRIMARY KEY (ScoreID, RangeNum),
96     FOREIGN KEY (ScoreID) REFERENCES ShotRounds(ScoreID),
97     FOREIGN KEY (RangeID) REFERENCES RoundRanges(RangeID)
98 );
99
100 CREATE TABLE ShotEnds (
101     ScoreID INT not null,
102     RangeNum TINYINT not null,
103     EndNum TINYINT not null,
104     TotalEndScore TINYINT not null,
105     PRIMARY KEY (ScoreID, RangeNum, EndNum),
106     FOREIGN KEY (ScoreID, RangeNum) REFERENCES ShotRanges(ScoreID, RangeNum)
107 );
108
109 CREATE TABLE ShotArrows (
110     ScoreID INT not null,
111     RangeNum TINYINT not null,

```

```

112     EndNum TINYINT not null,
113     ArrowNum TINYINT not null,
114     Score ENUM('M', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'X') not null,
115     PRIMARY KEY (ArrowNum, ScoreID, RangeNum, EndNum),
116     FOREIGN KEY (ScoreID, RangeNum, EndNum) REFERENCES ShotEnds(ScoreID, RangeNum, EndNum),
117 );
118
119 INSERT INTO RoundTypes(RoundName, TotalArrows, MaxScore)
120 VALUES
121 ("WA90/1440", 144, 1440),
122 ("WA70/1440", 144, 1440),
123 ("WA60/1440", 144, 1440),
124 ("AA50/1440", 144, 1440),
125 ("AA40/1440", 144, 1440),
126 ("Long Sydney", 120, 1200),
127 ("Sydney", 120, 1200),
128 ("Long Brisbane", 120, 1200),
129 ("Brisbane", 120, 1200),
130 ("Adelaide", 120, 1200),
131 ("Short Adelaide", 120, 1200),
132 ("Hobart", 90, 900),
133 ("Perth", 90, 900),
134 ("Canberra", 90, 900),
135 ("Short Canberra", 90, 900),
136 ("Junior Canberra", 90, 900),
137 ("Mini Canberra", 90, 900),
138 ("Grange", 90, 900),
139 ("Melbourne", 90, 900),
140 ("Darwin", 90, 900),
141 ("Geelong", 90, 900),
142 ("Newcastle", 90, 900),
143 ("Holt", 90, 900),
144 ("Samford", 90, 900),
145 ("Drake", 90, 900),
146 ("Wollongong", 72, 720),
147 ("Townsville", 72, 720),
148 ("Launceston", 72, 720),
149 ("WA70/720", 72, 720),
150 ("WA60/720", 72, 720),
151 ("WA50/720", 72, 720);
152
153 INSERT INTO RoundRanges(RoundName, Distance, TargetFace, TotalArrows)
154 VALUES
155 ("WA90/1440", "90", "120", 36),
156 ("WA90/1440", "70", "120", 36),
157 ("WA90/1440", "50", "80", 36),
158 ("WA90/1440", "30", "80", 36),
159 ("WA70/1440", "70", "120", 36),
160 ("WA70/1440", "60", "120", 36),
161 ("WA70/1440", "50", "80", 36),
162 ("WA70/1440", "30", "80", 36),
163 ("WA60/1440", "60", "120", 36),
164 ("WA60/1440", "50", "120", 36),
165 ("WA60/1440", "40", "80", 36),
166 ("WA60/1440", "30", "80", 36),
167 ("AA50/1440", "50", "120", 36),
168 ("AA50/1440", "40", "120", 36),
169 ("AA50/1440", "30", "80", 36),

```

```

170 ("AA50/1440", "20", "80", 36),
171 ("AA40/1440", "40", "120", 36),
172 ("AA40/1440", "30", "120", 36),
173 ("AA40/1440", "30", "80", 36),
174 ("AA40/1440", "20", "80", 36),
175 ("Long Sydney", "90", "120", 30),
176 ("Long Sydney", "70", "120", 30),
177 ("Long Sydney", "60", "120", 30),
178 ("Long Sydney", "50", "120", 30),
179 ("Sydney", "70", "120", 30),
180 ("Sydney", "60", "120", 30),
181 ("Sydney", "50", "120", 30),
182 ("Sydney", "40", "120", 30),
183 ("Long Brisbane", "90", "120", 30),
184 ("Long Brisbane", "70", "120", 30),
185 ("Long Brisbane", "60", "80", 30),
186 ("Long Brisbane", "50", "80", 30),
187 ("Brisbane", "70", "120", 30),
188 ("Brisbane", "60", "120", 30),
189 ("Brisbane", "50", "80", 30),
190 ("Brisbane", "40", "80", 30),
191 ("Adelaide", "60", "120", 30),
192 ("Adelaide", "50", "120", 30),
193 ("Adelaide", "40", "80", 30),
194 ("Adelaide", "30", "80", 30),
195 ("Short Adelaide", "50", "120", 30),
196 ("Short Adelaide", "40", "120", 30),
197 ("Short Adelaide", "30", "80", 30),
198 ("Short Adelaide", "20", "80", 30),
199 ("Hobart", "90", "120", 30),
200 ("Hobart", "70", "120", 30),
201 ("Hobart", "50", "120", 30),
202 ("Perth", "70", "120", 30),
203 ("Perth", "60", "120", 30),
204 ("Perth", "50", "120", 30),
205 ("Canberra", "60", "120", 30),
206 ("Canberra", "50", "120", 30),
207 ("Canberra", "40", "120", 30),
208 ("Short Canberra", "50", "120", 30),
209 ("Short Canberra", "40", "120", 30),
210 ("Short Canberra", "30", "120", 30),
211 ("Junior Canberra", "40", "120", 30),
212 ("Junior Canberra", "30", "120", 30),
213 ("Junior Canberra", "20", "120", 30),
214 ("Mini Canberra", "30", "120", 30),
215 ("Mini Canberra", "20", "120", 30),
216 ("Mini Canberra", "10", "120", 30),
217 ("Grange", "60", "120", 90),
218 ("Melbourne", "50", "120", 90),
219 ("Darwin", "40", "120", 90),
220 ("Geelong", "30", "120", 90),
221 ("Newcastle", "20", "120", 90),
222 ("Holt", "50", "80", 90),
223 ("Samford", "40", "80", 90),
224 ("Drake", "30", "80", 90),
225 ("Wollongong", "90", "120", 36),
226 ("Wollongong", "70", "120", 36),
227 ("Townsville", "70", "120", 36),

```

```
228 ("Townsville", "60", "120", 36),  
229 ("Launceston", "50", "80", 36),  
230 ("Launceston", "30", "80", 36),  
231 ("WA70/720", "70", "120", 72),  
232 ("WA60/720", "60", "120", 72),  
233 ("WA50/720", "50", "80", 72);  
234  
235 INSERT INTO Classes(Class, MinAge, MaxAge)  
236 VALUES  
237 ("Under 14", NULL, 13),  
238 ("Under 16", NULL, 15),  
239 ("Under 18", NULL, 17),  
240 ("Under 12", NULL, 20),  
241 ("Open", NULL, NULL),  
242 ("50+", 50, NULL),  
243 ("60+", 60, NULL),  
244 ("70+", 70, NULL);
```

Final Create Table Commands

```
1 /* To drop existing tables */
2 DROP TABLE IF EXISTS
3     ShotEnd,
4     ShotRange,
5     ShotRound,
6     Competition,
7     ClubChampionship,
8     RoundRange,
9     EquivalentRound,
10    RoundType,
11    Class,
12    Archer
13 ;
14
15 CREATE table Archer (
16     ArcherID VARCHAR(11) not null,
17     FirstName VARCHAR(255) not null,
18     LastName VARCHAR(255) not null,
19     BirthYear YEAR not null,
20     Gender ENUM('Male', 'Female') not null,
21     PRIMARY KEY (ArcherID)
22 );
23
24 CREATE TABLE Class (
25     Class VARCHAR(255) not null,
26     MinAge TINYINT,
27     MaxAge TINYINT,
28     PRIMARY KEY (Class)
29 );
30
31 CREATE TABLE RoundType (
32     RoundName VARCHAR(255) not null,
33     TotalArrows SMALLINT not null,
34     PRIMARY KEY (RoundName)
35 );
36
37 CREATE TABLE EquivalentRound (
38     RoundName VARCHAR(255) not null,
39     Class VARCHAR(255) not null,
40     Gender ENUM('Male', 'Female') not null,
41     Division ENUM('Recurve', 'Compound', 'Recurve Barebow', 'Compound Barebow', 'Longbow') not null,
42     PRIMARY KEY(RoundName, Class, Gender, Division),
43     FOREIGN KEY (RoundName) REFERENCES RoundType(RoundName),
44     FOREIGN KEY (Class) REFERENCES Class(Class)
45 );
46
47 CREATE TABLE RoundRange (
48     RangeID INT not null AUTO_INCREMENT,
49     RoundName VARCHAR(255) not null,
50     Distance ENUM('10', '20', '30', '40', '50', '60', '70', '90') not null,
51     TargetFace ENUM("80", "120") not null,
52     TotalArrows SMALLINT not null,
53     PRIMARY KEY (RangeID),
54     FOREIGN KEY (RoundName) REFERENCES RoundType(RoundName)
55 );
56
57 CREATE TABLE ClubChampionship (
58     RoundName VARCHAR(255) not null,
59     ChampionshipYear YEAR not null,
60     TopTwoScore BOOL not null,
61     PRIMARY KEY (ChampionshipYear, RoundName),
62     FOREIGN KEY (RoundName) REFERENCES RoundType(RoundName)
63 );
64
65 CREATE TABLE Competition (
66     CompetitionName VARCHAR(255) not null,
67     RoundName VARCHAR(255) not null,
68     PRIMARY KEY (CompetitionName),
69     FOREIGN KEY (RoundName) REFERENCES RoundType(RoundName)
70 );
71
72 CREATE TABLE ShotRound (
73     ScoreID INT not null AUTO_INCREMENT,
74     ArcherID VARCHAR(11) not null,
75     RoundName VARCHAR(255) not null,
76     Date DATE not null,
77     Division ENUM('Recurve', 'Compound', 'Recurve Barebow', 'Compound Barebow', 'Longbow') not null,
```

```

78     Class VARCHAR(255) not null,
79     Verified BOOLEAN not null,
80     CompetitionName VARCHAR(255),
81     PRIMARY KEY (ScoreID),
82     FOREIGN KEY (ArcherID) REFERENCES Archer(ArcherID),
83     FOREIGN KEY (RoundName) REFERENCES RoundType(RoundName),
84     FOREIGN KEY (Class) REFERENCES Class(Class),
85     FOREIGN KEY (CompetitionName) REFERENCES Competition(CompetitionName)
86 );
87
88 CREATE TABLE ShotRange (
89     ScoreID INT not null,
90     RangeNum TINYINT not null,
91     RangeID INT not null,
92     PRIMARY KEY (ScoreID, RangeNum),
93     FOREIGN KEY (ScoreID) REFERENCES ShotRound(ScoreID),
94     FOREIGN KEY (RangeID) REFERENCES RoundRange(RangeID)
95 );
96
97 CREATE TABLE ShotEnd (
98     ScoreID INT not null,
99     RangeNum TINYINT not null,
100    EndNum TINYINT not null,
101    Arrow1 ENUM('M', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'X') not null,
102    Arrow2 ENUM('M', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'X') not null,
103    Arrow3 ENUM('M', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'X') not null,
104    Arrow4 ENUM('M', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'X') not null,
105    Arrow5 ENUM('M', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'X') not null,
106    Arrow6 ENUM('M', '1', '2', '3', '4', '5', '6', '7', '8', '9', '10', 'X') not null,
107    TotalEndScore TINYINT not null,
108    XTotal TINYINT not null,
109    PRIMARY KEY (ScoreID, RangeNum, EndNum),
110    FOREIGN KEY (ScoreID, RangeNum) REFERENCES ShotRange(ScoreID, RangeNum)
111 );

```

Dummy Data

Where reasonable, Dummy Data was created manually on tables where there is less data required. The following shows the tables where dummy data was created manually and one example of data entry into the table:

RoundType (INSERT INTO RoundType(RoundName, TotalArrows) VALUES ("WA90/1440", 144);)

RoundRange (INSERT INTO RoundRange(RoundName, Distance, TargetFace, TotalArrows) VALUES ("WA90/1440", "90", "120", 36);)

Class (INSERT INTO Class(Class, MinAge, MaxAge) VALUES ("Under 14", NULL, 13);)

EquivalentRound (INSERT INTO EquivalentRound(RoundName, Class, Gender, Division) VALUES ("WA90/1440", "Open", "Male", "Recurve");)

Competition (INSERT INTO Competition (CompetitionName, RoundName) VALUES ("The Donovan Cup", "WA70/720");)

ClubChampionship (INSERT INTO ClubChampionship(RoundName, ChampionshipYear, TopTwoScore) VALUES ("WA90/1440", 2024, 1);)

For this project, ChatGPT 4o was used to generate dummy data. Generative AI tools like ChatGPT provide the ability to generate meaningful data with quite strict constraints at a high enough level of accuracy to be acceptable as dummy data.

The following prompt as well as all create table commands and all manually created dummy data was used to generate the data:

I need you to generate dummy data from the database specifications listed above. Here are the constraints.

I need you to create data for the following tables: Archers, ShotRounds, ShotRanges, ShotEnds, ShotArrows, Competitions. Do not create data for any other tables. Where there are foreign key constraints you should refer to the INSERT INTO statements provided to you above.

You should create 500 rows of data for the archers table and 1500 rows of data for the ShotRounds table. Date in ShotRounds should be a random date between 1/1/2024 and 25/5/2025 Verified in ShotRounds should be TRUE always.

There should be between 1 and 3 rows of data in ShotRanges for each row of data in ShotRounds. Each ScoreID from ShotRounds should not have more than 3, or less than 1 row of data in ShotRanges attached to it. RangeNum in ShotRanges should be the number of the range being shot (as in it should only be 1, 2 or 3). The field RangeID in ShotRanges should reference a value from the field in RoundRanges. This is an auto incremented value so is not shown in the INSERT INTO statements I have given you, but it should be a value associated with the RoundName in RoundRanges linked to the corresponding ScoreID field. This means the first row of data in the INSERT INTO statement would have RangeID = 1, the second RangeID = 2, and so forth.

There should be between 6 and 12 rows of data in ShotEnds for each row of data in ShotRanges. Each combination of ScoreID and RangeNum from ShotRanges should not have more than 12, or less than 6 rows of data in ShotEnds attached to it. Each instance of EndNum in ShotEnds should be the number of the end being shot (as in it should only be 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12). TotalEndScore in ShotEnds should be the sum of the values in Arrow1, Arrow2, Arrow3, Arrow4, Arrow5 and Arrow6, where 'M' would equal 0 and 'X' would equal 10 XTotal in ShotEnds should be the sum of all the values that are 'X' in Arrow1, Arrow2, Arrow3, Arrow4, Arrow5, and Arrow6.

FirstName and LastName in Archers should be a random name generated by you such as John Smith. ArcherID in Archer should be a random, unique 10 digit number.

The following shows each table where data was created using Generative AI and an example of the data created:

Archer (INSERT INTO Archer (ArcherID, FirstName, LastName, BirthYear, Gender) VALUES ('8643795781', 'Lauren', 'Johnson', 1986, 'Male');

ShotRound (INSERT INTO ShotRound (ScoreID, ArcherID, RoundName, Date, Division, Class, Verified, CompetitionName) VALUES (9, '8793068400', 'Darwin', '2024-10-24', 'Longbow', 'Under 16', 1, NULL);)

ShotRange (INSERT INTO ShotRange (ScoreID, RangeNum, RangeID) VALUES (9,1,8);)

ShotEnd (INSERT INTO ShotEnd (ScoreID, RangeNum, EndNum, Arrow1, Arrow2, Arrow3, Arrow4, Arrow5, Arrow6, TotalEndScore, XTotal) VALUES (9,1,3,9,7,1,X,7,5,39,1);)

The following constraints were identified with the dummy data:

GPT 4o was not able to correctly link the RangeID to the actual RangeID associated with ShotRound. RangeID in ShotRange is a correct value and so passes the foreign key constraint, but it is often not an actual range that would be shot in the round.

NULL values were not effectively handled, and every round inputted into ShotRounds was associated with a competition. To combat this, all fields of CompetitionName were originally set to NULL, and then some rounds were manually associated with a CompetitionName.

Overall the data does provide a meaningful set that can be used to test different queries, bar a few minor discrepancies.

Indexing

Some searches that were identified as frequently used are the following:

- Search in ShotRound by ArcherID (SELECT * FROM ShotRound WHERE ArcherID = 'x';)
- Search in ShotRound by Date (SELECT * FROM ShotRound WHERE Date BETWEEN 'xxxx-xx-xx' AND 'xxxx-xx-xx';)
- Search in ShotRound by Division (SELECT * FROM ShotRound WHERE Division = 'x';)
- Search in ShotRound by Class (SELECT * From ShotRound WHERE Class = 'x';)
- Search in ShotRound by CompetitionName (SELECT * From ShotRound WHERE CompetitionName = 'x';)
- Search in ShotRange by ScoreID (SELECT * FROM ShotRange WHERE ScoreID = 'x';)
- Search in ShotEnd by ScoreID & RangeNum (SELECT * FROM ShotEnd WHERE ScoreID = 'x' AND RangeNum = 'x';)
- Search in RoundRange by RoundName (SELECT * From RoundRange WHERE RoundName = 'x';)

Many of these selected candidates are primary keys and as such are automatically indexed. Because of this our potential candidates for indexing are low. This leaves us with the following for potential indexing candidates:

- Search in ShotRound by Date (SELECT * FROM ShotRound WHERE Date BETWEEN 'xxxx-xx-xx' AND 'xxxx-xx-xx';)
- Search in ShotRound by Division (SELECT * FROM ShotRound WHERE Division = 'x';)

The indexes for these are created like such:

Archers viewing their scores by Date

```
CREATE INDEX index_shotround_date ON ShotRound(Date);
```

Archers viewing scores by Division

```
CREATE INDEX index_shotround_division ON ShotRound(Division);
```

Testing the Index for Date on ShotRound:

To test the index for Date on ShotRound, the following statements were run without the index 3 times to capture an average time taken, and then run again 3 times with the index created. Query cache was blocked using the SQL_NO_CACHE keyphrase in an attempt to achieve the most accurate results. ShotRound at the time of running these tests contains 1500 rows of dummy data:

Statement: SELECT SQL_NO_CACHE * FROM ShotRound WHERE Date BETWEEN '2024-01-01' AND '2025-05-25';

Results Without Index: 0.0077, 0.0057, 0.0107 seconds respectively; 0.0080 seconds average

Results With Index: 0.0065, 0.0093, 0.0023 seconds respectively; 0.0060 seconds average

Statement: SELECT SQL_NO_CACHE * FROM ShotRound WHERE Date BETWEEN '2024-01-01' AND '2024-02-01';

Results Without Index: 0.0025, 0.0028, 0.0028 seconds respectively; 0.0027 seconds average

Results With Index: 0.0023, 0.0029, 0.0022 seconds respectively; 0.0025 seconds average

Statement: SELECT SQL_NO_CACHE * FROM ShotRound WHERE Date BETWEEN '2024-03-01' AND '2024-08-01';

Results Without Index: 0.0041, 0.0031, 0.0024 seconds respectively; 0.0032 seconds average

Results With Index: 0.0025, 0.0025, 0.0025 seconds respectively; 0.0025 seconds average

Conclusion: On the largest set of data, statement 1, a 25% improvement in read speeds was noted. While the difference between 0.0080 and 0.0060 seconds is small, as this dataset grows a 25% difference in speed may become important. As well as this, there is estimated to be more read queries (archers checking their scores) than write queries (archers entering a round), so based on this evidence the index on Date in ShotRound should be kept.

Testing the Index for Division on ShotRound:

The same process that was used for Date was used to capture average time for Division.

Statement: SELECT SQL_NO_CACHE * FROM ShotRound WHERE Division = "Recurve";

Results Without Index: 0.0022, 0.0024, 0.0026 seconds respectively; 0.0024 seconds average

Results With Index: 0.0024, 0.0025, 0.0022 seconds respectively; 0.00236 seconds average

Statement: SELECT SQL_NO_CACHE * FROM ShotRound WHERE Division = "Compound" OR Division = "Longbow";

Results Without Index: 0.0023, 0.0036, 0.0024 seconds respectively; 0.0028 seconds average

Results With Index: 0.0035, 0.0028, 0.0025 seconds respectively; 0.0029 seconds average

Statement: SELECT SQL_NO_CACHE * FROM ShotRound WHERE Division = "Compound" OR Division = "Recurve Barebow" OR Division = "Compound Barebow";

Results Without Index: 0.0036, 0.0035, 0.0026 seconds respectively; 0.0032 seconds average

Results With Index: 0.0033, 0.0026, 0.0021 seconds respectively; 0.0027 seconds average

Conclusion: On the largest set of data, statement 3, a 15% improvement in read speeds was noted. However, improvements in speed using a single division or multiple divisions were minimal. We hypothesize that this is due to the small number of choices available from Division. We also presume it will be less likely that Archers will search for rounds based on Division compared to Date or other search factors such as the Archer ID. Due to this, we have chosen to discard this index to reduce storage and improve write speeds.

Business Logic and Assumptions

- All archer ages are treated as though the archer was born on the first day (January 1st) of their birth year.
- Calculation of archer age based on birth year is dealt with in the front end when necessary.
- This project focuses solely on target archery, which involves shooting outdoors on a level field at a standard sized target.
- It is assumed that an archer cannot play an individual end or range, and must play an entire round.
- Enumeration only used for gender.

Archer

- Each archer has an archer ID, a gender and a birth year.
- As the archer name is sensitive information, it will not be stored in the database.

Archer	
PK	<u>ArcherId</u>
	BirthYear
	Gender

Category

- Each round has categories, determining who can play in the round, and what equipment they must use based on their class.
- A category is based on class (age and gender) and division (the type of bow used).
- An archer may shoot in a harder category than what they are eligible for, but may not shoot in an easier category.
- Archers must compete within their own gender.

Class

- Class consists of the required gender and age range.
- In a table, it could include class name (PK), gender, minimum age and maximum age.
- For the purpose of this project, the “Class” table will only define the minimum and maximum age for a class, with gender being defined separately where necessary.
- The age brackets are:
 - Under 14
 - Under 16
 - Under 18

- Under 21
- Open
- 50+
- 60+
- 70+
- The rules for each age bracket are listed below (found at [box age classes 2025.pdf | Powered by Box](#)):
 - Under 14 (U14) can shoot as U14, U16, U18, U21 or Open
 - Under 16 (U16) can shoot as U16, U18, U21 or Open
 - Under 18 (U18) can shoot as U18, U21 or Open
 - Under 21 (U21) can shoot as U21 or Open
 - 50+ can shoot as 50+ or Open
 - 60+ can shoot as 60+, 50+ or Open
 - 70+ can shoot as 70+, 60+, 50+ or Open
- If the “MinAge” or “MaxAge” fields are NULL, it will mean that there is not a minimum or maximum age required to play in that class. For example, the “Open” class will have NULL “MinAge” and “MaxAge” fields.

Class	
PK	<u>ClassName</u>
	MinAge
	MaxAge

Division

- Division is based entirely on the type of bow being used
- For this project, it is assumed that the club recognises 5 division:
 - Recurve
 - Compound
 - Recurve Barebow
 - Compound Barebow
 - Longbow
- This could be recorded as an enumerator.

Arrow Score

- Depending on where an arrow lands after being shot, it can have a score of M (miss), 1, 2, 3 ... 8, 9, 10 or X (bullseye). This may be implemented as an enumerator.
- An X shares the same point value as a 10, but if two competing archers end up with the same score after a game, whoever has the most Xs recorded is considered the winner.

Ends

- An archer shoots exactly 6 arrows at a time before moving to collect them, and this collection of 6 arrows is called an “end”.
- During a game, all participating players shoot each end at the same time. Once all players have shot an end, they take turns shouting each of their arrow scores to one player who is designated as the “recorder”. The recorder writes down the individual arrow scores obtained by each player.
- While archers generally shout out their arrow scores in order of descending score, the recorded order of scores within an end in the database does not matter.

Range

- A range is a collection of ends shot in sequence.
- A range has a consistent distance, face size and number of ends.

Face Size

- The target being shot at can have a face size diameter of either 80cm or 120cm.
- This could be implemented as an enumerator.

Distance

- The distance being shot can be 10m, 20m, 30m, 40m, 50m, 60m, 70m or 90m.
- 80m can not be shot.
- This could be implemented as an enumerator

Round

- A round is a collection of ranges shot in sequence.
- A round may consist of a minimum of 6 ends (36 arrows).
- A round must have a date and category
- Unlike ranges which are arbitrary combinations of distance, face sizes and number of arrows, round types are legally defined with each one having an associated name (such as “WA90/1440” or “Sydney”). Therefore, “RoundTypes” should have its own table containing the round names. This may be best implemented as an enumerator.
- There should also be a “RoundRanges” table with round name (PK), distance (PK), face size (PK) and number of ends. This table would record the range types making up a round type. This will implement the table shown below:

Rounds (Schedule 9A)	Total Arrows	Number of Arrows at Each Distance								Possible Score	
		Distances (metres)									
		90	70	60	50	40	30	20	10		
WA90/1440	144	36+	36+		36*		36*			1440	
WA70/1440	144		36+	36+	36*		36*			1440	
WA60/1440	144			36+	36+	36*	36*			1440	
AA50/1440	144				36+	36+	36*	36*		1440	
AA40/1440	144					36+	36+	36*	36*	1440	
Long Sydney	120	30+	30+	30+	30+					1200	
Sydney	120		30+	30+	30+	30+				1200	
Long Brisbane	120	30+	30+	30*	30*					1200	
Brisbane	120		30+	30+	30*	30*				1200	
Adelaide	120			30+	30+	30*	30*			1200	
Short Adelaide	120				30+	30+	30*	30*		1200	
Hobart	90	30+	30+	30+	30+					900	
Perth	90		30+	30+	30+					900	

- There should also be a “RoundCategories” table with round name (PK), class (PK) and division, describing the valid categories which can be played in for each round type. This will implement the table shown below:

- There should also be an “EquivalentRounds” table with base round (PK), equivalent round (PK), class (PK), division (PK) and date (PK).

Event	Open		50+		60+, 70+		Under 21		Under 18		Under 16		Under 14	
	Male	Female	Male	Female	Male	Female	Male	Female	Male	Female	M/F	M/F	M/F	M/F
Target Rounds														
WA90/1440	RC						RC							
WA70/1440	B	RC	RC				B	RC	RC					
WA60/1440	L	B	B	RCB	RCB		L	B	RCB	RC				
AA50/1440		L	L	L	RCBL		L	L	BL	RC				
AA40/1440										BL	RCBL			

- The club should have the option of adding new round types to the database.

Competitions

- A competition is a set of rounds shot by multiple people on a specific date.
- A “Competitions” table should include the competition name (PK) and competition date (PK).

Club Championships

- Club championships take place once per year.

User Needs

Add an archer:

```
1 INSERT INTO Archer(ArcherID, FirstName, LastName, BirthYear, Gender)
2 VALUES ("archerid", "firstname", "lastname", birthyear, "gender");
```

Archer entering their scores into the database:

The code below shows how a transaction is used to add a round score to the archery project database. If the creation of any of the ends fail, the transaction is rolled back. Otherwise, the transaction is committed.

```
1 $archer = $_POST["archer"];
2 $round = $_POST["round"];
3 $date = date("Y-m-d");
4 $class = $_POST["class"];
5 $division = $_POST["division"];
6 $competition = $_POST["competition"];
7 $conn->query("START TRANSACTION");
8 if ($competition === "") {
9     $conn->query("INSERT INTO ShotRound (ArcherID, RoundName, Date, Division, Class, Verified) VALUES ('$archer', '$round', '$date', '$division', '$class', TRUE)");
10 } else {
11     $conn->query("INSERT INTO ShotRound (ArcherID, RoundName, Date, Division, Class, Verified, CompetitionName) VALUES ('$archer', '$round', '$date', '$division', '$class', TRUE, '$competition')");
12 }
13 $round_id = mysqli_insert_id($conn);
14 $ranges = mysqli_query($conn, "SELECT * FROM RoundRange WHERE RoundName='$round'");
15 $range = mysqli_fetch_row($ranges);
16 $rangenum = 1;
17 while ($range) {
18     $conn->query("INSERT INTO ShotRange (ScoreID, RangeNum, RangeID) VALUES ($round_id, $rangenum, $range[0])");
19     $totalends = $range[4] / 6;
20     for ($endnum = 1; $endnum <= $totalends; $endnum++) {
21         $endtotal = $_POST["$rangenum-$endnum"];
22         $xtotal = $_POST["$rangenum-$endnum-x"];
23         for ($arrownum = 1; $arrownum <= 6; $arrownum++) {
24             $arrowscore[$arrownum] = $_POST["$rangenum-$endnum-$arrownum"];
25         }
26         $insertEnd = "INSERT INTO ShotEnd (ScoreID, RangeNum, EndNum, Arrow1, Arrow2, Arrow3, Arrow4, Arrow5, Arrow6, TotalEndScore, XTotal)
27             VALUES ($round_id, $rangenum, $endnum, '$arrowscore[1]', '$arrowscore[2]', '$arrowscore[3]', '$arrowscore[4]', '$arrowscore[5]', '$arrowscore[6]', $endtotal, $xtotal)";
28         if (!(mysqli_query($conn, $insertEnd))) {
29             $insertfailure = true;
30         }
31     }
32     $rangenum += 1;
33     $range = mysqli_fetch_row($ranges);
34 }
35 if ($insertfailure) {
36     $conn->query("ROLLBACK");
37     $scoreadded = false;
38 } else {
39     $conn->query("COMMIT");
40     $scoreadded = true;
41 }
```

The following SQL commands are used in this PHP snippet:

```
1 -- if not part of a competition
2 INSERT INTO ShotRound(ArcherID, RoundName, Date, Division, Class, Verified)
3 VALUES ("archerid", "roundname", date, "division", "class", "verified");
4 -- if part of a competition
5 INSERT INTO ShotRound(ArcherID, RoundName, Date, Division, Class, Verified, CompetitionName)
6 VALUES ("archerid", "roundname", date, "division", "class", "verified", "competitionname");
7 -- for each range
8 INSERT INTO ShotRange(ScoreID, RangeNum, RangeID)
9 VALUES ("scoreid", "rangenum", "rangeid");
10 -- for each end
11 INSERT INTO ShotEnd(ScoreID, RangeNum, EndNum, Arrow1, Arrow2, Arrow3, Arrow4, Arrow5, Arrow6, TotalEndScore, XTotal)
12 VALUES ("scoreid", "rangenum", "endnum", "arrow1", "arrow2", "arrow3", "arrow4", "arrow5", "arrow6", "totalendscore", "xtotal");
```

Recorder verifying a score shot by an archer:

```
1 UPDATE ShotRound
2 SET Verified = 1
3 WHERE ScoreID = ""; -- score id goes here
```

Recorder removing a score that was deemed to be unsuitable:

```
1 START TRANSACTION
```

```

2 DELETE FROM ShotEnd
3 WHERE ScoreID = ""; -- score id goes here
4 DELETE FROM ShotRange
5 WHERE ScoreID = ""; -- score id goes here
6 DELETE FROM ShotRound
7 WHERE ScoreID = ""; -- score id goes here
8 COMMIT

```

Recorder or Archer adding a new custom round:

```

1 START TRANSACTION
2 INSERT INTO RoundType(RoundName, TotalArrows)
3 VALUES ("roundname", totalarrows);
4
5 INSERT INTO RoundRange(RoundName, Distance, TargetFace, TotalArrows)
6 -- for each range the round should contain
7 VALUES ("roundname", "distance", "targetface", totalarrows);
8 COMMIT

```

Recorder adding a new equivalent round:

```

1 INSERT INTO EquivalentRound(RoundName, Class, Gender, Division)
2 VALUES ("roundname", "class", "gender", "division");

```

Recorder or Archer adding a new competition:

```

1 INSERT INTO Competition(CompetitionName, RoundName)
2 VALUES ("competitionname", "roundname");

```

Archer checking the results of a specific competition:

```

1 SELECT
2 sr.ArcherID,
3 a.FirstName,
4 a.LastName,
5 sr.RoundName,
6 sr.Date,
7 sr.Division,
8 sr.Class,
9 SUM(se.TotalEndScore) AS RoundScore,
10 SUM(se.XTotal) AS XTotal
11 FROM ShotRound AS sr
12 -- join archer by archer ID
13 JOIN Archer AS a ON sr.ArcherID = a.ArcherID
14 -- join shot range to get the valid primary keys for ShotEnd
15 JOIN ShotRange sra ON sr.ScoreID = sra.ScoreID
16 -- join shot end to get the total score and XTotal
17 JOIN ShotEnd se ON sra.ScoreID = se.ScoreID AND sra.RangeNum = se.RangeNum
18 WHERE sr.CompetitionName = "" -- Competition name added here
19 AND sr.Verified = 1 -- only verified scores are shown
20 GROUP BY
21 sr.ArcherID,
22 a.FirstName,
23 a.LastName,
24 sr.RoundName,
25 sr.Date,
26 sr.Division,
27 sr.Class
28 ORDER BY RoundScore DESC, XTotal DESC;

```

Recorder adding the rounds that will be shot for a club championship:

```

1 INSERT INTO ClubChampionship(RoundName, ChampionshipYear, TopTwoScore)
2 VALUES ("roundname", year, "bool");

```

Archer checking the current scores of a club championship:

This statement is one of the most complex ones with multiple joins and a check for an exact match in EquivalentRound. As PhpMyAdmin on the Swinburne database is used, this statement has been made to be in line with MariaDB 5.5, which has limited functionality such as the inability to use a Common Table Expression. The statement below returns the total scores for all archers who have shot their equivalent of the WA90/1440, ordered by the total score of the round and then by the total x numbers in the round. In the backend of the application hosting this, the roundname will need to be changed and the query re-executed for each round in the club championships.

```

1 SELECT
2 -- select everything that should be seen
3 sr.ArcherID,
4 a.FirstName,
5 a.LastName,
6 sr.RoundName,
7 sr.Date,
8 sr.Division,
9 sr.Class,
10 SUM(se.TotalEndScore) AS RoundScore,
11 SUM(se.XTotal) AS XTotal
12 FROM ShotRound AS sr

```

```

13 -- join archer by archer ID
14 JOIN Archer AS a ON sr.ArcherID = a.ArcherID
15 -- Ensure that all rounds shot are in the right year
16 JOIN ClubChampionship AS cc ON YEAR(sr.Date) = cc.ChampionshipYear
17 -- Check for an exact match in EquivalentRound
18 LEFT JOIN EquivalentRound AS er
19 ON cc.RoundName = "WA90/1440" -- other equivalent round names could go on here in application backend as a variable.
20 AND sr.RoundName = er.RoundName
21 AND sr.Class = er.Class
22 AND sr.Division = er.Division
23 AND a.Gender = er.Gender
24 -- join shot range to get the valid primary keys for ShotEnd
25 JOIN ShotRange sra ON sr.ScoreID = sra.ScoreID
26 -- Join shot end to get the total score and XTotal
27 JOIN ShotEnd se ON sra.ScoreID = se.ScoreID AND sra.RangeNum = se.RangeNum
28 WHERE er.RoundName IS NOT NULL
29 AND sr.Verified = 1
30 GROUP BY
31 sr.ArcherID,
32 a.FirstName,
33 a.LastName,
34 sr.ScoreID,
35 sr.RoundName,
36 sr.Date,
37 sr.Division,
38 sr.Class
39 ORDER BY RoundScore DESC, XTotal DESC;

```

Archer can look up their own scores over time:

```

1 SELECT
2 sr.ArcherID,
3 a.FirstName,
4 a.LastName,
5 sr.Date,
6 sr.Division,
7 sr.Class,
8 SUM(se.TotalEndScore) AS TotalScore,
9 SUM(se.XTotal) AS XTotal
10 FROM ShotRound AS sr
11 JOIN Archer AS a ON sr.ArcherID = a.ArcherID
12 JOIN ShotRange AS sra ON sr.ScoreID = sra.ScoreID
13 JOIN ShotEnd AS se ON sra.ScoreID = se.ScoreID AND sra.RangeNum = se.RangeNum
14 WHERE sr.ArcherID = "" -- ArcherID goes here
15 AND sr.Verified = 1
16 GROUP BY
17 sr.ArcherID,
18 a.FirstName,
19 a.LastName,
20 sr.Date,
21 sr.Division,
22 sr.Class
23 ORDER BY Date DESC;

```

Archer or Recorder looking into the score of specific ranges of a round that has been shot:

This assumes the archer/recorder has the ScoreID of the round they want to see the ranges of (or has clicked something in an application that provides this ScoreID). Verified has been left out as it is assumed if the archer can see their score in the application then it is verified.

```

1 SELECT
2 sr.RangeNum,
3 rr.Distance,
4 rr.TargetFace,
5 rr.TotalArrows,
6 SUM(se.TotalEndScore) AS RangeScore,
7 SUM(se.Xtotal) AS XTotal
8 FROM ShotRange AS sr
9 JOIN RoundRange AS rr ON sr.RangeID = rr.RangeID
10 JOIN ShotEnd se ON sr.ScoreID = se.ScoreID AND sr.RangeNum = se.RangeNum
11 WHERE sr.ScoreID = "" -- score id goes here
12 GROUP BY
13 sr.RangeNum,
14 rr.Distance,
15 rr.TargetFace,
16 rr.TotalArrows
17 ORDER BY RangeNum;

```

Archer/Recorder looking into the specific ends of a range that has been shot:

This assumes the archer/recorder has the ScoreID of the round they want to see the ends of, and the RangeNum they want to see the ends of (or has clicked something in an application that provides these). Verified has been left out as it is assumed if the archer can see their score in the application then it is verified.

```

1 SELECT
2 RangeNum,

```

```

3 EndNum,
4 Arrow1,
5 Arrow2,
6 Arrow3,
7 Arrow4,
8 Arrow5,
9 Arrow6,
10 TotalEndScore,
11 XTotal
12 FROM ShotEnd
13 WHERE RangeNum = "" -- range num goes here
14 AND ScoreID = "" -- ScoreID goes here
15 ORDER BY EndNum;

```

Archer checking the club's best score for a round and archer who shot it:

```

1 SELECT
2 sr.ArcherID,
3 a.FirstName,
4 a.LastName,
5 sr.Date,
6 sr.Division,
7 sr.Class,
8 SUM(se.TotalEndScore) AS TotalScore,
9 SUM(se.XTotal) AS XTotal
10 FROM ShotRound AS sr
11 JOIN Archer AS a ON sr.ArcherID = a.ArcherID
12 JOIN ShotRange AS sra ON sr.ScoreID = sra.ScoreID
13 JOIN ShotEnd AS se ON sra.ScoreID = se.ScoreID AND sra.RangeNum = se.RangeNum
14 WHERE sr.RoundName = "" -- round name goes here
15 AND sr.Verified = 1
16 GROUP BY
17 sr.ArcherID,
18 a.FirstName,
19 a.LastName,
20 sr.Date,
21 sr.Division,
22 sr.Class
23 ORDER BY TotalScore DESC, XTotal DESC;

```

Archer can check personal best score for a round:

```

1 SELECT
2 sr.ArcherID,
3 a.FirstName,
4 a.LastName,
5 sr.Date,
6 sr.Division,
7 sr.Class,
8 SUM(se.TotalEndScore) AS TotalScore,
9 SUM(se.XTotal) AS XTotal
10 FROM ShotRound AS sr
11 JOIN Archer AS a ON sr.ArcherID = a.ArcherID
12 JOIN ShotRange AS sra ON sr.ScoreID = sra.ScoreID
13 JOIN ShotEnd AS se ON sra.ScoreID = se.ScoreID AND sra.RangeNum = se.RangeNum
14 WHERE sr.RoundName = "" -- round name goes here
15 AND sr.ArcherID = "" -- ArcherID goes here
16 AND sr.Verified = 1
17 GROUP BY
18 sr.ArcherID,
19 a.FirstName,
20 a.LastName,
21 sr.Date,
22 sr.Division,
23 sr.Class
24 ORDER BY TotalScore DESC, XTotal DESC;

```


Team Health Monitor

- 💡 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	MAJRD
Sponsor	Swinburne University of Technology
Health monitor cadence	Fortnightly

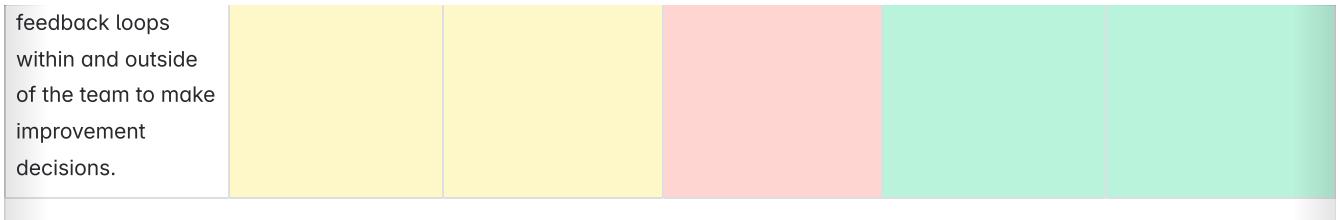
👩‍💻 Team health assessment 🕵️

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	Aiden	Panduka	Matt	John	Don
Team Cohesion ↗ We have the mutual trust and respect necessary to be an effective team for healthy collaboration. We have a strong sense of connectedness between members.	During meetings, all team members share ideas and are respectful when discussing them.	Team members effectively communicate both during workshops and in online chat. Everyone so far, seems to have done some part of the work assigned to them.	Working environment supports open communication. All members are very respectful and helpful to one another.	From the outset of the assignment, we have established a collaborative and supportive environment where everyone feels heard and supported. We have maintained regular check-ins and inclusive discussions.	Team members are effective communicators and are helpful with one another. Everyone is encouraged to communicate openly.

 Balanced team	The team is able to easily split up necessary tasks between the members. Though members don't really have distinct roles, there have been no problems with people being unable to complete their tasks.	The roles aren't too clearly defined. This may be okay for a small scale project like this. Everyone does a bit of contribution on everything. However, we definitely have the right people with the right skills.	Each member has demonstrated that they are adequately equipped with the skills and knowledge to contribute to tasks at hand.	Our team has effectively divided the work for necessary tasks. Although, we don't have any specific roles assigned, there were no problems with the tasks.	All team members are equipped with the tools needed to succeed. Roles can often overlap or not be clearly defined, everyone contributes to all tasks.
 Encouraging Difference	We seek and voice different viewpoints from diverse sources, both internally and externally, and we take the time to respectfully work through points of differences.	Though everyone is able to share their thoughts during meetings, we don't go out of our way to get an idea from everyone.	Everyone seriously takes into consideration what everyone else's opinions and thoughts are on a matter before we move forward.	Although the team has mostly been on the same page thus far, our environment enables and encourages differing opinions nevertheless.	Our team is open to suggestions from everyone and values each member's perspective. We actively listen to different point of views during discussions.
 Shared understanding	We share an understanding of our mission and purpose and our key milestones to deliver our strategic plan effectively as a team.	We usually dedicate a portion of each meeting to understanding what our goals will be for the coming week so that everyone has a healthy understanding of their tasks and expectations.	The milestones and key goals are clearly defined on the project page. We've gone over this multiple times.	Advice has been exchanged between team members and facilitated by our tutor on topics requiring more attention.	We have clear understanding of what needs to be done for each task and how it contributes to overall goals. We stay aligned on goals and milestones.

Value and metrics	<p>We understand the value we provide and the value back to the business, our definition of success and how that value is tracked and measured. We ultimately leverage our metrics to make decisions and action as necessary.</p>	<p>We have not focused much on the value that could come from this project, instead focusing on simply completing the goals and deliverables set out for us to the best of our ability.</p>	<p>Our definition of success seems to be reaching goals with fairness and efficiency. We track all of this through confluence.</p>	<p>Metrics haven't been utilized and leveraged very much, but the values in which both the team and its individual members provides is somewhat clear.</p>	<p>Although we have successfully completed every task, we are not fully clear on the value. We lack a clear understanding of the metrics.</p>	<p>We have not often utilized our metrics to make any decisions, however the value each team members provides is clear as we make a conscious effort to split up the workloads.</p>
Suitable ways of working	<p>Our ways of working together within the team enable us to do our jobs effectively, whether we are distributed or co-located. This includes the tools we use, how we meet and collaborate, and how we make decisions.</p>	<p>Our usual workflow is based around communicating tasks and updates during the weekly meetings, and working individually on our tasks the rest of the time. So far, this process has caused no significant issues.</p>	<p>Everyone is working at home. We all do our parts and we report back to the leader. So far, it's going smoothly. As for tools, everyone is using confluence.</p>	<p>Current system allows us to work independently effectively and also to come together to discuss key areas when necessary.</p>	<p>Our current way of working together has been effective, and we have faced no major issues. As for tools we use the confluence to check for updates.</p>	<p>Our ways of working have proven to work okay so far. However, communication can sometimes be sparse and who is doing what is often only delegated verbally, rather than written.</p>
Continuous improvement	<p>We always make time to celebrate our successes as well as earnestly reflect on, take action against, and fulfill our improvement opportunities. We have regular and intentional</p>	<p>We do not often take the opportunity to look back on our previous work to analyze how we could improve, instead focusing on the tasks at hand.</p>	<p>We do not always take the time to celebrate successes. However, we do reflect on our progress from time to time.</p>	<p>Haven't had very many opportunities dedicated to reflection, or sessions in which ways where improvement can be had has been discussed.</p>	<p>We often celebrate our success and take time to reflect on our actions. We encourage feedback both within the team and from the tutor, ensuring we continuously evolve and improve.</p>	<p>We often celebrate our successes, and we regularly reflect on our actions and decisions to revise and work to improve on what we have created so far.</p>



🎯 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 area and action items
15/04/2025	<p>Continuous Improvement</p> <ul style="list-style-type: none"> <input type="checkbox"/> Take time at the start of each meeting to focus on some highlights and celebrate the work of others. <input type="checkbox"/> Look to provide feedback to each other each meeting so as to take any opportunities to improve <input type="checkbox"/> Seek external feedback from our Tutor each week in order to ensure we are on the right track and get an outside perspective of our progress

Software Development Tasks

Ideas for webpages:

- Add scores # AIDEN
 - User selects archer and round (also competition if applicable).
 - Site looks up roundtype table to get number of ranges/range types/number of arrows.
 - Users should be forced to enter arrow scores from largest to lowest.
- View scores, css # MATT
 - User inputs at least one of Archer ID, Date range or Competition name to be able to view selected scores
- View rounds/add rounds # DONOVAN
 - User can view the rounds available and see how many ends/arrows they contain at what difference. Users can also add a custom round
- View/Add/Edit Archer details # REEVE
 - User can add new archers, remove existing archers
- View/Add/Edit Competitions # REEVE
 - Users can add/edit/remove competitions and their details.

Software Development Outcomes

- Add scores # AIDEN

The screenshot shows a web browser window titled "Add Score | MAIRD Archery". The URL is mercury.swin.edu.au/cos20031/s103597864/addscore.php. The form has fields for "Archer ID:" (empty), "Round:" (Brisbane), "Class:" (Open), "Division:" (Recurve), and "Competition (optional:)". A green "Add Score" button is at the bottom.

The screenshot shows the same "Add Score" form with the following details: Archer: Aiden Large, Round: Melbourne, Class: Open, Division: Longbow. Under "Range 1:", it shows a 10x7 grid of scores for 15 ends. The scores are: End 1: X, 10, 10, 9, 8, 7 (Total 54); End 2: X, X, 10, 9, --, -- (Total --); Rows 3-7: X (blue), 10 (blue), 9 (blue), 8 (blue), 7 (blue), 6 (blue), 5 (blue), 4 (blue), 3 (blue), 2 (blue), 1 (blue), M (blue). A red "Delete" button is at the bottom right of the grid.

The screenshot shows the "Add Score" form with the same details as the previous screenshot. It displays two ranges: Range 1 (15 ends) and Range 2 (15 ends). The scores for Range 2 are: End 7: 10, 10, 9, 6, 6, 6 (Total 47); End 8: X, 10, 10, 8, 7, 7 (Total 52); End 9: 8, 8, 5, 4, 4, 3 (Total 32); End 10: 10, 9, 9, 7, 6, 6 (Total 47); End 11: X, X, 10, 8, 7, 5 (Total 50); End 12: 10, 9, 9, 7, 6, 6 (Total 47); End 13: 10, 7, 6, 6, 5, 4 (Total 38); End 14: 8, 7, 7, 6, 5, 5 (Total 38); End 15: 10, 10, 9, 9, 7, 6 (Total 51). At the bottom are green "Add Score" and red "Abort Score" buttons.

- View scores, css # MATT

Archer Name	Date	Score X Total	Round Name	Division	Competition
Ryan Collins	2024-05-11	499	4	Short Recurve	The Matthew Shoot
Andre Russell	2024-05-15	183	0	Short Barebow	The Matthew Shoot
Alicia Campbell	2024-09-09	790	8	Short Canberra	The Matthew Shoot
Paula Miller	2024-10-11	300	7	Short Longbow	The Matthew Shoot
Michael Tate	2024-11-14	668	8	Short Compound	The Matthew Shoot
Carol Allen	2025-02-19	932	16	Short Barebow	The Matthew Shoot
Brianna House	2025-04-16	871	16	Short Compound Barebow	The Matthew Shoot

 Below the table is a small note: 'Web app created by: Aiden Large, Donovan Quilty, Matthew Xu, Reeve Kariyawasam'."/>

- Archers can search by ArcherID or CompetitionName to find all relevant scores.
- View rounds/add rounds # DONOVAN

Distance	Total Arrows	Target Face
20m	90	120cm

 Below the table is a small note: 'Web app created by: Aiden Large, Donovan Quilty, Matthew Xu, Reeve Kariyawasam'. At the bottom of the page is a blue button labeled 'View more rounds'."/>

The page further describing a round's details.

The screenshot shows a web application titled "MAJRD Database". At the top, there is a navigation bar with five buttons: "Search", "Add", "Scores", "Competitions", and "Archers". Below the navigation bar, a section titled "Search Rounds" is displayed. It contains a sub-section titled "Search through rounds to see the specifications of them here". There is a text input field labeled "Search by Round Name: [Enter Round Name]" and a blue button labeled "Search by Round Name". Below this, there is another blue button labeled "See all rounds". A table is shown with the following data:

Round Name	Total Arrows	Max Score
WA90/1440	144	1440
WA70/1440	144	1440
WA60/1440	144	1440
AA50/1440	144	1440

The page to search for types of rounds. Round types are displayed in a table with the total number of arrows and its maximum possible score.

The screenshot shows a web application titled "MAJRD Database". At the top, there is a navigation bar with five buttons: "Search", "Add", "Scores", "Competitions", and "Archers". Below the navigation bar, a section titled "Add a round" is displayed. It contains a sub-section titled "Enter round name. Round name must be unique". There is a text input field labeled "San Francisco" and a dropdown menu labeled "How many ranges will be in this round? 8". Below this, there is a blue button labeled "Press here to input the ranges". At the bottom of the page, there is a footer with the text "Web app created by: Aiden Large, Donovan Quilty, Matthew Xu, Reeve Kariyawasam".

A filled in example of a round to add.

MAJRD Database

Search Add Scores Competitions Archers

Enter Ranges for round

Enter values for range 1
 Note: only 1 range per distance can be inputted. Total arrows must be divisible by 6.
 Enter Distance Please Select ▾

80cm 120cm

Total Arrows

Enter values for range 2
 Note: only 1 range per distance can be inputted. Total arrows must be divisible by 6.
 Enter Distance Please Select ▾

80cm 120cm

Total Arrows

Press here to create round

Web app created by: Aiden Large, Donovan Quilty, Matthew Xu, Reeve Kariyawasam

Ranges are added one by one after a round has been created. A transaction is used to ensure a round is not added unless all ranges are valid too.

- View/Add/Edit Archer details # REEVE

The screenshot shows the 'MAJRD Database' application. At the top, there is a navigation bar with five buttons: 'Search', 'Add', 'Scores', 'Competitions', and 'Archers'. Below the navigation bar, the title 'MAJRD Database' is displayed in a large, bold font. Underneath the title, the section 'Manage Archers' is shown. A sub-section titled 'Add a new archer:' contains fields for 'Archer ID' (with a note: 'Enter unique 11 digit ID To remove. DB structure needs slight update first.'), 'First Name', 'Last Name', 'Birth Year', and 'Gender'. A blue 'Add' button is located below these fields. Another sub-section titled 'Existing Archers:' is present, with a note 'Sorted by Last Name'. It displays a table with columns: ID, First Name, Last Name, BirthYear, and Gender. The data in the table is as follows:

ID	First Name	Last Name	BirthYear	Gender
5033156105	Alec	Adams	1971	Female
9941940189	Ryan	Allen	1968	Male
2150383403	Matthew	Allen	2005	Female
6242751115	Carol	Allen	1954	Male

User can add new archers, remove existing archers

- View/Add/Edit Competitions # REEVE

The screenshot shows the 'MAJRD Database' application. At the top, there is a navigation bar with five buttons: 'Search', 'Add', 'Scores', 'Competitions', and 'Archers'. Below the navigation bar, the title 'MAJRD Database' is displayed in a large, bold font. Underneath the title, the section 'Manage Competitions' is shown. A sub-section titled 'Add a new competition:' contains fields for 'Name' (with a note: 'The Aiden Cup') and 'Round Type' (with a note: 'Mini Canberra - 90 arrows'). A blue 'Create' button is located below these fields. Another sub-section titled 'Existing Competitions:' is present. It displays a table with columns: Competition Name and Round Type. The data in the table is as follows:

Competition Name	Round Type
keyboard	AA50/1440
MercuryBoard 2	Hobart
mouseboard	WA90/1440
SomethingortheOther	WA70/1440
The Aiden Cup	Hobart
The Donovan Cup	WA70/720

Users can add/edit/remove competitions and their details.

Add a new competition:

Name:

Round Type: ▾

A competition with the name 'The Aiden Cup' already exists

Create

Existing Competitions:

Competition Name	Round Type
keyboard	AA50/1440
MercuryBoard 2	Hobart
mouseboard	WA90/1440
SomethingortheOther	WA70/1440
The Aiden Cup	Hobart

Competition names must be unique

Cybersecurity Major Task - John

User Case - Archer

Archer need to be able to enter scores via their hand-held device into a staging table, view their scores and look up round and competition info.

COMMANDS for Access of Archer

- Creating the role for Archer

```
CREATE ROLE Archer;
```

- Granting access to lookup information.

```
GRANT SELECT ON Archer TO Archer;
```

```
GRANT SELECT ON Round TO Archer;
```

```
GRANT SELECT ON RoundRange TO Archer;
```

```
GRANT SELECT ON EquivalentRound TO Archer;
```

```
GRANT SELECT ON ClubChampionship TO Archer;
```

```
GRANT SELECT ON Competition TO Archer;
```

- Allowing archer to look up and enter their score

```
GRANT SELECT, INSERT ON ShotRound TO Archer;
```

- Revoking access to Archer to modify the score tables

```
REVOKE INSERT, UPDATE, DELETE ON ShotRound FROM Archer;
```

```
REVOKE INSERT, UPDATE, DELETE ON ShotEnd FROM Archer;
```

User Case - Recorder

Recorder should to able manage all data, validate and approve staging scores, update Competition and Club Championship

COMMANDS for Access of Recorder

- Creating role for Recorder

```
CREATE ROLE Recorder
```

- Allowing full access to Recorder

```
GRANT SELECT, INSERT, UPDATE, DELETE ON Archer TO Recorder;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON Round TO Recorder;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON ShotRound TO Recorder;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON RoundRange TO Recorder;  
GRANT SELECT, INSERT, UPDATE, DELETE ON EquivalentRound TO Recorder;  
GRANT SELECT, INSERT, UPDATE, DELETE ON Competition TO Recorder;  
GRANT SELECT, INSERT, UPDATE, DELETE ON ShotRange TO Recorder;  
GRANT SELECT, INSERT, UPDATE, DELETE ON ClubChampionship TO Recorder;  
GRANT SELECT, INSERT, UPDATE, DELETE ON ShotEnd TO Recorder;
```

Prepared Statements to Prevent SQL Injection

The following statements were used in implementing the database.

- Archer Submitting new score into staging table via hand-held device

```
INSERT INTO ShotRound (ArcherID, RoundName, Date, Division, Class, Verified, CompetitionName )  
VALUES (?, ?, ?, ?, ?, ?, ?);
```

- Archers view their score by filtered by date

```
SELECT * FROM ShotRound  
WHERE ArcherID = ? AND Date BETWEEN ? AND ?  
ORDER BY Date DESC;
```

- Archers view their score by filtered by round

```
SELECT * FROM ShotRound  
WHERE ArcherID = ? AND RoundName = ?  
ORDER BY Score DESC;
```

- Recorder adding arrow-by-arrow score

```
INSERT INTO ShotEnd (ScoreID, RangeNum, EndNum, Arrow1, Arrow2, Arrow3, Arrow4, Arrow5, Arrow6, TotalEndScore,  
XTotal)  
VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?);
```

- Recorder adding equivalent rounds

```
INSERT INTO EquivalentRound(RoundName, ClassName, Gender, Division)  
VALUES(?, ?, ?, ?);
```

- Recorder adding a new round

```
INSERT INTO RoundType(RoundName, TotalArrows)  
VALUES(?, ?);
```

- Recorder adding a new range

```
INSERT INTO RoundRange(RoundName, Distance, TargetFace, TotalArrows)  
VALUES(?, ?, ?, ?);
```

- Adding a class

```
INSERT INTO Class(ClassName, MinAge, MaxAge)  
VALUES(?, ?, ?);
```

We have implemented input validation in the development of the archery database web application to ensure data integrity and prevent invalid or malicious input. This includes validating user input for required fields, checking data formats and value ranges (e.g., ensuring total arrows are numeric, within acceptable limits, and divisible by 6), as well as sanitising inputs to protect against security threats such as XSS and SQL injection.

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	MAJRD
Team members	@DONOVAN QUILTY @Aiden Large @Reeve Kariyawasam @Matthew Xu @John Jaswanth Carmel Madanu
Date	27/05/2025
Retrospective period	Q2

💡 4Ls retrospective 💡

Milestones	Loved	Longed for	Loathed	Learned
ER Diagram Design	<ul style="list-style-type: none">Creating relations and problem solving with normalizationConfluence enabled straightforward collaboration	<ul style="list-style-type: none">More collaboration or discussion between teammatesA greater understanding of normalization and indexing	<ul style="list-style-type: none">Trying to draw arrows and edit things with draw.io	<ul style="list-style-type: none">The fundamentals of relational database design and ERD modellingIterations with database and creating more meaningful ways of connecting tables
MYSQL Implementation	<ul style="list-style-type: none">Found the LinkedIn videos very useful in learning MYSQL commandsWorking as a group to come up with the create table statements	<ul style="list-style-type: none">Being able to come up with cleaner, less convoluted SQL statements.More options in projects outside of just archery	<ul style="list-style-type: none">Trying to understand and troubleshoot foreign key constraints	<ul style="list-style-type: none">Ways of filtering results from select statements.More complex topics such as indexing and inner joins
Major Specific Work	<ul style="list-style-type: none">Revisiting skills learned in previous units	<ul style="list-style-type: none">More collaboration between team members.	<ul style="list-style-type: none">Connecting to SQL databases with another	<ul style="list-style-type: none">Learned how to implement PHP with Vue and Bootstrap.

	(web development)	<ul style="list-style-type: none"> An opportunity to demo our work for marks. 	<ul style="list-style-type: none"> member's credentials Handling error troubleshooting with mysqli errors. 	<ul style="list-style-type: none"> Connecting multiple PHP redirects to each other Complex SQL queries in PHP
--	-------------------	--	--	---

⚡ Action plan ⚡

Action	Owner	Action items
Consider and evaluate different ERD drawing tools.	@DONOVAN QUILTY	Research different ERD tools, try and compare them to draw.io , generate a recommendation for the group
Enable PHP setting to make sure any mysqli error throws an exception with appropriate error details	@Reeve Kariyawasam	Enable MYSQLI_REPORT_ERROR only during development. (It's a security risk if left enabled as users could view sensitive data)
Explore related but alternative solutions/frameworks when applying previously learnt skills to expand skillset	@Matthew Xu	Research tools that can expand and or simplify processes which can be implemented.
Utilize confluence's project management tools to a further extent	@John Jaswanth Carmel Madanu	Research and discover different tools offered within confluence. Discuss how they could be implemented.
Assign a database as the group database	@Aiden Large	Allow every group member to access a single database, which will be updated whenever the ER diagram is changed. This will ensure consistency between all major specific projects and consistency with the current database design.

Product Video Division of Labor

Powerpoint Link: [Product Video.pptx](#)

Section	Member	Time
Title slide with team names	Matthew Xu	10 seconds
Introduction and project background, summary of requirements	Matthew Xu	20 seconds = 30 seconds
Mission statement and objectives	Reeve Kariyawasam	20 seconds = 50 seconds
1-2 slides about team process (collaboration, roles, contributions to solution)	Reeve Kariyawasam	30 seconds = 1 min 20 sec
1-3 slides about the logical/physical design, discussion of decisions made regarding the entities and their relationships	Donovan Quilty	30 seconds = 1 min 50 sec
Discussion of solutions to use cases (e.g. SQL, also the design solutions influenced by these use cases - there is overlap with item 5 here);	John	30 seconds = 2 min 20 sec
1-5 slides on major-specific technology;	Aiden Large	45 seconds = 3 min 05 sec
Summary of solution and lessons learnt.	Donovan	20 seconds = 3 min 25 sec