Curtin University – Department of Computing

Assignment Cover Sheet / Declaration of Originality

Complete this form if/as directed by your unit coordinator, lecturer or the assignment specification.

Last name:	Ching	Student ID:	22013213
Other name(s):	Mao Jin		
Unit name:	Database System	Unit ID:	ISYS2014
Lecturer / unit coordinator:	Ms. Nimalika	Tutor:	Mr. Foad
Date of submission:	21/10/2024	Which assignment ?	(Leave blank if the unit has only one assignment.)

I declare that:

- The above information is complete and accurate.
- The work I am submitting is *entirely my own*, except where clearly indicated otherwise and correctly referenced.
- I have taken (and will continue to take) all reasonable steps to ensure my work is *not accessible* to any other students who may gain unfair advantage from it.
- I have *not previously submitted* this work for any other unit, whether at Curtin University or elsewhere, or for prior attempts at this unit, except where clearly indicated otherwise.

I understand that:

- Plagiarism and collusion are dishonest, and unfair to all other students.
- Detection of plagiarism and collusion may be done manually or by using tools (such as Turnitin).
- If I plagiarise or collude, I risk failing the unit with a grade of ANN (“Result Annulled due to Academic Misconduct”), which will remain permanently on my academic record. I also risk termination from my course and other penalties.
- Even with correct referencing, my submission will only be marked according to what I have done myself, specifically for this assessment. I cannot re-use the work of others, or my own previously submitted work, in order to fulfil the assessment requirements.
- It is my responsibility to ensure that my submission is complete, correct and not corrupted.

Signature: CMJ Date of signature: 18/10/2024

(By submitting this form, you indicate that you agree with all the above text.)

Introduction: I have created a database that consists of the Athlete, Coach, Event, Medallist, Schedule, Register, Team, Venue tables. Besides, I have also created 3 views – virtual tables which include athlete summary details, Gold Medal and isIncluded to

show the athletes that are included in the medallist. I have created trigger to automate the delete process of the related athlete details in the child tables when an athlete is deleted from the parent table. This is useful as user do not have to manually delete the relevant tuples from the child table. The first stored procedure is named UpdateAthleteName which handles situation such as an athlete has entered the name wrongly and requires an update in his name. The second procedure is named DeleteAndReplaceSport which handles situation like when a sport is deleted from the Event table, the child tables that contains the old sport will be replaced with the new sport. I have written a python program for the database frontend to ease the user for inserting, deleting, updating the table and executing the pre-defined SQL queries. The pre-defined SQL queries are written previously and user can just select the question that they are interested to know. All the pre-defined SQL queries are in the script called 'DesigningAndImplementingQueries.sql". The goal of this project is to allow users to store large amount of data in the database system, at the same time allowing users to retrieve relevant data and modify incorrect data in the system.

Entity Set

Entity Set	Keys	Other Attributes
Athlete	Athlete_ID(PK) Coach_ID(FK)	Sport, Athlete_Name, Gender, Team_Name, Country, Competition_Type, Coach_Name, Age
Coach	Coach_ID(PK)	Coach_Name, Gender, Country, Sport,
Event	Event_ID(PK) Venue_ID(FK)	Sport, Competition_Type, Start_Date, End_Date, Event_Time
Venue	Venue_ID(PK)	Venue_Name, Sport
Schedule(Weak Entity)	Athlete_ID(PK) Event_ID(PK)	Sport, Competition_Type, Start_Date, End_Date
Team	Team_ID(PK)	Sport, Coach_Name, Team_Name, Country
Medallists(Weak Entity)	Athlete_ID(PK), Event_ID(PK)	Competition_Type, Gender, Medal_Place, Country, Sport, Athlete_Name

These are the entities which are the nouns involved with the scnario and they are implemented as tables for the database.

Relationship Set

Relationship Sets	Between Which Entity sets	Attributes of Relationship Sets
Coached	Athlete, Coach	-
Takes-place	Event, Venue	Event_Time
isIncluded	Athlete, Medallist	-
Joins	Athlete, Team	Joining_Date
Register	Athlete, Event, Schedule	Registration_Date

These are the relations between each entity. Register is a ternary relation and the rest are binary relation. Since register is a ternary relation, it will have its own table.

Relation Constraints

Relationship Sets	Cardinality Constraints	Participation Constraints
Coached	Many to one Many-Athlete One-Coach One coach can coach many athlete. Since one athlete can only be coached by one coach, coach will be one	Athlete – Total Participation Athlete cannot exist without a coach. Athlete must have at least a coach for guidance.
		Coach – Total Participation Coach cannot exist without an athlete. Coach must have at least an athlete to keep his job.

Takes-place	Many to One Many- Event One-Venue One venue can have many events organized but one event only requires one venue.	Event – Total Participation Event cannot exist without a venue. An event must take place in a venue.
		Venue- Partial Participation Venue can exist even without a event. An aquatic centre has been there for years even without an olympic game.
isIncluded	Many to many Many-Athlete Many-Medallist Since athlete can register in more than one event, there'll be chances where athlete may be in more than one medallist. A medallist can consists more than one athlete.	Athelete – Partial Participation Athlete can still exist without being included in a medallist. Not all athlete has a medal in olympics and yet they are still an athlete.
		Medallist – Total Participation Medallist cannot exist without at least an athlete. Medallist keeps track of the athlete awarding a medal. Without any athlete, there'll will be no medallist.
Joins	Many to one Many-Athlete One-Team A team will have more than one athlete. One athlete will only need to join for a team.	Athlete- Partial Participation Athlete can exist without joining a team. Not all sport is done in a team, it can be done individually.
		Team- Total Participation Team cannot exist without an athlete. Some sports like artistic swimming is done in a team, there must be athletes in a team to perform.
Register	Many to many to many Many-Athlete ManySchedule Many-Event More than one athlete can register for more	Athlete- partial Participation Athlete can exist without participating an event. Not all national athlete are qualified for olympic games according to their country policy.
		Schedule-Total Participation Schedule can't exist without at least one athlete. The schedule records the event participated by the athlete. If there's no athlete participated in Olympic game, schedule won't exist.

	Many to many to many Many-Athlete ManySchedule Many-Event More than one athlete can register for more than one event and one event can have many athletes. Since an athlete may have registered for more than one event, then they will have more than one schedule	Event-Total Participation Event can't exist without an athlete. Event refers to the games provided Olympic. If there's no participation, the event can't occur.

The above describes the cardinality constraint and participation constraint of each entity set.

Data Description

Athlete Table

Attribute	Data Type	Size	Null	Description
Athlete_ID(PK)	Int	-	No	Identity Number of an athlete, must be unique
Team_Name	Varchar	30	Can be null	The name of the team Eg: TeamMY . some of the athlete may not join a team due to the sport can be done individually like swimming.
Country	char	5	No	The country represented by the athlete. It has to be in short name.

				Eg: Malaysia, MY. When inserting country name, please enter MY not Malaysia.
Competition_Type	varchar	12	No	The type of competition joined by the athlete. It only accepts team or individual.
Coach_Name	Varchar	50	No	The name of the coach. The name can be short or full regardless.
Athlete_Name	varchar	50	No	The name of the athlete, can be short or long
Gender	Char	2	No	The gender of the athlete registered. Please only enter ‘M’ or ‘F’ . ‘M’ stands for Male and ‘F’ stands for female
Sport	Varchar	100	No	The sport name registered by the athlete. Eg: Swimming, Diving
Age	Int	-	No	The age of the Athlete. The age must be between 14 to 35 years old inclusive to be qualified for olympic games. This is the

				assumption I made and I have implemented a check constraint for it.
Coach_ID(FK)	Int	-	No	The identidy number of the coach that ranges from 1 and increment by 1 for each entries

The reason why I chosed varchar for Strings is Varchar will only take the amount of memory space required for a String. Some attributes like Gender are fixed size so using Char can improve the performance.

Coach Table

Attribute	Data Type	Size	Null	Description
Coach_ID(PK)	int	-	No	Identity number of a coach- Unique
Coach_Name	varchar	100	No	Name of the coach, can be short or long
Gender	Char	2	No	Gender of the coach
Country	char	5	No	Country where the coach belongs to
Sport	Varchar	100	No	Sport handled by the coach

Venue Table

Attribute	Data type	Size	Null	Description
Venue_ID(PK)	Int	-	No	The identity number of the venue -Unique

Venue_Name	Varchar	100	No	The name of the venue. Eg: Aquatic Centre
Sport	Varchar	100	No	Sport name

Event Table

Attribute	Data type	size	null	Description
Event_ID(PK)	Int	-	No	The identity number of the sport
Sport	Varchar	100	No	The name of the sport for a particular event
Competition_Type	Varchar	12	No	The type of the competition. For example: team or individual
Start_Date	Date	-	No	Starting date of a particular event in the format of 'yyyy-mm-dd'
End_Date	Date	-	No	End date of a particular event
Venue_ID (FK)	Int	-	No	Venue Identity number – Foreign Key
Event_Time	Time	8	No	Starting time for the event in the format of 'hh-mm-ss'

Medallist Table (Weak entity)

Attribute	Data type	Size	Null	Description
Athlete_ID(PK , FK)	Int	-	No	Since medallist is a weak entity set, the primary key

				depends on another entity called Athlete
Event_ID (PK, FK)	int	-	No	Unique indentity number of an event. This is a foreign key because it refers to the PK of event table which is Event_ID.
Athlete_Name	Varchar	50	No	Athlete Name Can be full or short
Medal_Place	Int	-	No	The medal place awarded by athlete. Eg: 1,2,3
Competition_Type	Varchar	12	No	The type of the competition. It only accepts the input of team or individual.
Country	char	5	No	The short form of a country name represented by athlete. Eg: AUS
Gender	Char	2	No	Athlete Gender. Eg: 'M' or 'F'
Sport	varchar	100	No	The sport name registered by the athlete. Eg:Badminton

Team Table

Attribute	Data type	Size	Null	Description

Team_ID(PK)	Int	-	No	Identity number of a team - Unique
Team_Name	Varchar	30	No	Team name
Coach_Name	Varchar	50	No	Coach Name
Country	char	5	No	Country short Name
Sport	Varchar	100	No	Sport name

Schedule Table(Weak Entity)

Attribute	Data type	Size	Null	Description
Athlete_ID(PK & FK)	Int	-	No	Since schedule is a weak entity, its primary key depends on another entity
Event_ID(PK & FK)	Int	-	No	It is depends on the primaru key of Event
Start_Date	Date	-	No	Starting date of the event
End_Date	Date	-	No	Ending date of the event
Competition_Type	Varchar	12	No	Type of Competition
Sport	Varchar	100	No	Sport name

Register Table

Attribute	Data type	Size	Null	Description
Athlete_ID(FK & PK)	Int	-	No	Athlete Identity Number
Event_ID (FK & PK)	Int	-	No	Event identiy number
Registration_Date	Date	-	No	Registration Date for the event

Relational Schema

Coach(Coach_ID, Coach_Name, Gender, Country, Sport)

Athlete(Athlete_ID, Athlete_Name, Country, Sport, Team_Name, Competition_Type, Coach_Name, Gender, Coach_ID, Age)

FK Coach_ID REF Coach(Coach_ID)

Venue(Venue_ID, Venue_Name, Sport)

Event(Event_ID, Sport, Competition_Type, Start_Date, End_Date, Venue_ID, Event_Time)

FK Venue_ID REF Venue(Venue_ID)

Athlete(Athlete_ID, Athlete_Name, Country, Sport, Team, Competition_Type, Coach_Name, Gender, Coach_ID, Age)

Medallist(Athlete_ID, Event_ID,Athlete_Name, Medal_Place, Competiton_Type, Country, Gender, Sport)

FK Athelete_ID REF Athlete(Athlete_ID)

FK Event_ID REF Event(Event_ID)

isIncluded(Athlete_ID, Event_ID, Athlete_Name)

FK Athlete_ID REF Athlete(Athlete_ID)

FK Event_ID REF Event(Event_ID)

Team(Team_ID, TeamName, Coach_Name, Country, Sport)

Athlete(Athlete_ID, Athlete_Name, Country, Sport, Team, Competition_Type, Coach_Name, Gender, Coach_ID, Team_ID, Age)

FK Team_ID REF Team(Team_ID)

Athlete(Athlete_ID, Athlete_Name, Country, Sport, Team, Competition_Type, Coach_Name, Gender, Coach_ID, Age)

Schedule(Athlete_ID, Event_ID, Start_Date, End_Date, Competition_Type, Sport)

FK Athlete_ID REF Athlete(Athlete_ID)

FK Event_ID REF Event(Event_ID)

Register(Athlete_ID, Event_ID, Registration_Date)

FK Athlete_ID REF Athlete(Athlete_ID)

FK Event_ID REF Event(Event_ID)

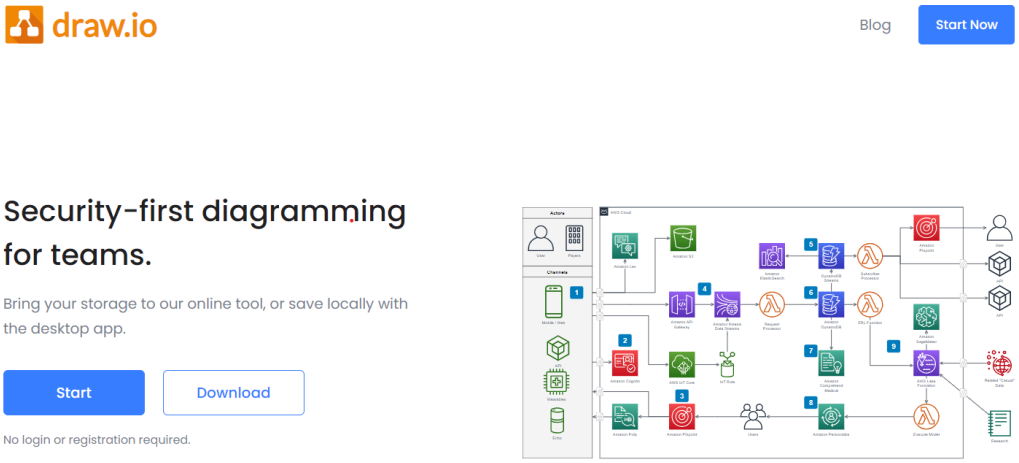
Explanation on the schema: All tables are in first normal form because none of the row have more than one attribute value. All tables are in second normal form because there's no partial dependency. All tables are in third normal form because they are in second normal form and there is no transistivity dependency. Transistivity dependency is an attribute depends on another non-prime attribute and the non-prime attribute depends on the prime attrbitutes. This is known as transistivity dependency.

Note*: Kindly take note that the explanation regarding the participation constraint and cardinality contraints is attached under the participation contraint column and cardinality contraints column for all the relations. Each table consists of the attributes related to its entity set. Please refer to the entity set table for specific attributes of a table that you wish to know. So far theres's only one attribute which allows inserting a null value which is the team name under the athlete table. Explanation regarding that are attached under the description column, please refer to it. Kindly take note of the attributes such as name can be either short or full, Gender can only be 'M' or 'F', Country must be in short form Eg: MY stands for Malaysia, Competition_Type is either 'individual' or 'team'. This is applicable for all entity that have the following attributes. The reasons that I have selected the entities above is that this database aims to provide interesting information about the participants and winner of the games. Hence, all the entity sets are required for the task. You may have noticed that varchar is being used for those varying string length, and char is mostly used for fixed length of string. This is to prevent memory space wastage as varchar will only take up to the memory space needed for a string. However, char is a fixed size regardless of the length of the string. The start_Date and end_date are of date data type which take in the formate of "YYYY—MM—DD" which enables me to use some of the date formats that I learnt in Lecture 2 of Database system. The ER Diagram is drawn using the website known as draw.io. To view it, please use the website. The link will be provided under references. The only assumption that I have made in this project is the age of the athlete must be between 14 to 35 to be qualified for any games in olympic. Here's the guide to open the ER Model:

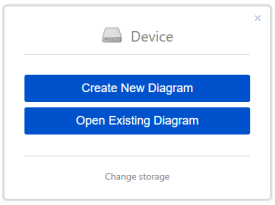
Step1:

Click on this link: https://www.drawio.com/

After clicking the link, it brings to the page below.

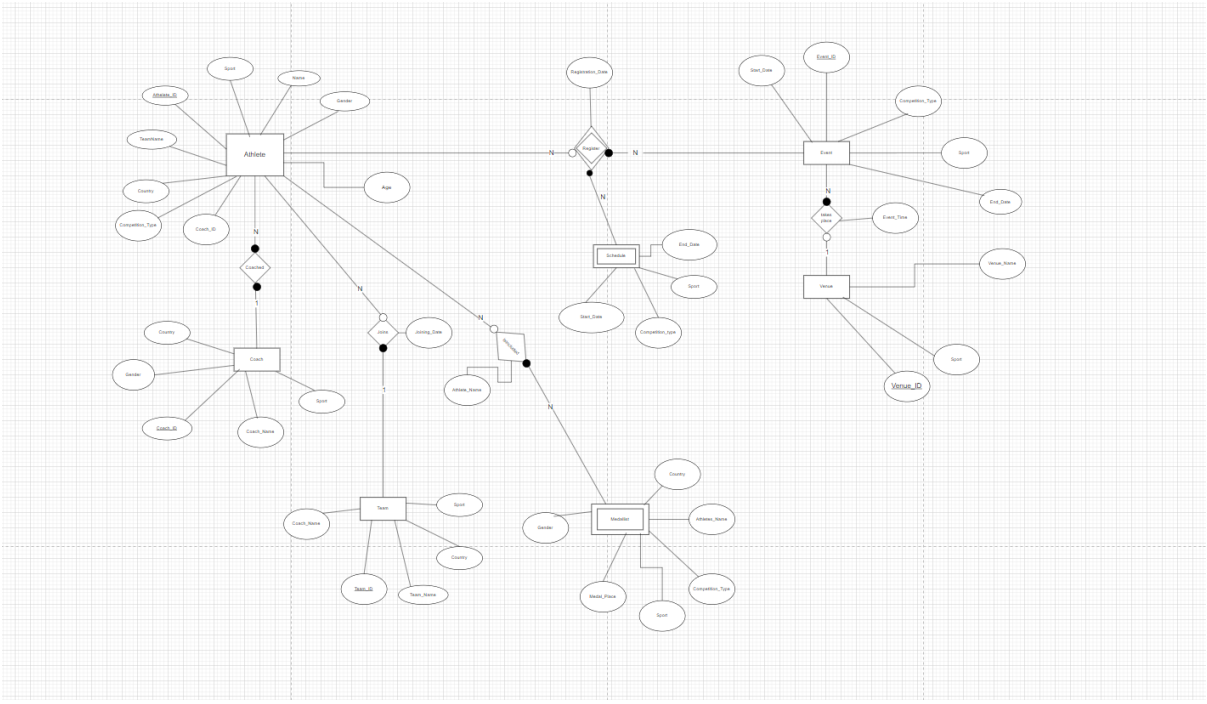


Click the ‘Start’ button. After clicking the ‘Start’ button, you will be brought to this page.



Click on “open Existing Diagram” and select the file called ER Diagram from your folder.

You will eventually open ER Model online. You should be able to see the Er model as the attachment below after opening it.



Implementating database and adding sample data

Implementation of the database and adding sample data with evidence is available in the user guide. Please refer to the user guide. According to my lecturer, the accuracy of data does not matter and he won't be questioning me about the accuracy of the data. The names used in the entity sets are generated randomly from the random name generator webiste as provided in the reference list. The sport names are also taken randomly from a webiste and the link is provided in the reference list.

meaningful queries

Here're the questions and answers to the questions in SQL query. You can also refer to the sql script "DesigningAndImplementingQueries.sql" .

-- Q1: What is the coach name of the athlete

select Coach_Name from Athlete;

The query is to list out all the coach name of the athlletes

-- Q2: What are the male athletes

select Athlete_Name,Athlete_ID, gender from Athlete where gender='M'

order by(Athlete_ID) desc;

The query is to list out all the Athlete Names, Athlete ID for male athletes only

In the descending order of Athlete ID.

-- Q3: What is the event time for all types of football match

select Event_Time, Sport from Event where Sport Like '%football%';

The query is to list out event time for all types of football match so that the participants can check the event time.

-- Q4: Remove the athlete name from medallists

Alter table Medallist Drop Column Athlete_Name;

-- Q5: What are the athletes that are included in the medallists

Select A.Athlete_Name, M.Medal_Place from Athlete A LEFT OUTER JOIN Medallist M

on A.Athlete_ID=M.Athlete_ID

where M.Athlete_ID IS NOT NULL;

This query is to find out which athletes are in the medallist.

-- Q6: What are the athletes that awarded gold medal

select A.Athlete_Name, M.Medal_Place, M.Athlete_ID from Athlete As A inner join

(select Medal_Place, Athlete_ID from Medallist where Medal_Place=1) as M

on M.Athlete_ID=A.Athlete_ID

order by(M.Athlete_ID) desc;

This query is to display the athletes that awarded a gold medal.

-- Q7: What is the coach for football

```
select Coach_Name, Sport from Coach
where Sport like '%football%';
```

This query display the coaches that in charge of all types of football.

```
-- Q8: What is the venue for badminton
select Venue_Name, Sport from Venue
where Sport='Badminton';
```

This query displays the venue for badminton.

```
-- Q9: What is the event date for football
select Date_Format(Start_Date, '%e %M %Y') as Event_Date, Sport from Event
where Sport like '%football%';
```

This query shows the event date of football in the example format of 27 October 2024.

```
-- Q10: What are the athletes that are not in the medallists
select A.Athlete_Name, A.Athlete_ID from Athlete as A left outer join Medallist as M
on A.Athlete_ID=M.Athlete_ID
where M.Athlete_ID is null;
```

This query displays the athletes are not in the medallists.

```
-- Q11: Group the athlete based on the same sport and display the column number of athlete and sport
select Count(Athlete_Name) as Num_of_Athletes, Sport from Athlete Group by (Sport);
```

This query will group the athlete based on the same sport and display the number of athletes taking the same sport.

```
-- Q12: How many events takes place between 2024-8-20 to 2024-9-20
```

```
select Count(Sport) as Num_Of_Events from Event
where Start_Date between '2024-08-20' and '2024-09-20';
```

This query will count how many events takes place between 2024-08-20 to 2024-09-20.

```
-- Q13: Total numbers of athletes that are not in the medallists
select count(A.Athlete_ID) as number_of_Athletes_not_on_medalList from Athlete as A
left outer join Medallist as M
on A.Athlete_ID=M.Athlete_ID
where M.Athlete_ID is null;
```

This query will count the total number of Athletes that are not in the medallist.

```
-- Q14: Find the average age of the athletes of each sport
select Avg(Age) as Average_Age, Sport from Athlete
Group By(Sport);
```

This query displays the average age of the athlete for each sport.

```
-- Q15: Display the athletes who are taking the sport and the sport
select A.Athlete_Name as Athlete1, B.Athlete_Name as Athlete2, A.sport
from Athlete A, Athlete B
where A.sport=B.sport
AND A.Athlete_ID <> B.Athlete_ID
order by (A.sport);
```

This query displays the athletes who are taking the same sport.

The evidence of successful implementation of these queries will be shown during the assignment demostration as the output is very lengthy.

Advanced Features

The advanced features are created in the storedProcedure sql scripts, views scripts and trigger scripts. The function of each advanced feature is commented in the scripts. The evidence is shown in the user guide. Please refer to them.

Database connectivity

The evidence of successful database connectivity is displayed in the user guide with screenshot. I have written a python code for the database connectivity which allows the user to execute the pre-defined SQL queries shown above, insert, update and delete the existing data in the database.

Discussion: I have created a database system to store a large amount of data and allow users to retrieve, update and delete the data from the existing data available in the system. The challenges that I face is thinking what attributes for each entity and what entity set I need for this project. Creating an ER Model was quite challenging at first because I needed to figure out the relationship between each entity and the relation constraints which requires a lot of thinking process. There're definitely limitations in this project which include not implementing a human computer interface of the database frontend.

References

Britannica, T. Editors of Encyclopaedia (2015, October 5). *list of sports*. *Encyclopedia Britannica*. https://www.britannica.com/topic/list-of-sports-2038581

Catonmat. (N.A). *Generate Random Names*. https://catonmat.net/tools/generate-random-names

Petro, P. (2024). *Paris 2024 Olympic Summer Games.* Kaggle. https://www.kaggle.com/datasets/piterfm/paris-2024-olympic-summer-games?resource=download