# Project, Part 1
# CS 4347 Fall 2024
# Due September 20th, 2024, 11:59pm

## 1 Objective

To begin the modelling aspect of constructing a database for the Database Class. The goal of the project is to build the database, not the database application.

### 1.1 Instructions:

This assignment is a Cohort Assignment. Students will form a team, called a cohort, to complete this assignment. While everyone may work together and share information, each member of the cohort must submit their own copy of the project.

### 1.2 Products:

To that end, certain phases must be completed. This phase consists of

- Form a Cohort, 10 points
- Mini-world, 10 points
- Software discovery, 10 points
- An extended entity-relationship diagram, 35 points
- Two Dictionaries, 35 points
  - An Entity Dictionary
  - A Relationship Dictionary

### 1.3 Grading

This project phase is graded based on the following results

- Documents: All documents are submitted. If a document is not submitted or is unreadable, then full points are deducted.
- Completion and Size: A list of what is required of the size and type of pieces in this project is listed in section 5. If these specifications are not met, points will be deducted based on the amount of material missing. For example, a diagram without a class lattice would lose 5 points. A missing cardinality or missing underlined primary key would lose 2 points. (Up to 35 points)
- Dictionaries: The dictionaries must be sorted by alphabetical order by the name of the entity, subclass, or relationship. If the submission is not alphabetized, it is not a dictionary, and full points will be deducted. (up to 35 points)
- Matching: The dictionary matches the diagram, and the diagram matches the Mini-World. Every time one portion of the project does not match another portion of the project, 2 points will be deducted.
- Effort and Clarity: If the documents are confusing, badly spelled, badly drawn to such an extent that the person grading cannot understand or even read the model, 2 points will be deducted for each time the person grading is confused.

# 2 Cohort: (10 points)

For the term project, each Cohort must create a Database. Each cohort must write up an English description of what they plan to create. The cohort's name will be used as a prefix for submitting documents, programs, and archive files.

## 2.1 How big should a Cohort be?

The cohort should be between three to five people. Cohorts bigger than this size tend to fall apart. Yes, a student can attempt a cohort of 1 person, but it is not recommended. When doing this type of project, it helps to have someone to trade ideas.

## 2.2 How will the Cohort function?

Once the Cohort is formed, cohorts are expected to stay connected with each other using current network software and in-person meetings. GitHub, Facebook, Discord, GroupMe, Slack, and whatever the next cool software will be. In addition, face-to-face meetings are strongly recommended. Class time will be made available for Cohort meetings.

## 2.3 Naming the Cohort and Writing the Description

Each Cohort should have a good, clear cohort name. The name chosen must be suitable to be put into the Blackboard gradebook. Blackboard has a 32-character limit, and yes, the cohort's name must be readable, useable characters. Such a name can be silly or serious, but it needs to be appropriate (syllabus), recognizable, and understandable. Also include a description, serious or silly, that goes with the cohort name. The description should be about 100 to 200 words long.

## 2.4 What if something goes wrong with the Cohort?

In previous semesters, several interpersonal issues have occurred while teaching this class.

- "Piggy-backing"-a cohort member has never shown up and never participated. The remaining cohort members can remove that member from server access on their own accord.
- "Peer-Grading Extortion"-a cohort member will tell the others they must surrender their project documents or else they will receive a bad peer review. No peer grading this semester. If this happens, the remaining Cohort members can remove that member from server access on their own accord.
- "Cohort break up" – if a cohort breaks into other cohorts, or students want to swap to another cohort, this can be done by the students without assistance from the TA or the Instructor. The Cohort members would just email the TA and the Instructor the new configuration.
- "Forgot to turn it in"-a cohort member who has not turned in their assignment because they thought the project was a group turn-in can ask their Cohort members for help. If this situation is an ***honest mistake***, the Cohort member who missed their turn in can ask the other Cohort members to vouch for them. If this issue should happen, the cohort members should send an email to the TA and the Instructor vouching for the student.

Include a paragraph, 100 to 200 words, that states that all cohort members understand their responsibilities as a member of the cohort. Basic items include routine communication and participation, Some cohorts include additional guidelines for members who have young children or a full-time job or other external need. This paragraph should be a guideline statement, not a contract with compulsory components.

## 2.5 What to turn in

Turn in a PDF *<CohortName>.Charter.pdf* with the following information:

- The name of the cohort.
- The reason for the name of the cohort.
- A statement of responsibility
- A listing of the members of the cohort.
-

# 3 Mini-World: (10 points)

The proposal is a write up of what the cohort wants to build. The proposal must include a situation, product, or network that requires the use of a database. E-Commerce, computer games, chat services, and social media all use databases to store and manipulate information. A mini-world is a subset of the real world. A database can always be expanded with more detail, but for the project, the cohort should start with a limited description. This description can be changed in later parts of the project. Any cohort can create their own database on a different topic than this example, but the mini-world must be complicated enough for a database to be required.

E

Example Mini-world: The Olympics

It's the Olympics! The Pride of Nations! Where the best representatives of the athletes of their home countries come to compete for GOLD! Whoo-hoo!

Each Athlete has a first name, a last name, a height, a weight, and a gender. Each athlete also has a media nickname, and the date and time the nickname was given.

Athletes are from a nation or a governing body. Each nation has a name, a capital city, a head of state, and an ambassador. Each nation provides a uniform for their Athletes. The uniforms are of a style and coloration to represent their home nation.

Each event has a rank, with the top three being gold, silver, and bronze. Each event has a scoring system of points. Points are awarded by one or more judges or referees. Each event can be either a team event or an individual event. Each event has a particular time and location in which they are held.

Events are held at a particular venue. Venues can be an indoor auditorium, an outdoor stadium, on a route through the city, on a route through the country, or on a body of water. Venues have a name, and may or may not have a year in which they were constructed. A canal was constructed, a river was not.

Events are associated with a particular sport. Each sport has a name, the first year the sport was at the Olympics. And each sport has a category such as track and field, or gymnastics, or swimming. And each category may have its own properties. Gymnastics may be by performance, while swimming may be against a clock.

## 3.1 What to turn in

Turn in a PDF *<CohortName>.Miniworld.pdf* with the following information:

- The name of the cohort must be in the document.
- The description of the Mini-World in English.

# 4 Software Discovery: (10 points)

For this project, an entity-relationship diagram must be constructed. Undoubtedly, a dozen systems exist for drawing such diagrams.

## 4.1 Choosing the Drawing Software

Whether it be pencil and paper, a simple paint program, to a professional tool, to online freeware: Draw IO, Visio, MySQL workbench and other software packages exist as either open source or as educational software. What software will the cohort use? Where is the software from? Be sure to include a screen capture. Yes, even if the design tool chosen is pencil and paper.

## 4.2 What to turn in

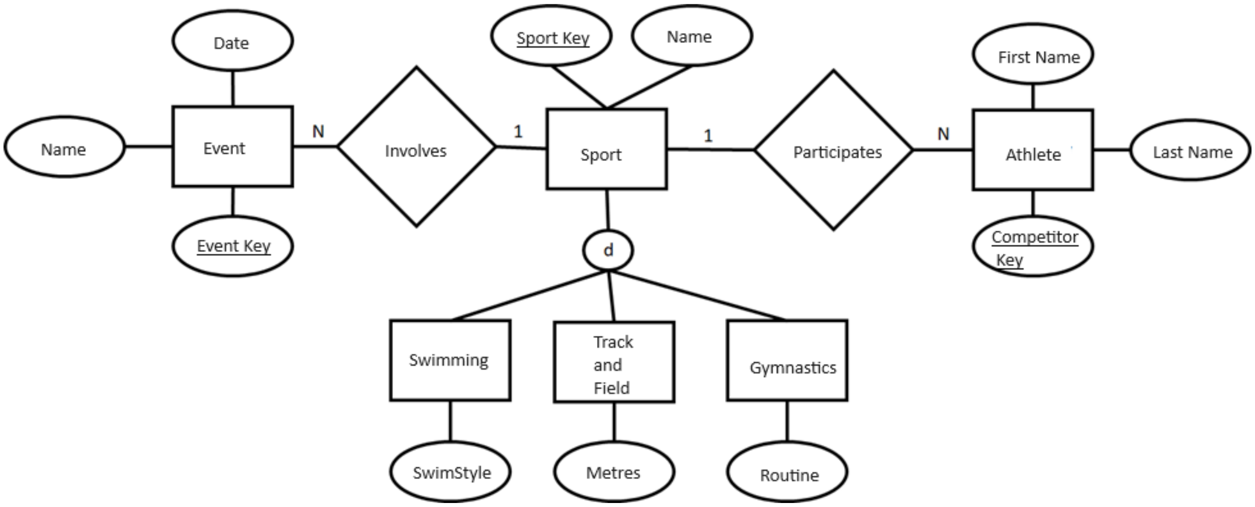
Turn in a PDF named *<CohortName>.SoftwareDiscovery.pdf* with the following information:

- What software is the Cohort using? Cite the software. The typical citation includes the name, the version (if any), the URL of the web page, and the date the software was downloaded.
- Include a picture of what tools the Cohort is using.
- Give a short description why the Cohort is using this software, even if the reason is “Its cheap and free.”

# 5 Extended Entity-Relationship Diagram (35 points)

Please be aware there are many, many database tools in existence that auto-generate diagrams. The Cohort will be expected to use the symbols and notations from the textbook and examples for this project. If the Cohort uses auto-generating software that does not follow the lecture, the diagram must be redrawn with the correct symbols and notation to receive credit.

A small fragment of such a diagram would be:



## 5.1 For this section:

- The entity-relationship diagram must be based on the mini-world.
- The diagram must match the Mini-World.
- It should be "complex enough."
    - How big should the diagram be?
    - Adding all the entities, subclasses, and relations together and wind up between 20 and 30, then the project is large enough. A total number of boxes and diamonds, about 25 should be sufficient.
- The Diagram MUST have
    - Every strong entity must have a primary key attribute.
    - Every weak entity must have an owning entity and may include a surrogate key.
    - Every entity must have useful information attributes.
    - Every relationship must have a cardinality.
    - Every relationship must have a participation.
    - At least one weak entity must exist in this diagram.
    - At least one owning relationship must exist in this diagram.
    - At least one class lattice must exist in this diagram.

## 5.2 Assumptions

If Cohort members find themselves arguing over a particular piece of the diagram during design, this action indicates an assumption has been made to make the design work. Be aware of such discussions, and if necessary, add them as a document to the Extended Entity Relationship Diagram. If no assumptions have been made, include a document stating this fact. If the cohort does not make any assumptions, please state so in the document.

## 5.3 What to turn in

Turn in a PDF named *<CohortName>.EERDiagram.pdf* with the following information:

- A complete image of the Extended Entity-Relationship Diagram

Turn in a PDF named *<CohortName>.Assumptions.pdf* with the following information:

- Describe any assumptions the Cohort feels would aid the grader in understanding the diagram. If no assumptions, state there are no assumptions.

# 6 Entity-Relationship Dictionary (35 points)

A Dictionary is a list of all the pieces of the diagram, with their definitions. Every entity, relationship, attribute, key, and other property in the diagram must have an entry. The entries must be detailed to match the diagram. If the entity has an attribute in the diagram, then the entity should have an attribute in the Dictionary. The entities and relationships must be sorted into alphabetical order in the Dictionary. There will be two dictionaries. The first dictionary is the Entity Dictionary, for entities and any of their subclasses. The second dictionary will be the Relationship Dictionary, for the relationships between entities.

***Example:***

**ENTITY DICTIONARY:**

**Entity:** Sport
**Description:**
A physical activity done in competition
**Attributes:**

- Name, Varchar(80), the name of the Sport
- Sport Key, Integer, an identifying number for the Sport
- Primary Key: PK_Com

**Entity:** Swimming
**Description:**
One of the kinds of sports being hosted at the Olympics
**Attributes:**

- SwimStyle, VarChar(32), a text label for the style of swimming (ie. Butterfly , Backstroke, etc)
- Primary Key: Inherits PK_Tch from superclass Sport

**RELATION DICTIONARY**

**Relationship:** Participates
**Description:** Athletes participate in sports, competing between two or more athletes in a single sport.
**Entities:** Sport, Athlete
**Cardinality:** One Sport can have Many Atheltes
**Participation:** Partial-to-Partial

## 6.1 Required

- The Entity Dictionary
  - The dictionary MUST be in alphabetical order by name of Entity.
  - Classes and Subclasses will be treated as Entities.
  - Every Entity on the diagram must be in the dictionary.
  - Every Attribute of every entity must be in the dictionary.
  - Every Strong Entity must have a defined primary key.
  - Every Weak Entity must be owned by a strong entity.
  - Weak Entities may have a surrogate key.
- The Relation Dictionary
  - The dictionary MUST be in alphabetical order by name of the Relation.
  - Every Relation in the diagram must be in the dictionary.
  - Any Attributes of a Relation must be in the dictionary.
  - The Entities of the Relation must be specified.
  - The Cardinality of the Relation must be specified.
  - The Participation of the Relation must be specified.

## 6.2 What to turn in

Turn in a PDF named *<CohortName>.EntityDictionary.pdf* with the following information:

- An alphabetically sorted list of all the entities and subclasses in the diagram, with all the correct properties listed.

Turn in a PDF named *<CohortName>.RelationshipDictionary.pdf* with the following information:

- An alphabetically sorted list of all the relationships in the diagram, with all the correct properties listed.