# CIS 492 - Entity-Relationship Diagram (ERD) Course Style Standards

## Course ERD Style Standards:

Your project's entity-relationship diagram (ERD) is required to meet the following course style standards:

- THOU SHALT AVOID **ANY** MENTION OF TABLES IN THY MODEL. The project is **not** yet at the table stage at this point!
- each **entity class** is represented by a rectangle, with the name of the entity class within the rectangle. There should be **only** 1 rectangle per entity class! (You can have multiple relationship class lines connected to a single entity class rectangle.)
- each **relationship class** is represented by a diamond on a line connecting the associated entity classes, labeled on or above or below the diamond with a descriptive (and preferably unique) name for that relationship class.
  - (that is, you draw a line between related entity classes, with the labeled diamond for that relationship class on that line…)
- the **maximum cardinality** of each relationship class must be depicted by writing the 1, M, or N near the end of the relationship line near the appropriate entity class rectangle. (See Example 1 below for an illustration of what is expected for this -- be especially careful to put these on the "correct" ends of the relationship line.)
- the **minimum cardinality** of each relationship class must be depicted by drawing either an oval or a line, whichever is appropriate, on each end of the relationship line (near the entity class rectangle). You draw an oval on the relationship line near the entity class rectangle if the relationship is optional at that end, and you draw a line across the relationship line near the entity class rectangle if the relationship is mandatory on that end. (Again, see Example 1 below for an illustration of what is expected for this.)
- below or on a separate page from the ER diagram, list the name of each entity class, and underneath each entity class name:
  - list the **attributes** for that entity class
  - for any attribute that can be **multi-valued**, write `(MV)` after that attribute
  - underline or write in all capital letters the attribute(s) that **identify** that entity (**if** there are any at this stage)
  - **Note:** Do **not** write these lists of entity class attributes as relation structures -- these are not relations yet! They are entities, and each entity will eventually be transformed into one or more relations during the database design phase, which **follows** the modeling phase.
  - Also remember: in the modeling phase, the attributes in these lists are attributes of that entity **alone** --- they do **not** indicate relationships between entities. The relationship lines in the ERD do that, at the modeling stage! A useful rule of thumb: each attribute should only appear once,

**total**, in this list of entities' attributes.

- **weak entities**, if needed/desired, should be depicted by having a double-border around their rectangle.
- **supertype/subtype/union entity classes** should be depicted as follows:
  - each supertype, union, and subtype entity class will be written within a labeled rectangle, as usual
  - the relationship lines do not have triangles on them, and are drawn differently: there is a line from the supertype or union entity rectangle to a labeled circle, and a line from each subtype entity rectangle to that same labeled circle
  - this labeled circle is labeled with the letter `o` if these are overlapping subtypes (if a supertype instance may be more than one of the subtypes), with a `d` if they are disjoint subtypes (if a supertype instance may be at most one of the subtypes), and with a `u` if they are union subtypes (the subtypes are more "truly" separate entities than in the case of usual supertype/subtypes, with their own identifying attributes and perhaps no overlap of attributes)
  - each relationship line to a subtype entity rectangle has a u shape on it (with the points of the u "facing" the labeled circle)
  - the relationship line to the supertype or union entity rectangle has a hash or line across it, near the supertype/union entity class rectangle, if every supertype or union entity instance is **required** to also be at least one of the subtype instances, and has an oval across it, near the supertype/union entity class rectangle, if every supertype or union entity instance is **not** required to also be at least one of the subtype instances
  - in their entity attribute lists, still be careful to only list attributes specifically for that entity class. Here, that means that attributes common to all subtypes of a supertype are listed **only** with the supertype entity class's attributes; the only attributes listed for subtype entity classes are those **particular** to that subtype. It is assumed that, in reality, a subtype entity inherits all of the attributes of its supertype entity class (and so there is no need to rewrite them for the subtype entity class's attributes). (Shades of object-oriented programming, where subclasses really do inherit all of the data fields and methods of its supertype!)
  - Examples 2, 3, and 4 below give examples using these standards

# Examples

For example, the following meets the above standards.

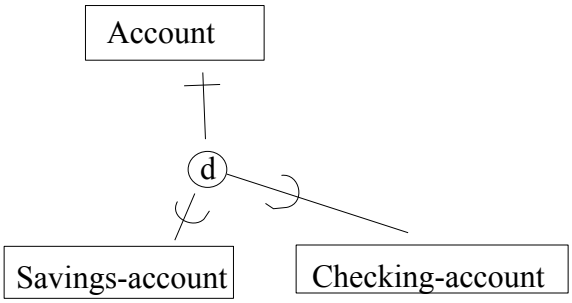
## *Example 1*


```
Professor                   Course-section
-----------------           ------------------
PROF_SSN                    Course_name
```

```
Prof_lname                  COURSE_ID
Prof_dept                   Course_text (MV)
```

In the above example, given the maximum and minimum cardinalities shown, a professor does not **have** to teach any course-sections, but may teach **many** course-sections. A course-section **must** have an associated professor, and may have **at most one** professor associated with it --- that is, there is exactly one professor associated with each course-session. Note that the Professor entity attributes do not include any indication of which courses that professor teaches, nor do the Course-section entity attributes include any indication of which professor teaches that course-section --- this is **not** an error. This is proper for the database modeling phase --- the relationship between professors and courses is indicated at this phase by the relationship line between their entity rectangles and by the minimum and maximum cardinalities shown on that relationship line.

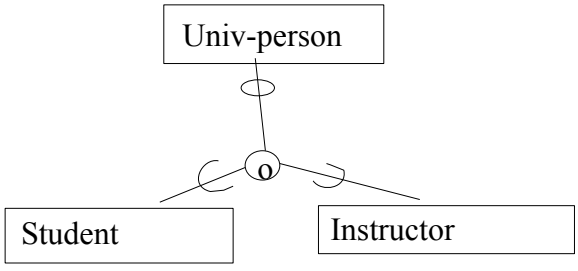
## *Example 2*




```
Account           Savings-account         Checking-account
-------           ---------------         ----------------
ACCT-NUM          interest-rate           per-check-charge
date-opened       minimum-balance         max-number-of-checks-per-month
current-balance
```

This depicts a supertype entity Account with two subtype entity classes Savings-account and Checking-account. In these particular scenarios, Savings-account and Checking-account are disjoint subtype entity classes: an Account entity instance may be one or the other, but not both. The hash near Account indicates that all Account entity instances must be either a Savings-account or a Checking-account; there are no Account entity instances that are neither a Savings-account nor a Checking-account.

## *Example 3*

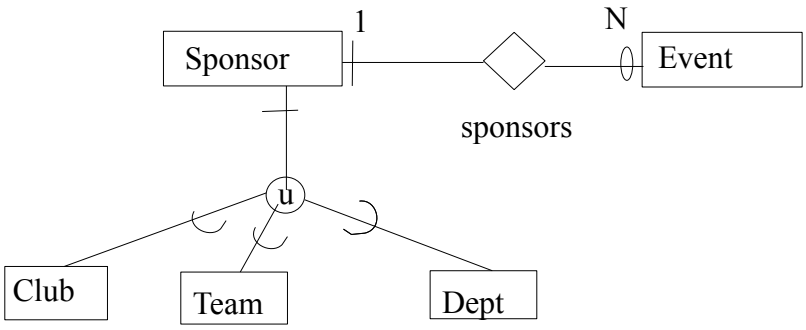



```
Univ-person       Student                 Instructor
-----------       -------                 ----------
UNIV-ID           gpa                     salary-per-course
Last-name
```

```
First-name
Campus-email
```

This depicts a supertype entity class Univ-person with two subtype entity classes Student and Instructor. A Univ-person entity instance may be both a Student entity instance and an Instructor entity instance, since Student and Instructor are overlapping subtype entity classes. The oval near Univ-person indicates that there can be a Univ-person entity instances that is neither a Student instance nor an Instructor instance (a University administrator, for example).

## *Example 4*




```
Sponsor     Club        Team        Dept             Event
-------     --------    --------    ---------        -----
            CLUB_NUM    TEAM_CODE   DEPT_CODE        EVENT_NUM
            Club_Name   Sport       Dept_title       Event_title
            Is_active   Season      Office_num       Event_date
```

This depicts a university-based scenario in which there are campus clubs, campus teams, campus departments, and campus events, and a business rule noting that campus events must be sponsored by either a club, a team, or a department.