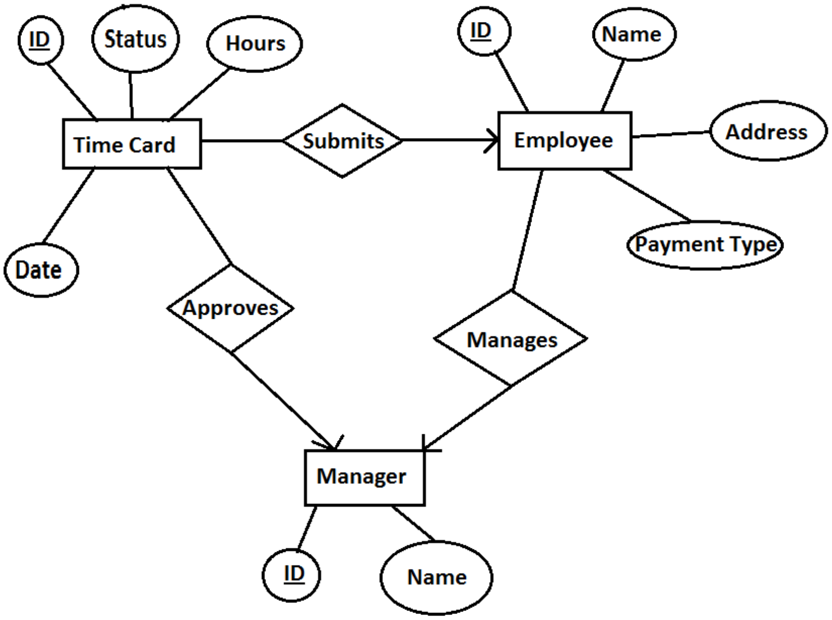
# Problem 1: ER Diagram Design

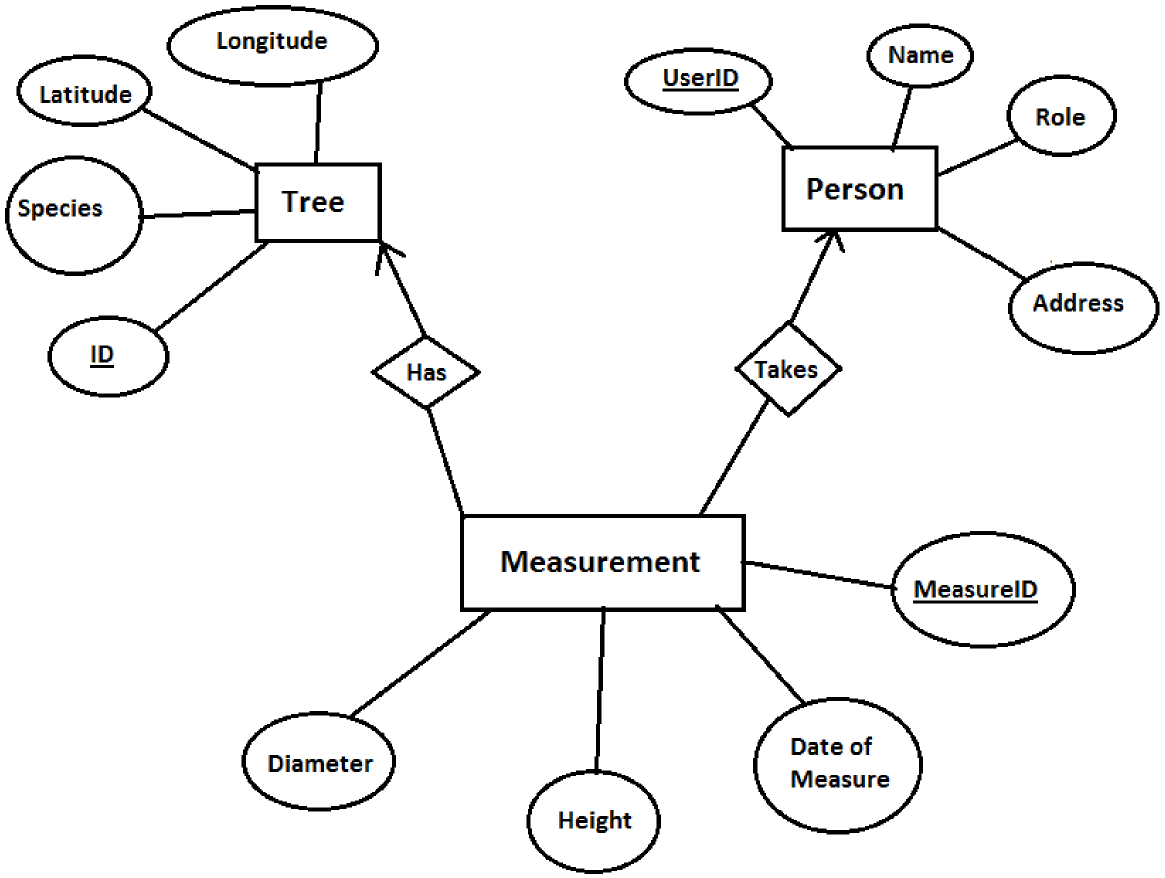
The company you work for wants to digitize their time cards. You have been asked to design the database for submitting and approving time cards. Draw the database ER diagram with the following information:

- A *timecard* should have hours worked and date submitted
- Each *timecard* is associated with exactly one *employee*
- Each *timecard* should have a unique id
- Each *timecard* has a status: it is either approved, not approved, or pending
- Each *employee* has a unique id
- Each *employee* has a name and address.
- Each *employee* submits a time card every pay period. i.e. In 1 year, they will submit multiple time cards
- Each *employee* either has direct deposit or physical check as their method of payment
- Each *employee* is associated with exactly one *manager*
- Each *manager* has a unique id and a name
- Each *manager* is in charge of multiple employees
- Each *manager* approves time cards for multiple employees

If you feel that you must make some assumptions, please state them clearly so that they are easily understood by the graders. Remember to indicate the key for each entity, as well as the multiplicity of each relationship (e.g. one-to-many) using the appropriate notation.

**Solution:**





## Use the above diagram for the following questions!

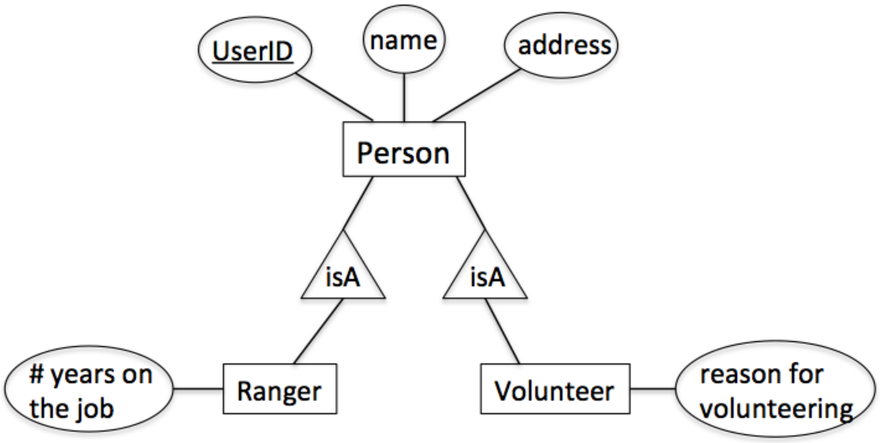
# Problem 2: More ER Fun

Answer the following questions based on the original ER Diagram in Figure 1:

1. Could the "Date of Measure" attribute of the *Measurement* entity be the key for the entity, instead of the "MeasureID"? Why or Why Not?
2. We want to add a "Tools Used" attribute, which will store the tool(s) used to make measurements, but are not sure where this attribute belongs. We do know that one person might use multiple tools for different trees, and also that more than one tool may be used to measure the same tree (for example, different tools may be necessary to measure the same tree in the summer, than in the winter.) Where could we add this attribute? Choose *one or more* answers from the 3 entities and 2 relationships (we would add the attribute only once, but if you believe there are multiple possible places to add the attribute, we ask you to identify all potential candidates):
   a. the entity Person
   b. the relationship Has
   c. the entity Measurement
   d. the relationship Takes
   e. the entity Tree
3. There are two roles that people can have, a Ranger and a Volunteer. Using what you know of subclassing, add these as two new entities in the ER Diagram. Add a reasonable attribute to each of the new entities. Just redraw the relevant part of the diagram that needs to change. Can we eliminate the **role** attribute from*Person*?

**Solution**

1. It would not be a good key because there may be many measurements made on the same day. Therefore this attribute is not unique, and would not make a good key.
2. The relationship Has, the entity Measurement, and the relationship Takes are all correct answers (b, c, d). The entity Person and the entity Tree are not (a, e).
3. See Diagram. Any appropriate attributes to the subclasses can be accepted. You can eliminate the role type attribute. However, we may not want to, as keeping the payment type attribute may facilitate some queries, and may make creating new payment types easier.



# Problem 3: ER Diagram Translation

Use the original ER Diagram in Figure 1 for this problem (without any changes you may have made during your work in problem 2).

1. Translate the ER diagram to a relational design. Try to minimize the number of relations your solution has, and merge relations where appropriate. Don't forget to specify the keys.
2. Assume now that we have changed the ER diagram so that the Takes relationship is now many-many, with the interpretation that it might take several people to determine a single measurement. How will this change the relational design?

**Solution:**

1. Person(UserID, Name, Role, Address), Tree(TreeID, Latitude, Longitude, Species), Measurement(MeasureID, Date_of_Measure, Height, Diameter, TreeID, UserID) (because Measurements had a 1-many relationship with both Person and Tree, the Measurement table can store those relationships).
2. Now the many-many relationship needs its own table, Takes(UserID, MeasureID) and the UserID column would be removed from the Measurement table. *6 points*

# Problem 4: SQL Statements

**Grading: 25 points**

Use the original ER Diagram in Figure 1 for this problem (without any changes you may have made during your work in problems 2 or 3).

1. Write the SQL command that would define each table. Your definition must include correct data types with correct sizes for each field, key and unique declarations, and NULL constraints for fields. Please assume the following data types and lengths:
    a. the three ID attributes are all of type char, length 10
    b. Latitude and Longitude are of type decimal, precision value 10, scale value 6
    c. Height and Diameter precision value 10, scale value 6
    d. DateOfMeasure is of type date
    e. the rest of the attributes are all of type varchar, length 50
2. Write the SQL Command to insert one tuple of data into each table. The data you insert can be of your choosing, but must adhere to the data types and constraints of the tables.
3. We have decided that the Address property is no longer relevant. We do however wish to be able to email staff members. Change the Person table to remove the Address field and add an Email field.
4. Write a SQL Statement to add an Email address to the person you inserted into the Person table, now that the table has an Email field and not an Address field.

**Solution:**

1. CREATE TABLE Person(UserID char(10) PRIMARY KEY, Name varchar(50) NOT NULL, Address varchar(50) NOT NULL, Role varchar(50));
CREATE TABLE Tree(ID char(10) PRIMARY KEY, Species varchar(50) NOT NULL, Latitude Decimal(10,6) NOT NULL, Longitude Decimal(10,6) NOT NULL);
CREATE TABLE Measurement(MeasureID char(10) PRIMARY KEY, DateOfMeasure date NOT NULL, Height Decimal(10,2) NOT NULL, Diameter Decimal(10,2) NOT NULL);
*9 points, 3 points per Create statement. Only the primary keys need to be specified as NOT NULL, the rest are ok either way*

2. INSERT INTO Person (UserID, Name, Address, Role) VALUES ('12345', 'John Smith', '123 Science Lane, Urbana, IL', 'Ranger');
INSERT INTO Tree(ID, Species, Latitude, Longiture) VALUES ('987', 'Fir', 9.353234, 39.234562);
INSERT INTO Measurement(MeasureID, DateOfMeasure, Height, Diameter) VALUES ('432', '2011-01-30', 15.50, 3.10);
*6 points, 2 points per Insert Statement*

3. ALTER TABLE Person DROP Address;
ALTER TABLE Person ADD Email varchar(50);
*5 points*

4. UPDATE Person SET Email = 'smithj@treeplace.com' WHERE UserID = '12345';