How about doing an ER design interactively?

Suggest an application to be modeled.

Making E/R Models

- To make an E/R model you need to identify
  - Enitities
  - Attributes
  - Relationships
  - Cardinality ratios
- from a description

- General guidelines
  - Since entities are things or objects they are often nouns in the description
  - Attributes are facts or properties, and so are often nouns also
  - Verbs often describe relationships between entities

Example

A university consists of a number of departments. Each department offers several courses. A number of modules make up each course. Students enrol in a particular course and take modules towards the completion of that course. Each module is taught by a lecturer from the appropriate department, and each lecturer tutors a group of students

Example - Entities

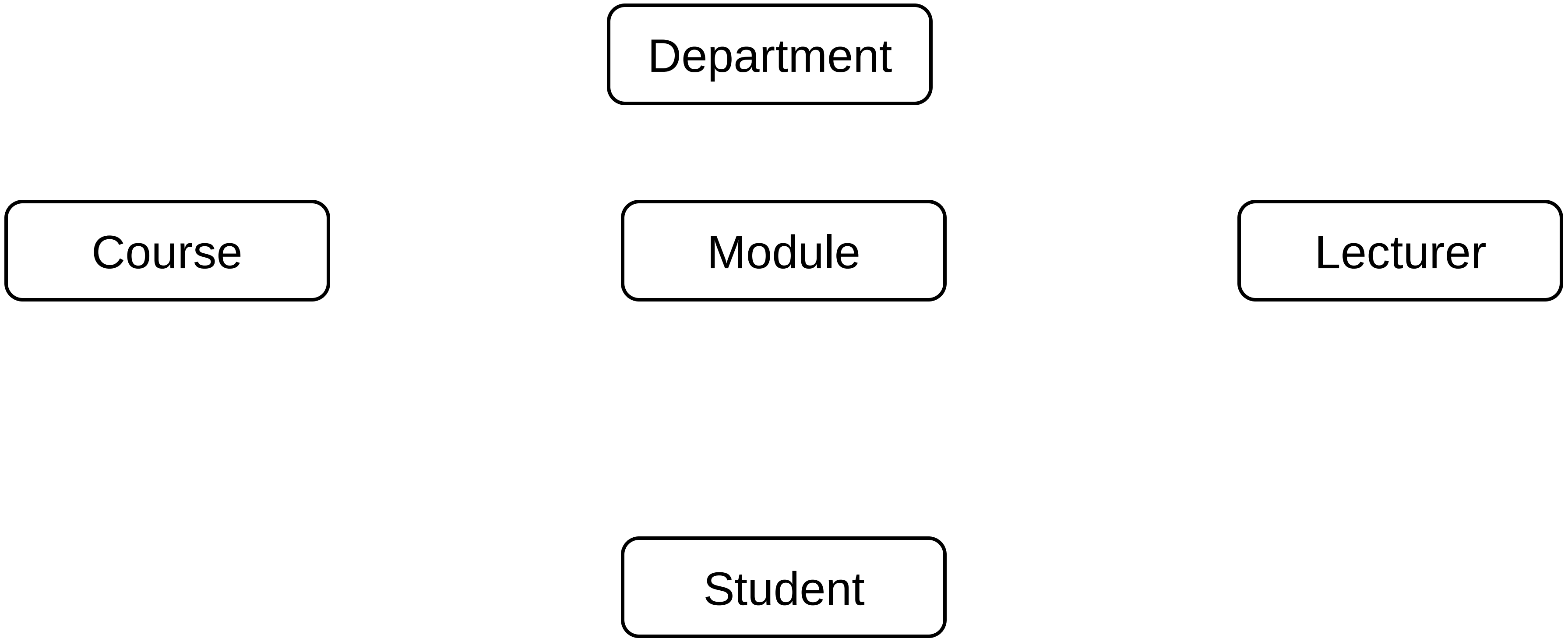
A university consists of a number of **departments**. Each department offers several **courses.** A number of **modules** make up each course. **Students** enrol in a particular course and take modules towards the completion of that course. Each module is taught by a **lecturer** from the appropriate department, and each lecturer tutors a group of students

Example - Relationships

- A university consists of a number of departments. Each department **offers** several courses. A number of modules **make up** each course. Students **enrol in** a particular course and **take** modules towards the completion of that course. Each module is **taught by** a lecturer **from the** appropriate department, and each lecturer **tutors** a group of students

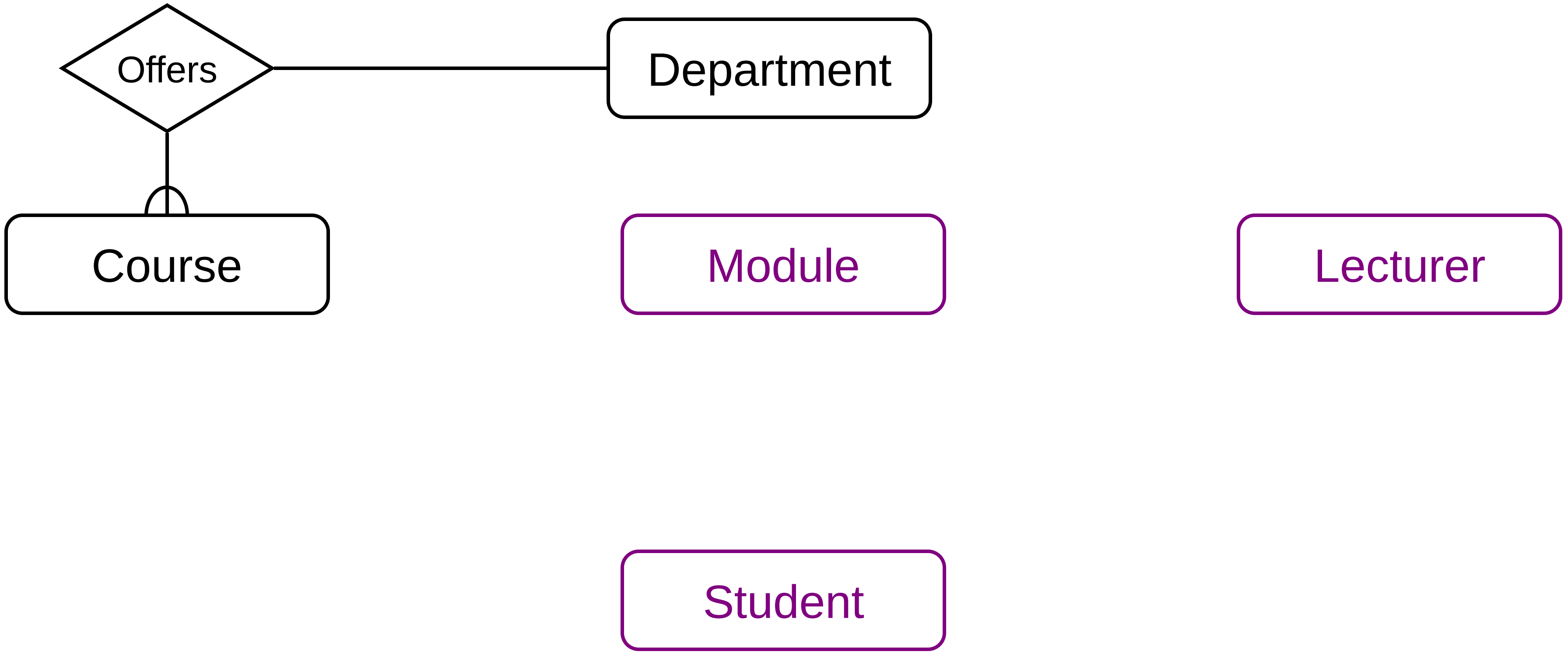
Example - E/R Diagram

Entities: Department, Course, Module, Lecturer, Student



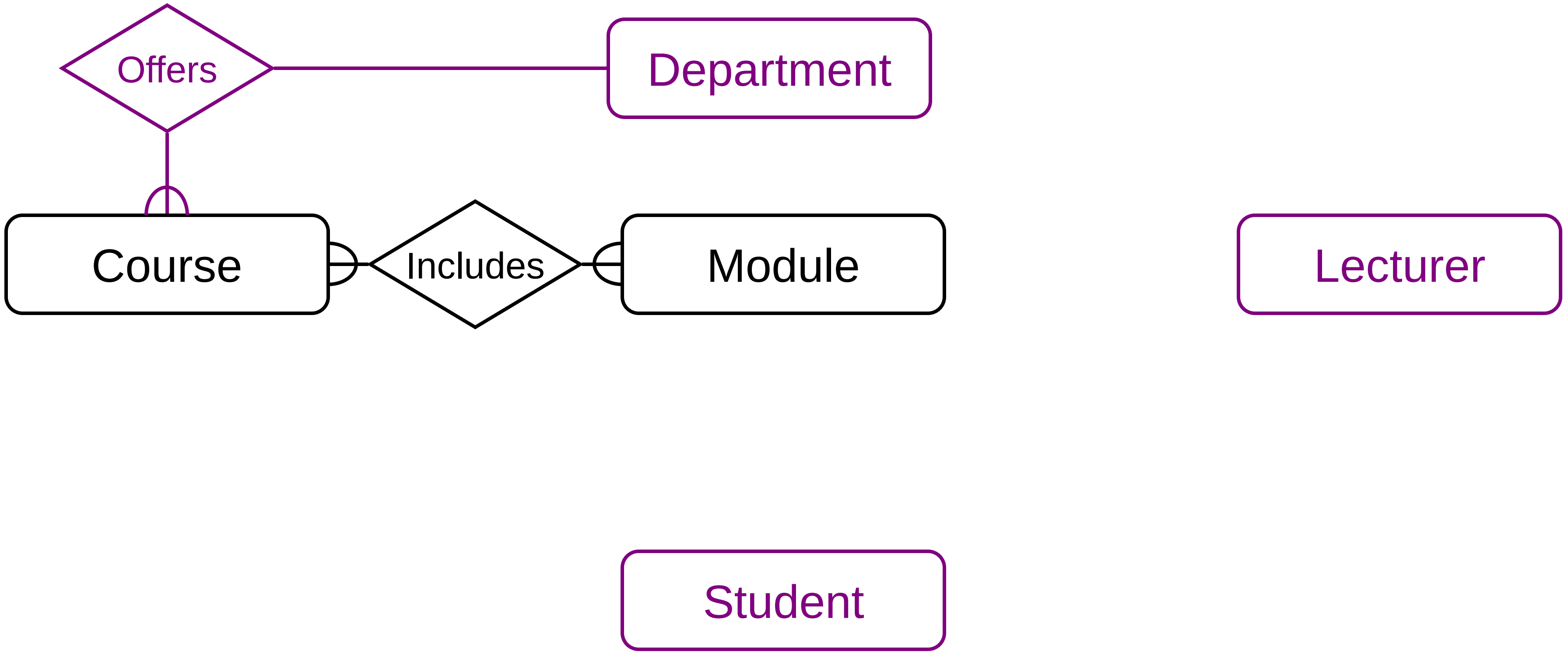
Example - E/R Diagram

Each department offers several courses



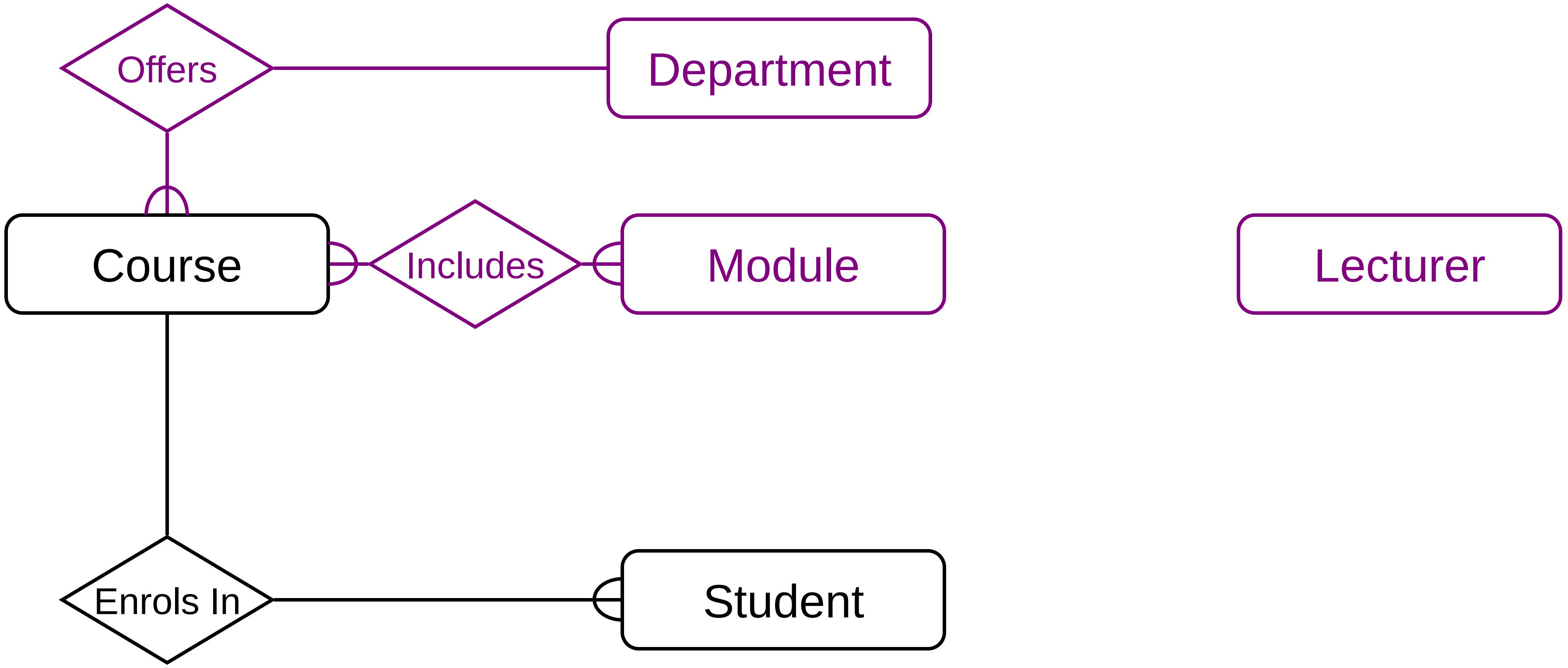
Example - E/R Diagram

A number of modules make up each courses



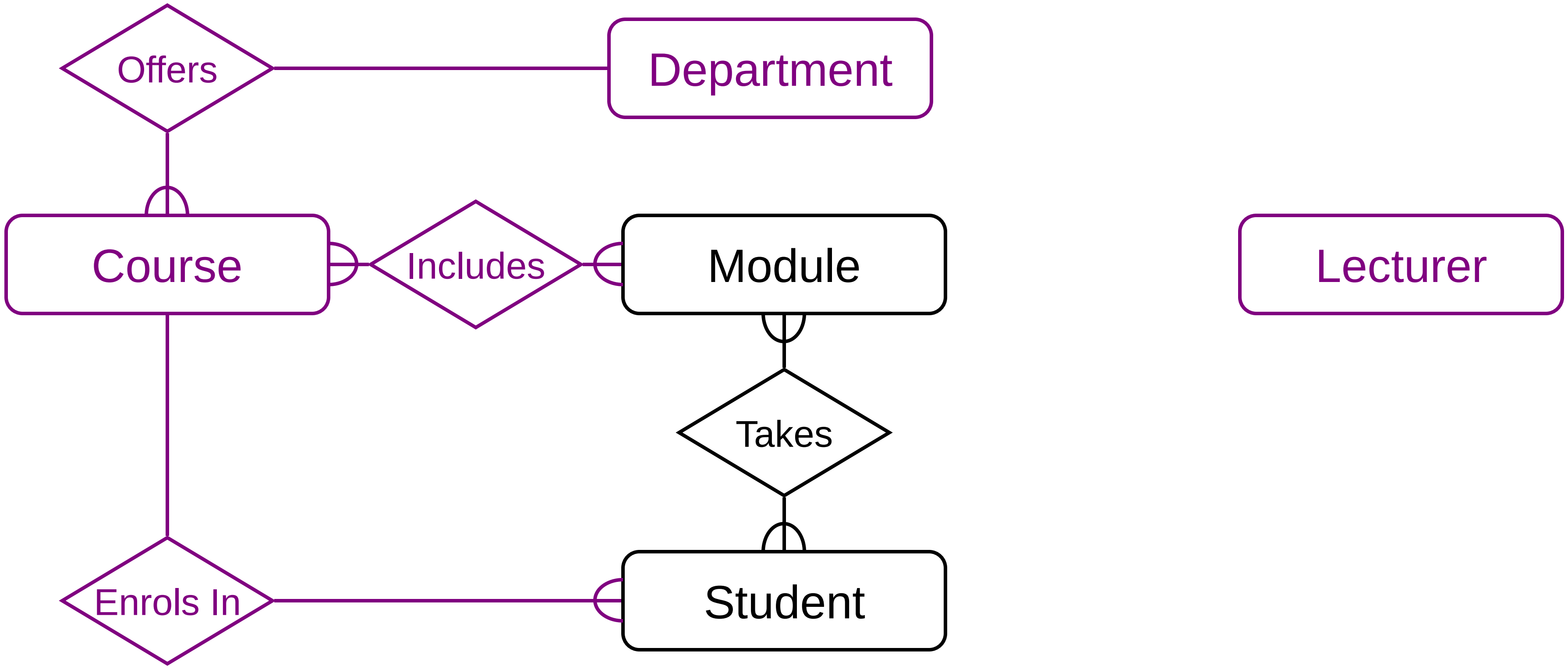
Example - E/R Diagram

Students enrol in a particular course



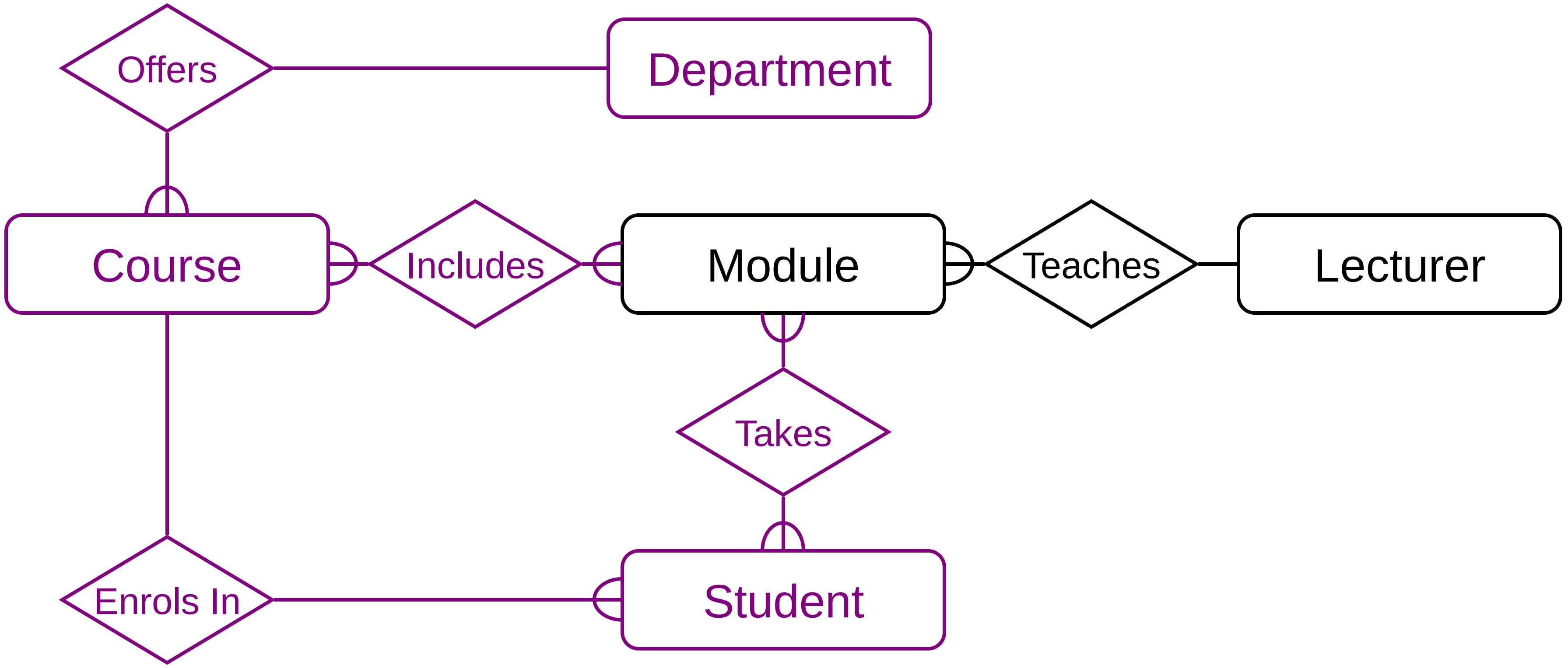
Example - E/R Diagram

Students … take modules



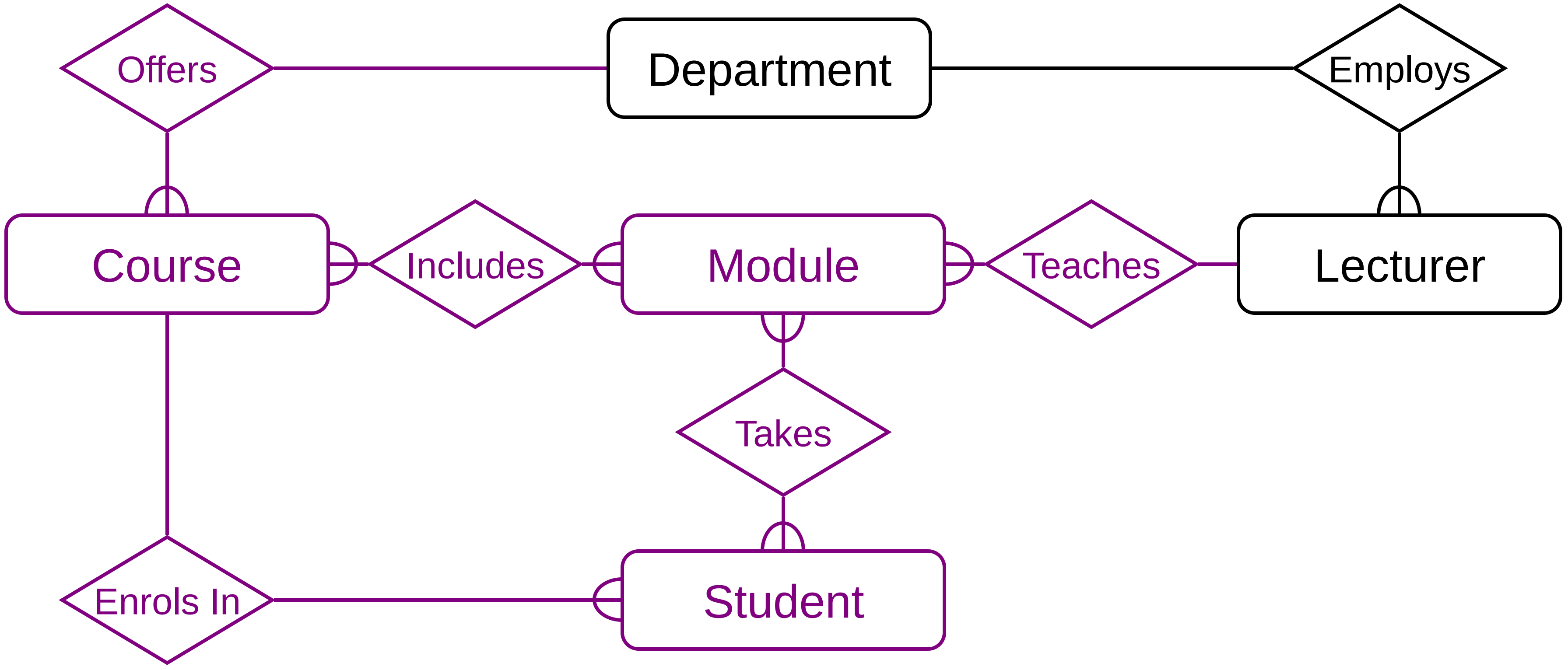
Example - E/R Diagram

Each module is taught by a lecturer



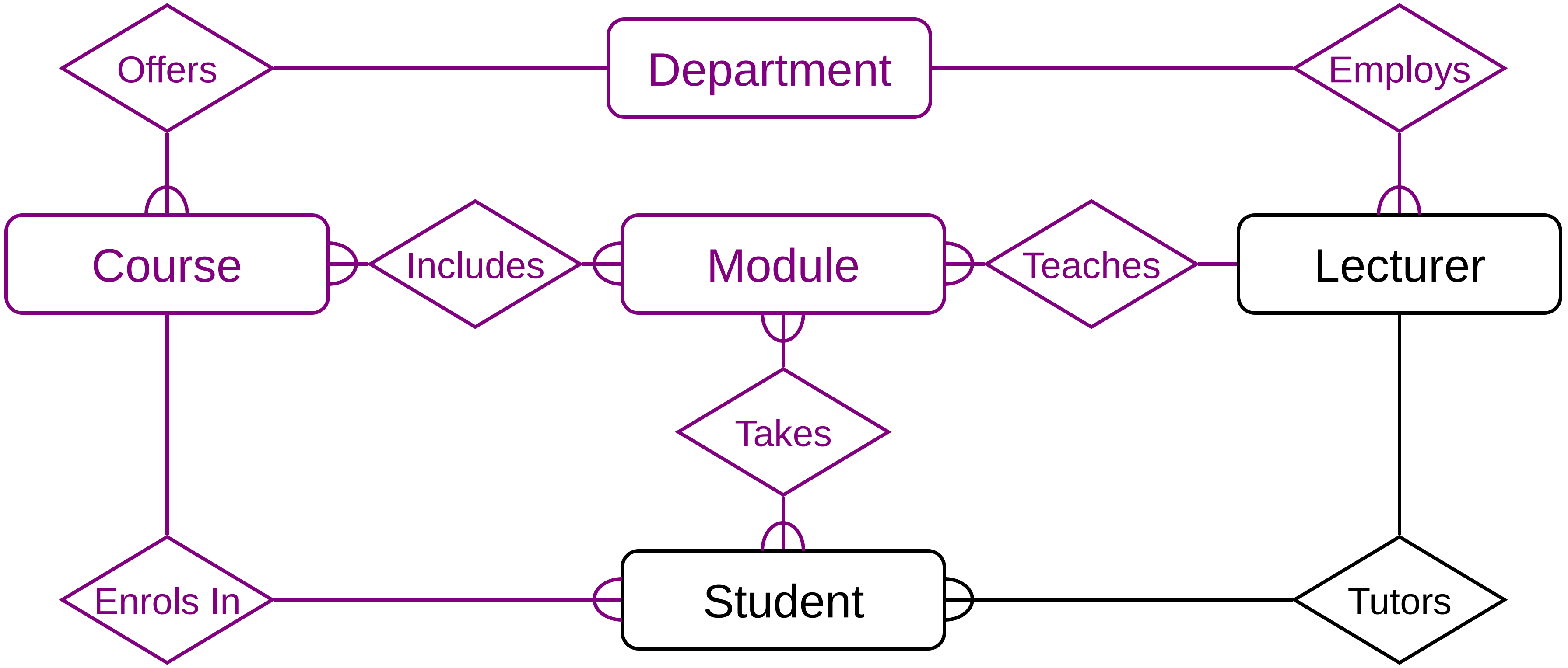
Example - E/R Diagram

a lecturer from the appropriate department

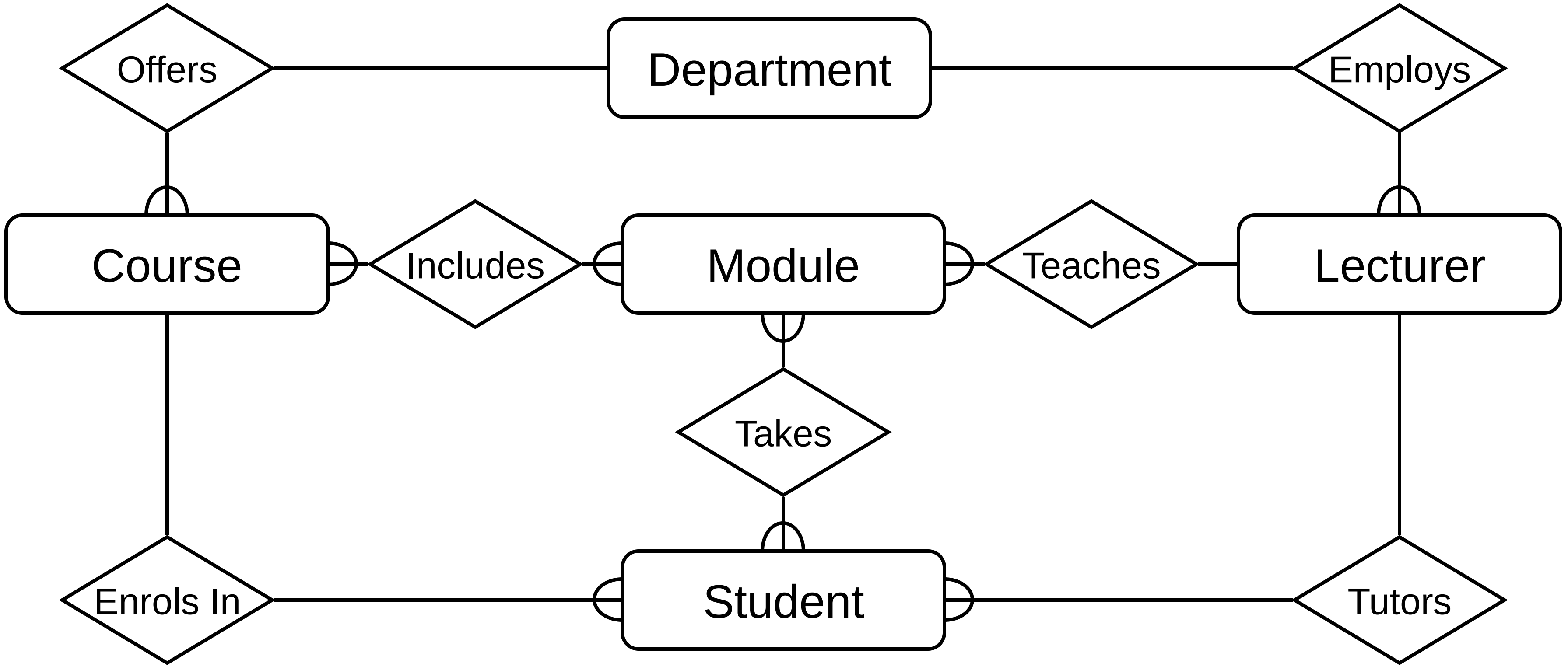


Example - E/R Diagram

each lecturer tutors a group of students



Example - E/R Diagram



Entities and Attributes

- Sometimes it is hard to tell if something should be an entity or an attribute
  - They both represent objects or facts about the world
  - They are both often represented by nouns in descriptions
- General guidelines
  - Entities can have attributes but attributes have no smaller parts
  - Entities can have relationships between them, but an attribute belongs to a single entity

Example

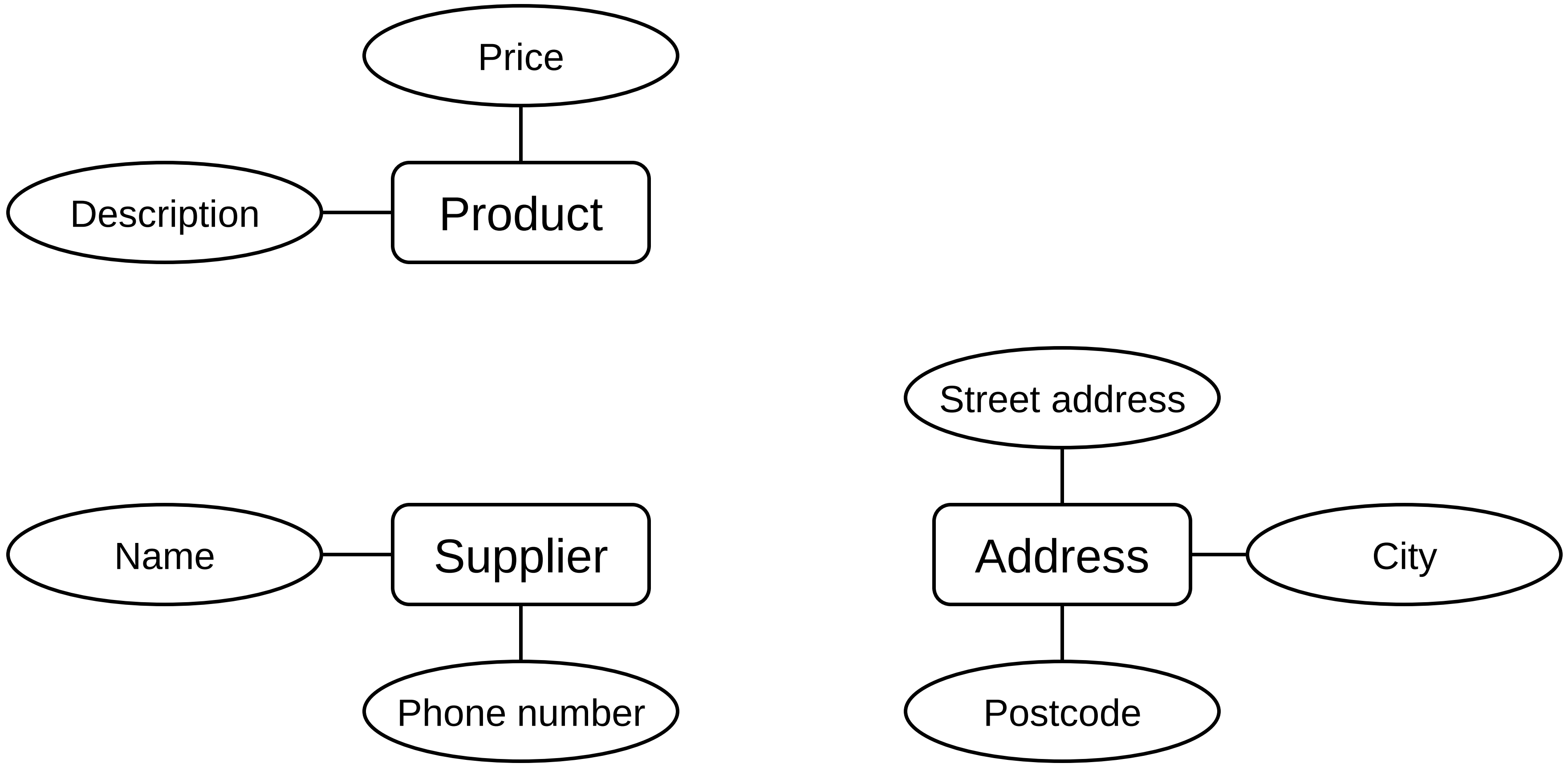
We want to represent information about products in a database. Each product has a description, a price and a supplier. Suppliers have addresses, phone numbers, and names. Each address is made up of a street address, a city, and a postcode.

Example - Entities/Attributes

- Entities or attributes:
  - product
  - description
  - price
  - supplier
  - address
  - phone number
  - name
  - street address
  - city
  - postcode

- Products, suppliers, and addresses all have smaller parts so we can make them entities
- The others have no smaller parts and belong to a single entity

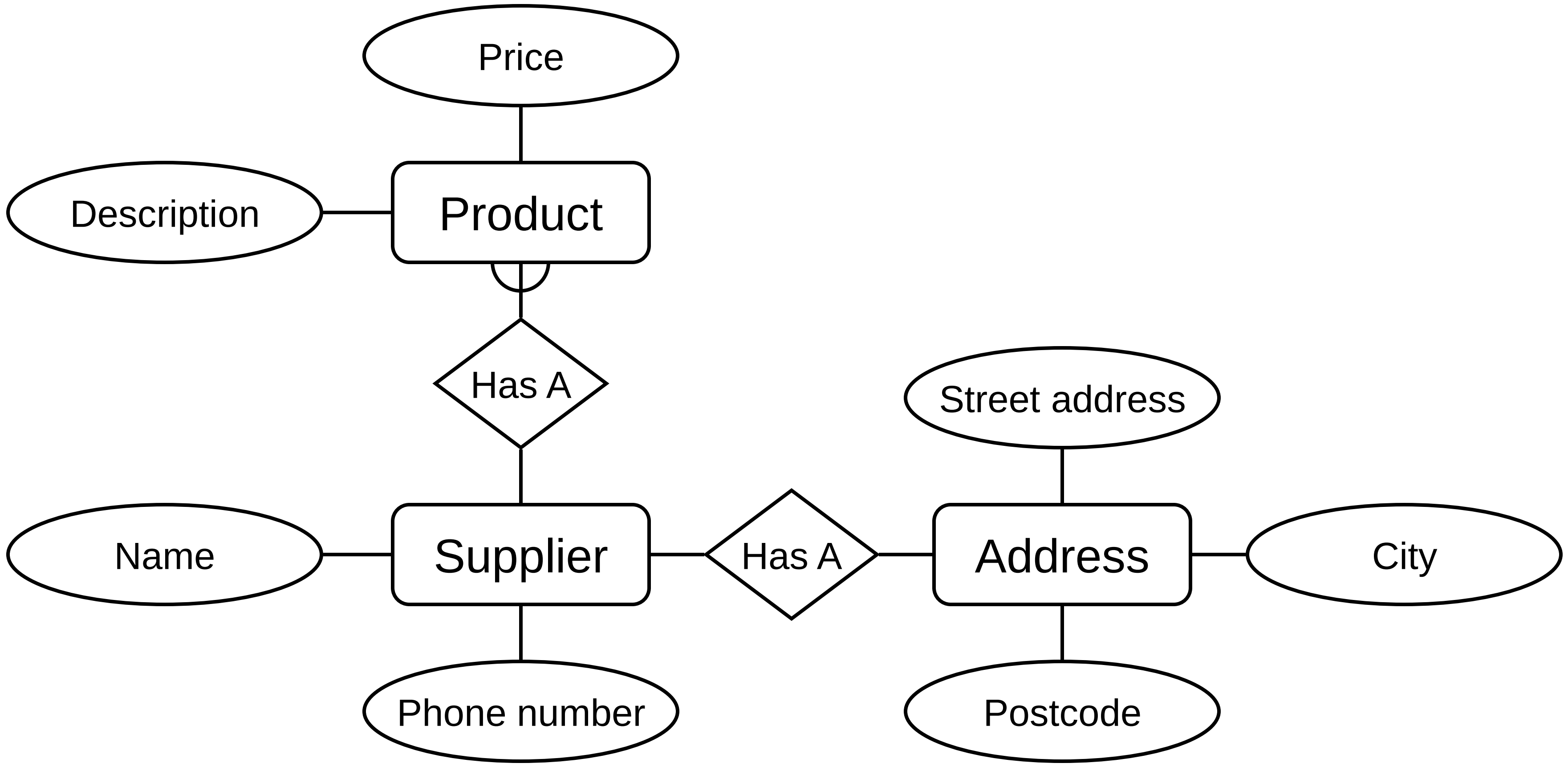
Example - E/R Diagram



Example - Relationships

- Each product has a supplier
  - Each product has a single supplier but there is nothing to stop a supplier supplying many products
  - A many to one relationship
- Each supplier has an address
  - A supplier has a single address
  - It does not seem sensible for two different suppliers to have the same address
  - A one to one relationship

Example - E/R Diagram



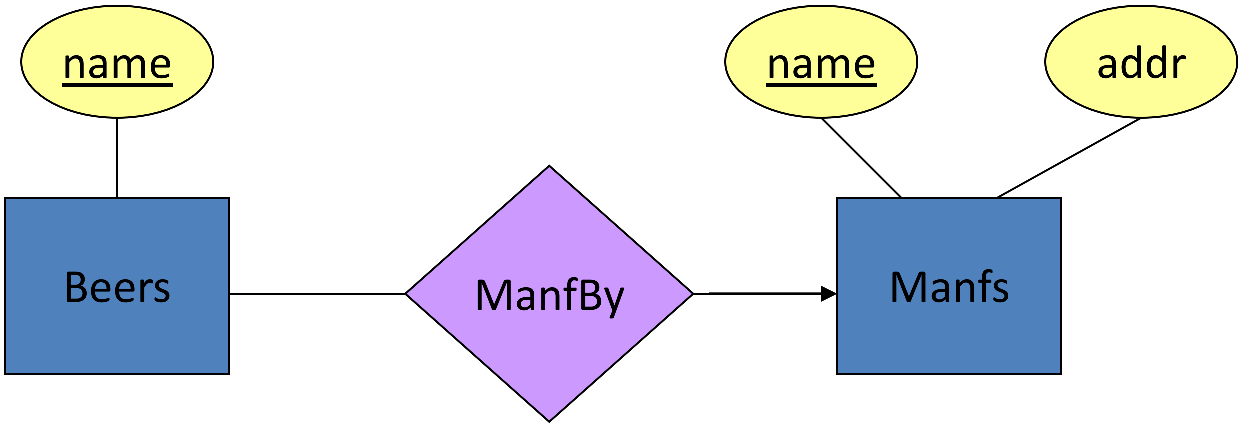
Design Techniques: Some Tips

1. Avoid redundancy.
2. Limit the use of weak entity sets.
3. Don’t use an entity set when an attribute will do.

Avoiding Redundancy

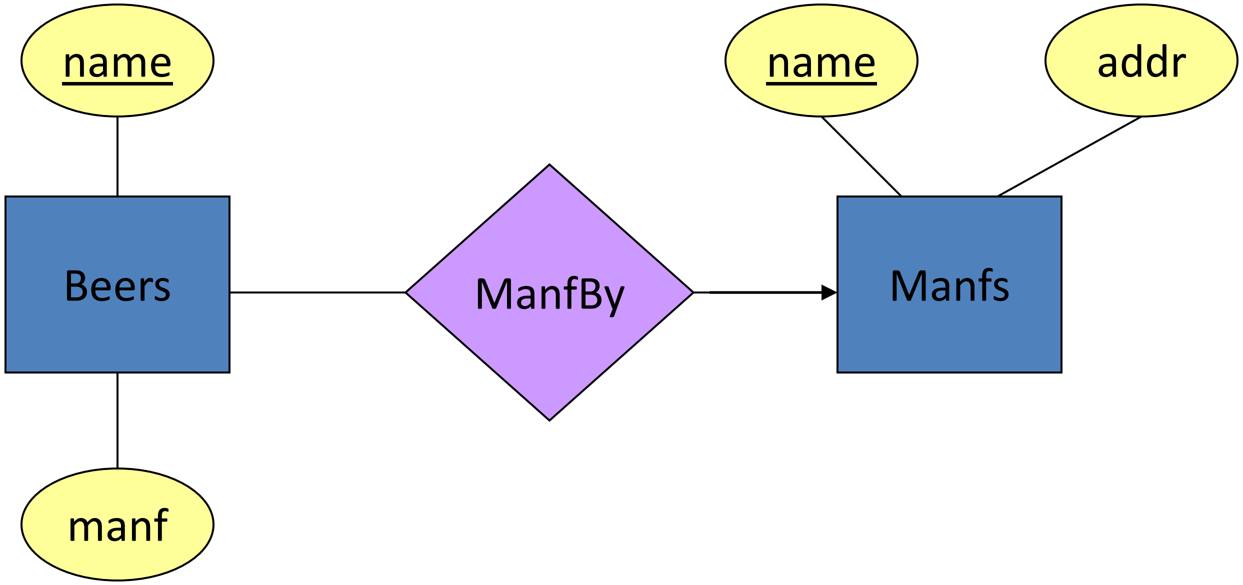
- *Redundancy* = saying the same thing in two (or more) different ways.
- Wastes space and (more importantly) encourages inconsistency.
  - Two representations of the same fact become inconsistent if we change one and forget to change the other.
  - Recall anomalies due to FD's.

Example: Good



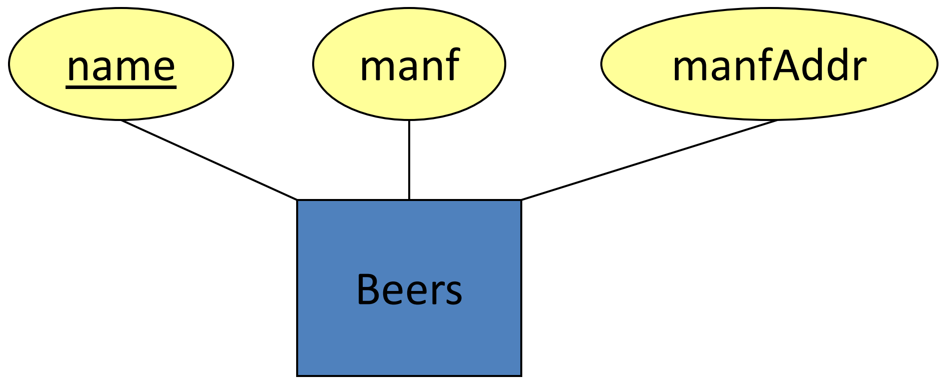
This design gives the address of each manufacturer exactly once.

Example: Bad



This design states the manufacturer of a beer twice: as an attribute and as a related entity.

Example: Bad



This design repeats the manufacturer's address once for each beer and loses the address if there are temporarily no beers for a manufacturer.

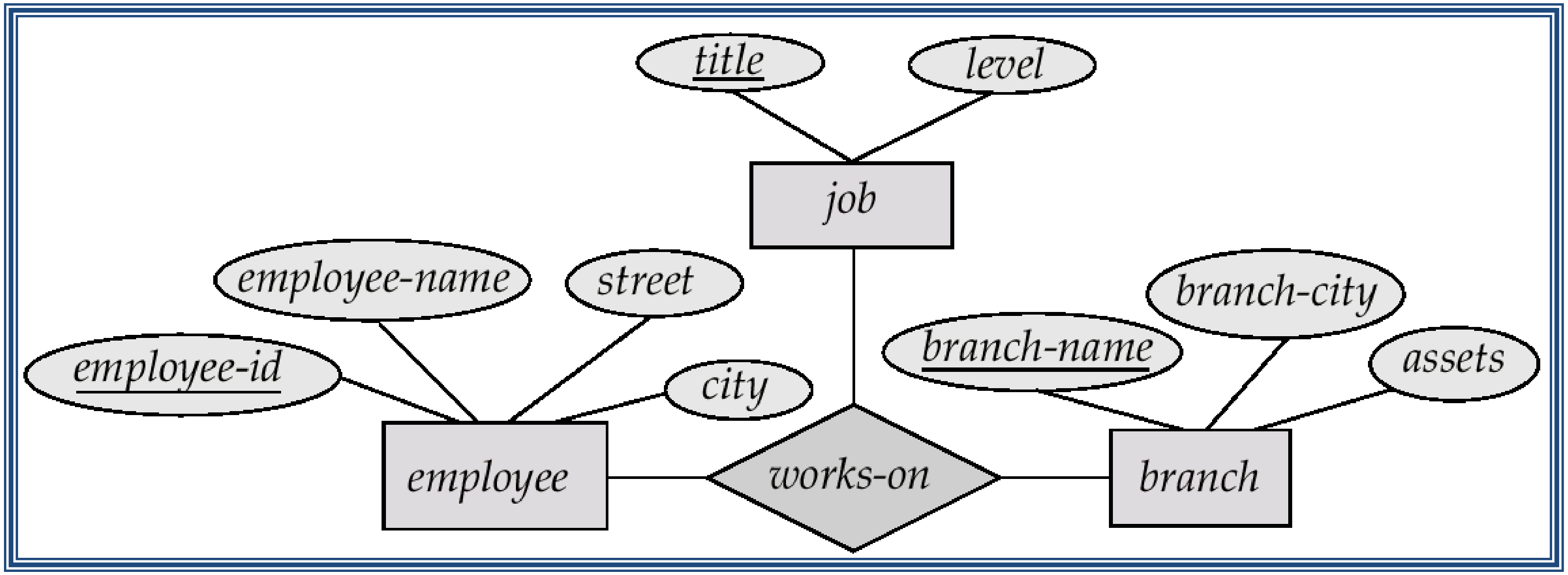
Keys

- A *super key* of an entity set is a set of one or more attributes whose values uniquely determine each entity.
- A *candidate key* of an entity set is a minimal super key
  - *Customer-id* is candidate key of *customer*
  - *account-number* is candidate key of *account*
- Although several candidate keys may exist, one of the candidate keys is selected to be the *primary key*.

Keys for Relationship Sets

- The combination of primary keys of the participating entity sets forms a super key of a relationship set.
  - (*customer-id, account-number*) is the super key of *depositor*
  - *NOTE: this means a pair of entity sets can have at most one relationship in a particular relationship set.*
    - E.g. if we wish to track all access-dates to each account by each customer, we cannot assume a relationship for each access. We can use a multivalued attribute though
- Must consider the mapping cardinality of the relationship set when deciding the what are the candidate keys
- Need to consider semantics of relationship set in selecting the *primary key* in case of more than one candidate key

E-R Diagram with a Ternary Relationship



Cardinality Constraints on Ternary Relationship

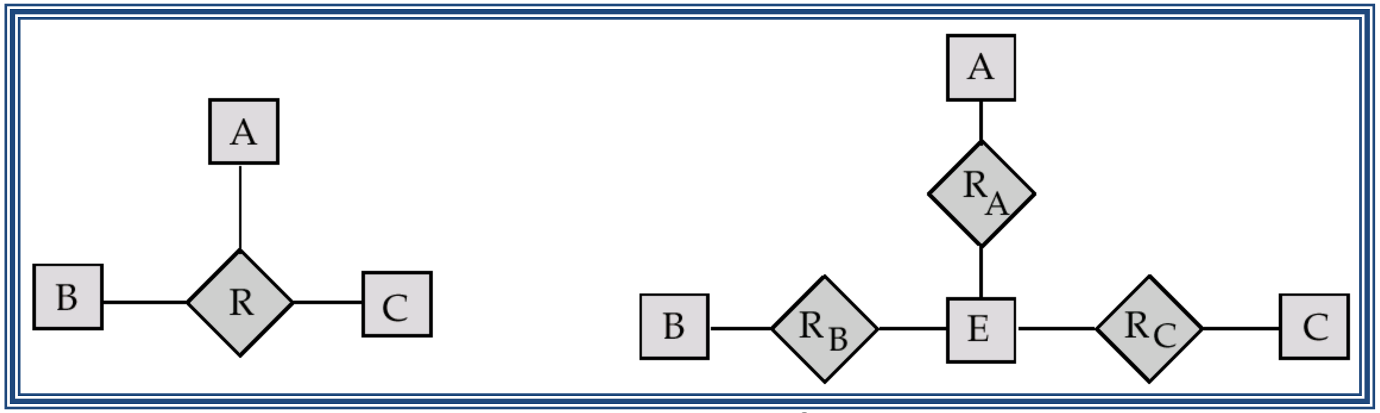
- We allow at most one arrow out of a ternary (or greater degree) relationship to indicate a cardinality constraint
- E.g. an arrow from *works-on* to *job* indicates each employee works on at most one job at any branch.
- If there is more than one arrow, there are two ways of defining the meaning.
  - E.g a ternary relationship *R* between *A*, *B* and *C* with arrows to *B* and *C* could mean
  - 1. each *A* entity is associated with a unique entity from *B* and *C* or
  - 2. each pair of entities from (*A, B*) is associated with a unique *C* entity, and each pair (*A, C*) is associated with a unique *B*
  - Each alternative has been used in different formalisms
  - To avoid confusion we outlaw more than one arrow

Binary Vs. Non-Binary Relationships

- Some relationships that appear to be non-binary may be better represented using binary relationships
  - E.g. A ternary relationship *parents*, relating a child to his/her father and mother, is best replaced by two binary relationships, *father* and *mother*
    - Using two binary relationships allows partial information (e.g. only mother being know)
  - But there are some relationships that are naturally non-binary
    - E.g. *works-on*

Converting Non-Binary Relationships to Binary Form

- In general, any non-binary relationship can be represented using binary relationships by creating an artificial entity set.
  - Replace *R* between entity sets A, B and C by an entity set *E*, and three relationship sets:
    1. R_A, relating *E* and *A*
    2. R_B, relating *E* and *B*
    3. R_C, relating *E* and *C*
  - Create a special identifying attribute for *E*
  - Add any attributes of *R* to *E*
  - For each relationship (a_i , b_i , c_i) in *R,* create
    1. a new entity e_i in the entity set *E*
    2. add (e_i , a_i) to R_A
    3. add (e_i , b_i) to R_B
    4. add (e_i , c_i) to R_C



Converting Non-Binary Relationships (Cont.)

- Also need to translate constraints
  - Translating all constraints may not be possible
  - There may be instances in the translated schema that cannot correspond to any instance of *R*
    - *Exercise: add constraints to the relationships* R_A, R_B *and* R_C to ensure that a newly created entity corresponds to exactly one entity in each of entity sets *A, B* and *C*
  - We can avoid creating an identifying attribute by making E a weak entity set (described shortly) identified by the three relationship sets

Design Issues

- Use of entity sets vs. attributes
  Choice mainly depends on the structure of the enterprise being modeled, and on the semantics associated with the attribute in question.
- Use of entity sets vs. relationship sets
  Possible guideline is to designate a relationship set to describe an action that occurs between entities
- Binary versus *n*-ary relationship sets
  Although it is possible to replace any nonbinary (*n*-ary, for $n > 2$) relationship set by a number of distinct binary relationship sets, a *n*-ary relationship set shows more clearly that several entities participate in a single relationship.
- Placement of relationship attributes