Conceptual Modeling

Developing Database-Driven Applications

- requirements specification
  - *functional requirements* describe what operations the user wants to perform
  - *data requirements* describe what information is needed, and what relationships exist between those pieces of information
  - *operational requirements* describe aspects of the system's performance e.g. availability, reliability, response time, security
- design
  - of the database
  - of the user interface for the application
  - of the application itself
- implementation
  - of the database – create the DB structure, configure access, create stored routines, populate the DB, etc
  - of the application

Data Modeling

A *data model* is a set of concepts for describing the structure of data.

Data Modeling

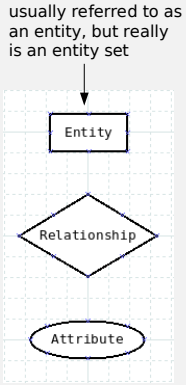
Data modeling for databases typically proceeds through three levels.

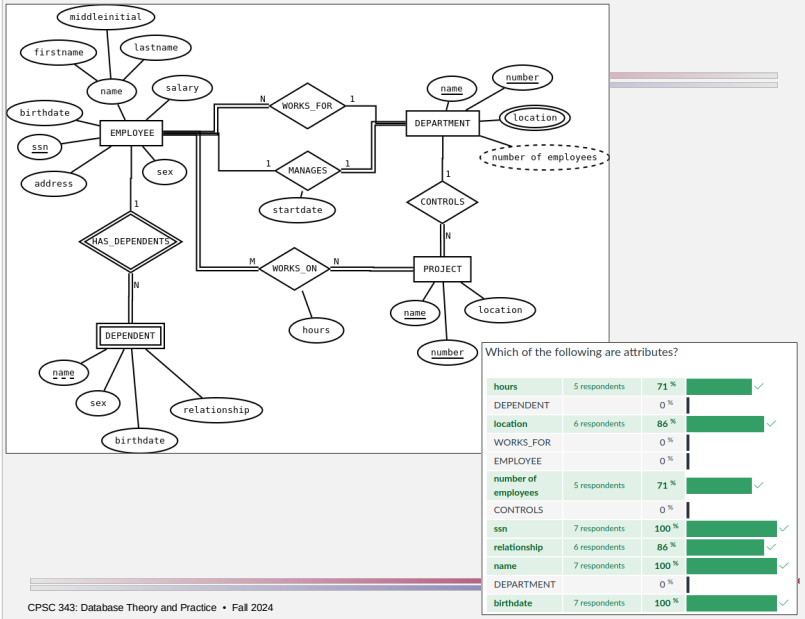
- *conceptual data model* has high level concepts to identify what the things are and how they relate to each other
  - concepts are close to those in the problem domain
  - implementation independent
  - we will study the entity-relationship (ER) model
- *logical data model* has concepts representing the structures available in the database
  - structure of the data is translated into something implementable in the database
  - interactions with the database (queries) can be thought of at this level of abstraction
  - we will study the relational model
- *physical data model* deals with how things will actually be built in the database
  - implementation details
  - we will use MySQL

Entity-Relationship Model

Core concepts and notation:

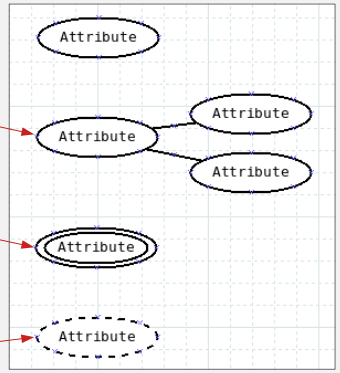
- an *entity* represents some abstract object
  - like a book, borrower, or library branch
- an *entity set* represents a collection of the same kind of entity
- a *relationship* represents a connection between two or more entity sets
- an *attribute* is a property of an entity set or relationship
  - like a title, card number, or branch name





Types of Attributes

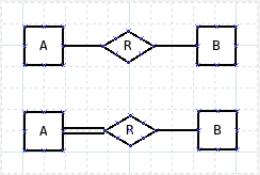
- simple or composite
  - is the attribute atomic, or can it be divided into meaningful subparts?
- single-valued or multivalued
  - can an entity have more than one value for the attribute at a time?
- stored or derived
  - can the attribute's value be determined from other information in the database?



Participation

Participation constraints dictate the minimum number of relationship instances a given entity participates in.

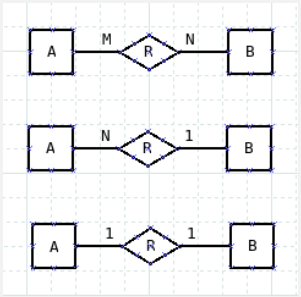
- normally this can be 0 (“may”)
- *total participation* requires ≥ 1 (“must”)



Cardinality

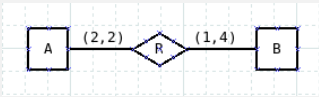
Cardinality constraints dictate the maximum number of relationship instances a given entity participates in.

- M:N (many-many)
  - one entity of A can participate in R with any number of entities of B, and vice versa
- N:1 (many-one)
  - one entity of A can participate in R with at most one entity of B
  - one entity of B can participate in R with any number of entities from A
- 1:1 (one-one)
  - one entity of A can participate in R with at most one entity of B, and vice versa



Participation and Cardinality

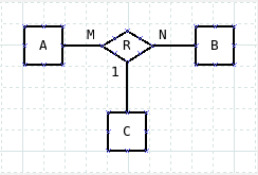
If you need more than may/must and one/many, an alternate notation allows explicit specification of lower and upper bounds.



- an entity of A participates in at least 1 and at most 4 relationship instances with an entity of B
- an entity of B participates in exactly 2 relationship instances with an entity of A

Multiway Relationships

Relationships can involve more than two entity sets.



a particular pair of entities from A and B can participate in R with at most one entity from C

a particular pair of entities from A, C or from B, C can participate in R with any number of entities from the third entity set

Note that not all cardinality constraints involving the various entity sets can be represented with this notation.

- e.g. can't express that one entity from A can participate in at most one relationship instance

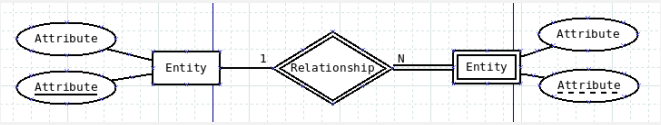
Roles

An entity set may participate more than once in a relationship.

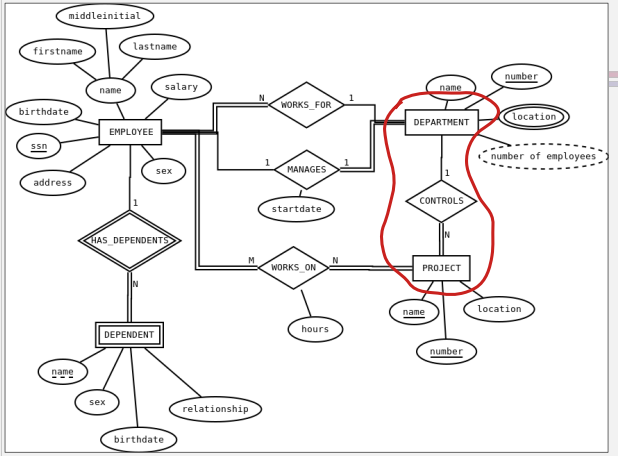
- note: same *entity set*, not necessarily the same *entity*
  - e.g. the EMPLOYEE entity set would participate in the SUPERVISES relationship as both supervisor and employee being supervised, but the same person would not be both supervisor and supervisee in a particular instance of the relationship
- also: this is not talking about cardinality, which deals with how many instances of a relationship a given entity may participate in

Keys and Weak Entities

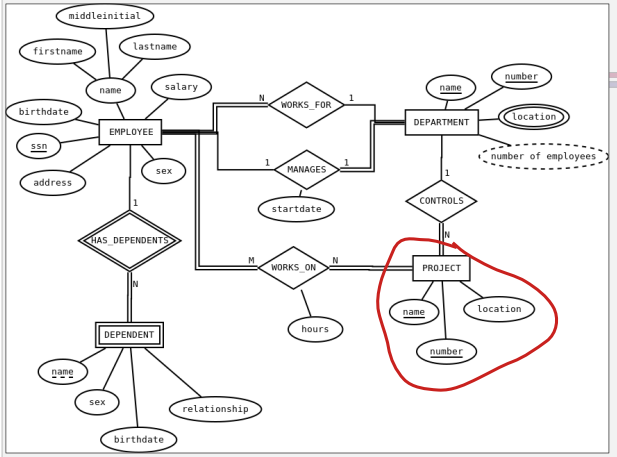
- a *key* is a set of one or more attributes whose values uniquely identify each entity in an entity set
- a *weak entity type* is an entity type whose key includes attributes from another entity type
  - must participate in a many- or one-to-one *identifying relationship* with the other entity type
  - key attributes belonging to the weak entity type form a *partial* or *weak* key



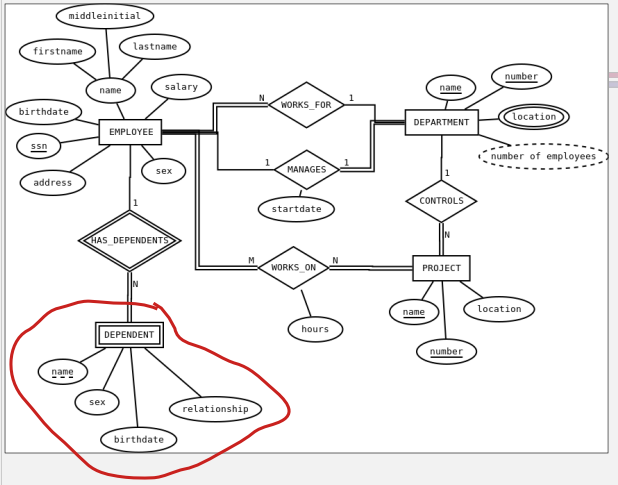
key attribute | identifying relationship | weak entity | weak key attribute



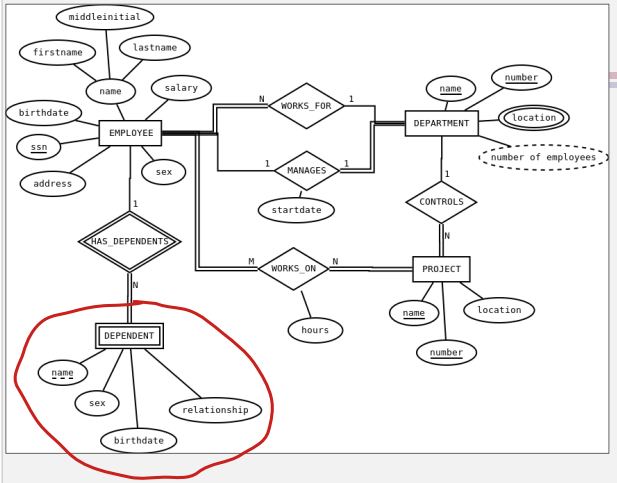
A department must control at least one project. 2 respondents 29 %



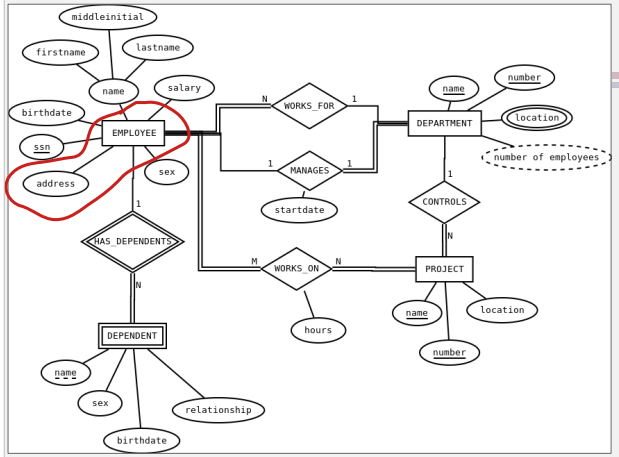
Two projects can have the same name. 2 respondents 29 %



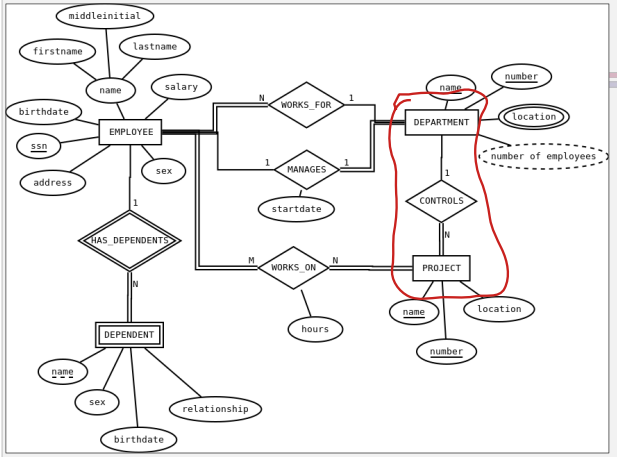
Two dependents can have the same name. 6 respondents 86 %



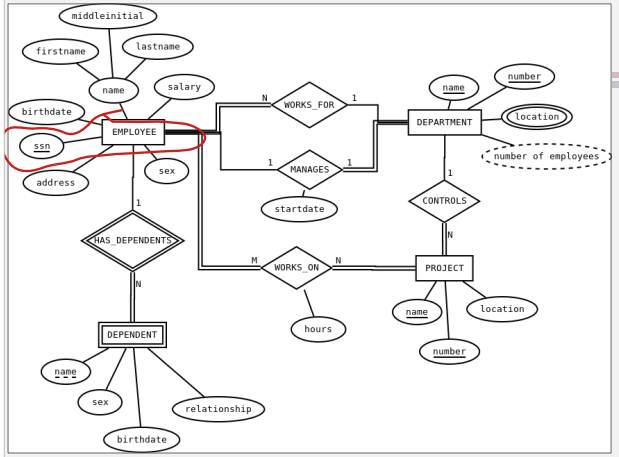
Two dependents of the same employee can have the same name as long as they have different relationships to that employee.	2 respondents	29 %



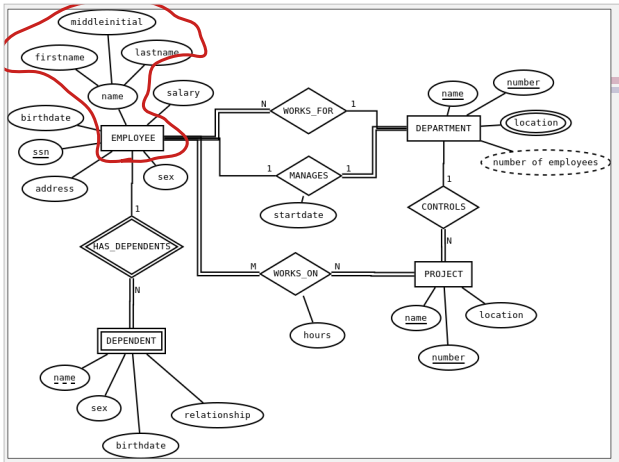
It is possible to find all of the employees living in a given city.	4 respondents	57 %



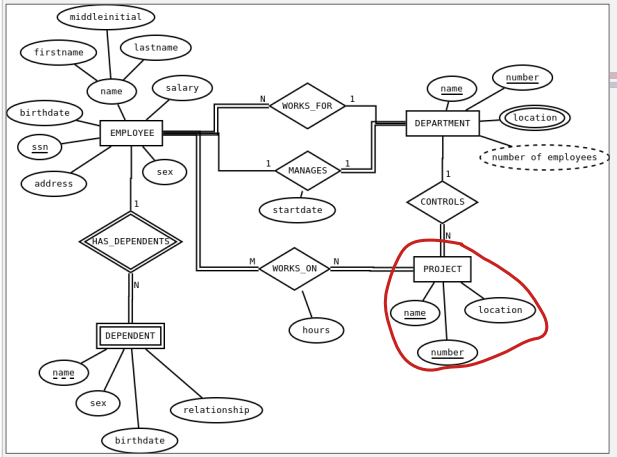
A project can be controlled by more than one department.	1 respondent	14 %



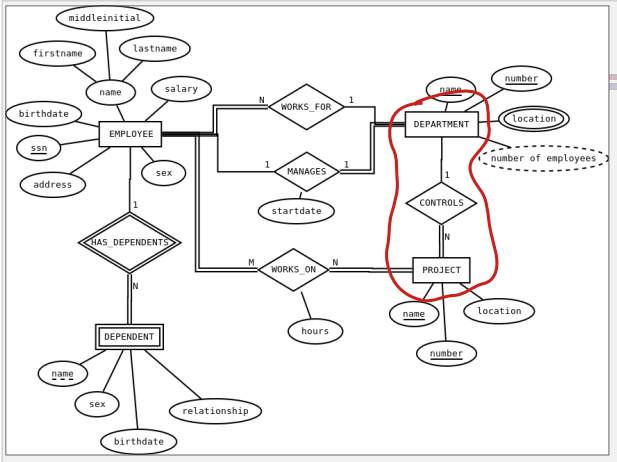
An employee must have a ssn recorded.	6 respondents	86 %



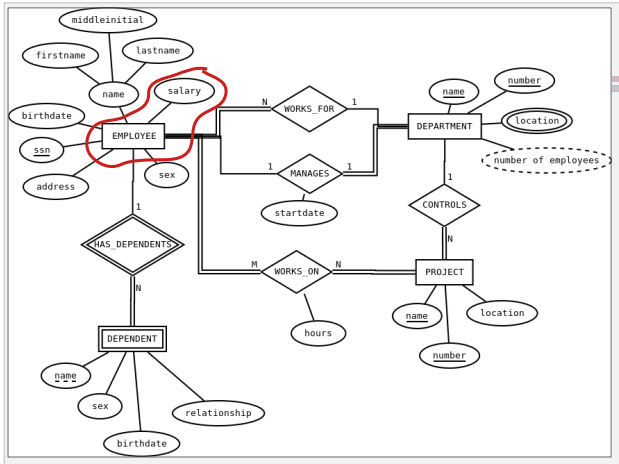
It is possible to find the employees with a given last name. | 5 respondents | 71 %



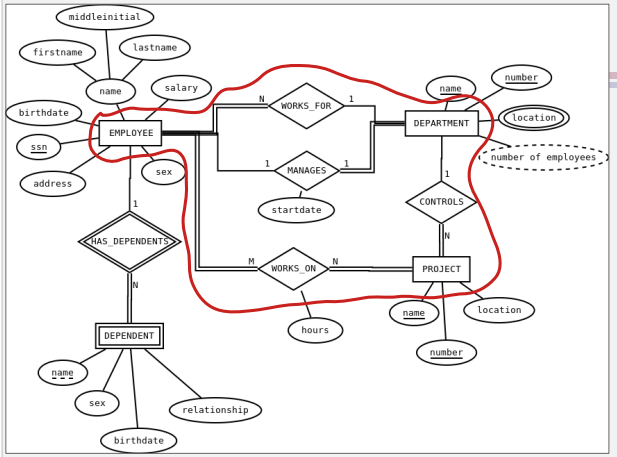
A project can have more than one location. | 1 respondent | 14 %



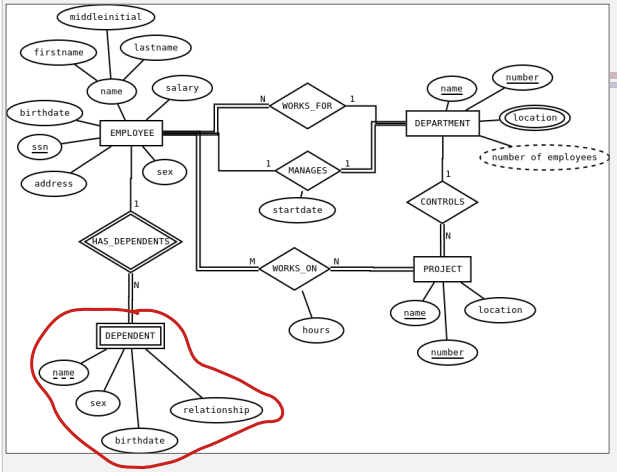
A project must be controlled by a department. | 5 respondents | 71 %



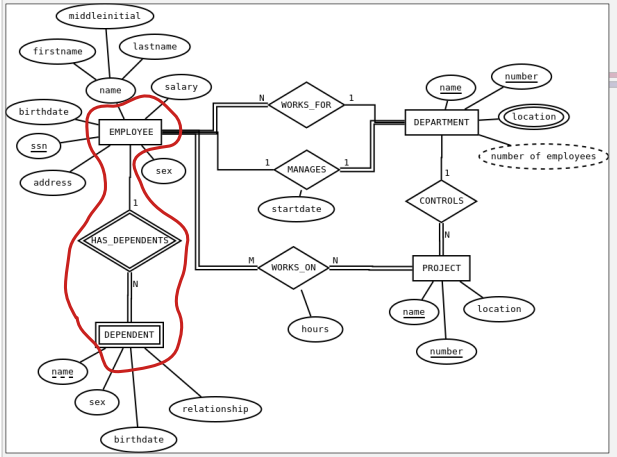
An employee must have a salary recorded. | 3 respondents | 43 %



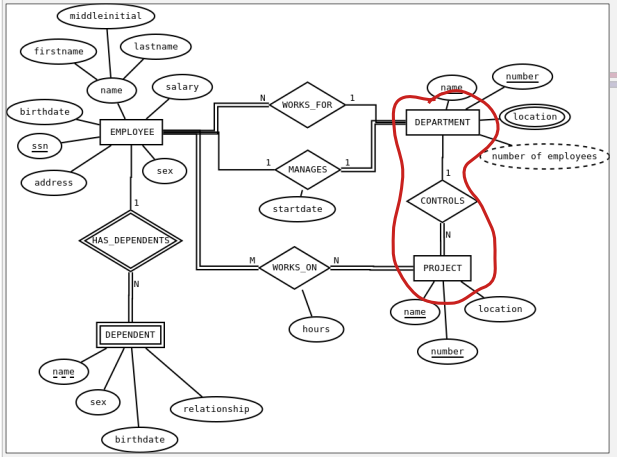
An employee can work on a project controlled by a department that the employee doesn't work for.	3 respondents	43 %	✓



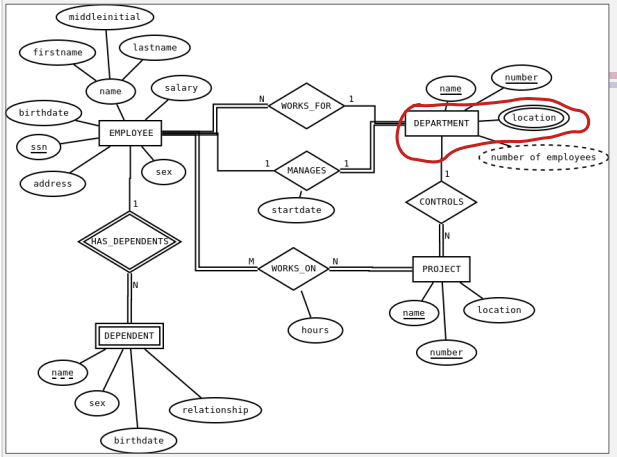
A dependent can be uniquely identified by his or her name.	3 respondents	43 %	



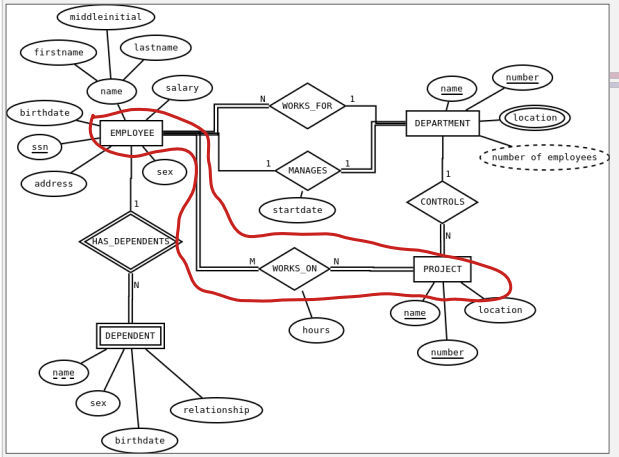
A dependent can be uniquely identified by the employee he or she is a dependent of.	2 respondents	29 %	



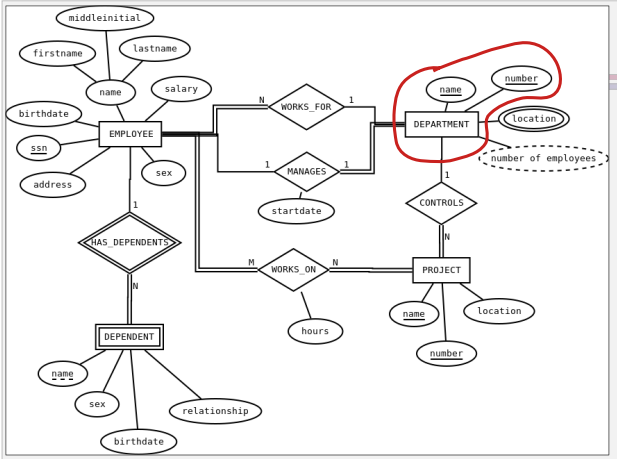
A department can control more than one project.	6 respondents	86 %	✓



A department can have more than one location. | 1 respondent | 14 %



An employee can work on multiple projects. | 6 respondents | 86 %



A department can be uniquely identified by its name or its number. | 6 respondents | 86 %

Questions

Can a relationship also be an entity?

- it may be possible or even reasonable to model something as an entity or as a relationship