HERIOT WATT UNIVERSITY

LEADERS IN IDEAS AND SOLUTIONS

Data Modelling: Entity Relationship Diagrams

Lecture 03 - F21DF 2019/20

HERIOT WATT UNIVERSITY

Outline

- **Database Design**
  - Conceptual Questions
  - Data Requirements
  - Conceptual Data Model
- **Entity Relationship Diagrams**
  - Types of Attributes
  - Keys
  - Entities
  - Relationships
  - Constraints

Slides content from F21DF by Dr. Alasdair Gray

HERIOT WATT UNIVERSITY

DATA MODELLING: WHAT DATA DO YOU NEED TO STORE?

HERIOT WATT UNIVERSITY

Data Modelling

- Different applications need different views
- Views combined to design data model
- User views to support applications:
  - Efficient query processing
  - Security by restricting data access

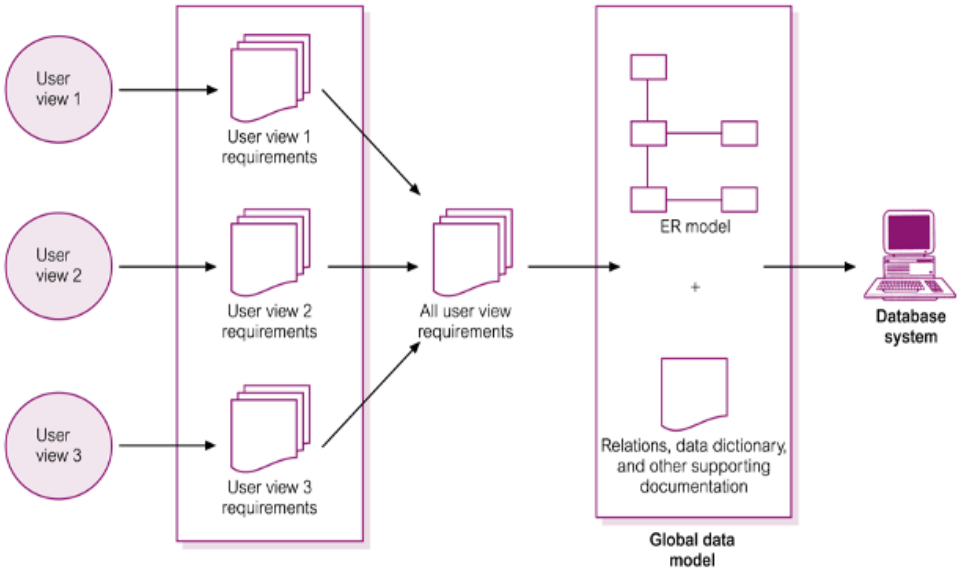


Image from Connolly and Begg (2005)

HERIOT WATT UNIVERSITY

System Definition: User Views

- **A user view defines what is required of a database system from perspective of:**
  - a particular job role (such as Manager or Supervisor) or
  - enterprise application area (such as marketing, personnel, or stock control).
  - For example:
    - A Director will want to run queries and produce reports at an overall level
    - A secretary working in one branch will maintain the data and run queries for that branch only
- **Defining User Views:**
  - Identify required data
  - Write named query (view) to display data
  - Restrict user access by granting read access to the view

HERIOT WATT UNIVERSITY

Database Design

Problem Definition

- Requirements analysis

Data Requirements

- Data analysis
- Conceptual design

Conceptual Data Model

- Identify entities
- Identify relationships

Logical Schema

- Schema for specific DBMS

Physical Schema

- Storage structure

HERIOT WATT UNIVERSITY

Conceptual Database Design

- Process of constructing a model of the data used in an enterprise, independent of all physical considerations.
- Conceptual database design is top-down design as you start by specifying entities (real-world objects) then build up the model by defining new entities, attributes, and relationships.
- The data model is built using the information in the users' requirements specification.
- The conceptual data model is the source of information for logical design phase.

— Connolly and Begg (2005)

HERIOT WATT UNIVERSITY

Conceptual Design Questions

1. **What are the entities?**
   e.g. students, courses, lecturer
2. **Which relationships exist among these entities?**
   e.g. student studies on a course
3. **What information do we want to store about these entities and relationships?**
   e.g. Student: name, gender, date of birth, …
4. **What are the business rules of the organisation?**
   e.g. a student can only study 4 courses per semester
5. **Which integrity constraints arise from them?**
   e.g. Students must have a unique identifier

HERIOT WATT UNIVERSITY

ENTITY RELATIONSHIP DIAGRAMS

HERIOT WATT UNIVERSITY

Entity Relationship Diagrams

- Capture the entities and their relationships
  - Do not contain foreign keys
- ER Modelling developed by Chen (1976)
- Lots of notation variations since proposed
  - Attributes often omitted from diagrams drawn using Chen notation
- UML notation used in the course

HERIOT WATT UNIVERSITY

Entity

- **Entity:** a thing about which information is kept
  - May be real or abstract
  - Described by a set of attributes
  - May have relationships with other entities
- **Entity type:** a group of objects with the same properties (not values)
  - e.g. all employees in a database
- **Entity occurrence:** a representation of an entity as a particular entity type
  - e.g. a particular employee record in a specific database

HERIOT WATT UNIVERSITY

Attributes

- **Attributes:** a property or characteristic of an entity or a relationship
  - e.g. name, date of birth
- **Attribute domain:** set of allowable values
  - Data type restriction: date of birth must be a date in the past
  - Enumeration: set of building names at Heriot-Watt (Earl Mountbatten, David Brewster, etc)

HERIOT WATT UNIVERSITY

Relationships

- **Relationship:** an association between two or more entities
  - May have attributes associated
  - Captures participation constraints
  - May have multiple relationships between the same entities
  - May have recursive relationships
- **Binary relationship:** an association between two entities
  - Most common form of relationship
  - Captured as a line connecting two entities
- **N-ary relationship:** an association between three entities
  - Requires diamond relationship notation

HERIOT WATT UNIVERSITY

Simple Example: Students on a Degree

Example Requirements
We want a database to store details of students and the degree they are reading for.

For a student we need to capture their student ID, first name, and last name.

For a degree programme we need to capture the unique degree code, type of degree, and the title.

A student can only register on one degree programme. Many students study for the same degree.

Entity:
Student is a thing that we want to capture that is distinct from Degree

Student
studentID {PK}
firstName lastName

Degree
degreeCode {PK}
degreeType title

HERIOT WATT UNIVERSITY

Simple Example: Students on a Degree

Example Requirements
We want a database to store details of students and the degree they are reading for.

For a student we need to capture their student ID, first name, and last name.

For a degree programme we need to capture the unique degree code, type of degree, and the title.

A student can only register on one degree programme. Many students study for the same degree.

Student
studentID {PK}
firstName lastName

Attributes:
The Degree entity has three characteristics that need to be stored. degreeCode uniquely identifies a Degree

Degree
degreeCode {PK}
degreeType title

HERIOT WATT UNIVERSITY

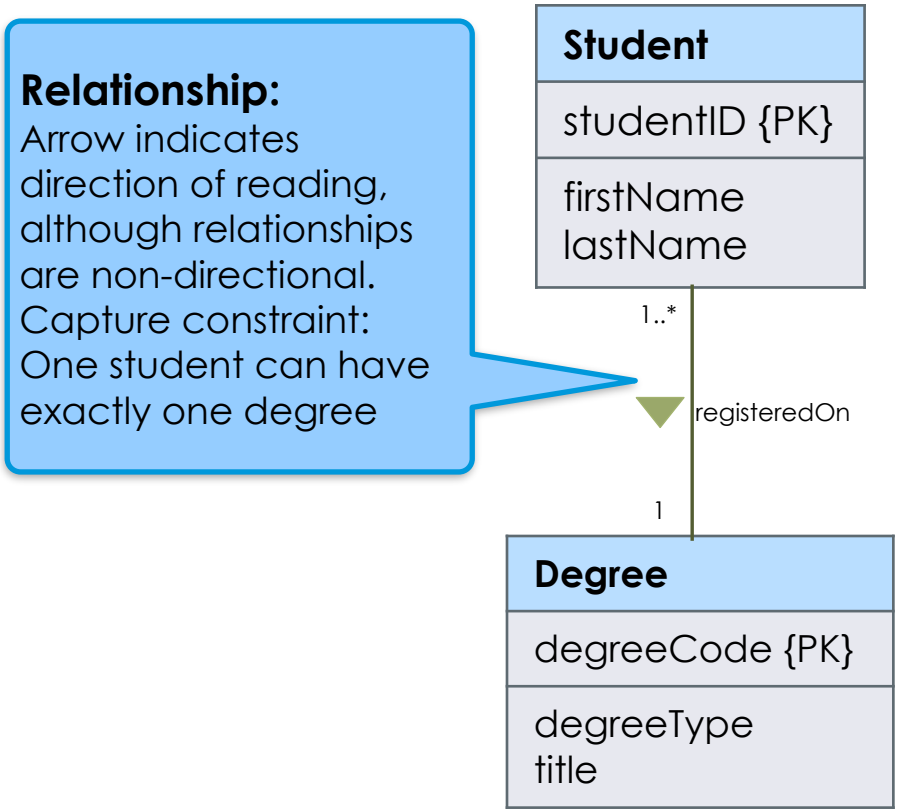
Simple Example: Students on a Degree

Example Requirements
We want a database to store details of students and the degree they are reading for.

For a student we need to capture their student ID, first name, and last name.

For a degree programme we need to capture the unique degree code, type of degree, and the title.

A student can only register on one degree programme. Many students study for the same degree.



HERIOT WATT UNIVERSITY

Company Example: Data Requirements

A company has several departments. Each department has a unique number and a unique name, and may be based in several locations (e.g. Hillhead, Overton, MillEnd). A department has a manager.

A department controls various projects. Projects also have a unique ID and name, and several deadlines (e.g. startup, intermediate, final).

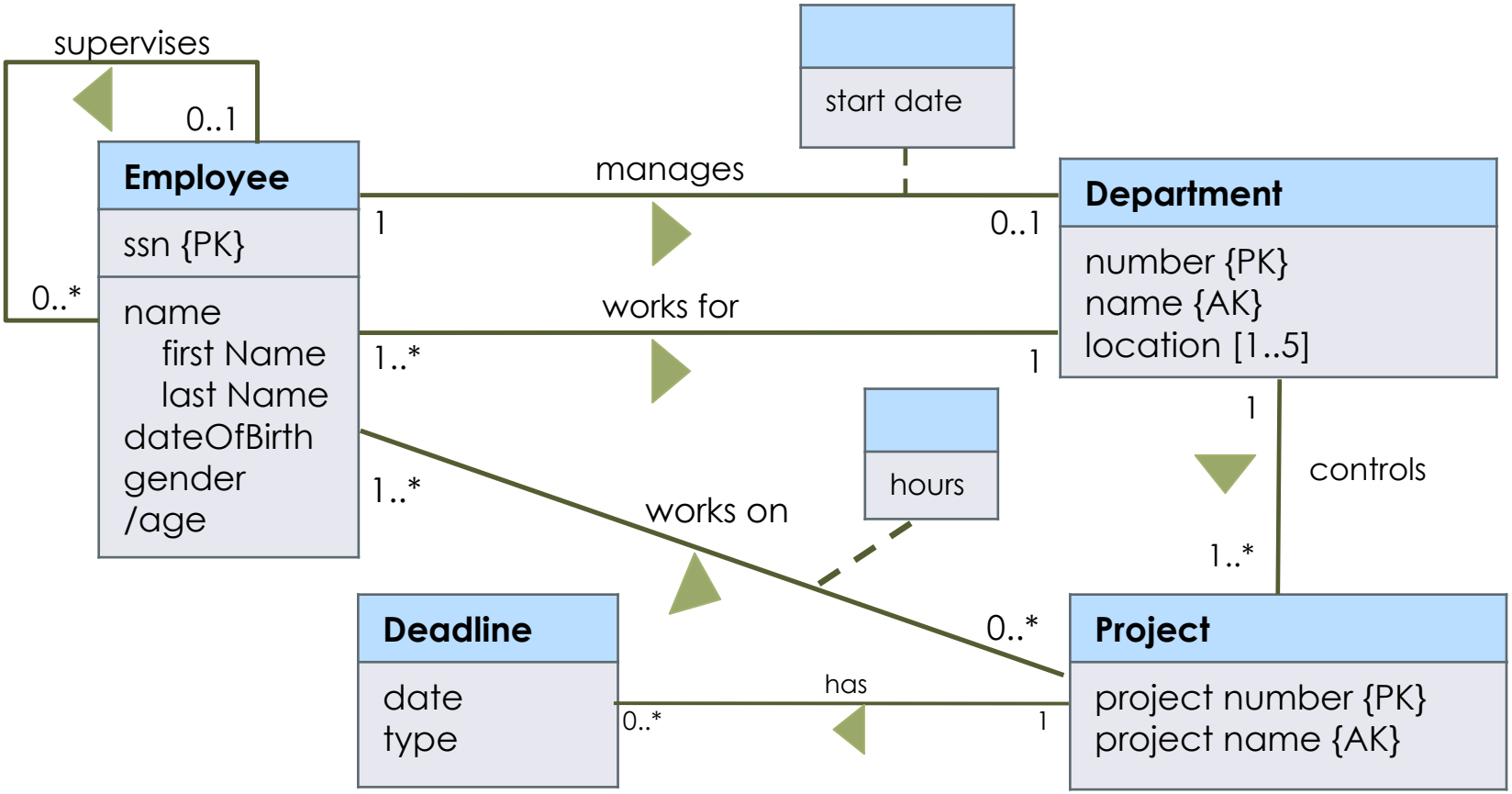
Information required about the employee is a unique ssn (social security number), name, date of birth, age, gender, salary. We don't need to know about which employee works at which location.

An employee works for one department and may work on several projects. The number of hours that they are allocated for each project must be recorded. Most employees are supervised by another employee.

Task: Identify entities, attributes, relationships and constraints in the above

HERIOT WATT UNIVERSITY

Company Example: ER Diagram

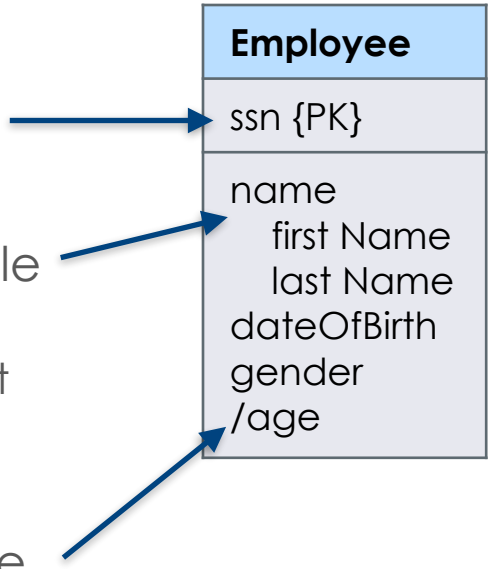


Slides content from F21DF by Dr. Alasdair Gray

HERIOT WATT UNIVERSITY

Types of attributes

- **Simple:** an attribute composed of a single component with an independent existence
- **Composite:** an attribute composed of multiple components
  - e.g. name is a composite of first name and last name
  - Shown with indentation
- **Derived:** an attribute whose value is derivable from a related attribute
  - e.g. age, which can more accurately be computed from date of birth
  - Show with a leading '/'

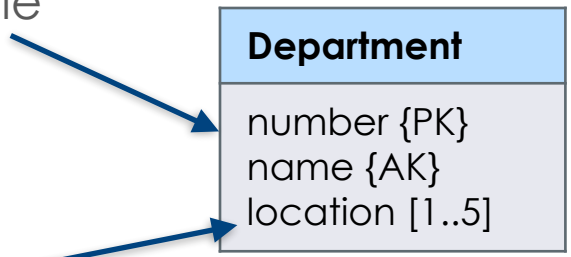


Slides content from F21DF by Dr. Alasdair Gray

HERIOT WATT UNIVERSITY

Types of attributes

- **Single valued:** attribute can only have a single value for an entity
  - e.g. date of birth
- **Multi-valued:** an attribute that holds multiple values for each occurrence of the entity
  - e.g. employee may have up to three telephone numbers
  - Permitted number shown with square brackets



HERIOT WATT UNIVERSITY

Keys

- **Candidate Key:** minimal set of attributes that uniquely identifies an instance
  - May be composite meaning that it is more than one attribute
- **Primary Key:** the chosen candidate key to uniquely identify an instance
  - Shown with {PK}
- **Alternative Key:** other choices for the primary key
  - Shown as {AK}, may need numeric index if it is a composite

Department
number {PK} name {AK} location [1..5]

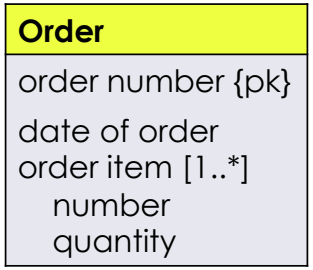
Choosing a key is problematic, e.g. names of individuals are not unique. Where a natural key does not exist, you may need to use an automatically incremented number.

HERIOT WATT UNIVERSITY

Strong and Weak Entities

- **Strong Entity:** does not depend upon another entity for its existence
  - e.g. department and employee entities

- **Weak Entity:** exists only if some other entity exists
  - Details of the weak entity are only stored if the main entity exists
  - Will not have a candidate key, key is dependent upon the related strong entity
  - Capture a composite multi-valued attribute e.g. items on an order



Slides content from F21DF by Dr. Alasdair Gray

HERIOT WATT UNIVERSITY

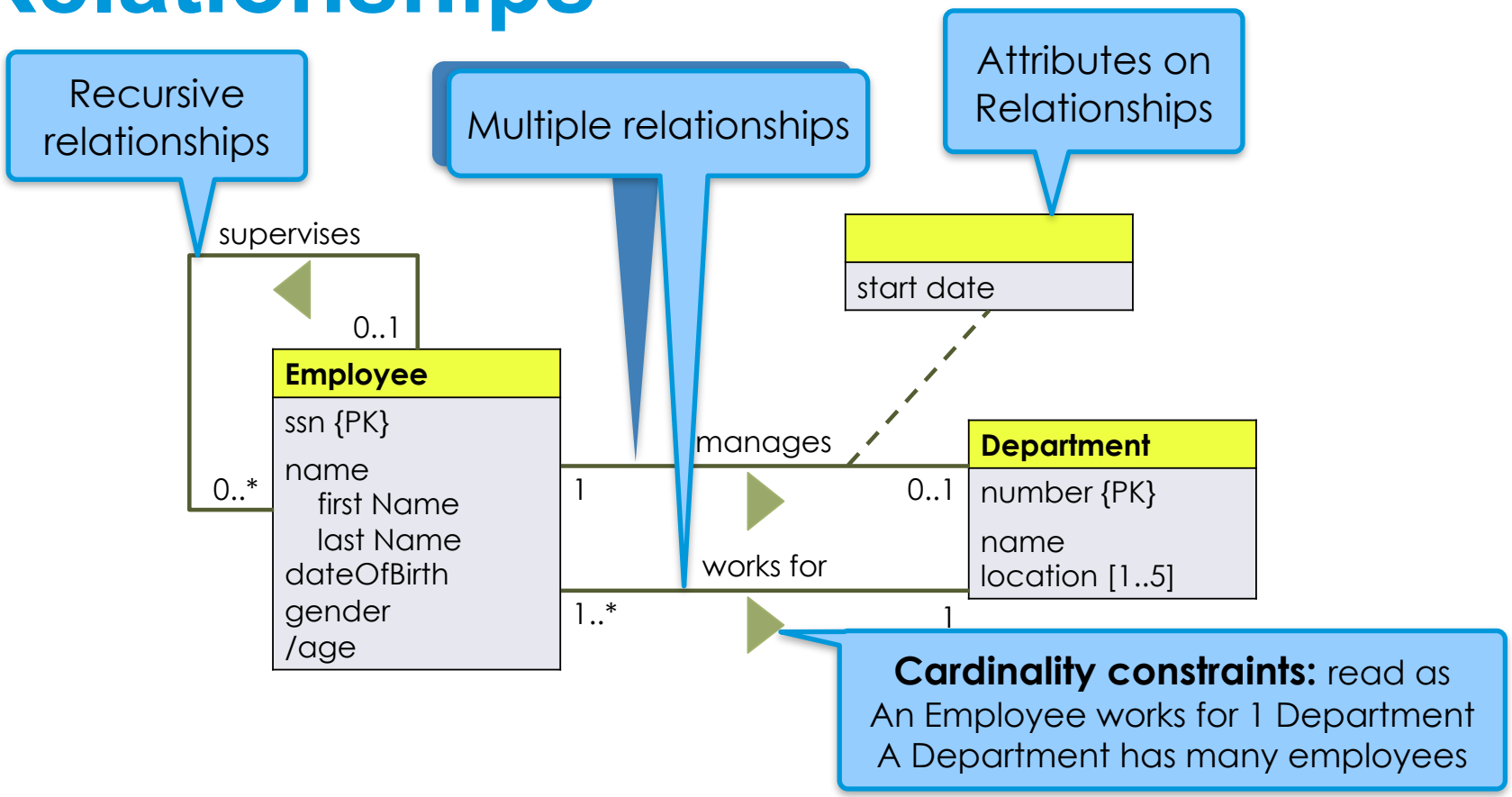
Relationships

Definition: Capture an association between two entities

- **Multiple relationships:** there can be many relationships between the same pair of entities
  - Each relationship captures a different association
- **Relationship attributes:** captures characteristics of the relationship
  - No key value
- **Cardinality constraints:** states how many times an entity is involved in a relationship
- **Recursive relationship:** captures an association between two instances of the same entity

HERIOT WATT UNIVERSITY

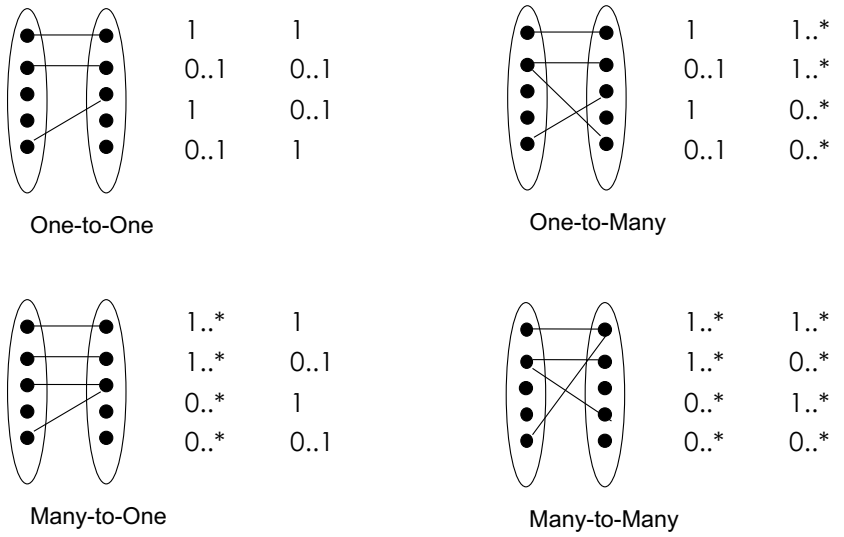
Relationships



Slides content from F21DF by Dr. Alasdair Gray

Relationship Constraints

- **Cardinality Constraint:**
  the number of possible occurrences of an entity type that may relate to a single occurrence of an associated entity type through a particular relationship.
  - One and only one: 1
  - Zero or one: 0..1
  - Zero or more (many): 0..*
  - One or more (many): 1..*



Slides content from F21DF by Dr. Alasdair Gray

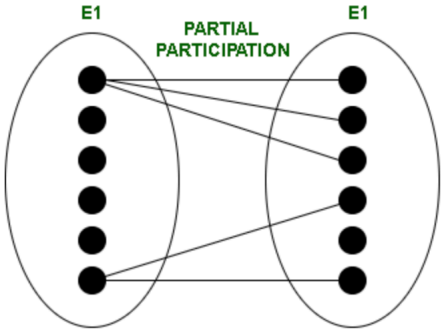
HERIOT WATT UNIVERSITY

Relationship Constraints

- **Participation Constraint:** whether **all** or **only some** entity occurrences participate in a relationship.

Must participate:

- One: 1
- One or more (many): 1..*



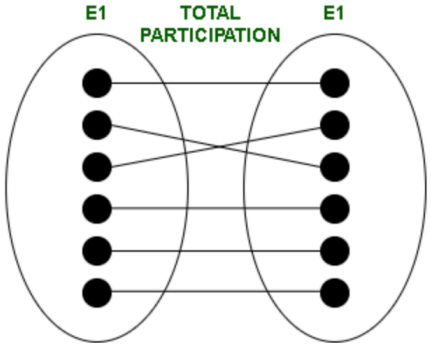


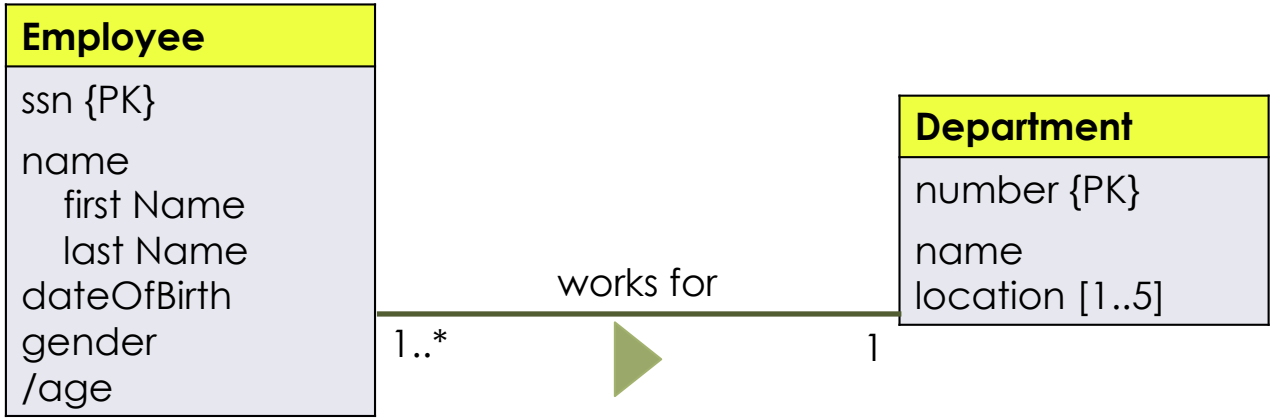
Image from GeeksforGeeks.org

Slides content from F21DF by Dr. Alasdair Gray

HERIOT WATT UNIVERSITY

Reading Cardinality Constraints

- When reading, ignore the number nearest to the first entity mentioned
- Start sentence with 'A' or 'An'
- End sentence with the details in the numbers nearest the second entity
  - An employee works for one department
  - A department is worked for by many employees

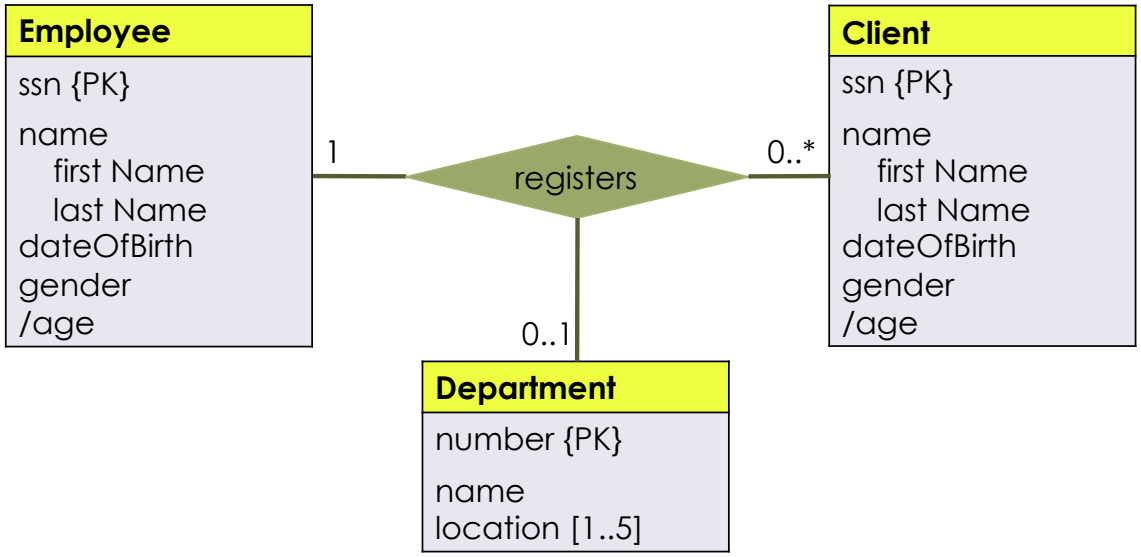


Slides content from F21DF by Dr. Alasdair Gray

HERIOT WATT UNIVERSITY

Complex Relationships

A staff member registers a client at a department. The staff member can register many clients. A client can only be registered at a single department.



Slides content from F21DF by Dr. Alasdair Gray

HERIOT WATT UNIVERSITY

Other Constraints

Not all constraints captured by ER diagram:

- **Custom domain constraints**
  E.g. gender can only be: "Male", "Female", "Other"

- **Multi-table constraints other than the relationships**
  E.g. Employees aged under 21 cannot work on Project X
  E.g. The time ranges allocated to a module cannot overlap

HERIOT WATT UNIVERSITY

Design Choices and Constraints

Design Choices

- Should a concept be modelled as an entity or an attribute?
- Should a concept be modelled as an entity or a relationship?
- Identifying relationships: binary or ternary

Constraints

- Capture the domain semantics
- Model the real world
- Don't impose 'normal' constraints if there could be exceptions.
  - e.g. students take 8 course, but occasionally students take more (repeating a course) or less (part-time)

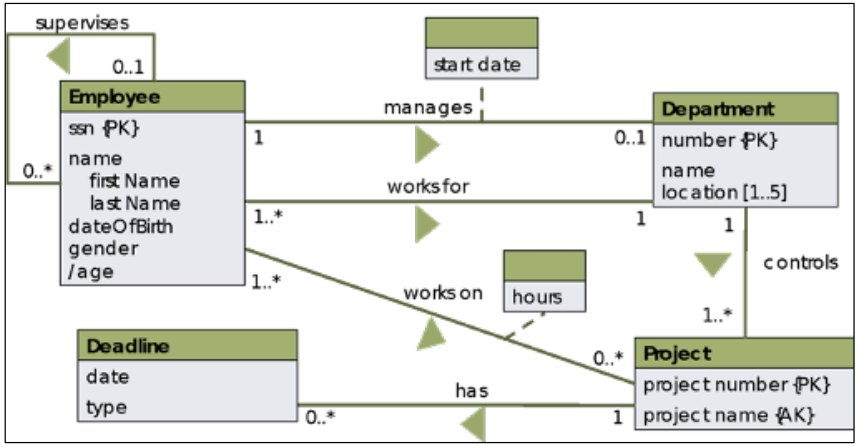
HERIOT WATT UNIVERSITY

LECTURE SUMMARY

HERIOT WATT UNIVERSITY

Summary: Entity Relationship Diagram

- **Conceptual design**
  - Follows requirements analysis
  - Performed at abstract level
    - No implementation details
- **Entity Relationship Model**
  - Captures conceptual design
  - Subjective
  - Include integrity constraints: keys, participation, etc



Slides content from F21DF by Dr. Alasdair Gray

HERIOT WATT UNIVERSITY

References

- Chen, P. (1976). The entity-relationship model—toward a unified view of data. ACM Transactions on Database Systems (TODS), 1(1), 9–36. http://doi.org/10.1145/320434.320440

- Connolly, T., & Begg, C. (2005). Database Systems: A Practical Approach to Design, Implementation, and Management (4th ed.). Addison Wesley. Chapter 11

- Ward, P. (2008). Database Management Systems (2nd ed.). Middlesex University Press. Chapter 3

HERIOT WATT UNIVERSITY

QUESTIONS?