Machine (DBMS, API, Clients, etc..)

Machine

(API)

Database Systems (Categories)

 Centralized vs. Distributed Single user vs. Multiuser

Open sourced (Free) vs. Proprietary

Machine

(Client)

(Expensive) Homogeneous vs. Heterogeneous

Homogeneous

## Heterogeneous

MYSQL)

Machine

(Client)

Machine Machine (DBMS, (API)

Machine (Client)

Machine (API)

Machine

(DBMS)

Machine (DBMS, MYSQL)

Machine (Client)

∘ simple - can't be

divided into more

simple attributes.

• composite - can be

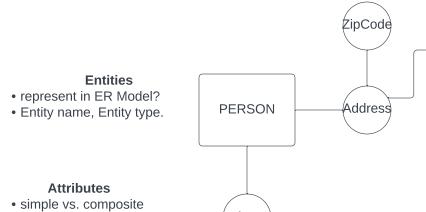
divided into more

simple attributes.

Machine (API)

(DBMS, Azure Cosmos DB)

Machine



**ER Modeling (Conceptual Design)** • Entities - object that has independent existence

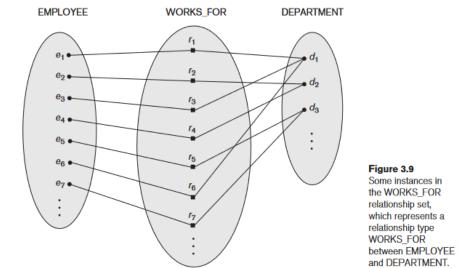
∘ physical - car, desk, etc.. ∘ conceptual - job, class, etc.. • Attributes - makes up the majority of

our databases (things that describe the entities) Name (attribute name), Addison

(attribute value) Relationships - how the entities interact with each other (attributes on entity interacting with another entity)

**External** Mapping Data Models (Database Design Process) Client API DBMS Requirements specification (gathering) Conceptual Data requirements - entities, attributes, relationships • Functional requirements - canned transactions, —Stored Procedure queries that users need to do. Data Independence Logical - ability to make change at conceptual level without having to Query change external. Compiler Physical - ability to make change at interal level without having to change higher levels (i.e. conceptual and Mapping external). Query Optimizer Data Execution

3.4 Relationship Types, Relationship Sets, Roles, and Structural Constraints 73



Let's diagram these logical data requirements as a class (assume we model relationships as attributes for now) this is fair game for a simple quiz question\*:

Identify each entity type E, for each entity type Ei in {E1, E2, ...., En} identify attributes A(E) and the type of attribute i.e. simple vs. composite, single valued vs. multi valued, derived vs. stored, complex\*.

■ The company (Entity) is organized into departments (Entity). Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations. ■ A department controls a number of projects, each of which has a unique name, a unique number, and a single location. ■ The database will store each employee's name, Social Security number, 2 address, salary, sex (gender), and birth date. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. It is required to keep track of the current number of hours per week that an employee works on each project, as well as the direct supervisor of each employee (who is another employee). ■ The database will keep track of the dependents of each employee for insurance purposes, including each dependent's first name, sex, birth date, and relationship to the employee.

ER Diagram (use cheat sheat below):

Symbol Meaning

Entity

cheat sheat

Attribute

Key Attribute

Multivalued Attribute

Composite Attribute

Derived Attribute