**10/2/2019**



Concepts to be modeled:

1. Entity: e.g. student, class (likely noun)
	* Have attributes
	* Form relationship with other entity.
	* More complicated
	* Exists on its own
2. Relationship: e.g. teaches, advises, … (likely verb)
	* ‘Connect’ entities
3. Attribute: e.g. FName, StuId, …
	* Only have values.
	* Simple
4. No need to be modeled.

E.g. Names, like concepts, should be refined.

Employee works for a department. Relationship -> ‘work’ => refinement: ‘supports’, and ‘work as the primary department’ -> ‘classified into’



ER Diagram: Object-based modeling

Entity: class
Attribute
Relationshp

Converted to the relational model:

A *relation schema*, R, is a set of attributes A1, A2,…,An with their domains D1, D2,…, Dn.

E1. An entity is mapped to a relation.

* The relation may use the same name.
* Two relations may be merged into a single relation later on

One object -> one row in the relation.

Six tables:

Department
Employee
Project
Level
Rank
PastRank

A1: A single-valued attribute of atomic data type (of an entity or relationship) is mapped to an attribute in the corresponding table.

Department(DeptId, DeptName, DeptMail)
Employee(EmloyeeId, FName, LName, Phone, Email, RankSince)
Project(ProjectId, ProjName, ProjDesc)
Level(Level, Description)
Rank(Rank, RankDesc)
PastRank(StartDate)

KC1. If a relation R implements an entity E, and E has candidate keys K's, set K's as candidate keys in R.

KC2. If a relation R implements an entity E, and E has no candidate key, create a *surrogate* candidate key SK for R.

* This is needed since every relation must have a candidate key.

Department(DeptId, DeptName, DeptMail)

CK: DeptId

Employee(EmloyeeId, FName, LName, Phone, Email, RankSince)

CK: EmployeeId

Project(ProjectId, ProjName, ProjDesc)

CK: ProjeectId

Level(Level, Description)

CK: Level

Rank(RankId, Rank, RankDesc)

Surrogate key: RankId

PK: RankId

CK: (1) RankId, (2) Rank

PastRank(PastRankId, StartDate)

Surrogate Key: PastRankId



There are departments (entity), with a unique DeptId (attribute of Department: PK). The department name and email address (attributes of Department) should also be stored. An employee is always assigned to a unique (one and only one in the multiplicity) department as his primary department. (noun: employee entity) – verb (relationship):assigned to as primary – noun: department entity) (Assumption: a department may have 0..\* employee as its primary department.) He (employee object) may (therefore support 0..\* department) support any number of additional (relationship) department (noun: entity).