# ER to Relational Mapping

CS157A

Chris Pollett

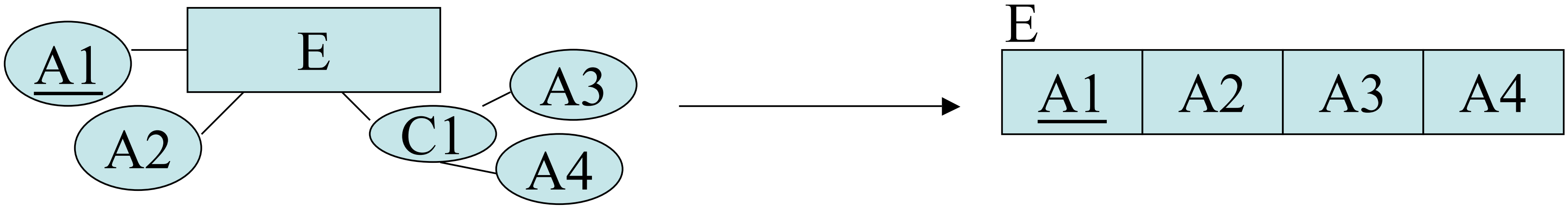
Oct. 10, 2005.

# Outline

- ER to Relational Mapping Algorithm
- Mapping EER Model Constructs to relations

# ER -> Relational Map Step 1

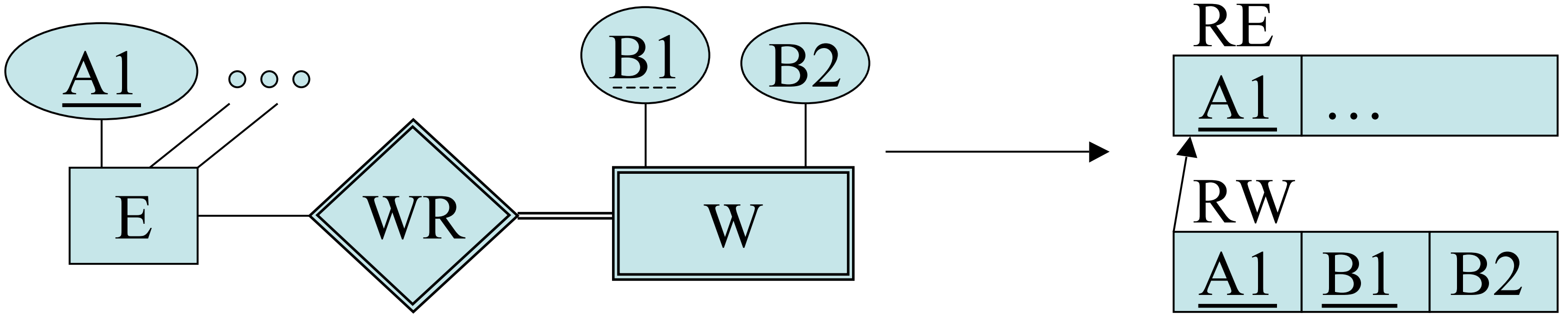
- For each entity type E in the ER schema, create a relation R that includes all the simple attributes of E.



- Add only simple components from any composite atrributes in E.
- Choose one of the key attributes of E to be a primary key of R.

# ER -> Relational Map Step 2

- For each weak entity type W with owner type E create a new relation RW that includes all the simple attributes of W as attributes of RW.
- In addition include a foreign key reference to the key of the translation RE of E.
- The key of RW will be the key of foreign key together with the mapped partial key from W.



RE

| A1 | … |
|---|---|

RW

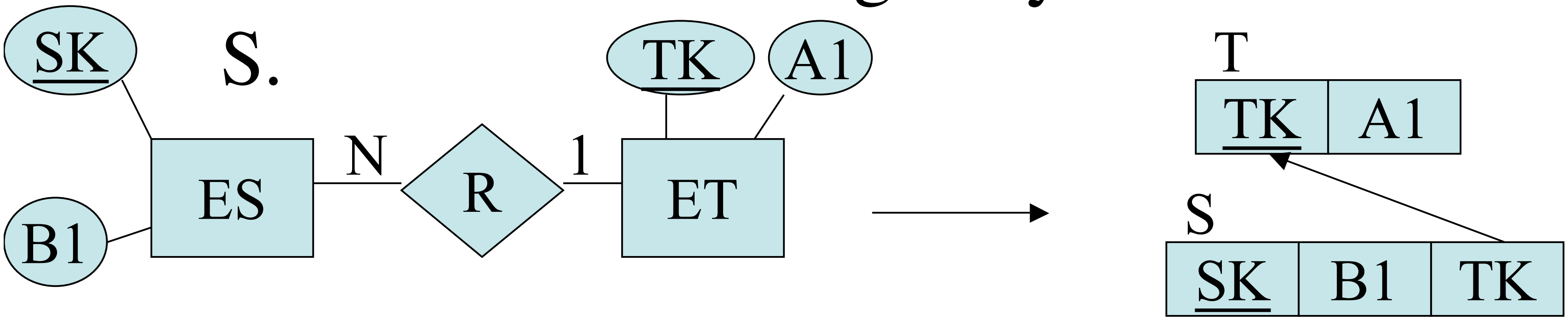
| A1 | B1 | B2 |
|---|---|---|

# ER -> Relational Map Step 3

- For each binary 1:1 relationship R in the ER Schema, identify the relations S and T that correspond to the entity types participating in R.
- There are three possible approaches:
  - **The foreign key approach:** Choose one of the relations, say S, and include in S a foreign key reference to the primary key of T. (Favor  S over T if its corresponding entity participated totally in the relationship.)
  - **Merged relation approach:** When both relations correspond to entities that participated totally in the relationship, one can just merge the two relations into one.
  - **Relationship relation approach:** Set up a new relation with for the purpose of cross referencing the primary keys of tables S and T.

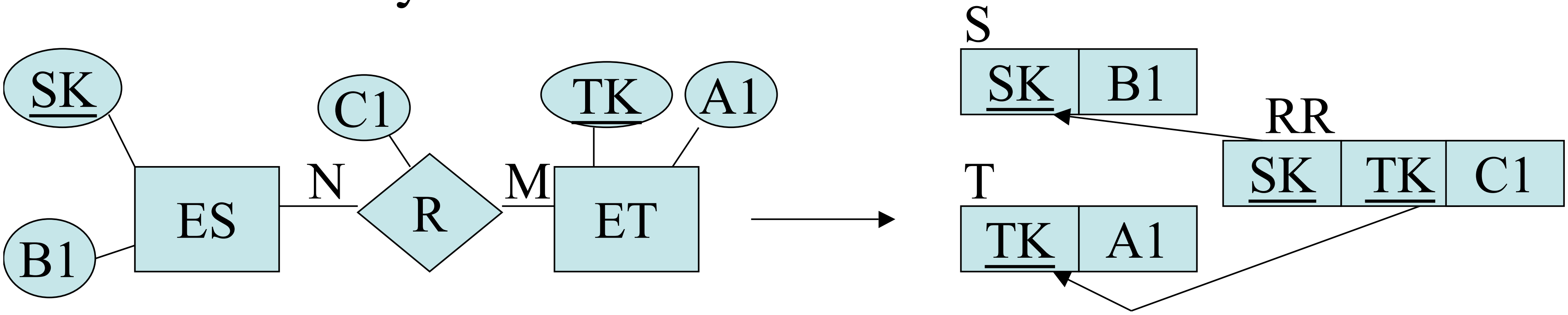
# ER -> Relational Map Step 4

- For each 1:N binary relationship type R, identify the relations S and T corresponding to the entity types in this relationship.
- Further, suppose S is the N-side of the relationship.
- Then include a foreign key reference to R in S.



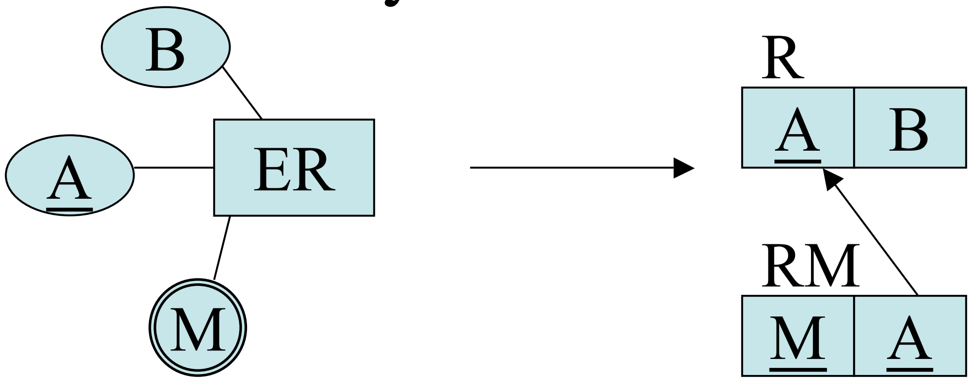
# ER -> Relational Map Step 5

- For each binary M:N relationship type R, create a new relation RR to represent R.
- Include as foreign keys of keys the primary keys of the relations corresponding to the two participating entities.
- Also add as attributes simple attributes of R as in the entity case.



# ER -> Relational Map Step 6

- For each multivalued attribute M, create a new relation RM. This relation will include an attribute corresponding to M as well as a foreign key reference to the relation corresponding to the entity that M was part of.
- The key will be both attributes.



# ER -> Relational Map Step 7

- For each n-ary relationship R with n >2, create a new relation RR to represent R.
- Include as foreign key attributes in RR the primary keys of the relations that correspond to the participating entities with cardinality constraints other than 1.
- The primary key of RR is the combination of these foreign keys.

# Discussion

- In this mapping to relations, relationship types are not represented explicitly as in the ER schema.
- In the ER setting, relationships have ordered tuples of entities and so all the entity attributes are available in the relationship.
- Nevertheless, the same effect can be had using our mapping by joining the relevant relations on the foreign key references corresponding to the original relationship.

# Mapping Specialization and Generalization

- There are several different ways one could map a specialization {S_1,…,S_n} of some class C with attributes {k (the key), a_1, …, a_n}.
    - Multiple relations --- Superclass and Subclass: Create a relation RC for C with attributes {k, a_1, …, a_n} and a relation RS_i for each i with attribute {k}U{attributes of S_i}
    - Multiple relations -- Subclass relations only. Create RS_i's as above but don't create an RC. This works if specialization is total.
    - Create a single relation with one type attribute -- let RC be a new relation with attribute {k, a_1,…, a_n}U{attributes S_1}U…U{attributes S_n}U{t}. Here t is a string of the form "C" or "S_i" for some i. This works only if specialization is into disjoint subclasses.
    - Single relation with multiple type attributes -- same as above except rather than adding a single t, we now add boolean flag types t_1, …, t_n one for each S_i which indicates which S_i are relevant for a particular row.

# Mapping Union Types

- For mapping a union type whose defining superclasses have different keys, one specifies a new key attribute called a **surrogate key**.
- This is then included as a foreign key attribute in each relation corresponding to a superclass of the union type.