Contents

Relational Schema Mapping

- How to move from a conceptual database design
    - Entity Relationship Model
- …to a logical database design
    - Relational Database Schema
- We follow a series of steps to map entity types, relationship snad attributes into relations
- The mapping will create:
    - Relations
        - We simple, single-valued attributes
    - Constraints
        - Primary keys
        - Unique keys
        - Referential integrity constraints

Mapping of Entity Types

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

    - When mapping composite attributes include only the simple component attributes in the new relation R

- Key attributes
    - Choose one of the key attributes of E as the primary key of R
    - Composite key attributes are included as a *composite primary key*
- Additional key attributes should be included as secondary unqiue keys of the relation

Mapping Multivalued Attributes

- For each multivalued attribute A, create a new relation R
- The new relation R will include
    - An attribute corresponding to A
    - The primary key K from the relation that represents the tntity type that A came from
        - This becomes a *foreign key* in R
    - The *primary key* of R is the combination of A and K

Mapping Relationships

- In addition to mapping the entity types from the ER model into the Relational Schema, we also need to map the relationship types
- Each relationship type is modeled differently
    - 1:1 - One to one
    - 1:N - One to many
    - M:N - Many to many

1:1 Relationships

- There are two main approaches to mapping binary 1:1 relationships
    - Foreign Key Approach
        - Most useful and most commonly used
    - Merged-Relation Approach
        - Used in cases of *total participation*
- For each binary 1:1 relationship type R
    - Identify the relations S and T that correspond to the entity types participating in R

- Foreign Key Approach
  - Choose one of the participating relations, say S
  - Include as a foreign key in S the primary key of T
  - If possible, choose and entity type with *total participation* in R for the role of S
  - Include all the simple attributes of the relationship type R as attributes of S
- Merged Relation Approach
  - This can only be used when both S and T have *total participation* in the relationship type R
  - Merge the two entity types S and T and the relatnship type R into one single relation V
  - V should include all the simple component attributes of S, T and R
  - This is possible as the joint total participation indicates that the type tables will have an identical number of tuples at all time

1:N Relationships

- For each binary 1:N relationship type R
  - Identify the relation S that corresponds to the entity types on the N-side of R
- Include as a foreign key in S, the primary key of T, which is the relation representing the entity type at the other side of R
- Include any simple attributes of the relationship type R as attributes of S
  - Or simple component attributes of a composite attribute

Recursive Relationships

- Resursive Relationships
  - Where an entity instance can refer to anotherinstance of the same entity type
- For each recursive relationship type R
  - Include the primary key of T, which is the relation representing the entity type involved, as a foreign key in the same relation, T
  - Include any simple attributes of the relationship type R as attributes of T
    - Or simple component attributes of a composite attribute

M:N Relationships

- Many to many relationship types are more complex to map than 1:1 or 1:N
- As each entity instance may reference many entity instances in the other participating entity type
  - You cannot use a foreign key attribute in either participating entity
  - You must create a new relation to represent the relationship type
- For each binary M:N relationship type R
  - Create a new relation S to represent R
- Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types
  - The combination of these foreign keys is the composite primary key of S
- Include any simple attributes of the relationship type R as attributes of S
  - Or simple component attributes of a composite attribute