Outline

- ER Modeling
- Relational Modeling
- Relational Algebra
- Mapping ER to Relational
- SQL

ER Modeling

Concepts

- Entity Type
  - Keys uniquely id a tuple
  - Attributes

- Weak Entity Type
  - No unique key (perhaps partial)
  - Is ID'd by the Regular Entity Type

- Relationship Type
  - Cardinality Ratios 1:1, 1:N, M:N

- Identifying Relationship Type

Structural Constraints

- Cardinality Ratios
  - Keywords: Every, At Most, At least, Exactly, ...
- Participation Constraints
  - Total a.k.a "Existence Dependency"
  - Partial

Relational Model

Constraints

- Domain Constraint
  - The value of the attribute must fall within the domain.
- Entity Integrity Constraint
  - No null values for a primary key.
- Key Constraint
  - No duplicate keys
- Referential Integrity Constraint
  - A reference must follow through

Update = Delete + Insertion, Delete only can violate EIC.

Keys

- Superkey minimally identifies a tuple (with 1≥ attributes)
- Primary Key: One attribute

Relational Algebra

Set Operations

- Union ∪
- Intersection ∩
- Difference −
- Cartesian Product ×

- Schema of first relation
- Column domains must match (e.g. FName, LName)

Relational

- Select σ_{COND}
- Projection π_{LIST} * Returns a set (no dups)
- Join
  - Theta $\bowtie_{COND}$ * No common attributes * $\times(\sigma(R))$ * Keeps all columns, only returning rows that match COND
    * Allowed $>, <, \ldots$
  - Equi
  - Natural * * Redundant join attrs are dropped * Join requires a same-named attr.
    * Implied equijoin
- Renaming $\rho_{A \to C}$ * Necessary for *
- Division ÷ * Keyword: "all" * Equivalent:

$$T_1 \leftarrow \pi_y(R)$$
$$T_2 \leftarrow \pi_y((S \times T_1) - R)$$
$$T \leftarrow T_1 - T_2$$

- Group/Aggregate $\mathcal{F}$
  - SUM - AVG
  - MAX - MIN
  - COUNT
  - GROUP==PARTITION
  - If no group, apply to all.
- Outer Joins
  - Left =⟕ Keep all rows from L, pad R with nulls.
  - Right ⟖= Same^
  - Full =⟗= Pad both sides as needed!
- Selectivity $= \frac{|\sigma_c(R)|}{|R|} = \frac{|\pi_L(R)|}{|R|}$
- Cardinality $= |\pi_L(R)|$

Mapping ER to Relational

1. Regular Entity Types
   - Convert each to a Relation, roll over all attributes

2. Weak Entity Types
   - Convert each to a Relation,

3. Binary 1:1 Relationships
   - Take the total participation side, and include a FK.
     The FK will be the PK of the partial side

4. Binary 1:N Relationships
   - Take the "N" relation, and include a FK.
     The FK will be the PK of the "1" relation.
   - Include the attrs of the "1" relation.

5. Binary M:N Relationships
   - Make a new relation
   - Include FKs that are PKs of both Entity Types

6. Multivalued Attributes
   - Create a new relation R, with attr A.
   - Make an FK to the PK of relation that owns A
   - Now, the PK of R should be A+K.

7. N-Ary Relationship Types
   - Create a relation S with all attributes.
   - Include an FK for all participating Relations
   - The PK becomes all referenced FKs.

8. Superclass / Subclass

   A.
   - Create a relation for the superclass.
   - Create a relation for each subclass, with FK = PK (Super)

   B.
   - Create relations for all subclasses.
   - Redundantly include the attributes of the superclass

   C.
   - Create a relation for the Superclass
   - Include a "type" for each subclass
   - Include attributes for ALL subclasses.

   d.
   - Create a relation for the superclass
   - Include all subclass attributes
   - Include a flag for each subclass, to know what "type" it is.