SWEN-220
Mathematical Models of Software

Conceptual to Relational Mapping
Relational (Logical) Design

• “Logical” design is where a conceptual design is mapped to a relational schema.
• Entry = Conceptual ERD (Cheng Notation)
• Exit = Well-structured (Normalized) Relations in Relational Schema (Structure) Notation (RSN)
• Resulting relations will closely resemble their physical representation – tables in a relational database (RDB).
Relational Model

• A relation is a set of attributes with values for each attribute such that:
  – Each attribute (column) value must be a single value only.
  – All values for a given attribute (column) must be of the same data type.
  – Each attribute (column) name must be unique.
  – The order of attributes (columns) is insignificant.
  – No two tuples (rows) in a relation can be identical.
  – The order of the tuples (rows) is insignificant.
What is Relational Schema Notation (RSN)?

1. Notation to show logical database structure.
2. "Flattened" linear text vs. diagram.
3. Entities become relations in the RDB:
   a. Entity attributes carried over (mostly).
   b. Entity key attributes become primary relation keys
4. Includes additional relations "implied" by the ERD:
   a. Multi-valued attributes.
   b. M : N relationships.
   c. Relationships with attributes.
RSN Notation

\[ R( \text{attr}_1, \text{attr}_2, \text{attr}_3, \ldots, \text{attr}_N ) \]
Relation with \( N \) attributes.

\[ R( \underline{\text{attr}_1}, \underline{\text{attr}_2}, \text{attr}_3, \ldots, \text{attr}_N ) \]
Primary key attribute(s) are underlined.

\[ R( \text{attr}_1, \circled{\text{attr}_2}, \circled{\text{attr}_3}, \ldots, \text{attr}_N ) \]
Foreign key attributes are circled.
We name the relation & primary key to which each foreign key refers.
NOTE: An attribute can be (part of) both the primary and a foreign key.
RSN Notation

\[ R( \text{attr}_1, \text{attr}_2, \text{attr}_3, \ldots, \text{attr}_N ) \]

Relation with N attributes.

\[ R( \underline{\text{attr}_1}, \underline{\text{attr}_2}, \text{attr}_3, \ldots, \text{attr}_N ) \]

Primary key attribute(s) are underlined.

\[ R( \underbrace{\text{attr}_1}, \underbrace{\text{attr}_2}, \text{attr}_3, \ldots, \text{attr}_N ) \]

Foreign key attributes are circled.

We name the relation & primary key to which each foreign key refers.

NOTE: An attribute can be (part of) both the primary and a foreign key.

Example:

\[ \text{Players( ssn, firstName, lastName, uniformNumber, team )} \]
RSN Notation

R( attr1, attr2, attr3, . . . , attrN )
  Relation with N attributes.

R( attr1, attr2, attr3, . . . , attrN )
  Primary key attribute(s) are underlined.

R( attr1, __attr2__, attr3, . . . , attrN )
  Foreign key attributes are circled.
  We name the relation & primary key to which each foreign key refers.
  NOTE: An attribute can be (part of) both the primary and a foreign key.

Example:

Players( __ssn__, firstName, lastName, uniformNumber, team )
RSN Notation

\[ R( \text{attr}_1, \text{attr}_2, \text{attr}_3, \ldots, \text{attr}_N ) \]
Relation with \( N \) attributes.

\[ R( \underline{\text{attr}_1}, \text{attr}_2, \text{attr}_3, \ldots, \text{attr}_N ) \]
Primary key attribute(s) are underlined.

\[ R( \text{attr}_1, \text{\underline{circled attr}_2}, \underline{\text{attr}_3}, \ldots, \text{attr}_N ) \]
Foreign key attributes are circled.
We name the relation & primary key to which each foreign key refers.
NOTE: An attribute can be (part of) both the primary and a foreign key.

Example:

Players( \text{ssn}, \text{firstName}, \text{lastName}, \text{uniformNumber}, \text{team} )
RSN Notation

R( attr1, attr2, attr3, . . ., attrN )
   Relation with N attributes.

R( attr1, attr2, attr3, . . ., attrN )
   Primary key attribute(s) are underlined.

R( attr1, attr2, attr3, . . ., attrN )
   Foreign key attributes are circled.
   We name the relation & primary key to which each foreign key refers.
   NOTE: An attribute can be (part of) both the primary and a foreign key.

Example:

Players( ssn, firstName, lastName, uniformNumber, team )
   *team refers to teamld in the Teams relation
Entity Conversion Rules

1. Create an RSN relation for each entity.
2. Simple attributes are carried over as is to the RSN
3. Derived attributes are not carried over.
4. Composite attributes are removed; only the simple fields of each composite are carried over to RSN.
5. The entity's key attributes define the primary key in RSN.
6. Multi-valued attributes are replaced by a new relation, say M:
   a. The attributes in M are those of the multi-valued attribute.
   b. Create a foreign key from M back to entity from which it was extracted.
   c. The key of M is all of its attributes.
1. Transform regular ERD entities to relations.
   – Attributes map directly
   – Entity type identifier becomes primary key(s).

PlayerID

Player

Name

Player(PlayerID, Name)
2. Create additional relations for multi-valued attributes.
   - Second relation has the primary keys as the primary key of the first relation plus the multi-valued attribute.
   - Second relation now contains a foreign key of the first’s relation’s primary key.

```
Player( PlayerID, Name)
Positions( PlayerID, Pos)
```

PlayerID refers to PlayerID in Player relation
3. Map One-to-Many (1:M) Relationships
   – Primary key on the one-side becomes **foreign key** on the many-side

Customer(CustID, Name)
Order(OrderID, Date, CustID)
CustID refers to CustID in Customer relation

- Create a relation to represent the relationship with a descriptive name.
- Primary keys of new relation are primary keys of each entity participating in relationship. They also become foreign keys.
- Add relationship attributes to new relation.

Student(StuID, StuName)
Course(CourseID, Cname)
Transcript(StuID, CourseID, Grade)

StudID refers to StuID in Student relation
CourseID refers to CourseID in Course relation