Logical Design

Projet base de donnèes – ENS Cachan

KOLČÁK Juraj

juraj.kolcak@lsv.fr

23 févier 2018

Introduction

Relational Schema

- The universe is modelled by relation schemas. A relation schema is a tuple of attributes $R = (A_1, \ldots, A_n)$.
- Each relation schema is populated by tuples of n values, representing the real world data. The set of all such tuples is referred to as relation state $(r(R))$ of relation schema R.
- Each attribute A_i has it's own domain, specifying the possible values of the i-th value in the n-tuple.

Introduction

Primary Key

Essentially, the same as primary keys in the EER – a subset of attributes that uniquely identifies each relation instance.

Foreign Key

A foreign key of relation schema R_1 that references relations R_2 is a set of attributes F such that :

- the domains of attributes in F are identical to the domains of attributes in P_2 (primary key of R_2).
- For each tuple $t_1 \in r_1(R_1)$, either $t_1[F] = \{NULL\}^{|F|}$ or there exists a tuple $t_2 \in r_2(R_2)$ such that $t_1[F] = t_2[P_2]$.

From EER to Relational Schema

Translation

An EER can be translated into relational schema almost algorithmically. The translation can be split into several steps :

1. **Regular Entity Types** : Relation schemas (all attributes preserved except multi-valued).
2. **Weak Entity Types** : Include a foreign key referencing the identifying entity.
3. **Relationship Types** : 3 options : foreign key/merge relations/cross-reference (relationship relation).
4. **Multivalued attributes** : New relation with foreign key referencing the owning entity.
5. **Inheritance** : 3 option : superclass + subclasses/subclasses only/superclass only.

From EER to Relational Schema

Relationship Types

- **Foreign Key** : If a relationship type has cardinality ratio of 1 :N (or 1 :1), the relationship can be represented by a foreign key in the relation of the entity type participating with cardinality 1, referencing the relation of the entity type participating with cardinality N. This is especially useful if the participation of the first is total (otherwise NULL values are introduced into the foreign key column).
- **Relation merging** : In case of a 1 :1 relationship type, it is possible to represent both participating entity types in a single relation schema (since 1 :1 defines pairing). Again this is useful if the entity types have total participation, to avoid excessive amount of NULL values.
- **Cross-reference** : This option is universal (works for any cardinality). The relationship type is represented by it's own relation schema, having foreign keys to both (all) participating entities.

From EER to Relational Schema

Inheritance

- **Superclass only** : Only one relation schema is introduced for the parent entity type. The attributes and relationships of the children entities are included in the relation schema, leading to NULL values for entities of different child types. Additionally a type attribute is usually included to indicate the type of child entity (several Boolean attributes have to be included for overlapping inheritance).
- **Subclasses only** : One relation schema is introduced for each child entity type. The attributes and relationships of the parent entity are included in each of the relation schemas. Not suitable for overlapping inheritance since it would lead to data duplication.
- **Both superclass and subclasses** : The universal option, each entity type (both parent and children) are converted into a relation schema with their respective attributes and relations. The relation schemas of children entities include a foreign key pointing to the parent entity.