Transformations

Process

Input: conceptual schema (diagram)

Tools: transformation rules

Output: relational schemas (headers of relational tables)

Methodology

Step 1: Replace multivalued attributes with classes of objects and *one-to-many* associations.

Step 2: Replace association classes with *one-to-many* associations and qualifications.

Step 3: Replace link attributes with *one-to-many* associations and qualifications.

Step 4: Replace *many-to-many* associations with *one-to-many* associations and qualifications.

Methodology

Step 5: Replace qualifications with a composite primary key (PK) and foreign key (FK).

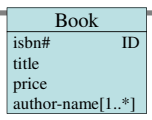
Step 6: Replace generalization hierarchies using either superset, subset, or association method.

Step 7: Migrate copies of identifiers from one side of associations to many side of associations and create foreign keys (FK).

Step 8: Replace identifiers with primary keys (PK) and candidate keys (CK).

Step 9: Replace classes of objects with relational schemas.

Step 1: Transform multivalued attributes

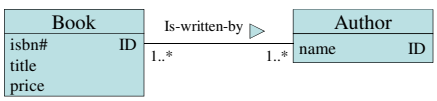


(1) Promote a multivalued attribute (author-name) to a class of objects (AUTHOR),

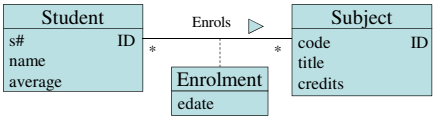
(2) Add a multivalued attribute name (author-name) as a singlevalued attribute (author-name) to a new class (AUTHOR)

(3) Remove a multivalued attribute (author-name) from the original class (BOOK),

(4) Create an association (Is-written-by) between the original class (BOOK) and a new class (AUTHOR).



Step 2: Transform association classes



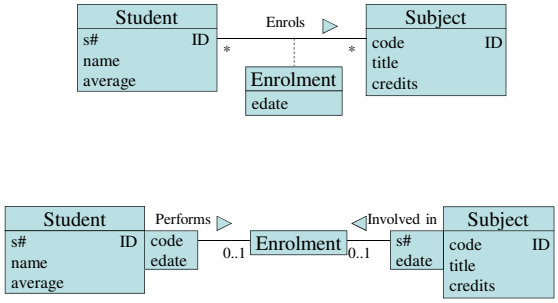
(1) Remove *many-to-many* association (Enrols),

(2) Create *one-to-many* association (Performs) between one of the classes (STUDENT) involved in many-to-many association (Enrols) and former association class (ENROLMENT),

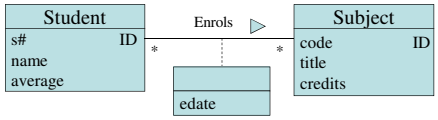
(3) Create *one-to-many* association (Involved-in) between the other class (SUBJECT) involved in many-to-many association (Enrols) and former association class (ENROLMENT),

(4) Add qualifications (s#,edate),(code,edate) on *one* sides of both associations and replace *many* with *zero-or-one*; when creating the qualifications carefully consider the semantics of the original association class.

Step 2: Transform association classes

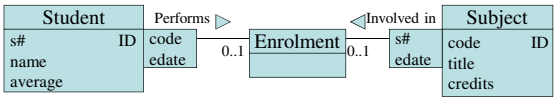


Step 3: Transform link attributes

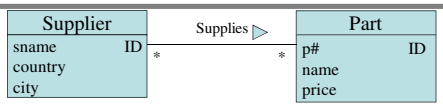


(1) Replace a link attribute (edate) with association class ENROLMENT and the attribute (edate) describing the association class,

(2) Perform Step 2

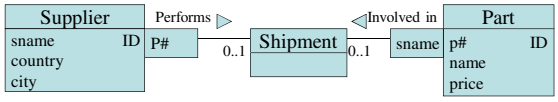


Step 4: Transform many-to-many associations

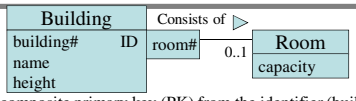


(1) Add association class (SHIPMENT) that encapsulates association Supplies,

(2) Perform Step 2.



Step 5: Transform qualifications



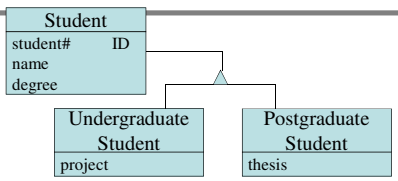
(1) Create a composite primary key (PK) from the identifier (building#) of class (BUILDING) located on *one* side of association (Consists-of) and qualification attribute (room#),

(2) Add the pimary key created in the previous step to a description of a class (ROOM) on zero-or-one side of association (Consists-of),

(3) Mark the identification attribute (building#) in the primary key as a foreign key (FK),

(4) Remove qualification (room#) and qualified association (Consists-of).

Step 6: Transform generalizations



Superset method

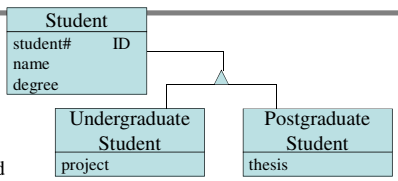
(1) Move all attributes (project, thesis) from all subclases (UNDERGRADUATE STUDENT, POSTGRADUATE STUDENT) to the top level superclass (STUDENT),

(2) Add an attribute "type-of" to a superclass,

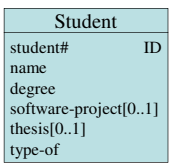
(3) Mark all attributes moved from the lover levels of generalization hierarchy as optional ([0..1]),

(4) Remove all subclasses.

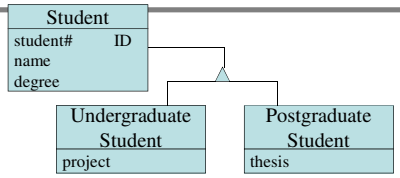
Step 6: Transform generalizations



Superset method



Step 6: Transform generalizations

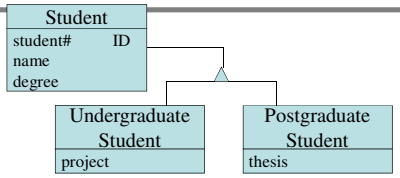


Subset method

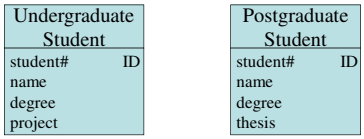
(1) Move all attributes (student#, name, degree) from the classes at nonleaf levels (STUDENT) of generalization hierarchy to leaf level classes (UNDERGRADUATE STUDENT, POSTGRADUATE STUDENT),

(2) Remove all classes at from nonleaf levels of generalization hierarchy.

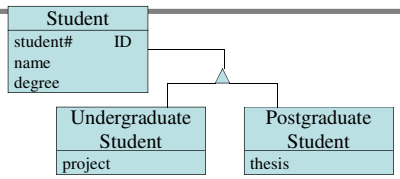
Step 6: Transform generalizations



Subset method



Step 6: Transform generalizations

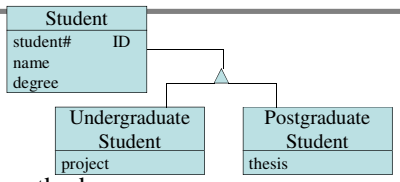


Association method

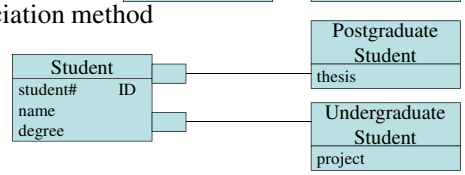
(1) Replace generalization symbols with *one-to-one* associations qualified with empty sets of attributes,

(2) Perform step 5.

Step 6: Transform generalizations

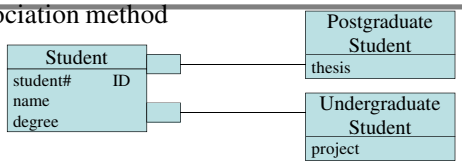


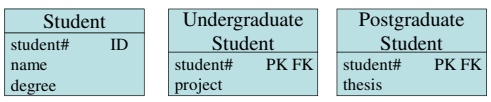
Association method



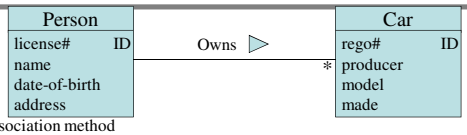
Step 6: Transform generalizations

Association method





Step 7: Transform associations

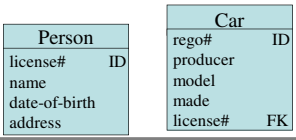


Association method

(1) Migrate a copy of identifier (license#) from one side (PERSON) of association (Owns) to many side (CAR) of association (Owns),

(2) Make a copied identifier (license#) a foreign key (FK) on many side (CAR) of association (Owns),

(3) Remove the association.



Step 8: Transform identifiers

Person	
ssno	ID1
first-name	ID2
last-name	ID2
date-of-birth	ID2
country	
city	
street	
building#	
phone#	ID3
income[0..1]	

Person	
ssno	PK
first-name	CK1
last-name	CK1
date-of-birth	CK1
country	
city	
street	
building#	
phone#	CK2
income[0..1]	

(1) Pick an identifier and mark it as a primary key (PK)

(2) Mark all other identifiers as candidate keys (CK)

Step 9: Transform classes of objects

(1) Replace a class of objects with a relational schema,

(2) Mark all optional attributes income[0..1] with NULL. PERSON(ssno, first-name,last-name,date-of-birth,country,city,street,building#, phone#, income:NULL)

primary key (ssno),

candidate key 1(first-name,last-name,date-of-birth), candidate key 2(phone#)

Person	
ssno	PK
first-name	CK1
last-name	CK1
date-of-birth	CK1
country	
city	
street	
building#	
phone#	CK2
income[0..1]	