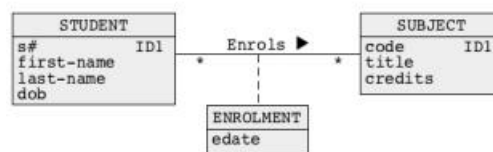


The logical design transforms a conceptual schema into a set of relational schemas

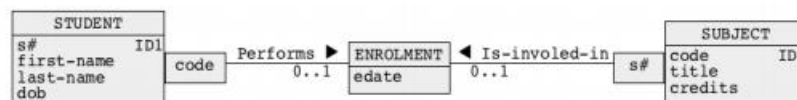
The logical design is performed as a sequence of the following steps:

- (1) The multivalued attributes are replaced with the classes of object sand depending on the semantics of multivalued attribute either with one-to-many or many-to-many associations
- (2) The associaton classes and link attributes are replaced with the triples (one-to-many association:class-of-objects :many-to-oneassociation)
- (3) Many-to-many associations are replaced with triples (one-to-many association:class-of-objects:many-to-one association)
- (4) The qualifications are replaced with one-to-many associations and composite identifiers in object classes on "many" side of one-to-many associations
- (5) The selected identifiers are copied from the classes of objects on "one" sides of association to the classes of objects on "many" side of the associations and such identifierds are tagged with FK_n tag (an index "n" is used to distinguish between different foreign keys)
- (6) The triples (class-of-objects:one-to-one association:class of objects) are merged into one class of objects and one-to-one associations are removed
- (7) The superset, subset and association methods are used to transform the generalizations
- (8) The primary and candidate keys are created
- (9) The associations are removed
- (10) The relational schemas with the referential integrity constraints are created

Association classes



- An association class Enrols is transformed in the following way:
- Many-to-many association Enrols is removed
- A one-to-many association Performs is added between a class STUDENT and a class ENROLMENT
- A one-to-many association Is-involved-in is added between a class SUBJECT and a class ENROLMENT
- A qualification with an attribute code is added on a STUDENT side of association Performs and a qualification with an attribute s# is added on SUBJECT side of association Is-involved-in



A qualification with an attribute code of an association Performs contributes to an identifier (s#, code) of a class ENROLMENT

A qualification with an attribute s# of an association Is-involved-in also contributes to an identifier (s#, code) of a class ENROLMENT

Qualified association classess

An association class Enrols qualified with an attribute edate is transformed in the following way:

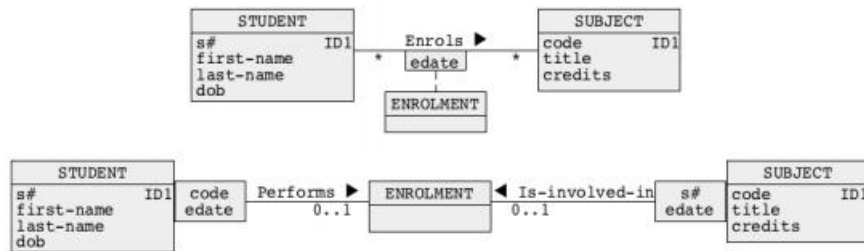
Many-to-many association Enrols is removed

A one-to-many association Performs is added between a class STUDENT and a class ENROLMENT

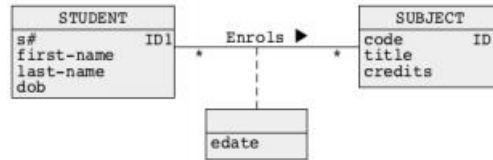
A one-to-many association Is-involved-in is added between a class SUBJECT and a class ENROLMENT

A qualification with the attributes (code,edate) is added on STUDENT side of association Performs and a qualification with the attributes (s#,edate) is added on SUBJECT side of association Is-involved-in

An attribute edate is removed from a class ENROLMENT

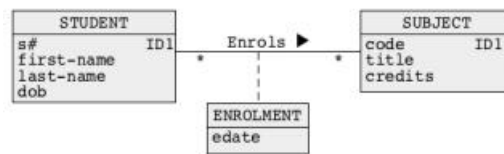


Link Attributes



A link attribute edate is transformed in the following way:

- A link attribute edate is promoted to an association class ENROLMENT

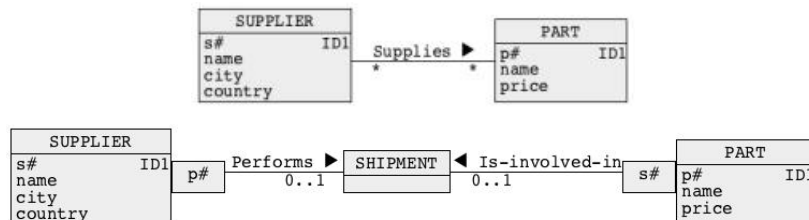


- An association class ENROLMENT is transformed in a way explained earlier

Many-to-many associations

A many-to-many association Supplies is transformed in the following way:

- A new class of objects SHIPMENT is created
- A one-to-many association Performs is added between a class SUPPLIER and a class SHIPMENT
- A one-to-many association Is-involved-in is added between a class PART and a class SHIPMENT
- A qualification with the attributes (p#) is added on SUPPLIER side of association Performs and a qualification with the attributes (s#) is added on PART side of association Is-involved-in



A qualification with an attribute p# of an association Performs contributes to an identifier (p#, s#) of a class SHIPMENT

A qualification with an attribute s# of an association Is-involved-in also contributes to an identifier (p#, s#) of a class SHIPMENT

Qualifications

A qualification with the attribute room# is transformed in the following way:

The attributes (bldg#, room#) are copied to a class ROOM

A pair of attributes (bldg#, room#) is tagged with IDn in a class ROOM (it becomes an identifier)

An attribute bldg# is tagged with FK n to denote a foreign key referencing an identifier bldg# in a class BUILDING

A multiplicity on a class ROOM side of qualified association has is changed to * or 1..*

A qualification with an attribute room# is removed

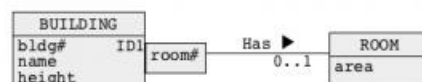
Building(bldg#, name, height)

PK:(bldg#)

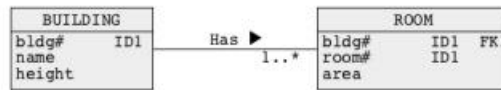
Room(bldg#, room#, area)

PK: (bldg#, room#)

FK: (bldg#) references Building (bldg#)



	bldg#	room#	area
A	1	20	
A	2	50	
A	3	100	
B	3	200	
C	3	50	



The attributes (bldg#, room#) form an identifier of a class ROOM

A attribute bldg# becomes a foreign key referencing an attribute bldg# in a class BUILDING

foreign key only exist in relational diagram, not in class diagram

Generalizations - superset method

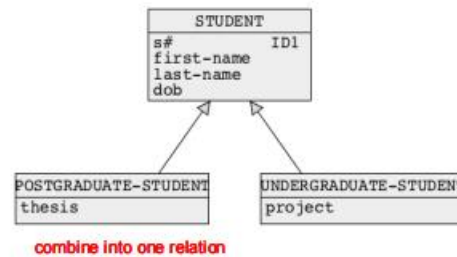
Student (s#, first-name, last-name, dob, thesis, project)

PK: (s#)

A superset method transforms entire generalization hierarchy into a single class of objects in the following way:

All attributes from the classes of objects at the lowest level of generalization hierarchy are copied to an immediate higher level and become optional attributes ([0..1] tag) there, e.g. the attributes project and thesis are copied from the classes UNDERGRADUATE-STUDENT and POSTGRADUATE-STUDENT to a class STUDENT

An attribute type-of-superclass is added to a superclass, e.g. and attribute type-of-students is added to a class STUDENT



Generalizations - subset method

Every subclass become a relation

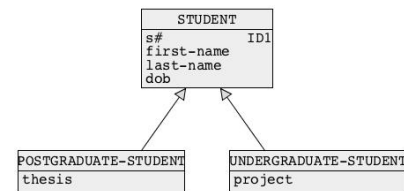
Post-Student(s#, first-name, last-name, dob, thesis)

PK:(s#)

Under-Student(s#, first-name. last-name, dob,thesis)

PK: (s#)

A subset method transforms entire generalization hierarchy into a number of classes of objects in the following way:
All attributes from the classes of objects at the higher levels of generalization hierarchy are copied to the classes of objects at the lowest levels of generalization hierarchy e.g. the attributes s# and first-name last-name, dob are copied from a class STUDENT to the classes POSTGRADUATESTUDENT and UNDERGRADUATE-STUDENT



Generalizations - association method

Every class is a relation

Student (s#, first-name, last-name, dob)

PK: (s#)

Post-Student (s#, thesis)

PK: (s#)

FK: (s#) references Student(s#)

Under-Student: (s#, project)

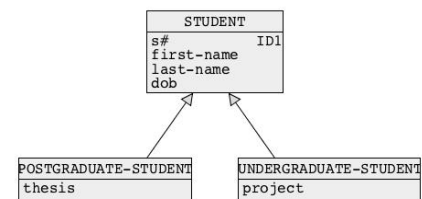
PK:(s#)

FK: (s#) references Student(s#)

An association method transforms entire generalization hierarchy into a number of classes of objects in the following way:

One of the identifiers from a superclass is copied to subclasses one level below a superclass, e.g. an attribute s# is copied from a class STUDENT to the classes UNDEGRADUATE-STUDENT and POSTGRADUATE-STUDENT

The copied identifier obtains a tag FK1 in the subclasses



Primary keys and candidate keys

Associations

Relational schemas