**Biến đổi từ sơ đồ lớp qua mô hình quan hệ hướng đối tượng**

Mục lục

[**I.** **Chuyển đổi Class** 3](#_Toc250542376)

[**1.** ***Giải pháp*** 3](#_Toc250542377)

[a. *Khai báo trực tiếp* 3](#_Toc250542378)

[b. *Khai báo gián tiếp* 3](#_Toc250542379)

[***2.*** ***Ví dụ*** 4](#_Toc250542380)

[a. *Phương pháp trực tiếp* 4](#_Toc250542381)

[b. *Phương pháp gián tiếp* 4](#_Toc250542382)

[**II.** **Chuyển đổi Attribute** 4](#_Toc250542383)

[***1.*** ***Chuyển đổi thuộc tính đơn trị*** 4](#_Toc250542384)

[a. Giải pháp 4](#_Toc250542385)

[b. Ví dụ 4](#_Toc250542386)

[***2.*** ***Chuyển đổi thuộc tính đa trị*** 4](#_Toc250542387)

[a. *Giải pháp* 4](#_Toc250542388)

[b. *Ví dụ* 5](#_Toc250542389)

[***3.*** ***Chuyển đổi tác vụ*** 5](#_Toc250542390)

[a. *Giải pháp* 5](#_Toc250542391)

[b. *Ví dụ* 5](#_Toc250542392)

[**III.** **Chuyển đổi Associate** 6](#_Toc250542393)

[***1.*** ***Quan hệ One-To-One*** 6](#_Toc250542394)

[a. *Giải pháp* 6](#_Toc250542395)

[b. *Ví dụ* 6](#_Toc250542396)

[***2.*** ***Quan hệ One-To-Many*** 7](#_Toc250542397)

[a. *Giải pháp* 7](#_Toc250542398)

[b. *Ví dụ* 7](#_Toc250542399)

[**3.** **Quan hệ Many -To-Many** 8](#_Toc250542400)

[a. *Giải pháp* 8](#_Toc250542401)

[b. *Ví dụ* 8](#_Toc250542402)

[**IV.** **Chuyển đổi Aggregation** 9](#_Toc250542403)

[***1.*** ***Giải pháp*** 9](#_Toc250542404)

[***2.*** ***Ví dụ*** 9](#_Toc250542405)

[**V.** **Chuyển đổi Composition** 9](#_Toc250542406)

[***1.*** ***Giải pháp*** 10](#_Toc250542407)

[***2.*** ***Ví dụ*** 10](#_Toc250542408)

[**VI.** **Chuyển đổi Generalisation** 10](#_Toc250542409)

1. **Chuyển đổi Class**
   1. ***Giải pháp***

Có 2 giải pháp

* + 1. *Khai báo trực tiếp*

CREATE TABLE <table name> (

// Khai báo khóa chính

<name\_column\_key> <data\_type> PRIMARY KEY,

…,

// Khai báo các column

<name\_column> <data\_type>,

…

);

* + 1. *Khai báo gián tiếp*

Gồm 2 bước

* + - * Chuyển class thành object type

CREATE TYPE OR REPLACE TYPE <type name> AS OBJECT (

// Khai báo các field

< attribute name> <data type>,

…

MEMBER PROCEDURE | FUNCTION

<procedure or function specification>,

…

);

CREATE TYPE BODY <type name>AS | IS MEMBER PROCEDURE | FUNCTION

<procedure or function name>(<parameters>)

RETURN <data type> IS <name\_function\_return> <data type>

BEGIN

<procedure or function body>

END;

END;

* + - * Tạo object table chứa các object type

CREATE TABLE <table name> [OF name\_object] (

// Khai báo khóa chính, nếu khóa chính là field của name\_object

PRIMARY KEY(name\_field),

…,

// Khai báo khóa chính, nếu khóa chính không là field của name\_object

<name\_column\_key> <data\_type> PRIMARY KEY,

…,

// Khai báo các column

<name\_column> <data\_type>,

…

);

* 1. ***Ví dụ***

Tạo object table Student:

|  |
| --- |
| Student |
| id  name  dateOfBirth  address  phone |
|  |

* + 1. *Phương pháp trực tiếp*

Create Table Student (

id varchar2(20) PRIMARY KEY,

name varchar2(50),

dateOfBirth date,

address address\_object,

phone varchar2(20)

)

* + 1. *Phương pháp gián tiếp*
       - Tạo student\_object:

Create Or Replace Type student\_object As Object (

id varchar2(20),

name varchar2(50),

dateOfBirth date,

address address\_object,

phone varchar2(20)

);

* + - * Tạo student\_table:

Create Table student\_table Of student\_object (

Primary Key (id)

);

1. **Chuyển đổi Attribute**
   1. ***Chuyển đổi thuộc tính đơn trị***
      1. Giải pháp

Thuộc tính đơn trị trong class sẽ chuyển thành

* + - * Một field trong object type nếu tạo object table bằng phương pháp gián tiếp
      * Một column trong table nếu tạo object table bằng phương pháp trực tiếp
    1. Ví dụ

Như ví dụ ở phần trước, thuộc tính id, name, … là các thuộc tính đơn.

* 1. ***Chuyển đổi thuộc tính đa trị***
     1. *Giải pháp*
        + Đối với thuộc tính đa trị, ta phải xác định số phần tử tối đa của thuộc tính này trước
        + Để chuyển đổi một thuộc tính đa trị, ta sẽ tạo 1 object type & dung từ khóa VARRAY
        + Sau đó tạo object table có chứa một attribute có kiểu object type này
     2. *Ví dụ*

Thuộc tính phone có 5 giá trị

|  |
| --- |
| Student |
| id  name  dateOfBirth  address  phone |
|  |

* + - * Định nghĩa object type cho thuộc tính phone có 5 giá trị này

Create Or Replace Type *phones\_object* **as varray(5) of** varchar2(20);

* + - * Định nghĩa object type có chứa thuộc tính có kiểu phones\_object

Create Or Replace Type student\_object As Object (

id varchar2(20),

name varchar2(50),

dateOfBirth date,

address address\_object,

phone ***phones\_object***

);

* 1. ***Chuyển đổi tác vụ***
     1. *Giải pháp*

Gồm 3 bước:

* + - * Khai báo object type

CREATE OR REPLACE TYPE name\_object AS OBJECT (

MEMBER PROCEDURE | FUNCTION

<procedure or function specification>, …

);

* + - * Hiện thực tác vụ

CREATE TYPE BODY <type name>AS | IS MEMBER PROCEDURE | FUNCTION

<procedure or function name>(<parameters>)

RETURN <data type> IS <name\_function\_return> <data type>

BEGIN

<procedure or function body>

END;

END;

* + - * Tạo object table thông qua object type

CREATE TABLE <table name> OF <name object>;

* + 1. *Ví dụ*

Student có thêm phương thức tính điểm trung bình tích luỹ cal\_gpa()

|  |
| --- |
| Student |
| id  name  dateOfBirth  address  phone |
| cal\_gpa(): float |

* + - * Bước 1: tạo student\_object

Create Or Replace Type student\_object As Object (

id varchar2(20),

name varchar2(50),

dateOfBirth date,

address address\_object,

phone varchar2(20),

member function cal\_gpa return float

);

* + - * Bước 2: hiện thực tác vụ tính điểm tích lũy cal\_gpa()

Create Or Replace Type Body student\_object As Member Function cal\_gpa

Return float Is gpa float;

Begin

gpa ≔8;

return gpa;

End;

End;

* + - * Bước 3: tạo student\_table

Create Table student\_table Of student\_object (

Primary Key(id)

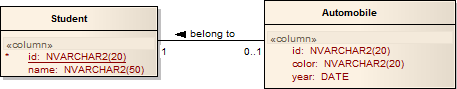
);

1. **Chuyển đổi Associate**
   1. ***Quan hệ One-To-One***
      1. *Giải pháp*

Có 3 bước

* + - * Bước 1: Chuyển class A & class B thành object type
      * Bước 2: Thêm vào object-type A 1 field tham khảo đến object-type B. Tương tự thực hiện cho object-type B.
      * Bước 3: tạo 2 object table
    1. *Ví dụ*

Sinh viên có hay không có xe moto.



* + - * Bước 1: Tạo object type

Create Or Replace Type student\_object As Object (

id varchar2(20),

name varchar2(50)

);

Create Or Replace Type automobile\_object As Object (

id varchar2(20),

color varchar2(20),

year date

);

* + - * Bước 2: Thêm field tham khảo

Create Or Replace Type student\_object As Object (

id varchar2(20),

name varchar2(50),

has REF automobile\_object

);

Create Or Replace Type automobile\_object As Object (

id varchar2(20),

color varchar2(20),

year date,

belong\_to REF student\_object

);

* + - * Bước 3: Tạo object table

Create Table student\_table Of student\_object

Create Table automobile \_table Of automobile\_object

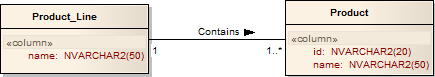
* 1. ***Quan hệ One-To-Many***
     1. *Giải pháp*

Quá trình chuyển class A (lượng số 1) và class B (lượng số nhiều) gồm 4 bước:

* + - * Bước 1: Ta thêm 1 field có tên là tên kết hợp của class A và class B vào class B. Field này có kiểu là object A.
      * Bước 2: Tạo 1 kiểu table tham chiếu đến object type B

CREATE OR REPLACE TYPE <name type> AS TABLE OF REF <object type B>

* + - * Bước 3: Thêm 1 field vào object type A có kiểu là kiểu bảng tham chiếu object type B.
      * Bước 4: Tạo object table cho các object type A & object type B Sau đó tạo 1 nested table.
    1. *Ví dụ*



* + - * Bước 1: tạo object type

Create Or Replace Type product\_line\_object As Object (

name varchar2(50)

);

Create Or Replace Type product\_object As Object (

id varchar2(20),

name varchar2(50),

isContained Ref product\_line\_object

);

* + - * Bước 2: tạo kiểu table tham chiếu đến product\_object

Create Or Replace Type product\_object\_table As Table Of Ref product\_object

* + - * Bước 3: thêm một field có kiểu product\_object\_table vào trong object type

Create Or Replace Type product\_line\_object As Object (

name varchar2(50),

contains product\_object\_table

);

* + - * Bước 4: tạo object table

Create Table product Of product\_object (

Primary Key (id)

);

Create Table product\_line Of product\_line\_object (

Primary Key (id)

);

Nested Table contains Store as table\_product;

* 1. **Quan hệ Many -To-Many**
     1. *Giải pháp*

Chuyển đổi class A & class B gồm 4 bước:

* + - * Bước 1: Tạo object type cho class A và class B
      * Bước 2: Tạo kiểu bảng tham chiếu đến object type A & object type B

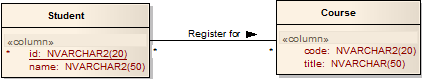
Create Or Replace Type <name type A> As Table Of Ref <object type A>

Create Or Replace Type <name type B> As Table Of Ref <object type B>

* + - * Bước 3: Thêm 1 field vào các object type với kiểu bảng vừa được tạo ở trên
      * Bước 4: Tạo các object table của object type A & object type B. Tạo thêm 1 field như sau:

Nested Table <name field> Store As <name table>

* + 1. *Ví dụ*



* + - * Bước 1: Tạo object type cho class student & class course

Create Or Replace Type student\_object As Object (

id varchar2(20),

name varchar2(50)

);

Create Or Replace Type course\_object As Object (

code varchar2(20),

title varchar2(50)

);

* + - * Bước 2: Tạo kiểu bảng tham chiếu đến student\_object & course\_object

Create Or Replace Type student\_table\_object As Table Of Ref student\_object

Create Or Replace Type course\_table\_object As Table Of Ref course\_object

* + - * Bước 3: Thêm field vào student\_object & course\_object

Create Or Replace Type student\_object As Object (

id varchar2(20),

name varchar2(50),

register\_for course\_table\_object

);

Create Or Replace Type course\_object As Object (

code varchar2(20),

title varchar2(50),

registrants student\_table\_object

);

* + - * Bước 4: Tạo các object table

Create Table student Of student\_object (

Primary Key (id)

);

Nested Table register\_for Store As table\_course;

Create Table course Of student\_object (

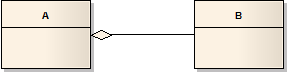
Primary Key (id)

);

Nested Table registrants Store As table\_student;

Với các ràng buộc lượng số có giới hạn trên, chúng ta có thể thay thế cách hiện thực nested table bằng cách hiện thực varray.

1. **Chuyển đổi Aggregation**



* 1. ***Giải pháp***

Chuyển đổi gồm 4 bước như sau:

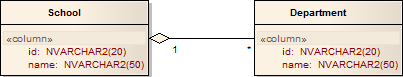
* + 1. Bước 1: Tạo object type B
    2. Bước 2:
       - Tạo kiểu bảng tham khảo đến object type B nếu lượng số của B là nhiều (\*)

Create Or Replace Type <list object type B> As Table Of Ref <object type B>;

* + - * Tạo kiểu varray tham khảo đến object type B nếu lượng số của B có cận trên

Create Or Replace Type <list object type B> As Varray(<n>) Of Ref <object type B>;

* + 1. Bước 3: Tạo object type A chứa 1 field có kiểu là <list object type B>
    2. Bước 4: Tạo object table A, object table B. Trong table A tạo nested table chứa B trong trường hợp <list object type B> là nested table.
  1. ***Ví dụ***



* + 1. Bước 1: tạo object department

Create Or Replace Type department\_object As Object (

id varchar2(20),

name varchar2(50)

);

* + 1. Bước 2: tạo kiểu bảng tham kháo đến object department

Create Or Replace Type department\_table\_object As Table Of Ref department\_object;

* + 1. Bước 3: tạo object school

Create Or Replace Type school\_object As Object (

id varchar2(20),

name varchar2(50),

consists\_of department\_table\_object

);

* + 1. Bước 4: tạo object table

Create Table department Of department\_object (

Primary Key (id)

);

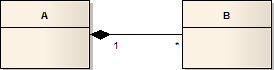
Create Table school Of school\_object (

Primary Key (id)

);

Nested Table consists\_of Store As table \_department;

1. **Chuyển đổi Composition**



* 1. ***Giải pháp***

Gồm 4 bước:

* + 1. Bước 1: Tạo object B
    2. Bước 2:
       - Tạo kiểu bảng tham khảo đến object B nếu không giới hạn số lượng B

Create Or Replace Type <list object type B> As Table Of Ref <object type B>;

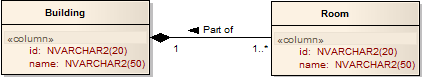
* + - * Tạo kiểu <list object type B> là varray tham khảo đến object B nếu giới hạn số lượng B

Create Or Replace Type <list object type B> As Varray(<n>) Of Ref <object type B>;

* + 1. Bước 3: Tạo object A
    2. Bước 4: Tạo object table A & object table B. Tạo nested table chứa B trong trường hợp <list object type B> là nested table.

Chúng ta phải định nghĩa assertion hay trigger để hiện thực các ngữ nghĩa của composition.

* 1. ***Ví dụ***



* + 1. Bước 1: tạo object Room

Create Or Replace Type room\_object As Object (

id varchar2(20),

name varchar2(50)

);

* + 1. Bước 2: tạo kiểu bảng tham khảo đến object room

Create Or Replace Type room\_table\_object As Table Of Ref room\_object;

* + 1. Bước 3: tạo object A

Create Or Replace Type building\_object As Object (

id varchar2(20),

name varchar2(50),

has room\_table\_object

);

* + 1. Bước 4: tạo object tables & nested table

Create Table room Of room\_object (

Primary Key (id)

);

Create Table building Of building \_object (

Primary Key (id)

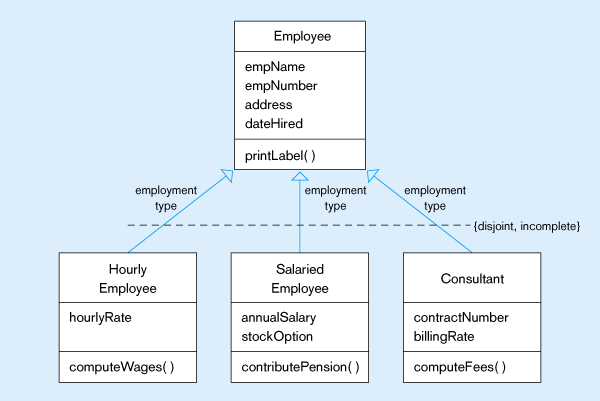
);

Nested Table has Store As table \_room;

1. **Chuyển đổi Generalisation**

Oracle 8i không hổ trợ kế thừa. Do đó, generalization hiện thực thông qua khóa ngoại (foreign keys) như mô hình quan hệ hay dùng kiểu REF. Ngoài ra, chúng ta cũng phải hiện thực các ràng buộc (check, assertions & triggers) hay các tác vụ thực hiện các ràng buộc này.

* 1. **Ví dụ 1**



Trường hợp Disjoint, Incomplete

* + 1. Tạo các object type

Create Or Replace Type employee\_object As Object (

id number,

empName varchar2(20),

empNumber number,

address varchar2(50),

dateHired date,

member function printLabel( ) return void

);

Create Or Replace Type hourly\_employee\_object As Object (

employee employee\_object,

hourlyRate number,

member function computeWages( ) return number

);

Create Or Replace Type salaried\_employee\_object As Object (

employee employee\_object,

annualSalary number,

stockOptions varchar2(20),

member function computePension( ) return number

);

Create Or Replace Type consultant\_object As Object (

employee employee\_object,

contractNumber number,

billingRate number,

member function computeFees( ) return number

);

* + 1. Thêm vào employee\_object các field tham khảo đến các con của nó. Các field này có thể là null để thể hiện incomplete

Create Or Replace Type employee\_object As Object (

id number,

empName varchar2(20),

empNumber number,

address varchar2(50),

dateHired date,

isHourlyEmployee Ref hourly\_employee\_object,

isSalariedEmployee Ref salaried\_employee\_object,

isConsultantEmployee Ref consultant\_object,

member function printLabel( ) return void

);

* + 1. Tạo các object table

Create Table employee Of employee\_object (

Primary Key (id)

);

Create Table hourly\_employee Of hourly\_employee\_object (

Primary Key (employee.id)

);

Create Table salaried\_employee Of salaried\_employee\_object (

Primary Key (employee.id)

);

Create Table consultant Of consultant \_object (

Primary Key (employee.id)

);

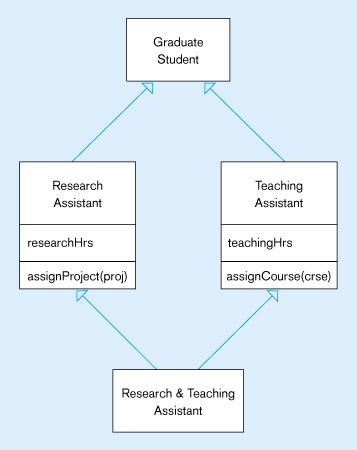
* + 1. Chúng ta thực hiện thêm ràng buộc để kiểm tra tính chất disjoint có vi phạm không. Cụ thể là kiểm tra xem trong 1 thời điểm thì isHourlyEmployee, isSalariedEmployee và isConsultantEmployee chỉ có thể có nhiều nhất 1 tham khảo là có giá trị khác null. Nghĩa là lúc đó employee sẽ là 1 trong các loại employee (***disjoint***) hay không là loại employee nào cả (***incomplete***).

Do đó, đối với trường hợp ***complete*** thì chúng ta sẽ hiện thực hàm kiểm tra xem tại 1 thời điểm thì isHourlyEmployee, isSalariedEmployee và isConsultantEmployee có đúng 1 tham khảo có giá trị khác null.

Trường hợp ***overlapping*** thì không cần hiện thực hàm kiểm tra này.

Nếu kế thừa có yếu tố dynamic thì ta thêm vào thuộc tính time\_start & time\_end có kiểu date.

* 1. **Ví dụ 2**: Đa kế thừa

****

* + 1. Tạo các object type

Create Or Replace Type graduate\_student\_object As Object ( );

Create Or Replace Type research\_assistant\_object As Object (

student graduate\_student\_object,

researchHrs number,

member function assignProject(proj in project) return void

);

Create Or Replace Type teaching\_assistant\_object As Object (

student graduate\_student\_object,

researchHrs number,

member function assignCourse(crse in course) return void

);

Create Or Replace Type research\_teaching\_assistant\_object As Object (

isRearchAssistant Ref research\_assistant\_object,

isTeachingAssistant Ref teaching\_assistant\_object

);

* + 1. Tạo các object table

Create Table graduate\_student Of graduate\_student\_object ( Primary Key (id) );

Create Table research\_assistant Of research\_assistant\_object ( Primary Key (student.id) );

Create Table teaching\_assistant Of teaching\_assistant\_object ( Primary Key (student.id) );