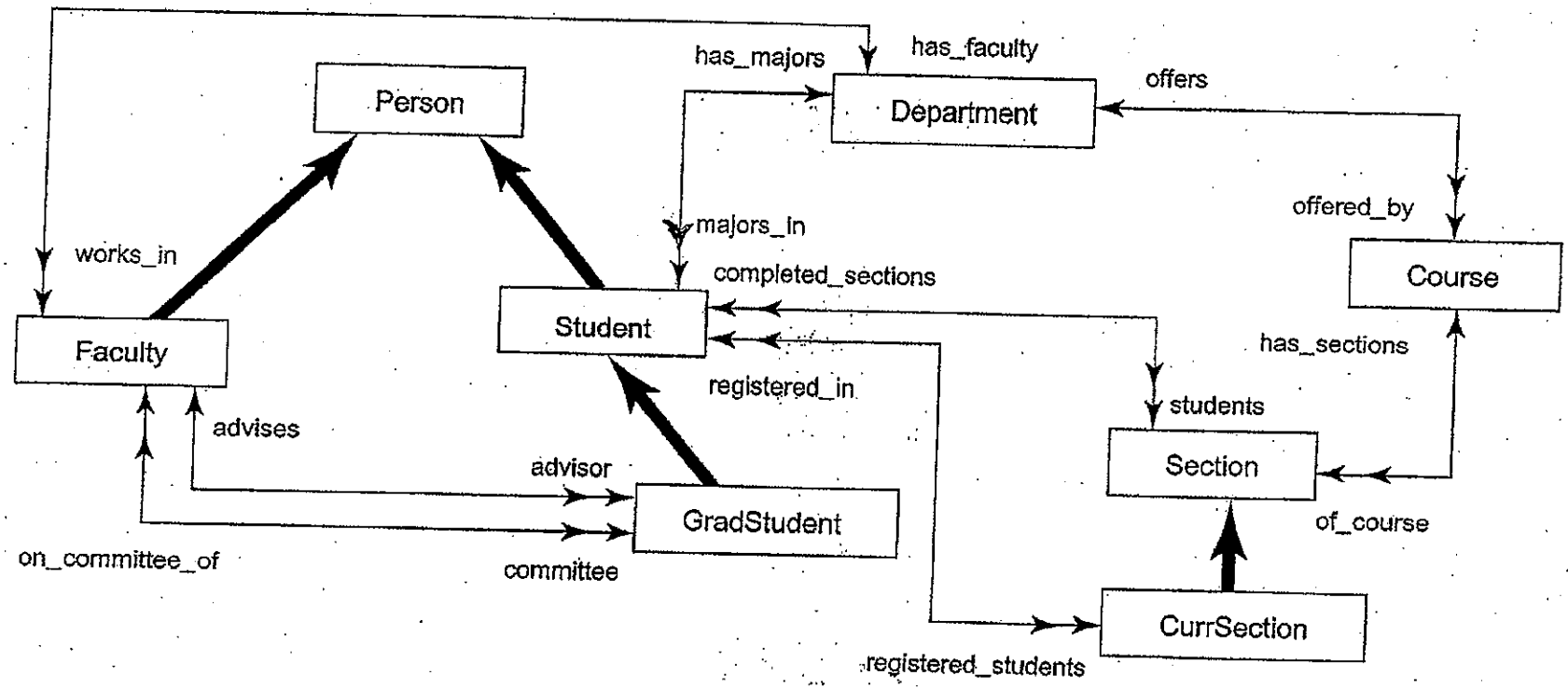
Figure 12.5 An example of a database schema. (b) A graphical object database schema for part of the UNIVERSITY database.



© Addison Wesley Longman, Inc. 2000, Elmasri/Navathe, Fundamentals of Database Systems, Third Edition

```
class PERSON
(  extent      PERSONS
   key         Ssn )
{  attribute   struct Pname {   string   Fname,
                                string   Mname,
                                string   Lname }   Name;
   attribute   string                              Ssn;
   attribute   date                                Birth_date;
   attribute   enum Gender(M, F}                   Sex;
   attribute   struct Address {   short    No,
                                  string   Street,
                                  short    Apt_no,
                                  string   City,
                                  string   State,
                                  short    Zip }   Address;
   short       Age();   };

class FACULTY extends PERSON
(  extent      FACULTY )
{  attribute   string          Rank;
   attribute   float           Salary;
   attribute   string          Office;
   attribute   string          Phone;
   relationship DEPARTMENT     Works_in inverse DEPARTMENT::Has faculty;
   relationship set<GRAD_STUDENT> Advises inverse GRAD_STUDENT::Advisor;
   relationship set<GRAD_STUDENT> On_committee_of inverse GRAD_STUDENT::Committee;
   void        give_raise(in float raise);
   void        promote(in string new rank);   };

class GRADE
(  extent      GRADES )
{
   attribute   enum GradeValues{A,B,C,D,Fl, P} Grade;
   relationship SECTION Section inverse SECTION::Students;
   relationship STUDENT Student inverse STUDENT::Completed_sections;   };

class STUDENT extends PERSON
(  extent      STUDENTS )
{  attribute   string          Class;
   attribute   Department      Minors_in;
   relationship Department Majors_in inverse DEPARTMENT::Has_majors;
   relationship set<GRADE> Completed_sections inverse GRADE::Student;
   relationship set<CURR_SECTION> Registered_in inverse CURR_SECTION::Registered_students;
   void        change_major(in string dname) raises(dname_not_valid);
   float       gpa();
   void        register(in short secno) raises(section_not_valid);
   void        assign_grade(in short secno; in GradeValue grade)
                  raises(section_not_valid,grade_not_valid);   };
```

Figure 21.6
Possible ODL schema for the UNIVERSITY database of Figure 21.5(b) (continued).

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley

Figure 21.6 (continued)
Possible ODL schema for the UNIVERSITY database of Figure 21.5(b).

```
class DEGREE
( attribute      string      College;
  attribute      string      Degree;
  attribute      string      Year;    );

class GRAD_STUDENT extends STUDENT
( extent         GRAD_STUDENTS )
{ attribute set<Degree>      Degrees;
  relationship Faculty Advisor inverse FACULTY::Advises;
  relationship  set<FACULTY> Committee inverse FACULTY::On_committee_of;
  void         assign_advisor(in string Lname; in string Fname)
                 raises(faculty_not_valid);
  void         assign_committee_member(in string Lname; in string Fname)
                 raises(faculty_not_valid); };

class DEPARTMENT
( extent         DEPARTMENTS
  key            Dname )
{ attribute      string      Dname;
  attribute      string      Dphone;
  attribute      string      Doffice;
  attribute      string      College;
  attribute      FACULTY     Chair;
  relationship   set<FACULTY> Has_faculty inverse FACULTY::Works_in;
  relationship   set<STUDENT> Has_majors inverse STUDENT::Majors_in;
  relationship   set<COURSE> Offers inverse COURSE::Offered_by; };

class COURSE
( extent         COURSES
  key            Cno )
{ attribute      string      Cname;
  attribute      string      Cno;
  attribute      string      Description;
  relationship set<SECTION> Has_sections inverse SECTION::Of_course;
  relationship <DEPARTMENT> Offered_by inverse DEPARTMENT::Offers; };

class SECTION
( extent         SECTIONS )
{ attribute      short       Sec_no;
  attribute      string      Year;
  attribute      enum Quarter{Fall, Winter, Spring, Summer}
                             Qtr;
  relationship set<Grade> Students inverse Grade::Section;
  relationship COURSE Of_course inverse COURSE::Has_sections; };

class CURR_SECTION extends SECTION
( extent         CURRENT_SECTIONS )
{ relationship set<STUDENT> Registered_students
                   inverse STUDENT::Registered_in
  void         register_student(in string Ssn)
                   raises(student_not_valid, section_full); };
```

Copyright © 2007 Pearson Education, Inc. Publishing as Pearson Addison-Wesley