Lab 3

Xiaohui Chen
CS ID : xc2388
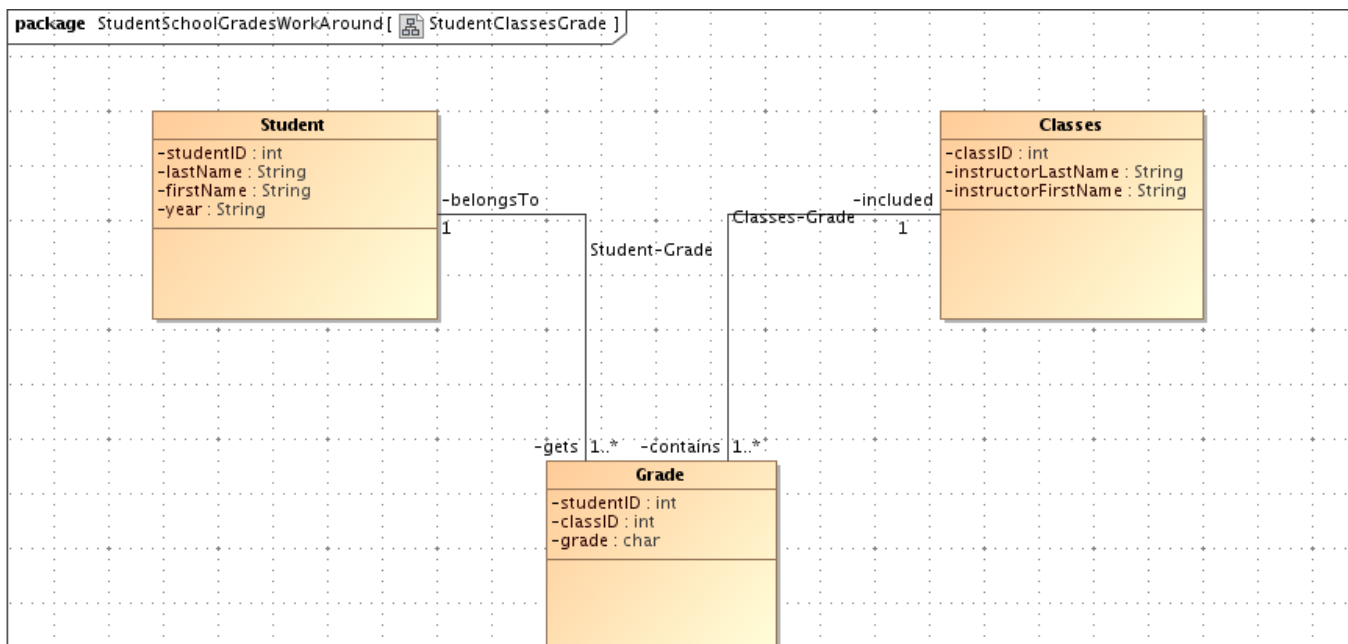EID: xc2388

Student and classes
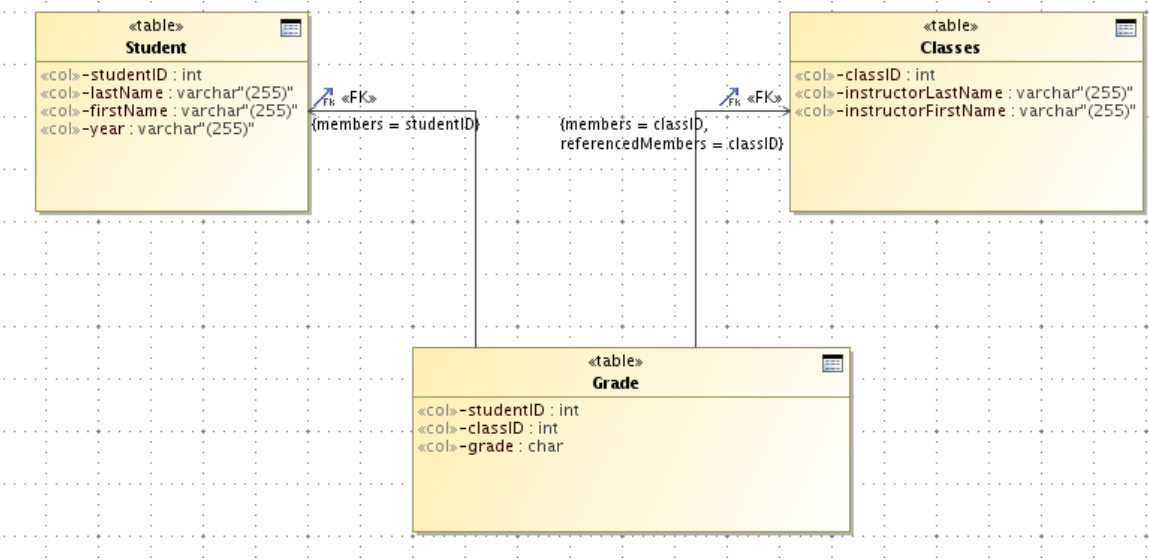
UML model:

```
package StudentSchoolGrades [ 🔠 StudentClassesDiagram ]

        Student                                              Classes
  -studentID : int                                    -classID : int
  -lastName : String                                  -instructorLastName : String
  -firstName : String      -attend      Grade     -contain  -instructorFirstName : String
  -year : String                                        1..*
                           1..*

                                     Grade
                                 -studentID : int
                                 -classID : int
                                 -grade : char
```

Work around model:

```
package StudentSchoolGradesWorkAround [ 🔠 StudentClassesGrade ]

        Student                                              Classes
  -studentID : int                                    -classID : int
  -lastName : String                                  -instructorLastName : String
  -firstName : String    -belongsTo         -included -instructorFirstName : String
  -year : String            1          Classes-Grade      1

                          Student-Grade

                  -gets 1..*  -contains 1..*
                               Grade
                           -studentID : int
                           -classID : int
                           -grade : char
```

Physical Model:

Code:

CREATE TABLE StudentSchoolGradesPhysical.StudentSchoolGradesWorkAround.Classes

```
(

    classID int,

    instructorLastName varchar (255),

    instructorFirstName varchar (255),

    PRIMARY KEY(classID)

);
```

---SKIPPED: Class REMOVED_BY_CUSTOM_TRANSFORMATION

CREATE TABLE StudentSchoolGradesPhysical.StudentSchoolGradesWorkAround.Student

```
(

    studentID int,

    lastName varchar (255),

    firstName varchar (255),

    year varchar (255),

    PRIMARY KEY(studentID)

);
```

CREATE TABLE StudentSchoolGradesPhysical.StudentSchoolGradesWorkAround.Grade

(

    studentID int,

    classID int,

    grade char,

    id_Grade integer,

    PRIMARY KEY(studentID, classID),

    FOREIGN KEY(studentID) REFERENCES StudentSchoolGradesPhysical.StudentSchoolGradesWorkAround.Student (studentID),

    FOREIGN KEY(classID) REFERENCES StudentSchoolGradesPhysical.StudentSchoolGradesWorkAround.Classes (classID)

);

Santa Claus list:

UML:
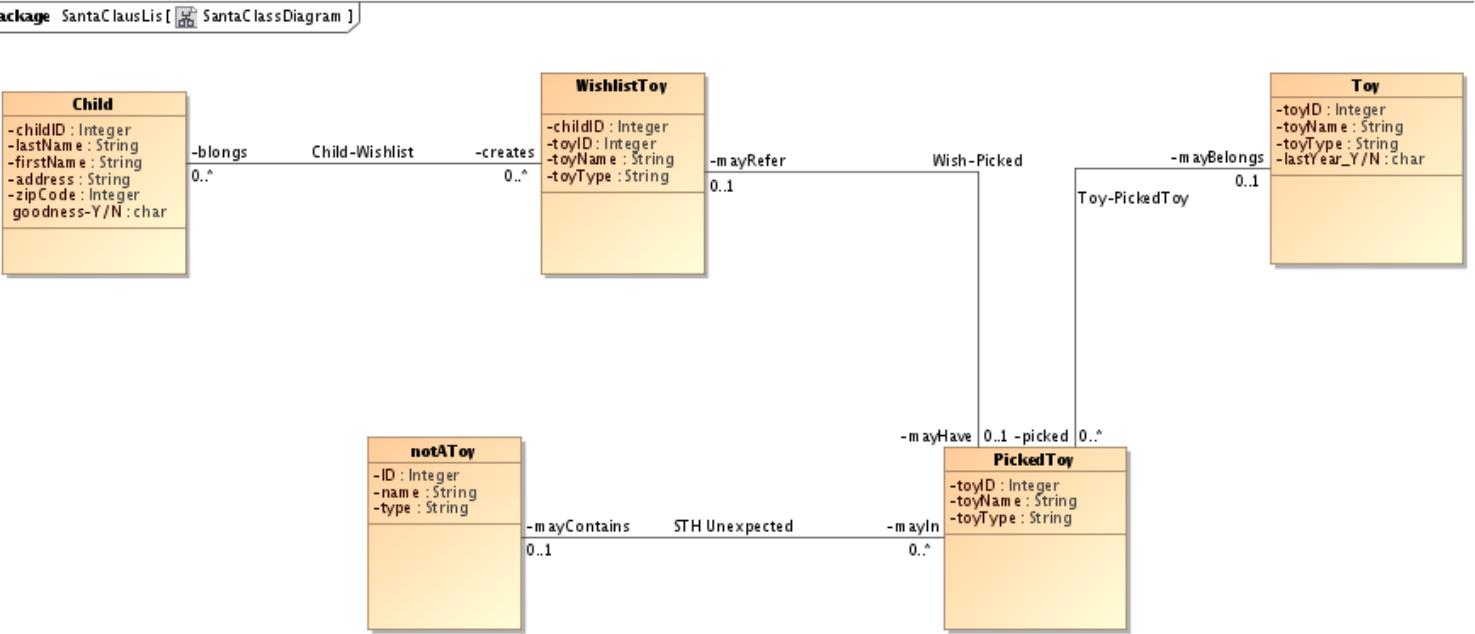
Code:

CREATE TABLE SantaPhysical.SantaClausLis.Child

(

```sql
    childID integer,

    lastName varchar (255),

    firstName varchar (255),

    address varchar (255),

    zipCode integer,

    goodness-Y/N char,

    PRIMARY KEY(childID)

);


CREATE SEQUENCE SantaPhysical.SantaClausLis.PickedToy_SEQ;


CREATE SEQUENCE SantaPhysical.SantaClausLis.WishlistToy
_SEQ;


CREATE SEQUENCE SantaPhysical.SantaClausLis.Child_SEQ;


CREATE SEQUENCE SantaPhysical.SantaClausLis.notAToy_SEQ;


CREATE SEQUENCE SantaPhysical.SantaClausLis.Toy_SEQ;

    ---SKIPPED: Class REMOVED_BY_CUSTOM_TRANSFORMATION


CREATE TABLE SantaPhysical.SantaClausLis.Toy

(

    toyID integer,

    toyName varchar (255),
```

```
        toyType varchar (255),

        lastYear_Y/N char,

        PRIMARY KEY(toyID)

);


CREATE TABLE SantaPhysical.SantaClausLis.notAToy

(

        ID integer,

        name varchar (255),

        type varchar (255),

        PRIMARY KEY(ID)

);


CREATE TABLE SantaPhysical.SantaClausLis.WishlistToy

(

        childID integer,

        toyID integer,

        toyName varchar (255),

        toyType varchar (255),

        PRIMARY KEY(childID, toyID),

        UNIQUE(null),

        FOREIGN KEY(childID) REFERENCES SantaPhysical.SantaClausLis.Child (childID)

);


CREATE TABLE SantaPhysical.SantaClausLis.PickedToy
```
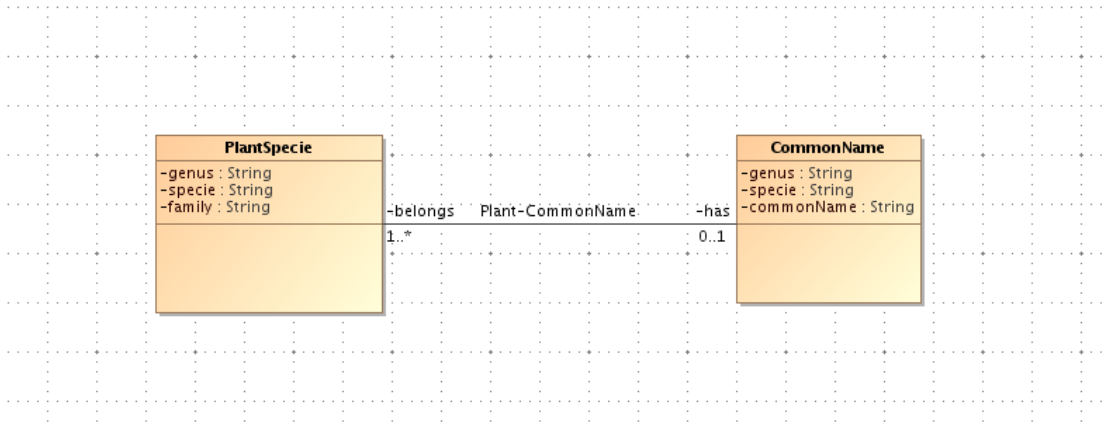
(

    toyID integer,

    toyName varchar (255),

    toyType varchar (255),

    PRIMARY KEY(toyID),

    UNIQUE(null),

    FOREIGN KEY(toyID) REFERENCES SantaPhysical.SantaClausLis.WishlistToy (toyID),

    FOREIGN KEY(toyID) REFERENCES SantaPhysical.SantaClausLis.notAToy (ID),

    FOREIGN KEY(toyID) REFERENCES SantaPhysical.SantaClausLis.Toy (toyID)
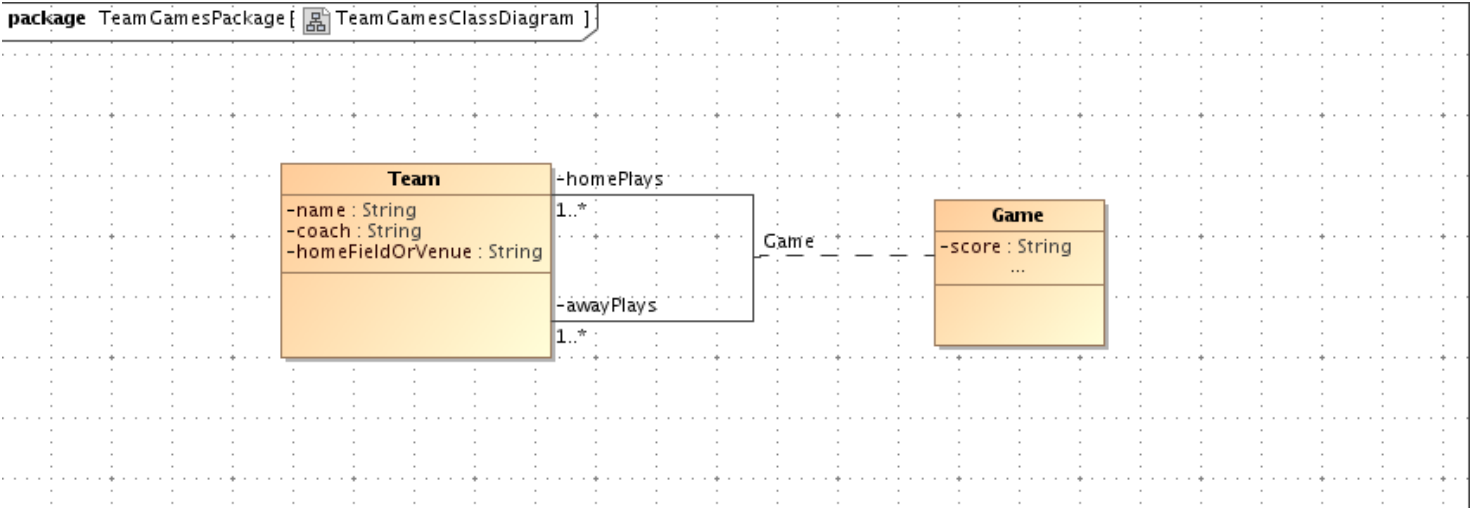
);

Plant Species:

Logical Model:
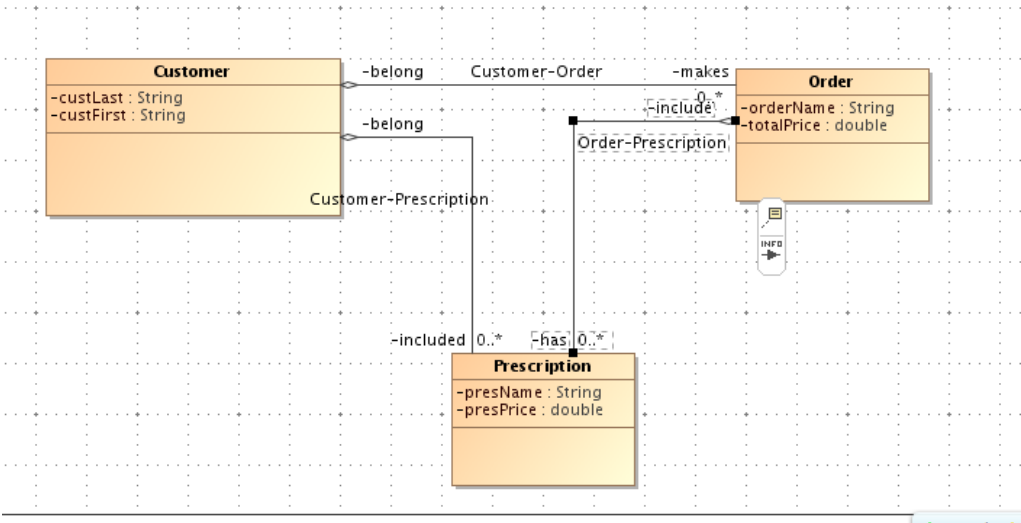

Physical Model:

Team Games:
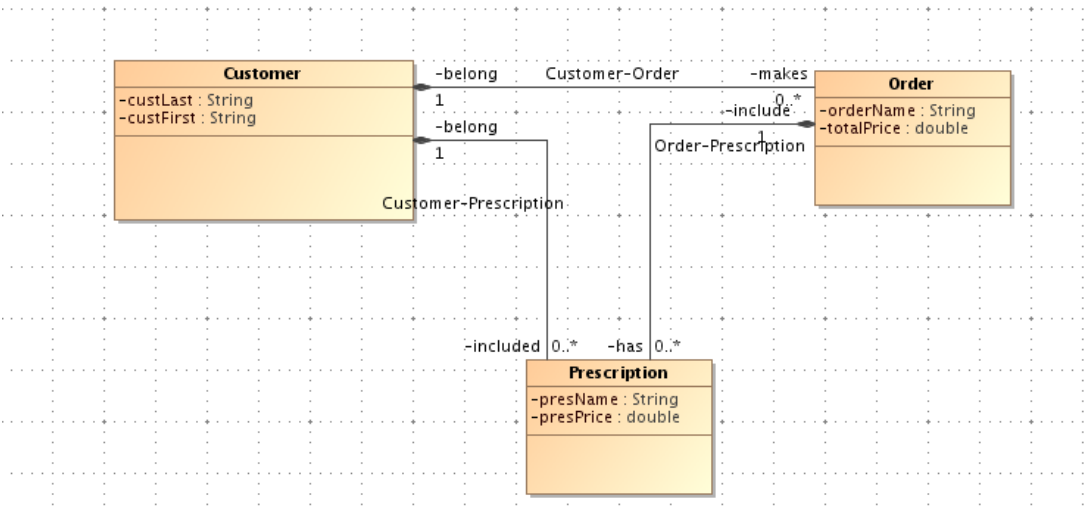

package TeamGamesPackage [🔲 TeamGamesClassDiagram]
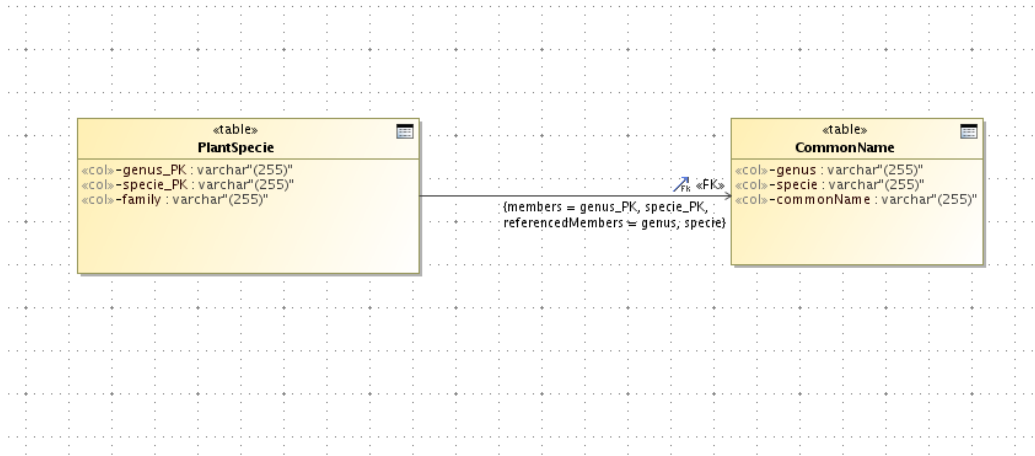
Part 2

1)

Logical Models:

Aggregation:

Composition:

Physical Model:

Aggregation:

| «table» Customer |
| «col» –custLast : varchar"(255)" |
| «col» –custFirst : varchar"(255)" |
| «col» –id_Customer : integer |

«FK» {members = fk_Customerid_Customer, referencedMembers = id_Customer}

«FK» {members = fk_Customerid_Customer, referencedMembers = id_Customer}

| «table» Order |
| «col» –orderName : varchar"(255)" |
| «col» –totalPrice : double precision |
| «col» –id_Order : integer |
| «col» –fk_Customerid_Customer : integer{nullable = false} |

«FK» {members = fk_Orderid_Order, referencedMembers = id_Order}

| «table» Prescription |
| «col» –presName : varchar"(255)" |
| «col» –presPrice : double precision |
| «col» –id_Prescription : integer |
| «col» –fk_Orderid_Order : integer{nullable = false} |
| «col» –fk_Customerid_Customer : integer{nullable = false} |

Composition:

package PharmacyPackage2 [Untitled1]

| «table» Customer |
| «col» –custLast : varchar"(255)" |
| «col» –custFirst : varchar"(255)" |
| «col» –id_Customer : integer |

«FK» {members = fk_Customerid_Customer referencedMembers = id_Customer}

«FK» {members = fk_Customerid_Customer referencedMembers = id_Customer}

| «table» Order |
| «col» –orderName : varchar"(255)" |
| «col» –totalPrice : double precision |
| «col» –id_Order : integer |
| «col» –fk_Customerid_Customer : integer{nullable = false} |

«FK» {members = fk_Orderid_Order referencedMembers = id_Order}

| «table» Prescription |
| «col» –presName : varchar"(255)" |
| «col» –presPrice : double precision |
| «col» –id_Prescription : integer |
| «col» –fk_Orderid_Order : integer{nullable = false} |
| «col» –fk_Customerid_Customer : integer{nullable = false} |

Code:

Aggregation:

CREATE TABLE unnamed1.PharmacyPackage2.Customer

(

    custLast varchar (255),

    custFirst varchar (255),

    id_Customer integer,

    PRIMARY KEY(id_Customer)

);

CREATE SEQUENCE unnamed1.PharmacyPackage2.Prescription_SEQ;

CREATE SEQUENCE unnamed1.PharmacyPackage2.Order_SEQ;

CREATE SEQUENCE unnamed1.PharmacyPackage2.Customer_SEQ;

---SKIPPED: Class REMOVED_BY_CUSTOM_TRANSFORMATION

CREATE TABLE unnamed1.PharmacyPackage2.Order

(

orderName varchar (255),

totalPrice double precision,

id_Order integer,

fk_Customerid_Customer integer NOT NULL,

PRIMARY KEY(id_Order),

FOREIGN KEY(fk_Customerid_Customer) REFERENCES unnamed1.PharmacyPackage2.Customer (id_Customer)

);

CREATE TABLE unnamed1.PharmacyPackage2.Prescription

(

presName varchar (255),

presPrice double precision,

id_Prescription integer,

fk_Orderid_Order integer NOT NULL,

fk_Customerid_Customer integer NOT NULL,

==PRIMARY KEY(id_Prescription),==

FOREIGN KEY(fk_Orderid_Order) REFERENCES unnamed1.PharmacyPackage2.Order (id_Order),

FOREIGN KEY(fk_Customerid_Customer) REFERENCES unnamed1.PharmacyPackage2.Customer (id_Customer)

);

Composition:
CREATE TABLE unnamed1.PharmacyPackage2.Customer

(

    custLast varchar (255),

    custFirst varchar (255),

    id_Customer integer,

    PRIMARY KEY(id_Customer)

);

CREATE SEQUENCE unnamed1.PharmacyPackage2.Prescription_SEQ;

CREATE SEQUENCE unnamed1.PharmacyPackage2.Order_SEQ;

CREATE SEQUENCE unnamed1.PharmacyPackage2.Customer_SEQ;

    ---SKIPPED: Class REMOVED_BY_CUSTOM_TRANSFORMATION

CREATE TABLE unnamed1.PharmacyPackage2.Order

(

    orderName varchar (255),

    totalPrice double precision,

    id_Order integer,

fk_Customerid_Customer integer NOT NULL,

PRIMARY KEY(id_Order, fk_Customerid_Customer),

FOREIGN KEY(fk_Customerid_Customer) REFERENCES unnamed1.PharmacyPackage2.Customer (id_Customer)

);

CREATE TABLE unnamed1.PharmacyPackage2.Prescription

(

presName varchar (255),

presPrice double precision,

id_Prescription integer,

fk_Orderid_Order integer NOT NULL,

fk_Customerid_Customer integer NOT NULL,

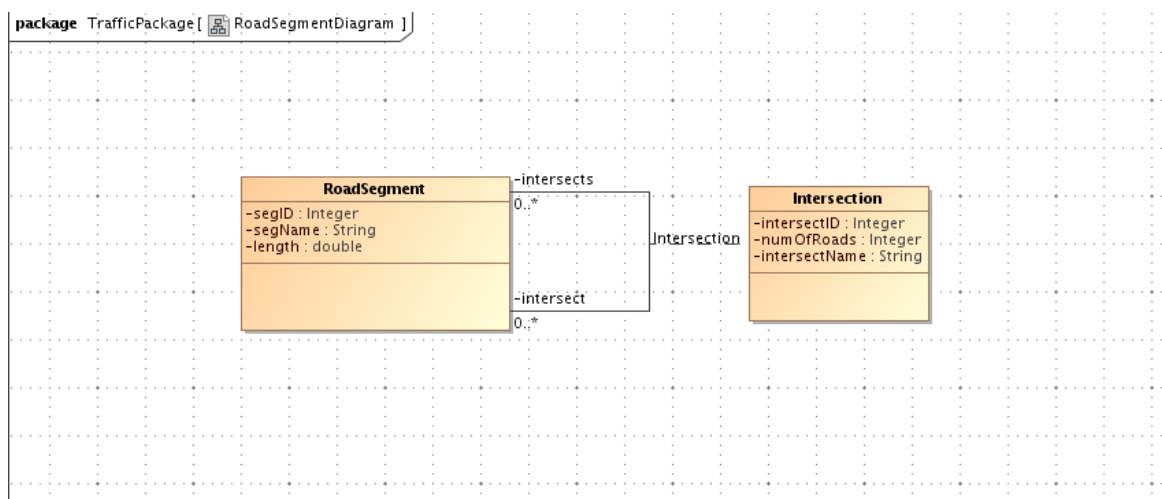PRIMARY KEY(id_Prescription, fk_Orderid_Order, fk_Customerid_Customer),

FOREIGN KEY(fk_Orderid_Order) REFERENCES unnamed1.PharmacyPackage2.Order (id_Order),

FOREIGN KEY(fk_Customerid_Customer) REFERENCES unnamed1.PharmacyPackage2.Customer (id_Customer)
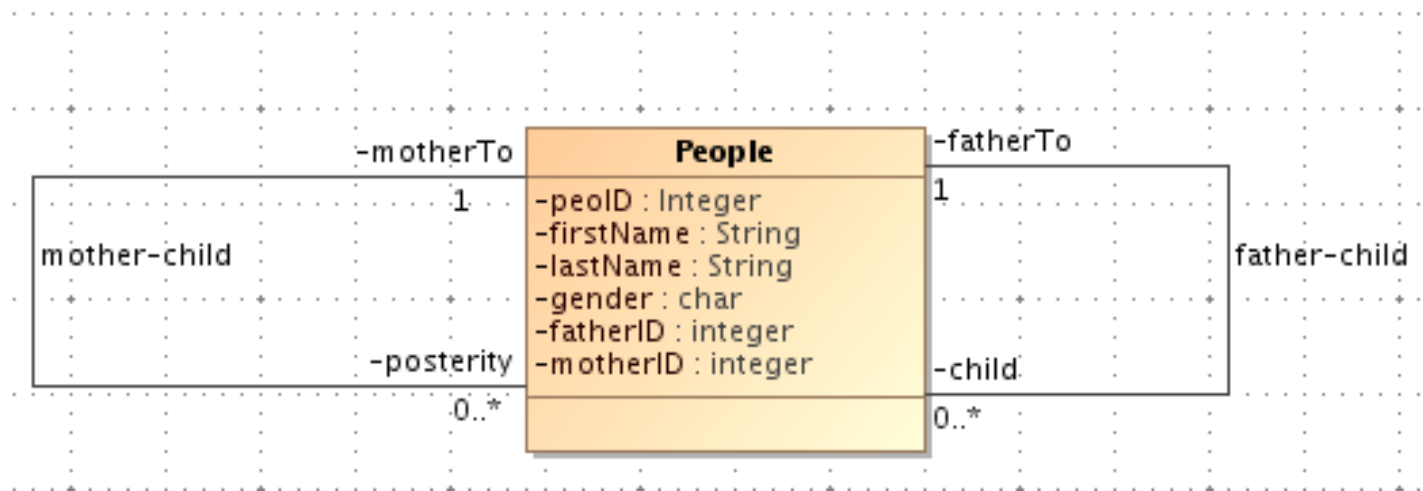
);

2)

3)

Logical Model:

CREATE SEQUENCE unnamed1.PeoplePackage.People_SEQ;

CREATE TABLE unnamed1.PeoplePackage.People

(

    peoID integer,

    firstName varchar (255),

    lastName varchar (255),

    gender char,

    fatherID integer,

    motherID integer,

    PRIMARY KEY(peoID),

    FOREIGN KEY(fatherID, motherID) REFERENCES unnamed1.PeoplePackage.People (peoID)

);

    ---SKIPPED: Class REMOVED_BY_CUSTOM_TRANSFORMATION