*, foreign key(lid) references Lesson(lid))"
προσθήκη ξένου κλειδιού

```
class database
   QSqlQuery q(dB);
   public:
    database(QString n)
   {
       dB=QSqlDatabase::addDatabase("QSQLITE");
         dB.setDatabaseName(n);
         dB.open();
       QSqlQuery q(dB);
       q.exec("Create table if not exists Lesson(
lid Integer,lessoname varchar(20),primary key(lid))");
       //κατασκευή πίνακα Lesson
       q.exec("create table if not exists Enroll(id inte
ger,lessonid integer,grade double,primary key(id,lessonid)
)"); //κατασκευή πίνακα Enroll   *
  };
   ~database()   {dB.close();}
    void insert_lesson(int lid,QString ln)
   {  //εισαγωγή τιμών.
       QSqlQuery q(dB);
             prepare
       q.~~exec~~("insert into Lesson(lid,ln) values(?,?)");
       q.addBindValue(lid);
       q.addBindValue(ln);
       q.exec();
   }
   void insert_Enroll(int id,int lid,double gr)
  {  // εισαγωγή εγγεγραμμένων.
     QSqlQuery q(dB);
     q.prepare("insert into Enroll(id,lid,grade) value(
~~les~~studentid,?,?)");
    q.addBindValue(lid);
    q.addBindValue(gr);
q.bindValue(":studentid", id);
q.exec(); //εισαγωγή στον πίνακα Enroll.
}
bool lesson_is_in(int lid)   // Έλεγχος αν το μάθημα
{                            // υπάρχει.
  QSqlQuery q(dB);
  q.exec("select * from LESSON where lid="+QString
::number(lid));
  return q.next();
}

bool enroll_is_in(int sid,int lid) // έλεγχος αν
{                                  // υπάρχει εγγεγραμμένος
                                   // ήδη
   QSqlQuery q(dB);
   q.prepare("select * from Enroll where id=?
and    lessonid=:lid");
   q.addBindValue(sid);      // Έλεγχος αν ο εγγεγραμ-
   q.bindValue(":lid", lid);  // μένος υπάρχει μέσα
   q.exec();
   return q.next();
}

 struct enroll
 {
   int lid;
   int sid;
   double grade;
 }

                        Class Lesson
                        {
                          private:
                           int lid;
                           QString ln;
                          public:
                           Lesson(int id,QString
                          n):lid(id),ln(n) {}
                           int getid() const
                           {return lid;}
                           QString get_ln() const
                           {return ln;}
                           QString to_str()
                           { return QString::number
                           (lid)+" -- "+ln;
                           }
                        };
```

κλάση οντότητας
και δομή οντότητας

```
QVector <Lesson>  getlessons()
{
    QVector <Lesson>  lessons;

    QSqlQuery  q(dB);
    q.exec("Select *  from Lesson ");
    While(q.next())
    {
        lesson  l(q.value(0).toInt(), q.value(1).toString());
        lessons<<l;
    }
    return  lessons;
}
void  update_grade(int sid, int lid, double g)
{
    QSqlQuery  q(dB);
    q.prepare("Update  Enroll  set  grade=:g  where
lessonid=?, and  id=?");
     q.addBindValue(lid);
     q.addBindValue(sid);
     q.bindValue(":g", g);
     q.exec();
}
```

OR

```
q.exec("Update Enroll
set grade="+QString::number
(g)+" where  id="+QString
::number(sid)+"  and  lessonid
="+QString::number(lid));
```

```
void  delete_lesson(int lid)
{
   QSqlQuery  q(dB);
   q.exec("delete  from  Lesson  where  lid="+
QString::number(lid));
   q.exec("delete  from  Enroll  where  lessonid="
+QString::number(lid));
}

QStringList  getEnrollLessons()
{
  QStringList   list;
  QSqlQuery   q(dB);
  q.exec("Select distinct lid from Enroll ");
  while(q.next())
 {
   QSqlQuery  query(dB);
   query.exec("Select lessoname from Lesson where
lid= "+ q.value(0).toString());
    query.next();
    list<<query.value(0).toString();
  }
  return list;
}
```

```
struct  dets
{
   QString ln;
   int  summary;
}
```

```
QVector <dets>
~~void~~     show_enroll_sorted()
{
  QVector   <dets>   details;
                                          as  n
  QSqlQuery  q(dB);
  q.exec("Select  lessonname, count(id) from
Lesson, Enroll   where  lid=lessonid  ~~am~~  Group  by
lessonname   order  by   count(id) (n) "); (Desc)
   while(q.next())                               ↓
  {                                          φθivouσα
     dets   d;
     d.ln=q.value(0).toString();
     d.summary=q.value(1).toInt();
     details<<d;
   }
return  details;
}
```