4 SQL语言与可编程对象

目 录

4.1.1 SQL语言概述

SQL语言概述

SQL是结构化查询语言（Structure Query Language）的简称，是关系数据库操作语言的国际标准。

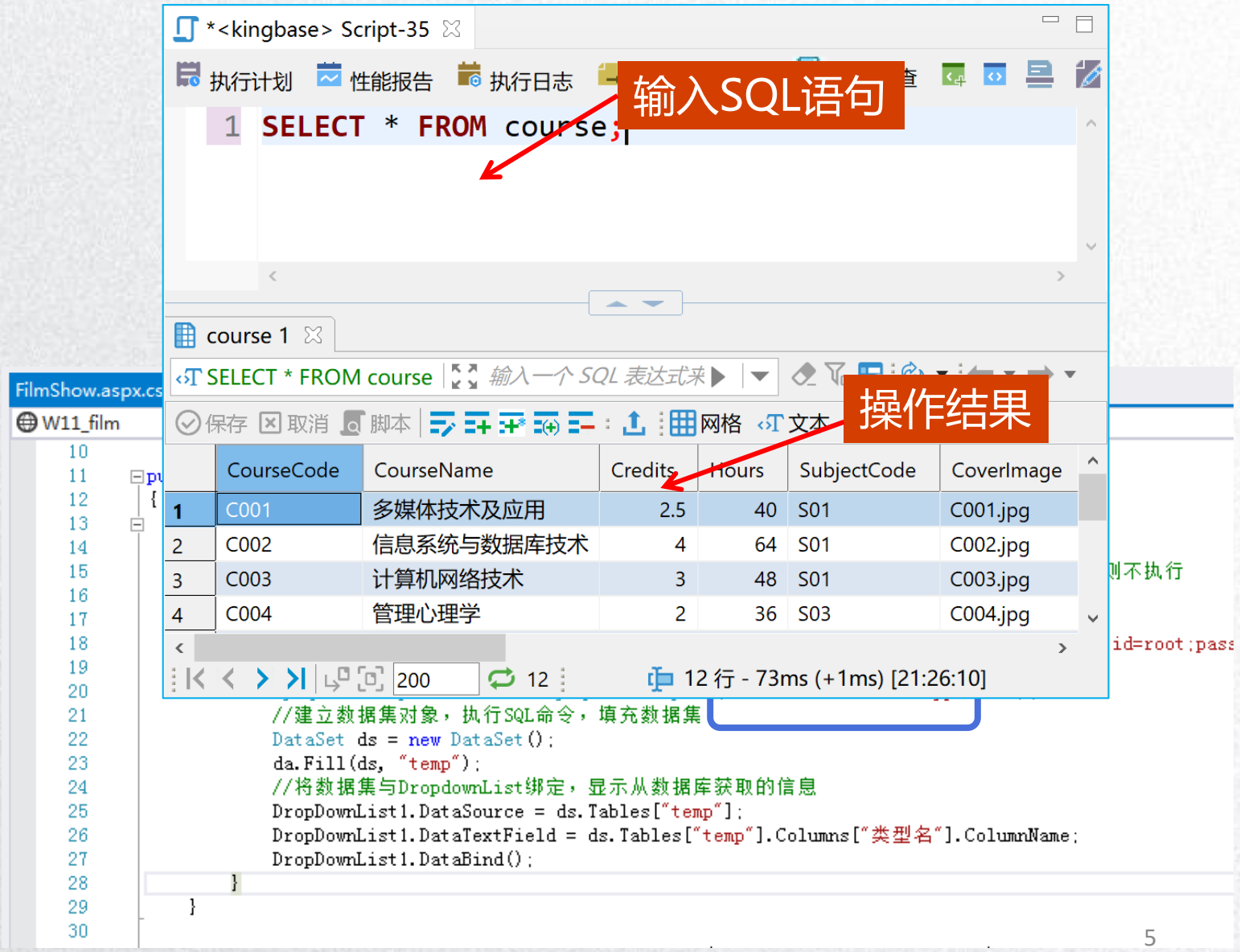
KingbaseES所使用的SQL语言遵从ANSI-92国际标准。

SQL语言的特点

非过程化的描述性语言。
语言简洁、易学易用。

使用方式

- ✓ 联机交互独立使用
- ✓ 嵌入到其他宿主语言

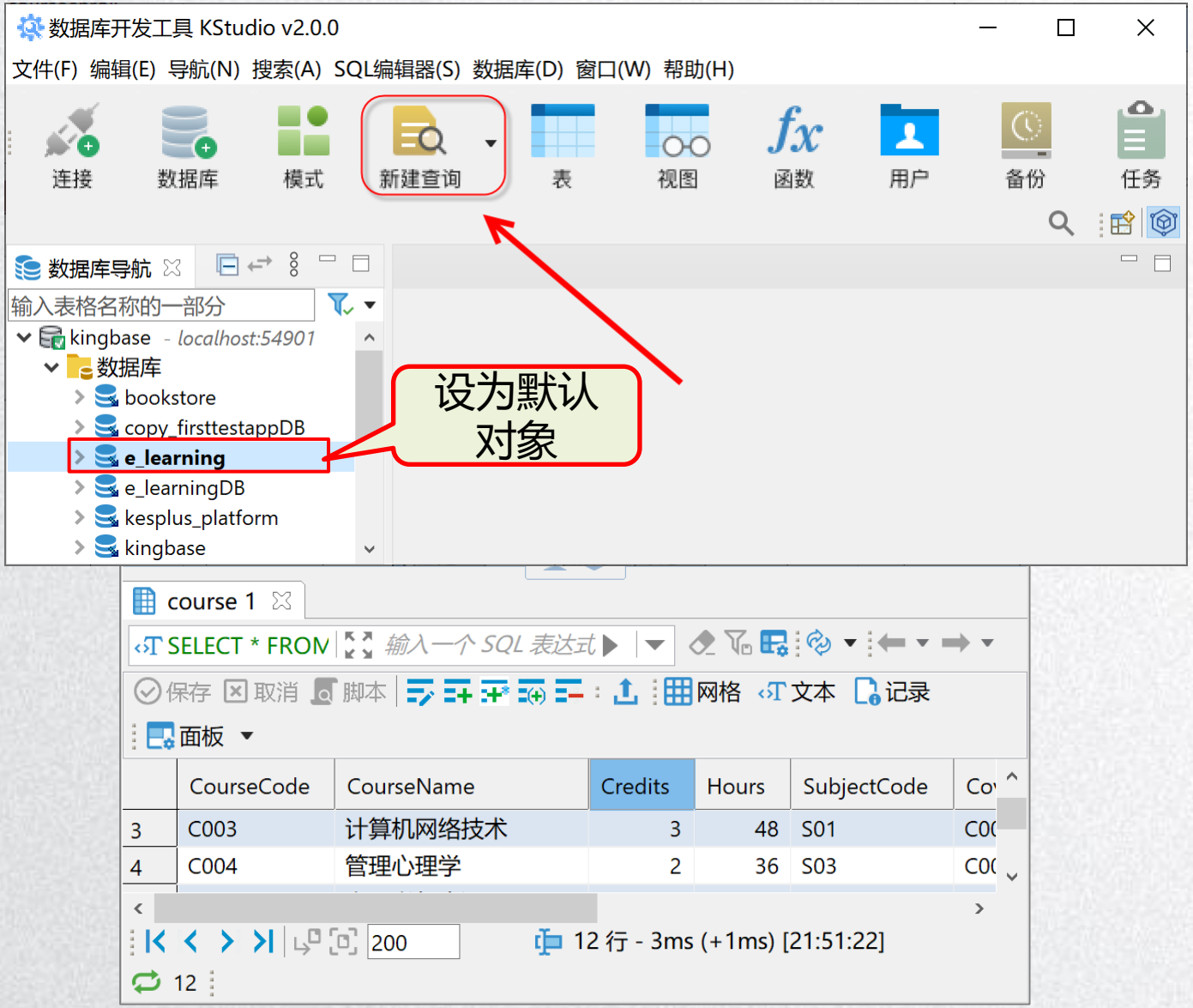


KStudio中打开交互式操作窗口

注意：首先将待操作的数据库设为默认对象。

1. 点击工具栏中的“新建查询”按钮。

2. 在“查询编辑器”中输入命令，点击“运行”。



SQL语言的主要语句类别

操作对象是数据库或者对象（表、视图等）

- ✓ **DDL(Data Definition Language) 数据库定义语言**

  维护数据结构。支持建立、修改、删除数据库及其对象的操作

操作对象是表中的数据

- ✓ **DML(Data Manipulation Language) 数据库操纵语言**

  数据访问。支持对数据库中数据的查询、插入、修改和删除等操作。

- ✓ **DCL(Data Control Language) 数据库控制语言**

  数据库监视。支持对数据对象的授权、安全控制等操作。

SQL语言的书写规则

关键字

【例】 `SELECT StudentCode, StudentName FROM student;` --查询学生表中学生的学号和姓名

标识符

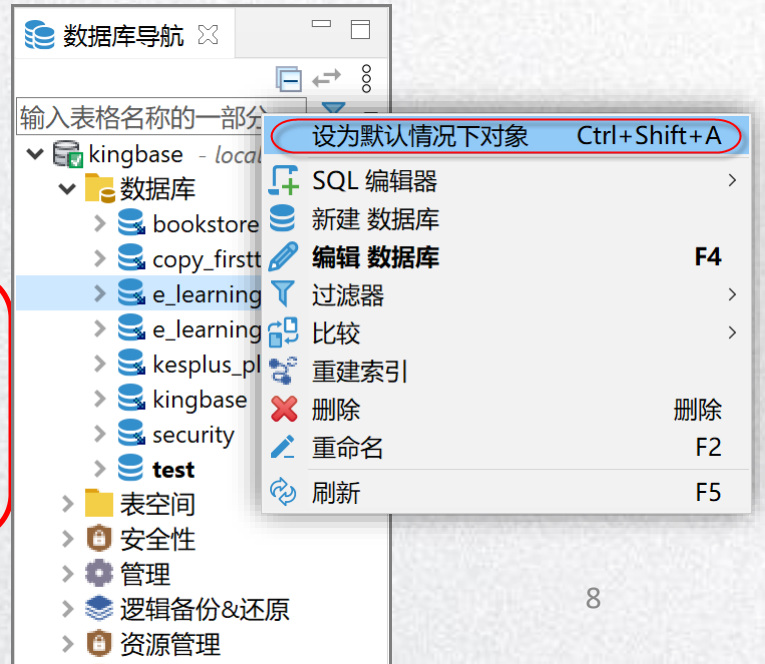
语句以“;”结尾

- ✓ 语句中的单词可以是关键词，也可以是标识符
- ✓ 不区分字母大小写
- ✓ 一条语句可写在一行或多行上
- ✓ 可用注释增强语句可读性
  - ① 单行注释（--）
  - ② 多行注释（/* … */）

注意：

当第一次访问某数据库时，需将该数据库“设为默认情况下对象”！

若自定义标识符与保留字同名，如 limit，用双引号括起，"limit"



THANK YOU!

4.1.2 数据库定义语言（DDL）简介

数据库定义语言DDL

维护数据结构。支持建立、修改或删除数据库及其对象的操作。

主要语句：（动作+操作对象类型+对象名）

创建数据库

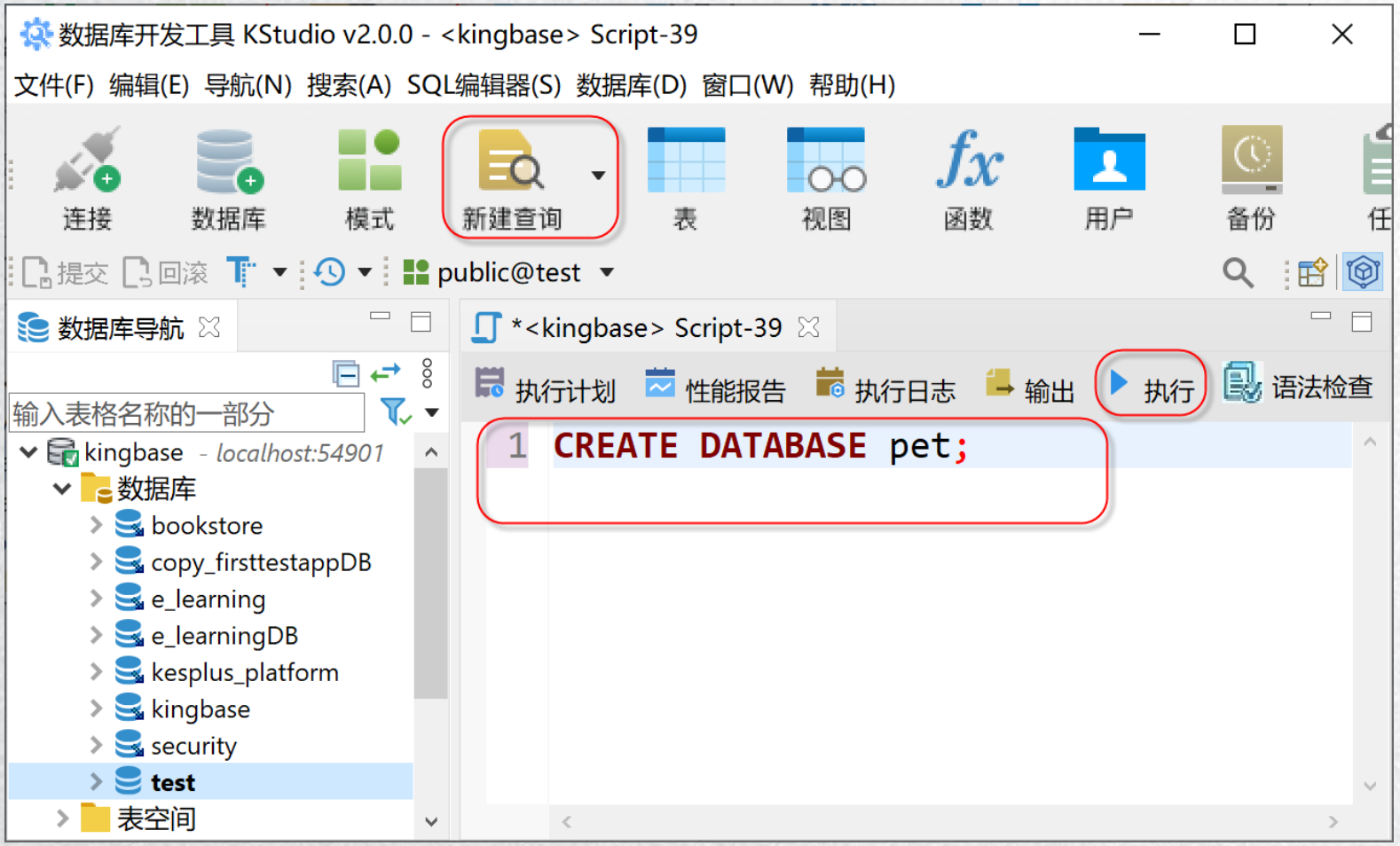
```
CREATE DATABASE 数据库名
 [ [ WITH ]
   [ OWNER [=] 拥有者名 ]
   [ TEMPLATE [=] 模板名 ]
   [ ENCODING [=] 字符集编码 ]
   [ LC_COLLATE [=] 排序规则 ] ];
```

[ ]表示可选项

```
CREATE DATABASE pet;   -- 创建一个名为pet的数据库，采用默认的字符集和排序规则

CREATE DATABASE plant Encoding = 'utf8';
/*创建一个名为plant的数据库，设置其字符集为utf8 */
```

演示——创建数据库命令


数据库开发工具 KStudio v2.0.0 - <kingbase> Script-39
文件(F) 编辑(E) 导航(N) 搜索(A) SQL编辑器(S) 数据库(D) 窗口(W) 帮助(H)
连接
数据库
模式
新建查询
表
视图
函数
用户
备份
提交
回滚
public@test
数据库导航
输入表格名称的一部分
kingbase - localhost:54901
数据库
bookstore
copy_firsttestappDB
e_learning
e_learningDB
kesplus_platform
kingbase
security
test
表空间
*<kingbase> Script-39
执行计划
性能报告
执行日志
输出
执行
语法检查
1 CREATE DATABASE pet;

删除数据库

```
DROP DATABASE 数据库名;   -- 删除数据库
```

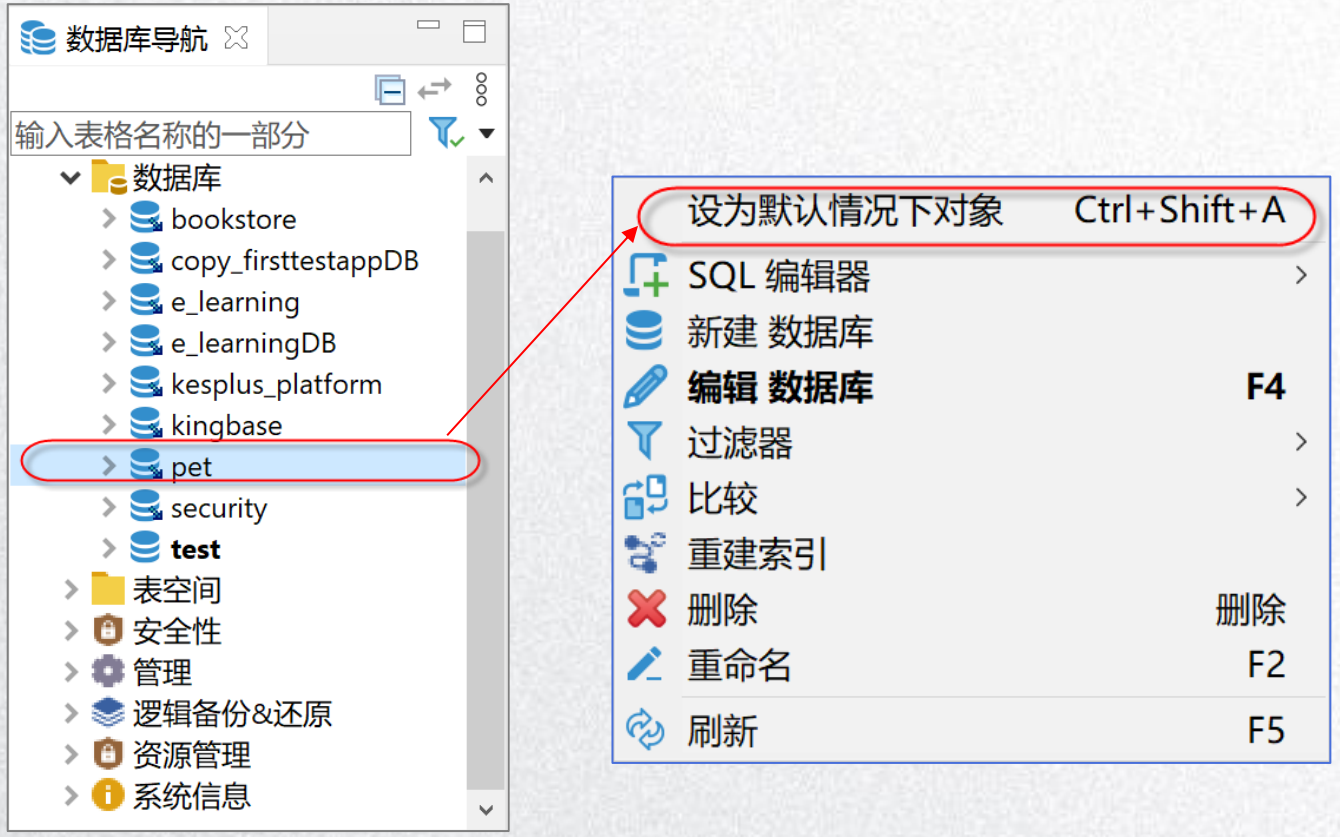
DROP DATABASE plant; -- 删除数据库plant

注意：用DROP语句删除数据库将把数据库文件彻底从磁盘删除。

使用数据库——设其为默认对象

使用某数据库前，**应先将该数据库设为默认对象**，以后的操作都是针对该数据库进行。

如，将pet设为默认对象



创建数据表

```
CREATE TABLE 表名(
字段名1 字段数据类型 [字段约束1],
字段名2 字段数据类型 [字段约束2],
 ……
 [Constraint 表级约束(主键、外键、唯一性等)]
);
```

示例——创建数据表

先将pet设为默认对象

字段名	数据类型	约束
CatCode	int4	非空, 自增，主键
CatName	varchar(16)	非空
Gender	char(1)	非空，默认’母’
Birthday	date	非空
Photo	varchar(30)	
Introduction	text	
RegTime	timestamp	默认 CURRENT_TIMESTAMP

```
CREATE TABLE cat(
  /*不可取空值，创建具有自增功能的列*/
  CatCode int4 NOT NULL GENERATED BY DEFAULT AS IDENTITY,
  CatName varchar(16) NOT NULL, --不可取空值
  Gender char(1) NOT NULL DEFAULT '母', --Gender默认为母
  Birthday date NOT NULL,
  Photo varchar(30),
  Introduction text,
  /*默认当前时间*/
  RegTime timestamp DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (CatCode) --设置CatCode为主键
);
```

描述字段的每一行末尾是逗号

语句末尾是分号

自增列语法

必填

GENERATED BY DEFAULT AS IDENTITY[(START WITH 起始值 INCREMENT BY 增量)]

- **GENERATED BY DEFAULT**
  插入数据时，若没有为该列提供值，数据库会自动生成一个值。但用户可以手动插入值
- **AS IDENTITY**
  说明该列是一个自增列，数据库会自动管理其值的生成。
- **START WITH 起始值**
  设置自增序列的起始值，若省略则默认1。
- **INCREMENT BY 增量**
  定义自增的步长，若省略则默认1。

如果省略括号及其内容，则从1开始，每次递增1。

如，GENERATED BY DEFAULT AS IDENTITY(START WITH 1001 INCREMENT BY 1)

创建索引

CREATE [UNIQUE] INDEX 索引名
ON 表名(字段名 ASC|DESC); -- ASC升序，DESC降序

默认为升序ASC

例：在cat表的CatName字段上建立唯一索引IX_Name，按名字降序排列

```
CREATE UNIQUE INDEX IX_Name
ON cat (CatName DESC);
```

表名

要建立的索引字段名

cat表

字段名	数据类型	约束
CatCode	int4	非空, 自增，主键
CatName	varchar(16)	非空
Gender	char(1)	非空，默认'母'
Birthday	date	非空
Photo	varchar(30)	默认为空
Introduction	text	
RegTime	timestamp	默认 CURRENT_TIMESTAMP

删除数据表、索引

```
DROP TABLE 数据表名;
```

如，**DROP TABLE cat;** **--删除cat表**

```
DROP INDEX 索引名;
```

如，**DROP INDEX IX_Name;** **--删除(cat表的)IX_Name索引**

修改数据库对象

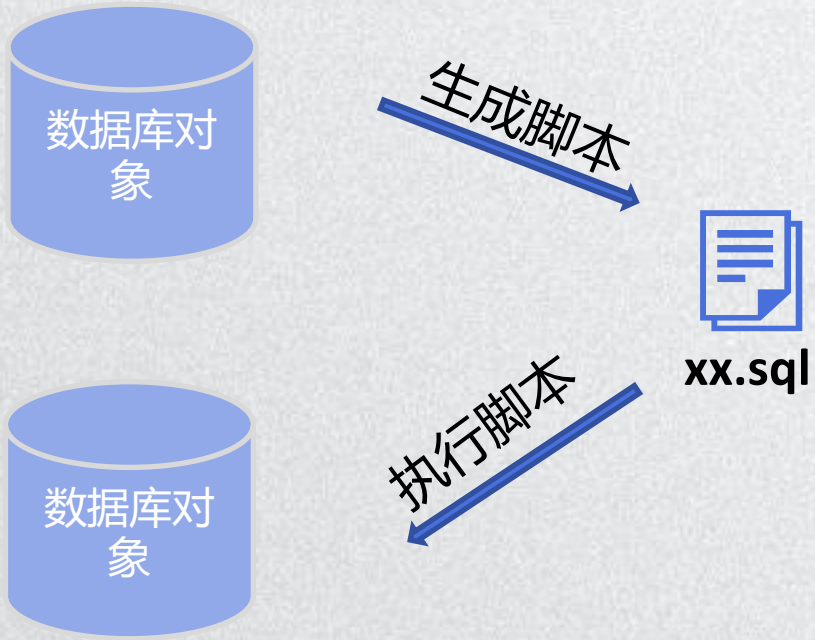
ALTER 数据库对象 对象名;

如　向 **cat** 表中添加一个名为 **Weight** 的列，
数据类型为 **decimal(3, 1)**。

```
ALTER TABLE cat ADD (Weight decimal(3,1));
```

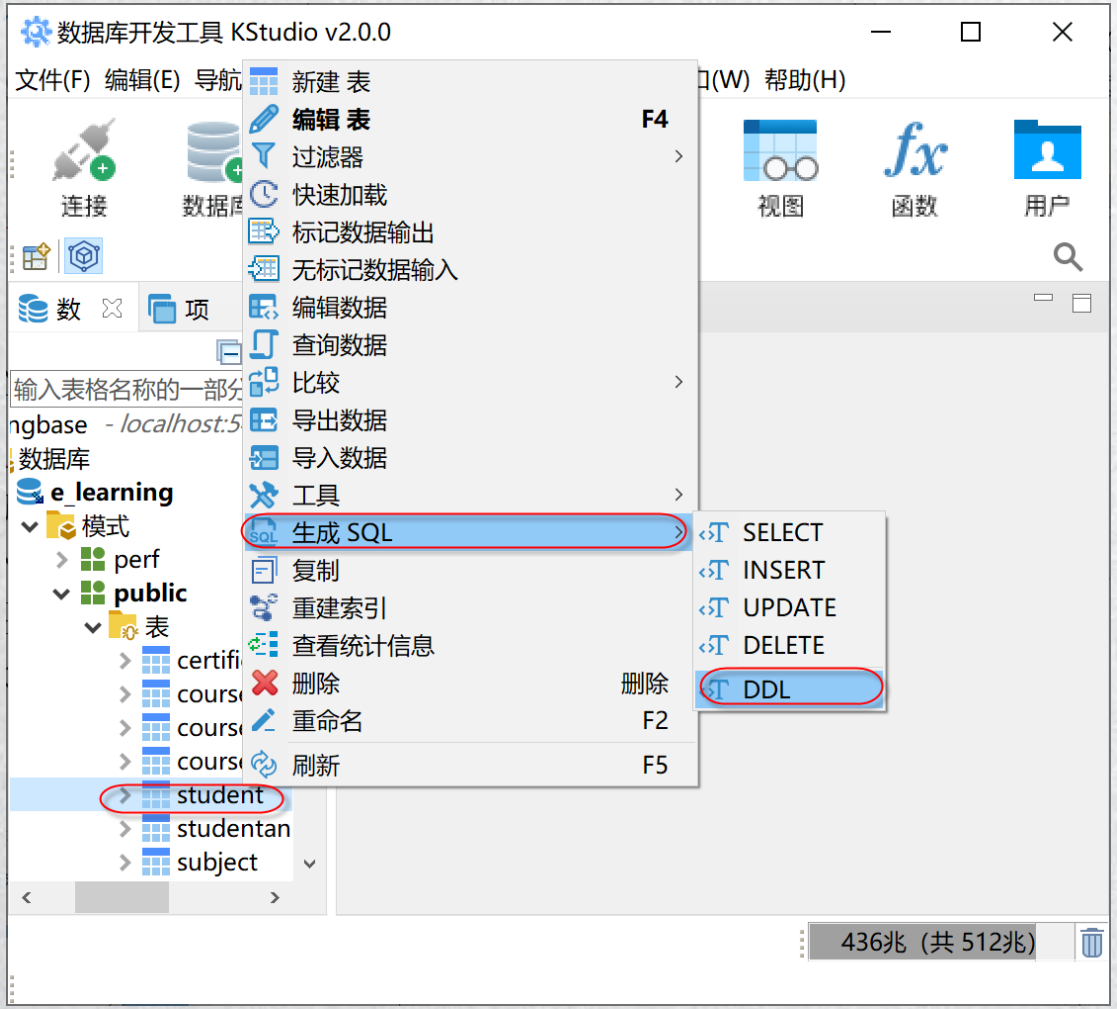
转储数据定义SQL文件

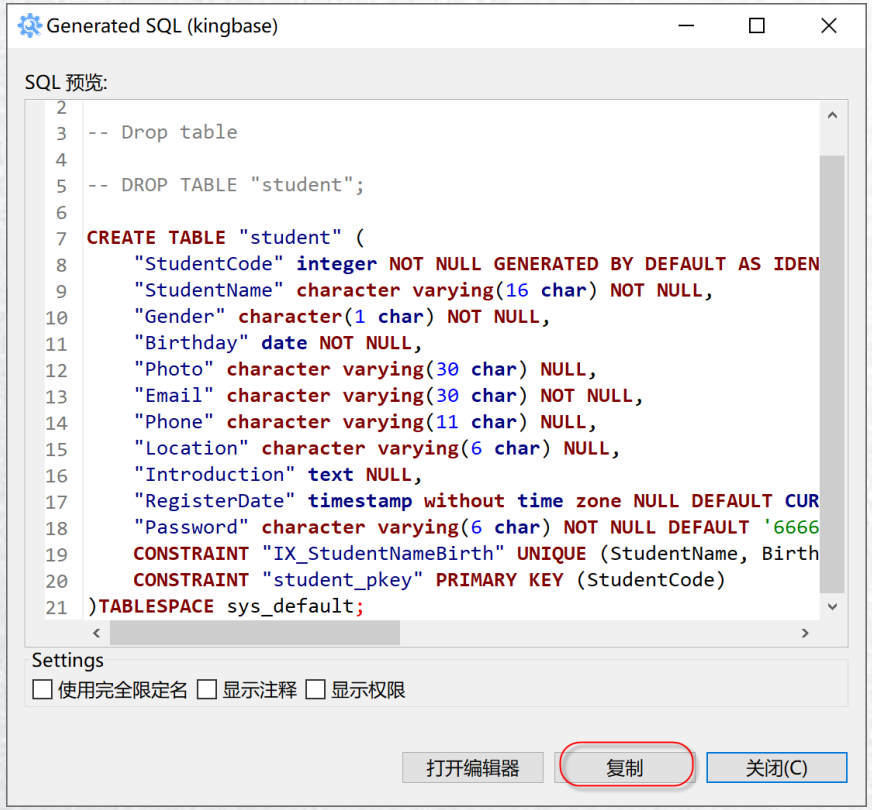
在KStudio中创建各种数据库对象，实质上是以可视化方式生成DDL语句，这些语句可以保存到一个SQL脚本文件中，以后运行该文件可重建数据库对象。



生成创建表的DDL语句

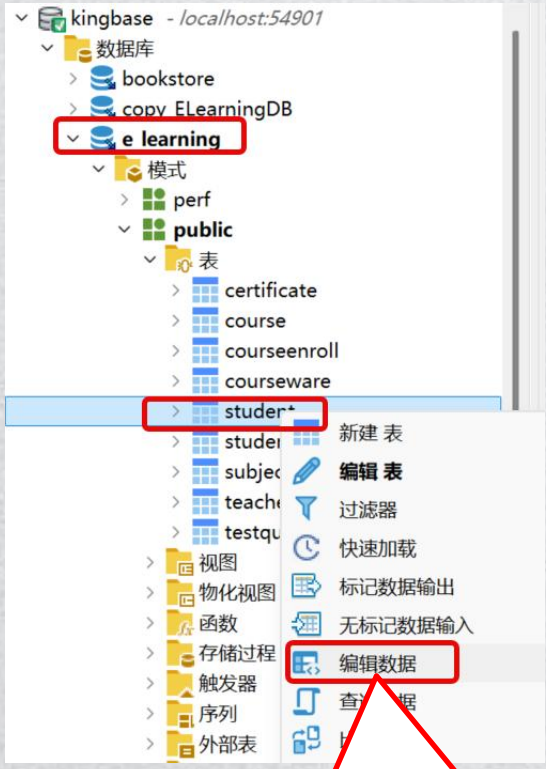
【例】生成创建student表的DDL语句 。



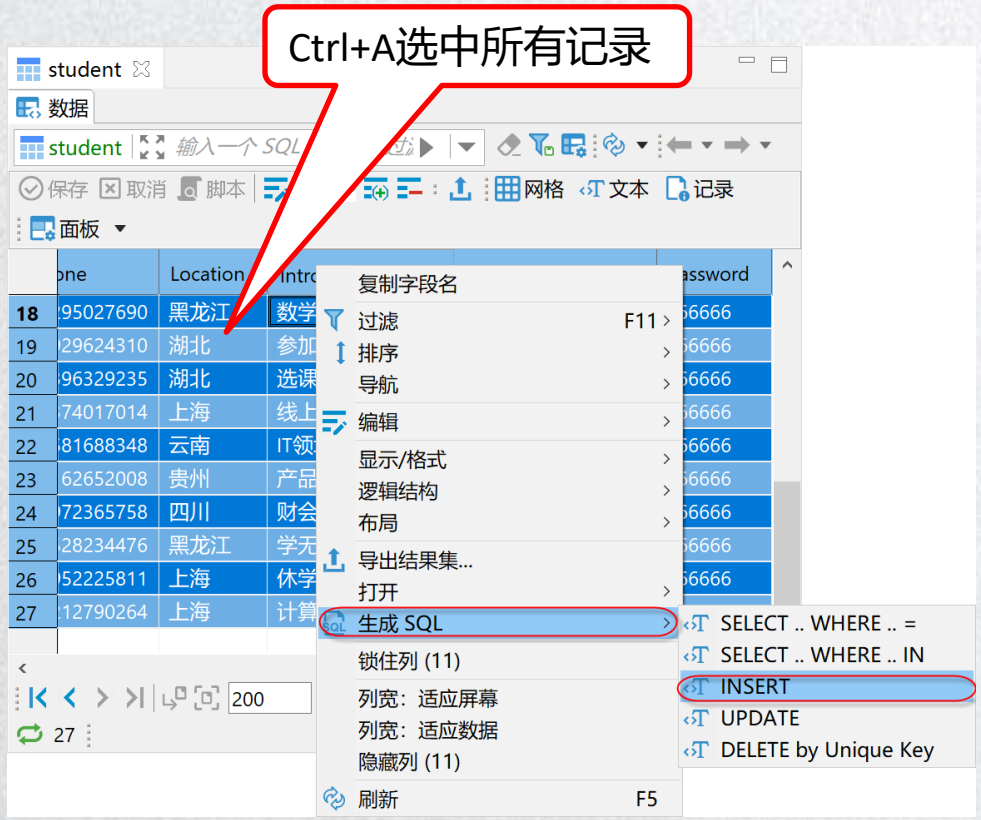


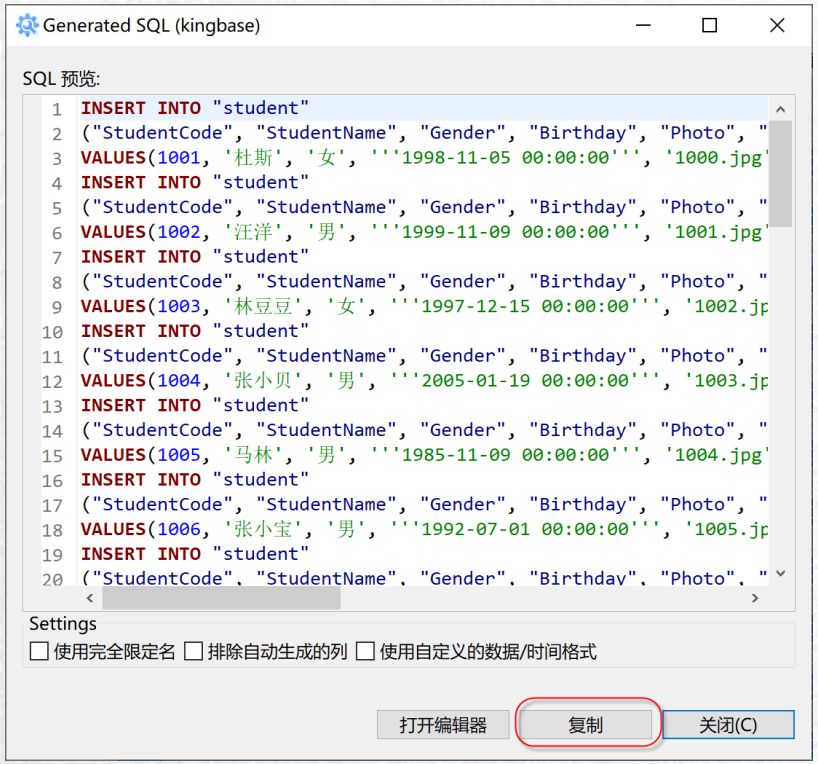
生成表中数据INSERT语句

【例】生成student表中数据INSERT语句 。



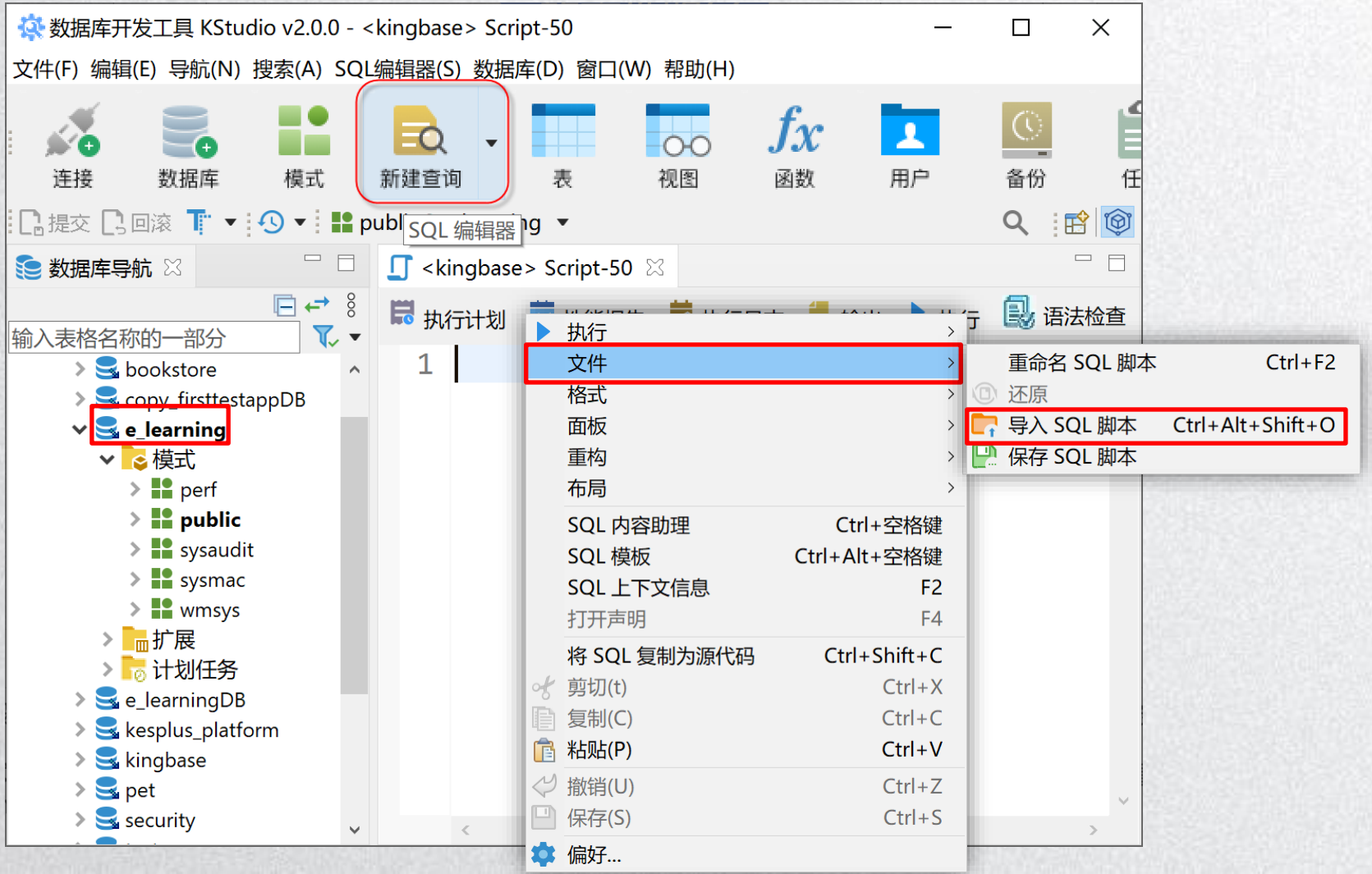
打开student表的数据





运行SQL脚本

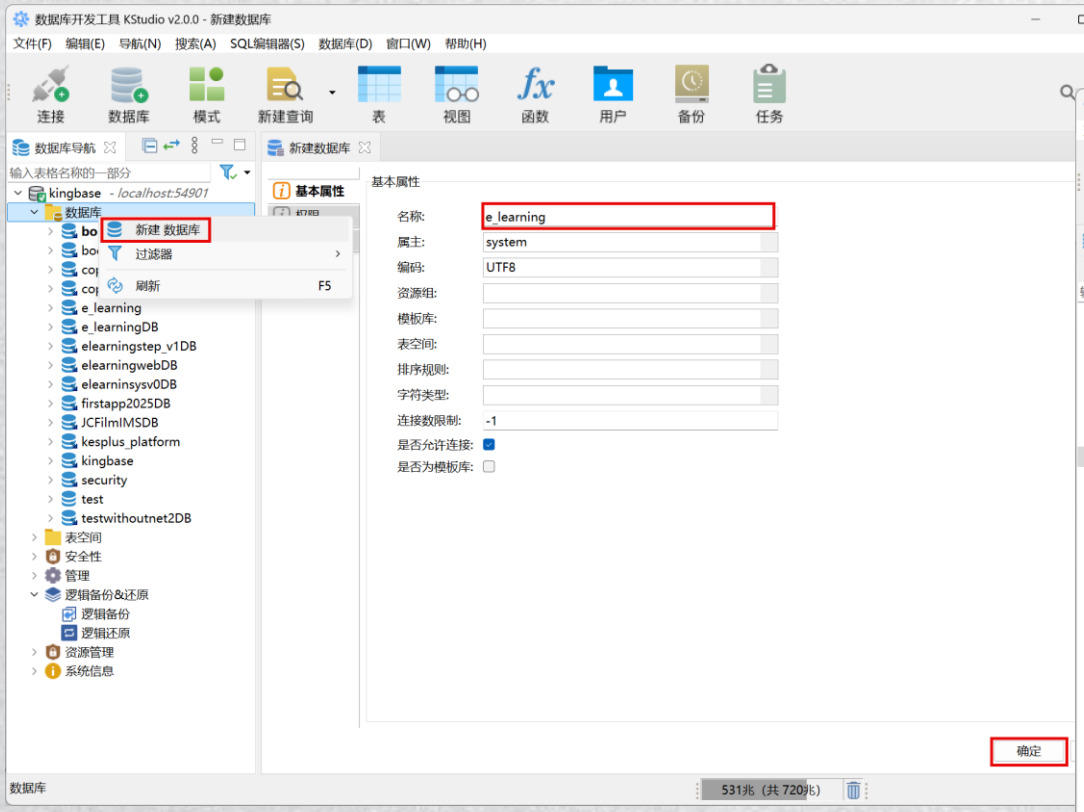
在已建的example数据库中，新建查询，运行文件student.sql建立student表。

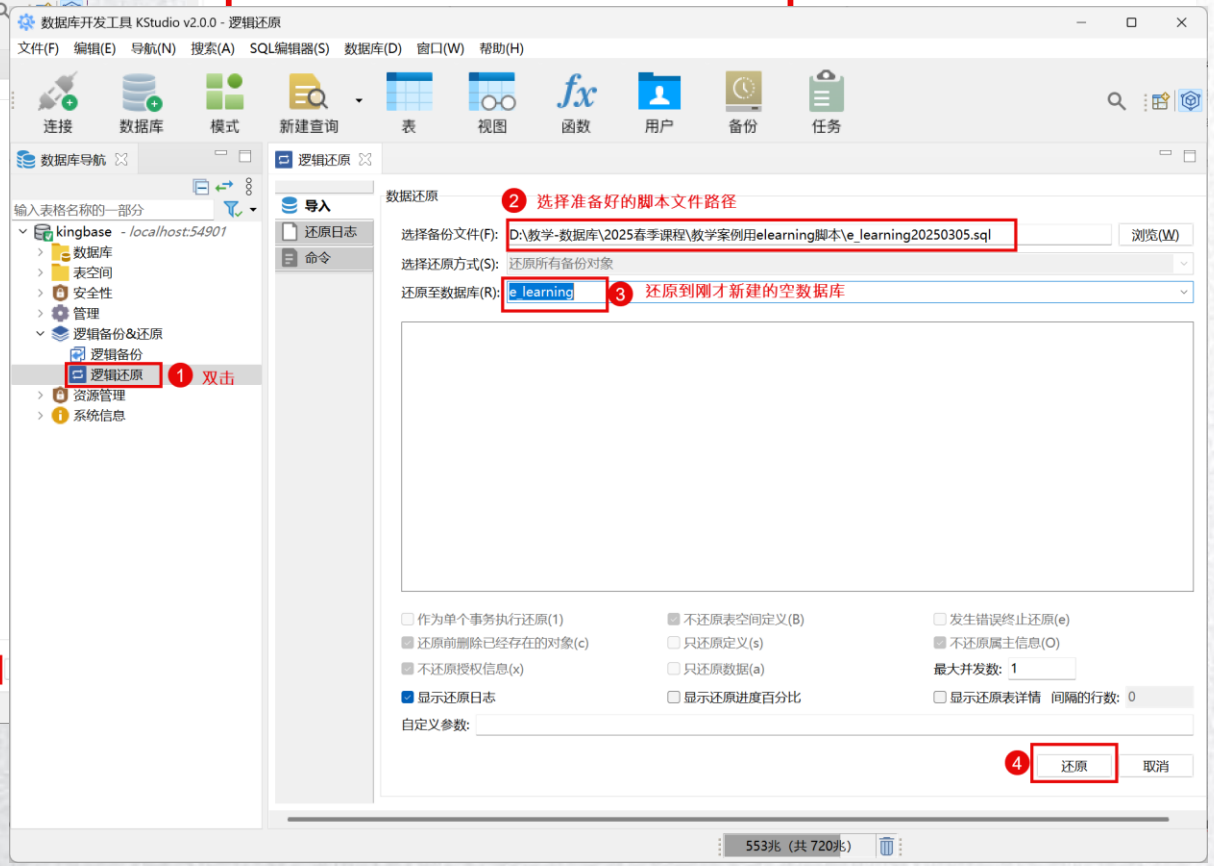


注意：也可以转储整个数据库。如果运行sql文件重建整个数据库，必须在KStudio中先新建一个数据库，再执行SQL语句。

用脚本文件重建教学案例数据库e_learning

- 下载数据库脚本文件(e_learning.sql)。
- 新建一个空数据库命名为e_learning。
- 使用“逻辑还原”功能重建数据库。





回顾：数据库定义语言DDL

维护数据结构。支持建立、修改、删除数据库及其对象的操作。

主要语句：

CREATE	DATABASE
	TABLE
DROP	VIEW
	PROCEDURE
ALTER	
	TRIGGER
	FUNCTION

THANK YOU!

4.1.3 数据库操作语言（DML）——单表查询

数据操纵语言DML

实现数据访问。支持对数据的查询和更新操作。

常用语句（查增改删）

语句	功能
SELECT	从表中查询数据
INSERT	向表中添加记录
UPDATE	修改表中数据
DELETE	从表中删除记录

查询

更新

查

增

改

删

查询语句SELECT

从一个或多个表中查询数据。单表查询只需要查询一个表。

例1 查询表中某些字段的信息

	StudentCode	StudentName	Gender	Birthday	Photo	Email	Phone	Location	Introduction	RegisterDate	Password
1	1,001	杜斯	女	1998-11-05 00:00:00	1000.jpg	1001@elearning.com	13358301911	上海	数学专业毕业在即	2017-07-21 15:40:44	666666
2	1,002	汪洋	男	1999-11-09 00:00:00	1001.jpg	1002@elearning.com	13834992564	北京	对文学感兴趣	2017-07-25 22:30:48	666666
3	1,003	林豆豆	女	1997-12-15 00:00:00	1002.jpg	1003@elearning.com	13243964904	江苏	网络选课拓展知识面	2018-06-04 14:36:36	123456
4	1,004	张小贝	男	2005-01-19 00:00:00	1003.jpg	1004@elearning.com	13142540852	山东	到这里提前上大学	2017-08-28 09:20:58	666666
5	1,005	马林	男	1985-11-09 00:00:00	1004.jpg	1005@elearning.com	13971113836	浙江	已经工作，充电	2017-09-01 20:11:04	666666
6	1,006	张小宝	男	1992-07-01 00:00:00	1005.jpg	1006@sina.cn	13713143513	黑龙江	回来恶补一些专业知	2017-09-17 18:15:07	666666
7	1,007	宋思明	男	1995-10-02 00:00:00	1006.jpg	1007@sina.cn	13916801546	江西	无话可说	2017-09-23 15:55:11	666666
8	1,008	徐米拉	女	2000-03-30 00:00:00	1007.jpg	1008@sina.cn	13225414354	湖北	到这里才发现学海无	2017-09-30 19:51:14	666666

200 27 27 行 - 2ms (+1ms) [00:08:51]

问题：如何得到所有同学的姓名和性别列表?

分析：查询student表中”StudentName”和“Gender”这两个字段的信息。

SELECT...FROM...

SELECT 字段列表 FROM 表名;

字段列表：表中字段
- ✓ 多个字段之间用逗号分隔
- ✓ 所有字段可用通配符“*”表示

SELECT StudentName, Gender FROM Student;

	studentname	gender
1	杜斯	女
2	汪洋	男
3	林豆豆	女
4	张小贝	男
5	马林	男
6	张小宝	男

200 27

Student表中学生姓名和性别

例2 查询表中所有字段的信息

	StudentCode	StudentName	Gender	Birthday	Photo	Email	Phone	Location	Introduction	RegisterDate	Password
1	1,001	杜斯	女	1998-11-05 00:00:00	1000.jpg	1001@elearning.com	13358301911	上海	数学专业毕业在即	2017-07-21 15:40:44	666666
2	1,002	汪洋	男	1999-11-09 00:00:00	1001.jpg	1002@elearning.com	13834992564	北京	对文学感兴趣	2017-07-25 22:30:48	666666
3	1,003	林豆豆	女	1997-12-15 00:00:00	1002.jpg	1003@elearning.com	13243964904	江苏	网络选课拓展知识面	2018-06-04 14:36:36	123456
4	1,004	张小贝	男	2005-01-19 00:00:00	1003.jpg	1004@elearning.com	13142540852	山东	到这里提前上大学	2017-08-28 09:20:58	666666
5	1,005	马林	男	1985-11-09 00:00:00	1004.jpg	1005@elearning.com	13971113836	浙江	已经工作，充电	2017-09-01 20:11:04	666666
6	1,006	张小宝	男	1992-07-01 00:00:00	1005.jpg	1006@sina.cn	13713143513	黑龙江	回来恶补一些专业知	2017-09-17 18:15:07	666666
7	1,007	宋思明	男	1995-10-02 00:00:00	1006.jpg	1007@sina.cn	13916801546	江西	无话可说	2017-09-23 15:55:11	666666
8	1,008	徐米拉	女	2000-03-30 00:00:00	1007.jpg	1008@sina.cn	13225414354	湖北	到这里才发现学海无	2017-09-30 19:51:14	666666

200 27 27 行 - 2ms (+1ms) [00:08:51]

问题：怎么知道每一位同学的所有信息？

分析：查询student表中所有字段的信息

SELECT...FROM...

SELECT 字段列表 FROM 表名;

SELECT * FROM Student;

字段列表：表中字段

- 多个字段之间用逗号分隔
- 所有字段可用通配符“*”表示

	StudentCode	StudentName	Gender	Birthday	Photo	Email	Phone	Location	Introduction	RegisterDate	Password
1	1,001	杜斯	女	1998-11-05 00:00:00	1000.jpg	1001@elearning.com	13358301911	上海	数学专业毕业在即	2017-07-21 15:40:44	666666
2	1,002	汪洋	男	1999-11-09 00:00:00	1001.jpg	1002@elearning.com	13834992564	北京	对文学感兴趣	2017-07-25 22:30:48	666666
3	1,003	林豆豆	女	1997-12-15 00:00:00	1002.jpg	1003@elearning.com	13243964904	江苏	网络选课拓展知识[	2018-06-04 14:36:36	123456
4	1,004	张小贝	男	2005-01-19 00:00:00	1003.jpg	1004@elearning.com	13142540852	山东	到这里提前上大学	2017-08-28 09:20:58	666666
5	1,005	马林	男	1985-11-09 00:00:00	1004.jpg	1005@elearning.com	13971113836	浙江	已经工作，充电	2017-09-01 20:11:04	666666
6	1,006	张小宝	男	1992-07-01 00:00:00	1005.jpg	1006@sina.cn	13713143513	黑龙江	回来恶补一些专业[	2017-09-17 18:15:07	666666
7	1,007	宋思明	男	1995-10-02 00:00:00	1006.jpg	1007@sina.cn	13916801546	江西	无话可说	2017-09-23 15:55:11	666666
8	1,008	徐米拉	女	2000-03-30 00:00:00	1007.jpg	1008@sina.cn	13225414354	湖北	到这里才发现学海[	2017-09-30 19:51:14	666666

200 27 27 行 - 2ms (+1ms) [00:08:51]

Student表中全部信息

例3 给查询结果中的字段名取个别名

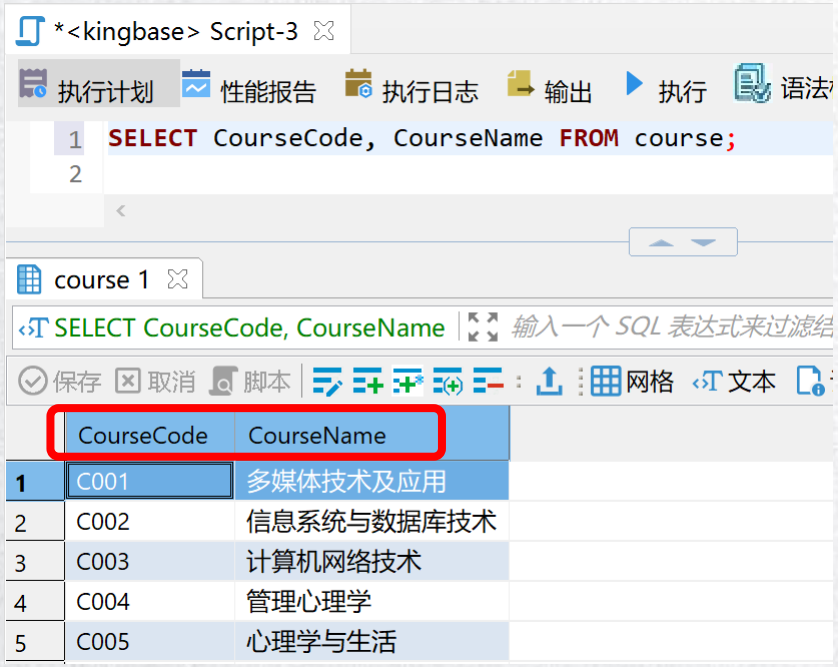
CourseCo	CourseName	Credits	Hours	SubjectCode	CoverImage	Introduction	TeacherCode	StudentNum
C001	多媒体技术及应	2.5	40	S01	C001.jpg	多媒体技术使计算	T001	5
C002	信息系统与数据	4	64	S01	C002.jpg	这门课融合了信息	T002	5
C003	计算机网络技术	3	48	S01	C003.jpg	邀请您畅游网络世	T003	5
C004	管理心理学	2	36	S03	C004.jpg	内心可以萌发最伟	T008	2
C005	心理学与生活	1.5	30	S03	C005.jpg	心理学很神秘？你	T004	4
C006	感觉与知觉	1.5	28	S03	C006.jpg	「感觉与知觉」是	T004	0
C007	沟通心理学	2	32	S03	C007.jpg	视角独特的"主持型	T008	1
C008	操作系统	4	64	S01	C008.jpg	操作系统是计算机	T002	1
C009	高等数学微积分	4	64	S06	C009.jpg	本课程系统地介绍	T007	2
C010	线性代数	3	48	S06	C010.jpg	线性代数是19世纪	T005	0
C011	外国文学史	2	36	S07	C011.jpg	外国文学横跨东西	T006	0
C012	唐诗经典	1.5	28	S07	C012.jpg	唐诗是中国古典诗	T006	1

问题：

查询课程号和课程名信息，查询结果中的字段名分别是“课程号”和“课程名”？

分析：

1. 查询course表中”CourseCode”和“CourseName”这两个字段的信息
2. 给查询结果的字段名取个别名



SELECT…FROM…

SELECT 字段列表 FROM 表名;

字段列表：表中字段
✓ “字段名 AS 别名”替换列标题显示

SELECT CourseCode AS课程号, CourseName 课程名 FROM course;

	课程号	课程名
1	C001	多媒体技术及应用
2	C002	信息系统与数据库技术
3	C003	计算机网络技术

200 12

用别名显示查询结果和列标题

例4 查询表中可能有重复值的字段

CourseCo	CourseName	Credits	Hours	SubjectCode	CoverImage	Introduction	TeacherCode	StudentNum
C001	多媒体技术及应	2.5	40	S01	C001.jpg	多媒体技术使计算	T001	5
C002	信息系统与数据	4	64	S01	C002.jpg	这门课融合了信息	T002	5
C003	计算机网络技术	3	48	S01	C003.jpg	邀请您畅游网络世	T003	5
C004	管理心理学	2	36	S03	C004.jpg	内心可以萌发最伟	T008	2
C005	心理学与生活	1.5	30	S03	C005.jpg	心理学很神秘？你	T004	4
C006	感觉与知觉	1.5	28	S03	C006.jpg	「感觉与知觉」是	T004	0
C007	沟通心理学	2	32	S03	C007.jpg	视角独特的"主持型	T008	1
C008	操作系统	4	64	S01	C008.jpg	操作系统是计算机	T002	1
C009	高等数学微积分	4	64	S06	C009.jpg	本课程系统地介绍	T007	2
C010	线性代数	3	48	S06	C010.jpg	线性代数是19世纪	T005	0
C011	外国文学史	2	36	S07	C011.jpg	外国文学横跨东西	T006	0
C012	唐诗经典	1.5	28	S07	C012.jpg	唐诗是中国古典诗	T006	1

问题：根据course表信息，怎么知道有多少种不同的学科？

分析：查询course表中的SubjectCode字段？

SELECT...FROM...

SELECT 字段列表 FROM 表名;

字段列表：
字段名前加限制，优化查询结果：
- DISTINCT：相同字段值只显示一条记录

SELECT SubjectCode FROM Course;

	SubjectCode
1	S01
2	S01
3	S01
4	S03
5	S03
6	S03

查询Course表中学科号

SELECT DISTINCT SubjectCode FROM Course;

	SubjectCode
1	S01
2	S03
3	S06
4	S07

查询Course表中学科号（只显示一条）

例5 查询时针对字段值的算术操作

StudentCode	CourseCode	Score	TestTime
1,001	C005	NULL]	[NULL]
1,002	C001	70	-07-21 16:21:40
1,002	C002	88	-06-22 09:15:57
1,002	C003	55	-08-15 09:20:33
1,002	C004	NULL]	[NULL]
1,002	C005	NULL]	[NULL]
1,003	C001	75	-11-01 15:55:01
1,003	C002	90	-01-10 10:55:18
1,003	C003	76	-06-22 09:16:20
1,003	C005	NULL]	[NULL]
1,003	C012	NULL]	[NULL]
1,004	C001	99	-06-22 09:16:29
1,004	C002	88	-06-22 09:16:42
1,004	C003	82	-06-22 09:17:06
1,005	C001	54	-06-22 09:17:25
1,005	C002	60	-06-22 09:17:33

问题：
已知数据库中courseenroll表中所有同学成绩偏低5分，根据该表信息如何查询得到正常成绩（不更改表的数据）？

分析：
查询courseenroll表中的Score字段，然后怎么调整呢?

SELECT...FROM...

SELECT 字段列表 FROM 表名;

字段列表：
- 可新增字段(常量、表达式)

SELECT StudentCode, CourseCode, Score+5 AS 成绩, '成绩调整' AS 说明
FROM courseenroll;

	StudentCode	CourseCode	成绩	说明
1	1,001	C001	77	成绩调整
2	1,001	C002	70	成绩调整
3	1,001	C003	98	成绩调整

查询courseenroll表，将成绩加5分显示，并增加一列成绩调整

注：courseenroll表中数据未更改。

例6 查询时针对字段值的聚合运算

StudentCode	CourseCode	Score	TestTime
1,001	C005	NULL]	[NULL]
1,002	C001	70	-07-21 16:21:40
1,002	C002	88	-06-22 09:15:57
1,002	C003	55	-08-15 09:20:33
1,002	C004	NULL]	[NULL]
1,002	C005	NULL]	[NULL]
1,003	C001	75	-11-01 15:55:01
1,003	C002	90	-01-10 10:55:18
1,003	C003	76	-06-22 09:16:20
1,003	C005	NULL]	[NULL]
1,003	C012	NULL]	[NULL]
1,004	C001	99	-06-22 09:16:29
1,004	C002	88	-06-22 09:16:42
1,004	C003	82	-06-22 09:17:06
1,005	C001	54	-06-22 09:17:25
1,005	C002	60	-06-22 09:17:33

问题：
如何知道课程的平均分/最高分/最低分/总分/总人数?

分析：
查询courseenroll表，对相应字段做聚合操作（求平均值/最大值/最小值/总数/计数）

SELECT语句...FROM...

SELECT 字段列表 FROM 表名;

字段列表：
- 可新增字段(常量、表达式)

```
SELECT Avg(Score) AS 平均分,
Max (Score) AS 最高分,
Min(Score) AS 最低分,
Count(StudentCode) AS 总人次
FROM courseenroll;
```

	平均分	最高分	最低分	总人次
1	76.6666666667	99	54	25

含有聚合函数的查询结果

表达式中常用聚合函数

函数名	函数功能
AVG	计算某一字段的平均值
COUNT	统计某一字段的个数
MAX	查找某一字段的最大值
MIN	查找某一字段的最小值
SUM	计算某一字段的总和

注：courseenroll表中数据未更改。

THANK YOU!

数据查询语句SELECT

——子句

未解决的其他查询问题

问题举例：

1. 怎么从student表中获取所有男同学的信息？
2. 如何按照生源地，统计各地学生的平均成绩？
3. 查询学生的成绩，并按年龄从小到大排序？

…

分析：

1. 查询条件
2. 分组
3. 排序

查询语句SELECT

从一个或多个表中查询数据。

查询语句SELECT——WHERE子句

SELECT 字段列表 FROM 表名
WHERE 查询条件;

查询条件：
是一个关系或逻辑表达式

常用的关系和逻辑运算

查询条件	谓词
比较	>、>=、<、<=、=、<>(不等于)、^= (不等于)、!=(不等于)
确定范围	BETWEEN ...AND、NOT BETWEEN... AND
确定集合	IN、NOT IN、EXISTS
字符匹配	LIKE、NOT LIKE、SIMILAR TO
空值	IS NULL、IS NOT NULL
逻辑运算	AND、OR、NOT

WHERE子句的相等查询

SELECT 字段列表 FROM 表名
WHERE 查询条件;

查询条件：比较运算
>、>=、<、<=、=、
<>(不等于)、!=(不等于)、^=(不等于)

例如：查询student表中学号为1003的学生信息。

```
SELECT * FROM student WHERE StudentCode=1003;
```

例如：查询Student表中所有女生的信息。

```
SELECT StudentCode, StudentName, Gender, Location
FROM student
WHERE  Gender='女';
```

StudentCode	StudentName	Gender	Location
1001	杜斯	女	上海
1003	林豆豆	女	江苏
1008	徐米拉	女	湖北

查询时间: 0.001s　第 1 条记录 (共 14 条)

女生信息

例如：查询生源地为上海的女生的信息。

```
SELECT StudentCode, StudentName, Gender, Location
FROM  Student
WHERE  Gender='女' AND  Location='上海';
```

AND表示两个条件都必须满足，是“且”的意思

StudentCode	StudentName	Gender	Location
1001	杜斯	女	上海
1021	刘萍	女	上海
1027	王红琳	女	上海

查询时间: 0.002s　第 1 条记录 (共 3 条)

生源地为上海的女生信息

WHERE子句的区间查询

SELECT 字段列表 FROM 表名
WHERE 查询条件;

查询条件：确定范围
BETWEEN 值1 AND 值2
在值1和值2之间

表名

查询条件，跟在where字句后

例如：在courseenroll表中查询选修了课程号C001、成绩在70～90分之间的所有学生的学号、课程号及成绩信息。

```
SELECT StudentCode, CourseCode, Score FROM courseenroll
WHERE CourseCode= 'C001' AND Score>= 70 AND Score<=90;
```

查询字段（列表），写在select主句中

等价于

Score BETWEEN 70 AND 90

	StudentCode	CourseCode	Score
1	1,001	C001	72
2	1,002	C001	70
3	1,003	C001	75

成绩在70～90分之间

WHERE子句的区间查询

SELECT 字段列表 FROM 表名
WHERE 查询条件;

查询条件：确定范围
[NOT] BETWEEN 值1 AND 值2
[不]在值1和值2之间

例如：查询Student表中不在1970—2004出生的学生学号、姓名和生日。

```
SELECT StudentCode, StudentName, Birthday FROM Student
WHERE Birthday NOT BETWEEN '1970-01-01' AND '2004-12-31';
```

使用Year()函数

```
Year(Birthday) NOT BETWEEN 1970 AND 2004
```

StudentCode	StudentName	Birthday
1004	张小贝	2005-01-19
1015	刘文强	1963-12-04
1022	汪洋	1968-05-01

不在1970—2004年之间出生

WHERE子句的集合查询

SELECT 字段列表 FROM 表名
WHERE 查询条件;

查询条件：IN运算
确定是否在集合中

例如：查询student表中来自东北三省的学生，即生源地为黑龙江、吉林、辽宁的学生信息。

SELECT StudentCode, StudentName, Location FROM Student
WHERE Location IN ('黑龙江','吉林','辽宁');

等价于关系运算

Location ='黑龙江' OR Location ='吉林' OR Location ='辽宁'

StudentCode	StudentName	Location
1006	张小宝	黑龙江
1014	徐纯纯	辽宁
1018	潘佳栋	黑龙江
1025	宋婷婷	黑龙江

来自东三省

WHERE子句的模糊查询

SELECT 字段列表 FROM 表名
WHERE 查询条件;

常见通配符

通配符	说明
%	匹配0个或多个任意字符
_	匹配1个任意字符
[ ]	匹配集合中任意单个字符
[^]	不匹配集合中的任意单个字符

LIKE+含通配符的查询条件
确定字符串模式是否匹配，仅支持基础通配符%和_

例如：姓名匹配某个模式
- 姓“王”： LIKE '王% '
- 姓“王”单字名： LIKE '王_ '
- 姓“王”双字名： LIKE '王__ '
- 中间字是“小”： LIKE ' _小% '
- 不姓“王”： NOT LIKE '王%'

SIMILAR TO+含通配符的查询条件
确定字符串模式是否匹配，支持上述所有通配符

例如：匹配某个模式
- 不姓“王”： SIMILAR TO '[^王]%'
- 姓“王”或“刘”： SIMILAR TO '[王刘]%'
- 不姓“王”也不姓“刘”： SIMILAR TO '[^王刘]%'
- 手机号尾数是8、9、0：SIMILAR TO '%[890] '

WHERE子句的模糊查询举例

SELECT 字段列表 FROM 表名
WHERE 查询条件;

查询条件：LIKE运算
确定字符串模式是否匹配

例如：查询course表中课程名中包含“技术”两个字的课程号及课程名。

```
SELECT CourseCode, CourseName FROM course
WHERE CourseName LIKE '%技术%';
```

	CourseCode	CourseName
1	C001	多媒体技术及应用
2	C002	信息系统与数据库技术
3	C003	计算机网络技术

名称包含“技术”的课程

例如：查询student表中不姓刘的学生的学号和姓名。

```
SELECT StudentCode, StudentName FROM student
WHERE StudentName NOT LIKE '刘%';
```

```
SELECT StudentCode, StudentName FROM student
WHERE StudentName SIMILAR TO '[^刘]%';
```

	StudentCode	StudentName
1	1,001	杜斯
2	1,002	汪洋
3	1,003	林豆豆

23 行 - 2ms [17:36:4

不姓刘的学生

WHERE子句的空值查询

SELECT 字段列表 FROM 表名
WHERE 查询条件;

查询条件：IS NULL
确定字段取值是否为空

例如：查询courseenroll表中成绩为空的学号和课号信息。

SELECT StudentCode, CourseCode FROM courseenroll WHERE Score ISNULL;

某个字段=null?

StudentCode	CourseCode	Score	TestTime
1,001	C005	NULL]	[NULL]
1,002	C001	70	-07-21 16:21:40
1,002	C002	88	-06-22 09:15:57
1,002	C003	55	-08-15 09:20:33
1,002	C004	NULL]	[NULL]
1,002	C005	NULL]	[NULL]
1,003	C001	75	-11-01 15:55:01
1,003	C002	90	-01-10 10:55:18
1,003	C003	76	-06-22 09:16:20
1,003	C005	NULL]	[NULL]
1,003	C012	NULL]	[NULL]

Courseenroll表的部分内容

	StudentCode	CourseCode
1	1,001	C005
2	1,002	C004
3	1,002	C005
4	1,003	C005
5	1,003	C012
6	1,005	C004
7	1,005	C005

查询结果

查询语句SELECT——GROUP BY子句

分类汇总。即按分组字段把具有相同值的记录汇总计算合并成一条记录。

包含分组字段、汇总计算字段

SELECT 字段列表 FROM 表名
GROUP BY 分组字段 [HAVING 分组条件];

用于分组的字段

注意：
- HAVING子句只能跟在GROUP BY子句后，不能单独使用。
- SELECT后查询的字段列表只能包含分组字段和聚合计算字段。

表名　分组字段　字段列表　分组条件

例如：统计选课表中每门课程的选课人数和平均分。只显示平均80分以上的课程。

```
SELECT CourseCode, Count(StudentCode) AS 选课人数, AVG(Score) AS 平均分
FROM courseenroll
GROUP BY CourseCode
HAVING  AVG(Score) >=80;     -- 或HAVING 平均分>=80;
```

	CourseCode	选课人数	平均分
1	C008	1	70
2	C001	5	74
3	C004	2	[NULL]
4	C005	4	[NULL]
5	C007	1	80
6	C003	5	76.6
7	C012	1	[NULL]
8	C009	1	86
9	C002	5	78.2

HAVING子句与WHERE的区别

SELECT 字段列表 FROM 表名
GROUP BY 分组字段 [HAVING 分组条件];

区别：
WHERE子句：先筛选再汇总
HAVING子句：先汇总再筛选

注意：如果筛选条件中包含汇总结果，用WHERE子句无法实现（WHERE子句中不能包含聚合函数），这时只能用HAVING子句.
例如： HAVING COUNT(StudentCode)>3

例如：查询课程号为C001的选课人数。

```
SELECT CourseCode, COUNT(StudentCode) AS 选课人数, AVG(Score) AS 平均分
FROM courseenroll
WHERE CourseCode='C001'
GROUP BY courseCode;
```

先筛选课程号，再汇总被选出的课程的选课人数

```
SELECT CourseCode, COUNT(StudentCode) AS 选课人数, AVG(Score) AS 平均分
FROM courseenroll
GROUP BY CourseCode
HAVING CourseCode='C001';
```

先汇总每门课程的选课人数，再根据课程号筛选

判断对错

```
SELECT COUNT(StudentCode)  AS 选课人数, AVG(Score) AS 平均分
FROM courseenroll
WHERE CourseCode='C001' ;
```

```
SELECT CourseCode, COUNT(StudentCode)  AS 选课人数, AVG(Score) AS 平均分
FROM courseenroll
WHERE CourseCode='C001' ;
```

为什么错误！

SQL 错误 [42803]: ERROR: 字段 "courseenroll.CourseCode" 必须出现在 GROUP BY 子句中或者在聚合函数中使用
Position: 8 At Line: 1, Line Position: 8

```
SELECT CourseCode, COUNT(StudentCode)  AS 选课人数, AVG(Score) AS 平均分
FROM courseenroll
WHERE CourseCode='C001'
GROUP BY courseCode;
```

GROUP BY子句中有多个分组字段

SELECT 字段列表 FROM 表名
GROUP BY 分组字段 [HAVING 分组条件];

GROUP BY后有多个分组字段，则表示多次分组。

分组字段　　查询的字段列表

例如：统计*各生源地男、女学生的人数*。

```
SELECT Location, Gender, COUNT(StudentCode) AS 学生人数
FROM Student
GROUP BY Location, Gender;
```

Location	Gender	学生人数
上海	女	3
上海	男	2
云南	男	1
北京	女	1
北京	男	2

第 1 条记录 (共 20 条)

使用GROUP BY子句的注意事项

```
SELECT  字段列表  FROM  表名
GROUP BY 分组字段 [ HAVING 分组条件 ];
```

注意：

✓SELECT 子句的字段列表中的字段，或包含在**聚合函数**中， 或包含在 GROUP BY子句中（**分组字段**），否则不能显示该字段值。

✓HAVING子句必须和GROUP BY子句一起使用，不能单独使用。

✓HAVING子句和WHERE子句的实现机制的差异

查询语句SELECT——ORDER BY子句

按特定字段值为查询结果排序。

SELECT 字段列表 FROM 表名
ORDER BY 字段 [ASC|DESC];

排序依据

ASC: Ascending，升序，默认值
DESC:Descending，降序

例如：查询“零零后”学生，并按年龄从小到大排序。

```
SELECT StudentCode, StudentName, Birthday
FROM student
WHERE Birthday>='2000-01-01'
ORDER BY Birthday DESC;
```

StudentCode	StudentName	Birthday
1004	张小贝	2005-01-19
1021	刘萍	2004-03-21
1020	朱丽娜	2003-12-09
1026	朱松	2002-08-02
1009	杨康	2001-11-09

询时间: 0.001s 第 1 条记录 (共 9 条)

按年龄从小到大排序的零零后学生

ORDER BY子句的多个排序字段

SELECT 字段列表 FROM 表名
ORDER BY 字段 [ASC|DESC];

多个排序字段。首先按第一字段排，值相同再依据第二字段排序……

例如：按生源地升序和学号降序查询学生电话信息。（如果生源地相同则按学生学号排序）

```
SELECT Location, StudentCode, StudentName, Phone
FROM student
ORDER BY Location, StudentCode DESC;
```

Location	StudentCode	StudentName	Phone
北京	1017	章咪咪	13347146490
北京	1010	郭靖	13974339695
北京	1002	汪洋	13834992564
贵州	1023	周一桐	13162652008
河南	1016	刘超	13550705371

查询时间: 0.001s　　第 1 条记录 (共 27 条)

查询语句SELECT——LIMIT子句

用于显示查询结果中的部分记录。

```
SELECT  字段列表  FROM  表名
LIMIT 记录数  OFFSET 开始位置;
```

若“记录数”大于查询结果记录总数，则显示所有记录。

若“起始位置”为0，可省略OFFSET子句

例如：查询显示平均成绩前3名的学生。

```
SELECT  StudentCode, avg(Score) 平均成绩
FROM courseenroll
GROUP BY StudentCode
ORDER BY avg(Score) DESC
LIMIT 3;  -- 或 LIMIT 3 OFFSET 0;
```

查询需求中隐含了排序需求

	StudentCode	平均成绩
1	1,004	89.6666666667
2	1,003	80.3333333333
3	1,010	78.6666666667

3 行 - 2ms [22:4

平均成绩前3名

思考：如何查询显示平均成绩第4至6名的学生？

回顾：查询语句SELECT

从一个或多个表中查询数据。

THANK YOU!