

## 数据库创建与维护

数据库管理系统是创建和使用数据库的必备软件，它提供集中进行数据组织、存储、维护和检索的功能。本章介绍国产数据库管理系统 KingbaseES，重点讲解建立和维护数据库的方法。

### 3.1 KingbaseES 及数据库设计案例



#### 3.1.1 KingbaseES 简介

金仓数据库管理系统（KingbaseES，也可简称为 KES）是中电科金仓（北京）科技股份有限公司研发的具有自主知识产权的企业级大型通用关系数据库管理系统，广泛用于电子政务、能源、金融、电信等重点行业和关键领域。它简单易用、安全可靠、兼容性好，支持高效构建基于网络的数据库应用系统，受到系统开发人员的青睐。

##### 1. KingbaseES 的特点

KingbaseES 是基于客户/服务器模式的关系数据库管理系统，能跨越多种软/硬件平台，具有大型数据综合管理能力，支持所有主流编程语言的开发框架，具备从 Oracle、MySQL、SQL Server 等其他数据库迁移的能力。与其他同类软件相比，它具有以下突出特点。

① 通用性强。兼容多种硬件体系（x64/x86\_64、国产海光、兆芯、龙芯、飞腾/鲲鹏、申威等）；可在主流操作系统（Windows、Linux、UNIX）下运行，还可运行在国产操作系统（凝思、统信、中标麒麟、银河麒麟、麒麟信安、欧拉等）下；支持主流程序设计语言（如 C/C++、.NET、Java、Python、PHP 等）；符合 SQL92 标准并通过 PL/SQL 对标准 SQL 进行了过程化的扩展。

② 可用性高。通过预写式日志（WAL）、备份和恢复、集群化和自动故障转移等机制实现高可用性。还提供最大可用性架构（MAA），可定义不同保护级别，以满足各类应用对可用性和数据保护的需求。

③ 支持快速开发。提供主流客户端编程接口及开发框架，简化了数据库访问流程。配套的快速开发与运维平台 KES Plus 进一步提高了开发和部署数据库应用系统的效率，并能实现低代码化。

##### 2. KingbaseES 的安装和配置

KingbaseES 在 Windows 平台的版本主要分为标准版和企业版，均可运行在通用 x86\_64 体系结构的 CPU 上。标准版具备数据存储、数据处理、数据管理等基本功能，支持多用户并发访问，并提供图形化管理工具，能满足中小型企业及互联网应用的数据处理需要。企业版提供了更高级的特性和功能，适用于对数据库性能、安全性和可靠性要求更高的企业级应用。



KingbaseES V9 于 2024 年 11 月发布, 提供了面向全市场的版本, 包括兼容 Oracle 模式、兼容 MySQL 模式、兼容 SQLServer 模式等。安装前需要先获取数据库 License, 即 KingbaseES 提供的授权文件。安装和配置单行版 KingbaseES V9, 可参考金仓文档“基于 Windows 系统的数据库软件安装指南”。

除上述单行版外, 数据库应用快速开发与运维平台 KES Plus 产品中集成了 KingbaseES。用户无须单独下载安装 KingbaseES, 因为在安装 KES Plus 服务端后, 其自带的 KingbaseES (兼容 Oracle 模式) 即被安装, 本书选用的是教育版 KES Plus。其安装和配置过程见前言二维码。

### 3. KingbaseES 服务启动

KES Plus 服务端安装完后, 桌面会出现两个快捷方式图标。双击“启动 KES Plus 服务”图标 (见图 3.1), 可启动 KES Plus 服务和 KingbaseES 服务; 双击“停止 KES Plus 服务”图标 (见图 3.2), 可停止两个服务。



图 3.1 启动服务

### 4. KStudio 数据库开发工具

数据库开发工具 KStudio (也称客户端工具) 是 KingbaseES 自带的一个图形化工具, 主要用于配置和管理 KingbaseES 服务器、管理数据库对象、编辑和调试 SQL 语句、进行数据库的安全管理等。KES Plus 不带 KStudio, 需单独下载安装包 kstudio-20250409.rar, 解压后, 直接运行 KStudio.exe 即可。



图 3.2 停止服务

要使用 KStudio, 需先建立与数据库服务器的连接。单击工具栏中“连接”按钮 , 在“新建数据库连接”页面选择数据库类型为 KingbaseES (见图 3.3), 单击“下一步”按钮进入“连接设置”界面, 在“常规”选项卡中进行连接设置 (见图 3.4)。

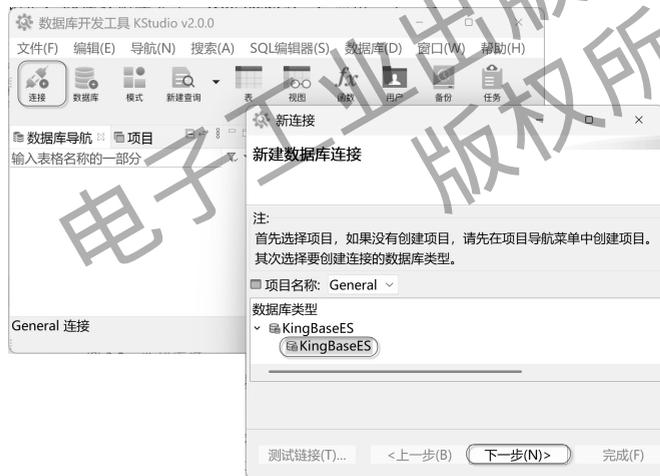


图 3.3 新建数据库连接



图 3.4 连接设置

- 数据库版本: 从下拉列表中选择所安装的 KingbaseES 版本, 这里选 V9R1。
- 主机 IP: 安装 KingbaseES 的计算机名或 IP 地址, localhost 或 127.0.0.1 均表示是本机服务器。
- 端口: KingbaseES 使用的 TCP/IP 协议端口号, 默认值为 54321。要连接 KES Plus 服务端, 端口号默认值为 54901。可根据安全需求进行更改。

- 数据库：设置登录的数据库，默认为 test。
  - 用户名和密码：默认用户名为 system，填写在安装服务器时设定的密码。
- “连接名”默认为 kingbase，可将连接信息保存下来，方便下次登录。单击“完成”按钮，连接名出现在 KStudio 左栏的导航树中（见图 3.5）。

KStudio 界面主要分为 4 个区域：菜单栏、工具栏、导航树、对象窗口（见图 3.5）。



图 3.5 KStudio 界面

左栏以可折叠的树状目录（简称为导航树）呈现了连接服务器的实例对象，包括数据库及其对象（如表、视图等）、表空间、逻辑备份&还原等；右栏为内容显示窗口，双击导航树中的对象，右栏将呈现所选对象的细节，也可以使用工具栏中的按钮选择对象。

### 3.1.2 KingbaseES 数据库管理模式

#### 1. KingbaseES 的两类数据库

KingbaseES 不仅存储数据库，所有与数据处理操作有关的信息也都存储在其中，因此它管理的数据库对象（见图 3.6）可分为两类：系统数据库和用户自定义数据库。

系统数据库存放的是 KingbaseES 工作时所需要的系统级信息，包括管理所需的元数据、系统配置和模板等（见表 3.1），以便支持用户创建和管理其他应用数据库，它们在创建数据库实例时由系统自动创建和维护。这些数据库很重要，损坏或丢失会影响其正常工作。test 数据库虽不存储系统级信息，但它也由系统自动创建。

用户自定义数据库是用户根据数据管理需要建立的数据库，由用户自行创建和维护。



图 3.6 数据库对象

表 3.1 KingbaseES V9 的系统数据库

数据库	主要作用
kingbase	系统级数据库。用于存储系统级的数据和配置信息，包括数据库实例的全局配置参数、系统级的表和视图等。可以删除，但删除后会影响系统的正常工作
security	系统级数据库。存储与数据库安全相关的配置和信息，包括用户和角色信息，访问控制策略、数据加密策略、审计策略，安全相关的操作日志。可以删除，但删除后会导致安全功能失效或数据丢失
template0	系统预定义的模板数据库。template0 是一个只读的、不可修改的模板，不包含任何自定义的设置或数据，用于创建干净的数据库副本。不允许修改和删除，不在 KStudio 的导航树中呈现
template1	系统预定义的模板数据库。template1 是一个可修改的模板，包含一些默认的设置和扩展，这些设置和扩展会被自动复制到新创建的数据库中。KingbaseES V9 中，默认使用 template1 创建新数据库。不允许删除，不在 KStudio 的导航树中呈现
test	默认测试数据库。用于测试和开发。可以删除，但需谨慎，因为数据库可能包含测试数据

## 2. 数据库的逻辑结构

数据库的逻辑结构是从用户操作视角看到的数据库的组成对象及其关系。一个 KingbaseES 数据库由一组对象组成，包括表、视图、存储过程和触发器等，用户或应用程序通过操作这些对象就可以实现对数据库的创建、管理和使用。

在 KingbaseES 中，一个数据库的所有对象被分类组织到多个模式中（见图 3.5）。模式（Schema）是一个逻辑上的命名空间，用于将数据库对象（如表、视图、函数等）进行分组和管理。通过模式，可以将相关的数据库对象组织在一起，便于管理；同时隔离不同用户或不同业务模块的对象，避免命名冲突。一个数据库可以包含多个模式，但每个模式只能属于一个数据库；一个模式可包含多个数据库对象，而每个数据库对象只属于一个模式。

在创建数据库时，KingbaseES 会默认创建三个模式：PUBLIC、SYS\_CATALOG 和 INFORMATION\_SCHEMA。

- PUBLIC：公共模式，当用户创建对象（如表、视图、函数等）时，如果没有指定模式，则这些对象会默认存储在 PUBLIC 模式中。
- SYS\_CATALOG：系统模式，用于存储系统表和所有内置的数据类型、函数和操作符。不在 KStudio 导航树中呈现。
- INFORMATION\_SCHEMA：系统模式，用于存储数据库的元数据（如表结构、列信息等）。不在 KStudio 导航树中呈现。

此外，创建数据库时，系统还自动创建了 perf（与性能监控相关）、sysaudit（与审计功能相关）、sysmac（与标记和强制访问控制功能相关）等扩展模式。

## 3. 数据库的物理结构

数据库的物理结构是指从机器操作视角看到的数据库结构，是数据库在磁盘中的存储和管理方式。

KingbaseES 数据库中所有数据都以操作系统文件的方式存储在磁盘上。具体来说，数据库中表或索引等对象都被记录在表空间中，并被拆分成若干个段存储，一个段对应一个物理文件。数据库集簇（指由单个数据库服务器实例管理的多个数据库的集合）的所有数据文件、日志文件、控制文件都存储在一个数据库数据文件夹里。

提示：在单独安装 KingbaseES V9 时，其数据文件夹为 data，默认存储位置为“C:\Program

Files\Kingbase\ES\V9\data”。KES Plus 服务端安装时自带的 KingbaseES V9，其数据文件夹为 data\dbdata，默认存储位置为“D:\kingbase\kesplus\data\dbdata”。

在数据文件夹中，存储数据库数据和集簇全局信息的文件夹分别是 base 和 global。

- ▶ base 文件夹：为集簇中每个用户自定义数据库创建一个独立的文件夹，以该数据库的 OID（对象标识符，系统生成的唯一标识符）命名，库中的每个表和索引都存储为独立的文件。e\_learning 数据库的 OID 为 43198，其中包含了很多文件（见图 3.7）。
- ▶ global 文件夹：存储整个数据库集簇相关的全局系统表和控制信息，这些文件为集簇中的所有数据库共享。



图 3.7 base 文件夹中的数据库文件

此外，配置文件 kingbase.conf、sys\_hba.conf 和 sys\_ident.conf 通常也存储在数据文件夹中。

提示：可以通过 SQL 语句来查询 OID 和数据库对应关系：SELECT oid, dataname FROM pg\_database。注意，复制其中的文件是无法直接使用的。后面会介绍如何通过 SQL 语句导出或备份数据库方式来转储数据库。

### 3.1.3 KingbaseES 数据库的数据类型

按照数据库设计要求，数据表中的每个字段都需要定义明确的数据类型来规定该字段的数据取值类型、范围和存储格式。在设计数据表时，应根据需要为各字段选择恰当的数据类型，既要满足存储需求又要方便程序处理，但也不能大材小用，以免浪费存储空间。

KingbaseES 不但支持所有标准 SQL 中的数据类型，还对其进行了扩展，常用数据类型（兼容 Oracle 模式）见表 3.2。

表 3.2 KingbaseES 的常用数据类型

数据类型	主要类型常用标识符
整型	tinyint、smallint、int、bigint
浮点及定点型	real、double、decimal、numeric
字符及文本型	character、varchar、text
日期及时间型	date、time、timetz、timestemp
二进制型	bytea、bolb
布尔型	boolean

#### (1) 整型

整型用于存储整数。有 4 种常用的整型，分别用于存储不同范围的整数，见表 3.3。

表 3.3 整型

数据类型	别名	数据范围	长度
tinyint	—	$-2^7 \sim 2^7 - 1$	1B
smallint	int2	$-2^{15} \sim 2^{15} - 1$	2B
integer	int, int4	$-2^{31} \sim 2^{31} - 1$	4B
bigint	int8	$-2^{63} \sim 2^{63} - 1$	8B

## (2) 浮点及定点型

浮点及定点型用于存储含有小数的十进制数，有 4 种常用的数据类型，见表 3.4。

表 3.4 浮点及定点型

数据类型	别名	数据范围	长度
real	float4	$-3.40E+38 \sim 3.40E+38$	4B
double precision	double, float8	$-1.79E+308 \sim 1.79E+308$	8B
decimal(p,s)	number(p,s)	精度 p 取值范围为 1~1000，标度 s 取值范围为 0~1000	(p+1)/2B
numeric(p,s)			

real 为单精度浮点型，double 为双精度浮点型。decimal 和 numeric 是定点型，用 p 表示总的位数，s 为小数位数，若超出精度范围则会四舍五入。例如，decimal(4,1)表示的数据范围为-999.9~999.9。decimal 占用空间大，可用于存储高精度数据，适合存储银行账目。

## (3) 字符及文本型

字符及文本型用于存储由字母、符号和数字组成的字符串。字符串常量要用单引号作为起止界限，例如，'9aC'。常用的字符及文本型有 3 种，见表 3.5。

表 3.5 字符及文本型

数据类型	说明
character(n[char   byte]) 别名: char(n[char   byte])	定长字符串长度固定为 n，不足时，自动在串尾添加空格。 n 的最大值为 10485760，最小值为 1，默认值为 1。若长度超出 n，则截断或报错
character varying(n [char   byte]) 别名: varchar(n[char   byte]) nvarchar[(n[char byte])]	变长字符串长度为字符串的实际字符个数，长度不超过 n。 n 的最大值为 10485760。如果 n 省略，则表示接受任何长度的串
text	变长字符串型。长度最大为 1GB。适合存储长文本信息

提示：表中参数[char|byte]是可选项，用于表示字符串长度的单位。char 表示以字符为单位，byte 表示以字节为单位。一个英文字符通常占 1 字节，而汉字会占用若干字节。文本数据通常选 char。如果是二进制数据，如图像、音频等，通常选 byte。若未显式指定字符串长度的单位，则以系统设置为准，可用 SQL 语句查看：SHOW nls\_length\_semantics。

若某个字段的字符串长度固定，则选用 char 型；若长度明显不一致，则选用 varchar 型。例如，student 表中，Gender 属性用 char(1)存储定长的 1 位字符：'男'或'女'，属性 StudentName 存储的姓名长度不同，使用 varchar(16)，最长可含 16 个字符。

`varchar` 可以存储不同字符集的字符串，具体取决于数据库的字符集设置。如果数据库使用的是单字节字符集（如 ASCII），则一个字符占 1 字节；如果使用多字节字符集（如 UTF8），一个字符可能占多字节，适合存储中文或中英文混合字符串。

`nvarchar` 专用于存储 Unicode 字符，常采用 UTF16 编码。每个字符都占 2 字节。

#### （4）日期及时间型

日期及时间型用于存储日期或时间相关的数据，有 4 种常用数据类型，见表 3.6。

表 3.6 日期及时间型

数据类型	说明
<code>date</code>	存储日期和时间值，不包含时区信息。可存储年、月、日、时、分、秒。推荐格式为 'YYYY-MM-DD HH:MM:SS'。
<code>time[(p)] [without time zone]</code> 别名： <code>time</code>	存储一天中的时间值（无时区），格式为 HH:MM:SS[.SSSSSS]或 HHMMSS[.SSSSSS]。 p 表示秒精度，即秒中的小数位数，取值范围为 0~6。
<code>time[(p)] with time zone</code> 别名： <code>timetz</code>	存储一天中的时间值，包括时区，格式为 HH:MM:SS[.SSSSSS]或 HHMMSS[.SSSSSS]。 p 表示秒精度，取值范围为 0~6。
<code>timestamp[(p)] [without time zone]</code> 别名： <code>timestamp</code>	存储日期和时间值。精度可以达到小数秒，默认精度为 6 位小数秒（微秒）。格式为 'YYYY-MM-DD HH:MM:SS[.SSSSSS]'。 取值范围为 '1000-01-01 00:00:00.000000' 到 '9999-12-31 23:59:59.999999'。 p 表示秒精度，取值范围为 0~6。

日期及时间型常量必须按其数据类型约定的完整格式书写，并用单引号括起来，例如，`date` 型常量值 '2028-12-08 12:35:29'。

#### （5）二进制型

二进制型适合存放用二进制数表达的数据，有两种常用数据类型，见表 3.7。

表 3.7 二进制型

数据类型	说明
<code>bytea</code>	变长的二进制串（字节数组），长度最大为 1GB。用于存储中小规模的数据，如图像、音频等。
<code>blob</code>	变长的二进制大对象，长度最大为 1GB。常用于存放大型的二进制数据，如大型文档、图书或长视频。

虽然图片、声音等多媒体数据可以转化为二进制数以 `bytea`、`blob` 类型存储，但读取和写入都不方便。一般将媒体文件存放在文件系统中，只在数据库中存放文件路径和文件名。

#### （6）布尔型

布尔型用于存储取值为真、假或未知的数据，见表 3.8。

表 3.8 布尔型

数据类型	说明
<code>boolean</code>	KingbaseES 布尔型可取三种值： <code>true</code> 、 <code>false</code> 、 <code>null</code> ，分别表示真、假、未知三种状态，用 1 字节存储。

若某字段取值“真”或“假”（“是”或“否”）时，可以选用 `boolean` 类型。例如，`teacher` 表中字段 `AdminYN` 存储“是否管理员”信息，使用了 `boolean` 类型。

### 3.1.4 在线学习系统数据库设计案例

第2章通过实例介绍了在线学习系统 e\_learning 数据库的设计过程，并得到关系模型逻辑结构。通过为每个字段定义恰当的数据类型和对各个表进行数据完整性约束定义，就可以得到该数据库的物理结构设计。最后按照物理设计建立数据库。

下面给出在线学习系统 e\_learning 数据库在 KingbaseES 中的完整设计，它基本采用图 2.13 的数据库关系模型，但为了后续实例应用，进行了以下变动：

① 各种标识符包括数据库名、数据表名及各字段名均采用英文命名。这主要是为了避免由于汉字编码方式不一致导致的乱码问题，同时也可减少开发时的中英文输入切换。student 表存储学生信息，certificate 表存储毕业证信息，subject 表存储学科信息，course 表存储课程信息，courseware 表存储课件信息，courseenroll 表存储学生选课及成绩信息，teacher 表存储教师信息，testquestion 表存储测试题信息，studentanswer 表存储答题信息。

② 在一些关系模式中增加了属性，使得信息更全面丰富。本书后续章节的例题都以该数据库为基础。读者可以通过该数据库回顾和理解数据库基本知识。e\_learning 数据库脚本见前言二维码。

#### 1. 数据库的逻辑结构

e\_learning 数据库包含 9 个表，数据库关系模式和表之间的关系见图 3.8，其中钥匙图标标注的字段是主键，表间连接线表示两个表之间通过外键建立了关系。

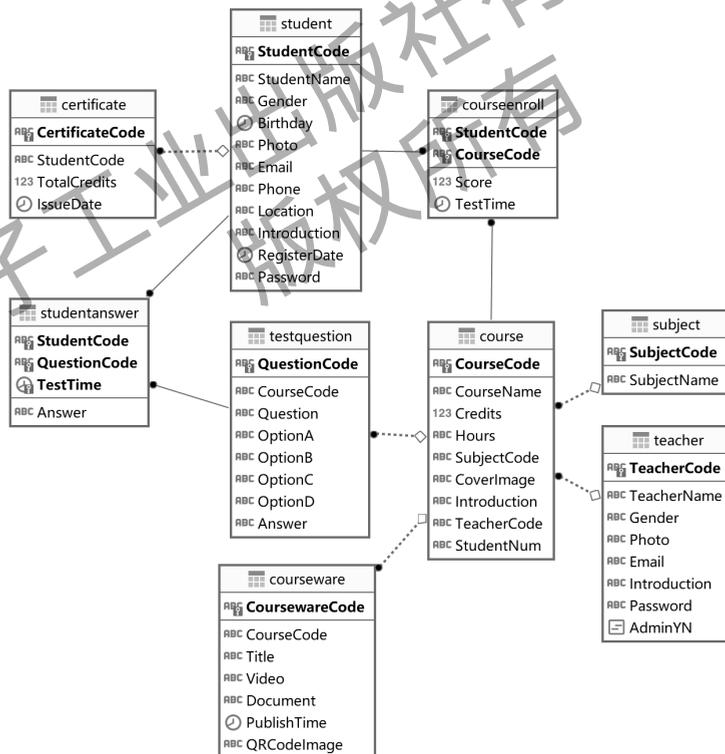


图 3.8 e\_learning 数据库关系图

#### 2. 数据库的物理结构

各表的物理结构见表 3.9 至表 3.17。这里只给出各表的结构设计、完整性约束和索引。

表 3.9 student 表

字段名称	字段说明	类型定义	属性限定	索引	关系(外键)
StudentCode	学号	int, 自增	Primary Key, 初值为 1001	唯一索引	
StudentName	姓名	varchar(16)	Not Null	唯一索引	
Birthday	生日	date	Not Null		
Gender	性别	char(1)	Not Null, '男'或'女', 默认为'男'		
Photo	照片文件路径	varchar(30)			
Email	电子信箱	varchar(30)	Not Null		
Phone	电话	varchar(11)			
Location	生源地	varchar(6)		索引	
Introduction	个人简介	text			
RegisterDate	注册日期	timestamp	默认为 CURRENT_TIMESTAMP 或者默认为 SYSTIMESTAMP		
Password	密码	varchar(6)	Not Null		

表 3.10 certificate 表

字段名称	字段说明	类型定义	属性限定	索引	关系(外键)
CertificateCode	证书号	char(8)	Primary Key	唯一索引	
StudentCode	学号	int	Not Null, Unique	唯一索引	student: StudentCode
TotalCredits	总学分	decimal(4,1)			
IssueDate	发证日期	date	Not Null	索引	

表 3.11 subject 表

字段名称	字段说明	类型定义	属性限定	索引	关系(外键)
SubjectCode	学科号	char(3)	Primary Key	唯一索引	
SubjectName	学科名	varchar(10)	Not Null		

表 3.12 teacher 表

字段名称	字段说明	类型定义	属性限定	索引	关系(外键)
TeacherCode	工号	char(4)	Primary Key	唯一索引	
TeacherName	姓名	varchar(16)	Not Null		
Gender	性别	char(1)	Not Null		
Photo	照片文件路径	varchar(30)			
Email	电子信箱	varchar(30)	Not Null		
Introduction	个人简介	text			
Password	密码	varchar(128)	Not Null, 默认为'666666'		
AdminYN	是否管理员	boolean	Not Null, 默认为 false		

表 3.13 course 表

字段名称	字段说明	类型定义	属性限定	索引	关系(外键)
CourseCode	课程号	char(4)	Primary Key	唯一索引	
CourseName	课程名	varchar(16)	Not Null		
Credits	学分	decimal(3,1)	Not Null, 默认为 0		
Hours	学时	int	Not Null		
SubjectCode	学科号	char(3)	Not Null	索引	subject: SubjectCode
CoverImage	封面图片	varchar(30)			
Introduction	课程简介	text			
TeacherCode	教师工号	char(4)	Not Null	索引	teacher:TeacherCode
StudentNum	选课人数	int			

表 3.14 courseenroll 表

字段名称	字段说明	类型定义	属性限定	索引	关系(外键)
StudentCode	学号	int	Primary Key	唯一索引	student: StudentCode
CourseCode	课程号	char(4)		索引	course: CourseCode
Score	成绩	decimal(4,1)	大于或等于 0		
TestTime	测试时间	date			

表 3.15 courseware 表

字段名称	字段说明	类型定义	属性限定	索引	关系(外键)
CoursewareCode	课件号	int, 自增	Primary Key, 初值为 10001	唯一索引	
CourseCode	课程号	char(4)	Not Null	索引	course: CourseCode
Title	课件标题	varchar(16)	Not Null		
Video	视频地址	varchar(30)	Not Null		
Document	文档地址	varchar(30)			
PublishTime	发布时间	date	Not Null		
QRCodeImage	二维码图	varchar(30)			

表 3.16 testquestion 表

字段名称	字段说明	类型定义	属性限定	索引	关系(外键)
QuestionCode	问题号	int, 自增	Primary Key	唯一索引	
CourseCode	课程号	char(4)	Not Null	索引	course: CourseCode
Question	题干	text	Not Null		
OptionA	选项 A	varchar(50)	Not Null		
OptionB	选项 B	varchar(50)	Not Null		

续表

字段名称	字段说明	类型定义	属性限定	索引	关系(外键)
OptionC	选项 C	varchar(50)	Not Null		
OptionD	选项 D	varchar(50)	Not Null		
Answer	正确答案	char(1)	Not Null		

表 3.17 studentanswer 表

字段名称	字段说明	类型定义	属性限定	索引	关系(外键)
StudentCode	学号	int	Primary Key	唯一索引	student: StudentCode
QuestionCode	问题号	int			testquestion: QuestionCode
TestTime	测试时间	date			
Answer	学生答案	char(1)			

## 3.2 KingbaseES 数据库的创建与维护

### 3.2.1 创建数据库

在 KingbaseES 中创建数据库主要有两种方法：① 采用 KStudio 可视化创建；② 执行 SQL 语句创建。本节介绍采用 KStudio 可视化创建数据库的方法。

**【例 3.1】**创建 e\_learning 数据库。

① 在 KStudio 导航树中，右击“数据库”，选“新建数据库”命令，或者单击工具栏中的“数据库”按钮。

② 在打开的页面中填写数据库的基本属性（见图 3.9），即可创建数据库。

在“名称”文本框中，输入需要创建的数据库名：e\_learning。

在“属主”下拉列表中，指定拥有该数据库的用户，取默认的 system（数据库管理员）。

在“编码”下拉列表中，指定数据库的编码方式，取默认的 UTF8 编码，它在处理中文字符时不会出现乱码问题。

其他属性取默认值。单击“确定”按钮后，在导航树中可看到 e\_learning（见图 3.5）。

**提示：**若数据库中需要存储中文字符，强烈推荐选择 UTF8 编码。

删除数据库的操作方法如下。

① 在 KStudio 导航树中，右击要删除的数据库，选“删除”命令。

② 在“删除对象”页中，单击“是”按钮即完成。

**提示：**删除操作将数据库文件及数据从服务器磁盘中全部删除，该数据库中所有的对象均被删除，即永久性删除，不能恢复，所以删除操作要慎重！

图 3.9 基本属性

### 3.2.2 创建和维护表

新建数据库只生成了必要的系统文件，接下来的工作就是创建数据库中的各个对象，其中最重要的是表。表的创建也可采用 KStudio 可视化和 SQL 命令两种方式。

#### 1. 创建表

【例 3.2】在 e\_learning 数据库中创建 student 表，完成各字段的定义。

① 在 KStudio 导航树中，单击 e\_learning 数据库左侧的 >，展开 e\_learning 数据库，再展开“模式→public”，右击“表”，选“新建 表”命令，打开新建表页（称为表编辑器）（见图 3.10）。



图 3.10 student 表的创建

② 在“基本属性”选项卡的“基本属性”区，“名称”输入“student”。在“字段”区，单击“新增”按钮添加字段。根据表设计说明（见表 3.9）依次完成每个字段名称和数据类型等的设置。例如，在第 1 个新增字段行中，“名称”输入“StudentCode”，“类型”选择“int4”，“非空”改为“是”，表示该字段的值不允许为空，“注释”填写当前字段的描述说明“学号”。设置完所有字段的属性后，单击“确定”按钮完成表的创建。

提示：“非空”取值为“否”，表示添加新记录时该列的值可以没有，即为空值（Null）。

注意，KStudio 教育版中，在“字段属性”区勾选“自增”复选框，不能将该字段设置为自增字段，需要通过创建表的 SQL 语句来设置。

③ 在导航树中展开“e\_learning→模式→public→表”，可以看到创建好的 student 表。

#### 2. 修改表结构

在导航树中，展开“e\_learning→模式→public→表”，找到待修改的表，右击，选“编辑 表”命令，打开编辑表页，再次进入表编辑器。选中需要修改的字段，可重新进行各种设置，如添加或删除字段、修改字段数据类型等。切换到“约束”、“外键”等选项卡，可以设置主键、外键等。修改完毕后保存。

### 3. 重命名表

右击需要重命名的表，选“重命名”命令，在表名处输入新表名即可。

### 4. 删除表

右击待删除的表，选“删除”命令，打开“删除对象”对话框，单击“是”按钮，即可删除该表。注意，不能删除系统表以及有外键参照约束的表。

**提示：**删除表要十分谨慎，因为当表被删除时，该表的结构定义、数据、索引和约束等都被永久地从数据库中删除了！

参照表 3.9 至表 3.17 的说明，可依次创建 e\_learning 数据库中的各个表。

## 3.2.3 创建数据完整性约束规则

KingbaseES 提供了定义约束、检查和保持数据符合约束的完整性控制机制，其主要通过限制表的字段（列）、记录（行）取值，以及表之间的参照数据来保证数据完整性。用户在创建表时应该设置各项完整性约束，以保证表中数据的一致性。常用的数据完整性约束见表 3.18。

表 3.18 KingbaseES V9 常用的数据完整性约束

完整性	约束	作用
实体完整性	主键（Primary Key）约束	约束主键字段不能出现重复值，以保证表中记录的唯一性
	唯一性（Unique）约束	约束非主键字段不出现重复值
域完整性	默认值（Default）约束	对没有插入值的字段自动添加表定义时对该列设置的默认值
	非空值（Not Null）约束	限定某一字段中必须有值，即不允许空值
	检查（Check）约束	限定某一字段中的取值或数据格式
参照完整性	外键（Foreign Key）约束	通过表间关系约束字段值的有效性

#### 1. 实体完整性约束

实体完整性主要体现为表中记录的唯一性。其实现方式有主键约束和唯一性约束。

##### (1) 主键约束

主键是表中能保证表中记录唯一性的一个或多个字段的组合。主键字段值不能重复且不能为空。一个表只能有一个主键。

**【例 3.3】**定义 student 表的主键为 StudentCode。

打开 student 表的表编辑器，切换到“约束”选项卡（见图 3.11），单击“新增”按钮，添加一个新约束，“名称”为“student\_pkey”；“类型”选择“主键”；单击“列”对应的浏览按钮，在弹出的对话框中将 StudentCode 从“选择列”移入“确认列”中，单击“确认”按钮，完成主键设置。当保存表时，该约束设置即保存在表中。

##### (2) 唯一性约束

使用唯一性约束可以保证非主键字段不出现重复值。在下列情况下应使用唯一性约束。

- ① 对非主键字段或字段的组合强制唯一性。可以为表定义多个唯一性约束。
- ② 对允许空值的字段强制唯一性。主键约束虽然强制唯一性，但不允许为空值。

**【例 3.4】**在 student 表中不允许出现姓名且生日相同的记录，即对 StudentName 和 Birthday 字段组合定义唯一性约束。

在 KingbaseES 中，唯一性约束通过设置唯一键来实现，具体步骤如下。



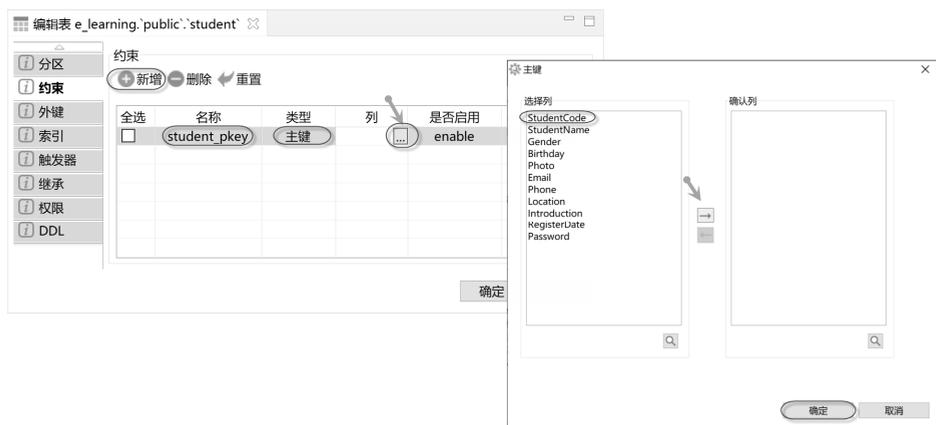


图 3.11 设置主键约束

① 打开 student 表的表编辑器，切换到“约束”选项卡。

② 单击“新增”按钮，添加一个新约束。输入新约束名称“IX\_StudentNameBirth”；“类型”选择“唯一键”；单击“列”对应的浏览按钮，在弹出的对话框中分别将 StudentName、Birthday 从“选择列”移入“确认列”，单击“确认”按钮，完成唯一约束设置（见图 3.12）。当保存表时，该约束设置即保存在表中。

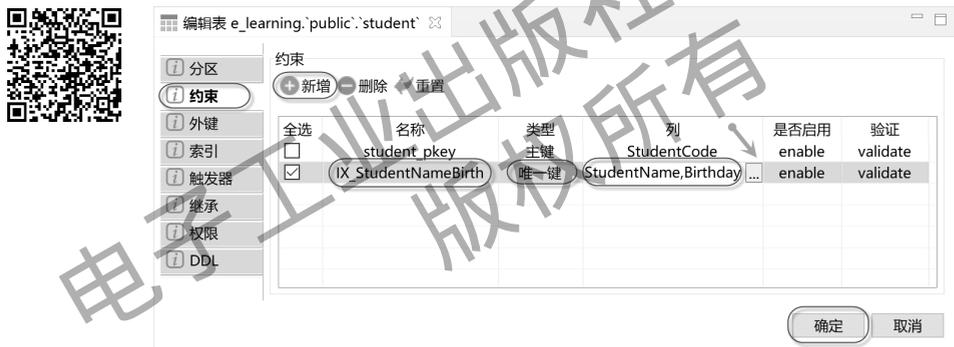


图 3.12 设置唯一性约束

## 2. 域完整性约束

域完整性主要体现为表中字段值的有效性。其约束的实现方式有默认值约束、非空值约束和检查约束。

### (1) 默认值约束

默认值约束就是为某个字段定义一个默认值，当添加的新记录在该字段中无输入值时，系统将该默认值填入。注意，默认值的数据类型与字段的一要一致。

**【例 3.5】**在 student 表中设置 Gender 字段的默认值为“男”，设置注册日期 RegisterDate 字段默认值为插入记录时的当前日期。

打开 student 表的表编辑器，选择 Gender 字段，在下方“字段属性”区中，“默认值”填写“'男'”，需加单引号（见图 3.13）；选择 RegisterDate 字段，“默认值”填写“CURRENT\_DATE”。如果设置为当前时间，则填写“CURRENT\_TIMESTAMP”或“NOW()”。最后保存表。



图 3.13 设置默认值约束

### (2) 非空值约束

非空值约束可以限定某个字段必须有值，即在向表中添加新记录时必须为该字段输入一个值，或已经为该字段定义了默认值，否则该记录不允许存入数据库。

**提示：**数字 0、空白或零长度的字符串 ("") 并不是空值。空值意味着没有输入，通常表示值未知或未定义。

**【例 3.6】**在 student 表中设置 StudentCode、StudentName、Gender、Birthday 和 Email 字段为非空值，其他字段允许为空值。

新增字段时，“非空”默认为“否”，即允许空值，如果要字段设置为不允许空值，只需在对应的“非空”列中单击“否”将取值切换为“是”即可，设置好后保存表。

### (3) 检查约束

检查约束用于限制表中字段的值必须满足某些条件。它通过指定一个逻辑表达式来定义字段值的合法性规则。在向表中添加或修改记录时，输入的字段值必须符合预定义的规则。

**【例 3.7】**在 student 表中设置 Gender 字段只能取值“男”或“女”。

打开 student 表的表编辑器，切换到“约束”选项卡，单击“新增”按钮，添加一个新约束，“名称”为“chk\_Gender”；“类型”为“检查”；单击“列”对应的浏览按钮，在弹出的对话框中输入 Gender IN ('男','女') (也可以为 Gender='男' OR Gender='女')，确认完成(见图 3.14)。最后保存表。

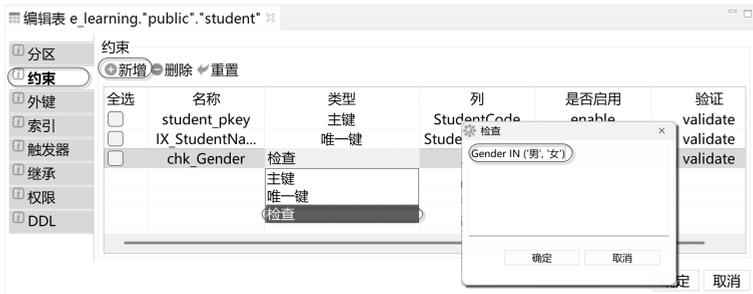


图 3.14 设置检查约束

### 3. 参照完整性约束

参照完整性约束通过表间关系来约束字段值的一致性，即通过外键约束实现。两个建立关联关系的表，外键的取值必须是主表中主键中存在的值或空值，并可以进一步设定违反约束时的处理方式。

**【例 3.8】**在 `courseware` 和 `course` 表之间建立关联关系，设置参照完整性约束。在该关系中，将主表 `course` 的 `CourseCode` 字段设置为主键，将外表 `courseware` 的 `CourseCode` 字段设置为外键，实施参照完整性约束的设置，拒绝违反参照完整性的操作。

打开 `courseware` 表的表编辑器，切换到“外键”选项卡，单击“新增”按钮，新增一个参照完整性约束。“名称”为“FK\_courseware\_course”，“列”选择“CourseCode”，“依赖表”选择“public.course”，“依赖列”选择“CourseCode”，“删除”和“更新”均选择“保持”（见图 3.15），最后保存表。

**提示：**`course` 表的 `CourseCode` 字段必须已经被设为主键或唯一性约束。外键名可自行定义，习惯的命名形式为“FK\_外表名\_主表名”，其中 FK 是 ForeignKey 的缩写。



图 3.15 设置外键约束

在“删除”和“更新”下拉列表中有 4 个处理策略选项，表明违反外键约束时的处理方式，它们在对主表进行删除或更新操作引起主键变化时起作用。

- 无动作：默认行为。显示错误信息，并终止该操作。
- 保持：显示错误信息，并终止该操作。
- 级联：如果是“更新”，则允许更新主表记录，同时自动更新外表中被其约束的所有相关记录的外键值；如果是“删除”，则允许删除主表记录，同时自动删除外表中被其约束的所有相关记录。
- 设为空：更新或删除主表记录时自动将外表中约束的相关记录的外键值设为空值。

“保持”和“无动作”的结果是相同的，它们的差别在于检查时间不同：“无动作”对删除或更新数据延期检查，而“保持”则在保存操作结果时立即检查。

**【例 3.9】**处理策略为“保持”，修改主表 `course` 中的课程号，观察外键约束的作用。

① 右击 `course` 表，选“编辑数据”命令，打开 `course` 表的数据页，即表数据编辑器，修改 C001 为 C015。

② 单击工具栏中的“保存”按钮时，会出现错误提示消息，对 `course` 表的更新或删除操作违反了 `courseware` 表上的外键约束（见图 3.16）。

③ 单击“确定”按钮，返回数据页，单击“取消”按钮，C015 恢复为 C001。关闭 `course` 表的数据页。

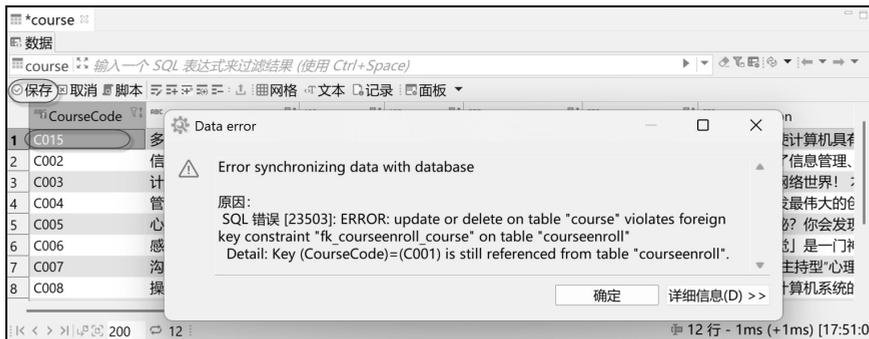


图 3.16 修改主键值时的错误提示信息

【例 3.10】处理策略为“保持”，在外表 courseware 中插入记录，观察外键约束的作用。

① 右击 courseware 表，选“编辑数据”命令，打开 courseware 表的数据页，单击工具栏中的，添加新行，依次输入以下各字段值：(10006, C018, 大数据应用, 10006.MP4, 10006.pptx, 2017-10-13)。

② 单击“保存”按钮，出现错误提示信息（见图 3.17），对 courseware 表的插入或更新操作违反了外键约束“FK\_courseware\_course”。

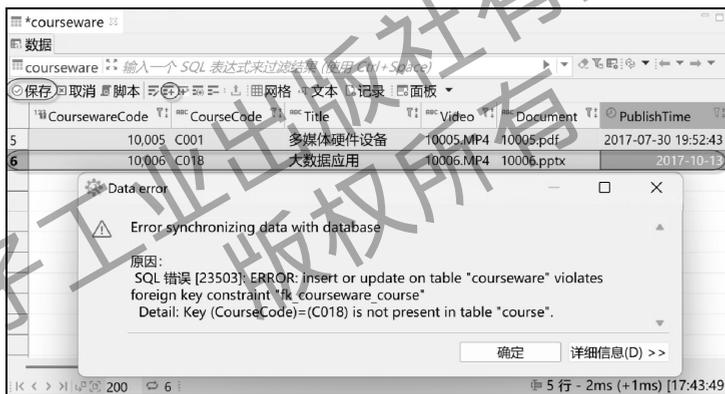


图 3.17 插入记录时的错误提示信息

③ 返回数据页，单击“取消”按钮删除该记录。关闭 courseware 表的数据页。

④ 打开 course 表的数据页，看到 course 表中无 CourseCode 为 C018 的新记录。

外键约束需要额外系统开销来进行检测，如果不要求特别严格的约束，也可在设计表时不考虑参照完整性约束，而是通过应用程序代码实现。

## 3.2.4 创建索引

### 1. 索引的概念

为了提高数据操作时的记录定位速度，一般需要在表上建立索引。索引与图书目录或字典检字索引的原理类似。它由表中的一列或者若干列的值与其对应的记录在表中的地址所组成。因此，数据库中一个表的存储由两部分组成：数据页和索引页。

例如，为 student 表建立以“姓名”为索引键的索引 1 和以“出生日期”+“性别”为索引键的索引 2（见图 3.18）。



图 3.18 索引示意图

可以在一个表上建立多个索引。索引建立后，由系统进行维护并自动使用。用户也可以在开发应用程序时指明使用哪个索引。使用索引的目的是改善查询性能，加快依据索引字段对表中记录的检索。例如，索引 1 支持以“姓名”为检索条件的快速查找。

索引虽然能改善查询性能，但也耗费了磁盘空间，并且当对表进行增加、修改或删除数据操作时，系统需要花费一些时间来维护索引页。所以通常不要在一个表上建立太多索引，也不要建立不常用的索引。一般需要建立索引的字段有主键和外键、频繁作为检索条件的字段、经常需要排序的字段。

## 2. 索引的分类

### (1) 聚集索引和非聚集索引

根据索引的顺序与表中数据的物理顺序是否相同，可分为聚集索引和非聚集索引。

聚集索引对表中的记录按指定索引字段值进行排序后再重新存储到磁盘中，使表的物理顺序与索引的一致。聚集索引的优点是检索速度快，缺点是重组记录物理顺序来维护索引的时间和空间消耗大。每个表只能建一个聚集索引，通常建在主键上。

非聚集索引通过增加索引页来存储组成非聚集索引的字段值和地址，不需要将记录重新排序。可以建立多个非聚集索引。图 3.18 中的索引 1 和索引 2 都是非聚集索引。

### (2) 单列索引和复合索引

根据索引字段由一列还是多列构成可分为单列索引和复合索引。图 3.18 中的索引 1 是单列索引，而索引 2 是由两个字段构成的复合索引，即若生日相同，再以性别排序。

### (3) 唯一索引和非唯一索引

根据表中任意两条记录的被索引字段值是否允许相同，可分为唯一索引和非唯一索引。

非唯一索引允许各记录的索引字段值相同或取空值，而唯一索引要求每条记录的索引字段值不能相同，也是强制取值唯一性的一种机制。

## 3. 索引的创建

KingbaseES 常见的索引方式有普通索引、唯一索引和全文索引。另外，主键默认建立了一个唯一索引。

- ① 普通索引：基本索引类型，属于非唯一索引，允许索引字段中有重复值或空值。
- ② 唯一索引：不允许表中任何两条记录具有相同的索引字段值，即创建唯一性约束。
- ③ 全文索引：一种用于高效检索文本数据的索引技术。

KingbaseES 提供了多种索引方法，包括 B-tree、Hash、GiST（树状结构）、SP-GiST、GIN（通用倒排索引）、brin 和 Bitmap，以适应不同的应用场景。默认采用 B-tree 索引，适用于等值查询和范围查询。全文索引支持对文本内容进行分词、模糊匹配和语义搜索，适用于处理大段文本的查询。全文索引通常基于 GIN 或 GiST 索引方法实现。

提示：KingbaseES 中没有传统意义上的“聚集索引”（如 MySQL 中的聚集索引），但利用以下 SQL 语句可以基于索引对表进行物理排序，实现类似的聚集功能：

CLUSTER 表名 USING 索引名;

【例 3.11】在 student 表中按生源地 Location 字段建立普通索引，索引方法为 B-tree。

打开 student 表的表编辑器，选择“索引”选项卡（见图 3.19），可看到例 3.3 建立主键时自动创建的唯一索引 student\_pkey，例 3.4 在（StudentName, Birthday）组合上建立唯一性约束时自动创建的唯一索引 IX\_StudentNameBirth。单击工具栏中的“新增”按钮，添加一个新索引，名称为“IX\_Location”，类型为“B-tree 索引”，是否唯一为“否”，列为“Location”。最后保存表。



图 3.19 创建索引

索引管理包括索引的查看、修改和删除操作。在表编辑器中单击“索引”选项卡，即可完成查看索引、更改索引名、删除索引操作。注意，索引的其他设置不能在原索引上直接修改，需要先“删除”原索引后，再“新增”一个索引。

### 3.2.5 添加记录

在实际开发系统时，只有当一个数据库中的所有表及完整性约束完全创建好后，才向其中添加记录，而且记录的添加和维护操作都是由用户通过应用程序进行的。

KStudio 提供了数据维护功能，包括数据的查看、插入、删除和修改操作，主要用于调试程序或临时操作数据库。

#### 1. 插入记录

右击 student 表，选“编辑数据”命令，打开数据页（见图 3.20），表中的数据按行显示，一条记录占一行。工具栏中有记录操作按钮，支持向表中插入、删除或修改记录，右击某条记录后选择快捷菜单命令也可实现相应的操作。



图 3.20 student 表的数据页

可以一次向表中插入多条记录，操作步骤如下：

① 选定记录插入位置，单击工具栏中的，弹出一个对话框，输入行数后确认，则在该记录后新增指定行数的空行，所有字段值都为 Null。

② 逐个输入字段值，按 Tab 键可移至下一个字段。

提示：要注意不允许为空值并且无默认值的字段必须输入值。

③ 输入完数据，单击“保存”按钮，然后关闭数据页。

## 2. 修改记录

定位到要修改的记录，对其中的字段值进行编辑修改。

## 3. 删除记录

定位到要删除的记录，单击工具栏中的，被删除记录的背景将变为红色。单击“保存”按钮后关闭数据页。

## 3.2.6 查看和导出 E-R 图

KStudio 支持可视化管理数据模型，可以查看和导出已创建数据库的 E-R 图。

【例 3.12】导出 e\_learning 数据库的 E-R 图，输出为 e\_learning.png。

① 在 KStudio 中，展开“数据库→e\_learning→模式”，右击 public，选“ER 图”命令，在对象窗口中显示数据表的结构及关系，可手工拖曳调整各表位置（见图 3.21）。

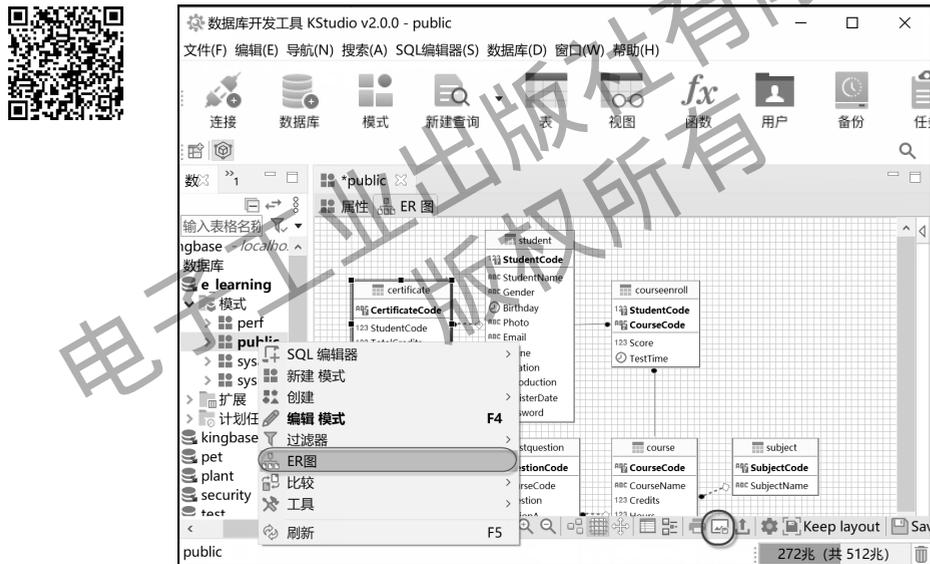


图 3.21 在对象窗口中显示的 E-R 图

② 在对象窗口下部的工具栏中，单击打开输出文件对话框，可保存图表为.png、.gif、.bmp 或.graphml 外部格式，输入文件名“e\_learning.png”，保存即可。

## 实验与思考

实验目的：了解 KingbaseES 数据库的逻辑结构和物理结构，进一步理解关系表及数据完整性约束的概念、索引的概念；掌握创建和维护数据库、数据表的方法。

实验环境及素材：KingbaseES V9 和 KStudio，bookstore 数据库设计文档（参见附录 A）。

1. 创建一个名称为 bookstore 的数据库，支持网上书店的信息管理。
2. 按附录 A 中表 A.1 至表 A.8 的表结构创建表，并为各个表定义相应的域完整性约束，定义主键实现实体完整性约束。至少完成 publisher 表、book 表、customer 表的创建。
3. 在各相关表之间建立关系，实施参照完整性约束，并保存 E-R 图为“bookstore.png”。
4. 为 publisher 表、book 表、customer 表各添加三条记录。  
提示：添加记录要注意先后顺序，应先输入主表数据，后输入外表数据，并且外键的值必须是主键中已存在的值或空值（如果外键允许为空）。另外，数据不能违反完整性约束。
5. 在 book 表的 ISBN 字段上建立一个名称为 IX\_ISBN 的唯一索引；在 PublisherCode 和 PublishTime 字段上建立一个自命名的普通索引。
6. 更新表记录。将 book 表中 BookCode 字段值为 0503 的记录删除；将 BookCode 字段值为 0202 的记录的 Discount 字段值修改为 0.50，PublisherCode 字段值更新为 55。  
提示：注意观察哪些操作不成功，思考为什么，加深对数据完整性约束作用的理解。

电子工业出版社有限公司  
版权所有