

# 实验 2 数据库表

数据库表（**Table**）是包含数据库中所有数据的数据库对象，创建数据库之后，即可创建数据表。

## 【知识要点】

### 1. 数据库表

数据库表（**Table**）是数据库内最重要的对象，它最主要的功能是存储数据内容。创建数据库之后，即可创建数据表。数据库表存储在数据库文件中，并可以将其存放在指定的文件组上。数据库表是列的集合，每一列都是不可再分的。数据在表中是按行和列的格式组织排列的，每行代表唯一的一条记录，而每列代表记录中的一个数据项。每一列具有相同的域，即有相同的数据类型。**SQL Server** 的每个数据库最多可存储**20**亿个表，每个表可以有 **1024** 列。表的行数及总大小仅受可用存储空间的限制。每行最多可以存储 **8,060** 字节。

### 2. 数据表结构

- 每个数据表至少包含下面内容：
- 数据表名称；
- 数据表中所包含列的列名称，同一表中的列名称不能相同；
- 每列的数据类型；
- 字符数据类型列的长度（字符个数）；
- 每个列的取值是否可以为空（**NULL**）。

### 3. 列的数据类型

类型名称		说明
整型	bigint	存储大小为8 个字节。从-2 <sup>63</sup> (-9223372036854775808) 到2 <sup>63</sup> -1 （9223372036854775807）的整型数据。
	int	存储大小为4 个字节。 从-2 <sup>31</sup> (-2,147,483,648) 到2 <sup>31</sup> - 1 (2,147,483,647) 的整型数据。

# 实验 2 数据库表

	smallint	存储大小为2 个字节。 从 $-2^{15}$ (-32,768) 到 $2^{15} - 1$ (32,767) 的整型数据。
	tinyint	存储大小为1 字节。 从0 到255 的整型数据。
位	bit	整型数据 1、0 或NULL。Microsoft® SQL Server™ 优化用于bit 列的 存储。如果一个表中有不多于8 个的bit 列，这些列将作为一个字节存 储。如果表中有9 到16 个bit 列，这些列将作为两个字节存储。更多 列的情况依此类推。
数 值 型	decimal[(p[,s ])]	带定点精度和小数位数的numeric 数据类型。  $P$ (精度) 指定小数点左边和右边可以存储的十进制数字的最大个数 。精度必须是从1 到最大精度之间的值。最大精度为38。  $S$ (小数位数) 指定小数点右边可以存储的十进制数字的最大个数。 小数位数必须是从0 到 $p$ 之间的值。默认小数位数是0，因而 $0 \leq s \leq p$ 。最大存储大小基于精度而变化。
	Numeric[(p[,s ])]	
货 币 型	money	存储大小为8 个字节。  货币数据值介于 $-2^{63}$ (-922,337,203,685,477.5808) 与 $2^{63} - 1$ (+922,337,203,685,477.5807) 之间，精确到货币单位的千分之十。
	Smallmoney	货币数据值存储大小为4 个字节。  介于 - 214,748.3648 与+214.748,3647 之间，精确到货币单位的千分之十。
数值型	float	用于表示浮点数字数据的近似数字数据类型。 浮点数据为近似值；并非数据类型范围内的所有数据都能精确地表示。
	Real	
字符串型	char	长度为 $n$ 个字节的固定长度且非Unicode 的字符数据。 $n$ 必须 是一个介于1 和8,000 之间的数值。存储大小为 $n$ 个字节。

# 实验 2 数据库表

	varchar	长度为n 个字节的可变长度且非Unicode 的字符数据。 n 必须是一个介于 1 和8,000 之间的数值。存储大小为输入数据的字节的实际长度，而不是n 个字节。所输入的数据字符长度可以为零。
	text	存储大量的可变长度的非 Unicode 文本数据。 其最大长度为 $2^{31}-1$ (2,147,483,647)个字符。
Unicode	nchar (n)	存储包含n 个字符的固定长度Unicode 字符数据。 n 的值必须介于 1 与4,000 之间。存储大小为n 字节的两倍。
	nvarchar (n)	存储n 个字符的可变长度Unicode 字符数据。 n 的值必须介于1 与 4,000 之间。字节的存储大小是所输入字符个数的两倍。 所输入的数据字符长度可以为零。
	ntext	存储大量的可变长度的 Unicode 文本数据。 可变长度Unicode 数据 的最大长度为 $2^{30} - 1$ (1,073,741,823)个字符。 存储大小是所输入字 符个数的两倍（以字节为单位） 。
二进制字符串	Binary (n)	存储固定长度的n 个字节二进制数据。 n 必须从1 到8,000。存储空间大小为n+4 字节。
	varbinary (n)	存储 n 个字节变长二进制数据。 n 必须从1 到8,000。存储空间大小为实际输入数据长度+4 个字节，而不是n 个字节。输入的数据 长度可能为0 字节。
	image	用来存储大量的二进制数据，通常用来存储图形。 可变长度二进制数据介于0 与 $2^{31}-1$ (2,147,483,647) 字节之间。
其它类型	timestamp	这种数据类型表现自动生成的二进制数，确保这些数在数据库中 是唯一的。 timestamp 一般用作给表行加版本戳的机制。 存储大小 为8 字节。
	Unique identifier	全局唯一标识符(GUID)。

# 实验 2 数据库表

---

## 4. 对数据表的操作

- 创建数据表之后，可进行下面操作：
- 修改表结构
- 删除数据表
- 查询数据表中的数据
- 更新（插入、修改、删除）数据表中的数据

## 5. 创建数据表的 T-SQL 语句

CREATE TABLE

[ *database\_name*.[ *owner* ] .| *owner*.] *table\_name*

( { < column\_definition >

| *column\_name* AS *computed\_column\_expression*

| < table\_constraint > ::= [ CONSTRAINT

*constraint\_name* ] }

| [ { PRIMARY KEY | UNIQUE } [ ,...*n* ]

]

[ ON { *filegroup* | DEFAULT } ]

< **column\_definition** > ::= { *column\_name data\_type* }

[ COLLATE < collation\_name > ]

[ [ DEFAULT *constant\_expression* ]

| [ IDENTITY [ ( *seed* , *increment* ) [ NOT FOR REPLICATION ] ] ] ]

[ ROWGUIDCOL]

[ < column\_constraint > ] [ ...*n* ]

< **column\_constraint** > ::= [ CONSTRAINT *constraint\_name* ]

{ [ NULL | NOT NULL ]

| [ { PRIMARY KEY | UNIQUE } [ CLUSTERED | NONCLUSTERED ] [ WITH FILLFACTOR =*fillfactor* ]

[ ON { *filegroup* | DEFAULT } ]

]

# 实验 2 数据库表

---

```
| [ [ FOREIGN KEY ] REFERENCES ref_table [ ( ref_column ) ]  
[ ON DELETE { CASCADE | NO ACTION } ]  
[ ON UPDATE { CASCADE | NO ACTION } ]  
[ NOT FOR REPLICATION ]  
]  
| CHECK [ NOT FOR REPLICATION ] ( logical_expression )  
}  
  
< table_constraint > ::= [ CONSTRAINT constraint_name ]  
{ [ { PRIMARY KEY | UNIQUE } [ CLUSTERED | NONCLUSTERED ]  
{ ( column [ ASC | DESC ] [ ,...n ] ) }  
[ WITH FILLFACTOR =fillfactor ]  
[ ON { filegroup | DEFAULT } ]  
]  
| FOREIGN KEY [ ( column [ ,...n ] ) ]  
REFERENCES ref_table [ ( ref_column [ ,...n ] ) ]  
[ ON DELETE { CASCADE | NO ACTION } ]  
[ ON UPDATE { CASCADE | NO ACTION } ]  
[ NOT FOR REPLICATION ]  
| CHECK [ NOT FOR REPLICATION ] ( search_conditions ) }
```

## 6. 修改数据表结构的 T-SQL 语句

```
ALTER TABLE table  
{ [ ALTER COLUMN column_name  
{ new_data_type [ ( precision [ , scale ] ) ]  
[ COLLATE < collation_name > ]  
[ NULL | NOT NULL ]  
| { ADD | DROP } ROWGUIDCOL }  
]  
| ADD { [ < column_definition > ]  
| column_name AS computed_column_expression  
} [ ,...n ]
```

# 实验 2 数据库表

---

```
| [ WITH CHECK | WITH NOCHECK ] ADD { < table_constraint > } [ ,...n ]  
| DROP { [ CONSTRAINT ] constraint_name  
| COLUMN column  
} [ ,...n ]  
| { CHECK | NOCHECK } CONSTRAINT { ALL | constraint_name [ ,...n ] } | { ENABLE |  
DISABLE } TRIGGER { ALL | trigger_name [ ,...n ] }  
}
```

## 7. 删除数据表的 T-SQL 语句

DROP TABLE table\_name

### 【实验目的】

- 掌握数据表的创建；
- 掌握数据表结构的修改；
- 掌握数据表的删除。

## 实验 2.1 创建数据库表

### 【实验目的】

- 掌握使用企业管理器创建数据库表；
- 掌握使用 SQL 命令创建数据库表；
- 掌握向数据库表中录入数据。

### 【实验内容】

1 . 创建数据库表。

表名分别如下：

教师表：T，学生表：S，课程表：C，选课表：SC，授课表：TC

2 . 各数据表的结构为：

表 1 教师表 T

# 实验 2 数据库表

字段名	数据类型	长度（字节数）	是否可 NULL	中文描述
TNO	char	2	NOT NULL	教师号
TN	char	8	NOT NULL	教师姓名
SEX	char	2	NULL	性别
AGE	tinyint	1	NULL	年龄
PROF	char	10	NULL	职称
SAL	smallint	2	NULL	工资
COMM	smallint	2	NULL	岗位津贴
DEPT	char	10	NULL	系名

表 2 学生表 S

字段名	数据类型	长度（字节数）	是否可 NULL	中文描述
SNO	char	2	NOT NULL	学生号
SN	char	8	NOT NULL	学生姓名
SEX	char	2	NULL	性别
AGE	tinyint	1	NULL	年龄
DEPT	char	10	NULL	系名

表 3 课程表 C

字段名	数据类型	长度（字节数）	是否可 NULL	中文描述
CNO	char	2	NOT NULL	课程号
CN	char	10	NOT NULL	课程名
CT	tinyint	1	NULL	课时数

表 4 选课表 SC

# 实验 2 数据库表

字段名	数据类型	长度（字节数）	是否可 NULL	中文描述
SNO	char	2	NOT NULL	学号
CNO	char	2	NOT NULL	课程号
SCORE	tinyint	1	NULL	成绩

表 5 授课表 TC

字段名	数据类型	长度（字节数）	是否可 NULL	中文描述
TNO	char	2	NOT NULL	教师号
CNO	char	2	NOT NULL	课程号

## 实验 2.2 修改数据库表结构

### 【实验目的】

- 掌握用企业管理器修改已有数据表的结构；
- 掌握用 SQL 修改已有数据表的结构。

### 【实验内容】

#### 1.向已有数据表 S 中追加学籍列

追加的列定义如下：

列名： **NATIVE**

数据类型： **CHAR**

长度： **40**

允许空否： **NOT NULL**

#### 2.修改已有表 S 中的列定义

把表 S 中的 **NATIVE** 列定义修改成下列定义：

列名： **NATIVE**

数据类型： **CHAR**



# 实验 2 数据库表

---

长度：16

允许空否：NULL

3.删除表 S 中的 NATIVE 列

## 实验 2.3 删除数据库表

### 【实验目的】

- 掌握在企业管理器中删除数据表；
- 掌握用 SQL 删除数据表。

### 【实验内容】

- 在企业管理器中删除数据表 test1；
- 用 SQL 删除数据表 test2。

### 【实验题】

#### •实验内容与要求

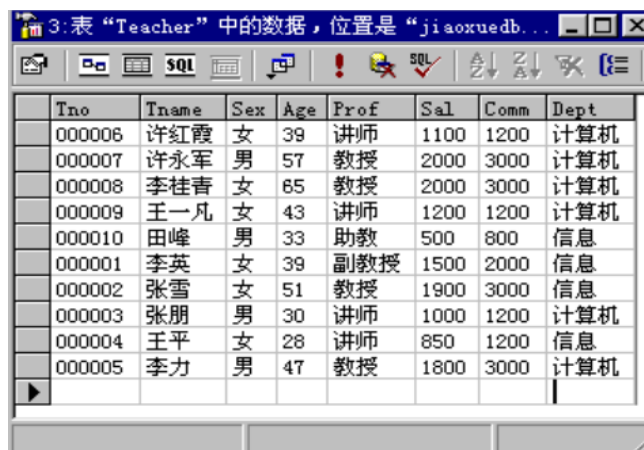
1. 创建数据库 jiaoxuedb。
2. 在数据库 jiaoxuedb 中创建表 Student、Teacher、Course、SC、TC，它们的表数据如下列图所示。

# 实验 2 数据库表



Sno	Sname	Sex	Age	Dept
001101	宋大方	男	19	计算机
002102	李王	男	20	信息
991101	张彬	男	18	计算机
991102	王蕾	女	19	计算机
991103	张建国	男	18	计算机
991104	李平方	男	18	计算机
991201	陈东辉	男	19	计算机
991202	葛鹏	男	21	计算机
991203	潘桃芝	女	19	计算机
991204	姚一峰	男	18	计算机
001102	许辉	女	22	计算机
001201	王一山	男	20	计算机
001202	牛莉	女	19	计算机
002101	李丽丽	女	19	信息

学生表 student



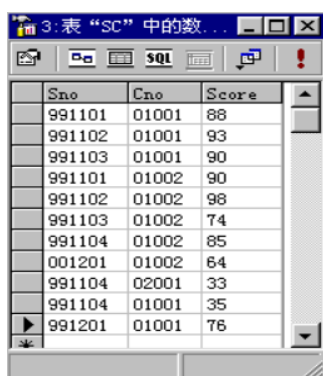
Tno	Tname	Sex	Age	Prof	Sal	Comm	Dept
000006	许红霞	女	39	讲师	1100	1200	计算机
000007	许永军	男	57	教授	2000	3000	计算机
000008	李桂青	女	65	教授	2000	3000	计算机
000009	王一凡	女	43	讲师	1200	1200	计算机
000010	田峰	男	33	助教	500	800	信息
000001	李英	女	39	副教授	1500	2000	信息
000002	张雪	女	51	教授	1900	3000	信息
000003	张朋	男	30	讲师	1000	1200	计算机
000004	王平	女	28	讲师	850	1200	信息
000005	李力	男	47	教授	1800	3000	计算机

教师表 Teacher



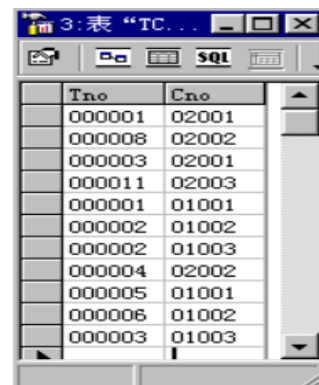
Cno	Cname	Chour	Ccredit
01001	计算机基础	60	3
01002	程序设计	80	5
02003	数据结构	60	6
02001	数据库	80	6
02002	计算机网络	60	6
01003	微机原理	60	8
02004	操作系统	60	6
03001	软件工程	60	3
03002	大型数据库	48	2
03003	图像处理	48	2

课程表 Course



Sno	Cno	Score
991101	01001	88
991102	01001	93
991103	01001	90
991101	01002	90
991102	01002	98
991103	01002	74
991104	01002	85
001201	01002	64
991104	02001	33
991104	01001	35
991201	01001	78

选课表 SC



Tno	Cno
000001	02001
000008	02002
000003	02001
000011	02003
000001	01001
000002	01002
000002	01003
000004	02002
000005	01001
000006	01002
000003	01003

教师任课表 TC

3. 给学生表 Student 增加两个字段，一个是长度为 2000 个字符的字段简历 Resume，另一个是长度为 20 个字符的籍贯 Native。
4. 把学生表中的籍贯列改为 40 个字符。
5. 删除学生表中的简历字段。
6. 删除表 TC。

## 【思考题】

1. 一个数据库中的数据表可以有相同的表名吗？同一个数据表中，列名称可以相同吗？
2. 用 ALTER TABLE 语句可以修改已有的列名或删除已有的列吗？
3. 解释语句 USE jxsk 的含义？给出与此语句功能相同的操作？

# 实验 2 数据库表

---

4. 有下列定义: create table test ( item1 char (10),  
item2 nchar (10),  
item3 nchar (20),  
item4 nvarchar (20) )

- (1) item1 与 item2 所占磁盘空间各是多少字节, 是否相同?
- (2) Item3 与 item4 所占磁盘空间各是多少字节, 是否相同?
- (3) 如果在 item1 对应的数据项中, 输入字符串“12345678901”, 可以吗? 为什么?