

# 专题2

## 数据库创建与管理

2.1 数据库管理工具

2.2 SQL Server的命名规范

2.3 创建数据库

2.4 修改与删除数据库

# 2.1

## 数据库管理工具

- ➡ SQL Server数据库管理系统
- ➡ 数据库存储结构
- ➡ 数据库类型
- ➡ 服务器连接

# 01 SQL Server数据库管理系统

常用管理工具：

- SQL Server Management Studio (SSMS)：是用于访问、配置、管理和开发 SQL Server 各种组件的集成环境。
- SQL Server 配置管理器：是为 SQL Server 服务、服务器协议、客户端协议和客户端别名提供基本配置管理的工具。

## □ SQL

结构化查询语言 (Structured Query Language) 最早由IBM研制的，具有高度非过程化，和面向集合的操作方式等特点，得到所有关系型数据库系统的支持。功能上包括4个部分：

- 数据查询(DQL)
- 数据操作(DML)
- 数据定义(DDL)
- 数据控制(DCL)

## □ T-SQL语言

T-SQL是Transact-SQL结构化查询语言，是SQL Server对SQL的功能扩展。

## 02 数据库存储结构

- 逻辑存储结构：数据库是由哪些性质的信息所组成，如表、索引、视图、约束等各种不同的逻辑成分，这些逻辑成分称为数据库对象。
- 物理存储结构：SQL Server数据库表现为数据库文件，由数据文件和事务日志文件组成，是数据库在物理设备（比如磁盘）上的存储结构与存取方法，它依赖于给定的计算机系统。

## 02 数据库存储结构

### 数据库对象

常用的数据库对象有表、视图、索引、约束、存储过程、触发器等。其中：



表

表由行和列组成，用于组织和存储数据



索引

索引是对数据表中  
一系列或多列的值进  
行排序的一种结构



视图

视图是用户查看数据表中数据的一种方式



存储过程

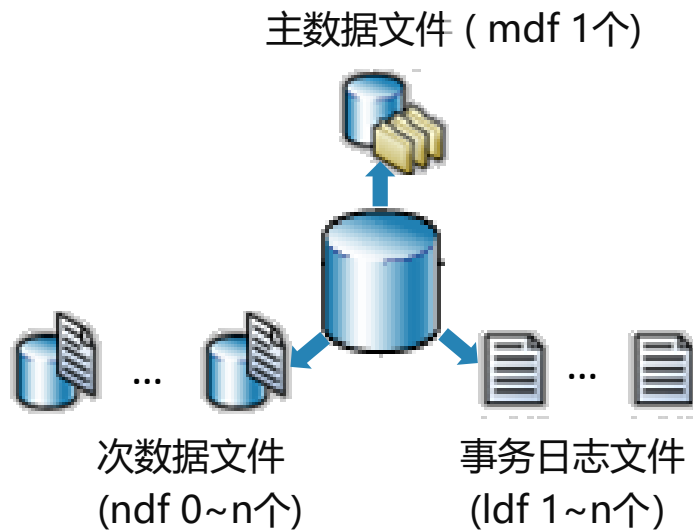
存储过程是一组为完  
成特定功能的已编译  
的SQL语句的集合

## 02 数据库存储结构

### 数据库文件

包括主数据文件、次数据文件、事务日志文件。

- 主数据文件的扩展名为mdf，一个数据库只能有1个主数据文件,用来存储数据库的数据和数据库的启动信息；
- 次数据文件的扩展名为ndf，一个数据库可以0个或多个次数据文件；
- 事务日志文件的扩展名为ldf，用来存放数据库的事务日志，一个数据库可以有1个或多个日志文件



### ◆ 文件组

文件组是数据文件的逻辑组合。

可以有1个主文件组、0个或多个用户定义文件组。

- 主文件组：是系统自动创建的。系统文件、每个数据库的主数据文件和没有被放入其他组的次数据文件分配在这一组。如果没有指定，则主文件组作为默认文件组
- 用户定义文件组：是指创建数据库时使用FILEGROUP关键字指定的文件组。可以定义0个或多个用户定义文件组。

注意：事务日志文件不属于任何文件组



## 03 数据库类型

SQL Server数据库实例包含系统数据库和用户数据库。

SQL Server数据库实例是指安装在计算机上的某个完整的SQL Server服务器。

每个数据库实例有4个系统数据库：

- **master** 数据库记录 SQL Server 系统的所有系统级信息。
- **model**数据库提供了SQL Server所有新建数据库模板。
- **msdb**数据库是“SQL Server代理服务”数据库，为报警、任务调度和作业记录提供存储空间。
- **tempdb**数据库是一个临时数据库，保存临时对象和中间结果集。

## 03 数据库类型

---

SQL Server数据库实例包含系统数据库和用户数据库。

用户数据库是用户创建的数据库，它们是为具体应用而创建的，可以有一个或多个，例如图书管理数据库。

## 04 服务器连接

操作方法：

- 启动SQL Server。
- 在“连接到服务器”对话框中，首先选择服务器类型，默认为数据库引擎；
- 选择服务器名称，一般为本地服务器，即当前计算机名，可以用(LOCAL)替代，如果要连接网络服务器，则输入服务器的IP地址。
- 选择身份验证方式。
- 单击“连接”按钮，进入SSMS窗口，默认打开对象资源管理器。



# 2.2

## SQL Server的命名规范

- ➡ 标识符
- ➡ 标识符类型
- ➡ 数据库对象命名规则

# 01 标识符

数据库对象是用标识符命名的。

标识符的命名规则：

- 首字符为字母、下划线( \_ )、@、 #，后续字符为字母、数字、下划线、@、#、\$所构成的字符串；  
例如： table1、@vb、\_selt是合法标识符  
@ ⇒ 局部变量/参数 如： @day  
# ⇒ 临时表/过程 如： 表#grade
- 字符个数在1~128之间, 不区分大小写；
- 不能有空格、其他特殊字符、SQL的保留字。

## 02 标识符的类型

标识符分为二种类型：

- 常规标识符：是符合标识符的规则标识符
- 分隔标识符：是包含在双引号（“”）或方括号（[]）中的标识符。

例如： Mr GIRL      ⇒    [Mr GIRL]  
         含有空格        分隔标识符

# 03

## 数据库对象的命名规则

数据库对象的完整名称由4部分组成，格式如下：

[[[服务器名.][数据库名] .] [数据库架构名] .] 数据库对象名

数据库架构是一个独立于数据库用户的非重复命名空间。系统默认架构为dbo。

例如：有计算机名为W7-062，数据库名为sems，表名CInfo，以下对象名有效：

[W7-062].sems.dbo.CInfo

--完整名称

.sems.dbo.CInfo

--省略服务器名

sems.dbo.CInfo

--省略服务器名

dbo.CInfo

--省略服务器名、数据库名

CInfo

--省略服务器名、数据库名、架构名

# 2.3

## 创建数据库

- ➡ 界面方式创建数据库
- ➡ 命令方式创建数据库



# 01 界面方式创建数据库

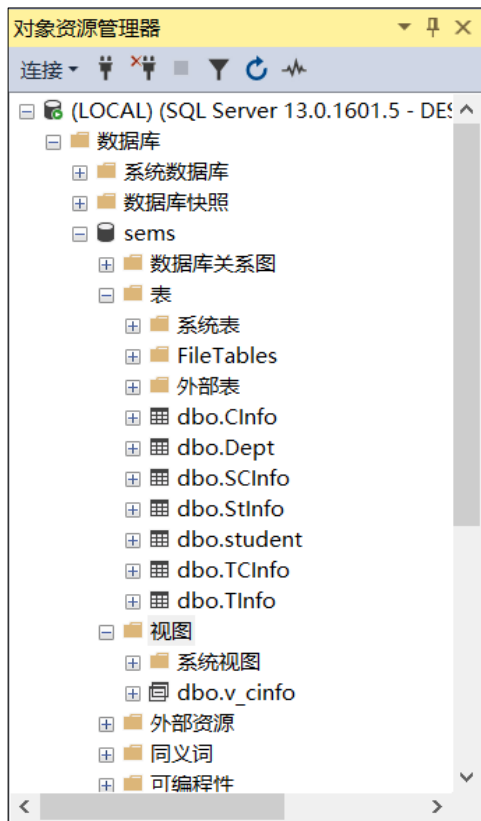
使用对象资源管理器。

## 例1 创建学生选课数据库sems。

设置初始大小为10MB, 按10%增长。

操作步骤如下：

- 在对象资源管理器中，单击数据库的快捷菜单的“新建数据库”命令；
- 在新建数据库对话框的数据库名称栏中，输入sems，初始大小栏输入10。
- 单击自动增长栏右侧按钮，在打开的对话框中，选择文件按百分比增长，单击“确定”按钮返回。其他参数保持默认。
- 单击“确定”按钮，完成数据库创建。



## 02 命令方式创建数据库

创建数据库命令的格式：

```
CREATE DATABASE <数据库名>
```

```
[ ON
```

```
    [ PRIMARY ]
```

```
    [ <文件选项> [ , ... n ] ]
```

```
    [ , FILEGROUP <文件组名> [ , ... n ] ]
```

```
    [ LOG ON <文件选项> [ , ... n ] ]
```

```
]
```

<文件选项> ::=

```
{ (
```

```
    NAME = 逻辑名称 ,
```

```
    FILENAME = { '存储路径' }
```

```
    [ , SIZE = 初始大小 ]
```

```
    [ , MAXSIZE = { 最大容量 | UNLIMITED } ]
```

```
    [ , FILEGROWTH = 增量 [容量 | % ] ]
```

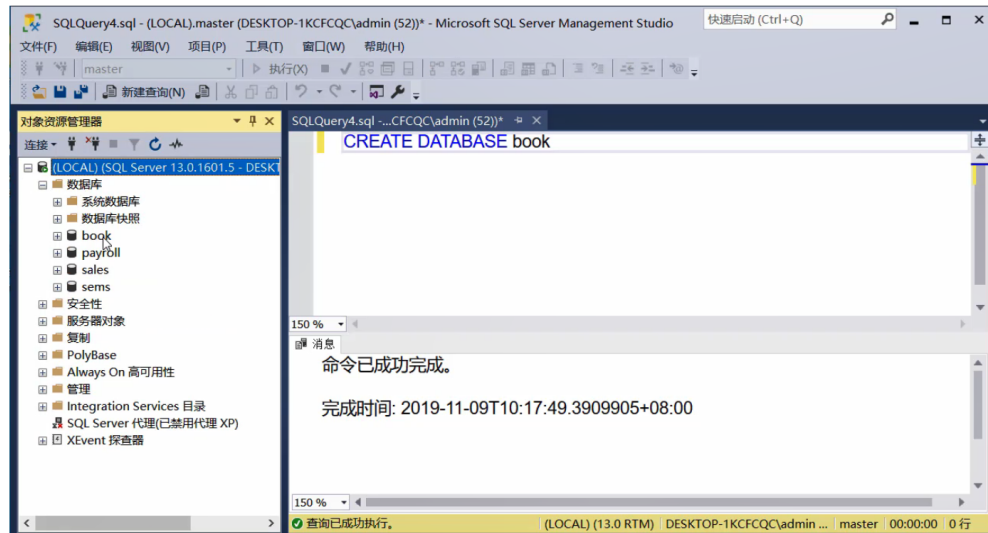
```
)} }
```

## 02 命令方式创建数据库

### 例2 创建图书数据库book。

CREATE DATABASE book

- 在查询编辑窗口输入语句；
- 单击“执行”按钮，在消息窗口中显示命令执行情况。
- 刷新对象资源管理器，显示book数据库。



# 2.4

## 修改与删除数据库

➡ 查看与修改数据库

➡ 删除数据库

# 01 查看和修改数据库

有两种方法：

- 使用界面方式查看和修改数据库
- 使用命令方式查看和修改数据库

**例1** 查看payroll数据库文件，修改主数据文件的初始大小为10MB，在paygp文件组添加数据文件paydat3 。

# 01 查看和修改数据库

## □ 界面方式查看数据库属性

- 在对象资源管理器的属性对话框的“文件”选择页中，显示payroll数据库文件的各项参数；
- 在“文件组”选择页中显示已定义文件组paygp。

# 01 查看和修改数据库

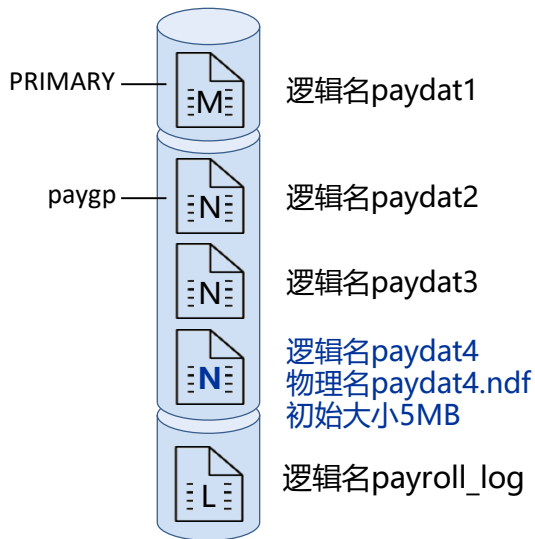
## □ 界面方式修改数据库

- 在paydat1的“初始大小”栏，输入10；
- 单击“添加”按钮添加一行，在“逻辑名称”栏输入paydat3，在“文件组”栏选择paygp，在“路径”栏输入路径。
- 单击“确定”按钮，完成修改。

# 01 查看和修改数据库

## □ 命令方式修改数据库

**例2** 向payroll数据库的paygp文件组添加次数据文件paydat4，文件属性如图所示，存储位置为D:\mydb。





## 01

## 查看和修改数据库

(1) 调用系统存储过程sp\_helpdb查看数据库信息。

语句如下：

```
sp_helpdb payroll
```

| 结果 |             | 消息       |                         |           |            |                |                     |           |
|----|-------------|----------|-------------------------|-----------|------------|----------------|---------------------|-----------|
|    | name        | db_size  | owner                   | dbid      | created    | status         | compatibility_level |           |
| 1  | pay...      | 34.00 MB | DESKTOP-1KCFCQC\admin   | 6         | 11 10 2019 | Status=ONLI... | 130                 |           |
|    | name        | fileid   | filename                | filegroup | size       | maxsize        | growth              | usage     |
| 1  | paydat1     | 1        | D:\mydb\paydat1.mdf     | PRIMARY   | 10240 KB   | Unlimited      | 65536 KB            | data only |
| 2  | payroll_log | 2        | D:\mydb\payroll_log.ldf | NULL      | 8192 KB    | 2147483648 KB  | 65536 KB            | log only  |
| 3  | paydat2     | 3        | D:\mydb\paydat2.ndf     | paygp     | 8192 KB    | Unlimited      | 65536 KB            | data only |
| 4  | paydat3     | 4        | D:\mydb\paydat3.ndf     | paygp     | 8192 KB    | Unlimited      | 65536 KB            | data only |

# 01 查看和修改数据库

(2) 使用ALTER DATABASE命令修改数据库。

语法格式：

```
ALTER DATABASE <数据库名>
{  ADD FILE <文件选项>[, ... n] [TO FILEGROUP 文件组]
  | ADD LOG FILE <文件选项> [, ... n]
  | ADD FILEGROUP <文件组名>
  | REMOVE FILE <逻辑文件名>
  | REMOVE FILEGROUP <文件组名>
  | MODIFY FILE <文件选项>
  | MODIFY NAME= <新数据库名>
  | MODIFY FILEGROUP <文件组名>
}
```

# 01 查看和修改数据库

## ➤ 添加次数据文件

例2中，向payroll数据库的paygp文件组添加次数据文件paydat4，使用以下语句：

```
ALTER DATABASE payroll
ADD FILE
( NAME=paydat4,
  FILENAME='D:\mydb\paydat4.ndf',
  SIZE=5 )
TO FILEGROUP paygp
```

|   | name        | fi... | filename                | filegroup | size     | maxsize       | growth   | usage     |
|---|-------------|-------|-------------------------|-----------|----------|---------------|----------|-----------|
| 1 | paydat1     | 1     | D:\mydb\paydat1.mdf     | PRIMARY   | 10240 KB | Unlimited     | 65536 KB | data only |
| 2 | payroll_log | 2     | D:\mydb\payroll_log.ldf | NULL      | 8192 KB  | 2147483648 KB | 65536 KB | log only  |
| 3 | paydat2     | 3     | D:\mydb\paydat2.ndf     | paygp     | 8192 KB  | Unlimited     | 65536 KB | data only |
| 4 | paydat3     | 4     | D:\mydb\paydat3.ndf     | paygp     | 8192 KB  | Unlimited     | 65536 KB | data only |
| 5 | paydat4     | 5     | D:\mydb\paydat4.ndf     | paygp     | 5120 KB  | Unlimited     | 65536 KB | data only |

# 01 查看和修改数据库

## ➤ 删除数据库中的数据文件

**例3** 删除sales数据库的次数据文件saledat2。

|   | name      | fileid | filename              | filegroup | size     | maxsize       | growth   | usage     |
|---|-----------|--------|-----------------------|-----------|----------|---------------|----------|-----------|
| 1 | saledat1  | 1      | D:\mydb\saledat1.mdf  | PRIMARY   | 20480 KB | Unlimited     | 5120 KB  | data only |
| 2 | sales_log | 2      | D:\mydb\sales_log.ldf | NULL      | 10240 KB | 2147483648 KB | 10%      | log only  |
| 3 | saledat2  | 3      | D:\mydb\saledat2.ndf  | PRIMARY   | 8192 KB  | Unlimited     | 65536 KB | data only |

使用以下语句：

```
ALTER DATABASE sales
```

```
REMOVE FILE saledat2
```

# 01 查看和修改数据库

## ➤ 修改数据库文件的属性

**例4** 将sales数据库的主数据文件最大容量修改为100MB。

| 逻辑名称      | 文件类型 | 文件组     | 大小(MB) | 自动增长/最大大小                 | ... | 路径      | 文件名           |
|-----------|------|---------|--------|---------------------------|-----|---------|---------------|
| saledat1  | 行数据  | PRIMARY | 20     | 增量为 64 MB, 增长无限制          | ... | D:\mydb | saledat1.mdf  |
| sales_log | 日志   | 不适用     | 10     | 增量为 64 MB, 限制为 2097152 MB | ... | D:\mydb | sales_log.ldf |

| 逻辑名称      | 文件类型 | 文件组     | 大小(MB) | 自动增长/最大大小                 | ... | 路径      | 文件名           |
|-----------|------|---------|--------|---------------------------|-----|---------|---------------|
| saledat1  | 行数据  | PRIMARY | 20     | 增量为 64 MB, 限制为 100 MB     | ... | D:\mydb | saledat1.mdf  |
| sales_log | 日志   | 不适用     | 10     | 增量为 64 MB, 限制为 2097152 MB | ... | D:\mydb | sales_log.ldf |

使用以下语句：

```
ALTER DATABASE sales
MODIFY FILE
( NAME = saledat1,
  MAXSIZE=100 )
```

# 01

## 查看和修改数据库

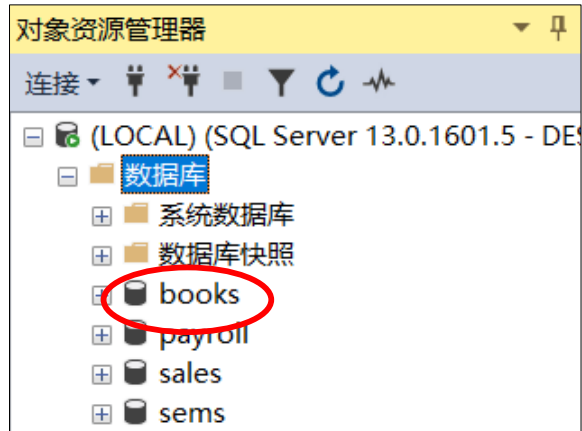
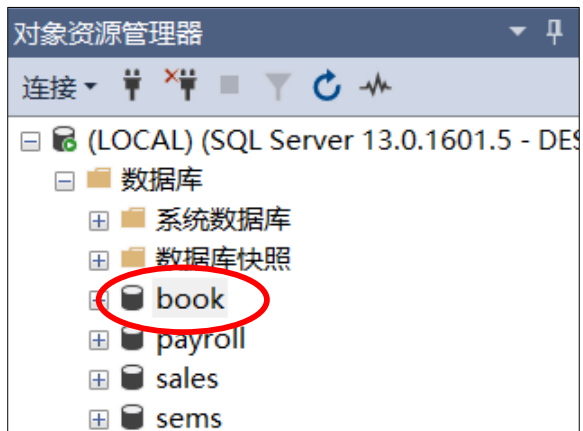
- 修改数据库的名称

**例5** 将book数据库更名为books。

命令如下：

```
ALTER DATABASE book
```

```
MODIFY NAME= books
```



# 01 查看和修改数据库

---

对于已存在的数据库可以进行以下修改：

- 增加或删除数据库文件或文件组；
- 改变数据库文件的大小和增长方式；
- 重命名数据库，但不能修改数据文件名和日志文件名。

## 02 删除数据库

□ 使用对象资源管理器删除数据库

**例6** 删除books数据库。

操作步骤如下：

- 在对象资源管理器中，选择books数据库；
- 单击快捷菜单的“删除”命令；
- 勾选“删除对象”窗口的“关闭现有连接”复选框；
- 单击“确定”按钮，完成删除操作。



## 02 删除数据库

□ 使用DROP DATABASE命令删除数据库。

格式如下：

```
DROP DATABASE <数据库名> [ , ...n ]
```

**例7** 删除例3中的sales数据库。

```
DROP DATABASE sales
```

**例8** 删除已有的test1和test2数据库。

```
DROP DATABASE test1, test2
```

**注意：**

- 数据库一旦被删除，数据库对象与数据库文件都将被删除。
- 使用DROP命令删除时不会出现确认信息，使用时要慎重。

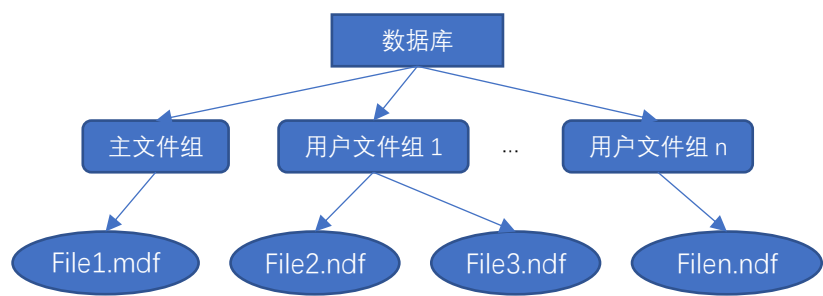
# SQL Server 文件和文件组

数据库是数据的仓库，用于存储数据，数据通常存储在硬盘上。SQL Server 将数据库映射为一组操作系统文件。数据库最常用的存储文件是数据文件和事务日志文件，数据文件用于存储数据，由一个主数据文件（.mdf）和若干个次数据文件（.ndf）构成；事务日志文件（.ldf）用于存储事务日志。不同的文件可以存分布到不同的物理硬盘上，这样便于分散硬盘输入输出，提高数据的读取速度。

数据文件的集合，称作文件组，数据库不能直接设置存储数据的数据文件，而是通过文件组来指定。

## 一、文件组文件的作用

SQL Server 的数据存储在文件中，文件是实际存储数据的物理实体，文件组是逻辑对象，SQL Server 通过文件组来管理文件。



一个数据库有一个或多个文件组，主文件组是系统自动创建的，用户可以根据需要添加文件组。每一个文件组管理一个或多个文件，其中主文件组中包含主数据文件，这个文件是系统默认生成的，并且在数据库中是唯一的；次数据文件是是通过在 **CREATE DATABASE** 或 **ALTER DATABASE** 语句中使用 **FILEGROUP** 关键字指定的任何文件组，即用户根据需要添加的。主文件组中也可以包含次数据文件，除了主文件组之外，其他文件组只能包含次数据文件。

例如，查看示例数据库的文件组，PRIMARY 是主文件组，勾选“默认值”表示主文件组是默认文件组，这意味着，如果在 **CREATE TABLE** 和 **CREATE INDEX** 中没有指定 **FILEGROUP** 选项，那么 SQL Server 将使用默认文件组来存储数据。

| 名称      | 文件 | 只读                       | 默认值                                 | 自动增长所有文件                 |
|---------|----|--------------------------|-------------------------------------|--------------------------|
| PRIMARY | 1  |                          | <input checked="" type="checkbox"/> | <input type="checkbox"/> |
| stgp    | 1  | <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/> |

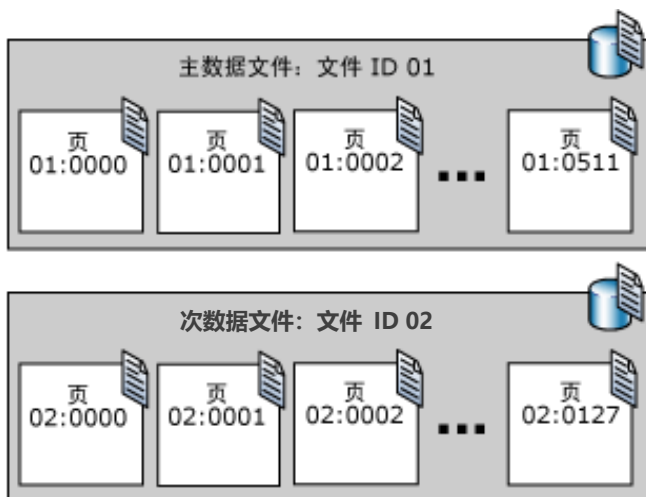
文件组是一个逻辑实体，实际上，数据存储和数据文件（.mdf 和.ndf）中，每一个文件组中都包含文件，因此，在 **CREATE TABLE** 和 **CREATE INDEX** 命令中指定文件组，那么数据就会存储到文件组包含的文件中。

| 数据库文件(F): |      |         |        |                         |                  |                  |
|-----------|------|---------|--------|-------------------------|------------------|------------------|
| 逻辑名称      | 文件类型 | 文件组     | 大小(MB) | 自动增长/最大大小               | 路径               | 文件名              |
| SEMS      | 行数据  | PRIMARY | 4      | 增量为 1 MB, 增长无限制         | ... E:\sems\data | SEMS.mdf         |
| SEMS_log  | 日志   | 不适用     | 4      | 增量为 10%, 限制为 2097152 MB | ... E:\sems\data | SEMS_log.ldf     |
| semsdat1  | 行数据  | stgp    | 8      | 增量为 64 MB, 增长无限制        | ... D:\semsdb    | ... semsdat1.ndf |

从上图中可以看到，数据库文件的元数据：

- ◆ 逻辑名称：是在所有 T-SQL 语句中引用物理文件时所使用的名称，在数据库中的逻辑文件名中必须是唯一的。
- ◆ 操作系统文件名：是包括目录路径的物理文件名，必须符合操作系统文件命名规则，逻辑名称和物理名可以是不同的。
- ◆ 数据库文件有两种类型：行数据（存储数据）和日志（存储日志）。
- ◆ 初始大小：是文件创建时的大小。
- ◆ 增量：表示文件自动增加的大小。
- ◆ 最大大小：是文件大小的最大值。
- ◆ 路径：表示文件的路径。

数据文件是按文件页（页的大小为 8KB）存储的。SQL Server 数据文件中的页按顺序编号，文件的首页以 0 开始。数据库中的每个文件都有一个唯一的文件 ID 号。若要唯一标识数据库中的页，需要同时使用文件 ID 和页码。下面图例显示了包含 4MB 主数据文件和 1MB 次数据文件在数据库中的页码。



每个文件的第一页是一个包含有关文件属性信息的文件的页首页。在文件开始处的其他几页也包含系统信息（例如分配映射）。

每个文件还可以指定一个最大大小。如果没有指定最大大小，文件可以一直增长到用完磁盘上的所有可用空间。如果 **SQL Server** 作为数据库嵌入某应用程序，而该应用程序的用户无法迅速与系统管理员联系，则此功能就特别有用。用户可以使文件根据需要自动增长，以减轻监视数据库中的可用空间和手动分配额外空间的管理负担。

## 二、使用文件组的优势

在实际开发数据库的过程中，通常情况下，用户需要关注文件组，而不用关心文件的物理存储，即使 **DBA** 改变文件的物理存储，用户也不会察觉到，也不会影响数据库去执行查询。除了逻辑文件和物理文件的分离之外，**SQL Server** 使用文件组还有一个优势，那就是分散输入输出负载，其实现的原理是：

- 对于单分区表，数据只能存到一个文件组中。如果把文件组内的数据文件分布在不同的物理硬盘上，那么 **SQL Server** 能同时从不同的物理硬盘上读写数据，把输入输出负载分散到不同的硬盘上。
- 对于多分区表，每个分区使用一个文件组，把不同的数据子集存储在不同的磁盘上，**SQL Server** 在读写某一个分组的数据时，能够调用不同的硬盘输入输出。

这两种方式，其本质上，都是使每个硬盘均摊系统负载，提高输入输出性能。在设计文件组时，应尽量把包含的文件存放在不同的硬盘上，以实现物理 **IO** 的最大分散化。

还有，应该根据硬盘的性能来创建文件组，日志文件存储到性能最好的硬盘上，而查询延迟要求高的数据，也需要存储到性能最好的硬盘上。

不是所有的数据都是同等重要的，应该根据业务需求和查询延迟，对数据分级，因此，在设计文件组时，应该把级别高的数据分散，而把那些基本用不到的数据存储到性能差的，用于存储归档数据的硬盘上，以实现服务器性能的合理配置。

## 三、为表指定文件组

在创建表时，在 **on** 子句中指定文件组，那么数据将存储在该文件组包含的文件中：

```
CREATE TABLE [dbo].[student](  
...  
) ON [PRIMARY]
```

在创建索引时，在 **on** 子句指定文件组选项，那么该表的索引结构将存储在文件组包含的文件中：

```
CREATE NONCLUSTERED INDEX [idx_student_stuname]  
ON [dbo].[student]([stuname] ASC )  
ON [PRIMARY]
```

在创建分区时，在 **TO** 子句中指定文件组，每一个分区将存储到文件组中：

```
REATE PARTITION SCHEME scheme_name  
AS PARTITION function_name  
TO ([fg_name1], <....>, [fp_nameN])
```

## 四、数据文件大小增长导致的问题

当数据文件没有空间存储数据时，执行 **INSERT** 命令，会导致数据文件的生长。如果 **FILEGOWTH** 选项设置的过大，会导致 **SQL Server** 耗费较长时间来实现文件的生长，在数据文件生长时，该文件是不能访问的，因此，即使用户仅插入一条数据，也要等待很长时间才能完成查询，对用户来说，体验不友好。

数据文件生长是非常耗费系统资源和影响性能，如果设置 **SQL Server** 自动增长，可能会导致系统性能不够稳定，所以，应该预测可能的空间使用需求，并提前做好规划。尽量避免空间用尽而使得 **SQL Server** 不得不自动生长的现象发生。同时也要确保每一次自动增长都能够在可接受的时间内完成，及时满足客户端应用的需求。

### 参考文献

[1]数据库文件和文件组. <https://wenku.baidu.com/view/bf3857ff534de518964bcf84b9d528ea81c72fae.html>