

索引和视图是数据库常用的对象。索引是为了快速地从数据库中找到所需要的数据,SQL Server 提供了类似于图书的目录作用的索引技术,用户可以使用索引技术在大量数据里快速查询。视图是为了更方便地服务于应用程序,SQL Server 提供了数据库的三级模式中外模式的对象——视图,用户可以简化所使用数据表。

本章主要介绍索引和视图的概念以及在 SQL Server 2012 数据库系统中索引和视图的基本操作。

5.1 索引

在数据库中,索引的含义与日常意义上的“索引”一词并无多大区别,如查字典时使用的索引。索引是用于提高数据库数据访问速度的数据库对象。

5.1.1 索引的基本概念

在关系数据库中,索引(Index)是单独地、物理地对数据库表中一列或多列的值进行排序的一种存储数据结构,它是某个表中一列或若干列值的集合和相应的指向表中物理标识这些值的数据页的逻辑指针清单。索引是数据库中一种特殊类型的对象,它与数据库中的表有着紧密的关系。

数据库索引的作用相当于图书的目录,可以根据目录中的页码快速找到所需的内容。一本书中,利用目录可以快速查找所需内容,而无须翻阅整本书。在数据库中,索引使数据库程序无须对整个表进行扫描,就可以在其中找到所需数据。书中的目录是一个标题列表,其中注明了各章节标题所对应的页码;而数据库中的索引是一个表中所包含的关键字对应的值的列表,其中注明了表中关键字和所包含的各个值在数据库表中对应物理行所在的存储位置,如图 5.1 所示。

当数据库表中有大量记录时,对表进行查询有两种方式:第一种方式是对全表进行搜索,将所有记录一一取出,和查询条件进行一一对比,然后返回满足条件的记录,这样做会消耗大量数据库系统时间,并造成大量的磁盘 I/O 操作;第二种方式是在表中建立索引,然后先在索引中根据关键字找到符合查询条件的索引值,再通过保存在索引中的数据库表中数据的相应物理位置所处的页码,快速找到数据表中所对应的记录。从图 5.1 中可以看到,索引可以避免对数据库表的全表扫描,一些查询可以仅在索引页中扫描少量索引页及数据页,而不是遍历数据库表中所有的数据页。但注意,并不是所有的数据库表建索引就检索快,若是检索表的数据记录比较少,则即使不用索引也可以一次性把所有数据读取出来,只需要做

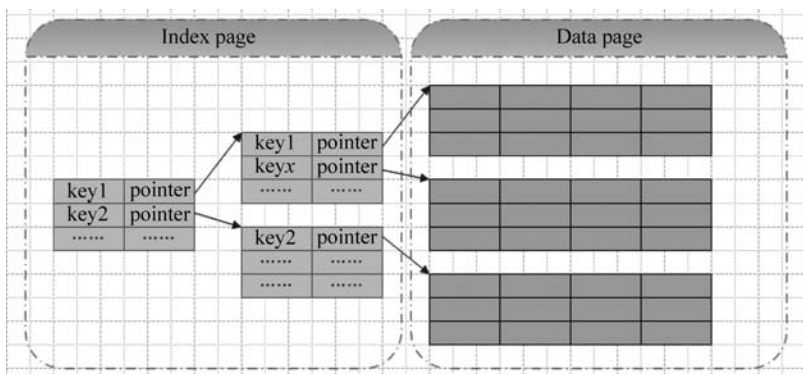


图 5.1 数据库索引与数据表的示意图

一次读写操作；如果使用索引，则首先检索索引，至少读取一次索引数据，再根据索引检索结果去读取相应的数据，再读取一次表数据，至少要做两次读写操作，这显然比不用索引至少多读写一次，这样用索引检索数据就不快了。

一般来说，创建索引可以大大提高系统的性能，主要表现在以下五方面。

(1) 通过创建索引，可以大大加快数据的检索速度，这是创建索引最主要的原因。

(2) 通过创建唯一性索引，可以保证数据库表中每行数据的唯一性。

(3) 可以加速表和表之间的连接，在实现数据的参照完整性方面特别有意义。

(4) 在使用分组和排序子句进行数据检索时，利用索引可以减少排序和分组的时间。

(5) 在数据查询的过程中，SQL Server 都会对查询语句进行优化。查询优化器如果发现所查数据已建索引，一般来说会根据索引优化所建立的查询执行路径，它将决定选择哪些索引可以使得该查询速度最快。

例如，从“学院”表中提取“学院名称”为“计算机学院”的“学院电话”和“学院地址”，则可使用下面的命令。

```
SELECT 学院名称,学院电话,学院地址
FROM 学院
WHERE 学院名称 = '计算机学院'
```

如果在“学院名称”那列上没有索引，那么 SQL Server 就可能对数据库中的数据表进行全表扫描，对表中的数据一行一行地查询，观察数据库表中每行的“学院名称”列的内容。为了找出满足检索条件的那些行，必须访问表中的每一行。对于数据量大的数据表来说，表的检索可能要花费数分钟甚至数小时。

如果在“学院名称”列上创建了索引，就可以提高 SQL Server 查询所需数据的速度。SQL Server 首先搜索针对“学院名称”列所建的索引，找到关键字“学院名称”为“计算机学院”的值，然后根据索引中的物理位置信息确定其在数据表中的物理页和行。由于索引是进行了排序和分类的，并且索引的行和列的数据比较少，所以对索引全部搜索一遍很快，这样就加速了数据的检索。

在数据库中建立索引会提高检索或查找的效率，但这并不是说表中的每个字段都需要建立索引，因为增删记录时除了对表中的数据进行处理外，还需要对每个索引进行维护，索引将额外占用磁盘空间，并且会降低增加、删除和修改的速度。在通常情况下，只对表中经

常查询的字段才创建索引。

5.1.2 索引的分类

数据库索引是数据库管理系统中一个已排序的数据结构,协助快速查询、更新数据库表中的数据。索引的实现通常使用 B+ 树或 B- 树及其变种,根据索引关键字的顺序与创建索引的数据表的物理顺序是否相同,索引可分为聚集(簇)索引(Clustered Index)和非聚集(非簇)索引(Nonclustered Index)。非聚集索引和聚集索引相比,聚集索引有着更快的检索速度。

1. 聚集索引

聚集索引是指数据库表中数据的行物理顺序与索引关键字值的逻辑顺序相同。在创建聚集索引时,一般要重新组织数据库表中的数据,这些数据要按指定的一个或多个列的值排序。聚集索引的叶结点也包含实际的数据,因此用它查找数据很快,但每个表只能建一个聚集索引,因为一个表的物理顺序只有一种情况。在聚集索引中,叶结点即数据结点,所有数据行的存储顺序与索引关键字值的顺序一致。如图 5.2 所示,该例子是以学生的姓名作为索引关键字,在索引中只描述了姓名和数据所在的页码,该例子中的索引是一个二级索引。一般使用聚集索引的场合有以下三种。

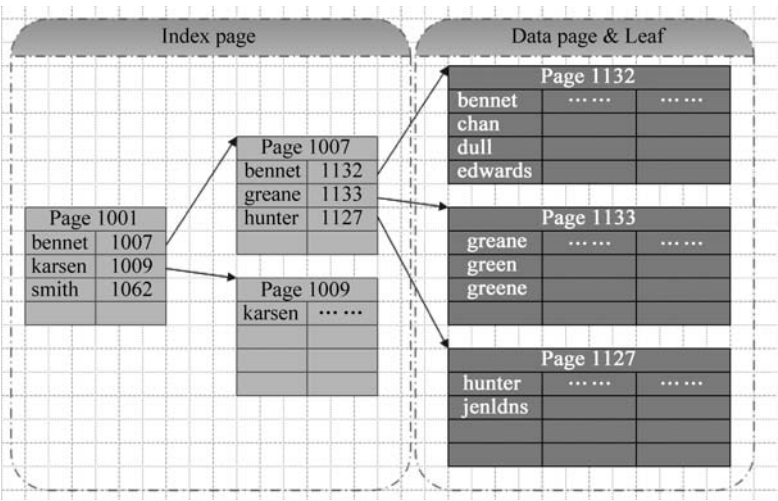


图 5.2 聚集索引(二级索引)示意图

- (1) 此列包含有限数目的不同值。
- (2) 所查询的结果返回为一个区间的值。
- (3) 所查询的结果返回某个值相同的大量结果集。

2. 非聚集索引

非聚集索引是指数据库表中的行物理顺序与索引关键字的值的逻辑顺序不匹配,如图 5.3 所示。聚集索引和非聚集索引都采用了 B+ 树或 B- 树的结构,但非聚集索引的叶子层并不与实际的数据页相重叠,而采用叶子层只包含指针,这个指针指向数据表中的记录所在的数据页。非聚集索引比聚集索引层次多,添加记录不会引起数据顺序的重组。该例子是以学生的姓名作为索引关键字,是一个多级索引结构,前面索引结构中不仅有数据的物

理页码,还有下一层索引页码,在叶子层中的指针直接指向数据的物理页码。一般使用非聚集索引的场合有以下三种。

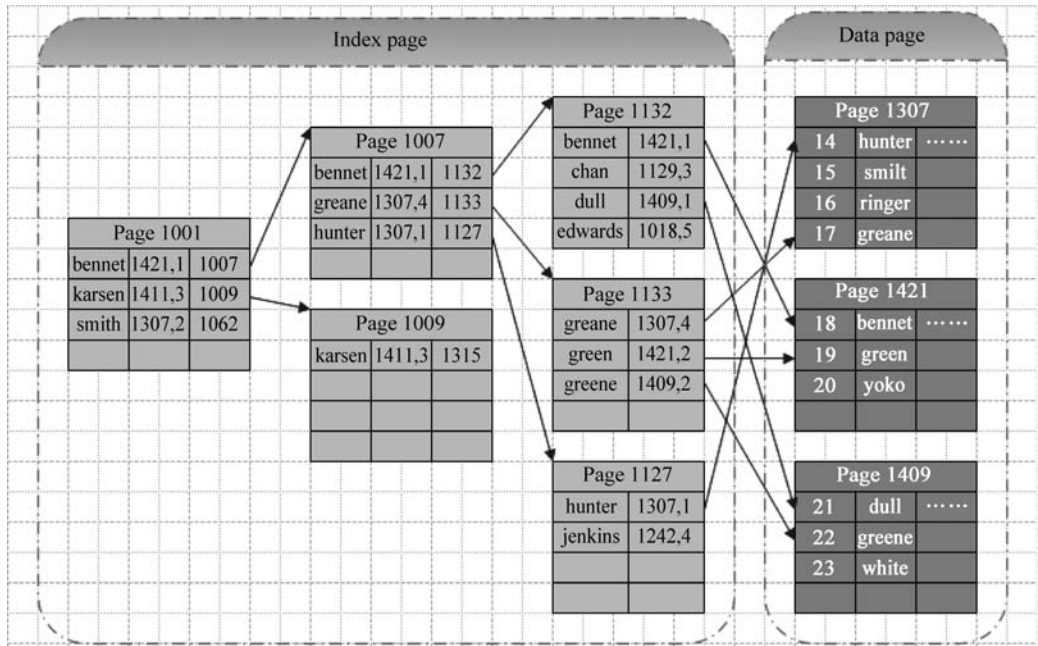


图 5.3 非聚集索引(三级索引)示意图

- (1) 此列包含了大量数目不同的值。
- (2) 所查询的结果返回的是少量的结果集。
- (3) order by 子句中使用了该列。

根据数据库的功能,在 SQL Server 2012 中可创建四种类型的索引,即主键索引、唯一索引、聚集索引和多关键字索引。下面将介绍如何创建索引。

5.1.3 创建索引

在 SQL Server 2012 中,可以在表或视图上创建索引。一般在设计数据库时,在向数据库表中插入数据之前创建索引。SQL Server 提供了 CREATE INDEX 语句来创建索引,该语句的语法格式如下:

```
CREATE [UNIQUE] [CLUSTERED | NONCLUSTERED] INDEX index_name ON [database_name.] table_or_view
_name(column[ASC|DESC][,...n])
```

各选项的含义如下。

- (1) UNIQUE: 为表或视图创建唯一索引。唯一索引不允许两行具有相同的索引键值。聚集索引必须唯一。
- (2) CLUSTERED: 创建聚集索引。创建聚集索引时,键值的逻辑顺序决定表中对应行的物理顺序。聚集索引的底层(或称叶级别)包含该表的实际数据行。一个表或视图只允许同时有一个聚集索引。如果没有指定 CLUSTERED,则创建非聚集索引。
- (3) NONCLUSTERED: 创建一个非聚集索引。对于非聚集索引,数据行的物理排序

独立于索引排序。

(4) index_name: 索引的名称。索引名称在表或视图中必须唯一,但在数据库中不必唯一。索引名称必须符合标识符的命名规则。

(5) database_name: 数据库的名称。

(6) column: 索引所基于的一列或多列。指定两个或多个列名,可为指定列的组合值创建组合索引,即多关键字索引。在 table_or_view_name 后的括号中,按排序优先级列出组合索引中要包括的列。

(7) [ASC|DESC]: 确定索引列的升序或降序排序方向,默认值为 ASC。

下面利用这个索引语句建立不同类型的索引,一般是主键索引和非主键索引、唯一索引和非唯一索引、聚集索引和非聚集索引、单关键字索引和多关键字索引。

1. 主键索引

主键索引是唯一索引的特殊类型。对于数据库表来说,在数据库定义表时,一般会定义一个主键,主键值要求具有唯一性。SQL Server 2012 在创建表时,将自动为表的主键创建主键索引,索引的名字由系统自动产生,其形式为 PK__表名。主键索引一般也是聚集索引,如图 5.4 所示。

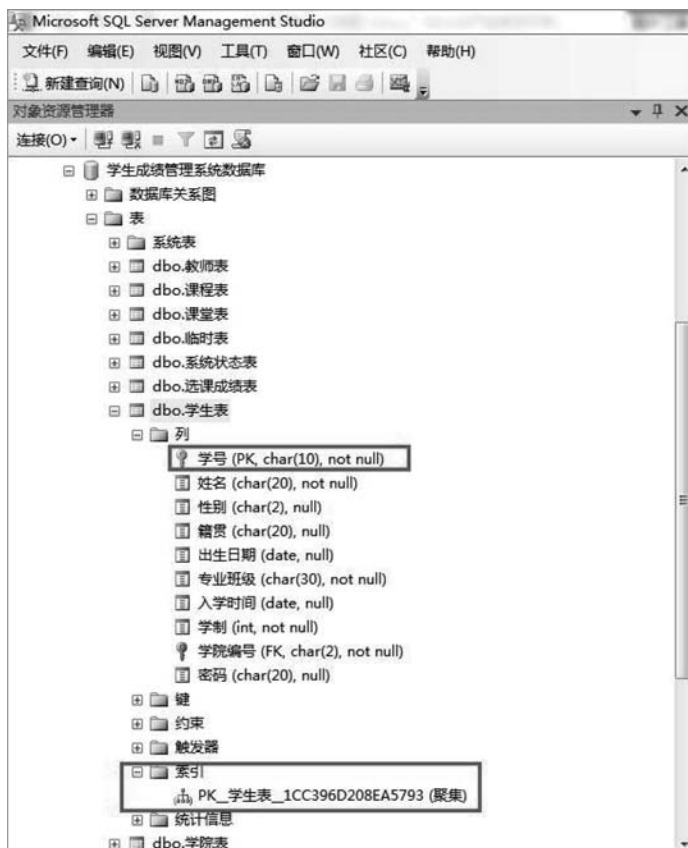


图 5.4 主键索引示意图

2. 唯一索引

在表中建立唯一索引时,一般是对除了主键以外的字段建立唯一索引,组成该索引的字段或字段组合在表中具有唯一值,也就是说,对于表中的任何两行记录来说,索引关键字的

值都各不相同。

如果表中一行以上的记录在索引关键字上具有相同的值,则不能建立唯一索引。如果表中的一个字段或多个字段的组合在有多行记录中具有 NULL 值,则不能建立唯一索引。

例如,如果在一个学生信息表中包含学号和姓名等相关字段,一般学号会作为主关键字,系统会自动创建相应的主键索引,因为学号肯定唯一。如果对学生信息表中的姓名字段创建唯一索引,则认为任何两个学生就不可能同姓名,但现实中学生还是有同名同姓的,因此姓名字段不适合建立唯一索引。

例 5-1 在学院表中,除了学院编号字段是唯一的,学院名称字段也具有唯一值,而且在查找数据时,用户更容易用学院名称作为关键字来查找,因此对学院表中的学院名称字段建立唯一索引。创建唯一索引的命令为

```
CREATE UNIQUE INDEX SCHOOLNAME ON 学院(学院名称)
```

查看已建唯一索引的步骤:单击“dbo.学院”表下属的索引,如果创建索引成功,则会看到刚才创建的索引 SCHOOLNAME,如图 5.5 所示。

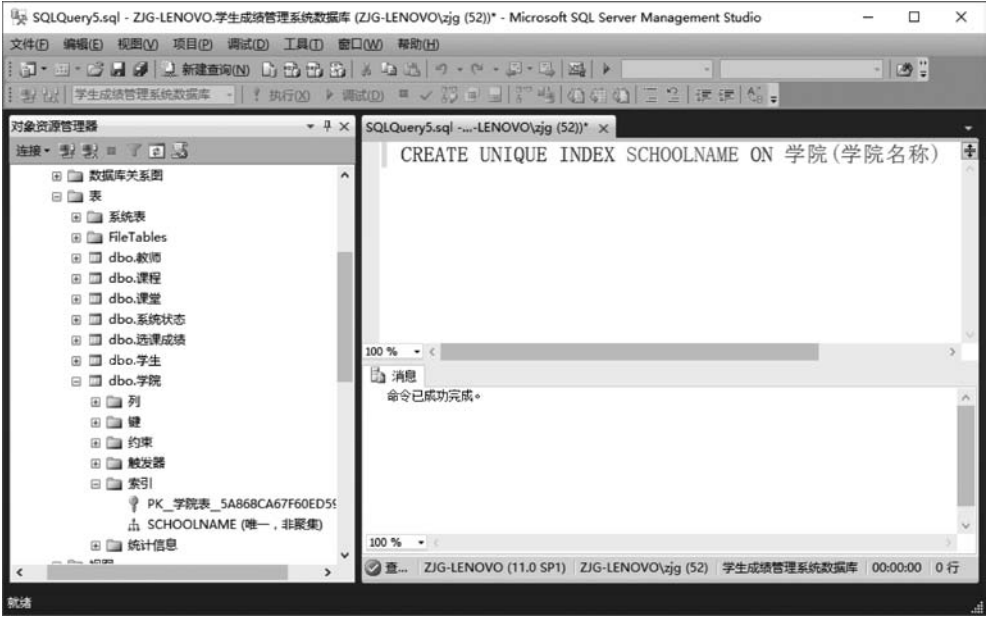


图 5.5 唯一索引创建示意图

双击 SCHOOLNAME 索引,则会弹出有关索引的具体信息(包括表名、索引名称、索引类型、索引键列),如图 5.6 所示。

3. 聚集索引

除了主键索引是聚集索引外,一般用户创建的索引在没有特别说明时均为非聚集索引。如果需要创建聚集索引,可用带有 CLUSTERED 的语句创建索引。

例 5-2 学院表中已存在一个主键,在学院名称字段上再创建聚集索引,看看结果如何。

执行如下 SQL 语句。

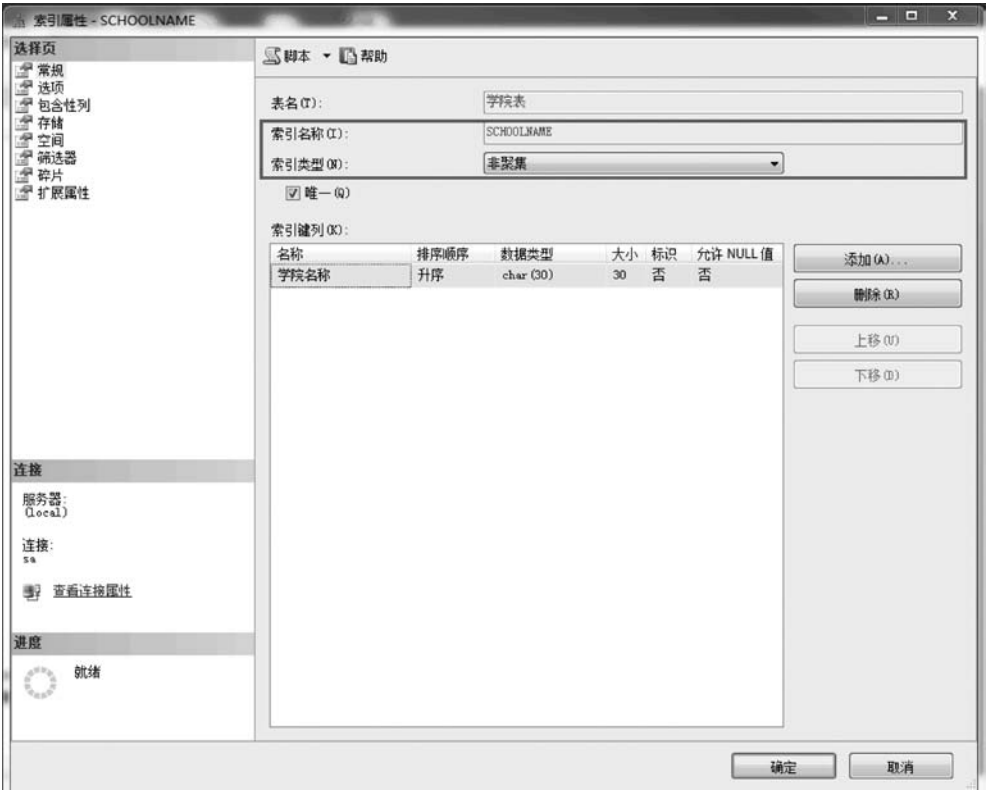


图 5.6 查看索引的具体信息

```
CREATE UNIQUE CLUSTERED INDEX 学院 ON 学院(学院名称)
```

说明：一个表中只能建立一个聚集索引,如果再创建第二个聚集索引时,将提示无法创建的信息,如图 5.7 所示。因此,建立第二个聚集索引时,需要删除表中已经存在的聚集索引。

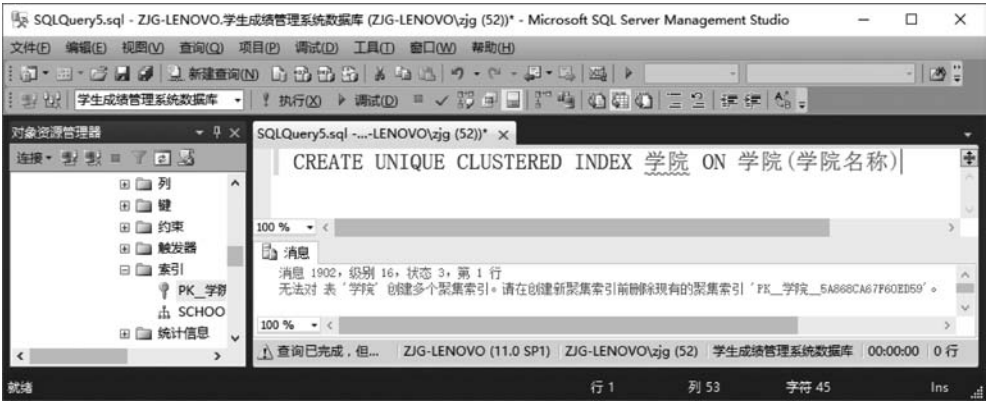


图 5.7 建立多个聚集索引时的系统提示

由于已经存在的聚集索引名为“PK__学院__5A868CA67F60ED59”,所以无法对一个表再建立聚集索引。

4. 多关键字索引

在数据库查询中,经常要把几个字段合在一起作为关键字来查询,为了提高查询效率可以建立多关键字的复合索引。

例 5-3 在学院表中,如果会经常对学院名称、学院地址和学院电话进行查询,可以建立以学院名称、学院地址和学院电话为关键字的索引,如图 5.8 所示。SQL 命令为

```
CREATE NONCLUSTERED INDEX SCHOOL ON 学院(学院名称, 学院地址, 学院电话)
```

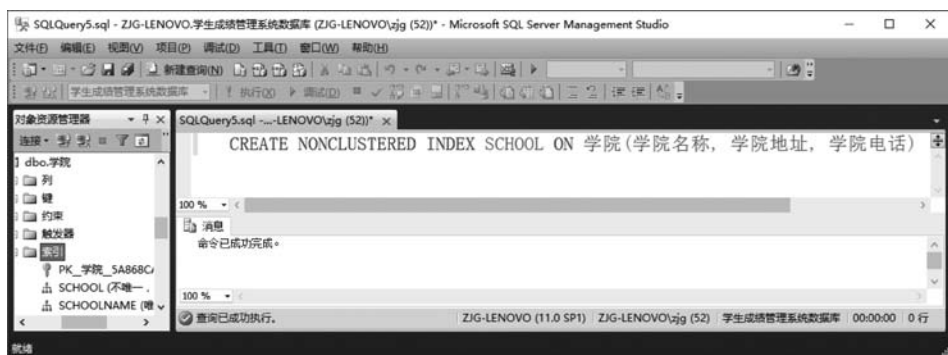


图 5.8 创建多关键字索引示意图

5.1.4 管理和使用索引

在表中创建索引后,一般数据库管理系统自动管理和使用索引,可以通过管理平台来查看索引,也可通过 Transact-SQL 语句来查看索引。

1. 用 SQL Server 管理平台查看修改索引

在 SQL Server 管理平台中选择数据库,展开要查看索引的表对象,选择展开“索引”选项将会列出该表的所有索引,如图 5.9 所示。

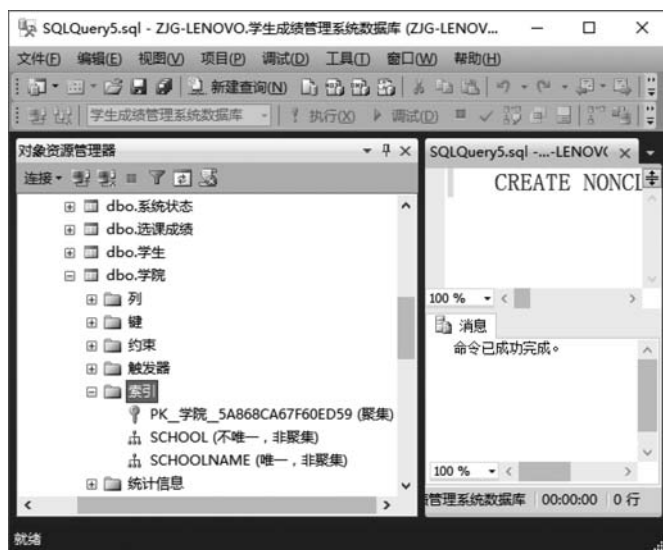


图 5.9 查看索引

然后在要查看的索引上右击,在弹出的快捷菜单中选择“属性”选项,将弹出如图 5.10 所示的“索引属性”窗口,在此窗口中可以查看、修改索引的相关属性。但是要注意的是,在该对话框中不能修改索引的名称,修改索引名称需要使用系统存储过程 sp_rename。例如,拟把学院表中索引名为 SCHOOL 的索引更名为 SCHOOL_add_tel 的语句如下:

```
use 学生成绩管理系统数据库
go
sp_rename '学院.SCHOOL', 'SCHOOL_add_tel'
```

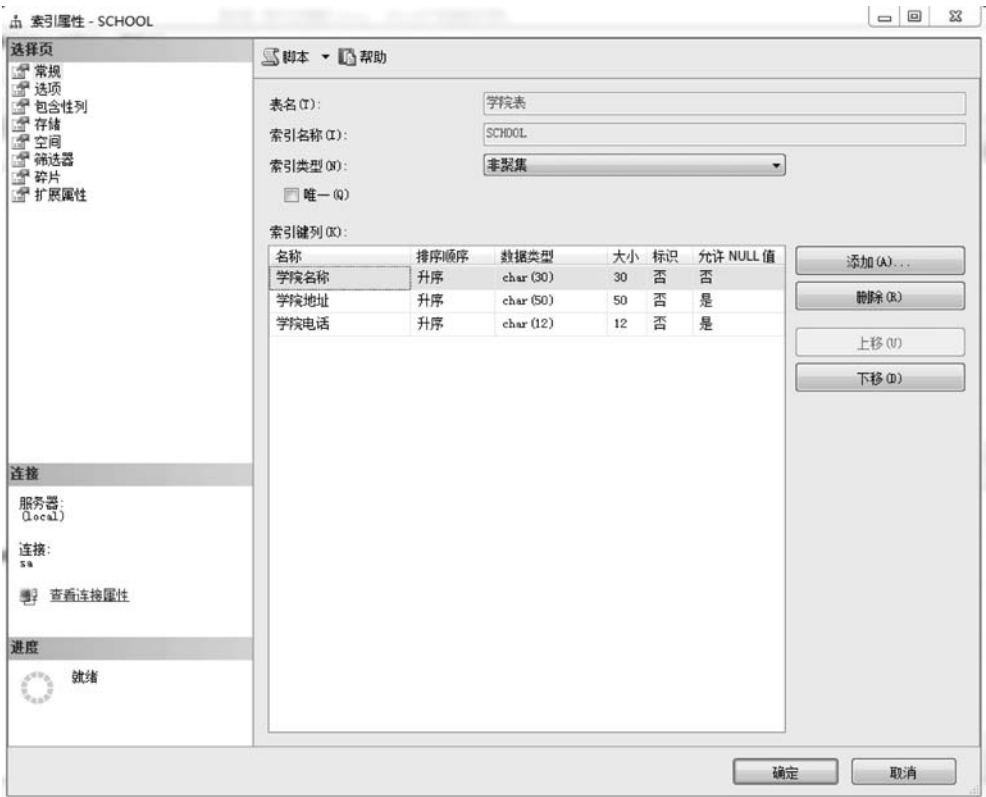


图 5.10 索引属性

注意: 在原来的索引名前一定要有表名作为前缀,以便找到相应的索引名。
若要在管理平台中更改索引名称,如图 5.11 所示,在快捷菜单中选择“重命名”选项。

2. 使用系统存储过程查看索引

sp_helpindex 系统存储过程可以查看表中所有索引的信息,其语法格式如下:

```
sp_helpindex 表名
```

例 5-4 查看学院表的索引,其操作为

```
use 学生成绩管理系统数据库
go
sp_helpindex 学院
```

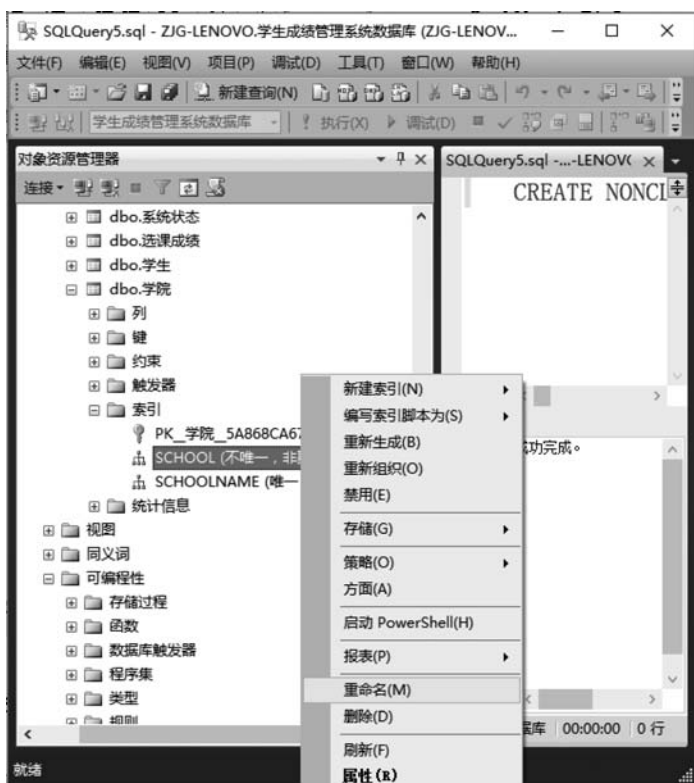


图 5.11 索引重命名

5.1.5 删除索引

索引可能会减慢 INSERT、UPDATE 和 DELETE 语句的执行速度,如果发现索引阻碍整体性能或不再需要索引,则可将其删除。

1. 使用 SQL Server 管理平台删除索引

在 SQL Server 管理平台中,可以从如图 5.12 所示的图中选择要删除的索引并右击,在弹出的快捷菜单中选择“删除”选项来删除索引。

2. 使用 Transact-SQL 语句删除索引

SQL Server 2012 的删除索引语句的语法格式如下:

```
DROP INDEX index_name[,...n]
ON[database_name.[schema_name.]]table_or_view_name
```

各选项的含义如下所述。

- (1) index_name: 要删除的索引名称。
- (2) database_name: 数据库的名称。
- (3) schema_name: 该表或视图所属模式的名称。
- (4) table_or_view_name: 与该索引关联的表或视图的名称。

删除索引时要注意以下四点。

- (1) 执行 DROP INDEX 后,SQL Server 将重新获得以前由索引占用的空间,此后可将

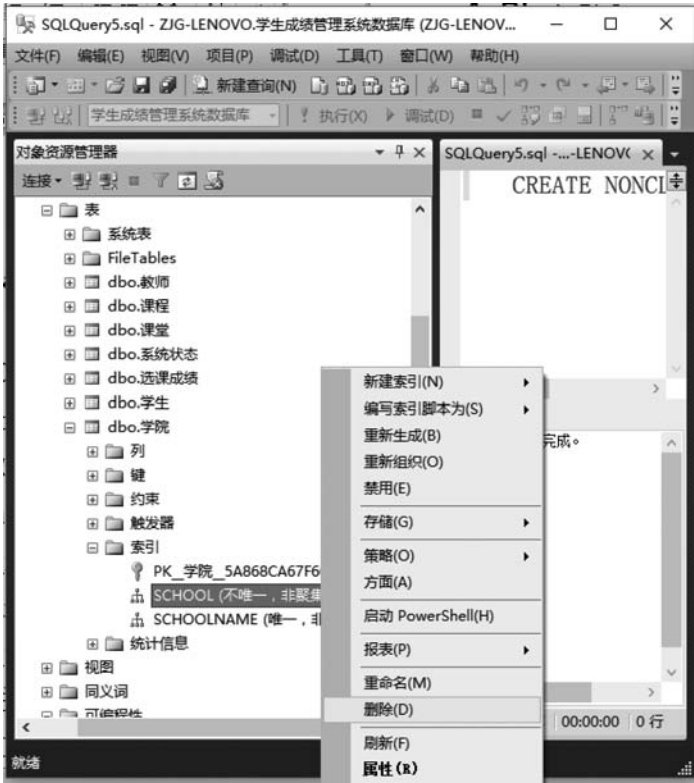


图 5.12 删除索引

- 该空间用于任何数据库对象。
- (2) 不适用于通过定义 PRIMARY KEY 或 UNIQUE 约束创建的索引。若要删除该约束和相应的索引,应使用带有 DROP CONSTRAINT 子句的 ALTER TABLE 语句。
 - (3) 删除视图或表时,将自动删除视图或表创建的索引。
 - (4) 删除索引视图的聚集索引时,将自动删除同一视图的所有非聚集索引和自动创建的统计信息。

例 5-5 删除学院表内名为 SCHOOL 的索引,执行如下 SQL 语句即可。

```
use 学生成绩管理系统数据库
go
IF EXISTS (SELECT name FROM sysindexes WHERE name = 'SCHOOL')
DROP INDEX SCHOOL ON 学院
```

5.2 视图

视图是关系数据库中提供给用户以多种角度观察数据库中数据的重要机制。用户通过视图来浏览表中感兴趣的数据,而数据的物理存放位置仍在原来表中。一般在开发应用系统时,如果某应用程序模块所需的数据在数据库的多个表中,或是数据库一个表中的部分数据,则会创建相关的数据库视图,也称为子模式。

5.2.1 视图的基本概念

视图(View)是从一个或多个表(或视图)导出的特殊的表,视图的作用相当于对数据库中的数据进行筛选。视图与表(也称为基表)不同,视图是一个虚拟表,即视图所对应的数据不额外占用实际物理存储空间,仍然存储在原来的表所在的物理空间中,数据库中只存储视图的定义,即建立视图的 SQL 语句。在对视图的数据进行操作时,系统根据视图的定义去操作与视图相关联的基表。

对视图数据的操作与对表的操作一样,可以对其进行查询、插入、修改和删除,但对数据的这些操作对于视图来说是有限制的。当对视图的数据进行修改时,与其相关的基表的数据也会发生变化;同样,若基表的数据发生变化,也会自动地反映到视图数据中。

视图通常用来集中、简化和自定义每个用户程序对数据库的不同理解,以及每个应用程序对数据的具体需求。视图可用于对数据的保护,也是数据的一种安全机制,允许用户通过视图访问数据,而不授予用户直接访问视图基表的权限。视图的作用主要表现在以下三方面。

1. 定制特定数据

视图使用户能够着重于他们所感兴趣的特定数据和所负责的特定任务,不必要的数据或敏感数据可以不出现在视图中。例如,在学生成绩管理系统中,各位教师只关心自己所讲授的课程和本课堂的学生数据,学生也只关心自己所选课程的数据,而不关注和自己不相干的数据。这样,可以根据实际情况,专门为不同的用户创建不同的视图,其视图就是他所关心的特定数据,以后他在查询数据时,只需要从视图中查询即可。

2. 简化操作

视图可以简化用户设计查询语句的方式。当设计的查询语句涉及聚合函数,同时还要显示其他字段的信息时,可能还会需要关联其他表。这样,查询语句可能会很长,设计也比较难。这时,也可以创建一些视图,简化查询语句。

视图可以简化用户操作数据的方式,可将经常使用的连接、投影、联合查询和选择查询定义为视图,这样,用户每次对特定的数据执行进一步操作时,不必指定全部条件和限定。

例如,一个用于报表目的的应用,并执行子查询、外连接及联合,以便从一组表中检索数据的复合查询,就可以创建一个视图。视图简化了对该报表数据的访问,因为每次生成报表时无须提交基表的查询,而是查询视图中的数据。

3. 安全性

通过视图,用户只能查询和修改他们所能见到的数据,数据库中的其他数据则既看不见也取不到。数据库授权命令可以使每个用户对数据库的检索限制到特定的数据库对象上,但不能授权到数据库特定行和特定列上。通过视图,用户可以被限制在数据的不同子集上。

- (1) 使用权限可被限制在基表的行的子集上。
- (2) 使用权限可被限制在基表的列的子集上。
- (3) 使用权限可被限制在基表的行和列的子集上。
- (4) 使用权限可被限制在多个基表的连接所限定的行上。
- (5) 使用权限可被限制在基表中的数据的统计汇总上。

(6) 使用权限可被限制在另一视图的一个子集上,或是一些视图和基表合并后的子集上。

5.2.2 视图的创建

数据库视图的创建可以通过 SQL Server 管理平台和 Transact-SQL 语句实现,通过 Transact-SQL 语句创建视图,其语句的语法结构如下:

```
CREATE VIEW[ schema_name]view_name[ (column[,...n])]  
AS  
select_statement  
[WITH CHECK OPTION]
```

各选项的含义如下所述。

(1) schema_name: 视图所属架构的名称。

(2) view_name: 视图的名称。视图名称必须符合有关标识符的命名规则,可以选择是否指定视图所有者名称。

(3) select_statement: 定义视图的 SELECT 语句。该语句可以使用多个表和其他视图,利用 SELECT 命令从表中或视图中选择构成新视图的列。

(4) WITH CHECK OPTION: 主要是针对通过视图修改的数据都必须符合 select_statement 中所设置的条件。

一般在创建数据库视图后,一旦使用该视图,系统会自动将满足视图 SQL 语句的对应基表中的数据导入相应内存中。

1. 为查询特定数据创建视图

例 5-6 如果经常查询课程的名称、学分、学时和课程性质,而不需要表中的其他数据,则只关注这些数据,为这些特定的数据建立一个视图。

视图创建语句为

```
CREATE VIEW 课程基本信息  
AS  
SELECT 课程名称,学时数,学分数,课程性质 FROM 课程 ;
```

具体查看视图,选择“学生成绩管理系统数据库”→“视图”选项,显示已经成功创建的视图——课程基本信息,如图 5.13 所示。

在所选中的视图上单击鼠标右键,选择“编辑前 200 行”选项,则显示视图下的数据如图 5.14 所示。

例 5-7 在学生成绩管理系统数据库中创建学分大于或等于 3 分的相关课程信息视图,该视图选择一个基表(课程)中的数据来显示学分大于或等于 3 分的虚拟表。

可执行如下 SQL 语句。

```
CREATE VIEW 大于或等于 3 学分课程信息  
AS  
SELECT * FROM 课程 WHERE 学分数>= 3
```

例 5-8 在学生成绩管理系统数据库中创建“C++ 程序设计基础”课程的学生成绩信息视图,该视图选择四个基表(学生、课程、课堂和选课成绩)中的数据来显示相关信息的虚拟

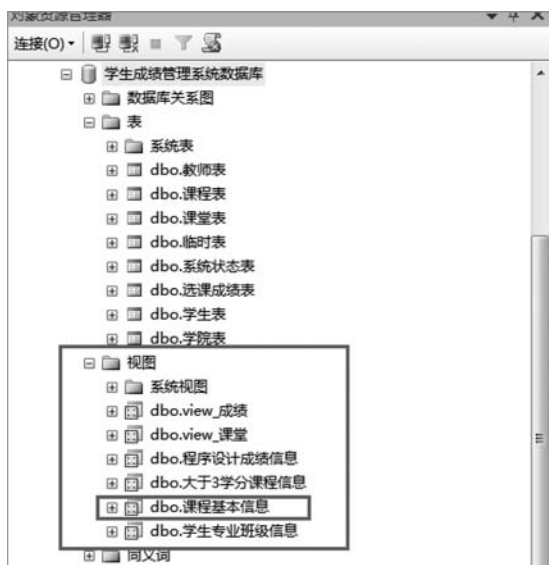


图 5.13 查看视图

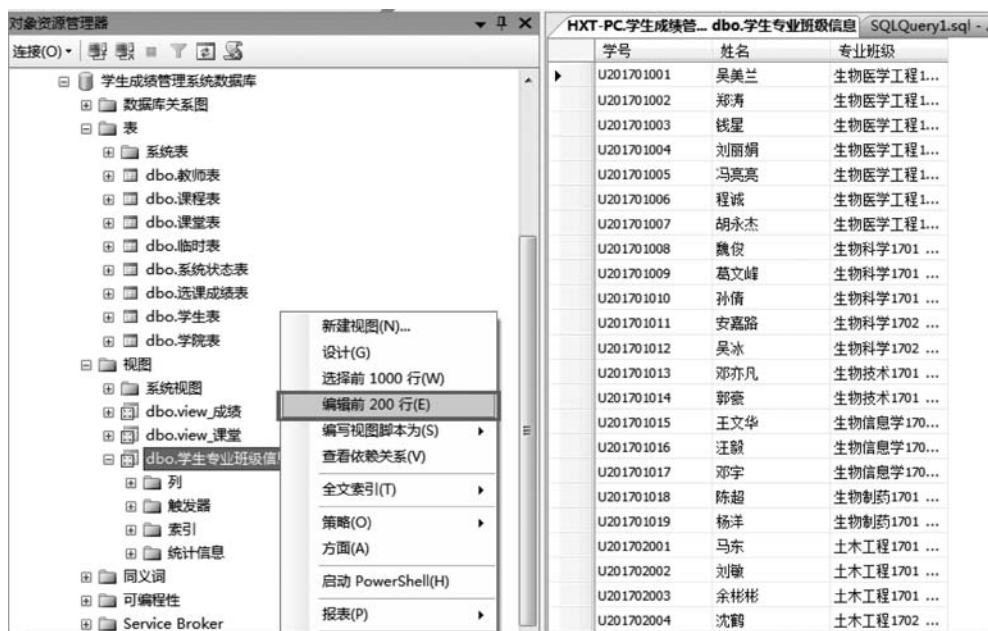


图 5.14 查看数据

表。如图 5.15 所示,执行如下 SQL 语句。

```
CREATE VIEW 程序设计成绩信息
AS
SELECT 学生.学号,姓名,专业班级,课程名称,成绩
FROM 学生 INNER JOIN 选课成绩 ON 学生.学号 = 选课成绩.学号
INNER JOIN 课堂 ON 选课成绩.课堂编号 = 课堂.课堂编号
INNER JOIN 课程 ON 课堂.课程编号 = 课程.课程编号
```

WHERE 课程名称 = 'C++ 程序设计基础'

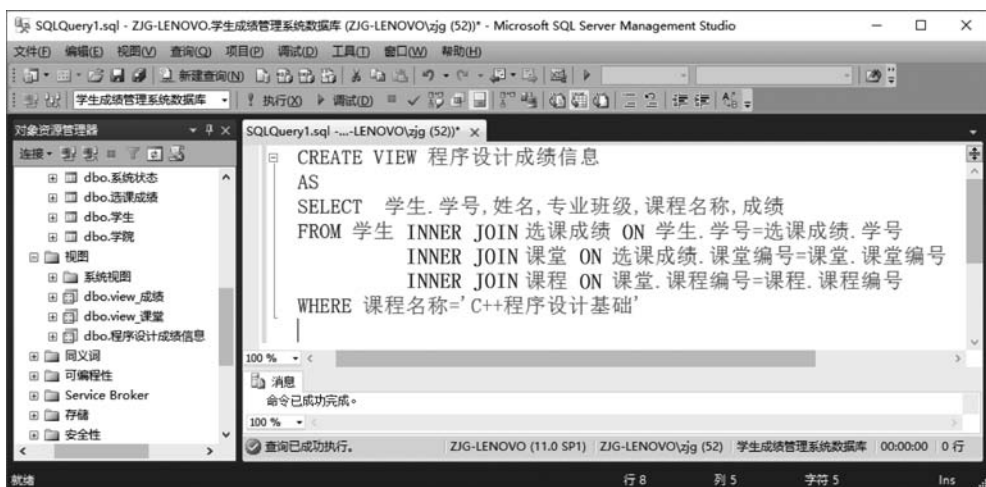


图 5.15 多表建视图示意图

2. 为简化 SQL 语句创建视图

例 5-9 查询土木工程、工程力学两个专业的学生在 2017—2018 学年均选修过的必修课程,要求显示课程编号、课程名称,并按课程编号的升序排列。

SQL 语句如下:

```
SELECT 课程.课程编号, 课程.课程名称
FROM 课程 INNER JOIN 课堂 ON 课程.课程编号 = 课堂.课程编号
      INNER JOIN 选课成绩 ON 课堂.课堂编号 = 选课成绩.课堂编号
      INNER JOIN 学生 ON 选课成绩.学号 = 学生.学号
WHERE 课堂.开课年份 = '2017-2018'
      AND 课程.课程性质 = '必修'
      AND 学生.专业班级 LIKE '土木工程 %'
INTERSECT
SELECT 课程.课程编号, 课程.课程名称
FROM 课程 INNER JOIN 课堂 ON 课程.课程编号 = 课堂.课程编号
      INNER JOIN 选课成绩 ON 课堂.课堂编号 = 选课成绩.课堂编号
      INNER JOIN 学生 ON 选课成绩.学号 = 学生.学号
WHERE 课堂.开课年份 = '2017-2018'
      AND 课程.课程性质 = '必修'
      AND 学生.专业班级 LIKE '工程力学 %'
ORDER BY 课程.课程编号;
```

如果想简化这个语句,可以利用视图,步骤如下。

1) 建立视图

```
CREATE VIEW 土木工程和工程力学选课信息
AS
SELECT 课程.课程编号, 课程.课程名称
FROM 课程 INNER JOIN 课堂 ON 课程.课程编号 = 课堂.课程编号
      INNER JOIN 选课成绩 ON 课堂.课堂编号 = 选课成绩.课堂编号
      INNER JOIN 学生 ON 选课成绩.学号 = 学生.学号
```

```

WHERE 课堂.开课年份 = '2017-2018'
      AND 课程.课程性质 = '必修'
      AND 学生.专业班级 LIKE '土木工程%'
INTERSECT
SELECT 课程.课程编号, 课程.课程名称
FROM 课程 INNER JOIN 课堂 ON 课程.课程编号 = 课堂.课程编号
      INNER JOIN 选课成绩 ON 课堂.课堂编号 = 选课成绩.课堂编号
      INNER JOIN 学生 ON 选课成绩.学号 = 学生.学号
WHERE 课堂.开课年份 = '2017-2018'
      AND 课程.课程性质 = '必修'
      AND 学生.专业班级 LIKE '工程力学%'

```

2) 对该视图查询并排序

```

SELECT * FROM 土木工程和工程力学选课信息
order by 课程.课程编号;

```

3. 为数据安全创建视图

例 5-10 如果经常要查询学生的学号、姓名和所在的专业班级,并且不希望学生表中的其他个人隐私信息被一般人查询,则可为这些特定的数据建立一个视图。

视图创建语句为

```

CREATE VIEW 学生专业班级信息
AS
SELECT 学号,姓名,专业班级 FROM 学生;

```

4. 用 WITH CHECK OPTION 创建视图

在例 5-7 中建立的视图是不带 WITH CHECK OPTION 的视图,如图 5.16 和图 5.17 所示。

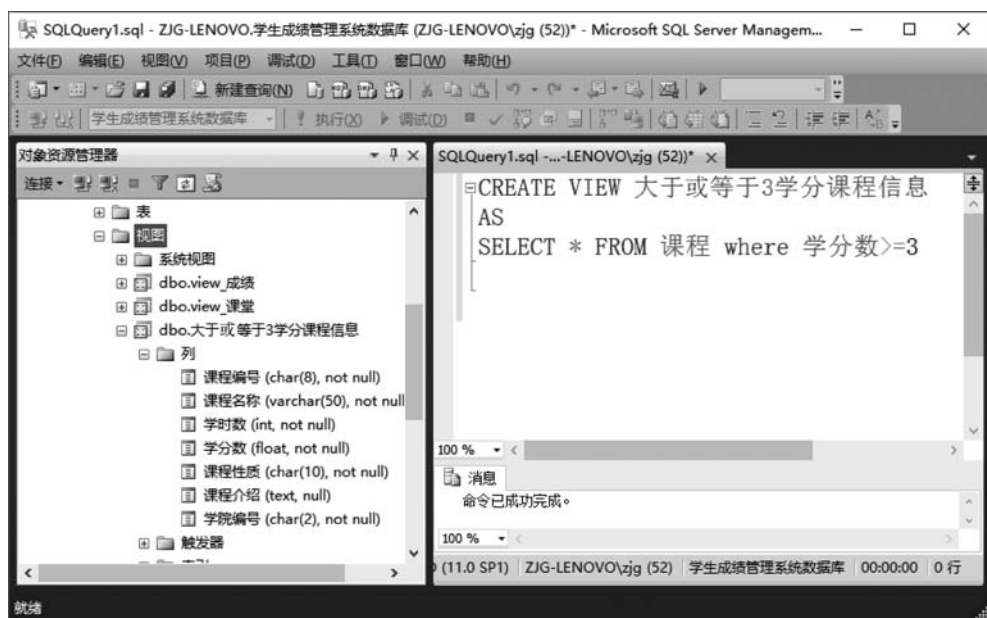


图 5.16 不带 WITH CHECK OPTION 创建视图示意图

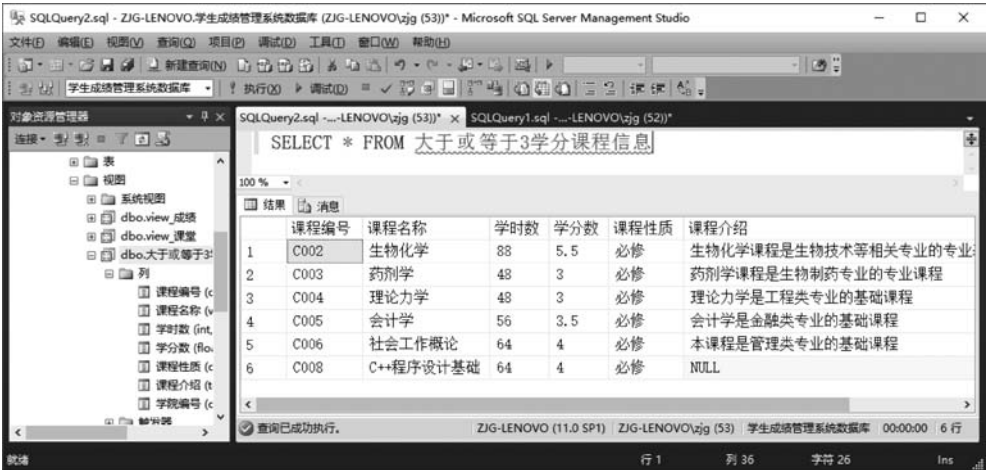


图 5.17 不带 WITH CHECK OPTION 创建视图的数据展示图

请思考：如果要对学分数小于 3 分的课程信息通过视图插入相关的记录,是否能插入成功？根据当前计算机技术的发展,需要增加大数据技术相关课程,需要插入大数据课程的相关信息,语句如下：

```
INSERT INTO 大于或等于 3 学分课程信息(课程编号, 课程名称, 学时数, 学分数, 课程性质, 课程介绍, 学院编号)
VALUES ('0016', '大数据基础', 32, 2, '选修', '本课程意在普及大数据知识, 帮助学生理解大数据时代的现实意义, 了解大数据的处理流程, 以及大数据采集、存储、分析、处理和管理的技术, 以积极投身于大数据的应用.', '06')
```

如图 5.18 所示,对于插入小于 3 学分的相关课程信息,其插入语句能成功执行,即视图的条件没有限制所插入的数据。不过,在视图中没有检索到所插入的相关信息,但在课程中可以检索到相关信息,如图 5.19 所示。

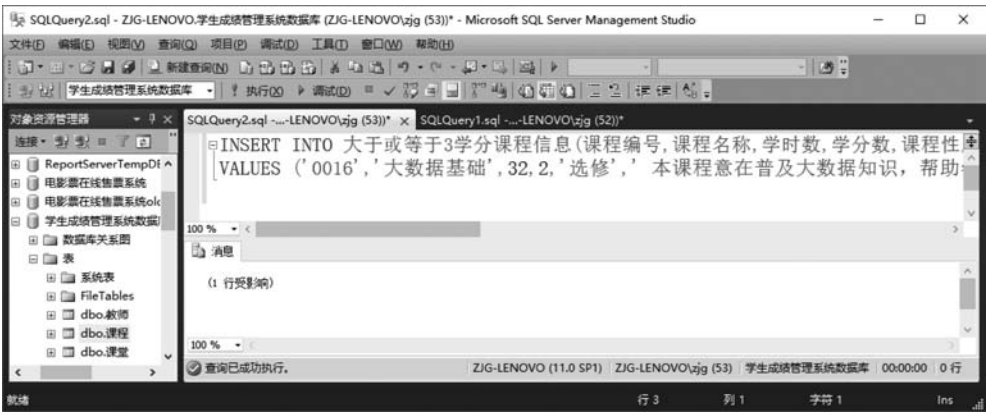


图 5.18 输入不满足 SQL 条件的数据后视图执行成功

为了比较,现在把已创建的“大于或等于 3 学分课程信息”视图删除,如图 5.20 所示。

例 5-11 在学生成绩管理系统数据库中创建“大于或等于 3 学分课程信息”视图,该视图选择一个基表(课程表)中的数据来显示学分数大于或等于 3 学分的数据,并选择用

WITH CHECK OPTION 创建视图,如图 5.21 所示。

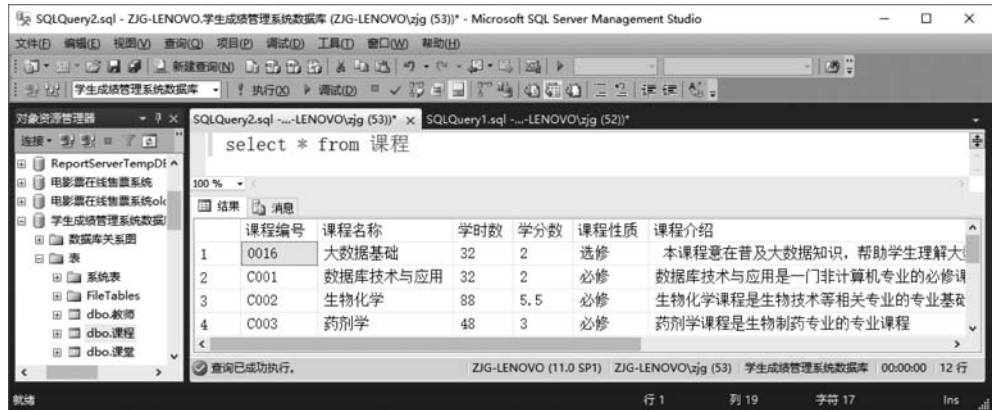


图 5.19 输入不满足 SQL 条件的数据后数据表情况

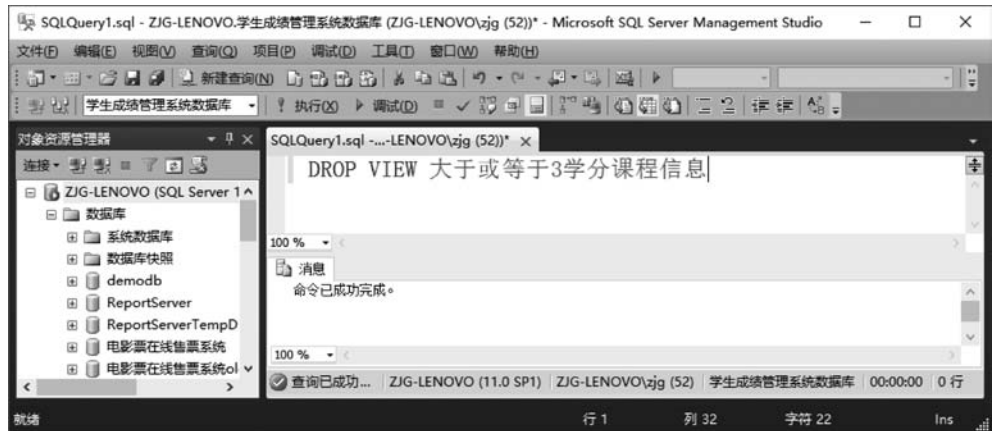


图 5.20 删除不用 WITH CHECK OPTION 的视图示意图

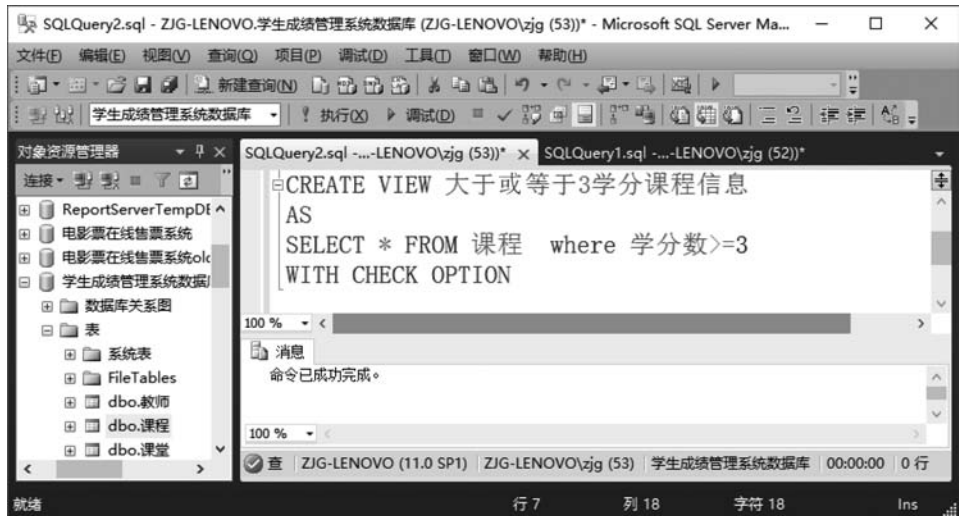


图 5.21 创建用 WITH CHECK OPTION 的视图示意图

可执行如下 SQL 语句：

```
CREATE VIEW 大于或等于 3 学分课程信息
AS
SELECT * FROM 课程 where 学分数>= 3
WITH CHECK OPTION
```

再插入大数据相关课程信息,如图 5.22 所示,发现插入数据无法执行成功,因为这个视图创建时使用了 WITH CHECK OPTION 子句,这就导致在对视图插入数据时要检查创建视图的 SQL 语句的条件。

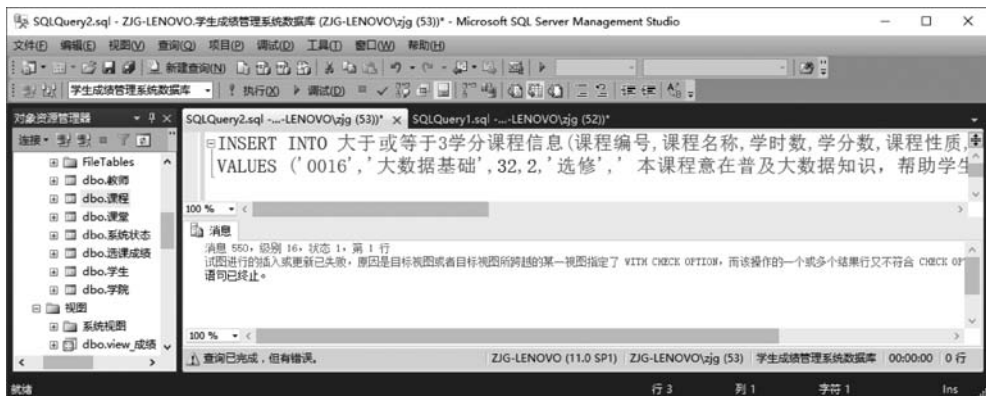


图 5.22 输入不满足 SQL 条件的数据插入情况

注意：不是所有的视图都可以进行更新操作的。

5.2.3 视图的修改

创建好的视图可以通过 SQL Server 2012 管理平台或 Transact-SQL 语句来进行修改。使用 ALTER VIEW 语句来修改视图,其语法格式如下：

```
ALTER VIEW[ schema_name. ]view_name[ (column[, ...n])]
AS
select_statement
[ WITH CHECK OPTION]
```

各选项的含义如下所述。

- (1) schema_name: 视图所属架构的名称。
- (2) view_name: 要更改的视图。
- (3) column: 一列或多列的名称,用逗号分开,将成为给定视图的一部分。
- (4) WITH CHECK OPTION: 对视图数据更新限制,满足视图的 select 语句的条件。

例 5-12 修改视图“程序设计成绩信息”,添加新字段“性别”。

如图 5.23 所示,命令为

```
ALTER VIEW 程序设计成绩信息
AS
SELECT 学生.学号,姓名,性别,专业班级,课程名称,成绩
FROM 学生 INNER JOIN 选课成绩 ON 学生.学号 = 选课成绩.学号
```

```

INNER JOIN 课堂 ON 选课成绩.课堂编号 = 课堂.课堂编号
INNER JOIN 课程 ON 课堂.课程编号 = 课程.课程编号
WHERE 课程名称 = 'C++程序设计基础'

```

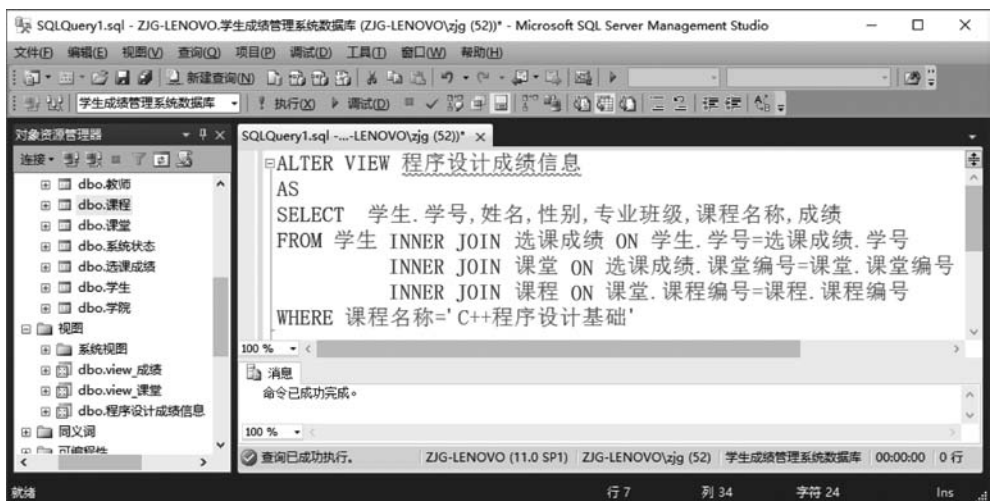


图 5.23 修改视图数据插入字段情况

5.2.4 视图的删除

不再需要的视图可以通过管理平台或 Transact-SQL 语句来删除。
可以使用 DROP VIEW 语句来删除视图,其语法格式如下:

```
DROP VIEW[schema_name.]view_name[,...n]
```

其中,各选项的含义如下所述。

- (1) schema_name: 视图所属架构的名称。
- (2) view_name: 要删除的视图的名称。

例 5-13 删除视图“程序设计成绩信息”,删除视图的 SQL 语句如下:

```
DROP VIEW 程序设计成绩信息
```

可在视图目录下查看该视图是否已经删除。

第一种方法:

查询视图是否存在的 SQL 语句如下:

```
SELECT * FROM sys.views
```

显示“程序设计成绩信息”视图已经不存在了,如图 5.24 所示。

第二种方法:

选择“学生成绩管理系统数据库”→“视图”选项,显示“程序设计成绩信息”视图不存在,如图 5.25 所示。

5.2.5 视图的管理

视图属性包括视图名称、权限、所有者、创建日期和用于创建视图的文本等方面。在

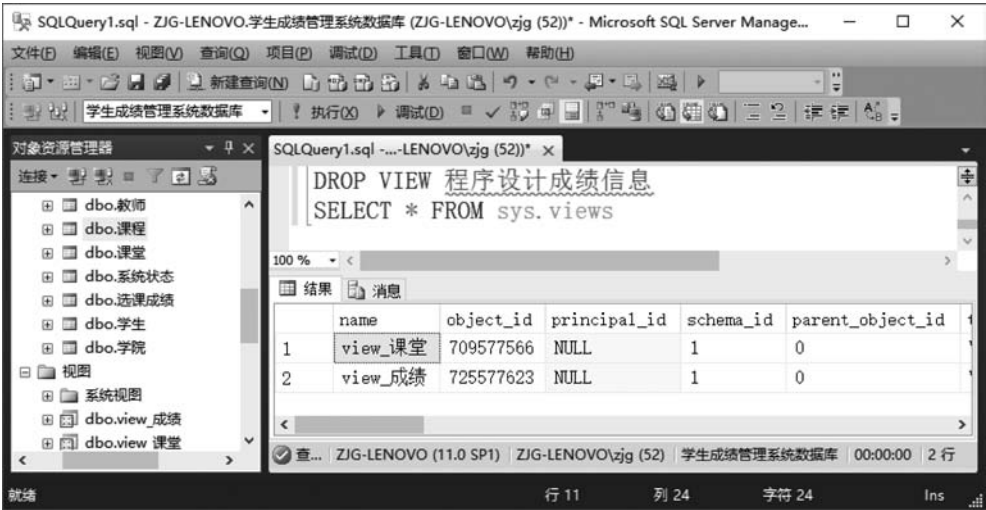


图 5.24 查看程序设计信息视图不存在

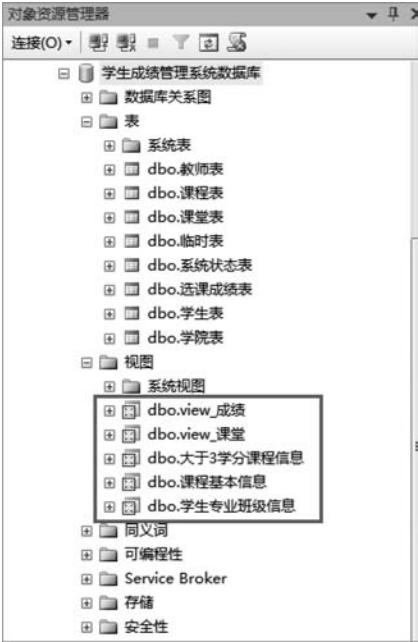


图 5.25 在对象资源管理器中查看视图

SQL Server 中,通过管理平台和系统存储过程可以查看和修改视图的这些信息。

1. 可以使用系统存储过程 `sp_helptext` 查看视图

使用系统存储过程 `sp_helptext` 可以查看视图的文本信息,其语法格式如下:

`sp_helptext` 对象名

对象名可以是视图、规则、默认、未加密的存储过程、触发器等数据库对象名,它用于显示各个数据库对象的文本定义信息。对象必须在当前数据库中。

例 5-14 查看视图“大于或等于 3 学分课程信息”的文本定义信息,可使用以下语句。

sp_helptext 大于或等于 3 学分课程信息

运行结果如图 5.26 所示。

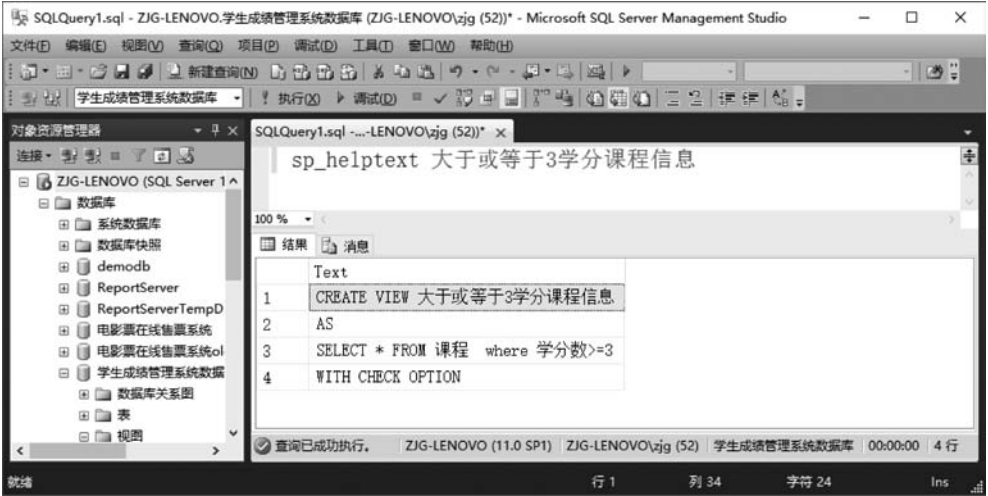


图 5.26 使用 sp_helptext 查看的视图文本信息

2. 使用系统存储过程重命名视图

可以使用系统存储过程 sp_rename 重命名视图,其语法格式及详细介绍参阅 3.3.4 节。

例 5-15 将视图“学生专业班级信息”重命名为“学生班级信息”,可执行如下 SQL 语句。

sp_rename 学生专业班级信息, 学生班级信息

5.2.6 视图的应用

利用视图可以完成某些和基表相同的数据操作。通过视图可以对基表中的数据进行检索、添加、修改和删除,但在做添加、修改和删除时,要注意限制条件,这里主要介绍如何利用视图来查询基表的数据。

在建立视图后,可以用任意一种查询方式检索视图数据,对视图可使用连接、GROUP BY 子句、子查询等以及它们的任意组合。

例 5-16 创建一个“大于或等于 3 学分课程信息”的视图,并通过视图查询相关数据。

- (1) 创建视图,见例 5-7。
- (2) 查看已经创建好的“大于或等于 3 学分课程信息”的视图数据。

SELECT * FROM 大于或等于 3 学分课程信息

在建立视图时,系统并不检索视图所参照的数据库对象是否存在。在通过视图检索数据时,SQL Server 将首先检查这些对象是否存在,如果视图的某个基表(或视图)不存在或已被删除,将导致语句执行错误,系统向用户返回一条错误消息。当新表重新建立后,视图可恢复使用。

在 CREATE VIEW 语句中使用 SELECT 子句建立视图后,如果重新创建或修改该视图的基表结构,并且增加了一些列,这些新增的列将不出现在已定义的视图中,除非这些视图被删除后重建,所以在通过视图检索数据时也不可能检索到新表中所增加列的内容。

本章小结

本章介绍了 SQL Server 2012 中两个重要的概念——索引和视图。索引是可以加快数据检索的一种结构,理解和掌握索引的概念与操作对于学习和进行数据查询很有帮助。视图作为一个查询结果集,虽然与表具有相似的结构,但它是一张虚拟表,以视图结构显示在用户面前的数据并不是以视图的结构存储在数据库中,而是存储在视图所引用的基表当中,视图的存在为保障数据库的安全性提供了新手段。

索引是对数据库表中一个或多个字段的值进行排序而创建的一种分散存储结构。建立索引的主要目的是加速数据检索和连接、优化查询、强制实行唯一性等操作。

索引主要有四种类型:主键索引、唯一性索引、聚集索引和多关键字索引。

(1) 在 SQL Server 2012 中对索引的基本操作包括创建索引、查看索引、更改索引和删除索引,可以在 SQL Server 管理平台或通过 Transact-SQL 语句实现索引操作。

(2) 视图是一种数据库对象,是从一个或多个表或视图中导出的虚拟表。视图所对应的数据并不真正地存储在视图中,而是存储在其所引用的表中,被引用的表称为基表,视图的结构和数据是对基表进行查询的结果。视图被定义后便存储在数据库中,和真实的表一样,视图在显示时也包括几个被定义的列和多个数据行,但通过视图看到的数据只是存放在基表中的数据。对视图的操作和对表的操作一样。

(3) 视图的操作主要包括视图的创建、修改、删除和重命名等,其操作可以通过 SQL Server 管理平台或 Transact-SQL 语句来实现。

(4) 通过视图可以完成某些和基表相同的数据操作,如数据的检索、添加、修改和删除。

习 题 5

一、选择题

(1) 建立索引的主要作用是_____。

- A. 节省存储空间
- B. 便于管理
- C. 提高查询速度
- D. 提高查询和更新的速度

(2) 在数据库设计阶段,需要考虑为关系表建立合适的索引。关于建立索引的描述,有下列说法:

- I. 对于经常在其上需要执行查询操作并且数据量大的表,可以考虑建立索引。
- II. 对于经常在其上需要执行插入、删除和更新操作的表,可以考虑建立索引。
- III. 对于经常出现在 WHERE 子句中的字段,可以考虑建立索引。
- IV. 对于经常出现在 ORDER BY 子句、GROUP BY 子句中的属性,应尽量避免建立

索引。

上述说法中正确的有_____。

- A. I、II和III B. I、II和IV C. I和III D. II和IV

(3) 索引是对数据库表中_____字段的值进行排序。

- A. 一个 B. 多个 C. 一个或多个 D. 零个

(4) 关于视图的叙述中正确的是_____。

- A. 视图是一张虚拟表,所有的视图中都不含数据
B. 用户一定能通过视图更新所有数据
C. 视图是一张实际的物理表,所有的视图中都含有实际数据
D. 视图只能通过表得到,不能通过其他视图得到

(5) SQL Server 中的视图提高了数据库系统的_____。

- A. 完整性 B. 可靠性 C. 安全性 D. 一致性

(6) Transact-SQL 语言中,删除一个视图的命令是_____。

- A. DELETE B. DROP C. CLEAR D. REMOVE

二、填空题

(1) 如果创建唯一索引,只能用_____语句实现。如果创建聚集索引,可以用_____语句实现。

(2) 在 SQL SERVER 中,除了基表以外,_____有对应的物理存储,而_____没有对应的物理存储。

(3) _____是关系数据库中提供给用户以多种角度观察数据库中数据的重要机制。

(4) 数据库中只存放视图的_____,而不存放视图对应的数据,这些数据仍存放在导出视图的基础表中。

(5) 关系数据库系统支持三级模式结构,其中外模式对应于_____,模式对应于基表,内模式对应于存储文件。

(6) 视图是虚拟表,它一经定义就可以和基表一样被查询,但_____操作将有一定限制。

三、问答题

(1) 聚集索引与非聚集索引之间有哪些不同点? 在一个表中可以建立多少个聚集索引和非聚集索引?

(2) 在什么场合下适合创建索引? 请举例说明。

(3) 什么叫视图? 视图有哪些用途?

(4) 在创建视图中,有 WITH CHECK OPTION 子句和没有 WITH CHECK OPTION 子句的区别是什么?

四、应用题

(1) 针对学生成绩管理系统数据库的需求,如果学生常常因为自己所学的课程来查询相关的课堂信息,经常会根据课堂名称来查询相关信息,请思考是否应该对课堂名称创建索引,是否需要创建聚集索引,并说明理由。

(2) 针对学生成绩管理系统数据库的需求,学生试听课程后,可以根据自己喜好对选课信息进行删除,并且再按照自己的喜好插入选课信息,请思考是否应该对课堂编号创建索

引,并说明理由。

(3) 针对学生成绩管理系统数据库的需求,发现用户常常会关心自己专业的同学的相关信息,怎样设计视图来满足用户的需求?

176

(4) 针对学生成绩管理系统数据库的需求,系统有三个角色:一是教师,专门录入学生的成绩数据;二是学生,专门查询自己的成绩数据;三是教务员,主要检查教师录入成绩数据是否正确并激活。如何建立相应的视图?如何给三个角色授权?