



宿州职业技术学院

Suzhou Vocational and Technical College

SQL Server 数据库技术与应用



数据完整性控制



1

数据完整性与约束

2

主键约束的管理

3

唯一约束的管理

4

外键约束的管理

5

其他约束的管理



数据完整性的概念

- ☑ 数据完整性是指存储在数据库中的数据的正确性、一致性、相容性与可靠性。
- ☑ 维护数据完整性，是为了防止数据库中存在不符合语义规则的数据，防止错误信息的输入与输出，从而避免对数据库造成无效的操作与不良破坏。



数据完整性的作用

- ☑ 数据完整性主要用于保证数据库中数据的正确性、一致性与可靠性。
- ☑ 数据完整性是保证数据质量的一种重要方法，是现代数据库系统的一个重要特征。同时，数据完整性的设计是评价数据库设计好坏的一项重要指标，直接影响到数据库能否真实地反映现实世界。



数据完整性的类别

- ☑ SQL Server 2008支持以下4类数据完整性：
 - ① 实体完整性；
 - ② 域完整性；
 - ③ 参照/引用完整性；
 - ④ 用户自定义完整性。



实体完整性的概念

- ☑ 实体完整性，又称为行完整性，用于保证基本表中的每一行记录都与现实世界的唯一实体相对应。实体完整性中的实体，是指数据库中所表示的一个实际事物或事件。实体完整性通过主键来实现，要求表中的所有记录行具有完全不同的主键值，并要求在组成主键的每个列上都不能取空值。



域完整性的概念

- ☑ 域完整性，又称为列完整性，是指数据表特定列输入的有效性。用来保证列值的有效性与正确性。域完整性通常使用有效性检查来保证，也可以通过限制数据的类型、格式、取值范围来实现。



参照完整性的概念

- ☑ 参照完整性，又称为引用完整性，是建立在外键与主键或外键与唯一键之间的一种引用规则。当增加、修改或删除表中数据时，参照完整性用来保证相关联的多个表中数据的一致性与更新的同步性，维护表间的参照关系，确保外键值与主键值或唯一键值在所有表中保持一致，禁止引用不存在的键值。



用户自定义完整性的概念

- ☑ 用户自定义完整性，是根据实际应用的语义要求或数据库的约束条件，由用户自行定义并体现应用需求的特定业务规则。



实现数据完整性的方式

- ☑ SQL Server 2008中采用两种方式实现数据完整性，这两种方式分别为：
 - ① 声明数据完整性；
 - ② 过程数据完整性。



声明完整性的概念

- ☑ 通过在数据库对象定义中定义一系列的数据约束与验证标准，并由数据库系统自身在插入、修改、删除数据时自动实施这些标准的一种数据完整性实现手段。声明数据完整性主要通过定义与使用约束、默认值与规则来实现。



过程数据完整性的概念

- ☑ 通过在脚本语言中定义一系列的数据约束与验证标准，并在脚本执行过程中强制完成这些标准的检验。
- ☑ 过程数据完整性主要通过定义与使用视图、触发器和存储过程来实现。本章仅介绍声明数据完整性的实现方法与技术。



约束的概念

- ☑ 约束是用来保障数据完整性的一种强制性的限制规则。
- ☑ 约束用来描述数据库中数据之间的语法、词义联系，数据间的制约和依存关系，以及数据动态变化的规则，以保证数据的正确性、有效性和相容性。



约束的基本类别

- ☑ SQL Server 系统提供了6种约束类型：
- ① 主键约束；
 - ② 唯一约束；
 - ③ 外键约束；
 - ④ 检查约束；
 - ⑤ 默认约束；
 - ⑥ 非空约束。



约束的其他类别

- ☑ 根据约束所作用的对象是列、行或者表，可将完整性约束分为如下3种：
 - ① 列约束：列约束是对列所设定的某种限制，用来保证输入到该列的数据符合某种指定条；
 - ② 行约束：是对记录行所设定的某种限制，用来保证插入到表中的行数据符合某类指定条件；
 - ③ 表约束：表约束则是用来定义两个表或多个表之间依赖关系的一种手段。外键约束则可看作是一种表约束。



数据完整性与约束的关系

数据完整性类型	数据完整性的涵义	完整性约束或支持技术
实体完整性	规定了每行数据在表中具有唯一性，体现为每条记录具有非空且取值绝不重复的主键列	UNIQUE约束 PRIMARY KEY约束 标识属性(IDENTITY)定义
域完整性	表中指定字段列的数据具有合法的数据类型，符合正确的格式与精度，满足有效的取值范围	FOREIGN KEY约束 CHECK约束 DEFAULT约束 NOT NULL约束
参照/引用完整性	相互关联的多表之间保持数据的统一性与一致性，对主键与外键的引用与更改要保持同步与一致	PRIMARY KEY约束 FOREIGN KEY约束 CHECK约束
用户定义完整性	根据特定应用的语义需求，由用户(而非系统)创建的业务逻辑规则与约束限制	列约束 行约束 表约束 存储过程 触发器



数据完整性控制

1

数据完整性与约束



2

主键约束的管理

3

唯一约束的管理

4

外键约束的管理

5

其他约束的管理



主键约束的概念

- ☑ 主键约束为表定义主键，用以强制实现实体完整性。
- ☑ 每个表最多只能定义一个主键约束。
- ☑ 主键可定义为一个字段列，也可定义为多个字段列的组合；当主键为多列组合时，所有列的组合值必须唯一，不能重复。
- ☑ 记录在主键上必须有值，不允许为空；任意两条记录都不能包含相同的主键值。
- ☑ 主键约束在指定列或列组合上创建唯一性索引。



主键约束定义的条件

- ☑ 定义主键约束的列(称主键列)需满足以下条件:
 - ① 列的数据类型不能为text、ntext、varchar(max)、nvarchar(max)、image、xml类型。
 - ② 列值不能为空值,即列上不能定义非空约束。



主键约束的管理

- ☑ 主键约束的管理既可使用SSMS的可视化集成环境来完成，也可使用T-SQL命令来实现。管理内容包括以下常规操作：
 - ① 创建主键约束；
 - ② 查看主键约束；
 - ③ 修改主键约束；
 - ④ 删除主键约束。



T-SQL创建主键约束概述

- ☑ 使用T-SQL命令创建主键约束，有两种时机：
 - ① 在用CREATE TABLE语句创建表的时候创建主键约束；
 - ② 在用ALTER TABLE语句修改表的时候创建主键约束。



CREATE TABLE创建概述

- ☑ CREATE TABLE语句创建主键约束，又分为为以下两种形式：
 - ① 在单一字段列上创建主键约束；
 - ② 在多个字段列的组合上创建主键约束。



单一列上创建主键约束

- ☑ 在单一字段列上创建主键约束，只须直接在字段列的定义后命名约束名，并使用即可。语法格式如下：
- ☑ `CREATE TABLE <数据表名> (<列名> 数据类型 [CONSTRAINT <约束名>] PRIMARY KEY [ CLUSTERED | NONCLUSTERED ] [ ,...n ] ) [ ; ]`



组合列上创建主键约束

- ☑ 在多个字段列的组合上创建主键约束，须在所有字段定义结束后，再用 PRIMARY KEY 关键字专门指定定义主键约束的那些列名。语法格式如下：
- ☑ CREATE TABLE <数据表名>
([CONSTRAINT <约束名>] PRIMARY KEY
[CLUSTERED | NONCLUSTERED] (<列名
1>[ASC | DESC][,...n])
[,...n])[;]



修改表语句创建主键约束

- ☑ 用修改数据表的ALTER TABLE语句，也可以为已有的表添加主键约束。语法如下：
- ☑ ALTER TABLE <数据表名> ADD CONSTRAINT <约束名> PRIMARY KEY [CLUSTERED | NONCLUSTERED] (<列名1>[ASC | DESC] [, ... n]) [;]



T-SQL查看主键约束

- ☑ 对于已经定义的主键约束，可通过在【查询编辑器】中运行SQL Server预定义的系统存储过程Sp_help来查看约束的信息。
- ☑ 存储过程Sp_help提供约束及其他数据库对象的详情报告，包括对象名称、创建者、对象类型、创建时间等信息。
- ☑ 查看主键约束的基本语法如下：
- ☑ [EXECUTE] Sp_help <约束名>



T-SQL 修改主键约束

- ☑ 如果要修改表中已有的PRIMARY KEY约束，必须先删除这个主键约束，然后再使用ALTER TABLE语句重新创建新的主键约束。



T-SQL删除主键约束

- ☑ 使用ALTER TABLE语句的DROP子句，可以从一个表中删除已有的一个或多个约束。
- ☑ 命令格式如下：
- ☑ ALTER TABLE <数据表名> DROP CONSTRAINT <约束名1> [ASC | DESC] [, ...n] [;]



数据完整性控制

1

数据完整性与约束

2

主键约束的管理



3

唯一约束的管理

4

外键约束的管理

5

其他约束的管理



唯一约束的概念

- ☑ 唯一约束又称为候选的主键约束，用来强制确保在表的非主键列上不输入重复值。唯一约束可施加在一列或多列组合上，但这些列不能是表的主键列；不同记录在这些列或列组合上的值必须是唯一的。
- ☑ 由于一个表中最多只能有一个主键约束，当表中已有主键约束时，如果再需要在其他非主键列上实现实体完整性（即不允许有重复值），此时可通过创建唯一约束来实现。
- ☑ 唯一约束通过在指定的列上创建唯一性索引来强制实体完整性。默认情况下，唯一约束创建的是非聚集索引，也可以显式指定创建聚集索引。



主键与唯一约束的区别

- ☑ 主键约束与唯一约束都是实现实体完整性的手段，但它们又有以下不同之处：
 - ① 一个表中最多只能定义一个主键约束，但却可定义多个唯一约束。
 - ② 主键约束不允许主键列取空值，唯一约束允许所在列取空值。
 - ③ 主键约束与唯一约束都要创建索引；默认情况下，主键约束创建的是聚集索引，唯一约束创建的是非聚集索引。



唯一约束的管理

- ☑ 唯一约束的管理既可使用SSMS的可视化集成环境来完成，也可使用T-SQL命令来实现。管理内容包括以下常规操作：
 - ① 创建唯一约束；
 - ② 查看唯一约束；
 - ③ 修改唯一约束；
 - ④ 删除唯一约束。



T-SQL创建唯一约束概述

- ☑ 使用T-SQL命令创建唯一约束，有两种时机：
 - ① 在用CREATE TABLE语句创建表的时候创建唯一约束；
 - ② 在用ALTER TABLE语句修改表的时候创建唯一约束。



CREATE TABLE创建概述

- ☑ CREATE TABLE语句创建唯一约束，又分为为以下两种形式：
 - ① 在单一字段列上创建唯一约束；
 - ② 在多个字段列的组合上创建唯一约束。



单一列上创建唯一约束

- ☑ 在单一字段列上创建唯一约束，只须直接在字段列的定义后命名约束名，并使用即可。语法格式如下：
- ☑ `CREATE TABLE <数据表名> (<列名> 数据类型 [CONSTRAINT <约束名>] UNIQUE [ CLUSTERED | NONCLUSTERED ] [ ,...n ]) [;]`



组合列上创建唯一约束

- ☑ 在多个字段列的组合上创建唯一约束，须在所有字段定义结束后，再用 PRIMARY KEY 关键字专门指定定义唯一约束的那些列名。语法格式如下：
- ☑ CREATE TABLE <数据表名>
([CONSTRAINT <约束名>] UNIQUE
[CLUSTERED | NONCLUSTERED] (<列名
1>[ASC | DESC][,...n])
[,...n])[;]



修改表语句创建唯一约束

- ☑ 用修改数据表的ALTER TABLE语句，也可以为已有的表添加唯一约束。语法如下：
- ☑ ALTER TABLE <数据表名> ADD CONSTRAINT <约束名> UNIQUE [CLUSTERED | NONCLUSTERED] (<列名1>[ASC | DESC] [, ... n]) [;]



T-SQL查看唯一约束

- ☑ 对于已经定义的唯一约束，可通过在【查询编辑器】中运行SQL Server预定义的系统存储过程Sp_help来查看约束的信息。
- ☑ 存储过程Sp_help提供约束及其他数据库对象的详情报告，包括对象名称、创建者、对象类型、创建时间等信息。
- ☑ 查看唯一约束的基本语法如下：
- ☑ [EXECUTE] Sp_help <约束名>



T-SQL修改唯一约束

- ☑ 如果要修改表中已有的唯一约束，必须先删除这个唯一约束，然后再使用ALTER TABLE语句重新创建新的唯一约束。



T-SQL删除唯一约束

- ☑ 使用ALTER TABLE语句的DROP子句，可以从一个表中删除已有的一个或多个唯一约束。
- ☑ 命令格式如下：
- ☑ ALTER TABLE <数据表名> DROP CONSTRAINT <约束名1> [ASC | DESC] [, ...n] [;]



数据完整性控制

1

数据完整性与约束

2

主键约束的管理

3

唯一约束的管理

4

外键约束的管理

5

其他约束的管理





外键约束的概念

- ☑ 外键约束强制实现参照完整性，能够在同一个数据库的多个表之间建立关联，并维护表与表之间的依赖关系。
- ☑ 外键约束定义一个列或多个列的组合为当前表的外键，该外键值引用同表或者其他表中的主键约束所映射的列或唯一约束所映射的列。
- ☑ 根据外键所映射列的数目，外键约束提供单列参照完整性和多列参照完整性。



外键约束的管理

- ☑ 外键约束的管理既可使用SSMS的可视化集成环境来完成，也可使用T-SQL命令来实现。管理内容包括以下常规操作：
 - ① 创建外键约束；
 - ② 查看外键约束；
 - ③ 修改外键约束；
 - ④ 删除外键约束。



T-SQL创建外键约束

- 外键约束通常用ALTER TABLE语句来创建。语法格式如下：
- ALTER TABLE <参照表名>ADD CONSTRA INT <外键约束名> FOREIGN KEY (列名1, 列名2, ……., 列名n) REFERENCES <被参照表名> (列名1, 列名2, ……., 列名n) [ON DELETE {CASCADE | NO ACTION | SET NULL | SET DEFAULT }] [ON UPDATE {CASCADE | NO ACTION | SET NULL | SET DEFAULT }] [;]



T-SQL查看外键约束

- ☑ 对于已经定义的外键约束，可通过在【查询编辑器】中运行SQL Server预定义的系统存储过程Sp_help来查看约束的信息。
- ☑ 存储过程Sp_help提供约束及其他数据库对象的详情报告，包括对象名称、创建者、对象类型、创建时间等信息。
- ☑ 查看外键约束的基本语法如下：
- ☑ [EXECUTE] Sp_help <约束名>



T-SQL修改外键约束

- ☑ 如果要修改表中已有的外键约束，必须先删除这个外键约束，然后再使用ALTER TABLE语句重新创建新的外键约束。



T-SQL删除外键约束

- ☑ 使用ALTER TABLE语句的DROP子句，可以从一个表中删除已有的一个或多个外键约束。
- ☑ 命令格式如下：
- ☑ ALTER TABLE <数据表名> DROP CONSTRAINT <约束名1> [ASC | DESC] [, ...n] [;]



数据完整性控制

1

数据完整性与约束

2

主键约束的管理

3

唯一约束的管理

4

外键约束的管理

5

其他约束的管理





检查约束的概念

- ☑ 检查约束通过控制列值的范围来实现域完整性。检查约束限制对特定列输入数据的范围或格式，确保该列获得有效值，避免非法数据的产生与扩散。
- ☑ 对同一个列可定义多个检查约束。但标识列、ROWGUIDCOL列或数据类型为TIMESTAMP的列不能定义检查约束，因为这几类列的列值由数据库系统自动添加。



检查与外键两约束的区别

- ☑ 检查约束的作用类似于外键约束，它们都能限制列的取值范围。但两种约束确定列值是否有效的方法却不相同：
 - ① 检查约束通过指定的逻辑表达式来限制列的取值范围；
 - ② 键约束则通过其他表来限制列的取值范围。



检查约束的管理

- ☑ 检查约束的管理既可使用SSMS的可视化集成环境来完成，也可使用T-SQL命令来实现。管理内容包括以下常规操作：
 - ① 创建检查约束；
 - ② 查看检查约束；
 - ③ 修改检查约束；
 - ④ 删除检查约束。



T-SQL创建检查约束概述

- ☑ 创建检查约束主要通过以下两种命令：
- ☑ CREATE TABLE语句，该语句在新建表的同时创建一个或多个检查约束；
- ☑ ALTER TABLE语句，该语句对已有表添加一个或多个检查约束。



CREATE TABLE创建约束

- ☑ CREATE TABLE语句创建检查约束的语法格式如下：
- ☑ CREATE TABLE <数据表名>项(<列名> 数据类型 [<其他约束>,] [CONSTRAINT <检查约束名>] CHECK (<约束表达式> [, ... n]) [;]



ALTER TABLE创建约束

- ☑ ALTER TABLE语句添加检查约束的语法格式如下：
- ☑ ALTER TABLE <数据表名>[WITH {CHECK | NOCHECK}] ADD { CONSTRAINT <检查约束名> CHECK (<约束表达式>) [, ... n]) [;]



T-SQL查看检查约束

- ☑ 对于已经定义的检查约束，可通过在【查询编辑器】中运行SQL Server预定义的系统存储过程Sp_help来查看约束的信息。
- ☑ 存储过程Sp_help提供约束及其他数据库对象的详情报告，包括对象名称、创建者、对象类型、创建时间等信息。
- ☑ 查看检查约束的基本语法如下：
- ☑ [EXECUTE] Sp_help <约束名>



T-SQL 修改检查约束

- ☑ 如果要修改表中已有的检查约束，必须先删除这个检查约束，然后再使用ALTER TABLE语句重新创建新的检查约束。



T-SQL删除检查约束

- ☑ 使用ALTER TABLE语句的DROP子句，可以从一个表中删除已有的一个或多个检查约束。
- ☑ 命令格式如下：
- ☑ ALTER TABLE <数据表名> DROP CONSTRAINT <检查约束名> [;]



默认约束的概念

- ☑ 默认约束是由数据库系统为特定列提供默认值，当对表使用INSERT语句插入记录时，如果未显式指定特定列的值，则该列将被自动赋予默认值。
- ☑ 默认约束用来实现域完整性。
- ☑ 一个列只能定义一个默认约束。
- ☑ 不能对标识列、ROWGUIDCOL列或数据类型为TIMESTAMP的列定义默认约束。



默认约束的管理

- ☑ 默认约束的管理既可使用SSMS的可视化集成环境来完成，也可使用T-SQL命令来实现。管理内容包括以下常规操作：
- ① 创建默认约束；
 - ② 查看默认约束；
 - ③ 修改默认约束；
 - ④ 删除默认约束。



T-SQL创建默认约束概述

- ☑ 创建默认约束主要通过以下两种命令：
- ☑ CREATE TABLE语句，该语句在新建表的同时创建一个或多个默认约束；
- ☑ ALTER TABLE语句，该语句对已有表添加一个或多个默认约束。



CREATE TABLE 创建约束

- ☑ CREATE TABLE 语句创建默认约束的语法格式如下：
- ☑ CREATE TABLE <数据表名> (<列名>
数据类型 [<其他约束>]
[CONSTRAINT <默认约束名>]
DEFAULT <默认值表达式>
[, ... n]) [;]



ALTER TABLE创建约束

- ☑ ALTER TABLE语句添加默认约束的语法格式如下：
- ☑ ALTER TABLE <数据表名> ADD
CONSTRAINT <默认约束名>
DEFAULT (<默认值表达式>) FOR <
列名> [;]



T-SQL查看默认约束

- ☑ 对于已经定义的检查默认约束，可通过在【查询编辑器】中运行SQL Server预定义的系统存储过程Sp_help来查看默认约束的信息。
- ☑ 存储过程Sp_help提供约束及其他数据库对象的详情报告，包括对象名称、创建者、对象类型、创建时间等信息。
- ☑ 查看默认约束的基本语法如下：
- ☑ [EXECUTE] Sp_help <约束名>



T-SQL 修改默认约束

- ☑ 如果要修改表中已有的默认约束，必须先删除这个默认约束，然后再使用ALTER TABLE语句重新创建新的默认约束。



T-SQL删除默认约束

- ☑ 使用ALTER TABLE语句的DROP子句，可以从一个表中删除已有的一个或多个默认约束。
- ☑ 命令格式如下：
- ☑ ALTER TABLE <数据表名> DROP CONSTRAINT <默认约束名> [;]



非空约束的概念

- ☑ 非空约束指定特定列的值不允许为空，即让该列拒绝接受空值。
- ☑ 非空约束用来实现域完整性。
- ☑ 创建表时，如果未对列指定默认值，则 SQL Server 系统为该列提供 NULL 默认值；可通过为该列定义非空约束这改变这种默认的空值。默认情形下，主键列或标识列将自动具有非空约束。



非空约束的管理

- ☑ 非空约束的管理既可使用SSMS的可视化集成环境来完成，也可使用T-SQL命令来实现。管理内容包括以下常规操作：
 - ① 创建非空约束；
 - ② 查看非空约束；
 - ③ 修改非空约束；
 - ④ 删除非空约束。



T-SQL创建非空约束概述

- ☑ 创建非空约束主要通过以下两种命令：
- ☑ CREATE TABLE语句，该语句在新建表的同时创建一个或多个非空约束；
- ☑ ALTER TABLE语句，该语句对已有表添加一个或多个非空约束。



CREATE TABLE 创建约束

- ☑ CREATE TABLE 语句创建非空约束的语法格式如下：
- ☑ CREATE TABLE <数据表名> (<列名> 数据类型 [<其他约束>] [CONSTRAINT <非空约束名>] {NULL | NOT NULL} [, ... n]) [;]



ALTER TABLE创建约束

- ☑ ALTER TABLE语句添加非空约束的语法格式如下：
- ☑ ALTER TABLE <数据表名> ALTER COLUMN <列名> 数据类型 {NULL | NOT NULL} [;]



T-SQL查看非空约束

- ☑ 对于已经定义的检查非空约束，可通过在【查询编辑器】中运行SQL Server预定义的系统存储过程Sp_help来查看非空约束的信息。
- ☑ 存储过程Sp_help提供约束及其他数据库对象的详情报告，包括对象名称、创建者、对象类型、创建时间等信息。
- ☑ 查看非空约束的基本语法如下：
- ☑ [EXECUTE] Sp_help <约束名>



T-SQL 修改非空约束

- ☑ 如果要修改表中已有的非空约束，必须先删除这个非空约束，然后再使用ALTER TABLE语句重新创建新的非空约束。



T-SQL删除非空约束

- ☑ 使用ALTER TABLE语句的DROP子句，可以从一个表中删除已有的一个或多个非空约束。
- ☑ 命令格式如下：
- ☑ ALTER TABLE <数据表名> DROP CONSTRAINT <非空约束名> [;]



下一章再见!