# 数据完整性

关系型数据库系统和文件系统的一个不同点是，关系型数据库本身能保证存储数据的完整性，不需要应用程序的控制，而文件系统一般需要在程序端进行控制。当前几乎所有的关系型数据库都提供了约束机制，该机制提供了一条强大而简易的途径来保证数据库中数据的完整性。

一般来说，数据完整性有以下三种形式：

1、实体完整性保证表中有一个主键。在InnoDB存储引擎表中，用户可以通过定义Primary Key或Unique Key约束来保证实体的完整性。用户还可以通过编写一个触发器来保证数据完整性。

2、域完整性保证数据每列的值满足特定的条件。在InnoDB存储引擎表中，域完整性可以通过以下几种途径来保证：

选择合适的数据类型确保一个数据值满足特定条件

外键约束

编写触发器

还可考虑用DEFAULT约束作为强制域完整性的一个方面

3、参照完整性保证两张表之间的关系。InnoDB存储引擎支持外键，因此允许用户定义外键以强制参照完整性，也可以通过编写触发器以强制执行。

# 概述

约束保证数据的完整性和一致性。

## 约束和索引

当用户创建了一个唯一索引就创建了一个唯一的约束。但是约束和索引还是有所不同的，约束更是一个逻辑的概念，用来保证数据的完整性，而索引是一个数据结构，既有逻辑上的概念，在数据库中还代表着物理存储的方式。

## ENUM和SET约束

## 触发器与约束

# 分类

约束分为表级约束和列级约束：

对一个数据列建立的约束，称为列级约束。

对多个数据列建立的约束，称为表级约束。

列级约束既可以在列定义时声明，也可以在列定义后声明。

表级约束只能在列定义后声明。

对于InnoDB存储引擎而言，提供了以下几种约束：

Primary Key（主键约束）

Unique Key（唯一约束）

Foreign Key（外键约束）

Default（默认约束）

NOT NULL（非空约束）

## 主键约束

每张数据表只能存在一个主键

主键保证记录的唯一性

主键自动为NOT NULL

## 唯一约束

唯一约束可以保证记录的唯一性

唯一约束的字段可以为空值（NULL）

每张数据表可以存在多个唯一约束

注：唯一约束可以存在多个，主键只能是一个，且不为NULL

## 外键约束

外键约束保持数据一致性，完整性，实现一对一或者一对多关系。

外键约束的要求：

1. 父表和子表必须使用相同的存储引擎，而且禁止使用临时表；
2. 数据表的存储引擎只能为InnoDB；
3. 外键列和参照列必须具有相似的数据类型，其中数字的长度或是否有符号位必须相同，而字符的长度则可以不同；
4. 外键列和参照列必须创建索引，如果外键列不存在索引的话，MySQL将自动创建索引。

外键约束的参照操作：

1. CASCARD：从父表删除或更新且自动删除或更新子表中匹配的行
2. SET NULL：从父表删除或更新行，并设置子表中的外键列为NULL，如果使用该项，必须保证子表列没有指定NOT NULL；
3. RESTRICT：拒绝对父表的删除或更新操作；
4. NOT ACTION：标准SQL的关键字，在MySQL中与RESTRICT相同

# 添加约束

添加主键约束：

ALTER TABLE tb\_name ADD [CONSTRAINT [symbol]] PRIMARY KEY [index\_type](index\_col\_name,…)

添加唯一约束：

ALTER TABLE tb\_name ADD [CONSTRAINT [symbol]] UNIQUE [INDEX|KEY] [index\_name] [index\_type](index\_col\_name,…)

添加外键约束：

ALTER TABLE tb\_name ADD [CONSTRAINT [symbol]] FOREIGN KEY [index\_name] (index\_col\_name,…) reference\_definition

添加/删除默认约束：

ALTER TABLE tb\_name ALTER [COLUMN] col\_name {SETDEFAULT literal | DROP DEFAULT}

# 删除约束

删除主键约束：

ALTER RABLE tb\_name DROP PRIMARY KEY

删除唯一约束：

ALTER RABLE tb\_name DROP {INDEX|KEY} index\_name

删除外键约束：

ALTER RABLE tb\_name DROP FOREIGN KEY fk\_symbol