

数据库完整性约束

完整性约束

——数据库完整性约束简介

- 理解数据库完整性约束的作用；
 - 理解数据库中的各种约束的作用。
-
- 完整性约束是保证用户对数据库所做的修改不会破坏数据的一致性，是保护数据**正确性和相容性**的一种手段。
-
- 维护完整性**
- 在一个DBMS之中，为了能够维护数据库的完整性，必须能够提供以下的几种支持：

提供定义完整性约束条件机制	在数据表上定义规则，这些规则是数据库中的数据必须满足的语义约束条件；
提供完整性检查的方法	在更新数据库时检查更新数据是否满足完整性约束条件；
违约处理	DBMS发现数据违反了完整性约束条件后要采取的违约处理行为，如拒绝（NO ACTION）执行该操作，或者级联（CASCADE）执行其他操作。

- 主要约束分类**

开发之中可以使用以下的五种约束进行定义：

非空约束：	如果使用了非空约束的话，则以后此字段的内容不允许设置成null；
唯一约束：	即：此列的内容不允许出现重复；
主键约束：	表示一个唯一的标识，例如：人员ID不能重复，且不能为空；
检查约束：	用户自行编写设置内容的检查条件；
主-外键约束（参照完整性约束）：	是在两张表上进行的关联约束，加入关联约束之后就产生父子的关系。
隐式约束	例如数据类型

——非空约束：NK

- 在正常情况下，**NULL**是每个属性的**合法数据值**。如果说现在某个字段不能为NULL，且必须存在数据，那么就可以依靠**非空约束来进行控制**，这样在数据更新时，此字段的内容出现NULL时就会产生错误。
- 非空约束不允许字段为null；
- 非空约束出现错误时会提示完整的错误位置。
- 使用非空约束：**只需要在定义列的时候后面增加一个**NOT NULL**即可。**

◦ 定义member表，其中姓名不允许为空
<pre>DROP TABLE member PURGE ; CREATE TABLE member(mid NUMBER , name VARCHAR2(200) NOT NULL);</pre>

公告

昵称： 寻香径
园龄： 4年4个月
粉丝： 19
关注： 3
[+加关注](#)

<	2021年11月			
日	一	二	三	
31	1	2	3	
7	8	9	10	
14	15	16	17	
21	22	23	24	
28	29	30	1	
5	6	7	8	

搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

我的标签

Database(73)
Oracle(70)
Powershell(34)
SQL Fundamentals(26)
English(26)
Exchange(19)
Pronunciation(17)
Windows(12)
Active Directory(12)
Linux(10)
[更多](#)

随笔分类 (214)

ActiveDirectory(11)
English(8)
English Corner(18)
Exchange(18)
Life(4)
Linux(10)
Manual List(2)
Network(1)
Oracle(44)
PL/SQL(7)
Powershell(34)
Python(1)
SQL Fundamentals(26)
Vmware(4)
Web(4)
Windows(20)
正则表达式(2)

- 向member表中插入数据，name栏位不能为空

```
INSERT INTO member(mid,name) VALUES(1,'wendy')
```

—— 唯一约束：UK和CONSTRAINT子句

唯一约束,CONSTRAINT (constraint约束)

- 唯一约束（**UNIQUE**，简称UK）表示的是在表中的数据不允许出现重复的情况
- 唯一约束可以设置NULL；
- 唯一约束的列不允许重复。
- 只需要在定义列的时候后面增加一个**UNIQUE**即可。

- 定义member表，其中姓名不允许为空

```
DROP TABLE member PURGE ;
```

```
CREATE TABLE member(  
    mid    NUMBER ,  
    name   VARCHAR2(200) NOT NULL ,  
    email  VARCHAR2(50)  UNIQUE  
);
```

向表中添加数据：

```
INSERT INTO member(mid,name,email) VALUES(1,'t','z@qq.com')
```

```
INSERT INTO member(mid,name,email) VALUES(2,'w','z@qq.com')
```

当添加第二个数据时候出现错误：ORA-00001: 违反唯一约束条件 (SCOTT.SYS_C0011099)

- 因为约束在数据库中也是一个对象，每个对象都有自己的名字，如果用户没有指定名字，那么就由系统动态分配一个，所以如上出现的错误是SYS_C0011099
- 对于约束的名字，建议写法：“约束简写_字段名”

为唯一约束指定一个名字

```
CONSTRAINT uk_email UNIQUE(email)
```

```
DROP TABLE member PURGE ;
```

```
CREATE TABLE member(  
    mid    NUMBER ,  
    name   VARCHAR2(200) NOT NULL ,  
    email  VARCHAR2(50) ,  
    CONSTRAINT uk_email  UNIQUE (email)  
);
```

唯一约束是不受空类型控制。

如下两条记录都可以执行

```
INSERT INTO member(mid,name,email) VALUES(1,'t',null)
```

```
INSERT INTO member(mid,name,email) VALUES(2,'w',null)
```

主键约束：PK

- 主键约束：PK
 - 如果一个字段即要求唯一，又不能设置为null，则可以使用主键约束（主键约束 = 非空约束 + 唯一约束），
 - 主键约束使用**PRIMARY KEY**（简称PK）进行指定。

随笔档案 (183)

2017年8月(20)
2017年7月(72)
2017年6月(91)

IT

xiongzaqiren
python学习网址
huyong
美式音标
李兴华
Henry
潇湘隐者
布布扣
Tom
microsoft Developer
小麦苗
阿里云
AskTom
virtuallyimpossible
Netwrix Blog
EventLog
Tutorialspoint
程序园
windowsitpro
TechTarget数据库
HELLO ENGLISH
CareySon
Justin Young
bhargavs
port25guy
exchangeblog.pl
CLAY
jaapbrasser
englishclub
马开东
techonthenetoracle
ORACLE doc
欢醉
ORACLEMySQLBlog
mysqlplanet.
旧博客地址
选择了就坚持
苍月代表我

常用

音标网
w3schools
Linux命令在线手册
西哥

在member表中的mid字段应该表示一个成员的唯一编号，而这个编号即不能为空，也不能重复。

设置member表中的mid为主键

```
DROP TABLE member PURGE ;
CREATE TABLE member(
    mid    NUMBER    PRIMARY KEY ,
    name   VARCHAR2(200) NOT NULL ,
    email  VARCHAR2(50) ,
    CONSTRAINT uk_email UNIQUE (email)
);
```

指定主键约束的名称

```
DROP TABLE member PURGE ;
CREATE TABLE member(
    mid    NUMBER    ,
    name   VARCHAR2(200) NOT NULL ,
    email  VARCHAR2(50) ,
    CONSTRAINT pk_mid PRIMARY KEY (mid) ,
    CONSTRAINT uk_email UNIQUE (email)
);
```

- 复合主键

- 在实际的开发之中，一般在一张表中只会设置一个主键，但是也允许为一张表设置多个主键，这个时候将其称为复合主键。在复合主键中，只有两个主键字段的内容完全一样，才会发生违反约束的错误。

不建议使用复合主键

检查约束：CK

- 检查约束指的是对数据增加的条件过滤，表中的每行数据都必须满足指定的过滤条件。
- 在进行数据更新操作时，如果满足检查约束所设置的条件，数据可以成功更新，如果不满足，则不能更新。
- 在SQL语句中使用CHECK（简称CK）设置检查约束的条件。
- 检查约束会设置多个过滤条件，所以检查约束过多时会影响数据更新性能。
- 在member表中增加age字段（年龄范围是0~200岁）和sex字段（只能是男或女）

```
DROP TABLE member PURGE ;
CREATE TABLE member(
    mid    NUMBER    ,
    name   VARCHAR2(200) NOT NULL ,
    email  VARCHAR2(50) ,
    age    NUMBER    CHECK (age BETWEEN 0 AND 200) ,
    sex    VARCHAR2(10) ,
    CONSTRAINT pk_mid_name PRIMARY KEY (mid,name) ,
    CONSTRAINT uk_email UNIQUE (email) ,
    CONSTRAINT ck_sex CHECK (sex IN ('男','女'))
);
```

主-外键约束：FK（重要）

- 级联操作：ON DELETE CASCADE、ON DELETE SET NULL；
- 使用外键约束后删除表时应先删除子表再删除父表；
- 在进行外键设置的时候，对应的字段，在父表中必须是主键或是唯一约束。
- 如果A表和B表被设置为互为外键，那么一张表都无法删除。
 - 这个时候只能强制删除：DROP TABLE member CASCADE CONSTRAINT；
 - 不建议使用，在设计表的时候就应该考虑好先后关系
- 外键约束的产生分析
 - 例如，现在公司要求每一位成员为公司发展提出一些更好的建议，并且希望将这些建议保存在数据表之中，那么根据这样的需求，可以设计出如图所示的设计模型。
 - 一个成员可以提出多个建议，这是一个一对多的关系。
 - 设计出了两张数据表，两张表的作用如下：
 - **人员表**：用于保存成员的基本信息（编号、姓名）；
 - **建议表**：保存每一个成员提出的建议内容，所以在此表之中保存在了一个成员编号，即：通过此成员编号就可以和成员表进行数据的关联。

建立member和advice表

```
DROP TABLE member PURGE ;
DROP TABLE advice PURGE ;
CREATE TABLE member(
    mid    NUMBER    ,
    name   VARCHAR2(200) NOT NULL ,
    CONSTRAINT pk_mid PRIMARY KEY (mid)
);
CREATE TABLE advice(
    adid   NUMBER    ,
    content CLOB  NOT NULL ,
    mid    NUMBER,
    CONSTRAINT pk_adid PRIMARY KEY (adid)
);
```

增加一组正确的数据；

WENDY提出了2个意见，GAGA提出了3个意见

```
INSERT INTO member(mid,name) VALUES(1,'WENDY');
INSERT INTO member(mid,name) VALUES(2,'GAGA');
INSERT INTO advice(adid,content,mid) VALUES(1,'Miss Blair, we must go or we'll be late',1);
INSERT INTO advice(adid,content,mid) VALUES(2,'I've been invited to a charity event',1);
INSERT INTO advice(adid,content,mid) VALUES(3,'Okay, so whose mess is it?',2);
INSERT INTO advice(adid,content,mid) VALUES(4,'Can i borrow you for a minutes?',2);
INSERT INTO advice(adid,content,mid) VALUES(5,'There is nothing I can do',2);
```

查询每个成员的完整信息和所提出的意见的数量

确定所需要的数据表：

member表

advice表：统计建议数量

确定关联字段：m.mid=a.mid

```
SELECT m.mid,m.name,COUNT(a.mid)
FROM member m,advice a
WHERE m.mid=a.mid
GROUP BY m.mid,m.name;
```

增加一个建议，member表中并不存在99的信息，这时候就会出现错误数据
INSERT INTO advice(adid,content,mid) VALUES(6,q'[nothing]',99);
现在对于表可以分为父表（member）和子表（advice），因为子表中的数据必须参考member的数据。建议提出的成员编号应该是在member表中mid列上存在的数据。
为了保证表的数据的有效性，只能利用外键约束完成。 外键使用 FOREIGN KEY 来进行设置
增加外键配置
DROP TABLE member PURGE ; DROP TABLE advice PURGE ; CREATE TABLE member(mid NUMBER , name VARCHAR2(200) NOT NULL , CONSTRAINT pk_mid PRIMARY KEY (mid)); CREATE TABLE advice(adid NUMBER , content CLOB NOT NULL , mid NUMBER, CONSTRAINT pk_adid PRIMARY KEY (adid) , CONSTRAINT fk_adid FOREIGN KEY (mid) REFERENCES member(mid));
执行错误的数据插入： INSERT INTO advice(adid,content,mid) VALUES(6,q'[nothing]',99);
ORA-02291: 违反完整约束条件 (SCOTT.FK_ADID) - 未找到父项关键字

一旦为表中增加了外键约束，就会有新的问题：

级联操作问题

问题一：删除父表数据前需要先清出所有子表的对应数据
DELECT FROM member WHERE mid=1;
RA-02291: 违反完整约束条件 (SCOTT.FK_ADID) - 未找到父项关键字。
如果要删除这条记录，就必须删除子表的记录,但是这样的做法并不可取。
DELETE FROM advice WHERE mid=1; DELETE FROM member WHERE mid=1;
解决方法：根据业务需求选择合适的级联操作 【级联操作一】级联删除（ON DELETE CASCADE） 当主数据表被删除后，对应的子表数据也应该同时被清理.这样不行，比如一个公司部门取消，不能将雇员都删除。 【级联操作二】级联设置NULL（ON DELETE SET NULL） 当主表数据被删除后，对应的子表数据的相应字段的内容会设置为null.
级联删除配置（ON DELETE CASCADE）
DROP TABLE advice PURGE ; DROP TABLE member PURGE ; CREATE TABLE member(mid NUMBER , name VARCHAR2(200) NOT NULL , CONSTRAINT pk_mid PRIMARY KEY (mid)); CREATE TABLE advice(adid NUMBER , content CLOB NOT NULL , mid NUMBER, CONSTRAINT pk_adid PRIMARY KEY (adid) , CONSTRAINT fk_adid FOREIGN KEY (mid) REFERENCES member(mid) ON DELETE CASCADE);

```
adid  NUMBER  ,
content  CLOB  NOT NULL ,
mid  NUMBER,
CONSTRAINT pk_adid PRIMARY KEY (adid) ,
CONSTRAINT fk_adid FOREIGN KEY (mid) REFERENCES member(mid) ON DELETE CASCADE
);
```

现在删除父表记录，父表记录以及子表记录的信息都被删除。

```
DELETE FROM member WHERE mid=1;
```

级联更新：级联设置NULL（ON DELETE SET NULL）

```
DROP TABLE advice PURGE ;
DROP TABLE member PURGE ;
CREATE TABLE member(
    mid  NUMBER  ,
    name  VARCHAR2(200) NOT NULL ,
    CONSTRAINT pk_mid PRIMARY KEY (mid)
);
CREATE TABLE advice(
    adid  NUMBER  ,
    content  CLOB  NOT NULL ,
    mid  NUMBER,
    CONSTRAINT pk_adid PRIMARY KEY (adid) ,
    CONSTRAINT fk_adid FOREIGN KEY (mid) REFERENCES member(mid) ON DELETE SET NULL
);
```

删除父表中的记录，这个时候父表中的记录会被删除，而子表中的记录会变为null空。

```
DELETE FROM member WHERE mid=1;
```

问题二：删除父表时需要先将子表删除

查看约束

- 约束依然属于数据库对象，可以直接利用数据字典查看。
- 约束是由数据库自己创建的兑现个，所有对象都会在数据字典中进行保存
- 可以利用user_constraints数据字典或者user_cons_columns数据字典查看。

创建一张member表，主键不设置名字

```
CREATE TABLE member(
    mid  NUMBER  PRIMARY KEY ,
    name  VARCHAR2(200) NOT NULL ,
);
```

向表中插入数据，

```
INSERT INTO member(mid,name) VALUES(1,'WENDY');
INSERT INTO member(mid,name) VALUES(1,'GAGA');
```

- 当插入第二条数据时候会报错，ORA-00001: 违反唯一约束条件 (SCOTT.SYS_C0011147)
- 因为约束在数据库中也是一个对象，每个对象都有自己的名字，如果用户没有指定名字，那么就由系统动态分配一个，所以如上出现的错误是SYS_C0011099
 - SYS_C0011099就是数据库对象（约束）的名称

user_constraints查看约束信息，就能看到SYS_C0011099

Constraints_type字段可以看到约束的类型，P为主键，C为检查

- ```
SELECT * FROM user_constraints;
```
- 查看全部的约束名称、类型、约束设置对应的表名称
- ```
SELECT constraint_name,constraint_type,table_name FROM user_constraints ;
```
- 查询emp表上的全部约束

<pre>SELECT constraint_name,constraint_type,table_name FROM user_constraints WHERE table_name='EMP' ;</pre>
要查询约束所对应的字段，这时候查user_cons_columns
<pre>SELECT * FROM user_cons_columns;</pre>

修改约束（了解）

- 约束在建立表的时候一定要同时建立；
- 对于约束不建议对其进行修改。

为表中增加约束	<ul style="list-style-type: none"> ◦ 语法：ALTER TABLE 表名称 ADD CONSTRAINT 约束名称 PRIMARY KEY(约束字段) ; ◦ 范例：为member表的mid字段增加主键约束 <ul style="list-style-type: none"> ◦ ALTER TABLE member ADD CONSTRAINT pk_mid PRIMARY KEY(mid) ; ◦ 范例：为member表的age增加检查约束 <ul style="list-style-type: none"> ◦ ALTER TABLE member ADD CONSTRAINT ck_age CHECK(age BETWEEN 0 AND 200) ;
启用 / 禁用约束	<ul style="list-style-type: none"> ◦ 禁用约束 <ul style="list-style-type: none"> ◦ ALTER TABLE 表名称 DISABLE CONSTRAINT 约束名称 [CASCADE]; ◦ 范例：禁用advice表中的adid主键约束 “pk_adid” <ul style="list-style-type: none"> ◦ ALTER TABLE advice DISABLE CONSTRAINT pk_adid ; ◦ 启用约束 <ul style="list-style-type: none"> ◦ ALTER TABLE 表名称 ENABLE CONSTRAINT 约束名称 ; ◦ 范例：禁用member表中的 “pk_mid” 约束，此字段在advice表中是外键 <ul style="list-style-type: none"> ◦ ALTER TABLE member DISABLE CONSTRAINT pk_mid ;
删除约束	<ul style="list-style-type: none"> ◦ ALTER TABLE 表名称 DROP CONSTRAINT 约束名称 [CASCADE]; ◦ 范例：删除advice表之中的 “pk_adid” 约束 —— 无关联外键 <ul style="list-style-type: none"> ◦ ALTER TABLE advice DROP CONSTRAINT pk_adid ; ◦ 范例：删除member表之中的 “pk_mid” 约束 —— 有关联外键 <ul style="list-style-type: none"> ◦ ALTER TABLE member DROP CONSTRAINT pk_mid CASCADE ;

分类: Oracle

标签: Database, Oracle



+加关注

« 上一篇: 数据字典Data Dict

» 下一篇: Oracle Schema Objects——View

0 0

[刷新评论](#) [刷新页面](#) [返回顶部](#)

登录后才能查看或发表评论, 立即 [登录](#) 或者 [逛逛](#) 博客园首页

【推荐】百度智能云11.11上云钜惠盛典, 注册实名领万元礼包

【推荐】创新驱动数字化2021天翼云论坛, 11.12相约广州, 不见不散

【推荐】博客园x阿里云联合征文活动: 我修复的印象最深的一个bug

【推荐】跨平台组态\工控\仿真\CAD 50万行C++源码全开放免费下载!

【推荐】博客园老会员送现金大礼包, VTH大屏助力研发企业协同数字化

【推荐】华为HMS Core线上Codelabs挑战赛第3期: 用3D建模构建元宇宙

编辑推荐:

- 3D 穿梭效果? 使用 CSS 轻松搞定
- Asp.net core 配置信息读取的源码分析梳理
- [WPF] 玩玩彩虹文字及动画
- 记一次 .NET 某风控管理系统 内存泄漏分析
- 微服务与架构师

最新新闻:

- "卖惨带货"包装术: 剧本百元可定制 借苦情吸粉靠直播变现 (2021-11-11 17:25)
- 锤子科技第4次被申请破产 前三次均已撤回 (2021-11-11 17:17)
- HTC最新VR眼镜确定11月18日发售 近视眼也可裸眼享受 (2021-11-11 17:11)
- 疑清华贫困生本人回应"树洞"刷屏: 希望激励和我一样的清华人 (2021-11-11 17:03)
- FF宣布汉福德工厂获得最终生产使用资质 (2021-11-11 16:50)

» [更多新闻...](#)