第5章

数据库完整性

CHAPTER 5

学习目标

- 理解数据库完整性的概念。
- 掌握 Oracle 各种约束的类型。
- 熟练掌握 Oracle 各种约束的创建方法。

数据库完整性指的是数据的正确性和相容性。正确性是指数据是符合现实世界语义，反映了当前实际状况的。相容性是指数据库同一对象在不同关系表中的数据是符合逻辑的。在数据库中存储的数据，必须保证数据的完整性。完整性通过一组完整性规则来约束，而完整性规则是对关系的某种约束条件，Oracle 使用完整性约束防止不合法的数据写入数据库。本章首先讲解数据库完整性的概念，然后讲解 Oracle 各种约束的创建方法及查看方法。

5.1　完整性概述

Oracle 完整性约束可以分为实体完整性、域完整性、参照完整性、用户定义的完整性 4 类。

5.1.1　实体完整性

实体完整性指的是如果一个字段是主键字段，则此字段不能取空值(NULL)。

例如，EMP 表中雇员编号 EMPNO 为主键，则该字段在输入时不能取空(NULL)，如表 5-1 所示。

表 5-1　EMP 表

雇员编号 EMPNO	雇员姓名 ENAME	雇员职务 JOB	雇员经理编号 MGR	雇员雇佣日期 HIREDATE	雇员工资 SAL	雇员津贴 COMM	部门编号 DEPTNO
7369	SMITH	CLERK	7902	17-12 月-80	800		20
7499	ALLEN	SALESMAN	7698	20-2 月-81	1600	300	30
7521	WARD	SALESMAN	7698	22-2 月-81	1250	500	30
7566	JONES	MANAGER	7839	02-4 月-81	2975		20
7654	MARTIN	SALESMAN	7698	28-9 月-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-5 月-81	2850		30
7782	CLARK	MANAGER	7839	09-6 月-81	2450		10
7788	SCOTT	ANALYST	7566	19-4 月-87	3000		20
7839	KING	PRESIDENT		17-11 月-81	5000		10
7844	TURNER	SALESMAN	7698	08-9 月-81	1500	0	30
7876	ADAMS	CLERK	7788	23-5 月-87	1100		20

5.1.2　域完整性

域完整性指的是数据类型、范围、长度等约束。

例如，表 5-1 所示 EMP 表的定义如下。

```
 名称      是否为空?            类型
-------  --------  ------------------------------------------------
EMPNO     NOT NULL             NUMBER(4)
```

```
ENAME                          VARCHAR2(10)
JOB                            VARCHAR2(9)
MGR                            NUMBER(4)
HIREDATE                       DATE
SAL                            NUMBER(7,2)
COMM                           NUMBER(7,2)
DEPTNO                         NUMBER(2)
```

EMP 表中的字段类型、长度这些定义都是指域完整性的定义。

5.1.3 参照完整性

1. 外键

一个字段或一组字段不是表 R 的主键,但它和另外一个表 S 的主键相对应,则该字段或字段组合为 R 的外键。

例如,表 5-1 所示 EMP 表中的 DEPTNO 不是 EMP 的主键,但是它和另一表 DEPT 表中的 DEPTNO 相对应(EMP 表中的 DEPTNO 的值都取自 DEPT 表中的 DEPNO 字段)。DEPT 表如表 5-2 所示。

表 5-2 DEPT 表

部门编号 DEPTNO	部门名称 DNAME	所在城市 LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

2. 参照完整性

若一个表的外键和另一个表的主键相对应,则该表在外键上的取值为:或者取空值(NULL)(外键的每个属性值均为空值),或者等于另外一个表的某个元组的主键值。

例如,表 5-1 所示 EMP 表中的外键 DEPTNO 的取值,或者取 NULL 表示没有所属的部门,或者取自表 5-2 所示 DEPT 表中的 DEPTNO 中的某一个字段值(10,20,30,40)。

5.1.4 用户定义的完整性

用户定义的完整性指的是用户根据某一具体应用设置的约束条件。Oracle 中必须提供这样的定义机制,来实现用户定义的约束条件。

例:表 5-1 所示 EMP 表中如果定义 COMM 在 0～5000 的范围就属于用户定义的完整性约束。

5.2　约束类型

Oracle 数据完整性约束可以分为主键(PRIMARY KEY)、非空(NOT NULL)、唯一(UNIQUE)、检查(CHECK)约束和外键(FOREIGN KEY)约束 5 种类型。一个约束条件根据具体情况,可以在列级或表级定义。列级约束约束表的某一字段,出现在表的某一字段,或出现在表的某字段定义之后,约束条件只对该字段起作用。表级约束约束表的一个字段或多个字段,如果涉及多个字段,则必须在表级定义。表级约束出现在所有字段定义之后。

Oracle 数据完整性约束可以通过使用 CREATE TABLE 语句定义,也可以创建表之后使用 ALTER TABLE 添加、修改或删除。

5.2.1　主键约束

视频讲解

候选键是唯一标识表中每一条记录的字段或字段组合。若一个表有多个候选键,则选定其中一个为主键,定义主键的子句为 PRIMARY KEY。指定了主键后,包含在主键里的字段为主键字段,主键字段不为空,主键值唯一。

1. 创建表的定义

创建表时需要定义主键约束,其语法格式如下。

```
CREATE TABLE [schema.]table_name(column datatype [DEFAULT expr]
          [[CONSTRAINT constraint_name]PRIMARY KEY],                -- 列级约束
                    …
[[CONSTRAINT constraint_name] PRIMARY KEY(colum1,colum2,…)]);       -- 表级约束
```

例:创建学生表 STUDENT,包括学号(Sno)、姓名(Sname)、Sage(年龄)、性别(Ssex)、出生日期(Birthday)、院系(Sdept)等学生信息。其中,学号(Sno)为主键。

```
CREATE TABLE STUDENT
(Sno char(6) PRIMARY KEY,
Sname varchar2(8),
Sage number(2,0),
Ssex char(2),
Birthday date,
Sdept char(20));
```

Sno 没有提供主键约束的名字,系统会自动为该约束提供一个名字,如果指定一个约束名,需要使用 CONSTRAINT 关键字。

如果为 Sno 主键约束指定约束名为 PK_Sno,上面的例子可使用如下命令。

```
CREATE TABLE STUDENT
(Sno char(6) CONSTRAINT PK_Sno PRIMARY KEY,
Sname varchar2(8),
Sage number(2,0),
```

```
Ssex char(2),
Birthday date,
Sdept char(20));
```

上面的例子主键约束定义都是列级的,也可以使用表级约束来定义,命令如下。

```
CREATE TABLE STUDENT
(Sno char(6),
Sname varchar2(8),
Sage number(2,0),
Ssex char(2),
Birthday date,
Sdept char(20),
CONSTRAINT PK_Sno PRIMARY KEY(Sno));
```

例:创建一选课表 SC,包括学号(Sno)、课程号(Cno)、成绩(Grade),主键为(Sno,Cno)。

```
CREATE TABLE SC
(Sno varchar(12),
 Cno varchar(3),
 Grade number(3,0),
CONSTRAINT PK_SC PRIMARY(Sno,Cno));
```

该例中的主键(Sno,Cno)定义在表级。当主键为多字段时,主键约束必须定义在表级。

2. 使用 ALTER TABLE

创建表之后可以使用 ALTER TABLE 添加或删除主键约束,向已有表中添加主键约束时,表中的主键值不能有重复的值,表中的主键字段不能有 NULL,删除约束时必须指定要删除的约束名称,语法格式如下。

```
ALTER TABLE table_name
ADD [CONSTRAINT constraint_name] PRIMARY KEY(colum1,colum2,…)
[DROP CONSTRAINT constraint_name]
```

例:为 SC 表添加主键约束(Sno,Cno)。

```
ALTER TABLE SC ADD PRIMARY KEY(Sno,Cno);
```

也可以指定约束名称 PK_SnoCno。

```
ALTER TABLE SC ADD CONSTRAINT PK_SnoCno PRIMARY KEY(Sno,Cno);
```

例:删除 SC 表的主键约束 PK_SnoCno。

```
ALTER TABLE SC DROP CONSTRAINT PK_SnoCno;
```

5.2.2 非空约束

非空约束指定某字段不能取空值(NULL),它只能在列级定义。在默认情况下,

Oracle 允许字段的内容为空值，定义非空约束的子句为 NOT NULL。

1. 创建表的定义

创建表时需要定义非空约束，其语法格式如下。

```
CREATE TABLE [schema.]table_name(column datatype[DEFAULT expr]
          [[CONSTRAINT constraint_name] NOT NULL], -- 列级约束
                  …);
```

例：创建课程表 COURSE，包括课程号(Cno)、课程名(Cname)、学分(Ccredit)，要求课程名不能为空。

```
CREATE TABLE COURSE
(Cno char(3) PRIMARY KEY,
Cname varchar2(20) NOT NULL,
Ccredit number(3));
```

2. 使用 ALTER TABLE

创建表之后可以使用 ALTER TABLE 添加或删除非空约束，向已有表中添加非空约束时，相应的字段不能有 NULL，语法格式如下。

```
ALTER TABLE table_name
MODIFY cloumn [CONSTRAINT constraint_name] NOT NULL;
[DROP CONSTRAINT constraint_name]
```

例：给 STUDENT 表的 Sname 字段添加非空约束，约束名为 NOTNULL_Sname。

```
ALTER TABLE STUDENT
MODIFY Sname CONSTRAINT NOTNULL_Sname NOT NULL;
```

例：删除 STUDENT 表 Sname 的非空约束。

```
ALTER TABLE STUDENT
DROP CONSTRAINT NOTNULL_Sname;
```

5.2.3　唯一约束

唯一约束条件要求表的一个字段或多字段的组合内容必须是唯一的，即不能有重复的值。指定了 UNIQUE 的字段如果没有指定 NOT NULL 约束时允许输入 NULL 值，并且可多次输入 NULL 值。Oracle 认为 NULL 不等于任何值。UNIQUE 约束可以定义在列级，也可以定义在表级。但如果唯一约束包含表的多个字段，则必须在表级定义。

1. 创建表时定义

创建表时定义 UNIQUE 约束的语法格式如下。

```
CREATE TABLE [schema.]table_name(column datatype [DEFAULT expr]
                    [[CONSTRAINT constraint_name] UNIQUE],   -- 列级约束
```

```
                    …
    [[CONSTRAINT constraint_name] UNIQUE(column, … )]); -- 表级约束
```

例：创建课程表 COURSE，包括课程号(Cno)、课程名(Cname)、学分(Ccredit)，要求课程名取值唯一。

```
CREATE TABLE COURSE
(Cno char(3) PRIMARY KEY,
Cname varchar2(20) UNIQUE,
Ccredit number(3));
```

2. 使用 ALTER TABLE

创建表之后可以使用 ALTER TABLE 添加或删除唯一约束，向已有表中添加唯一约束时相应的字段不能有重复的值，语法格式如下。

```
ALTER TABLE table_name
ADD [CONSTRAINT constraint_name] UNIQUE(column, … );
[DROP CONSTRAINT constraint_name]
```

例：给 STUDENT 表的 Sname 字段添加唯一约束，约束名为 UNIQUE_Sname。

```
ALTER TABLE STUDENT
ADD CONSTRAINT UNIQUE_Sname UNIQUE(Sname);
```

例：删除 STUDENT 表 Sname 的唯一约束。

```
ALTER TABLE STUDENT
DROP CONSTRAINT UNIQUE_Sname;
```

5.2.4 检查约束

检查约束是用来定义表上的每一行必须满足约束条件，一个字段上的 CHECK 约束可以定义多个。CHECK 约束不能定义在伪列上，可以调用 SYSDATE、USER 等系统函数。一个 CHECK 约束可以包含一个字段或多字段。CHECK 约束可以定义在列级，也可以定义在表级，如果 CHECK 约束包含表的多个字段，则必须在表级定义。

1. 创建表的定义

创建需要定义 CHECK 约束的表，其语法格式如下。

```
CREATE TABLE [schema.]table_name(column datatype [DEFAULT expr]
            [[CONSTRAINT constraint_name] CHECK(condition)],   -- 列级约束
                …
            [[CONSTRAINT constraint_name] CHECK(condition)]); -- 表级约束
```

例：创建学生表 STUDENT，包括学号(Sno)、姓名(Sname)、Sage(年龄)、性别(Ssex)、出生日期(Birthday)、院系(Sdept)等学生信息。其中，学号(Sno)为主键，性别只能输入‘男’或‘女’。

```
CREATE TABLE STUDENT
(Sno char(6) PRIMARY KEY,
Sname varchar2(8),
Sage number(2,0),
Ssex char(2) CONSTRAINT CH_Ssex CHECK(Ssex = '男' or Ssex = '女'),
Birthday date,
Sdept char(20));
```

2. 使用 ALTER TABLE

创建表之后可以使用 ALTER TABLE 添加或删除检查约束，向已有表中添加检查约束时现有表中的数据必须满足定义的条件，否则禁止添加，语法格式如下。

```
ALTER TABLE table_name
ADD [CONSTRAINT constraint_name] CHECK(condition);
[DROP CONSTRAINT constraint_name]
```

例：为 SC 表中的 Grade 添加一个检查约束 CH_Grade，限制 Grade 的取值范围为 0～100。

```
ALTER TABLE SC
ADD CONSTRAINT CH_Grade CHECK(Grade BETWEEN 0 AND 100);
```

例：删除 SC 表的检查约束 CH_Grade。

```
ALTER TABLE SC
DROP CONSTRAINT CH_Grade;
```

5.2.5　外键约束

视频讲解

通过使用公共字段在表之间建立一种父子关系，在表上定义的外键可以指向主键或者其他表的唯一键。定义外键约束(FOREIGN KEY)必须用 REFERENCES 指定所参照的表及字段，REFERENCES 所参照的表为父表或主表，外键所在的表为子表或从表。外键约束可以定义在列级，也可以定义在表级。

创建表时定义外键约束的语法格式如下。

```
  CREATE TABLE [schema.]table_name(column datatype [DEFAULT expr]
  [[CONSTRAINT constraint_name] REFERENCES table_name(column)[ON DELETE CASCADE|ON DELETE
SET NULL]],   -- 列级约束
                ...
  [[CONSTRAINT constraint_name] FOREIGN KEY(column1,column2,…) REFERENCES table_name
(column1,column2,…) [ON DELETE CASCADE|ON DELETE SET NULL]]); -- 表级约束
```

ALTER TABLE 的语法格式如下。

```
  ALTER TABLE table_name
  ADD [CONSTRAINT constraint_name] FOREIGN KEY(column1,column2,…) REFERENCES table_name
(column1,column2,…) [ON DELETE CASCADE|ON DELETE SET NULL];
  [DROP CONSTRAINT constraint_name]
```

其中,ON DELETE CASCADE、ON DELETE SET NULL 为外键规则子句,可以省略,省略时,也就是 Oracle 的基本参照规则:禁止改变从表中的外键值(此值在主表中主键中不存在);禁止修改在从表中有对应记录的主表记录的主键值;禁止删除在从表中有对应记录的主表记录。

ON DELETE CASCADE:如果子表中子记录存在,则删除主表中的主记录时,级联删除子记录。

ON DELETE SET NULL:如果子表中子记录存在,则删除主表中的主记录时,将子记录(外键值)置成空。

1. 创建表的定义

例:创建一选课表 SC,包括学号(Sno)、课程号(Cno)、成绩(Grade),主键为(Sno,Cno),同时指定其外键为 Sno,参照 Student 表中的 Sno 字段。

```
    CREATE TABLE SC
    (Sno varchar(12) REFERENCES Student(Sno),
     Cno varchar(3),
     Grade number(3,0),
CONSTRAINT PK_SC PRIMARY(Sno,Cno));
```

2. 使用 ALTER TABLE

创建表之后可以使用 ALTER TABLE 添加或删除外键约束,向已有表中添加外键约束时现有表中的数据必须满足外键约束的条件,否则禁止添加。

例:为 EMP 表中的 EMPNO 字段添加外键约束,参照 DEPT 表中的 DEPTNO,约束名为 FK_DEP_DEPTNO。

```
ALTER TABLE EMP
ADD CONSTRAINT FK_DEP_DEPTNO REFERENCES DEPT(DEPTNO);
```

例:为 EMP 表中的 EMPNO 字段添加外键约束,参照 DEPT 表中的 DEPTNO,约束名为 FK_DEP_DEPTNO_DELETE,参照规则为 ON DELETE CASCADE。

```
ALTER TABLE EMP
ADD CONSTRAINT FK_DEP_DEPTNO_DELETE REFERENCES DEPT(DEPTNO) ON DELETE CASCADE;
```

例:为 EMP 表中的 EMPNO 字段添加外键约束,参照 DEPT 表中的 DEPTNO,约束名为 FK_DEP_DEPTNO_SET_NULL,参照规则为 ON SET NULL。

```
ALTER TABLE EMP
ADD CONSTRAINT FK_DEP_DEPTNO_SET_NULL REFERENCES DEPT(DEPTNO) ON SET NULL;
```

例:删除 EMP 表的外键约束 FK_DEP_DEPTNO。

```
ALTER TABLE EMP
DROP CONSTRAINT FK_DEP_DEPTNO;
```

视频讲解

5.2.6　查看约束

Oracle 约束创建之后，可以查看其约束的定义。用户定义的 Oracle 的约束放在数据字典 USER_CONSTRAINTS 和 USER_CONS_COLUMNS 中。USER_CONSTRAINTS 是用户表的所有约束，USER_CONS_COLUMNS 是用户表的字段对应的约束。

1. 查看某表的所有约束

其语法格式如下。

```
SELECT CONSTRAINT_NAME,CONSTRAINT_TYPE,STATUS
FROM USER_CONSTRAINTS
WHERE TABLE_NAME = '用户表';
```

例：查看 EMP 表的所有约束。

```
SELECT CONSTRAINT_NAME,CONSTRAINT_TYPE,STATUS
FROM USER_CONSTRAINTS
WHERE TABLE_NAME =  'EMP';
```

注意此处的 EMP 一定要用大写字母。

2. 查看某表的所有字段对应的约束

其语法格式如下。

```
SELECT CONSTRAINT_NAME,COLUMN_NAME
FROM USER_CONS_COLUMNS
WHERE TABLE_NAME = '用户表';
```

例：查看 EMP 表上所有列对应的约束。

```
SELECT CONSTRAINT_NAME,COLUMN_NAME
FROM USER_CONS_COLUMNS
WHERE TABLE_NAME =  'EMP';
```

5.2.7　约束状态设置

约束设置完成后如果不再适用表，可以把该约束禁用，如果需要可以再激活。约束有激活(Enable)和禁用(Disable)两种状态。默认情况下，约束创建之后就一直起作用。禁用约束是一种暂时的方法，在禁用约束状态下完成操作之后，还应该设为激活状态。约束定义时使用关键字 DISABLE 来设置为禁用状态，使用关键字 ENABLE 来设置为激活状态。可以使用 CREATE TABLE 和 ALTER TABLE 来设置约束状态，其语法格式如下。

```
CREATE TABLE [schema.]table_name(column datatype [DEFAULT expr]
[[CONSTRAINT constraint_name] constraint_type [ENABLE|DISABLE]],   -- 列级约束
                              …
[[CONSTRAINT constraint_name] constraint_type [ENABLE|DISABLE]]); -- 表级约束
```

ALTER TABLE 的语法格式如下。

```
ALTER TABLE table_name
[ENABLE|DISABLE] CONSTRAINT constraint_name;
```

例：创建表 DEPT2 时将 DEPTNO 的主键约束设置为禁用。

```
CREATE TABLE DEPT2(
DEPTNO NUMBER(2) CONSTRAINT PK_DEPTNO PRIMARY KEY DISABLE,
DNAME VARCHAR(14),
LOC VARCHAR(13))
```

例：把表 DEPT2 的主键约束 PK_DEPTNO 激活。

```
ALTER TABLE DEPT2
ENABLE CONSTRAINT PK_DEPTNO;
```

习题

1. Oracle 完整性有哪些？试举例说明。

2. 根据第 4 章习题中的 WORKER、DEPART、SALARY 表完成如下问题。

(1) 建立 WORKER 和 SALARY 表中的外键约束，约束规则为级联删除。

(2) 实施 SALARY 表的“工资”字段值限定有 0～9999 的约束。

(3) 实施 WORKER 表的“性别”字段默认值为“男”的约束。

(4) 限定 WORKER 表中的 Phone 为 11 位电话号码。

(5) 为 WORKER 表和 DEPART 表建立外键约束，约束名为 WD_FK_SETNULL，约束规则为 SET NULL。

(6) 删除约束 WD_FK_SETNULL。