**第05章 操作数据和管理表**

**本章内容**

* 数据类型
* 表的创建
* 列的创建
* 表约束管理
* 数据操作语言DML

# 数据类型

## 字符类型

a) char类型：固定长度的内容，效率高。存储空间范围（1-2000）字节。

b) varchar2类型：动态长度的字符类型，存储空间范围（1-4000）字节。

c) long类型：存储2G的文件大小。

## 数值数字类型

number:number数据类型即可能描述整数，也可以描述小数。

语法结构:

number[(p[,s]]

例：number(5)、number(5,2) 范围：-999.99 999.99

## 日期类型

Date日期：年-月-日 小时：分钟：秒

Timestamp日期：时间戳,可以精确到秒数据后6位.

时间格式的修改：yyyy-mm-dd | yyyy-mm-dd hh24:mi:ss

修改默认日期的显示格式:alter session set nls\_date\_format = 'yyyy-mm-dd';

不过只是暂时会话性质的修改，服务器重启就还原了，永久修改可以：

定位注册表到HKEY\_LOCAL\_MACHINE\SOFTWARE\ORACLE\KEY\_OraDb10g\_home1

然后新建一个reg\_sz

名为NLS\_DATE\_FORMAT

值为YYYY-MM-DD。

## 其它类型

通过大数据处理用来保存二进制数据的类型，目前用的比较少

# 表的创建

## 创建表

--语法：

create table 表名 (

字段 字段类型,

字段1 字段类型);

--示例：创建学生表

create table student ( ---表名

stuid number(4), --学号

stuname varchar2(20), --姓名

stusex char(2), --性别

stubirthday date, --出生日期

stusal number(7,2) --奖学金

);

--示例:创建班级表

create table class

(

cid number(2), --班级编号

cname varchar2(20) --班级名称

);

## 复制表结构及数据

create table 新表 as select \* from 旧表

## 复制表结构不需要数据

create table 新表 as select \* from 旧表 where 2<>2

## 使用union插入数据

insert into emp(empno,ename,deptno)

select 1000,'zs',10 from dual union

select 1001,'ls',20 from dual union

select 1002,'ww',30 from dual;

## 查看表结构

desc 表名

## 复制表数据

insert into 目标表 select \* from 参考表

## 查看指定用户(SCOTT)下所有的表

SELECT table\_name FROM all\_tables WHERE owner = upper('用户名');

注意：用户名必须大写

## oracle查询用户下的所有表

select table\_name from user\_tables;

select table\_name from tabs;

select \* from all\_tab\_comments -- 查询所有用户的表,视图等

select \* from user\_tab\_comments -- 查询本用户的表,视图等

select \* from all\_col\_comments --查询所有用户的表的列名和注释.

select \* from user\_col\_comments -- 查询本用户的表的列名和注释

select \* from all\_tab\_columns --查询所有用户的表的列名等信息

select \* from user\_tab\_columns --查询本用户的表的列名等信息

--使用1:

select t.table\_name,t.comments from user\_tab\_comments t

--使用2:

select r1, r2, r3, r5

from (select a.table\_name r1, a.column\_name r2, a.comments r3

from user\_col\_comments a),

(select t.table\_name r4, t.comments r5 from user\_tab\_comments t) where r4 = r1

# 列的创建

## 增加字段

alter table 表名 add (字段名称 字段类型)

## 修改字段

alter table 表名 modify (字段名称 字段类型)

注意：

实际上修改指的是对字段类型的大小进行修改，而不是类型本身

如果非在改成其它类型，则需要满足兼容性。或者把对应这列数据全部清除

## 删除字段

alter table 表名 drop column 字段名称

# 表约束管理

Oracle数据库使用约束（constraints）来防止无效的数据进入到表中，保护数据的实体完整性

|  |  |
| --- | --- |
| 约束名称 | 说明 |
| NOT NULL | 指定字段不能包含空值 |
| UNIQUE | 指定字段的值（或字段组合的值）表中所有的行必须唯一 |
| PRIMARY KEY | 表的每行的唯一标识，即主键 |
| FOREIGN KEY | 在字段和引用表的一个字段之间建立并且强制外键关系，即外键 |
| CHECK | 指定一个必须为真的条件 |

## 约束分类

### 表级约束（建表同时建立约束）

CREATE TABLE tb\_test

(

test\_id number(4),

test\_name varchar2(50) unique not null, --唯一、非空

test\_age number(4) not null, --非空

test\_sex char(2) default '男' not null, --默认'男'，非空

constraint pk\_id primary key(test\_id), --主键约束（表级）

constraint ck\_age check(test\_age>0 AND test\_age<100) --检查约束（表级）

)

### 行级约束（建表后建立约束）

--主键约束

--单个主键

alter table 表名 add constraint 约束名称 primary key(字段);

--复合（联合）主键

alter table 表名 add constraint 约束名称 primary key(字段1, 字段2);

--唯一约束

alter table 表名 add constraint 约束名称 unique(约束字段);

--检查约束

alter table 表名 add constraint 约束名 check(stusex='男' or stusex='女');

alter table 表名 add constraint 约束名 check(stuage between 15 and 30);

--外键约束

alter table 子表 add constraint 约束名 foreign key(子表外键字段)

references 主表(主键) 【on delete cascade】;级联 (一起生，一起死)

--非空设置

--只设置一个字段

alter table student modify (stuaddress not null);

--一次设置多个字段

alter table student modify (stuname not null,stusex not null);

--添加默认值

alter table 表名 modify (字段 default '北京');

--添加默认值的同时为字段设为非空

alter table student modify (stuage default 20 not null) ;

## 管理约束

### 查看约束

select constraint\_name,table\_name from user\_constraints where table\_name ='表名';

### 删除约束

ALTER TABLE table

DROP PRIMARY KEY | UNIQUE(字段名) | CONSTRAINT 约束名 [CASCADE]

例：ALTER TABLE stu DROP PRIMARY KEY;

ALTER TABLE table\_name DROP CONSTRAINT ck\_name;

### 禁用约束

ALTER TABLE table\_name DISABLE CONSTRAINT constraint\_name;

### 启用约束

ALTER TABLE table\_name ENABLE CONSTRAINT constraint\_name;

## 四种完整性约束

### 实体完整性

约束方法：唯一约束、主键约束

### 域完整性

约束方法：限制数据类型、检查约束、外键约束、默认值、非空约束

### 引用完整性

约束方法：外键约束

### 自定义完整性

约束方法：规则、存储过程、触发器

## 注意事项

* 如果不指定名，Oracle将自动按照SYS\_Cxxx格式命名
* 约束可以定在行级，也可以定义在表级
* 可以在建表的同时建立约束，也可以在建表后建立约束
* 主键约束包含唯一约束、非空约束
* 唯一约束允许包含空值，而主键约束不允许

# 数据操作语言DML

* 数据操纵语言（Data manipulation language ，DML）是SQL语言的一个核心部分之一，当想要添加、更新或者删除数据库中的数据时，需要执行DML语句
* 严格意义上讲，SELECT查询语句属于数据查询语言DQL，不属于数据操纵语言DML，但在实际工作中，多数研发人员、数据库管理员都将SELECT语句归入数据操纵语言中

## insert语句

--语法

INSERT INTO table[(column1, column2…)] VALUES (value1,value2…)

--插入所有字段一,使用默认时间格式

--录入员工的信息

insert into emp values(6666,'tom','ANALYST',7566,'31-1月-16',4200,500,40);

--插入所有字段二，时间格式自定义(使用to\_date函数)

insert into emp(empno,ename,job,mgr,hiredate,sal,comm,deptno)

values(7777,'jack','ANALYST',7566,to\_date('2016-5-22','yyyy-mm-dd'),4400,800,40);

--插入部份字段,时间格式为系统时间

insert into emp(empno,ename,job,mgr,hiredate,sal,deptno)

values(8888,'lucy','ANALYST',7566,sysdate,4500,40);

--创建员工信息备份表

create table empBack(

empno number(7) primary key,

empName varchar2(20),

empSal number(7,2)

);

--为其它表一次性插入多条数据

--语法(注:table1表一开始要存在)

INSERT INTO table1[(column1, column2…)] SELECT col1, col2… FROM table2…

--备份员工信息表中的所有员工的编号，姓名，工资

insert into empBack(empno,empName,empSal) select empno,ename,sal from emp;

## update语句

--语法

UPDATE table

SET column1 = value1, column2 = value2…

[WHERE…]

--1、把编号为6666的员工奖金修改为600

update emp set comm=600 where empno=6666;

--2、把编号为7777员工的工作岗位、上级领导编号、部门编号与JAMES一样

--select job=?,mgr=?,deptno=? from emp where empno=7777;

update emp set job=(select job from emp where ename='JAMES'),

mgr=(select mgr from emp where ename='JAMES'),

deptno=(select deptno from emp where ename='JAMES')

where empno=7777;

--或者第二种方式

update emp set (job,mgr,deptno)=(select job,mgr,deptno from emp where ename='JAMES') where empno=7777;

--3、为编号为6666、7777、8888的员工工资加500、奖金加200

update emp set sal=sal+500,comm=nvl(comm,0)+200 where empno in (6666,7777,8888);

## delete语句

--语法

DELETE [FROM] table [WHERE…]

--删除emp表,员工编号为7777的员工信息

delete from emp where empno=7777;

--删除emp表部门编号为40的员工信息(注:如果删除的数据有外键则会报错，报的是完整性约束错误)

delete from emp where deptno=40;

--清空表中的所有内容

delete table tableName; --删除所有记录,表结构还在,写日志,可恢复,速度慢

truncate table tableName; --删除表所有记录,结构还在,不写日志,不可恢复,