Oracle数据库实践

# Oracle数据库安装

## 1、数据库简介

数据库是以一定方式储存在一起、能与多个用户共享、具有尽可能小的冗余度、与应用程序彼此独立的数据集合。

数据库可以看作电子化的文件柜，或存储电子文件的处所，用户可以对文件中的数据进行新增，删除，修改和查找等操作。

简单的说，数据库（英文Dtabase）就是一个存放数据的仓库，这个仓库是按照一定的数据结果（数据结构是指数据的组织形式或数据之间的联系）来组织、存储的、我们可以通过数据库提供的多种方法来管理数据库里的数据更简单的形象理解，数据库和我们生活中存放杂物的仓库性质一样，区别只是存放的东西不同。

早期比较流行的数据库模型有三种，分别为层次式数据库、网络式数据库和关系型数据库。而在当今的互联网中，最常用的数据库模型主要是两种，即**关系型数据库**和非关系型数据库（NoSQL）。

## 2、关系数据库简介

网络数据库和层次数据库很好地解决了数据的集中和共享问题，但是在数据独立性和抽象级别上仍有很大欠缺。用户对这两种数据库进行存取时，依然需要明确数据的存储结构，支出存储路径。而关系数据库就可以较好地解决这些问题。

关系型数据库模型是把复杂的数据结构归结为简单的二元关系（即二维表格形式）。  
  Oracle 在数据库领域里上升到了霸主地位，形成每年高达数百亿美元的庞大市场，而MySQL也是不容忽视的数据库，以至于被Oracle重金收购。

生活中的关系型数据库举例：表格。

关系型数据库，简单说就是二位表格：

1. MySQL和Oracle是目前常见的关系型数据库，Oracle在传统企业，MySQL在互联网应用中被广泛使用。
2. 使用SQL（Structured Query Language结构化查询语句）存储数据和查询，更新，管理数据。
3. 遵循事务(Transaction)特性，保证数据的完整，一致和正确。事务（transaction）是用户定义的一个数据库操作序列，这些操作要么全做，要么全不做，是一个不可分割的工作单位。事务的特性包括原子性(Atomicity)、一致性(Consistency)、隔离性(Isolation)和持续性(Durability)，简称ACID特性。

其他关系型数据库举例：Microsoft Access，Microsoft SQL Server，Sybase，Dbase等。 几乎所有的数据库管理系统都配备了一个[开放式数据库连接](https://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E5%BC%8F%E6%95%B0%E6%8D%AE%E5%BA%93%E8%BF%9E%E6%8E%A5)（ODBC）驱动程序，令各个数据库之间得以互相集成。

## 3、数据库管理系统

[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728)管理[系统](https://baike.baidu.com/item/%E7%B3%BB%E7%BB%9F)(Database Management System)是一种操纵和管理数据库的大型软件，用于建立、使用和维护[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728)，简称[DBMS](https://baike.baidu.com/item/DBMS)。它对[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728" \t "_blank)进行统一的管理和[控制](https://baike.baidu.com/item/%E6%8E%A7%E5%88%B6/948689)，以保证[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728)的安全性和完整性。用户通过[DBMS](https://baike.baidu.com/item/DBMS)访问[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728" \t "_blank)中的数据，[数据库管理员](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E5%91%98/1216449)也通过DBMS进行数据库的维护工作。它可使多个[应用程序](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F/5985445" \t "_blank)和用户用不同的方法在同时或不同时刻去建立，修改和询问[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728)。大部分[DBMS](https://baike.baidu.com/item/DBMS)提供[数据定义语言](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%AE%9A%E4%B9%89%E8%AF%AD%E8%A8%80/104009" \t "_blank)[DDL](https://baike.baidu.com/item/DDL/21997)（Data Definition Language）和[数据操作语言](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E6%93%8D%E4%BD%9C%E8%AF%AD%E8%A8%80/10826383" \t "_blank)[DML](https://baike.baidu.com/item/DML)（Data Manipulation Language），供用户定义[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93/103728" \t "_blank)的模式结构与权限约束，实现对数据的追加、删除等操作。

数据库管理系统是数据库系统的核心，是管理数据库的软件。数据库管理系统就是实现把用户意义下抽象的逻辑数据处理，转换成为计算机中具体的物理数据处理的软件。有了数据库管理系统，用户就可以在抽象意义下处理数据，而不必顾及这些数据在计算机中的布局和物理位置。

## 4、ORACLE关系型数据库

ORACLE数据库是[甲骨文公司](https://baike.baidu.com/item/%E7%94%B2%E9%AA%A8%E6%96%87%E5%85%AC%E5%8F%B8/430115)的一款[关系数据库管理系统](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E6%95%B0%E6%8D%AE%E5%BA%93%E7%AE%A1%E7%90%86%E7%B3%BB%E7%BB%9F/11032386)。它是在数据库领域一直处于领先地位的产品。可以说Oracle数据库系统是目前世界上流行的关系数据库管理系统，系统可移植性好、使用方便、功能强，适用于各类大、中、小、微机环境。它是一种高效率、可靠性好的 适应高吞吐量的数据库解决方案。

## 5、Windows环境下ORACLE数据库安装

我们目前介绍安装的版本是：11g简化版，该版本下载地址：<https://www.oracle.com/technetwork/database/database-technologies/express-edition/downloads/index.html>

安装过程很简单，需要注意如下：

1. 整个安装路径都不要出现中文名字。
2. 安装时各种杀毒软件都退出。
3. 安装ORACLE需要占用端口1521和8080。提前查看这些端口是否被占用的方法： netstat –ano，如果被占用，可以查看到占用进程的进程号。

查看端口被谁占用：获得占用端口的进程号码后，利用 tasklist | findstr “<进程号码>”可以查看到进程号码对应的进程名，确定是谁占用了这些端口。

1. oracle数据库是用户式的, 默认安装后有两个管理员账户,sys, system，安装时指定了口令(密码)。**在学习过程设置简单好记，统一建议设置为1234。**

安装后，测试安装是否正确：

打开cmd窗口，运行sqlplus system/1234

能够连接到Oracle数据库表明ORACLE数据库安装成功。

安装常见问题的解决：

1. ‘sqlplus’不是内部或外部命令，也不是可运行的程序。

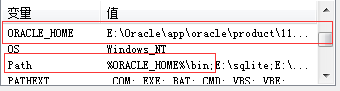
问题所在：环境变量设置有问题。

解决：检查环境变量Path,查看Oracle产品下bin目录的后面，是否有2个”;”。如果有2个分号，删除一个即可。

另外，可以设置一个ORCALE\_HOME的新环境变量，假设Oracle安装目录是E:\Oracle，那么ORACLE\_HOME设置为：

E:\Oracle\app\oracle\product\11.2.0\server

然后在环境变量Path中，增加%ORACLE\_HOME%\bin;（注意分号）

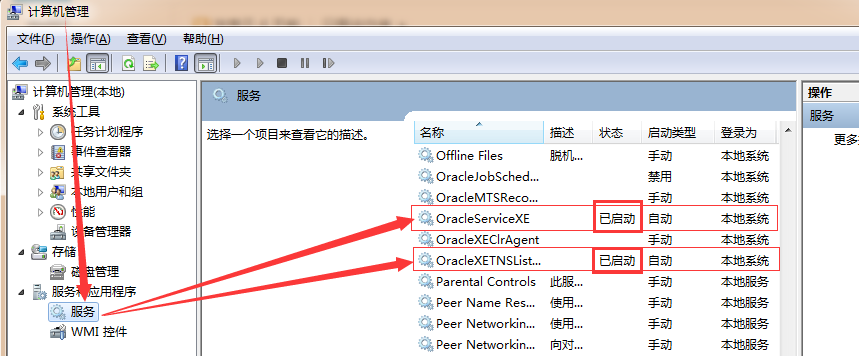


这样的好处是避免了Path太长。

1. Sqlplus连接不到Oracle数据库。

问题所在：Oracle数据库相关的服务没有启动。

解决：右键点击“计算机”，选择“管理”以打开计算机管理。在计算机管理中选择“服务和应用程序”。



注意服务“OracleServiceXE”和” OracleXETNSListener”的状态。状态为“已启动”，才可以连接到Oracle数据库。

如果这两个服务器的启动类型是自动,那么开机时这服务启动可能失败。如果是手动，那么需要手动启动它们。

如果需要卸载Oracle，需要先手动关闭这两个服务，再卸载。否则卸载后，再次安装Oracle会失败，并且问题难以解决。

## 6、sqlplus的基本使用

oracle自带一个sqlplus客户端，命令行式的，适合学习过程中使用。

可以通过sqlplus连接到Oracle数据库，连接需要提供用户名和密码。两种方式：

1. 从DOS命令行连接到Oracle数据库，普通用户或者system都可以采用这种方式连接数据库(sys用户比较特殊)。

命令格式： sqlplus 用户名/口令

或者： sqlplus，再根据提示输入用户名，再根据提示输入口令，口令输入时并不回显。

或者： sqlplus 用户名，再根据提示输入口令，口令输入时不回显。

注： sqlplus /nolog -- 不用密码登录，然后conn到sys。

超级系统管理员sys用户连接到数据库：

A、命令格式： sqlplus / as sysdba

注意上面命令中的空格。sys可以直接在数据库安装系统下免密码登录。注意：仅仅在数据库安装系统环境下。实际上是使用了电脑用户的密码进行了校验。

B、提供密码： sqlplus sys/<口令> as sysdba

可以远程使用sys连接数据库，但必须要提供密码。

连接到数据库后，如何查看用户身份呢：

show user;

从数据库退出：

命令： exit或者quit

清屏命令： clear scre

1. 已经连接到数据库中，切换用户：

命令中用conn替换sqlplus。

断开当前的连接，用新用户连接。

TIPS

|  |
| --- |
| 查看DB 版本：  SQL> select \* from v$version; |

# Oracle数据库用户管理

## 1、创建新用户

创建新用户，需要管理员权限。用户system是系统管理员，而用户sys是超级系统管理员，这两个用户都可以创建新用户。

创建用户的语句：

create user 用户名 identified by 口令;

注意：语句是以分号结尾的。

新创建的用户在没有指定角色的情况下，没有任何权限。

查看用户列表：

select username from dba\_users; (仅仅dba用户可用)

select \* from all\_users;

select \* from user\_users;

## 2、给用户分配权限

**GRANT CONNECT, RESOURCE TO 用户名;  
GRANT SELECT ON 表名 TO 用户名;  
GRANT SELECT, INSERT, DELETE ON表名 TO 用户名1, 用户名2;**

对于Oracle数据库，关于权限和角色有如下几点：

1. Oracle数据库中所用的用户等级是平级的！只是每个用户的权限不同而已！
2. 在一个用户登录后，可以在自己的登录状态下访问其他用户的数据缓冲区、表、以及表的操作！（只要该用户用权限！）
3. 我们可以在一个有创建用户权限的用户(管理员用户)登录后，进行创建（赋予权限等）操作，创建后的用户和当前用户是平级的！权限不同而已！

常见的系统权限，选择部分如下：

CREATE SESSION : 创建会话，允许登录。

CREATE TABLE: 在用户模式中创建表

DROP TABLE: 在用户模式中删除表

CREATE PROCEDURE： 创建存储过程

CREATE USER: 创建用户

DROP USER: 删除用户

实体权限分类  
select, update, insert, alter, index, delete, all  //all包括所有权限  
execute  //执行存储过程权限

Oracle中的角色：CONNECT, RESOURCE和DBA，角色其实就是某些权限的集合体。可以自己创建角色，以实现一些权限的集合。

每种角色的权限大致如下：

**CONNECT角色：** --是授予最终用户的典型权利，最基本的权力，只可以登录ORACLE(仅具有创建SESSION的权限)。

**RESOURCE角色：**--是授予开发人员的。

**DBA角色：**是授予系统管理员的，拥有该角色的用户就能成为系统管理员了，它拥有所有的系统权限

赋予用户权限的语句 : grant <权限> to <用户>

针对具体的实体： grant <权限> on <实体> to <用户>

**对于普通用户：授予connect, resource权限。**

**对于DBA管理用户：授予connect，resource, dba权限。**

我们通过管理员来创建用户Jason：

Create user Jason identified by abc123;

然后，通过管理员给用户Jason赋予一些基本权限：连接数据库，创建表，查询数据等。

**Grant connect, resource to Jason;**

扩展：

如果系统用户system存在表 product :

Grant select, update, insert on product to Jason;

也可以：

Grant all on product to Jason;

查看用户具有哪些系统权限：

SQL> select \* from dba\_role\_privs; (DBA用户)  
SQL> select \* from dba\_sys\_privs; (DBA用户)  
SQL> select \* from role\_sys\_privs;

**查自己拥有哪些系统权限:**

**SQL> select \* from session\_privs;**

扩展：

with admin option 用于[系统](https://www.2cto.com/os/" \t "_blank)权限授权，with grant option 用于对象授权。

给一个用户授予系统权限带上with admin option 时，此用户可把此系统权限授予其他用户或角色，但收回这个用户的系统权限时，这个用户已经授予其他用户或角色的此系统权限不会因传播无效。

## 3、回收用户权限

**REVOKE CONNECT, RESOURCE FROM 用户名;  
REVOKE SELECT ON 表名 FROM 用户名;  
REVOKE SELECT, INSERT, DELETE ON 表名 FROM 用户名1, 用户名2;**

## 4、修改用户密码

修改用户密码的语句：

**alter user 用户名 identified by 新密码；**

如果系统管理员密码忘记了，怎么办？

1. 在数据库安装系统下，用sys免密登录数据库。
2. 利用修改密码来设定的新密码。

## 5、删除用户

要删除用户以及该用户关联的数据库对象，最简单的办法是采用级联删除：

**drop user 用户名 cascade;**

如在管理员用户下，删除Jason用户：

drop user Jason cascade;

在删除用户时可能会碰到无法删除当前连接的用户，这是由于还有数据库连接到该用户，有会话存在，需要先删除会话。

# SQL语句

## 什么是SQL？

* SQL 指结构化查询语言
* SQL 使我们有能力访问数据库
* SQL 是一种 ANSI 的标准计算机语言

SQL是一种标准计算机语言，但存在着很多不同版本的 SQL 语言，为了与 ANSI 标准相兼容，它们必须以相似的方式共同地来支持一些主要的关键词（比如 SELECT、UPDATE、DELETE、INSERT、WHERE 等等）。

除了 SQL 标准之外，大部分 SQL 数据库程序都拥有它们自己的私有扩展！

## SQL 能做什么？

* SQL 面向数据库执行查询
* SQL 可从数据库取回数据
* SQL 可在数据库中插入新的记录
* SQL 可更新数据库中的数据
* SQL 可从数据库删除记录
* SQL 可创建新数据库
* SQL 可在数据库中创建新表
* SQL 可在数据库中创建存储过程
* SQL 可在数据库中创建视图
* SQL 可以设置表、存储过程和视图的权限

## RDBMS

RDBMS 指的是关系型数据库管理系统。

RDBMS 是 SQL 的基础，同样也是所有现代数据库系统的基础，比如 MS SQL Server, IBM DB2, Oracle, MySQL 以及 Microsoft Access。

RDBMS 中的数据存储在被称为表（tables）的数据库对象中。

表是相关的数据项的集合，它由列和行组成。

## SQL语句的3种分类

SQL语言按照实现的功能不同，主要分为3类：

数据控制语言（DCL）

数据定义语言（DDL）

数据操纵语言（DML）

### 数据定义语言（DDL）

用于创建和定义数据库对象，并且将对这些对象的定义保存到数据字典中。通过DDL语句可以创建数据库对象，修改数据库对象和删除数据库对象等。

|  |  |
| --- | --- |
| **常用DDL语句** | **功能说明** |
| **CREATE** | 创建数据库结构 |
| **ALTER** | 修改数据库结构 |
| **DROP** | 删除数据库结构 |
| **RENAME** | 更改数据库对象的名称 |
| **TRUNCATE** | 删除表的全部内容 |

### 数据控制语言（DCL）

数据控制语言用于修改数据库结构的操作权限。

|  |  |
| --- | --- |
| **常用DCL语句** | **功能说明** |
| **GRANT** | 授予其他用户对数据库结构的访问权限 |
| **REVOKE** | 收回用户访问数据库结构的权限 |

### 数据操纵语言（DML）

主要用来处理数据库中的数据内容。允许用户对数据库中的数据进行查询 ，插入，更新和删除等操作。

|  |  |
| --- | --- |
| **常用DML语句** | **功能说明** |
| **SELECT** | 从表或视图中检索数据行 |
| **INSERT** | 插入数据到表或视图 |
| **UPDATE** | 更新数据 |
| **DELETE** | 删除数据 |
| **CALL** | 调用过程 |
| **COMMIT** | 将当前事务所做的更改永久化（写入数据库） |
| **ROLLBACK** | 取消上次提交以来的所有操作 |

## SQL语句规则

编写SQL语句时需要遵循一定的规则：

1.SQL关键字、对象名、和列名不区分大小写。在写SQL语句时，建议关键字大写，而自己取得名字，如表名，字段名等，小写。

2.字符值和日期值要区分大小写。

3.在应用程序中如果SQL语句文本很长，可以将语句分布到多行上，并且可以通过使用跳格和缩进提高代码的可读性

4.语句以分号（;）结束。

# DDL表操作

## CREATE TABLE创建表

表是由行和列组成的。

列：字段，一列中的数据都是同类型的。

行：记录，比如一张学生表中的一行，就表示一个学生的记录。

一个数据库中，可能有多个表，表与表之间可能有关联关系的。

### 创建表的语法

CREATE TABLE <表名>(

列名 类型 [默认值] [约束],

列名 类型 [默认值] [约束],

……

列名 类型 [默认值] [约束],

);

### 创建表的前提条件

1) 要有建表的权限

2) 要有建表的空间

### 命名规则

表名和列名的命名，要遵循如下规则：

1. 组成：字母，数字，\_，$,#,以字母开头。
2. 不区分大小写，建议小写，多个单词用\_分隔。
3. 长度：不能超过30个字符
4. 不能使用系统关键字，通常表名会加前缀，比如t,s，表示是一张表的名字。如用户表，取名 t\_user。
5. 见名知意
6. 同一个用户下不能同时存在相同名字的对象。

### 数据类型

1. **文本型**

**VARCHAR2：可变长度的文本型**

根据实际内容大小开辟空间，比如：VARCHAR2(255),最大可开辟空间255字节，如果存储的数就是“tom”,那么实际只开辟3个字节空间。这样可以节约数据空间。

**CHAR：定长的文本型**

指定长度是多少，就开辟多少空间，不满的空间用空格填充。 比如：CHAR(255)，实际开辟空间一定是255字节，这样操作数据的效率高。

1. **数值型**

**NUMBER类型：可以是整数，也可以是小数**

NUMBER(9):可以存储9位的整数

NUMBER(11,2):有效位是11位，整数位：9位，小数位：2位

1. **时间**

**DATE：精确到秒**

**TIMESTAMP：精确到毫秒**

1. **大数据类型**

上限：2GB

**BLOB：存储大数据量的字节型。**

如一张图片。完全可

**CLOB：存储大数据量的字符型，**以选择把图片存储到本次，把路径存储到数据库**一本小说。**

**练习：**

创建一个学生表，包含姓名,学号,性别,年龄,班级。

|  |
| --- |
| CREATE TABLE t\_student(  name varchar2(50),  stu\_no number(9),  sex number(1),  age number(3),  class\_no number(9)  ); |

### 删除表

DROP TABLE <表名>;

用DROP删除表，不但删除了表的数据，而且删除了表的结构。该命令实际上是把删除的表放到了回收站中。

扩展：彻底删除 DROP TABLE <表名> PURGE; 或者用PURGE TABLE <表名>；删除回收站中指定的表名。

TRUNCATE TABLE <表名>;

删除表中的所有数据，并不是删除表。

### 查看表的结构

DESC <表名>

例如： DESC t\_student

或者：

SELECT COLUMN\_NAME, DATA\_TYPE, DATA\_LENGTH

FROM USER\_TAB\_COLS

WHERE TABLE\_NAME='t\_student';

### 给字段设定默认值

给字段设定默认值的语法：在字段的类型后，DEFAULT <字段的默认值>

设置的默认值并不是一开始创建表的初始值，而是以后插入数据时，不指定列的话，默认值不为null，而是你default后面的值。

### 约束

#### 什么是约束？

约束是强加在表上的规则或条件，确保数据库满足业务规则，保证数据的完整性。当对表进行DML或DDL操作时，如果此操作会造成表中的数据违反约束条件或规则的话，系统就会拒绝执行这个操作。

约束可以是列一级别的，也可以是表级别的。

定义约束时没有给出约束的名字，ORACLE系统将为该约束自动生成一个名字，其格式为SYS\_Cn，其中n为自然数(强烈建议各位在创建表或增加约束时，给约束定义名称。)

列级约束: 在定义列的同时定义约束，

column [CONSTRAINT constraint\_name] constraint\_type

表级约束：在定义了所有列后，再定义约束，这里需要注意，not null约束只能在列级上定义；

column ,...,

[CONSTRAINT constraint\_name] constraint\_type (column,...)

#### 约束的分类

1、NOT NULL 非空约束

如果在列上定义了not null，那么当插入数据时，必须为列提供，数据不能为NULL。约束只能在列级定义，不能在表级定义。

2、UNIQUE 唯一约束

当定义了唯一约束后，该列值是不能重复的，但是可以为null。

3、CHECK 检查约束

用于强制行数据必须满足的条件，假定在sal列上定义了check约束，并要求sal列值在1000～2000之间，如果不在1000～2000之间就会提示出错。

4、PRIMARY KEY 主键约束

用于唯一的标识表行的数据，当定义主键约束后，该列不但不能重复而且不能为NULL。一张表最多只能有一个主键，但是可以由多个unique约束。

创建主键或唯一约束后，ORACLE会自动创建一个与约束同名的索引（UNIQUENES为UNIQUE唯一索引）。需要注意的是：每个表只能有且有一个主键约束。

5、FOREIGN KEY 外键约束

用于定义主表和从表之间的关系，外键约束要定义在从表上，主要则必须具有主键约束或是unique约束，当定义外键约束后，要求外键列数据必须在主表的主键列存在或是为NULL。

用来维护从表（Child Table）和主表（Parent Table）之间的引用完整性. 外键约束是个有争议性的约束，它一方面能够维护数据库的数据一致性，数据的完整性。防止错误的垃圾数据入库； 另外一方面它会增加表插入、更新等SQL性能的额外开销，不少系统里面通过业务逻辑控制来取消外键约束。例如在数据仓库中，就推荐禁用外键约束。

#### 列级约束

重新创建学生表，并且增加如下的列级约束：

姓名， 非空

学号， 唯一

性别， 默认值是0， 并且只能是0或者1

|  |
| --- |
| CREATE TABLE t\_student(  name varchar2(50) NOT NULL,  stu\_no number(9) UNIQUE,  sex number(1) DEFAULT 0 CHECK(sex=0 or sex=1) ,  age number(3),  class\_no number(9)  ); |

#### 约束名字

约束也是一种对象，也有名字。如果没有给约束指定名字，系统将取默认的名字。

建议自己给约束起名字：表名\_列名\_约束类型

扩展：查询当前用户下的所有约束名：

SELECT constraint\_name FROM user\_constraints;

给约束起名字：CONSTRAINT 约束名 约束类型

**实例练习：**

重新创建学生表，增加如下的列级约束，并且给约束取名字：

姓名， 非空

学号， 唯一

性别， 默认值是0， 并且只能是0或者1

|  |
| --- |
| CREATE TABLE t\_student(  name varchar2(50)  CONSTRAINT t\_student\_name\_nn NOT NULL,  stu\_no number(9)  CONSTRAINT t\_student\_stu\_no\_uniq UNIQUE,  sex number(1) DEFAULT 0  CONSTRAINT t\_student\_sex\_check CHECK(sex=0 or sex=1) ,  age number(3),  class\_no number(9)  ); |

#### 在表中增加主键约束

主键约束：非空+唯一，一张表中只能有一个主键。

通常，会给每一张表增加一个与业务无关的字段id,主键约束，可以唯一代表这一行，保证行的唯一性。

|  |
| --- |
| CREATE TABLE t\_student(  id number(9)  CONSTRAINT t\_student\_id\_pk PRIMARY KEY,  name varchar2(50)  CONSTRAINT t\_student\_name\_nn NOT NULL,  stu\_no number(9)  CONSTRAINT t\_student\_stu\_no\_uniq UNIQUE,  sex number(1) DEFAULT 0  CONSTRAINT t\_student\_sex\_check CHECK(sex=0 or sex=1) ,  age number(3),  class\_no number(9)  ); |

#### 表级约束

除了非空，剩下的都可以作为表级约束。

单独一行写约束,注意要指定修饰的列。

|  |
| --- |
| CREATE TABLE t\_student(  id number(9),  name varchar2(50)  CONSTRAINT t\_student\_name\_nn NOT NULL,  stu\_no number(9),  sex number(1) DEFAULT 0,  age number(3),  class\_no number(9),  CONSTRAINT t\_student\_id\_pk PRIMARY KEY(id),  CONSTRAINT t\_student\_stu\_no\_uniq UNIQUE(stu\_no),  CONSTRAINT t\_student\_sex\_check CHECK(sex=0 or sex=1)  ); |

#### 外键约束

需求：

记录每个学生各科考试成绩。

思考：

需要设计几张表？

如何设计这几张表？

设计一张科目表，记录考试的科目： id（主键，科目的编号），科目名

设计一张成绩表，记录每个学生的每个科目的成绩：id(主键)，学生编号（学生表的主键），科目编号（科目表的主键）,成绩

|  |
| --- |
| CREATE TABLE t\_subject(  id NUMBER(5),  name VARCHAR2(255),  CONSTRAINT t\_subject\_id\_pk PRIMARY KEY(id)  ); |

|  |
| --- |
| CREATE TABLE t\_score(  id NUMBER(11),  stu\_id NUMBER(9),  sub\_id NUMBER(5),  score NUMBER(5,2),  CONSTRAINT t\_score\_id\_pk PRIMARY KEY(id)  ); |

表之间的引用：

被引用的表：父表

引用别的表的表：子表

先有父表，再有子表。

表和表的关系

一对一

一对多

多对多

外键约束：

|  |
| --- |
| CREATE TABLE t\_score(  id NUMBER(11),  stu\_id NUMBER(9)  CONSTRAINT t\_score\_stu\_id\_fk REFERENCES t\_student(id), --列级  sub\_id NUMBER(5),  score NUMBER(5,2),  CONSTRAINT t\_score\_id\_pk PRIMARY KEY(id),  CONSTRAINT t\_score\_sub\_id\_fk FOREIGN KEY(sub\_id) REFERENCES t\_subject(id) --表级  ); |

# 数据库范式

设计关系[数据库](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E5%BA%93)时，遵从不同的规范[要求](https://baike.baidu.com/item/%E8%A6%81%E6%B1%82/3598753)，设计出合理的关系型数据库，这些不同的规范要求被称为不同的范式，各种范式呈递次规范，越高的范式数据库冗余越小。

目前[关系数据库](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E6%95%B0%E6%8D%AE%E5%BA%93/1237340" \t "_blank)有六种范式：第一范式（1NF）、第二范式（2NF）、第三范式（3NF）、巴斯-科德范式（BCNF）、[第四范式](https://baike.baidu.com/item/%E7%AC%AC%E5%9B%9B%E8%8C%83%E5%BC%8F/3193985" \t "_blank)(4NF）和[第五范式](https://baike.baidu.com/item/%E7%AC%AC%E4%BA%94%E8%8C%83%E5%BC%8F/5025271" \t "_blank)（5NF，又称完美范式）。满足最低要求的范式是第一范式（1NF）。在第一范式的基础上进一步满足更多规范要求的称为第二范式（2NF），其余范式以次类推。一般说来，数据库只需满足第三范式(3NF）就行了。

所谓第一范式（1NF）是指在[关系模型](https://baike.baidu.com/item/%E5%85%B3%E7%B3%BB%E6%A8%A1%E5%9E%8B" \t "_blank)中，对于添加的一个规范要求，所有的域都应该是原子性的，即数据库表的每一列都是不可分割的原子数据项，而不能是集合，数组，记录等非原子数据项。

第二范式（2NF）是在第一范式（1NF）的基础上建立起来的，即满足第二范式（2NF）必须先满足第一范式（1NF）。

第二范式（2NF）要求数据库表中的每个实例或记录必须可以被唯一地区分。选取一个能区分每个实体的属性或属性组，作为实体的唯一标识。

例如在员工表中的身份证号码即可实现每个一员工的区分，该身份证号码即为候选键，任何一个候选键都可以被选作主键。在找不到候选键时，可额外增加属性以实现区分，如果在员工关系中，没有对其身份证号进行存储，而姓名可能会在数据库运行的某个时间重复，无法区分出实体时，设计如ID等不重复的编号以实现区分，被添加的编号或ID选作主键。

第二范式（2NF）要求实体的属性完全依赖于主关键字。所谓完全依赖是指不能存在仅依赖主关键字一部分的属性，如果存在，那么这个属性和主关键字的这一部分应该分离出来形成一个新的实体，新实体与原实体之间是一对多的关系。

第三范式（3NF）是第二范式（2NF）的一个子集，即满足第三范式（3NF）必须满足第二范式（2NF）。简而言之，第三范式（3NF）要求一个关系中不包含已在其它关系已包含的非主关键字信息。

例如，存在一个部门信息表，其中每个部门有部门编号（dept\_id）、部门名称、部门简介等信息。那么在员工信息表中列出部门编号后就不能再将部门名称、部门简介等与部门有关的信息再加入员工信息表中。如果不存在部门信息表，则根据第三范式（3NF）也应该构建它，否则就会有大量的数据冗余。

简而言之，第三范式就是属性不依赖于其它非主属性，也就是在满足2NF的基础上，任何非主属性不得传递依赖于主属性。

实例练习：

设计一个学校的学生系统，要求系统中包含这些信息：学号、学生姓名、年龄、性别、课程名称、课程学分、系别、学科成绩，系办地址、系办电话等信息。

思考：

学生有哪些基本信息？

学生选了哪些课？这些课的成绩是多少？

每个课的学分是多少？

学生是哪个系？系的基本信息是什么？

第一范式：

数据库表中的字段都是单一属性的，不可再分。这个单一属性由基本类型构成，包括整型、[实数](https://baike.baidu.com/item/%E5%AE%9E%E6%95%B0" \t "_blank)、字符型、逻辑型、日期型等。在当前的任何关系数据库管理系统（DBMS）中，傻瓜也不可能做出不符合第一范式的数据库，因为这些DBMS不允许你把数据库表的一列再分成二列或多列。因此，你想在现有的DBMS中设计出不符合第一范式的数据库都是不可能的。

第二范式：

如果把所有这些信息放到一个表中，会产生如下问题:

依赖： 课程名称并不依赖学号。姓名，年龄不依赖课程名称。因此不满足第二范式要求。（用学号做主关键字，课程名等依赖主关键字。用课程做主关键字，学号，年龄等不依赖主关键字）。

数据冗余：同一门课程由n个学生选修，"学分"就重复n-1次；同一个学生选修了m门课程，姓名和年龄就重复了m-1次。

更新异常：如果某门课的学分做了调整，会导致表中所有行的学分都需要调整。如果新增了一门课，再还没有学生选修的情况下，但因为没有学号关联，无法录入系统。

解决方法：

分拆为3张表，学生表，课程表，学生课程关联表。

学生：Student（学号，姓名，年龄，性别，系别，系办地址、系办电话）；

课程：Course（课程名称,学分）；

关联表：SelectCourse（学号，课程名称，成绩）。

第三范式：

上面的学生表Student（学号，姓名，年龄，性别，系别，系办地址、系办电话），关键字为单一关键字"学号"，因为存在如下决定关系：

（学号）→ （姓名，年龄，性别，系别，系办地址、系办电话

但是还存在下面的决定关系：

（学号） → (系别）→（系办地点，系办电话）

即存在非关键字段"系办地点"、"系办电话"对关键字段"学号"的传递依赖。

根据第三范式把学生关系表分为如下两个表就可以满足第三范式了：

学生：（学号，姓名，年龄，性别，系别）；

系别：（系别，系办地址、系办电话）。

# SQL脚本

## 什么是SQL脚本？

脚本（Script）是[批处理文件](https://baike.baidu.com/item/%E6%89%B9%E5%A4%84%E7%90%86%E6%96%87%E4%BB%B6/5363369" \t "_blank)的延伸，是一种纯文本保存的程序，一般来说的计算机脚本程序是确定的一系列控制计算机进行运算操作动作的组合，在其中可以实现一定的逻辑分支等。

SQL脚本是包含一到多个SQL命令的SQL语句，我们可以将这些SQL脚本放在一个文本文件中（我们称之为“SQL脚本文件”），然后通过相关的命令执行这个SQL脚本文件。

## 创建SQL脚本

文件中包含一个或多个SQL语句，每条语句最后以;结尾，使用“-- ”进行注释。文件建议使用sql作为扩展名。

示例如下：

exam01.sql

|  |
| --- |
| CREATE TABLE t\_student(  id number(9),  name varchar2(50)  CONSTRAINT t\_student\_name\_nn NOT NULL,  stu\_no number(9),  sex number(1) DEFAULT 0,  age number(3),  class\_no number(9),  CONSTRAINT t\_student\_id\_pk PRIMARY KEY(id),  CONSTRAINT t\_student\_stu\_no\_uniq UNIQUE(stu\_no),  CONSTRAINT t\_student\_sex\_check CHECK(sex=0 or sex=1)  );  CREATE TABLE t\_subject(  id NUMBER(5),  name VARCHAR2(255),  CONSTRAINT t\_subject\_id\_pk PRIMARY KEY(id)  );  CREATE TABLE t\_score(  id NUMBER(11),  stu\_id NUMBER(9)  CONSTRAINT t\_score\_stu\_id\_fk REFERENCES t\_student(id), --列级  sub\_id NUMBER(5),  score NUMBER(5,2),  CONSTRAINT t\_score\_id\_pk PRIMARY KEY(id),  CONSTRAINT t\_score\_sub\_id\_fk **FOREIGN KEY**(sub\_id) REFERENCES t\_subject(id) --表级  ); |

## 执行SQL脚本

通过命令行执行sqlplus命令，连接进入到Oracle数据库环境后，通过如下两种方式执行SQL脚本：

1. start <脚本文件>
2. @<脚本文件>

注意脚本文件包含了路径。如的d:\exam01.sql

执行命令：

1. start d:\exam01.sql
2. @d:\exam01.sql

执行脚本oaec.sql的关键：

1. 创建用户 oaec/oaec
2. 授权connect,resource给oaec
3. 切换用户到oaec: conn oaec/oaec
4. 简单当前的日期格式是不是英文的：

select sysdate from dual;

如果看到的是 01-MAR-19 就表明已经是英文格式了。

如果看到的是 01-3月-19，就表明是中文简体格式。在跑脚本前需要转换。

alter session set nls\_date\_language=English;

1. 运行oaec.sql脚本：

start <路径>\oaec.sql

或者： @<路径>\oaec.sql

请确保运行结果正确。

1. 请确保是在oaec这个用户下跑脚本的。

如果脚本过程中失败了，请切换到system用户下，把oaec用户删除掉。

Drop user oaec **cascade;**

**然后再从1开始。**

下次课前演讲内容:

1.作业

2.表的创建(表与表之间的关系)

3.约束(4个)