

# Oracle应用开发实战

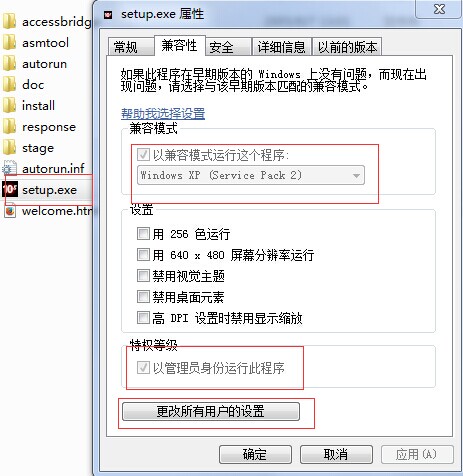
# oracle介绍

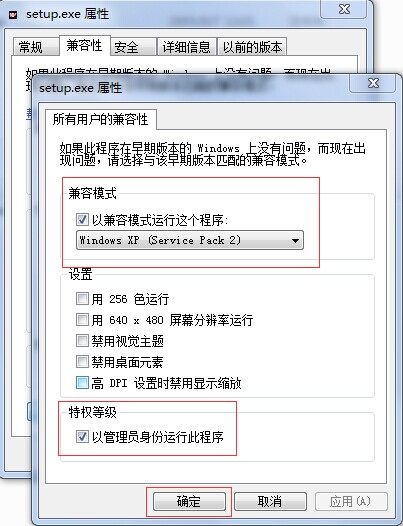
ORACLE[数据库系统](http://baike.baidu.com/view/7809.htm)是美国ORACLE公司（甲骨文）提供的以[分布式数据库](http://baike.baidu.com/view/68389.htm" \t "_blank)为核心的一组软件产品，是目前最流行的客户/服务器(CLIENT/SERVER)或B/S[体系结构](http://baike.baidu.com/view/1188494.htm)的数据库之一。比如SilverStream就是基于数据库的一种中间件。ORACLE数据库是目前世界上使用最为广泛的[数据库管理系统](http://baike.baidu.com/view/68446.htm" \t "_blank)，作为一个通用的[数据库系统](http://baike.baidu.com/view/7809.htm)，它具有完整的[数据管理](http://baike.baidu.com/view/14717.htm)功能；作为一个关系数据库，它是一个完备关系的产品；作为[分布式数据库](http://baike.baidu.com/view/68389.htm)它实现了[分布式处理](http://baike.baidu.com/view/185352.htm)功能。但它的所有知识，只要在一种机型上学习了ORACLE知识，便能在各种类型的机器上使用它。

# oracle10g安装

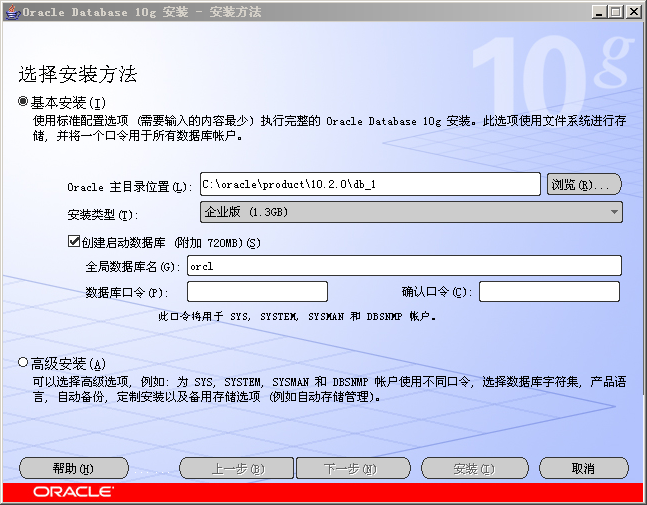
参考：下的安装说明书。

1. 解压oracle数据库安装包，如果是win7或者win8系统右键点击setup.exe选择兼容性，以xp方式，并且以管理员方式运行，以及其他所有用户都按着此规则如图



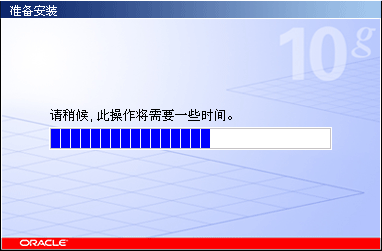


1. 并双击解压目录下的setup.exe，出现安装界面，如下：

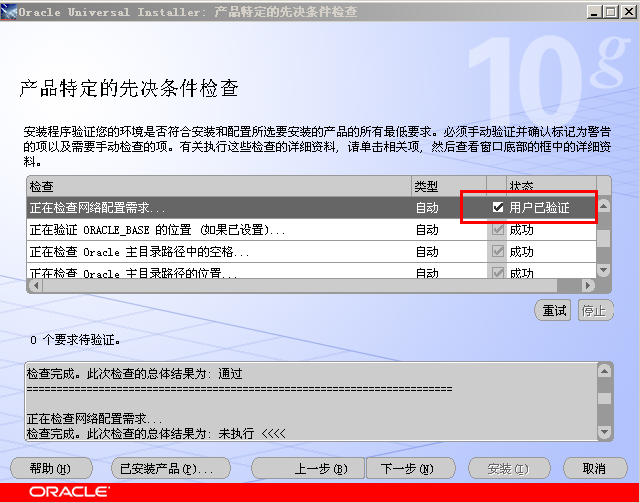


1. 输入口令和确认口令，如：password，点击下一步，出现如下进度条，

注：此口令即是管理员密码。



1. 检查先决条件，选中红框所示的选择框，如下图：



1. 点击“下一步”，出现“概要”界面，点击“安装”。



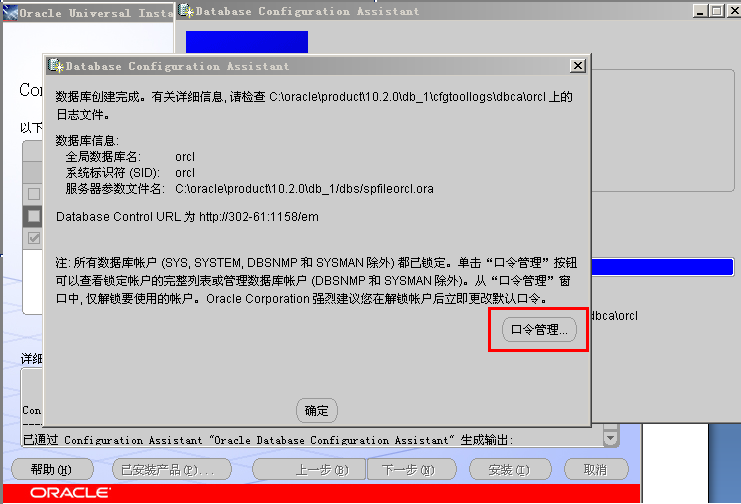
1. 出现安装进度条，等待安装完成，如下图：



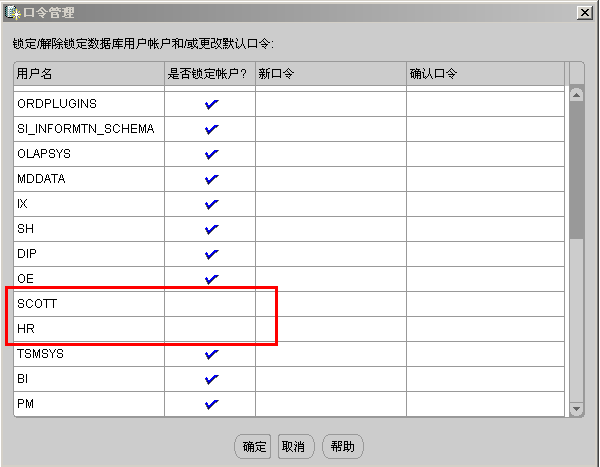
1. 安装完成后，自动运行配置向导，如下图，等待其完成：



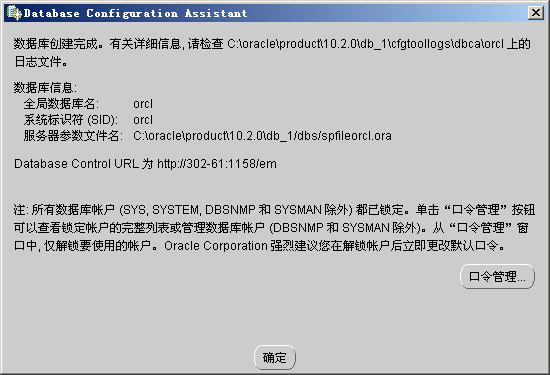
1. 完成后，出现“口令管理”界面，点击“口令管理”，如下图：



1. 将SCOTT和HR用户的沟去掉（解锁这两个账户），如下图所示，点击“确定”：



1. 回到“口令管理”界面，点击“确定”，如下图：



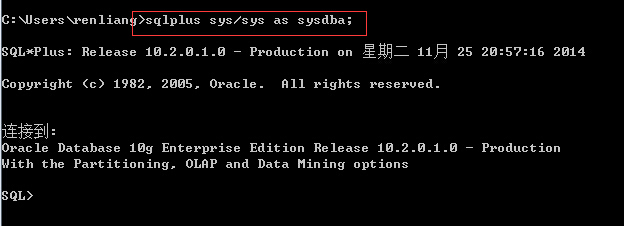
1. 安装结束，点击“退出”。



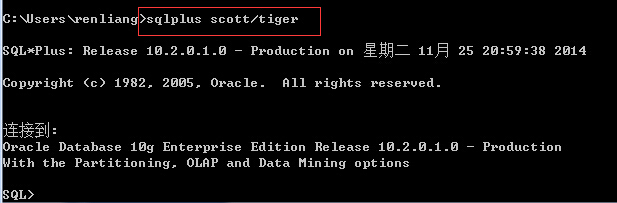
## 登录

运行cmd进入命令行

Sqlplus 用户名/密码@ip地址:1521/orcl [as sysdba]



如果是超级管理员需要在用户名/密码后面加上 as sysdba,是以系统管理员的身份来登录的，如图。



如果是普通用户不需要as sysdba

## 查看当前连接数据库的用户

使用show user查看当前的用户



## 用户的切换

在登录的状态下输入：conn 用户名/密码 [as sysdba]

如图：

* 切换为超级管理员

C:\Users\renliang\AppData\Roaming\Tencent\Users\491793587\QQ\WinTemp\RichOle\4405G$FR0O@98)R(Q7]{6PA.jpg

* 切换为普通的用户



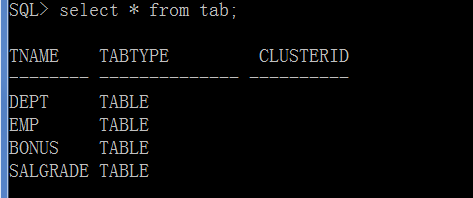
## 查看用户下的表

为了防止命令行中数据展示表格错乱的情况可以设计行宽和列宽

set linesize 300; 每行展示300个字符

Col 列名 for a[数字]，某一列占几个字符

在用户登录的情况下输入：select \* from tab;



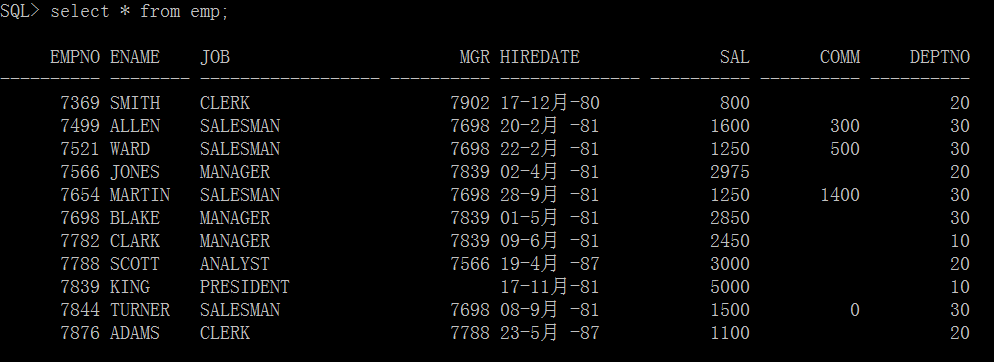
查看表中的数据

Col ename for a8

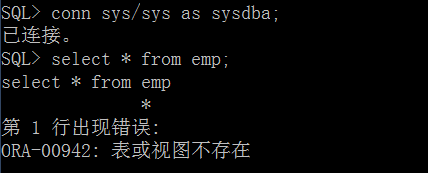
Col mgr for 9999

Col sal for 9999

Select \* from emp;

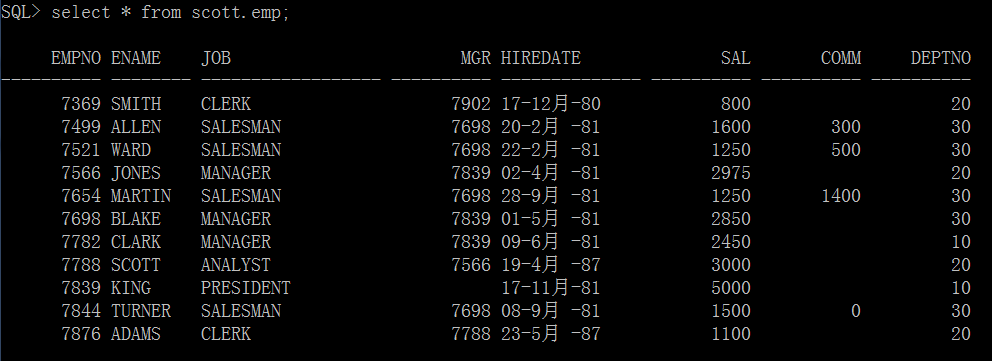


如果使用dba用户登录查询该表无法查看到，因为sys用户下没有这个emp表



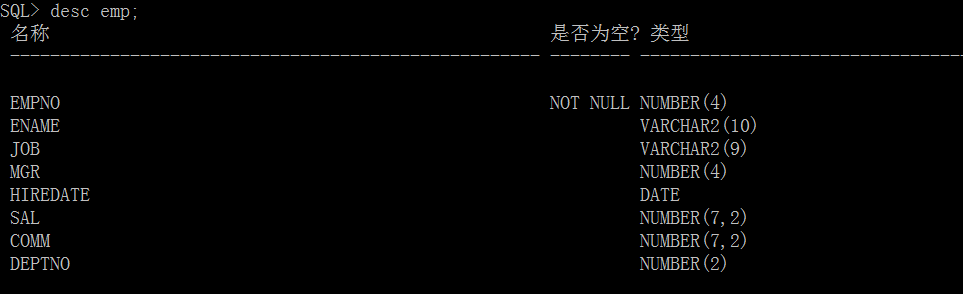
由于超级管理员的权限很大可以查看Scott用户下的表

可以使用select \* from [用户名].[表名] 来查看某用户下的表的数据



## 查看表的结构

Desc 表名



Number(4) 最大长度为4为的数值类型

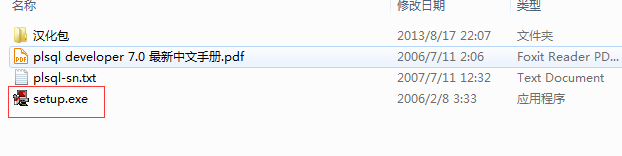
Varchar2(10) 最大长度为10的字符串，varchar2用于存储可变长度的字符串，.varchar2把所有字符都占两字节处理(一般情况下)，varchar只对汉字和全角等字符占两字节，数字，英文字符等都是一个字节，VARCHAR2把空串等同于null处理，而varchar仍按照空串处理；建议在oracle中使用varchar2

Number(7,2) 数值类型整数位占5位，小数位占2位，一共7位。

Date 时间类型

# PLSQL Developer安装

傻瓜式安装即可，不建议汉化，最后会提示输入序列号，打开plsql-sn.txt复制序列号，输入即可。重要提示：不要把plsql developer安装在有括号的目录下，否则会出现连不上数据库的情况，安装目录也不要出现中文。



Plsql developer是最好用也是公司里面大量使用的数据库图形化工具，功能十分强大，在开发人员中广受欢迎，基本上是使用oracle数据库的必备神器。Oracle支持命令行方式操作，在实际的开发中很少使用命令行，绝大多数使用图形化的工具Plsql developer

# Oracle体系结构

* **数据库：**

Oracle数据库是数据的物理存储。这就包括（数据文件ORA或者DBF、控制文件、联机日志、参数文件）。其实Oracle数据库的概念和其它数据库不一样，这里的数据库是一个操作系统只有一个库。可以看作是Oracle就只有一个大数据库。

* **实例：**

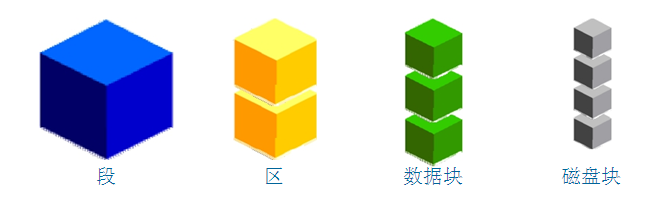
一个Oracle实例（Oracle Instance）有一系列的后台进程（Backguound Processes)和内存结构（Memory Structures)组成。一个数据库可以有n个实例。

* **用户：**

用户是在实例下建立的。不同实例可以建相同名字的用户。

* **表空间：**

表空间是Oracle对物理数据库上相关数据文件（ORA或者DBF文件）的逻辑映射。一个数据库在逻辑上被划分成一到若干个表空间，每个表空间包含了在逻辑上相关联的一组结构。每个数据库至少有一个表空间(称之为system表空间)。  
  
每个表空间由同一磁盘上的一个或多个文件组成，这些文件叫数据文件(datafile)。一个数据文件只能属于一个表空间。

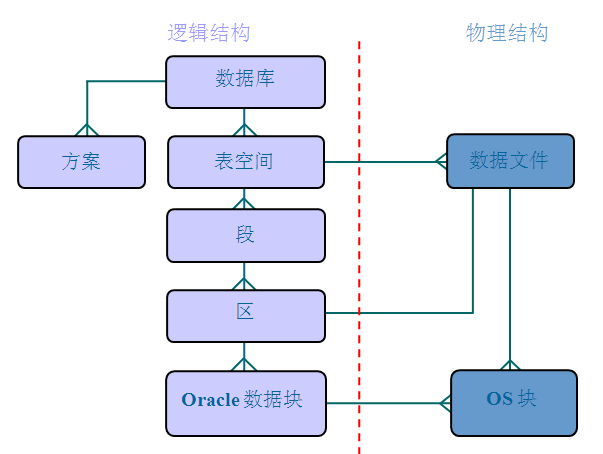


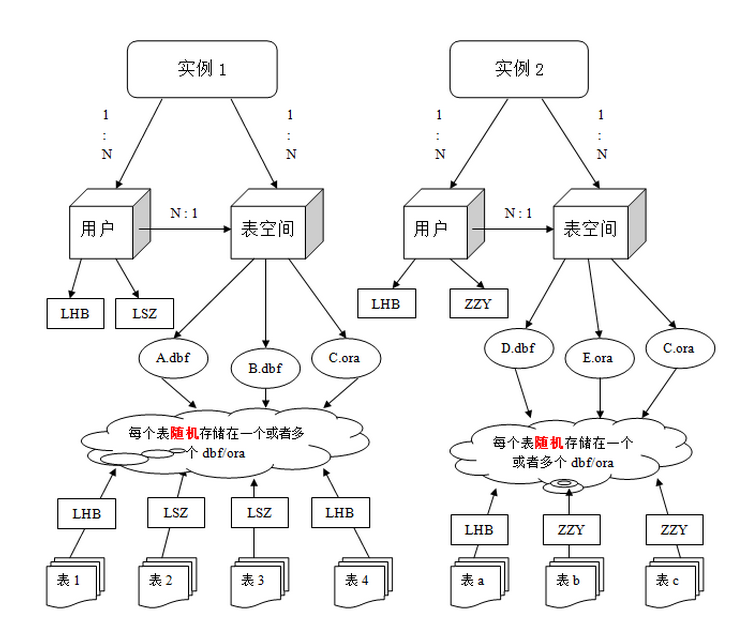
* **数据文件（dbf、ora）:**

数据文件是数据库的物理存储单位。数据库的数据是存储在表空间中的，真正是在某一个或者多个数据文件中。而一个表空间可以由一个或多个数据文件组成，一个数据文件只能属于一个表空间。一旦数据文件被加入到某个表空间后，就不能删除这个文件，如果要删除某个数据文件，只能删除其所属于的表空间才行。

注： 表的数据，是有用户放入某一个表空间的，而这个表空间会随机把这些表数据放到一个或者多个数据文件中。

由于oracle的数据库不是普通的概念，oracle是有用户和表空间对数据进行管理和存放的。但是表不是有表空间去查询的，而是由用户去查的。因为不同用户可以在同一个表空间建立同一个名字的表！这里区分就是用户了！





# 创建表空间

表空间？ ORACLE数据库的逻辑单元。 数据库---表空间 一个表空间可以与多个数据文件（物理结构）关联

一个数据库下可以建立多个表空间，一个表空间可以建立多个用户、一个用户下可以建立多个表。

|  |
| --- |
| create tablespace itcast001  datafile 'c:\itcast001.dbf'  size 100m  autoextend on  next 10m |

itcast 为表空间名称

datafile 指定表空间对应的数据文件

size 后定义的是表空间的初始大小

autoextend on 自动增长 ，当表空间存储都占满时，自动增长

next 后指定的是一次自动增长的大小。

# 用户

## 1创建用户

|  |
| --- |
| create user itcastuser  identified by itcast  default tablespace itcast001 |

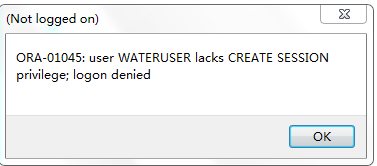
identified by 后边是用户的密码

default tablespace 后边是表空间名称

oracle数据库与其它数据库产品的区别在于，表和其它的数据库对象都是存储在用户下的。

## 2用户赋权限

新创建的用户没有任何权限，登陆后会提示



Oracle中已存在三个重要的角色：connect角色，resource角色，dba角色。

CONNECT角色： --是授予最终用户的典型权利，最基本的

ALTER SESSION --修改会话

CREATE CLUSTER --建立聚簇

CREATE DATABASE LINK --建立数据库链接

CREATE SEQUENCE --建立序列

CREATE SESSION --建立会话

CREATE SYNONYM --建立同义词

CREATE VIEW --建立视图

RESOURCE角色： --是授予开发人员的

CREATE CLUSTER --建立聚簇

CREATE PROCEDURE --建立过程

CREATE SEQUENCE --建立序列

CREATE TABLE --建表

CREATE TRIGGER --建立触发器

CREATE TYPE --建立类型

DBA角色：拥有全部特权，是系统最高权限，只有DBA才可以创建数据库结构，并且系统权限也需要DBA授出，且DBA用户可以操作全体用户的任意基表，包括删除

|  |
| --- |
| grant dba to itcastuser |

进入system用户下给用户赋予dba权限，否则无法正常登陆

## 3中文乱码问题解决

1.查看服务器端编码

select userenv('language') from dual;

我实际查到的结果为:AMERICAN\_AMERICA.ZHS16GBK

2.执行语句 select \* from V$NLS\_PARAMETERS

查看第一行中PARAMETER项中为NLS\_LANGUAGE 对应的VALUE项中是否和第一步得到的值一样。

如果不是，需要设置环境变量.

否则PLSQL客户端使用的编码和服务器端编码不一致,插入中文时就会出现乱码.

3.设置环境变量

计算机->属性->高级系统设置->环境变量->新建

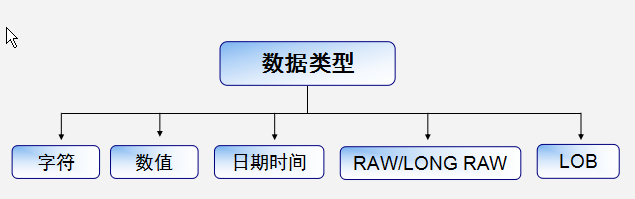
设置变量名:NLS\_LANG,变量值:第1步查到的值， 我的是

SIMPLIFIED CHINESE\_CHINA.ZHS16GBK

4.重新启动PLSQL,插入数据正常

# 7.Oracle数据类型

## 1.oracle 数据类型



## 2.常用的数据类型

|  |  |  |
| --- | --- | --- |
| No | 数据类型 | 描述 |
| 1 | Varchar， varchar2 | 表示一个字符串 |
| 2 | NUMBER | NUMBER(n)表示一个整数，长度是n |
| NUMBER(m,n):表示一个小数，总长度是m，小数是n，整数是m-n  Number(10,2):整数部分占 8 位，小数部分占2份 |
| 3 | DATE | 表示日期类型 |
| 4 | CLOB | 大对象，表示大文本数据类型，可存4G |
| 5 | BLOB | 大对象，表示二进制数据，可存4G |

# 8.表的管理

### 1建表

语法：

Create table 表名（

字段1 数据类型 [default 默认值],

字段2 数据类型 [default 默认值],

...

字段n 数据类型 [default 默认值]

）;

范例：创建person表

create table person(

pid number(10),

name varchar2(10),

gender number(1) default 1,

birthday date

);

insert into person(pid, name, gender, birthday)

values(1, '张三', 1, to\_date('1999-12-22', 'yyyy-MM-dd'));

### 2.表删除

语法：DROP TABLE 表名

### 3.表的修改

在sql中使用alter可以修改表

* 添加语法：ALTER TABLE 表名称 ADD(列名1 类型 [DEFAULT 默认值]，列名1 类型 [DEFAULT 默认值]...)
* 修改语法：ALTER TABLE 表名称 MODIFY(列名1 类型 [DEFAULT 默认值]，列名1 类型 [DEFAULT 默认值]...)
* 修改列名: ALTER TABLE 表名称 RENAME COLUMN 列名1 TO 列名2

范例：在person表中增加列address

alter table person add(address varchar2(10));

范例：把person表的address列的长度修改成20长度

alter table person modify(address varchar2(20));

### 4.截断表

在person表中使用delete语句删除数据，则可以使用rollback来回滚，如果想要清空一张表的数据，同时不想回滚可以立即释放资源需要使用截断表的语法

语法：TRUNCATE TABLE 表名

范例：截断person表

truncate table person;

### 5.数据库表数据的更新

#### INSERT（增加）

语法：

INSERT INTO表名[(列名1，列名2，...)]VALUES(值1，值2，...)

标准写法



简单写法（不建议）

INSERT INTO 表名VALUES(值1，值2，...)

insert into **person**

values(1,'张三',1,'9-5月-1981','北京北七家');

注意：使用简单的写法必须按照表中的字段的顺序来插入值，而且如果有为空的字段使用null

insert into person

values(2,'李四',1,null,'北京育新');

#### 2.事务

这是因为oracle的事务对数据库的变更的处理，我们必须做提交事务才能让数据真正的插入到数据库中，在同样在执行完数据库变更的操作后还可以把事务进行回滚，这样就不会插入到数据库。如果事务提交后则不可以再回滚。

提交：commit

回滚：rollback

##### 1.事务保存点与回滚：

语法：

savepoint 保存点名字

示例：

savepoint a1;    --设置一个保存点 a1;

update person set name='张三' where pid='101'  --修改一条数据

savepoint a2;   --设置一个保存点 a1;

update tmp set name='李四' where id='102  --再修改一条数据

rollback to a2;   --回滚到a2 保存点。 此时在 a2 保存点之后的所有数据修改视为无效。

rollback to a1;   --这里可以再从a2保存点再回滚到a1 保存点。 此时在 a1 保存点之后的所有数据修改视为无效。

rollback;           --回滚全部。即撤消至上一次提交事务后的所有数据修改。

commit;  --提交事务 将回滚后的事务提交，并会删除所有保存点。

注意：我们可以从a2向前再回滚到a1 ，但无法从a1回滚到a2。也就是只能向前回滚，不能从前面的点再向后回滚！

##### 2．事务的隔离级别：

事务的隔离性是什么？

事务隔离级别有哪些？

Oracle支持的隔离级别是什么？

#### 3.UPDATE（修改）

全部修改：UPDATE 表名 SET 列名1=值1，列名2=值2，....

局部修改：UPDATE 表名 SET 列名1=值1，列名2=值2，....WHERE 修改条件；

全部更新

局部更新

#### DELETE（删除）

语法 : DELETE FROM 表名 WHERE 删除条件;

在删除语句中如果不指定删除条件的话就会删除所有的数据

Truncate table实现数据删除

比较truncat与delete实现数据删除？

1. delete删除的数据可以rollback,也可以闪回
2. delete删除可能产生碎片，并且不释放空间
3. truncate是先摧毁表结构，再重构表结构

## 4.通过闪回实现数据恢复

### 1.闪回的原理

### 2.闪回的实现方式

### 3.闪回的注意事项

## 5. 碎片问题的思考

### 1.什么是碎片

### 2.碎片的产生

### 3.碎片问题的避免与解决

# 9.序列

在很多数据库中都存在一个自动增长的列,如果现在要想在oracle 中完成自动增长的功能, 则只能依靠序列完成,所有的自动增长操作,需要用户手工完成处理。

语法：CREATE SEQUENCE 序列名

[INCREMENT BY n]

[START WITH n]

[{MAXVALUE/ MINVALUE n|NOMAXVALUE}]

[{CYCLE|NOCYCLE}]

[{CACHE n|NOCACHE}];

范例:创建一个seqpersonid的序列,验证自动增长的操作

CREATE SEQUENCE seqpersonid;

序列创建完成之后,所有的自动增长应该由用户自己处理,所以在序列中提供了以下的两种操作：

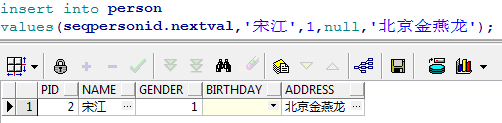
nextval :取得序列的下一个内容

currval :取得序列的当前内容

select seqpersonid.nextval from dual;

select seqpersonid.currval from dual;

在插入数据时需要自增的主键中可以这样使用



在实际项目中每一张表会配一个序列，但是表和序列是没有必然的联系的，一个序列被哪一张表使用都可以，但是我们一般都是一张表用一个序列。

序列的管理一般使用工具来管理。

# 10.约束

在数据库开发中，约束是必不可少，使用约束可以更好的保证数据的完整性。

### 1.主键约束

主键约束都是在id上使用，而且本身已经默认了内容不能为空，可以在建表的时候指定。

创建一张表，把pid作为主键

create table person(

pid number(10) primary key,

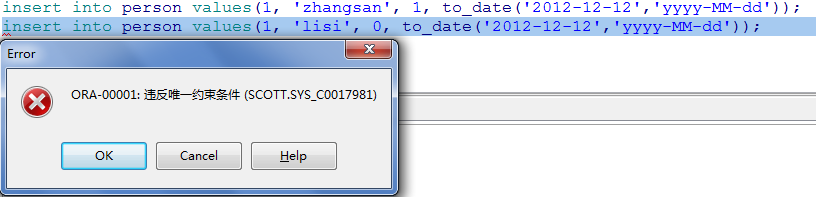
name varchar2(10),

gender number(1) default 1,

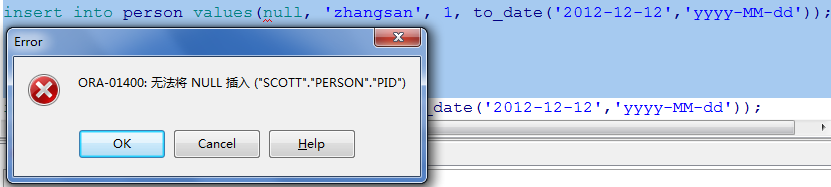
birthday date

);

主键不可重复， SCOTT.SYS\_C0017981是系统自动分配的约束的名字



主键不可为空



我们可以自己来指定主键约束的名字

create table person(

pid number(10),

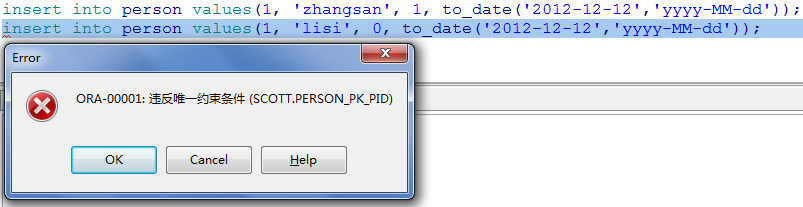
name varchar2(10),

gender number(1) default 1,

birthday date,

constraint person\_pk\_pid primary key(pid)

);



### 2.非空约束

使用非空约束，可以使指定的字段不可以为空。

范例：建立一张pid和name不可以为空的表

create table person(

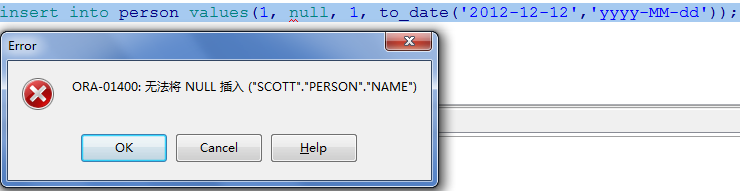
pid number(10) not null,

name varchar2(10) not null,

gender number(1) ,

birthday date,

);



### 3.唯一约束（unique）

表中的一个字段的内容是唯一的

范例：建表一个name是唯一的表

create table person(

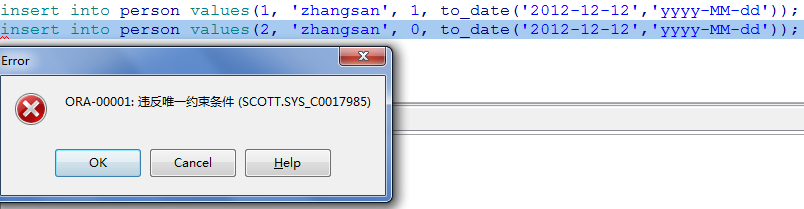
pid number(10) ,

name varchar2(10) unique,

gender number(1) ,

birthday date

);



唯一约束的名字也可以自定义

create table person(

pid number(10) ,

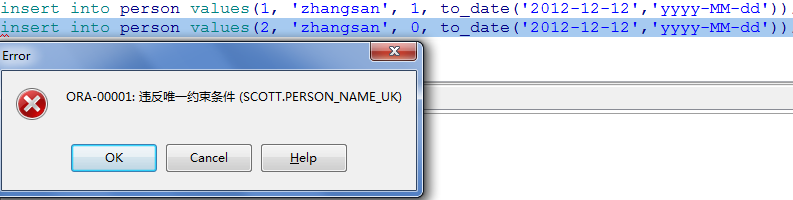
name varchar2(10),

gender number(1) ,

birthday date,

constraint person\_name\_uk unique(name)

);



### 4.检查约束

使用检查约束可以来约束字段值的合法范围。

范例：创建一张表性别只能是1或2

create table person(

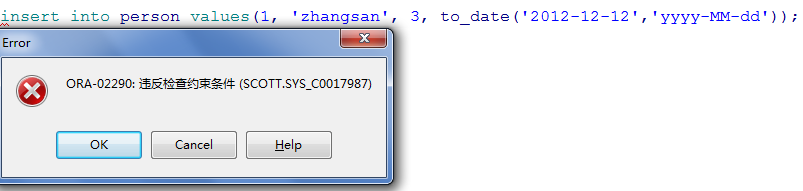
pid number(10) ,

name varchar2(10),

gender number(1) check(gender in (1, 2)),

birthday date

);



检查约束也可以自定义

create table person(

pid number(10) ,

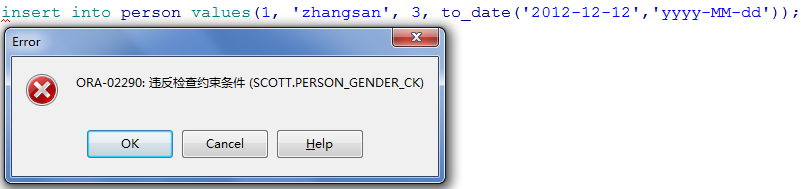
name varchar2(10),

gender number(1),

birthday date,

constraint person\_gender\_ck check(gender in (1,2))

);



### 5.外键约束

之前所讲的都是单表的约束，外键是两张表的约束，可以保证关联数据的完整性。

范例：创建两张表，一张订单表，一张是订单明细表，订单和明细是一对多的关系

create table orders(

order\_id number(10) ,

total\_price number(10,2),

order\_time date,

constraint orders\_order\_id\_pk primary key(order\_id)

);

create table order\_detail(

detail\_id number(10) ,

order\_id number(10),

item\_name varchar2(10),

quantity number(10),

constraint order\_detail\_detail\_id\_pk primary key(detail\_id)

);

insert into orders values(1, 200, to\_date('2015-12-12','yyyy-MM-dd'));

insert into order\_detail values(1, 2, 'java',1);

我们在两张表中插入如上两条数据，我们发现在order\_detail表中插入的order\_id在order表中并不存在，这样在数据库中就产生了脏数据。此时需要外键来约束它。

我们再次建表

create table orders(

order\_id number(10) ,

total\_price number(10,2),

order\_time date,

constraint orders\_order\_id\_pk primary key(order\_id)

);

create table order\_detail(

detail\_id number(10) ,

order\_id number(10),

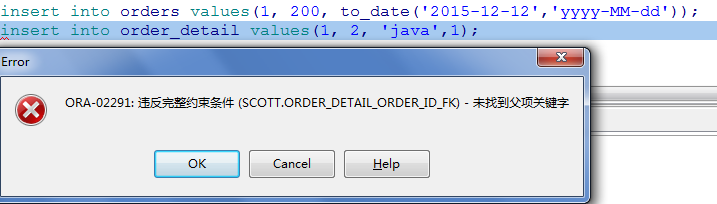
item\_name varchar2(10),

quantity number(10),

constraint order\_detail\_detail\_id\_pk primary key(detail\_id),

constraint order\_detail\_order\_id\_fk foreign key(order\_id) references orders(order\_id)

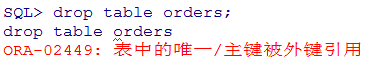
);



外键关联一定注意：

外键一定是主表的主键

删表时一定先删子表再删主表，如果直接删主表会出现由于约束存在无法删除的问题



但是可以强制删除drop table orders cascade constraint;(不建议)

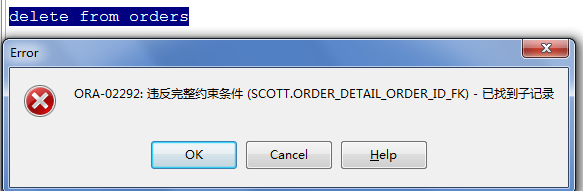
删除主表的数据可以先删除子表的关联数据，再删主表，也可以使用级联删除。

级联删除在外键约束上要加上on delete cascade 如

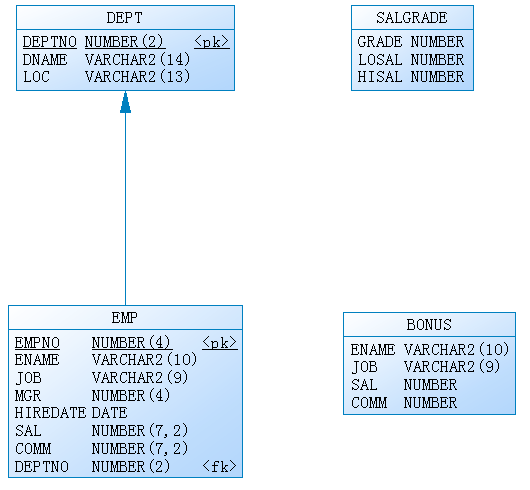
constraint order\_detail\_order\_id\_fk foreign key(order\_id)

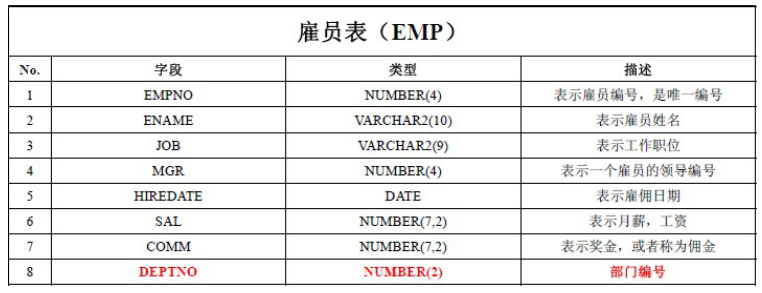
references orders(order\_id) on delete cascade

这样删除主表数据的时候会把字表的关联数据一同删除



# 11.Scott用户下的表结构











# 12. sql

## 1.sql简介

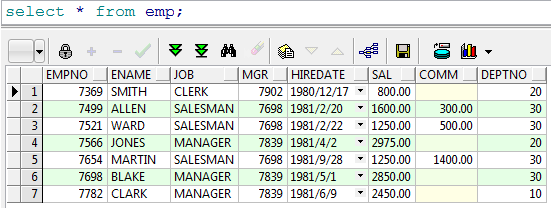
结构化查询语言(Structured Query Language)简称SQL(发音：/ˈɛs kjuː ˈɛl/ "S-Q-L")，结构化查询语言是一种数据库查询和程序设计语言，用于存取数据以及查询、更新和管理[关系数据库系统](http://baike.baidu.com/view/549699.htm" \t "_blank)；同时也是[数据库脚本文件](http://baike.baidu.com/view/3542225.htm)的扩展名。结构化查询语言是高级的非过程化编程语言，允许用户在高层[数据结构](http://baike.baidu.com/view/9900.htm)上工作。它不要求用户指定对数据的存放方法，也不需要用户了解具体的数据存放方式，所以具有完全不同底层结构的不同[数据库系统](http://baike.baidu.com/view/7809.htm), 可以使用相同的结构化查询语言作为数据输入与管理的接口。结构化查询语言语句可以嵌套，这使它具有极大的灵活性和强大的功能。

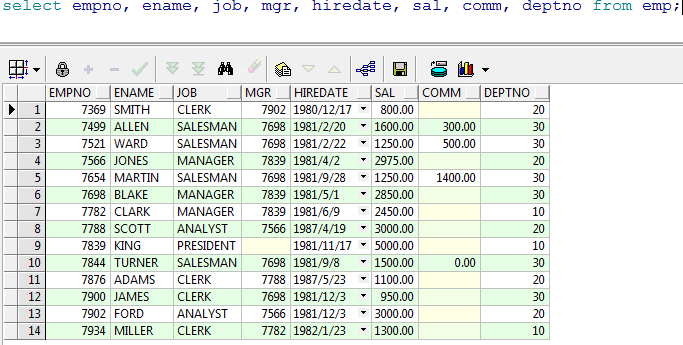
* **DML**(数据库操作语言): 其语句包括动词[INSERT](http://baike.baidu.com/view/270874.htm" \t "_blank)，[UPDATE](http://baike.baidu.com/view/1028155.htm" \t "_blank)和[DELETE](http://baike.baidu.com/view/4794.htm" \t "_blank)。它们分别用于添加，修改和删除表中的行。也称为动作查询语言。
* **DDL**(数据库定义语言): 其语句包括动词CREATE和DROP。在数据库中创建新表或删除表（CREAT TABLE 或 DROP TABLE）；为表加入索引等。DDL包括许多与人[数据库目录](http://baike.baidu.com/view/4369490.htm" \t "_blank)中获得数据有关的保留字。它也是动作查询的一部分。
* **DCL**(数据库控制语言):它的语句通过GRANT或REVOKE获得许可，确定单个用户和用户 组对[数据库对象](http://baike.baidu.com/view/50913.htm)的访问。某些RDBMS可用GRANT或REVOKE控制对[表单](http://baike.baidu.com/view/296684.htm" \t "_blank)个列的访问。

## 简单查询

### 1.查询语法

Select \* |列名 from 表名

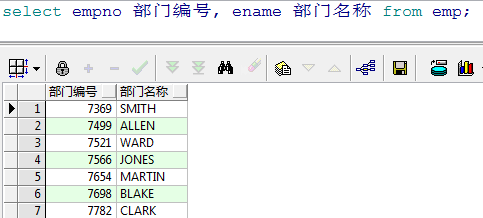




### 2.别名用法

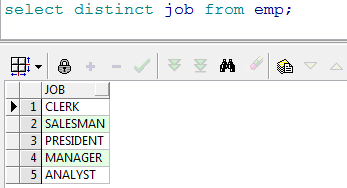
在查询的结果列中可以使用别名

Select 列名 别名，列名别名，... from emp;



### 消除重复的数据

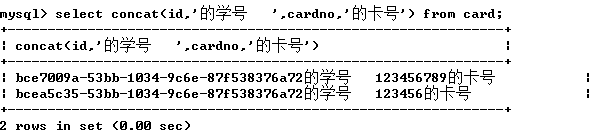
Select distinct \*|列名, ... from emp;



使用distinct可以消除重复的行，如果查询多列的必须保证多列都重复才能去掉重复

### 字符串连接查询

Mysql中实现方法：



查询雇员编号，姓名，工作

编号是：7369的雇员， 姓名是：smith，工作是：clerk



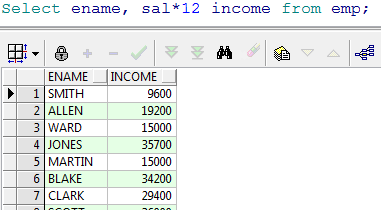
字符串的连接使用‘||’

### 查询中四则运算

查询每个雇员的年薪

select ename, sal\*12 from emp;

select ename, sal\*12 income from emp;



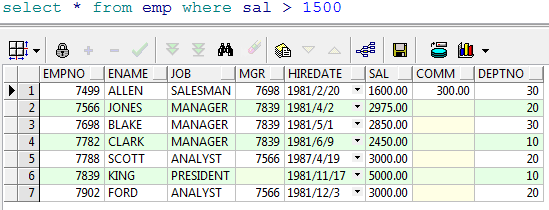
Sql中支持四则运算“+，-，\*，/”

## 3. 条件查询

在查询绝大多数都会有条件的限制

语法：select \*|列名 from 表名 where 条件

例如：查询工资大于1500的所有雇员



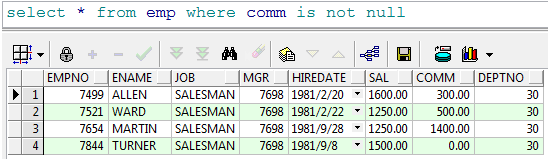
### 非空和空的限制

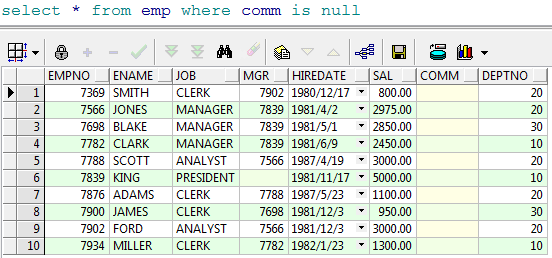
* 示例：查询每月能得到奖金的雇员

分析：只要字段中存在内容表示不为空，如果不存在内容就是null，

语法：列名 IS NOT NULL

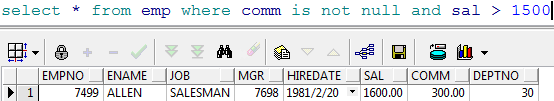
为空 列名 IS NULL





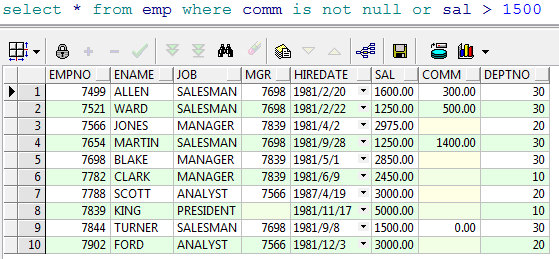
* 范例：查询工资大于1500并且有奖金领取的雇员

分析：多个查询条件同时满足之间使用‘AND’



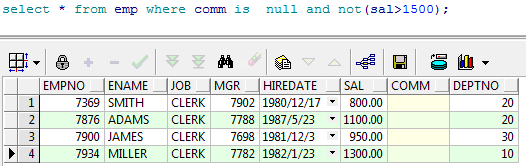
* 范例:查询工资大于1500或者有奖金的雇员

分析：多个查询条件或满足，条件之间使用“OR”



* 范例：查询工资不大于1500和没有奖金的人

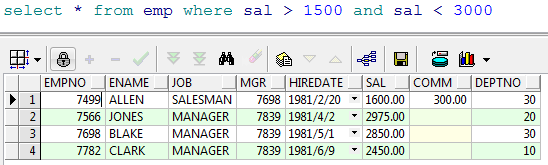
语法：NOT(查询条件)



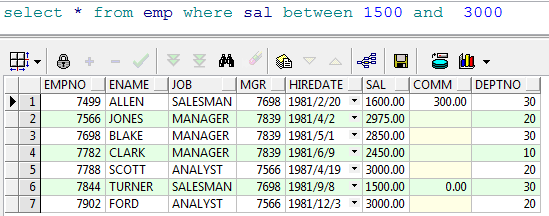
### 2.范围限制

范例：基本工资大于1500但是小于3000的全部雇员

分析：sal>1500, sal<3000

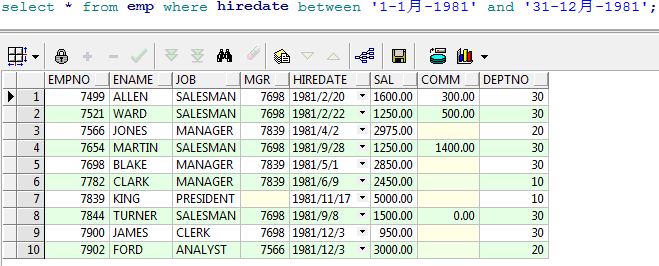


Between and等于 sal > =1500 and sal <= 3000



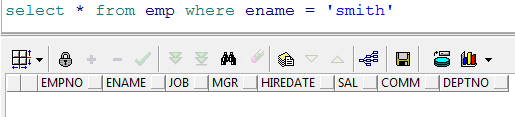
范例：查询1981-1-1到1981-12-31号入职的雇员

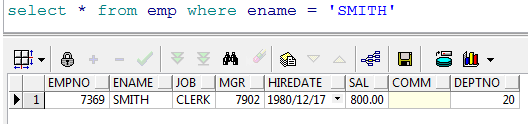
分析：between and 不仅可以使用在数值之间，也可以用在日期的区间



范例：查询雇员名字叫smith的雇员

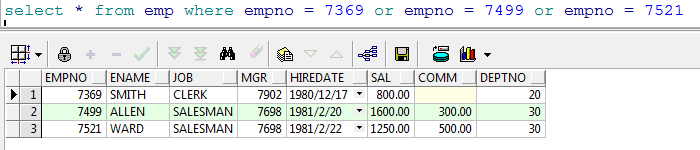
在oracle中的查询条件中查询条件的值是区分大小写的





范例:查询雇员编号是7369,7499,7521的雇员编号的具体信息

如果使用之前的做法可以使用OR关键字

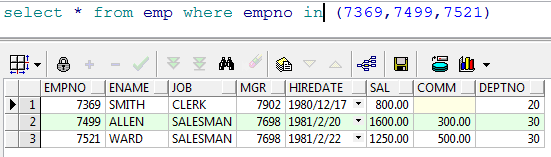


实际上，此时指定了查询范围，那么sql可以使用IN关键字

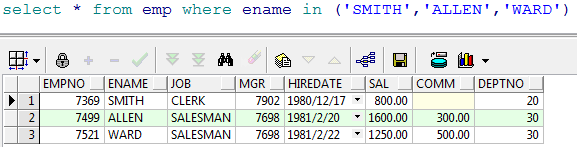
语法: 列名 IN (值1，值2，....)

列名 NOT IN (值1， 值2，...)

其中的值不仅可以是数值类型也可以是字符串



范例:查询雇员姓名是’SMITH’,’ALLEN’,’WARD’的雇员具体信息



### 3．模糊查询

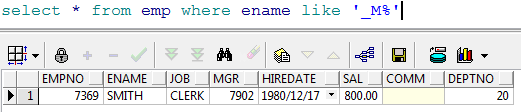
在常用的站点中经常会有模糊查询，即：输入一个关键字，把符合的内容全部的查询出来，在sql中使用LIKE语句完成。

在LIKE中主要使用以下两种通配符

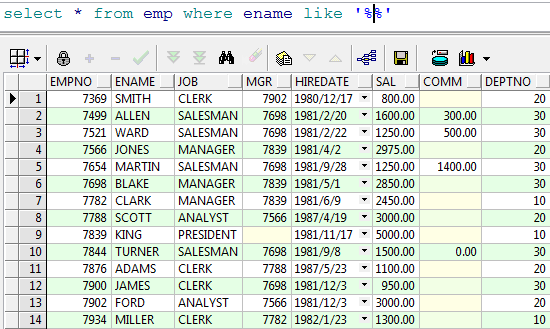
“%”：可以匹配任意长度的内容

“\_”:可以匹配一个长度的内容

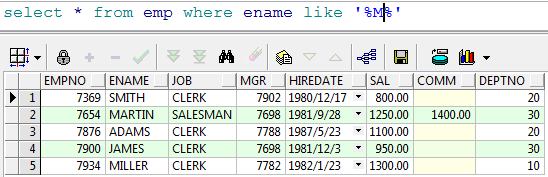
范例：查询出所有雇员姓名中第二个字符包含“M”的雇员



在LIKE中如果没有关键字表示查询全部

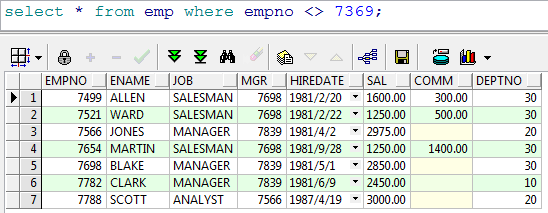


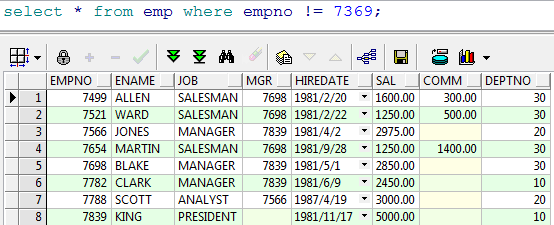
查询名字中带有“M”的雇员



在oracle中不等号的用法可以有两种形式“<>”和“!=”

范例：查询雇员编号不是7369的雇员信息





## 对结果集排序

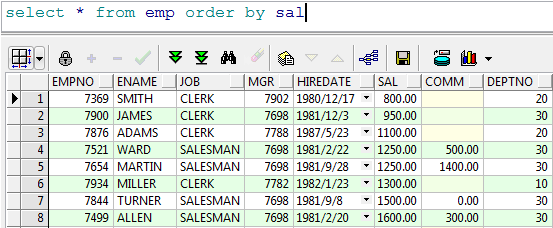
### 1.排序的语法

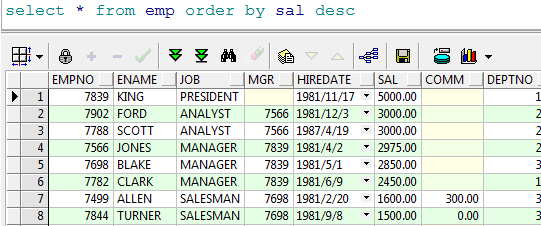
在sql中可以使用ORDER BY对查询结果进行排序

语法：SELECT \* |列名 FROM 表名 {WEHRE 查询条件} ORDER BY 列名1 ASC|DESC，列名2...ASC|DESC

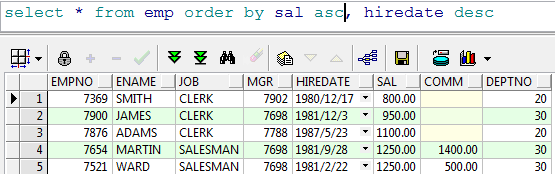
范例：查询雇员的工资从低到高

分析：ORDER BY 列名 默认的排序规则是升序排列，可以不指定ASC，如果按着降序排列必须指定DESC





如果存在多个排序字段可以用逗号分隔



注意ORDER BY语句要放在sql的最后执行。

### 2.排序中的空值问题

排序 order by 经验：当排序时存在null时就会产生问题 nulls first , nulls last

--查询雇员的工资从低到高

select \* from emp order by sal nulls first;

select \* from emp order by sal desc nulls last ;

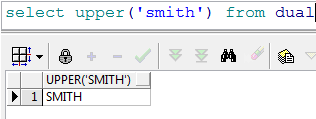
# 13.函数

### 1.字符函数

**接收字符输入返回字符或者数值，dual是伪表**

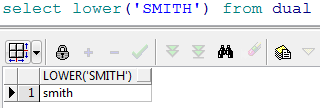
1. 把小写的字符转换成大小的字符

upper('smith')



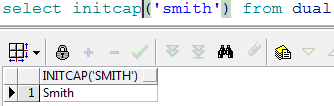
1. 把大写字符变成小写字符

lower('SMITH')



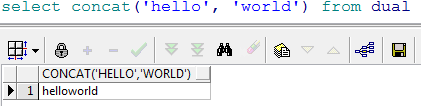
1. 把首字符大写

initcap('smith')



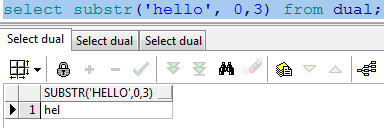
1. 字符串的连接可以使用concat可以使用“||”建议使用“||”

concat('hello', 'world')



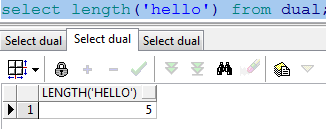
1. 字符串的截取，使用substr，第一个参数是源字符串，第二个参数是开始索引，第三个参数长度，开始的索引使用1和0效果相同

substr('hello', 1,3)



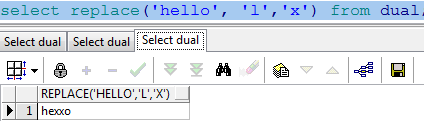
1. 获取字符串的长度

length('hello')



1. 字符串替换，第一个参数是源字符串，第二个参数被替换的字符串，第三个是替换字符串

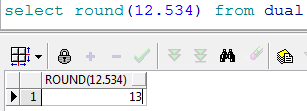
replace('hello', 'l','x')

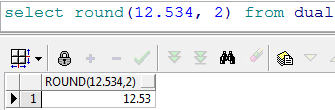


### 数值函数

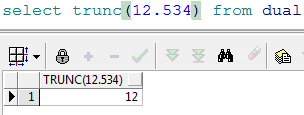
1. 四舍五入函数：ROUND()

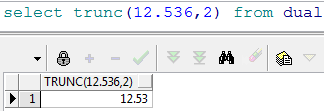
默认情况下ROUND四舍五入取整，可以自己指定保留的位数。



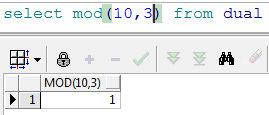


1. 取整：TRUNC()，默认全部去掉小数，也可以指定保留的位数





1. 取余数MOD()



### 3.日期函数

Oracle中提供了很多和日期相关的函数，包括日期的加减，在日期加减时有一些规律

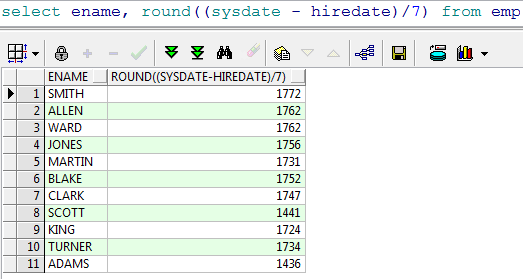
日期 – 数字 = 日期

日期 + 数字 = 日期

日期 – 日期 = 数字

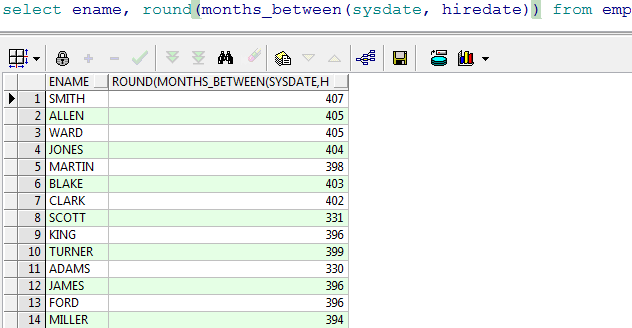
1. 范例：查询雇员的进入公司的周数。

分析：查询雇员进入公司的天数(sysdate – 入职日期)/7就是周数



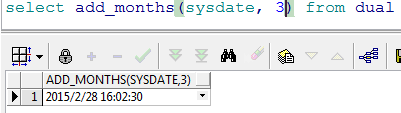
1. 获得两个时间段中的月数：MONTHS\_BETWEEN()

范例：查询所有雇员进入公司的月数



1. 获得几个月后的日期：ADD\_MONTHS()

范例：求出三个月后的日期



### 4.转换函数

1. TO\_CHAR:字符串转换函数

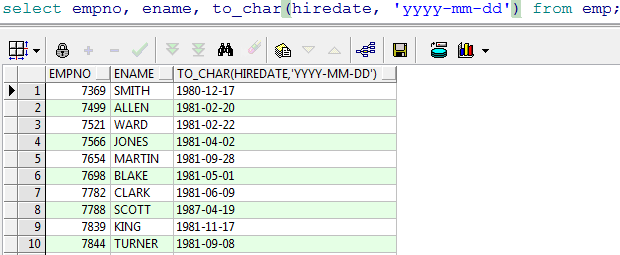
范例：查询所有的雇员将将年月日分开，此时可以使用TO\_CHAR函数来拆分

拆分时需要使用通配符

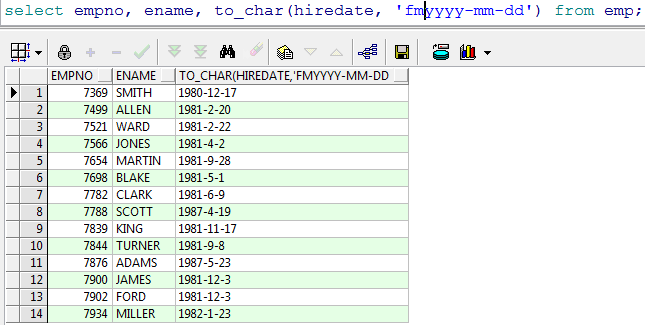
年：y, 年是四位使用yyyy

月：m, 月是两位使用mm

日：d, 日是两位使用dd

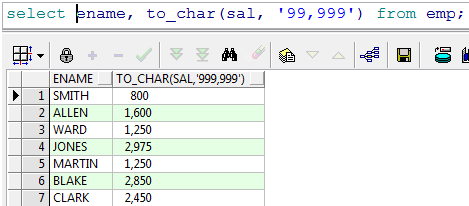


在结果中10以下的月前面被被补了前导零，可以使用fm去掉前导零

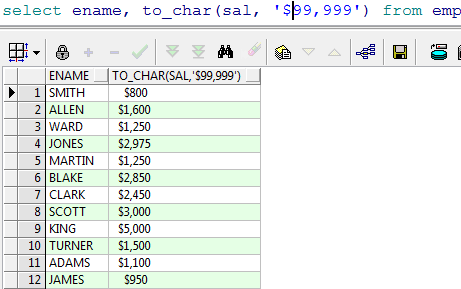


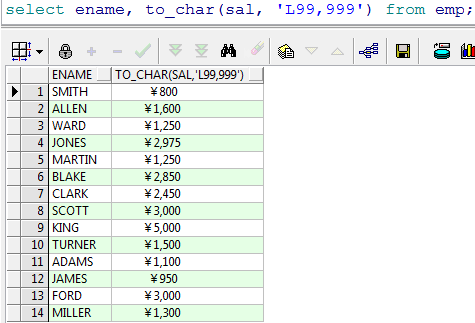
TO\_CHAR还可以给数字做格式化

范例：把雇员的工资按三位用“,”分隔，在oracle中“9”代表一位数字



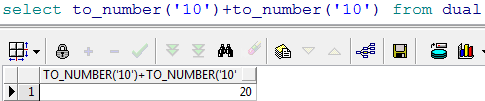
如果在钱的前面加上国家的符号可以使用“$”代表是美元，如果要使用本地的钱的单位使用“L”





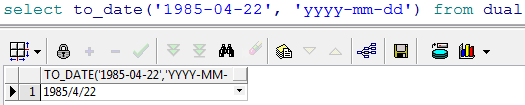
2. TO\_NUMBER:数值转换函数

TO\_NUMBER可以把字符串转换成数值



3. TO\_DATE:日期转换函数

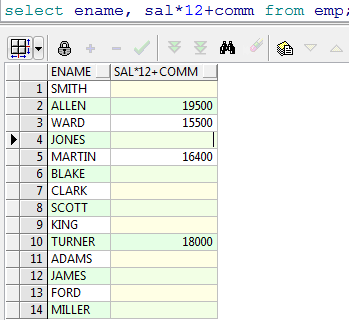
TO\_DATE可以把字符串的数据转换成日期类型



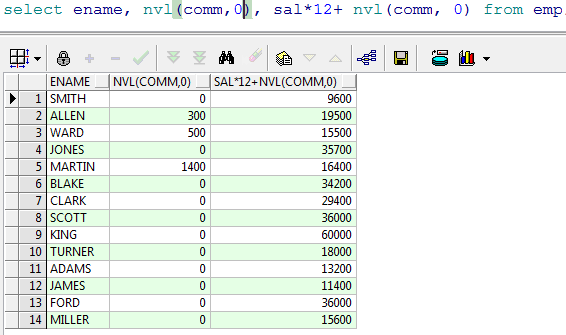
### 5.通用函数

#### 1．空值处理nvl

范例：查询所有的雇员的年薪



我们发现很多员工的年薪是空的，原因是很多员工的奖金是null，null和任何数值计算都是null，这时我们可以使用nvl来处理。



#### 2.Decode函数

该函数类似if....else if...esle

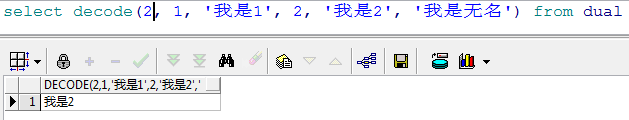
语法：DECODE(col/expression, [search1,result1],[search2, result2]....[default])

Col/expression:列名或表达式

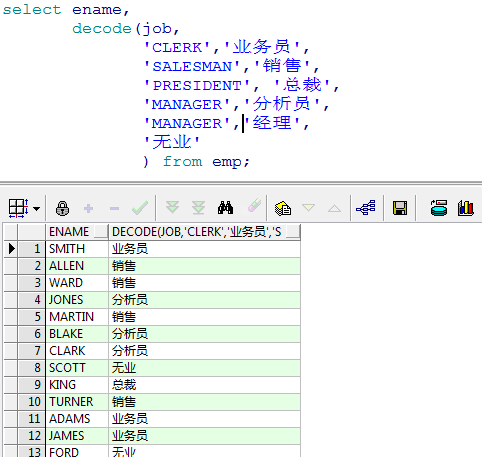
Search1，search2...:用于比较的条件

Result1, result2...:返回值

如果col/expression和Searchi匹配就返回resulti,否则返回default的默认值



范例：查询出所有雇员的职位的中文名



#### 3.case when

CASE expr WHEN comparison\_expr1 THEN return\_expr1

[WHEN comparison\_expr2 THEN return\_expr2

WHEN comparison\_exprn THEN return\_exprn

ELSE else\_expr]

END

select t.empno,

t.ename,

case

when t.job = 'CLERK' then

'业务员'

when t.job = 'MANAGER' then

'经理'

when t.job = 'ANALYST' then

'分析员'

when t.job = 'PRESIDENT' then

'总裁'

when t.job = 'SALESMAN' then

'销售'

else

'无业'

end

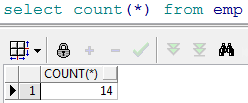
from emp t



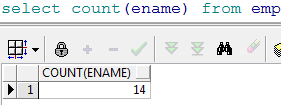
### 6.聚合函数

#### 1.统计记录数count()

范例：查询出所有员工的记录数

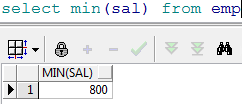


不建议使用count(\*)，可以使用一个具体的列以免影响性能。



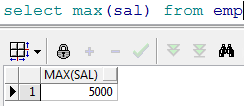
#### 2.最小值查询min()

范例：查询出来员工最低工资



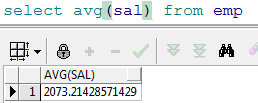
#### 3.最大值查询max()

范例：查询出员工的最高工资



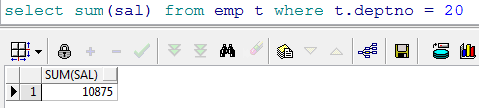
#### 4.查询平均值avg()

范例：查询出员工的平均工资



#### 5.求和函数sum()

范例：查询出20号部门的员工的工资总和



## 14.集合操作

### 1.并集

UNION

UNION ALL

### 2.交集

INTERSECT

### 3.差集

MINUS

### 4.集合操作的注意事项总结