Oracle第一天

v3.1

# Oracle概述

## 什么是数据库

数据库的英文是DataBase，简称DB，顾名思义，就是数据（包括数字、文字、图像、声音、视频等）存放的地方。因此，数据库的作用只要就是用来存储数据的。

## 关系型数据库管理系统（RDBMS）

RDBMS即关系数据库管理系统(Relational Database Management System)，是将数据组织为相关的行和列的系统，而管理关系数据库的计算机软件就是关系数据库管理系统，常用的数据库软件有Oracle，Microsoft SQL Server，DB2，Sybase，Informix，MySQL，PostgreSQL……等。

关系型数据库是建立在关系模型基础上的数据库，借助于集合代数等数学概念和方法来处理数据库中的数据。

## 什么是Oracle



Oracle公司-甲骨文公司，全称甲骨文股份有限公司(甲骨文软件系统有限公司)，是全球最大的企业级软件公司，总部位于美国加利福尼亚州的红木滩。1989年正式进入中国市场。2013年，甲骨文已超越 IBM ，成为继 Microsoft 后全球第二大软件公司。

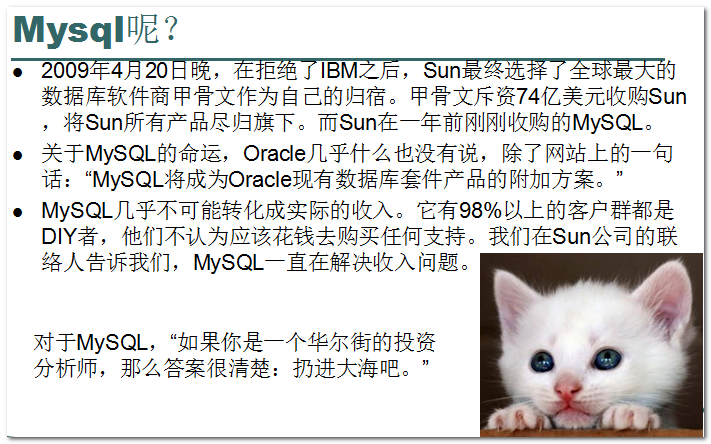
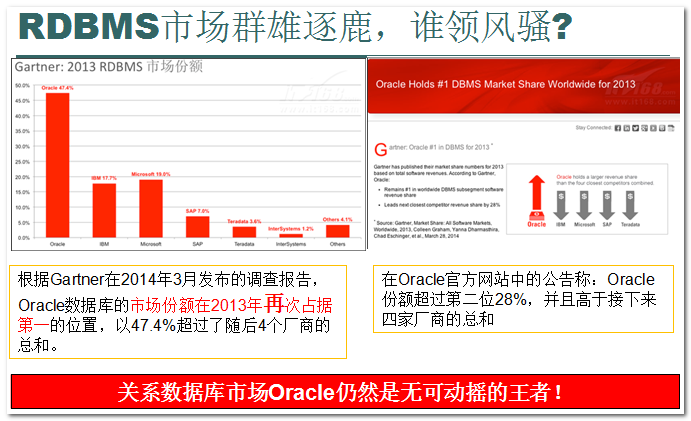
Oracle公司的网址为http://www.oracle.com。

Oracle在企业的大型应用、海量存储、高性能、高并发、安全性等方面都相当不错，被广泛应用于各个领域。Oracle不仅有数据库产品，而且也是ERP软件的供应商。

我们本次课程所说的Oracle都是指其数据库产品。

p．s：想了解更多，可以参考课前资料。

## Oracle的地位



## Oracle的应用场景

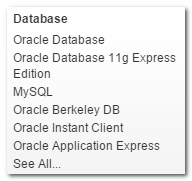
Oracle数据库方案在大量的企业，涉及航空与国防、汽车、化学、消费品、高科技、工业制造、生命科学、自然资源、石油和天然气、公用事业、通信（移动、联通、电信）、媒体和娱乐、教育和人力资源、工程建筑、金融服务（银行、证券）、卫生医疗、公共部门、零售、交通运输和专业服务等诸多行业。

# Oracle服务端的安装和配置

## 版本和下载

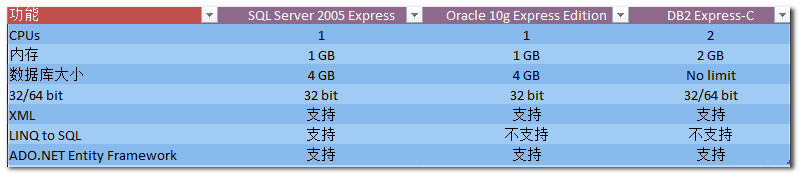
Oracle的版本发展8、9i，10g，11g，12c。

Oracle分免费简化版本（Oracle Database Express）和完整版本（Oracle Database），都可以自由下载。



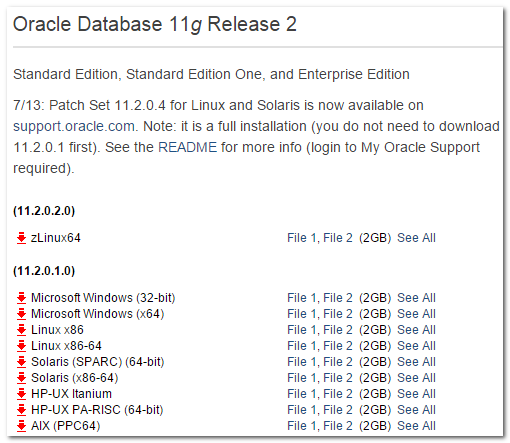
* Express版本你可以免费用于商业用途，但这个版本对cpu/内存/数据量的有限制。

常见的关系型数据库的Express版本对比：



* 完整版本可以免费用于学习、教学等，如果商用则需要收费，那么就可以享受Oracle公司提供的服务，Licence按照CPU数量报价(2009年)，一个CPU大约50w左右，打完折，25w以上。

Oracle可以在大部分主流的操作系统上安装，比如Linux和Windows。官方的网站是：[www.oracle.com。](http://www.oracle.com。/)



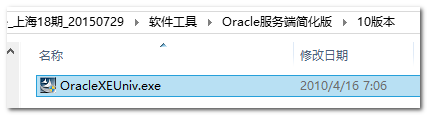
完整版本Oracle的windows版本（32位）是从官方直接下载的参考网址：

http://download.oracle.com/otn/nt/oracle11g/112010/win32\_11gR2\_database\_1of2.zip

http://download.oracle.com/otn/nt/oracle11g/112010/win32\_11gR2\_database\_2of2.zip

## 软件安装

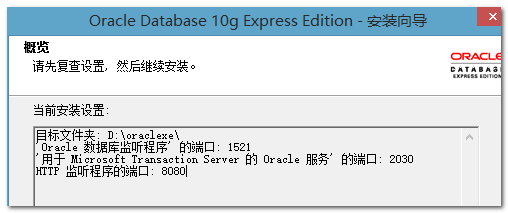
我们安装简化版本：



安装过程中：



这个口令是sys和system这两个超级用户的密码。sys相当于mysql的root。



Oracle的服务端口，默认是1521

提醒：简化版安装完成之后，你的8080端口可能会被占用。你可以将这个服务禁用，也可以将端口改掉。可以参考：



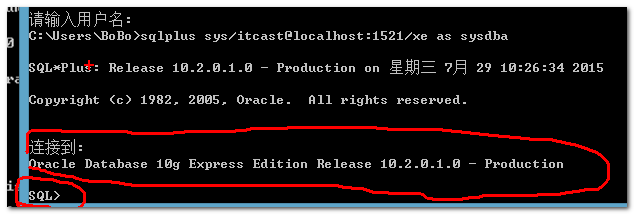
sys和system是系统的管理员帐号，默认密码是安装Oracle时指定的密码。

测试是否安装成功：

两种方式：一个是看服务，有没有oracle服务。一个是客户端连接测试。方法：

在安装有oracle的机器上，运行cmd，输入：

|  |
| --- |
| sqlplus sys/baby@localhost:1521/xe as sysdba |



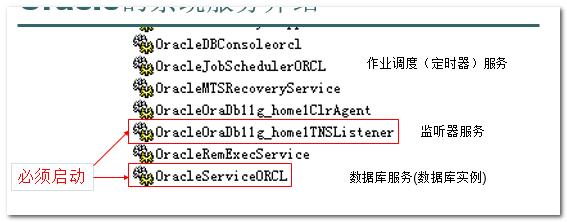
软件的卸载（再运行一遍简化版的安装程序，直到出现下面的界面）：



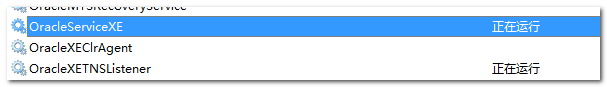
## Oracle的服务

在windows下打开“运行”，输入“Services.msc”，找到Oracle相关的服务。

完整版本：



简化版本：



关于这些服务：数据库实例服务和监听服务必须启动，其他都可以禁用。(如果安装到本机，那么建议全手动。)

新手如何记忆这两个服务的名字呢？

数据库实例服务：Oracle+Service+SID，简化版的Oracle默认的sid是xe，完整版的Oracle默认的SID是orcl(可以更改)。

监听服务：Oracle+SID+TNSListener。

手动启动或重启服务的顺序问题：

建议先启动实例服务：OracleServiceORCL，再启动监听服务



提醒：每次重启或开启服务的时候，需要等一会再连接。

## 服务配置-了解

Windows下的服务端安装完成后，大部分默认情况下是只允许本机访问，如果要让别的机器访问本机的Oracle服务，最简单的方法就是：

**先停掉Oracle两个服务，再修改两个配置文件，最后重启服务即可。**

具体操作如下：

停止oracle服务后，在服务端的如下路径中找到两个文件：

* listener.ora
* tnsnames.ora

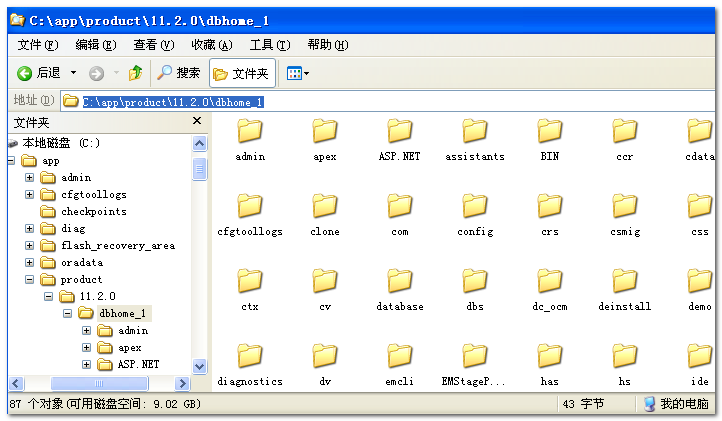
这两个文件在oracle\_home\NETWORK\ADMIN（C:\app\product\11.2.0\dbhome\_1）下。

（D:\oraclexe\app\oracle\product\10.2.0\server\）

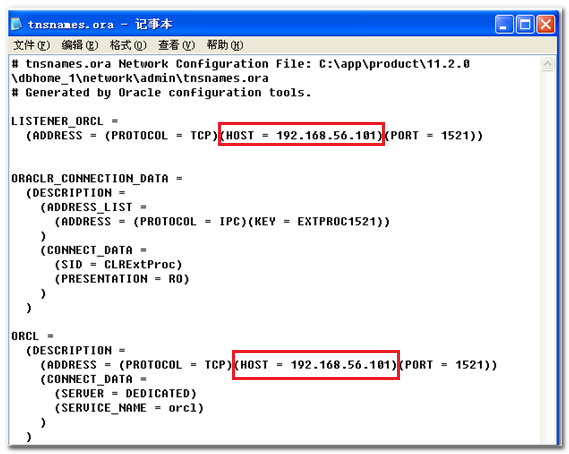
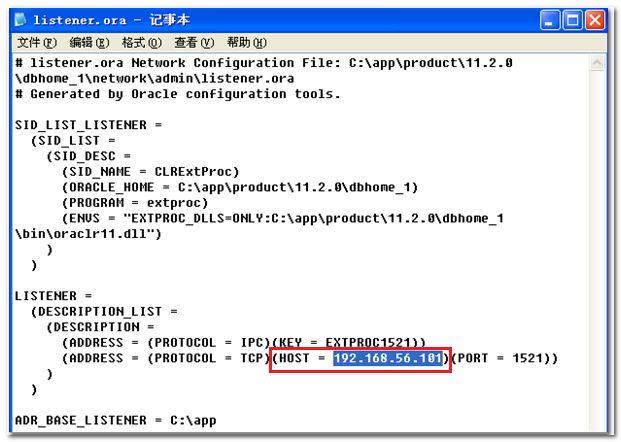
【解释】

什么是oracle\_home？

它是Oracle程序安装的目录，拥有Oracle程序运行的相关环境，所有的Oracle软件相关库文件、配置文件都在这里。我这里的位置如下：



更改这两个文件内容中的HOST的部分，如果默认是机器名，则不需要更改，如果是localhost，则需要更改为本机IP。修改的图解如下：



注意：

* 为避免修改错误或者修改后不生效，建议将先要将这两个文件备份一份。
* 默认一般共需要修改两处或三处地方，请仔细检查。
* 尽量先停止服务后再更改，更改完之后，要生效需要重启服务（两个服务都要重启）。

测试：



提示：

如果还是访问不了，请检查windows的防火墙是否关闭。

补充提示：

除了直接修改配置文件的方式外，Oracle也提供了图形配置的方式。具体可参考其他文档，如《Oracle的网路配置》。

# Oracle客户端的安装和配置

## 常见的客户端工具

常见的客户端工具为Oracle自带的命令行工具sqlplus、jdbc、第三方的图形化工具plsql developer，下面依次讲解一下安装配置和基本使用。

## Sqlplus

### 工具介绍

它是Oracle自带的命令行工具，兼容性和连接速度都是很好的。

如果你在当前操作的机器上安装有Oracle，则默认情况下就拥有这个命令行工具，不需要额外安装。否则，你需要安装Oracle独立客户端（Oracle Instant Client）。

### 使用sqlplus

基本语法：sqlplus 用户名/密码@主机地址:端口/sid（oracle的服务标识） 登录身份

提示：

端口默认是1521，可省略！

sid：是安装的时候提供的一个字符串，简化版默认是xe，无法更改；完整版默认orcl，可以更改

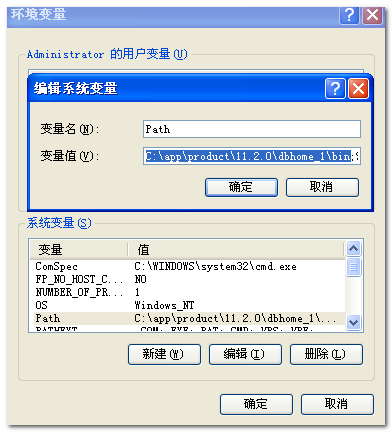
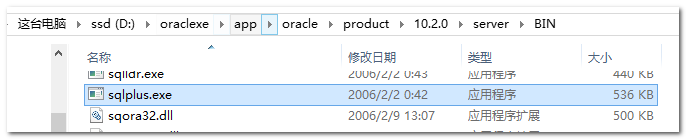
登录身份：普通用户不需要写，超管用户需要使用dba的身份，写法：as sysdba

在服务机上的cmd窗口中可以直接输入sqlplus命令：

|  |
| --- |
| c:\> --查看当前登录的用户：  SQL> show user;  连接的时候可以省略端口号（默认是1521）： |

【了解】

能直接执行这个命令的原因，是因为命令所在目录在安装后被自动配置到了环境变量中了：



### 乱码问题

在中文Windows环境下，完整版的oracle（服务端编码都是GB。。16。。,）,客户端会出现乱码问题。（ps：简化版本：服务端和客户端默认的编码utf-8，没有乱码问题）

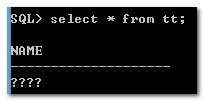
但简化版本的Oracle不会乱码。

原因：

完整版的Oracle的服务端我们一般都会指定编码为ZHS16GBK，但oraclce客户端，默认是utf-8的编码。编码不一致，导致乱码。

简化版，安装过程中不能更改编码，默认就是utf-8整合和客户端一致。无需更改编码。

新建一张表，插入中文测试：



解决方案：设置本地客户端编码。

添加环境变量：

|  |
| --- |
| NLS\_LANG="SIMPLIFIED CHINESE\_CHINA.ZHS16GBK" |



如果你的系统语言不是中文的，或者cmd窗口的编码不是GBK的，那么还需要设置一个环境变量来解决：（绝大多数不需要设置这个）

|  |
| --- |
| LANG=zh\_CN.GBK |

## JDBC连接

学习目标：

* 认识驱动Jar包（驱动）
* 记住JDBC的连接字符串

通过JDBC来连接Oracle有两种方式：

* Thin（瘦客户端）方式。纯java直连方式（直接的TCP/IP通讯），推荐。优点是只需要Oracle的驱动包（jar）就行

连接字符串：jdbc:oracle:thin:@localhost:1521:xe

* Oci（胖客户端，Oracle Call Interface）方式。Java来调用本机的Oracle客户端，然后再访问数据库，优点是速度快，但是需要安装和配置Oracle数据库的环境。不推荐。PLSQL Developer就是用的这种方式，它需要依赖于客户端OCI程序（前提必须安装）。

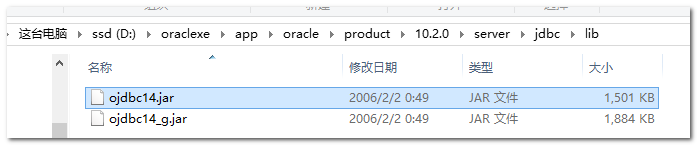
连接字符串：jdbc:oracle:oci:@localhost:1521:orcl, jdbc:oracle:oci8:@orcl

扩展阅读：ORACLE调用接口(Oracle Call Interface简称OCI)提供了一组可对ORACLE数据库进行存取的接口子例程（函数），通过在第三代程序设计语言（如C语言）中进行调用可达到存取ORACLE数据库的目的。

本例使用thin的方式连接数据库。

Oracle数据库提供的jar的包的位置：

简版的10G：



完整版的11g：



版本问题：ojdbc6（11）的版本高于ojdbc14（10）

Java程序采用jdbc方式连接ORACLE步骤：

1， 导入驱动包

2， 建立和ORACLE连接的封装类

Class.forName();//注册驱动

DriverManger.getConnection();//获取数据库的连接对象

3， 使用连接对象对数据进行操作

【示例】

需求：使用java连接Oracle，查询并打印sys用户下的aa表（自己建立的表）的数据。

【java代码】

|  |
| --- |
| //jdbc测试：  //注意：连接字符串和驱动的字符串  //查询刚才我们建立的表的tt的数据，打印出来  **public** **class** OracleJdbcTest {  **public** **static** **void** main(String[] args) **throws** Exception {  //驱动  //mysql:com.mysql.jdbc.Driver  //Oracle：  // oracle.jdbc.driver.OracleDriver（官方推荐）  // oracle.jdbc.OracleDriver  Class.*forName*("oracle.jdbc.driver.OracleDriver");    //连接conn  //连接字符串  //mysql:jdbc:mysql://localhost:3306/数据库名  //Oracle:注意：和sqlplus不太一样，sid前面符号：sqlplus:“/”，连接串：“：”  String url="jdbc:oracle:thin:@127.0.0.1:1521:xe";  //用户名:  //普通用户：自需要写用户名  //超管用户：用户名 + as sysdba  String user="sys as sysdba";  //密码  String password="baby";  Connection conn = DriverManager.*getConnection*(url, user, password);    //stmt  Statement stmt = conn.createStatement();  //查询出结果集  ResultSet rs = stmt.executeQuery("select \* from tt");    //提示：oracle必须提交事务才有数据  **while**(rs.next()){  System.*out*.println("用户的姓名："+rs.getString(1));  }    //释放资源  rs.close();  stmt.close();  conn.close();  }  } |

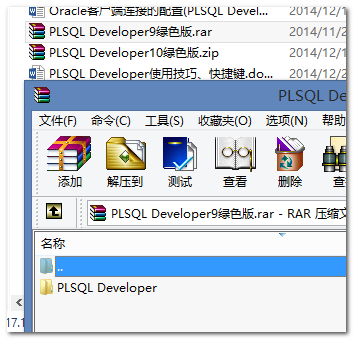


## PL/SQL Developer

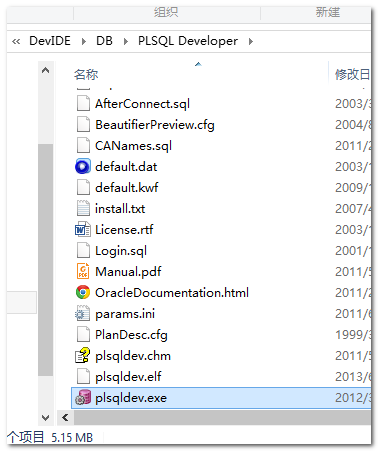
### 工具的安装

该工具是第三方的工具，非官方的。后面都简称“工具”；

本次教学提供安装版本和绿色版本，推荐使用绿色版本，直接解压即可使用。



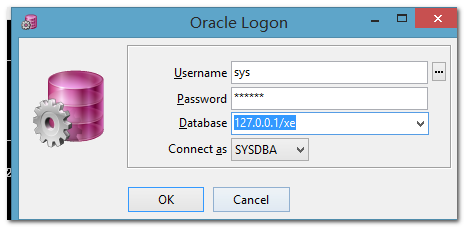
直接解压到一个目录即可，主执行文件为plsqldev.exe，可以自行创建快捷方式。



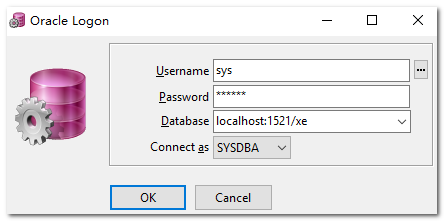
### 使用工具连接Oracle数据库

方式一：直接输入地址和sid

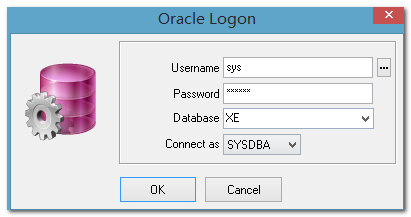
打开该工具，输入用户名，密码，数据库连接字符串（这里可以省略端口号1521）：



或



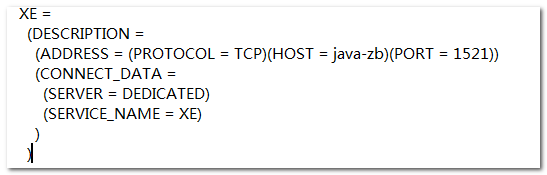
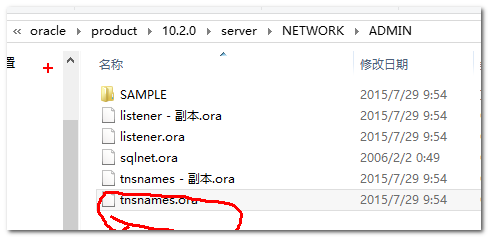
方式二：使用配置文件的方式。



提示：

按钮的作用：ok是开始登录；cancel是不登录直接进入软件界面，无法操作数据库。

下拉服务列表怎么产生的，怎么配置的：



要连接的服务器的地址，如果你要连接其他的机器，则，将这里直接修改为其他机器的ip



一般不需要修改，如果服务器的端口号不是1521，则这里需要修改。



：设置osid（oracle服务的唯一标识）



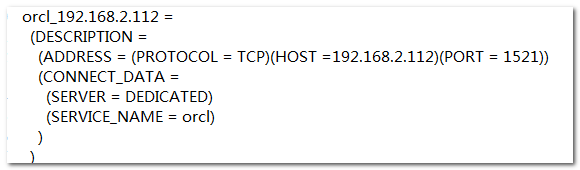
名字，任意起名，主要用来显示用的。前面一定不要带空格。



配置多个服务器连接：

场景：如果你在企业中，可能不止连接一台Oracle服务器，那么你就需要配置多个oracle的连接，如何配置呢？

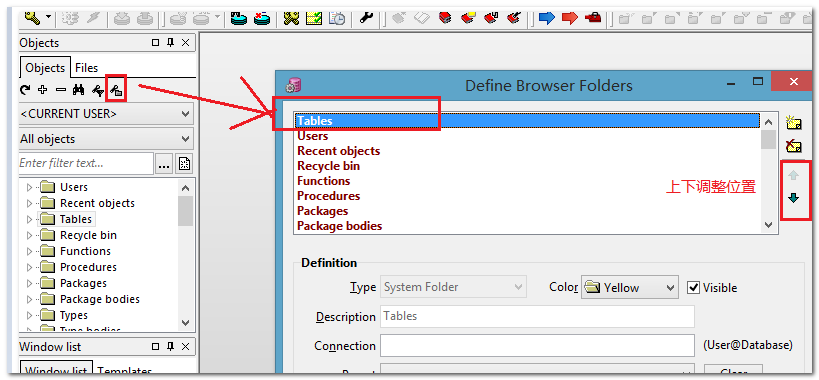
将xe的原先配置复制一份，改改就行了：



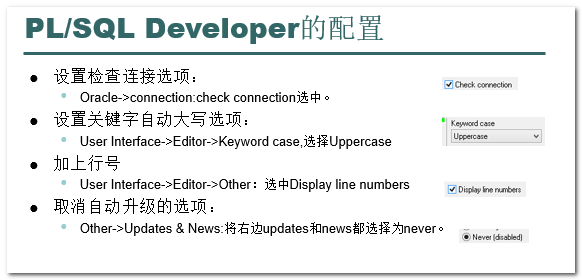
提醒：注意，虽然名字可以随意起，但前面一定不要加空格

### 工具的使用优化设置

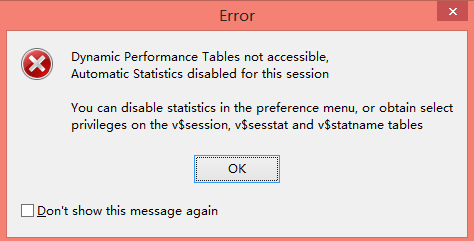
对象的显示顺序：



Tools->Preferences->

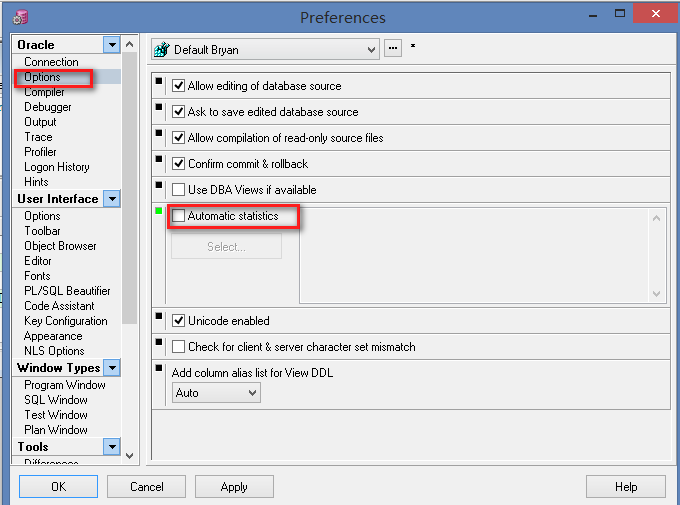


登录普通用户（如scott）后第一次查询表时，会弹出：



修改PLSQL Developer的设置--关闭PLSQL Developer 的统计功能：

在 Tools->Preferences->Options里 把Automatic Statistics前的那个勾子去掉，保存



原因是：plsql的自动统计功能需要读取SYS用户下的数据字典表，但普通用户没有权限读取，导致工具出现错误提示。

## Oracle客户端连接知识（客户端连接验证机制、客户端登录身份）。

### 客户端连接验证机制

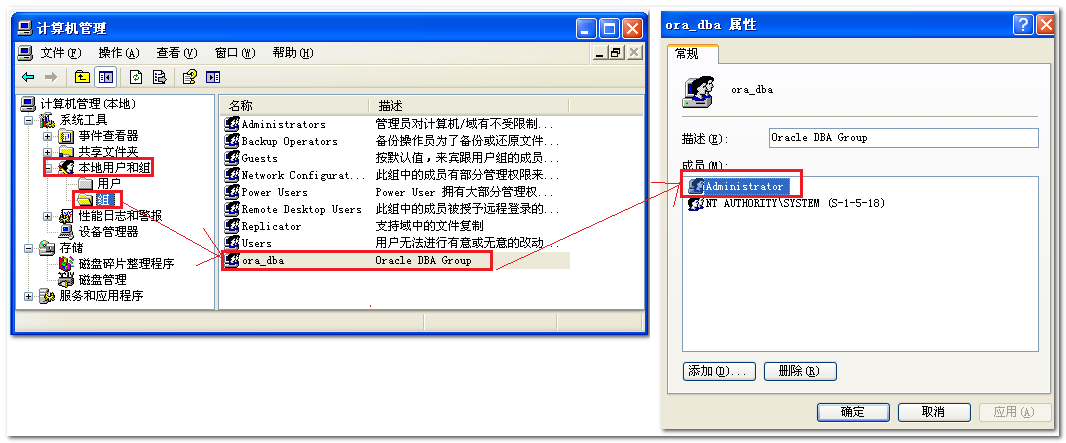
Oracle有三种连接验证机制：

* 操作系统验证（具有sysdba和sysopera的用户）
* ~~密码文件验证（具有sysdba和sysopera的用户）~~
* 数据库密码验证（普通用户和超级用户）

超级管理员连接数据库的方式：



也成功登录。（免登录必须在安装有Oracle服务的服务器上才有效）



免密码登录的原因是：oradcle安装时自动创建了一个ora\_dba的组，并将当前用户放入改组。如果你连接oracle的时候，会优先使用来查找当前用户是否在组中。如果在组中，则不需要密码就可以登录。

免登录要求：必须在安装有Oracle的本机上才能使用。

注意：

我们平时用的连接机制是数据库密码验证的机制。

关于验证的优先级顺序：

对于 SYSDBA身份 和 SYSOPER来说，OS验证优先于密码文件认证。因此，在服务器的机器上，你可以无需指定登录用户名或者任意指定登录用户名，均可以登录。

该特性的一个应用场景就是免登录解锁用户：sqlplus / as sysdba

【案例】

【示例】通过超级管理员对Oracle用户解锁和找回密码

比如sys用户密码忘记了

|  |
| --- |
| Sqlplus / as sysdba  --免登陆超级管理员的密码  语法：  alter user 用户名 identified by 密码  示例：  --更改用户密码：  Alter user sys identified by baby;  加锁和解锁用户：  语法：  alter user 用户名 account lock;--加锁：该用户不能登录。  alter user 用户名 account unlock;--解锁：该用户可以登录了。  示例：  Alter user scott account lock;  Alter user scott account unlock; |

提示：

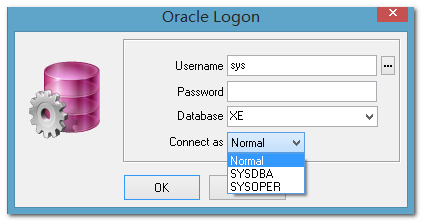
因为不需要密码是不安全的，所以一般用作服务器的话，要在计算机管理中的用户组ora\_dba把Administrator删除，删除之后登录就需要输入密码了。

提示：

conn / as sysdba连接时，使用的是当前登陆操作系统的用户名和密码认证。如果oracle是由当前用户安装的，那它可以成功登陆，通俗点说，当你用“as sysdba”登录时是按照操作系统用户验证的，也就是说oracle认为你都已经是这台电脑的老大了，我oracle没法限制你。

### 客户端登录的身份

Oracle有三种身份登录方式：Normal、sysdba、sysoper。



normal身份：普通用户身份，默认选项(默认可以不写)，用于普通用户登录使用。---记录日志（你的任何操作oracle都会记录一份日志）

sqlplus scott/tiger@localhost:1521/xe

sysdba身份：数据库管理员身份，用于给拥有DBA权限的用户登录使用。(拥有数据库)

该身份可以进行的操作： 打开、关闭数据库服务器， 备份、恢复数据库， 日志归档，会话限制，管理功能，创建数据库等。----操作是不计日志的。



sysoper身份：数据库操作员身份，用于给拥有DBA权限的用户登录使用。

该身份可以进行的操作： 打开、关闭数据库服务器， 备份、恢复数据库， 日志归档，会话限制等。---记录日志

作为开发人员，如何选择Oracle的登录身份？

* 普通用户进行数据操作，就用normal。（默认值）
* Sys用户就用sysdba。--(sys相当于mysql：root)

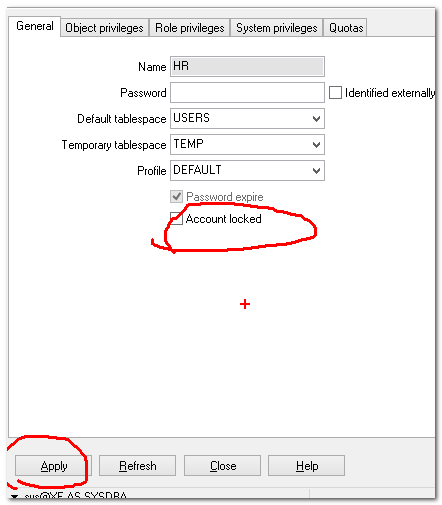
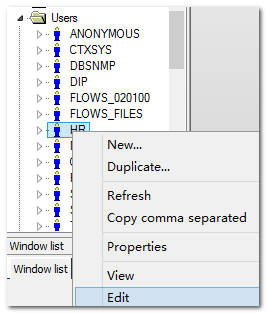
# 贯穿学习过程中的用户和表。

本次课程中主要涉及到两个用户下的表：scott和hr

【了解】：默认情况下，完整版的Oracle安装完成后，只解锁有效两个用户：sys,system，hr和scott用户需要解锁。而简化版的Oracle安装完成之后，无scott用户，而hr用户仍然需要手动解锁。

【解锁hr用户】

使用sys用户登录上之后，

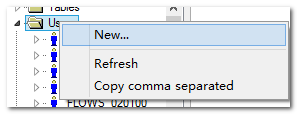


这里也可以更改密码，一般我们习惯为了操作方便，将用户名和密码设置一样。

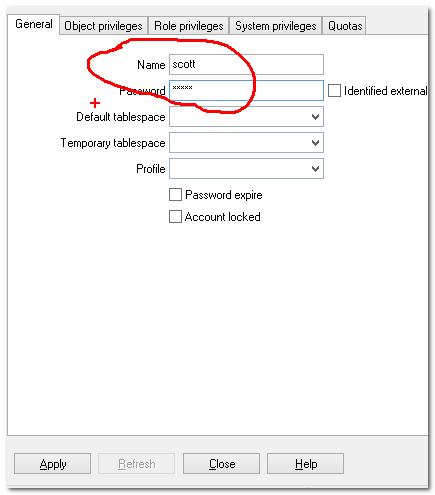
【新建Scott用户】

scott用户：需要我们手动建立和导入：

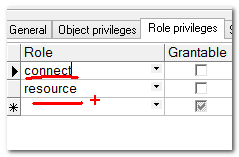
先建立scott用户（必须是sys用户登录进的）：

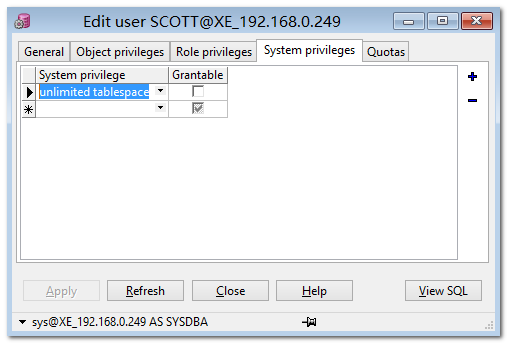


设置用户名和密码：用户名为scott密码为tiger



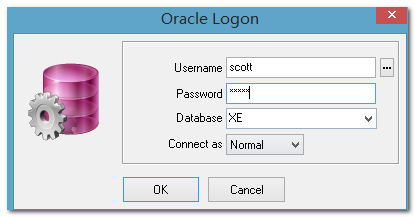
添加两个角色：



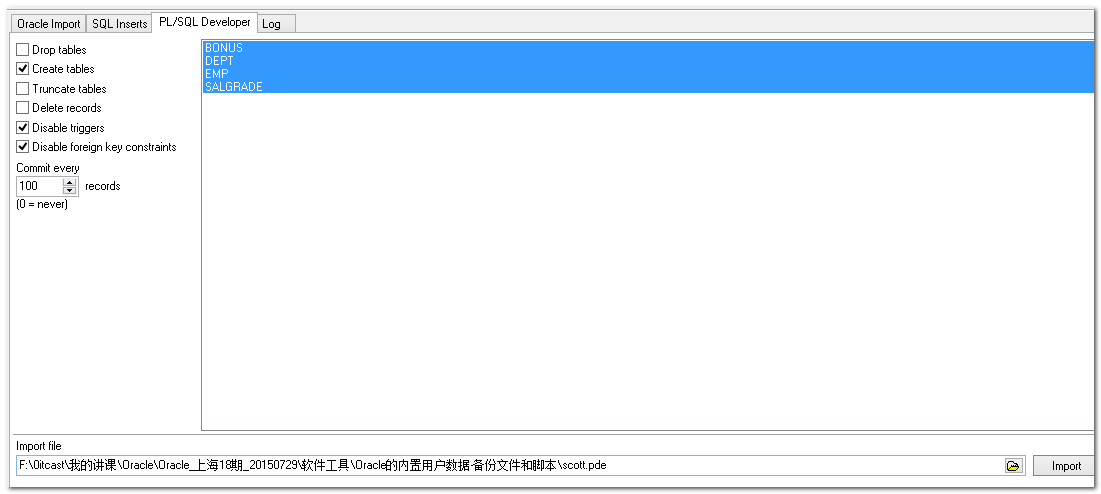
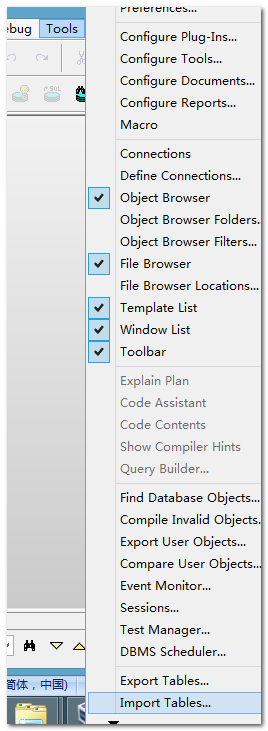


完成后，点击apply：

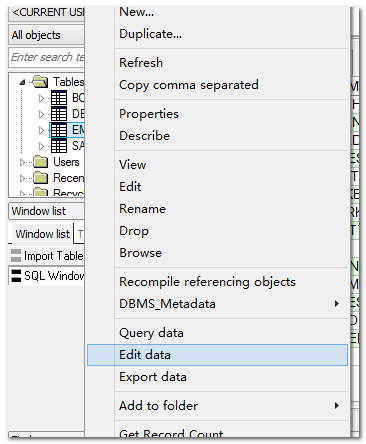
使用该用户登录（普通用户登录）：



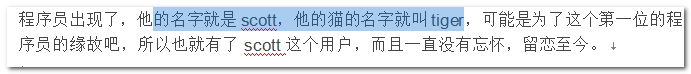
登录进去之后，到导入样例数据：



查询数据：



scott是上课时用的，默认密码是tiger。



hr是作业中涉及的，默认密码是hr。

## scott用户下的表结构

DEPT 部门表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| DEPTNO | NUMBER(2) | 部门编号，长度为2 |
| DNAME | Varchar2(14) | 部门名称，长度为14 |
| LOC | Varchar2(13) | 位置 |

EMP 雇员表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| EMPNO | NUMBER(4) | 雇员编号 长度为4 |  |
| Ename | Varchar2(10) | 雇员姓名 长度为10 |
| Job | Varchar2(9) | 职位 长度为9 |
| mgr | Number(4) | 领导编号 领导也是雇员 |
| hiredate | date | 入职日期 |
| sal | Number(7,2) | 雇员基本工资 |
| comm | Number(7,2) | 奖金 |
| deptno | Number(4) | 部门编号  和部门表中的编号关联 |

Salgrade 工资等级表

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| grade | number | 等级 |
| losal | number | 最低工资 |
| hisal | number | 最高工次 |

~~bonus工资表~~

|  |  |  |
| --- | --- | --- |
| 字段 | 类型 | 描述 |
| ENAME | VARCHAR2(10) | 雇员姓名 |
| JOB | Varchar2(9) | 职位 |
| sal | number | 基本工资 |
| comm | number | 奖金 |

## ORACLE常用数据类型

|  |  |  |
| --- | --- | --- |
| **类型名称** | **ORACLE类型** | **描述** |
| 整型 | Number(n) | N表示数字的长度，默认值number-相当于number(11) |
| 小数 | Number(n,m) | N表示总长度，m小数位，number(5,2) |
| 字符串(固定长度) | Char(n) | N表示长度 |
| 字符串(可变长度) | Varchar2(n) | N表示长度 最高存储2000字符(1000汉字左右) |
| 日期 | date | ORACLE日期格式是固定。日期在存储时有一个本地化操作 |
| 大文本类型 | Clob | 存储海量文本数据。最大值可达4G |
| 大量二进制类型 | blob | 存储二进制数据，最大4g |

常见的Oracle数据类型了解：

Char类型：定义长度时，如果存储的数据小于长度，空位补空格。固定长度的类型

缺点：浪费存储空间

Char类型中存在空格，那么在程序取值比较时容易出错

在ORACLE中使用trim()函数，会造成char类型字段上的索引失效

优点：char类型在效率上比着其它字符类型的会快一点

Varchar2类型： 可变长度。存储时如果数据小于长度，varchar的长度按实际存储长度计算

优点：节省存储空间

缺点：效率比char类型低

注意：varchar2只有Oracle才有的。带2的是Oracle自己增强的数据类型，不是sql99规范中的。

Clob是longtext类型的代替品,存储超大量的字符串。如果varchar存储不了，可以用这个。但效率很低下，而且不能使用上索引。这玩意的存储大小不是按长度存储，按字节数存储的。

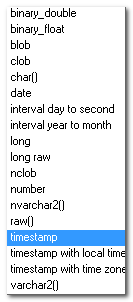
Blob 存储海量二进制类型。例：电影、音乐、高清图片。

缺点：每次存储时都需要转换为二进制进行存储

获取时把二进制转为数据

通常存储电影、音乐时存储的是文件的路径

Oracle所有的数据类型：



课外：试试各种语句和命令与mysql是否一样，比如crud的。

# SQL增强-单表查询

## 基本(基础)查询

学习目标：

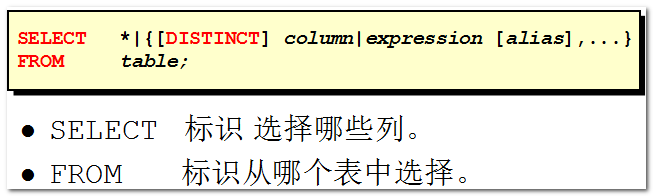
简单（基本）查询中涉及的几个注意点、不同点以及Plsql Developer工具的使用（后面简称工具）

后面的所有基本练习，除非特别说明，都使用scott用户登录。

### 基本查询语法

基本查询是指最基本的SQL select语句。

【语法】

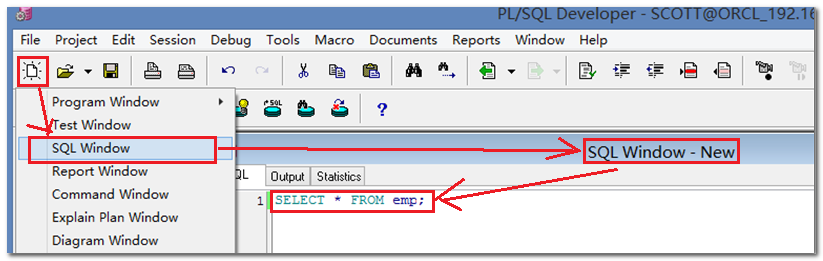


使用sqlplus登录普通用户：

|  |
| --- |
| sqlplus scott/tiger@localhost:1521/xe  sqlplus scott/tiger@192.168.0.249/xe--省略端口号，默认1521 |

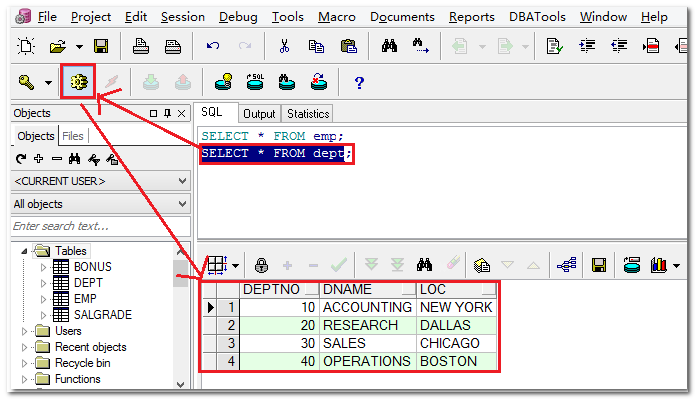
【知识点】如何使用工具进行查询

在plsql developer中打开查询窗口：



执行语句的操作：

选中要执行的语句，点击执行按钮或者按快捷键F8，下方会显示查询的结果。



显示下一页记录和显示剩余的所有页的记录：



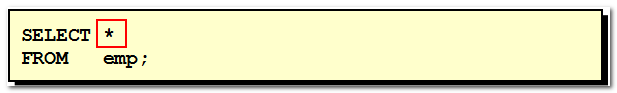
提示：

本课程不单独讲解该工具的详细使用，只是在用到哪些功能的时候再讲解相应的功能。如果想学习工具的详情使用，可参阅文档《plsql developer中文手册.pdf》

### 选择列

【语法】

选择全部列：



选择特定列：



【知识点】

两种语法效率是不同的，哪种效率高？

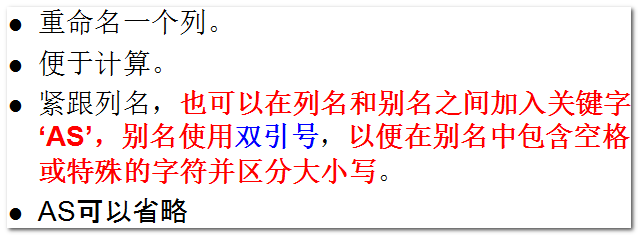
结论和原因：如果select \* 会全字段扫描，效率低，因此，尽量用指定的字段。

面试：请说出几种有哪些优化sql的方法？（答案：这就是其中一种）

关于sql语句优化的问题，将贯穿我们整个课程，实际工作和面试中就会遇到。需要大家自己整理一下sql优化的点。（任务）

### 别名

别名涉及到列的别名和表的别名。需要注意引号和用法。



【示例】

*--列的别名*

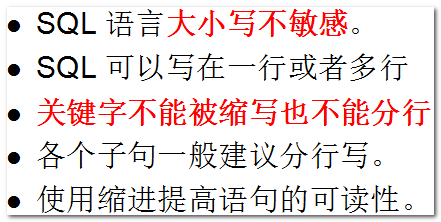
*--表的别名*

|  |
| --- |
| *--列的别名*  SELECT empno AS 员工编号 FROM emp; *--省略了双引号*  SELECT empno 员工编号 FROM emp;*--省略了as*  SELECT empno AS "员工编号" FROM emp; *--标准*  SELECT empno AS "员工 编号" FROM emp;*--空格的别名必须加引号*  *--表的别名*  SELECT empno,ename FROM emp t;  SELECT t.empno,t.ename FROM emp t;*--别名引用列*  SELECT emp.empno,emp.ename FROM emp;*--表直接引用列*  SELECT emp.empno,emp.ename FROM emp t;*--一旦指定了表的别名，则列的引用不能用原来的表名了。* |

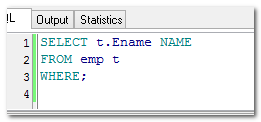
【知识点】

* 引号的问题。别名必须用双引号，也可以省略，而且还可以省略as。
* 表的别名一旦指定，列的引用中必须使用表的别名。

### 书写SQL的注意事项

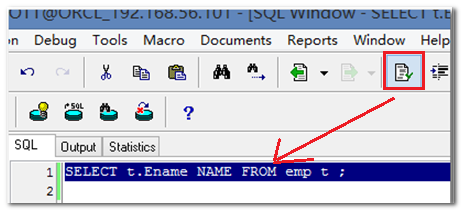


【示例】



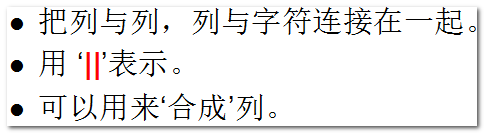
使用工具来格式化语句：

选中要格式化的语句，点击工具栏上的“美化”按钮，工具会自动将语句格式化：



格式化美化功能非常适合比较长的、复杂的语句的格式化。

### 字符串连接符||



【示例】

需求1：查询出员工的名字，要求显示的员工名字前面加上“姓名：”的字符串，显示结果参考：姓名：scott

需求2：将和员工的编号和员工的姓名都放在一个结果字段中显示。

|  |
| --- |
| *--需求1：查询出员工的名字，要求显示的员工名字前面加上“姓名：”的字符串，显示结果参考：姓名：scott*  SELECT '姓名：'|| ename 姓名 FROM emp;  *--需求2：将和员工的编号和员工的姓名都放在一个结果字段中显示。合成列*  SELECT empno||' '||ename FROM emp; |

提示：单引号代表的是字符串。

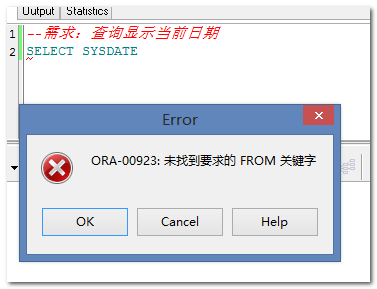
【知识点】

引号的问题。Oracle中如何选择单引号和双引号呢？基本上，只要是别名或不需要Oracle解析（运算）的字符串，用双引号，剩下的都用单引号（比如字符串）。

### 伪表-dual

mysql查询当前系统时间：SELECT SYSDATE();

但在Oracle中会报错：



提示：sysdate代表当前的系统时间函数。

报错原因：

Oracle和mysql的一个区别：

Oralce的查询语句必须是完整的，即必须满足语法select from

【示例】

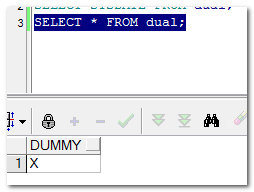
需求：查询显示当前的日期：

|  |
| --- |
| SELECT SYSDATE FROM dual;--sysdate代表当前日期的一个系统函数，dual是伪表，主要用来占位的，补充sql的。  SELECT 'a'||'b' FROM dual;  SELECT 1+2 FROM dual; |

DUAL 是一个‘伪表’（也称之为万能表），可以用来测试函数和表达式。也有人称之为万能表。

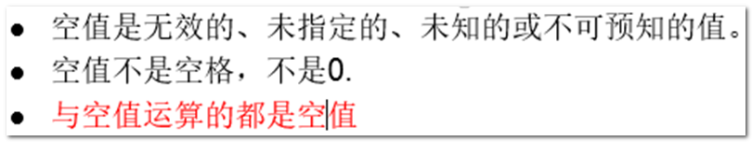
使用的时候可以用来占个语法的位置，来补充完整的sql。

伪表也是一张表，只是做了一些特殊处理。我们来看看：



注意：大家不要手动来维护这张表，这个表是由Oracle自动维护的。

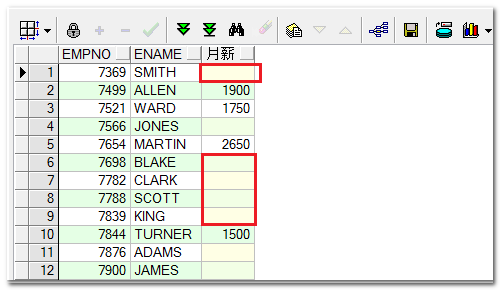
### 空值运算问题



【示例】

需求：查询所有员工的月薪（月薪=基本工资+奖金）

|  |
| --- |
| *--需求：查询所有员工的月薪（月薪=基本工资+奖金）*  SELECT ename, sal+comm 月薪 FROM emp;*--原因：与null运算的结果都是null* |



问题：

为何有的人月薪没值？但这些人明明有基本薪资（sal）。

原因：和null进行运算的都是null。

如何解决呢？我们会在单行函数中这一章节中进行解决。

提示：

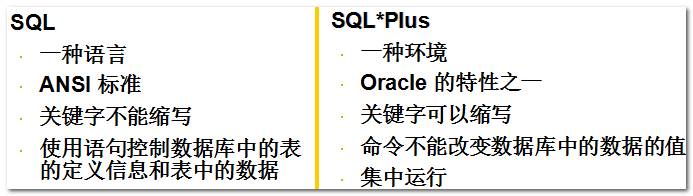
后续课程中会有更多与null相关的例子。请同学们注意整理。

### SQL语句和SQL\*Plus命令

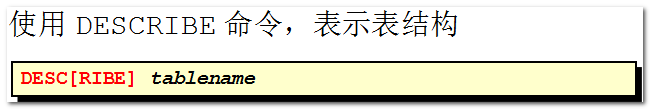
目标：

了解什么是命令，什么是语句。

两者对比如下：



关键字可以缩写，比如显示表结构的命令：



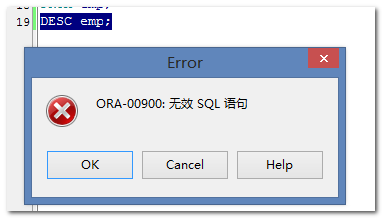
【示例】

需求：分别用完整命令和简写命令来显示emp表的结构：

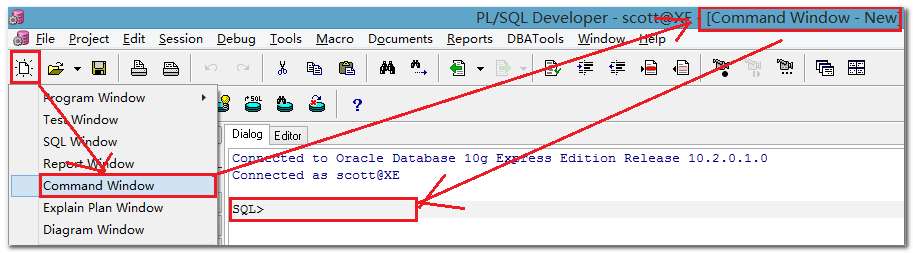
|  |
| --- |
|  |

【关于工具窗口使用选择】

工具的命令行窗口下，既能执行命令，也能执行sql，但在sql窗口下只能运行sql，如果在sql窗口运行命令，会出现错误信息：



工具中的命令窗口的调用方法：



友情提示：

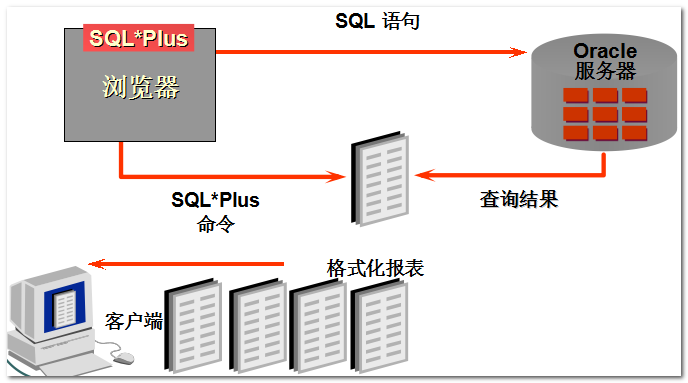
工具的命令窗口和sqlplus自带的命令窗口在有时候还是有少许区别的。后面会提到。

### 导出报表-扩展-了解

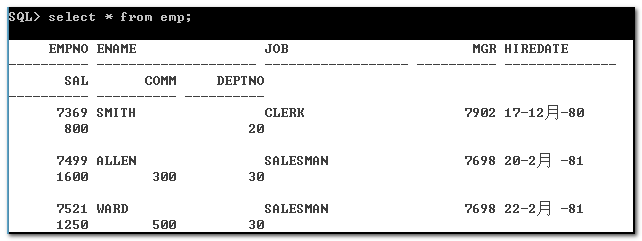
报表是向上级报告情况的表格。简单的说：报表就是用表格、图表等格式来动态显示数据，可以用公式表示为：“报表 = 多样的格式 + 动态的数据”。

计算机提供的报表可以由数据库直接生成，也可以由专业的报表软件生成。

* 数据库软件：它们可以拥有动态变化的数据，但是这类软件一般只会提供，最简单的表格形式来显示数据。它们没有实现报表软件的"格式多样化"的特性。
* 报表软件：它们需要有专门的报表结构来动态的加载数据，同时也能够实现报表格式的多样化。（eclipse官方提供birt）



使用sqlplus导出报表，不太方便。

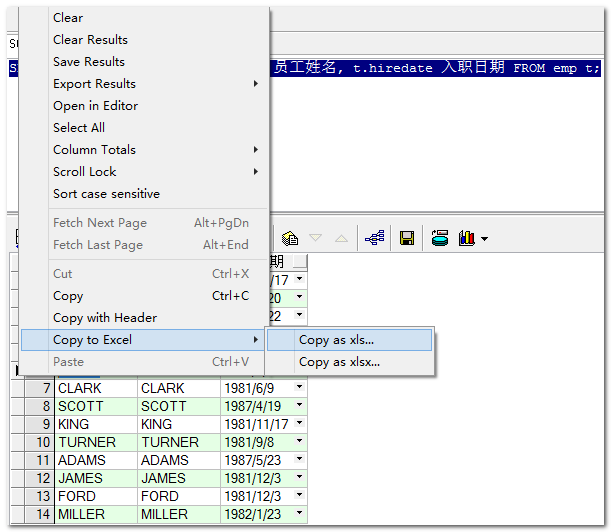


在没有专业的报表系统或报表工具的情况下，推荐直接使用SQLplus Developer工具进行简单报表的导出，导出步骤参考如下：

1. 执行期望的sql语句。

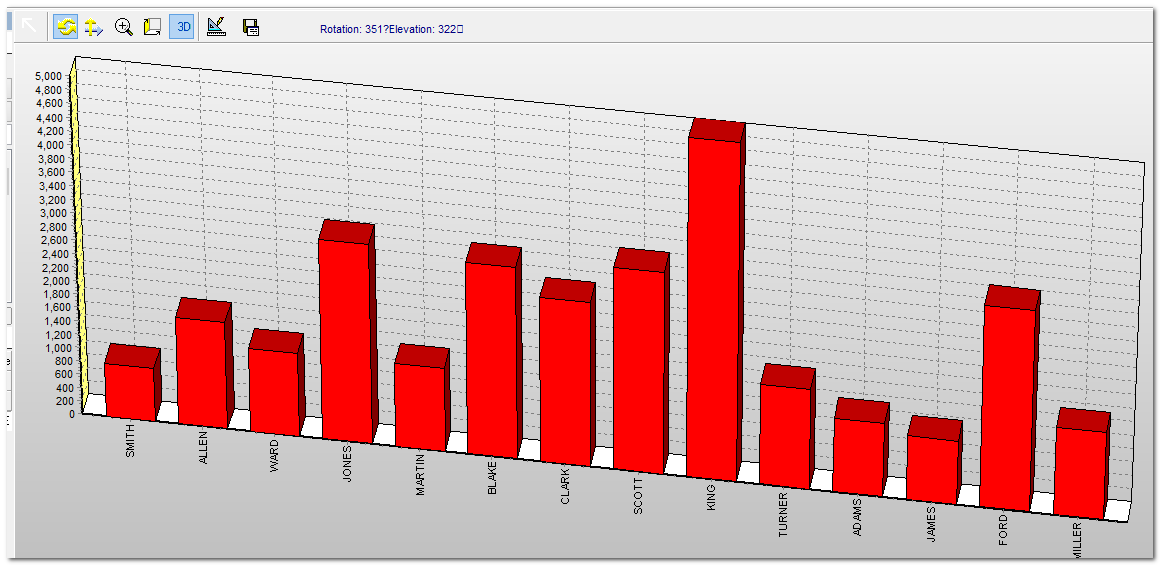
|  |
| --- |
| SELECT t.empno 员工编号, t.ename 员工姓名, t.hiredate 入职日期 FROM emp t; |

2. 选中需要导出的报表数据（如果不选中就是导出所有的），在左侧的工具栏上点击右键，选择copy to excel...copy as xls



图形报表：





学习提醒：

复杂的查询以及查询技巧主要是报表查询的时候要用的到！同学们注意理解学习的意义。

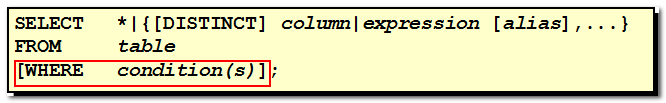
## 过滤子句where

学习目标：

过滤和排序要注意的几个问题，如空值问题、转义字符、条件运算优先级事项等。

### 过滤语法

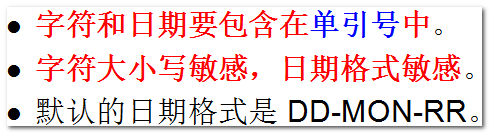
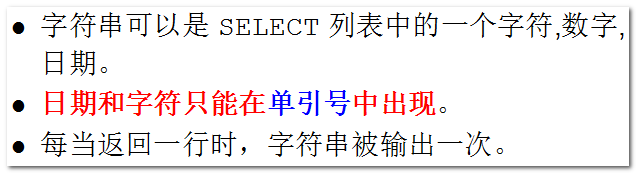
过滤就是使用where子句，将不满足条件的行过滤掉。



注意：

* Where子句紧跟from子句。
* where的过滤条件是对于每一行数据的。

### 字符和日期



这里强调两个事情：

字符大小写的问题和默认日期格式的问题。

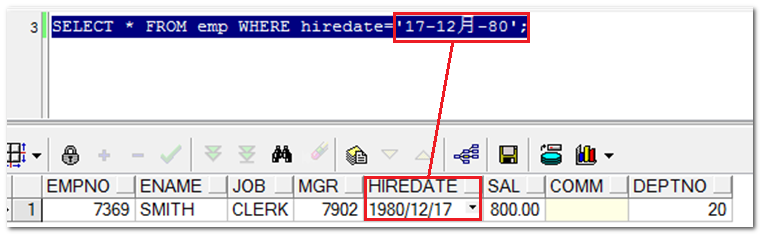
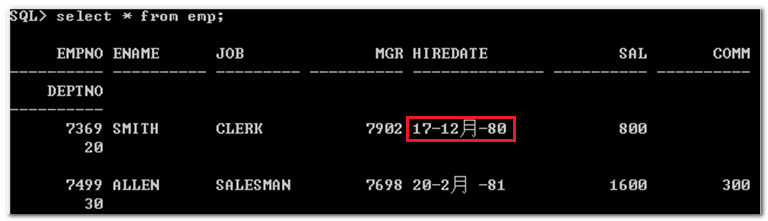
【示例】

需求1：查询关于KING这个人的记录。

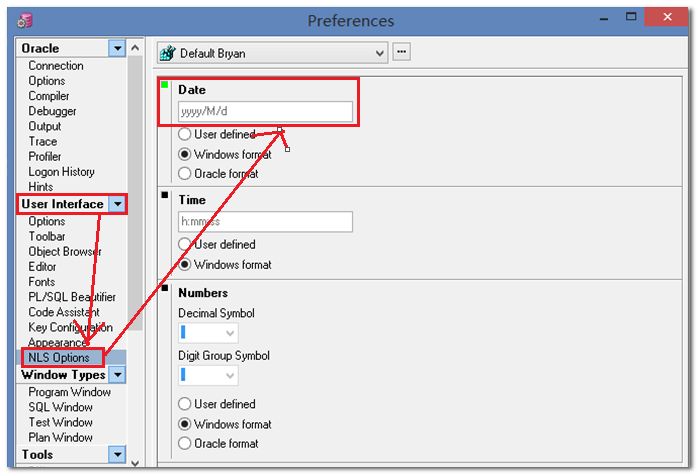
需求2：查询入职日期是1987/4/19的员工的信息。

|  |
| --- |
| *--需求1：查询关于KING这个人的记录。*  SELECT \* FROM emp WHERE ename='king';*--错误*  SELECT \* FROM emp WHERE ename='KING';*--正确，具体数据库的值是区分大小写。*  *--需求2：查询入职日期是1987/4/19的员工的信息*  SELECT \* FROM emp WHERE hiredate ='1987/4/19';*--数据库默认是日期的格式不对。导致无法将字符串隐式转换为日期*  SELECT \* FROM emp WHERE hiredate ='17-12月-80';*--数据库默认的日期格式，字符串可以隐式转换为日期*  *--1987/4/19格式是工具给你转的* |

【疑问】为什么工具显示的不是默认值（和sqlplus显示的不一样）？



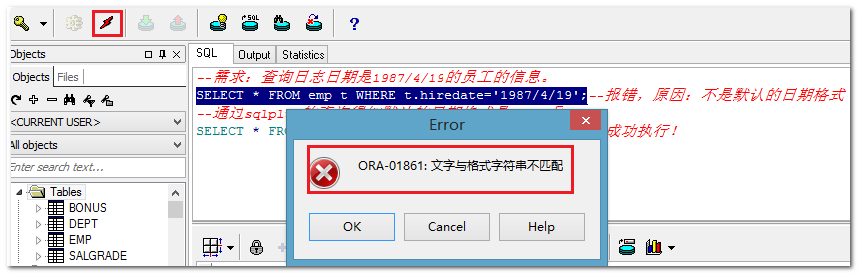
原因是工具自己转换了！工具显示的日期格式配置如下：



【工具使用提示】

1．注意工具的错误提示方式：

2．异常会导致语句执行过程被卡住。见工具栏上的闪电图标。此时可以选择终止运行或排除异常后语句继续执行。



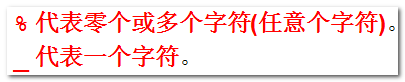
### Escape-转义字符

准备测试数据。

添加一条ename的值为xiao\_ming的测试数据，可以使用insert语句：

|  |
| --- |
| INSERT INTO emp(empno,ename) VALUES(1001,'xiao\_ming');  commit; |

【回顾】通配符：



【示例】

需求1：查询名称是带有”x”字符的员工的记录信息。

需求2：查询员工名称中含有下划线（“\_”）的员工.

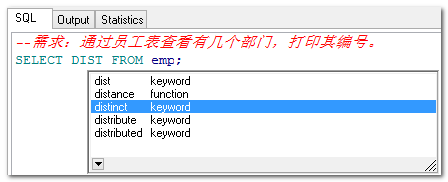
需求3：查询姓名是4个字符的员工的信息。

|  |
| --- |
| *--需求1：查询名称是带有”x”字符的员工的记录信息。*  SELECT \* FROM emp WHERE ename LIKE '%x%';  *--需求2：查询员工名称中含有下划线（“\_”）的员工.*  SELECT \* FROM emp WHERE ename LIKE '%\_%'; *--为什么全查出来：sql的通配符%(任意多个字符) \_(任意一个)*  SELECT \* FROM emp WHERE ename LIKE '%\\_%' ESCAPE '\';*--用ESCAPE来声明一个转义字符，语句中，该转义字符之后的字符，都作为普通字符来处理。*  SELECT \* FROM emp WHERE ename LIKE '%|\_%' ESCAPE '|';  *--需求3：查询姓名是4个字符的员工的信息。*  SELECT \* FROM emp WHERE ename LIKE '\_\_\_\_';  /\*  作用：假如你允许用户注册的时候带下划线，或者表单有个字段是备注，那么用户，在写备注的时候，可能会写下划线。  注册zhang\_bo,此时，你想知道数据库中有多少人的用户名是带下划线的。  \*/  SELECT COUNT(\*) FROM emp WHERE ename LIKE '%/\_%' ESCAPE '/'; |

【工具的使用提示】：

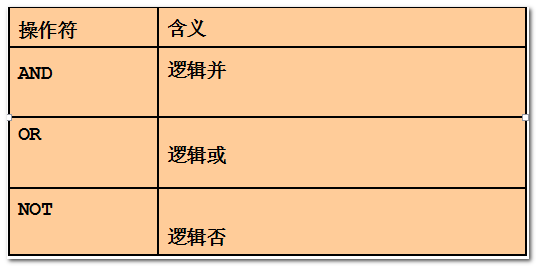
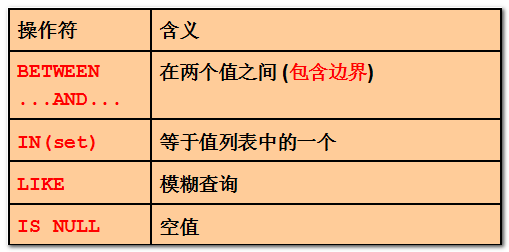
对于某个关键字不确定单词的编写的时候，可以使用工具的自动提示功能。

当输入前几个字符，则工具会自动提示相关关键字的列表。如果不小心关闭了提示或者未出现提示，则可以通过在单词上按F6。

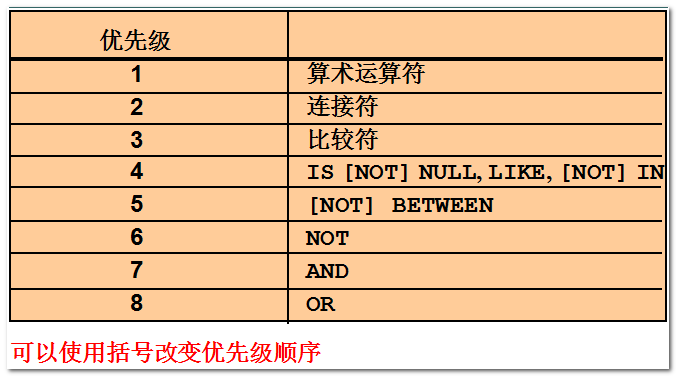


### 条件运算符

常见的条件运算符如下：



运算符的优先级：



### in和not in过滤时的空值问题

【示例】要求使用in和not in

需求1：查询10号部门和20号部门的员工；

需求2：查询10号和20号以及没有部门的员工部门的员工；

需求3：查询不是10号和20号以及没有部门的员工部门的员工；

|  |
| --- |
| *--需求1：查询10号部门和20号部门的员工；*  SELECT \* FROM emp WHERE deptno IN (10,20);  SELECT \* FROM emp WHERE deptno =10 OR deptno=20;  *--需求2：查询10号和20号以及没有部门的员工部门的员工；*  SELECT \* FROM emp WHERE deptno IN (10,20,NULL);  SELECT \* FROM emp WHERE deptno =10 OR deptno=20 OR deptno=NULL;*--和null运算的永远是false*  *--需求3：查询不是10号和20号以及没有部门的员工部门的员工；*  SELECT \* FROM emp WHERE deptno NOT IN (10,20,NULL);  SELECT \* FROM emp WHERE deptno deptno =10 and deptno=20 AND deptno=NULL;*--和null运算的永远是false* |

不正常结果的原因分析：

SELECT \* FROM emp WHERE deptno IN(10,20,NULL);*--小明没查出来*

原因：

deptno IN(10,20,NULL)===🡺deptno =10 or deptno=20 or deptno=null

true or true or 永远是false-====结果true

分析：

SELECT \* FROM emp WHERE deptno NOT IN(10,20,NULL);*--没有一条数据*

原因：

deptno NOT IN(10,20,NULL)====》deptno!=10 and deptno!=20 and deptno=!null

true and true and永远 false===🡺false

解决方案：

|  |
| --- |
| *--解决：*  SELECT \* FROM emp WHERE deptno IN (10,20) OR deptno IS NULL;  SELECT \* FROM emp WHERE deptno NOT IN (10,20) AND deptno IS NOT NULL; |

### 条件运算的优先级

思考：下面这条语句的两个条件的执行顺序是什么？（注：condition1和condition2是两个条件表达式）



答案：先执行2，再执行1。

原因是：

where条件子句的解析顺序：从右到左

【知识点】

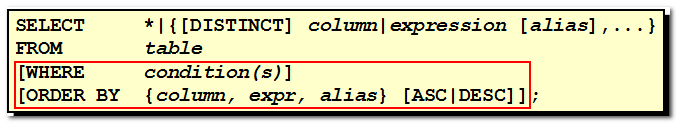
SQL优化：（where条件特别多的情况下，就有效果了）

对于and，应该尽量把假的放到右边。

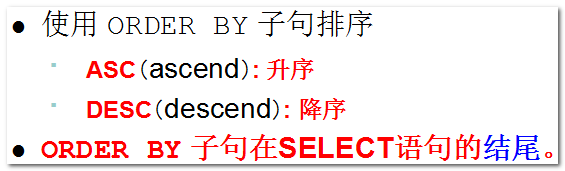
对于or，应该尽量把真的放到右边。

## 排序子句Order by

### 排序语法



两个注意点：



### 关键字作用范围

【示例】

需求：查询所有员工信息，要求按照部门和员工号的倒序排序，

|  |
| --- |
| *--需求：查询所有员工信息，要求按照部门和员工号的倒序排序，*  SELECT \* FROM emp ORDER BY deptno,empno DESC;*--desc关键字，只能对最靠近的一个字段起作用*  SELECT \* FROM emp ORDER BY deptno desc,empno DESC; |

### 别名列号排序-了解

【示例】

需求：查询所有员工信息，要求显示姓名和基本年薪（基本薪资\*12），并且要求根据基本年薪正序排列。

语句要求：分别使用别名、不使用别名、使用列号来排序

|  |
| --- |
| *--需求：查询所有员工信息，要求显示姓名和基本年薪（基本薪资\*12），并且要求根据基本年薪正序排列。*  *--语句要求：分别使用别名、不使用别名、使用列号来排序*  SELECT ename,sal\*12 FROM emp ORDER BY sal \*12 ASC;*--asc可以省略*  SELECT ename 姓名,sal\*12 年薪 FROM emp ORDER BY 年薪 ASC;*--根据别名排序*  SELECT ename 姓名,sal\*12 年薪 FROM emp ORDER BY 2 ASC;*--根据列号：第二列* |

### 空值排序显示的问题

【知识点】关键字nulls last的使用。

【示例】

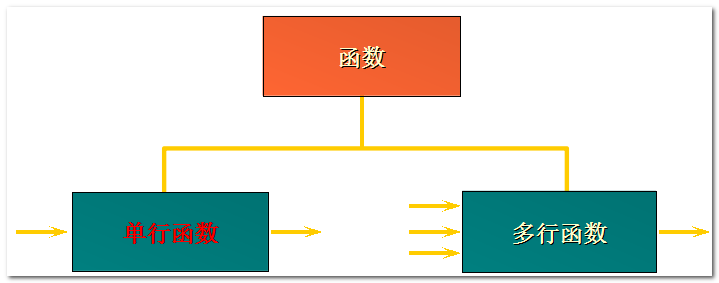
需求：根据基本薪资的年薪倒序序排列

|  |
| --- |
| *--需求：根据基本薪资的年薪倒序序排列*  SELECT ename,sal\*12 FROM emp ORDER BY sal \*12 DESC NULLS LAST;*--排序列的空值放在最后* |

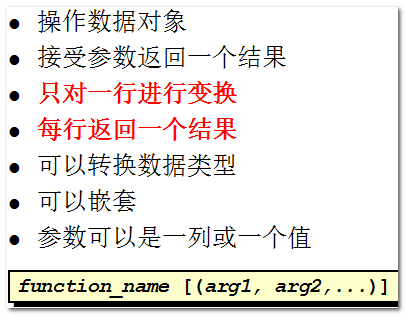
## 单行函数

### 函数的分类

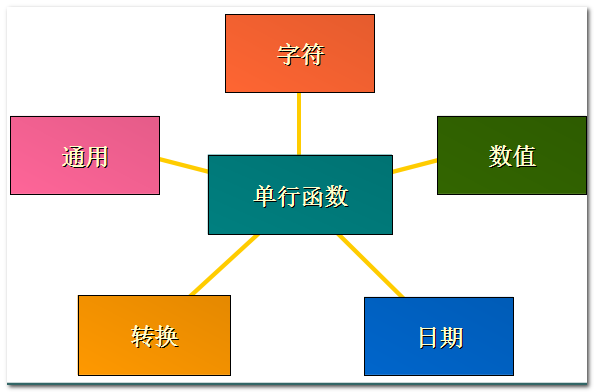
Oracle的内置函数分为单行函数和多行函数（多行函数还称之为组函数、聚集函数等）。



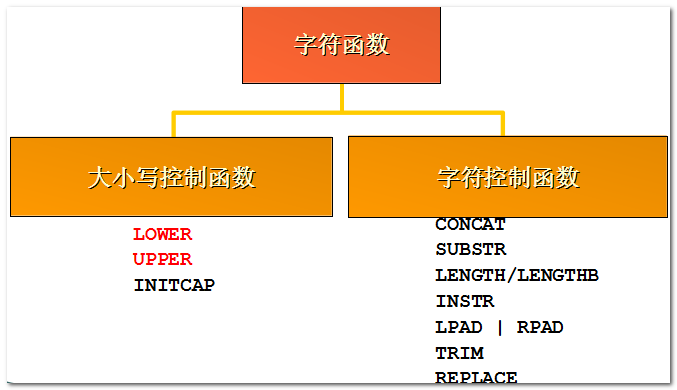
### 单行函数的概念



### 单行函数的分类



### 字符函数



* 大小写控制函数

【示例】

需求1：查询出KING的这个员工的信息。

|  |
| --- |
| *--需求1：查询出KING的这个员工的信息。*  SELECT \* FROM emp WHERE ename ='king';  SELECT \* FROM emp WHERE LOWER(ename) ='king';*--会将数据库的值转换成小写*  SELECT \* FROM emp WHERE ename =UPPER('KinG');*--不管用户输入的是大写还是小写，还是大小混合写*  SELECT UPPER('KiNg') FROM dual;  SELECT empno,INITCAP(ename) FROM emp *--首字母大写*  SELECT INITCAP('zhanG bo shi ge hao ren') FROM dual; |

【讨论】：

上述的需求，到底是使用upper还是使用lower呢？

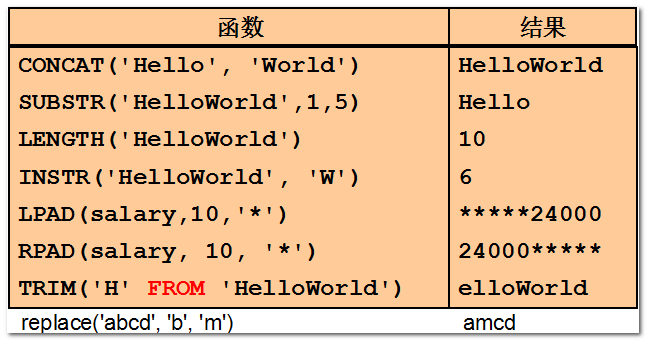
一般根据需求来选择的。

如果将函数放到字段上，会每行的该字段都会转换，效率低一些。--sql优化

因此，一般情况下，建议将转换函数放到固定值上面(好处之一就是只需要转换一次，还有一个好处，就是你不知道用户到底输入的是大写还是小写还是混合写，更适应业务)。



* 字符控制函数：



【示例】演示部分，其他**课后练习！**

需求1：替换字符串'abcd'中的’bc’为’BABY，最终显示为’aBABYd’

需求2：去掉' Hello World '前后的空格

需求3：去掉'Hello WorldH'前后的H字符（提示：使用from关键字）

|  |
| --- |
| *--需求1：替换字符串'abcd'中的’bc’为’BABY，最终显示为’* *aBABYd’*  SELECT REPLACE ('abcd','bc',' BABY ') FROM dual;  *--需求2：去掉' Hello World '前后的空格*  SELECT TRIM(' Hello World ') FROM dual;  *--需求3：去掉'Hello WorldH'前后的H字符（提示：使用from关键字）*  SELECT TRIM('H' FROM 'Hello WorldH') FROM dual; |

【提示】：

Oracle的函数非常多，建议大家只记住课堂上讲解的常用的几个就基本够用了，其他的可以查阅手册：



### 数字函数



【示例】

需求：钱数：*1385.56，*分别根据不同场景进行处理显示不同结果：买东西（抹零头：1385，1380）、发工资（发钱了：1386）

|  |
| --- |
| *--需求：钱数：1385.56，分别根据不同场景进行处理显示不同结果：买东西（抹零头：1385，1380）、发工资（发钱了：1386）*  SELECT TRUNC(1385.56) 买东西抹零头,TRUNC(1385.56,-1) 抹零头,  ROUND(1385.56) 发钱,ROUND(1385.56,1) 发钱  FROM dual; |

【提示】：

Round和trunc函数，除了对数字起作用外，对于日期也是起作用的。（后面会提到）

### 日期函数

【示例】

问题：日期可以相减么？日期可以相加么？

|  |
| --- |
| --问题：日期可以相减么？日期可以相加么？  SELECT SYSDATE-SYSDATE FROM dual;--日期相减一般是为了计算两个日期之间间隔  SELECT SYSDATE+SYSDATE FROM dual;--日期相加没意义 |

常用函数（了解，用时查询）



next\_day(基础日期,星期几)

星期几,是从周日开始，分别数字为1，2，3。。。。

【示例】

需求1：计算员工的工龄（工龄：当前的日期和入职的日期的差），要求分别显示员工入职的天数、多少月、多少年。

需求2：查看当月最后一天的日期。

需求3：查看指定日期的下一个星期天或星期一的日期。（next\_day(基础日期,星期几)）

|  |
| --- |
| *--需求1：计算员工的工龄（工龄：当前的日期和入职的日期的差），要求分别显示员工入职的天数、多少月、多少年。*  SELECT ename, SYSDATE-hiredate 工龄天,  (SYSDATE-hiredate)/30 工龄月不精确,months\_between(SYSDATE,hiredate) 工龄月精确,  TRUNC (months\_between(SYSDATE,hiredate)/12) 工龄年精确  FROM emp;  --需求2：查看当月最后一天的日期。  SELECT last\_day(SYSDATE) FROM dual;  --需求3：查看指定日期的下一个星期天或星期一的日期。（next\_day(基础日期,星期几)）--星期日是1，星期一2  SELECT next\_day(SYSDATE,1) FROM dual;  SELECT next\_day(SYSDATE,2) FROM dual; |

【扩展知识】

扩展：时间戳systimestamp关键字。

【示例】

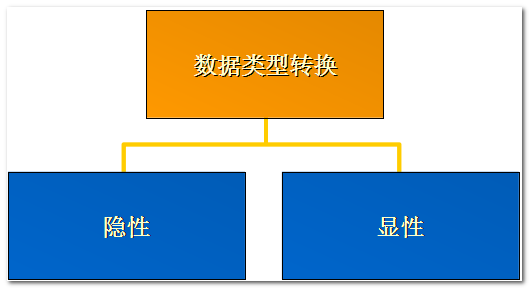
查看当前系统默认精度的日期时间和更高精度的时间戳，要求显示结果如下：



|  |
| --- |
| SELECT SYSDATE,Systimestamp FROM dual; |

### 转换函数

数据类型转换分类：



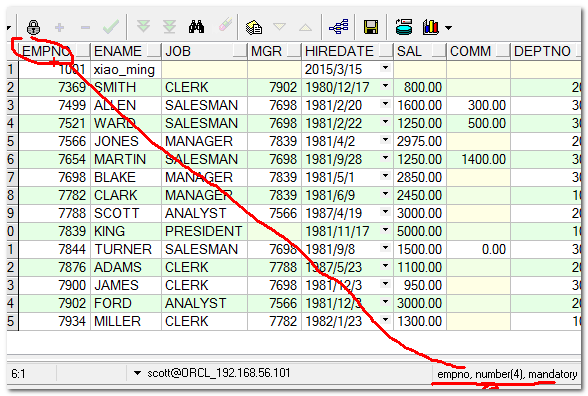
* **隐式转换**

【示例】

需求：查询10号部门的信息，分别使用数字和字符串作为条件的值。

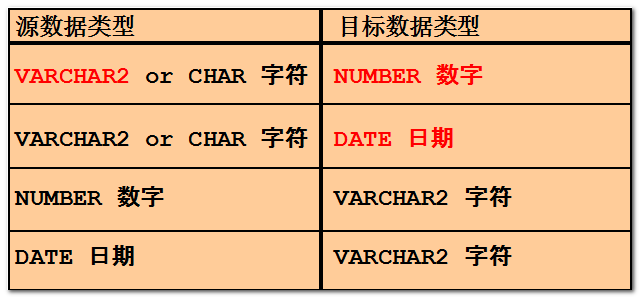
|  |
| --- |
| *--需求：查询10号部门的信息，分别使用数字和字符串作为条件的值。*  SELECT \* FROM emp WHERE deptno=10;  SELECT \* FROM emp WHERE deptno='10';*--字符串隐式转换为数字了*  SELECT \* FROM emp WHERE deptno='10q';*--隐式转换的前提，是能转换才可以。* |

【工具的使用补充】在查询数据的时候，通过工具来快捷查看字段的数据类型：



**隐式转换的条件：**

Oracle可以自动的完成下列类型（三种）的转换：

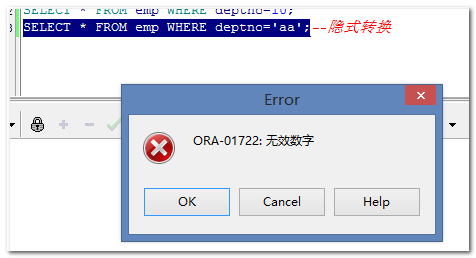


**非法转换：**

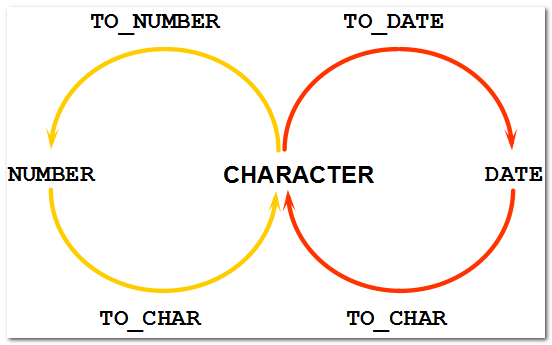
隐式转换的前提是：被转换的对象是可以转换的。下面的语句会报错：



运行会抛一个异常：



* **显示转换（三个函数）**

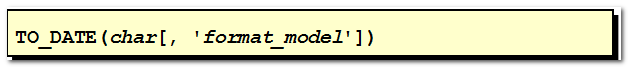


三个转换函数的语法：

将日期或数字转换成字符



将字符转换成日期



将字符转换成数字



**【提示：记忆方式】：第一个参数都是要转换的目标（到底用哪个函数，跟目标有关系），第二个都是转换的格式。**

【示例】

需求1：显示今天的完整日期，结果参考：“2015-07-06 11:07:25”。

需求2：显示今天是几号，不包含年月和时间，结果参考：“8日”。

需求3：显示当月最后一天是几号，结果参考：”30“。

需求4：xiaoming的入职日期是2015-03-15，由于其入职日期当时忘记录入，现在请将其插入到emp表中。

需求5：查看2015年2月份最后一天是几号，结果参考“28“

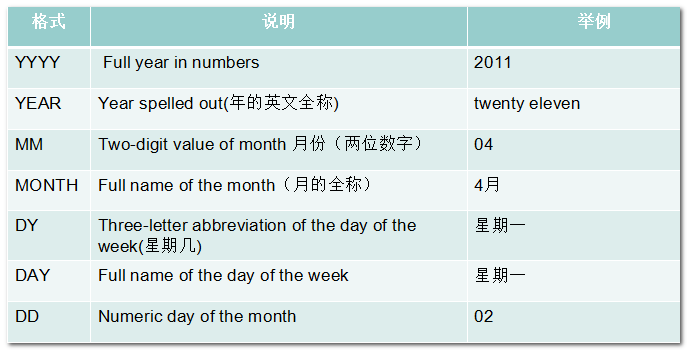
|  |
| --- |
| *--需求1：显示今天的完整日期，结果参考：“2015-07-06 11:07:25”。*  SELECT to\_char(SYSDATE,'yyyy-MM-dd HH:mm:ss') FROM dual;*--java的日期格式，和sql的不一样*  SELECT to\_char(SYSDATE,'yyyy-mm-dd hh24:mi:ss') FROM dual;*--sql--24小时制*  SELECT to\_char(SYSDATE,'yyyy-mm-dd hh:mi:ss') FROM dual;*--sql--12小时制*  SELECT to\_char(SYSDATE,'yyYy-Mm-Dd hH24:mi:ss') FROM dual;*--格式不区分大小写*  *--需求2：显示今天是几号，不包含年月和时间，结果参考：“8日”。*  SELECT to\_char(SYSDATE,'dd')||'日' FROM dual;*--字符串拼接方式*  SELECT to\_char(SYSDATE,'dd"日"') FROM dual;*--格式中直接加入固定值*  *--需求3：显示当月最后一天是几号，结果参考：”30“。*  SELECT to\_char(last\_day(SYSDATE),'dd') FROM dual;  *--需求4：xiaoming的入职日期是2015-03-15，由于其入职日期当时忘记录入，现在请将其插入到emp表中。*  UPDATE emp SET hiredate =to\_date('2015-03-15','yyyy-mm-dd') WHERE ename ='xiao\_ming';  COMMIT;  *--需求5：查看2015年2月份最后一天是几号，结果参考“28“*  SELECT last\_day(to\_date('201502','yyyymm')) FROM dual;  SELECT to\_date('201502','yyyymm') FROM dual;*--日期的默认值，不指定日期，默认1号* |

【注意】和java不同，Oracle的日期格式对大小写不敏感。

【使用上的选择】

到底要用哪个函数，关键是传进来的目标的类型和最终需要的结果类型。

**日期格式的常见元素：**

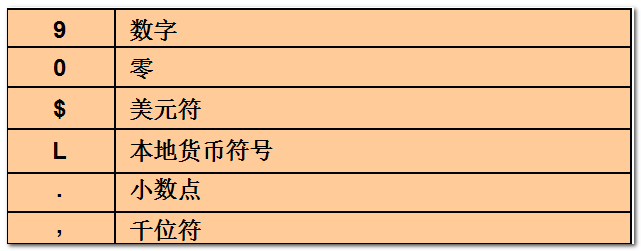


【示例】

需求：查看显示今天是星期几

|  |
| --- |
| SELECT to\_char(SYSDATE,'day') FROM dual; |

**数字格式的常见元素：**



提示：9代表任意数字，可以不存在。0代表数字，如果该位置不存在，则用0占位。

【示例】

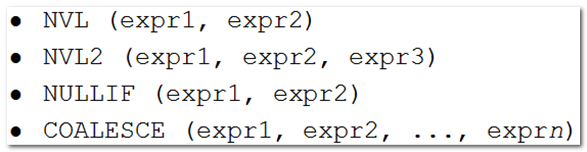
需求：查询员工的薪水，格式要求：两位小数，千位数分割，本地货币代码。

|  |
| --- |
| *--需求：查询员工的薪水，格式要求：两位小数，千位数分割，本地货币代码。*  SELECT ename,sal,to\_char(sal,'L99,999.00') FROM emp;  SELECT ename,sal,to\_char(sal,'L00,000.00') FROM emp; |

### 滤空函数（通用函数）

滤空函数也称为通用函数，其特点是：适用于任何数据类型，同时也适用于空值。

常见的滤空函数：



使用方法：

* nvl(a,c)，当a为null的时候，返回c，否则，返回a本身。
* nvl2(a,b,c),当a为null的时候，返回c，否则返回b

其中，nvl2中的2是增强的意思，类似于varchar2。

* nullif(a,b),当a=b的时候，返回null，否则返回a
* coalesce(a,b,c,d)，从左往右查找，当找到第一个不为null的值的时候，就显示这第一个有值的值。

【示例】

需求：查询员工的月收入（基本薪资+奖金）

|  |
| --- |
| *--需求：查询员工的月收入（基本薪资+奖金）*  SELECT ename,sal+nvl(comm,0) 月收入 FROM emp;  SELECT ename ,NVL2(sal,sal,0)+nvl(comm,0) FROM emp;*--为了小明*  SELECT coalesce(NULL,NULL,1,2) FROM dual;*--返回第一个不为空的值* |

### 条件表达式

条件表达式的作用是：在SQL语句中使用判断的逻辑（类似于IF-THEN-ELSE）来呈现个性化的数据。

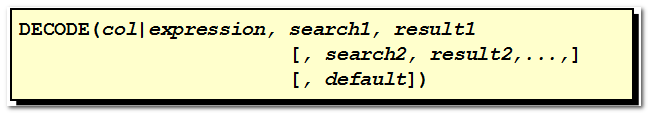
条件判断语句有两种：

* CASE 表达式：SQL99的语法，类似Basic，比较繁琐
* DECODE 函数：Oracle自己的语法，类似Java，比较简单

1. Decode函数

也可以理解为解码翻译函数。

语法：



语法解释：

decode (字段名,要翻译的原始值1，翻译后的值1，......，其他不满足翻译条件的默认值)

【示例】

需求:要将工种job的英文转换为中文

|  |
| --- |
| *--需求:要将工种job的英文转换为中文*  SELECT ename,job,  DECODE(job,'CLERK','职员','SALESMAN','销售人员','MANAGER','经理','其他工种')  FROM emp; |

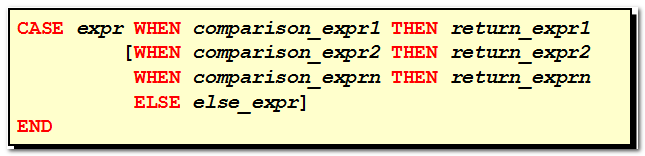
业务场景补充：

比如人的性别：一般数据库存放的是：0和1，2，在直接出报表的时候，就需要转换显示。

|  |
| --- |
| SELECT NAME 姓名,DECODE(sex,1,'男',0,'女','人妖') 性别 FROM TABLE; |

1. Case子句

语法：



语法解释：

case 字段 when 要翻译的值 then 翻译的结果

when 要翻译的值 then 翻译的结果

......

else 默认的结果值

end

【示例】

|  |
| --- |
| SELECT \* FROM emp;  --需求:要将工种job转换为中文  SELECT t.ename,  CASE job WHEN 'CLERK' THEN '办事员'  WHEN 'SALESMAN' THEN '销售人员'  ELSE '其他人员'  END  FROM emp t;  --两种语法--第二种很复杂。。。。---虽然复杂但灵活  SELECT t.ename,  CASE WHEN job='CLERK' THEN '办事员'  WHEN job='SALESMAN' THEN '销售人员'  ELSE '其他人员'  END  FROM emp t; |

1. case子句增强

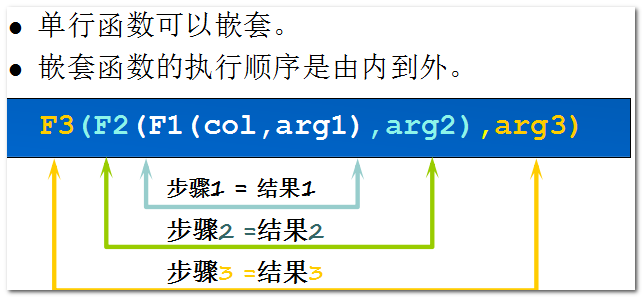
需求：查看公司员工的工资情况，要求显示员工的姓名、职位、工资、以及工资情况。如果是工资小于1000，则显示“工资过低”，工资大于1000小于5000为“工资适中”，工资大于5000的，则显示“工资过高”：

|  |
| --- |
| SELECT ename,job,sal,  CASE WHEN sal<1000 THEN '工资过低'  WHEN sal BETWEEN 1000 AND 5000 THEN '工资适中'  when sal IS NULL THEN '没工资酱油瓶'  ELSE '工资太高'  END  FROM emp; |

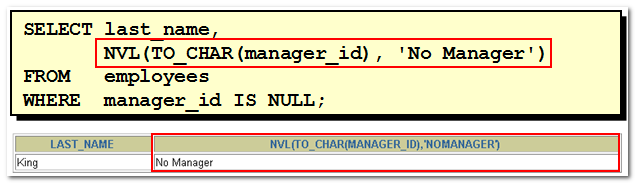
Decode和Case的使用选择：

在Oracle中，翻译值的这种条件判断，优先使用decode，因为简单明了，且Oracle有一定的优化；更复杂的条件判断或者其他的关系型数据库，只能使用Case子句。

### 嵌套函数



【示例】了解即可

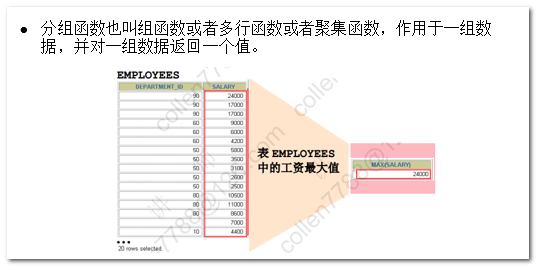


## 多行函数

### 多行函数的概念

多行函数也称之为分组函数、聚集函数。

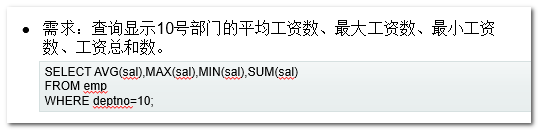
简答的说就是把多行的值汇聚计算成一个值。。



常见的分组函数：



【示例】



### 空值问题

多行函数会自动滤空。

【示例】

需求：统计计算员工的平均奖金。（不同需求不同结果）

|  |
| --- |
| *--需求：统计计算员工的平均奖金。（不同需求不同结果）*  SELECT AVG(comm) FROM emp;*--统计的是有奖金的人的平均奖金*  *--相当于*  SELECT SUM(comm)/COUNT(comm) FROM emp;*--多行函数会自动滤空*  *--统计所有人的平均奖金*  SELECT AVG(nvl(comm,0)) FROM emp; |

### Count的使用注意点

count统计时可以使用不同的对象：\*,column,1，不同的对象统计的方式和效率都不同。

【示例】

需求：统计员工的数量，要求使用count的多种统计方式，并分析原因。

|  |
| --- |
| *--需求：统计员工的数量，要求使用count的多种统计方式，并分析原因。*  SELECT COUNT(\*) FROM emp;*--效率最低，全表全字段扫描*  SELECT COUNT(empno) FROM emp; *--按照主键列来统计--效率也挺高，语法角度来说，不通用*  SELECT COUNT(1) FROM emp;*--统计的是字符是1的这一列，效率高（原因，这一列只有一个字符，运算的时候，数据流很小，而且是固定列）*  SELECT 1,ename FROM emp;  SELECT COUNT(11111111111) FROM emp;*--统计的参数不是列号* |

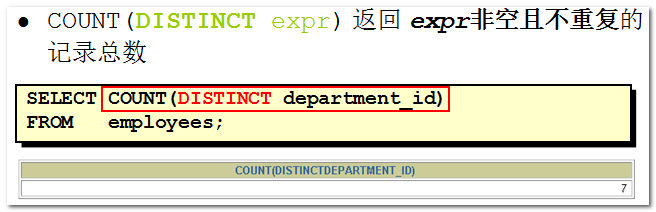
count（主键）效率很高。主键自动会有索引（提升查询效率的），这个效率最高。

count（\*）全表字段扫描，效率低，但现在的数据库都对此做了优化，底层根据主键

count（1）效率高，如果没有索引，这个效率比较高

### 嵌套函数

distinct可用来过滤掉多余的重复记录只保留一条，但往往只用 它来返回不重复记录的条数，而不是用它来返回显示不重记录的所有值。因此，一般和count配合使用，作为统计非空且不重复的记录数。



【示例】

需求：*--查看有几个部门，通过emp表*

查询公司发放了几种数量的奖金，要求员工是有奖金的，且奖金都不重复。

|  |
| --- |
| *--查看有几个部门，通过emp表*  SELECT Distinct(deptno) FROM emp;  *--需求：查询公司发放了几种数量的奖金，要求员工是有奖金的，且奖金都不重复。*  SELECT COUNT(Distinct(comm)) FROM emp; |

【友情提示注意】：DISTINCT关键字效率会比较低，如果仅仅是为了显示不重复的记录，建议使用group by；



慢的原因是：

distinct只有用二重循环查询来解决，而这样对于一个数据量非常大的表来说，无疑是会直接影响到效率的。

### 关于聚合函数的思考

下面的语句是否正确？

|  |
| --- |
| SELECT deptno,MAX(sal) FROM emp; |

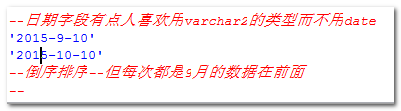
【分析】

因为聚合函数处理的是数据组，在本例中，MAX函数将整个EMP表看成一组，而deptno的数据没有进行任何分组，因此SELECT语句没有逻辑意义。

要想解决这个问题，需要对deptno进行分组。

[了解]：MAX()和MIN()函数不仅可以作用于数值型数据，也可以作用于字符串或是日期时间数据类型的数据。

补充字符串对比：

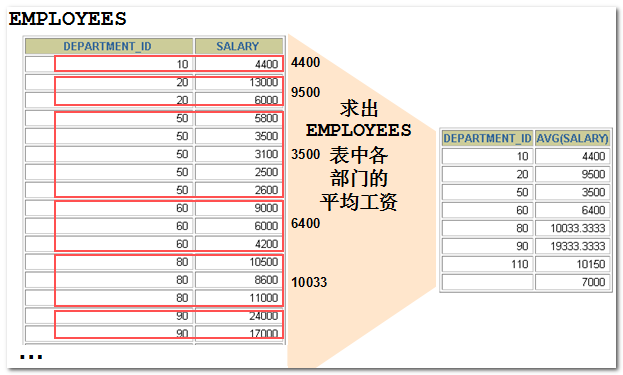


解决方案：

* + - * 1. 日期直接存成日期格式（date）
        2. 日期标准，2015-09-10

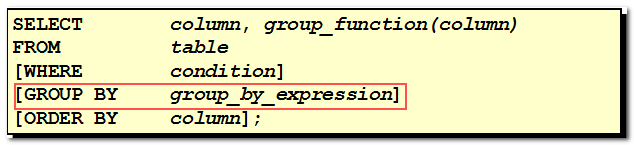
## 分组子句

### 分组数据的概念



可以通过group by子句达到效果。

### 分组子句的语法



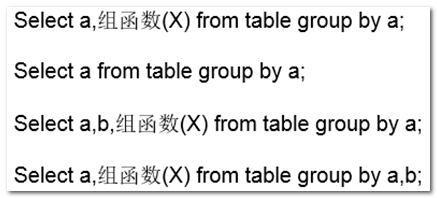
作用：可以使用GROUP BY 子句将表中的数据分成若干组

### 分组子句的要求

在SELECT 列表中所有**未包含在组函数**中的**列**都应该包含在 GROUP BY 子句中。

反之，包含在 GROUP BY 子句中的列不必包含在SELECT 列表中

请判断下面的示例的语法是否正确：



【示例】

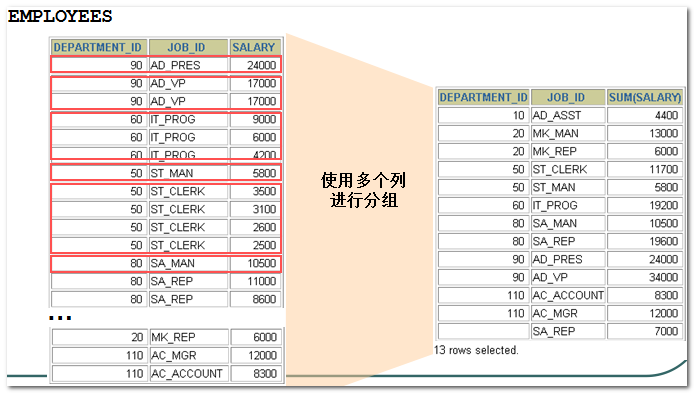
需求1：查询显示各个部门的平均薪资情况，并且按照部门号从低到高排列。

需求2：查询显示各个部门的不同工种的平均薪资情况，并且按照部门号从低到高排列。

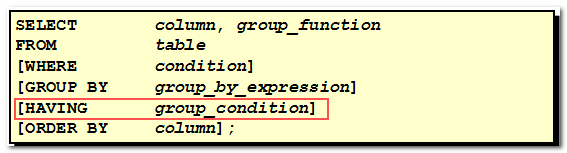
|  |
| --- |
| *--需求1：查询显示各个部门的平均薪资情况，并且按照部门号从低到高排列。*  SELECT deptno, AVG(sal) FROM emp GROUP BY deptno ORDER BY deptno;  *--需求2：查询显示各个部门的不同工种的平均薪资情况，并且按照部门号从低到高排列*  SELECT deptno,job, AVG(sal) FROM emp GROUP BY deptno,job ORDER BY deptno; |

## 过滤分组（having）

### 过滤分组的概念



### 过滤分组的语法



### Where和having的选择

1. 语法上的不同选择

1）是否能使用组函数的区别：

不能在 WHERE 子句中使用组函数（注意），即where子句不能完全代替having子句。

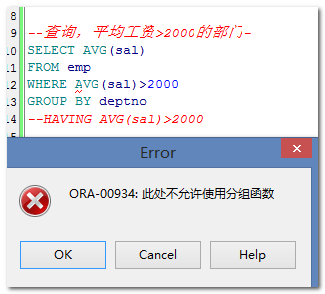
可以在 HAVING 子句中使用组函数。（having可以使用任何的条件写法）

【示例】

需求：查询平均工资大于2000的部门信息，要求显示部门号和平均工资

|  |
| --- |
| *--需求：查询平均工资大于2000的部门信息，要求显示部门号和平均工资*  SELECT deptno,AVG(sal) FROM emp WHERE AVG(sal)>2000 GROUP BY deptno;*--无法执行*  SELECT deptno,AVG(sal) FROM emp GROUP BY deptno HAVING AVG(sal)>2000; |

非法使用组函数的错误提示：



2）HAVING子句不能离开GROUP BY子句单独使用，HAVING子句无法完全代替WHERE子句。

【示例】

需求：查询所属部门号大于等于20的员工信息。（无法使用having子句）

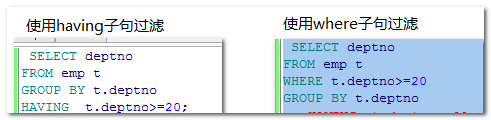
|  |
| --- |
| SELECT \* FROM emp WHERE deptno >=20;*--正确*  SELECT \* FROM emp HAVING deptno >=20;*--错误* |

非法使用having的错误提示：



1. 性能优化方面的选择—sql语句优化

思考：下面两组语句哪个效率更高。



分析：

使用having子句过滤，是先分组，再过滤，注意：分组的时候是全表扫描的，效率较低。

使用where子句过滤，是先过滤再分组，注意：分组的时候仅需要扫描部分数据，效率较高。

【结论（如何选择）】：

从语法上看，两者选择简单归纳为，就是group by分组之后需要的条件中有组函数的，就必须得用having，其他都可以直接用where。

从性能上看，实际开发中，使用分组的时候尽量先加一个where的过滤条件。没有组函数的情况下，尽量选择where。

全天总结重点：

* + - * 1. sql语句优化的几点：选择列，别名的使用、count(1)、优先使用where过滤
        2. 字符串的连接||，转义字符
        3. 伪表dual的使用。
        4. 空值运算in和not in
        5. 单行函数：字符函数、转换函数（to\_char()，to\_date(),to\_number()）,条件表达式（decode、case ）
        6. where和having的使用选择