# 1.oracle环境准备

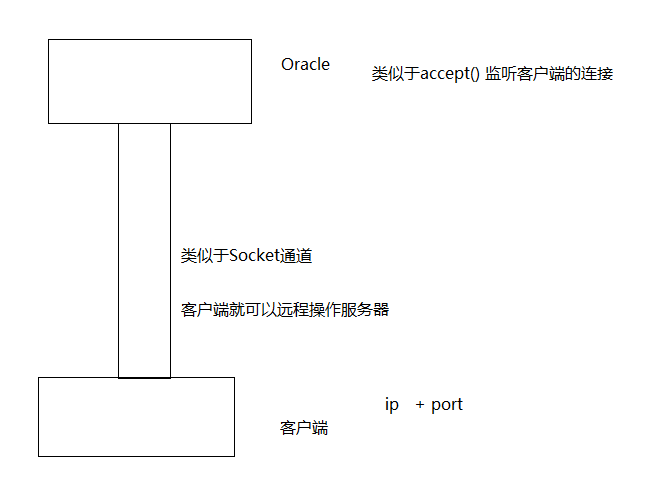
## 1.1 Oracle安装

包含Oracle的软件和Oracle的数据库ORCL

省略。。。

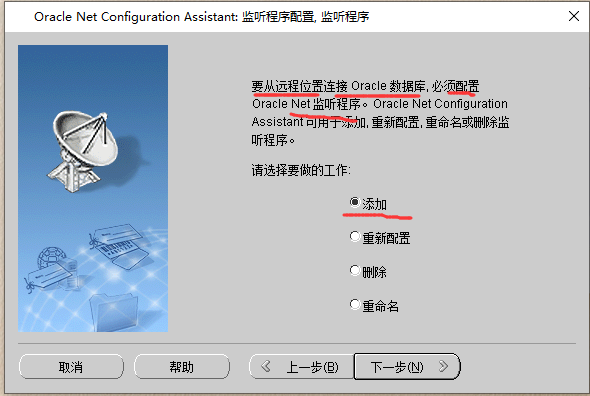
## 1.2 Oracle的监听

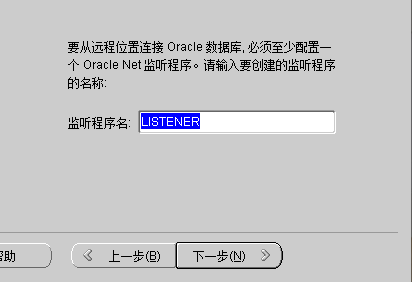
目的：客户端软件远程连接Oracle数据库

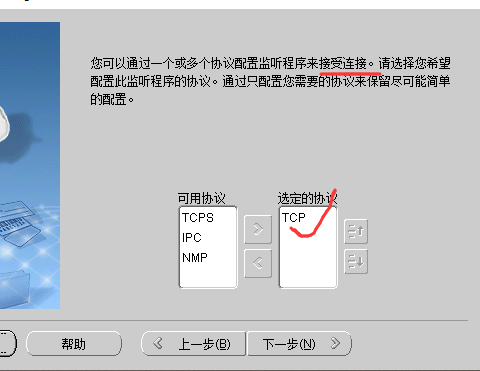


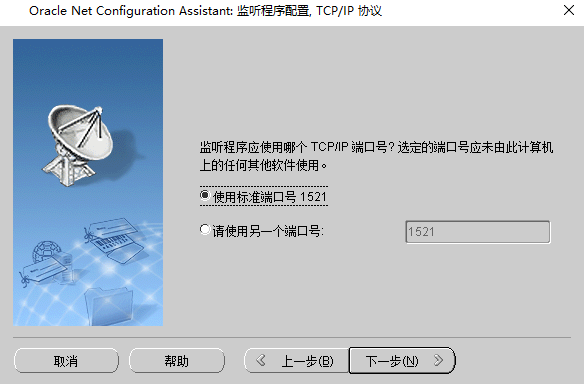




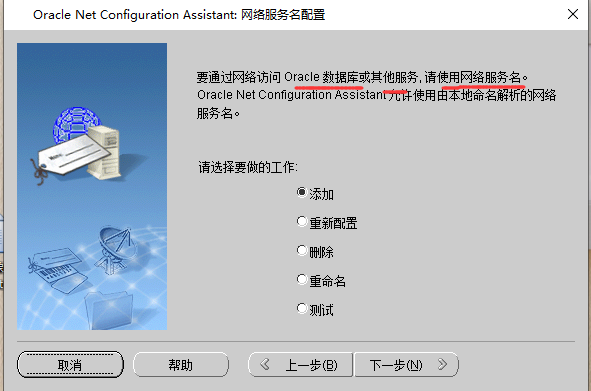




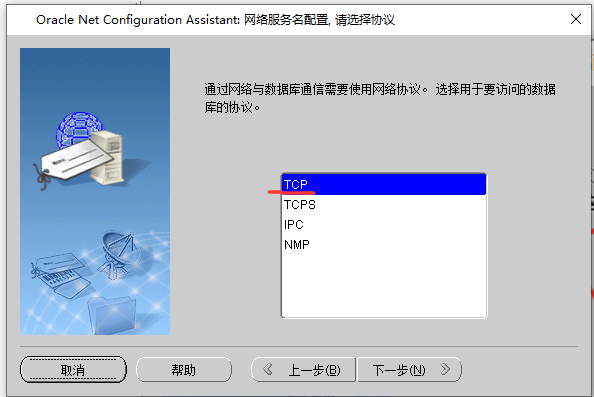


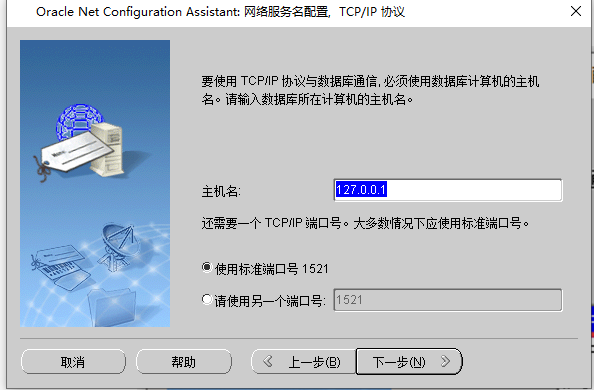


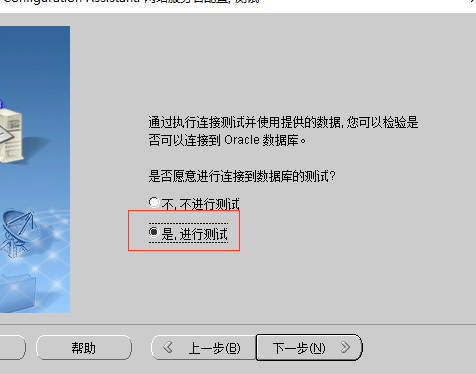






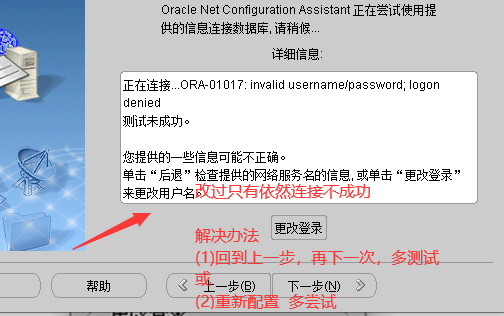






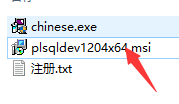


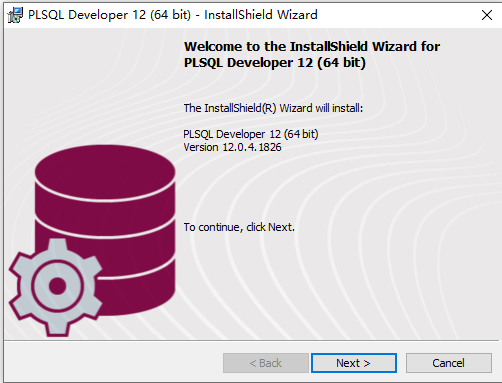


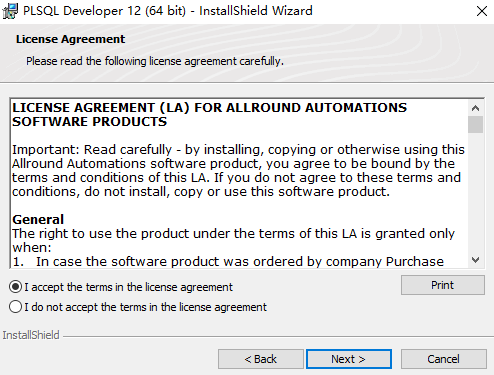


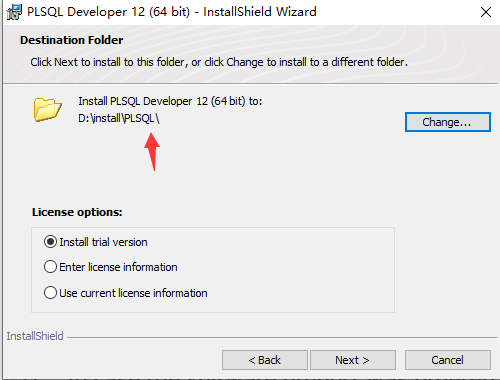
当使用第二种解决办法时：删除之前配置的监听和网络服务名。

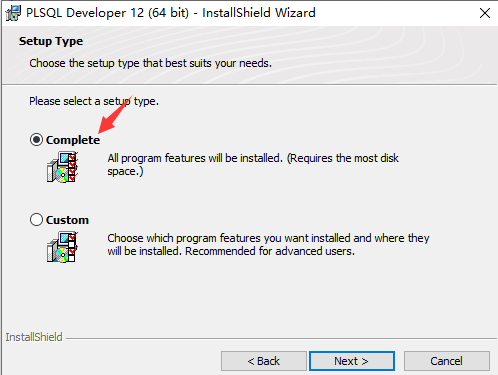
## 1.3 PLSQL Devleper

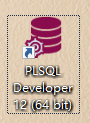


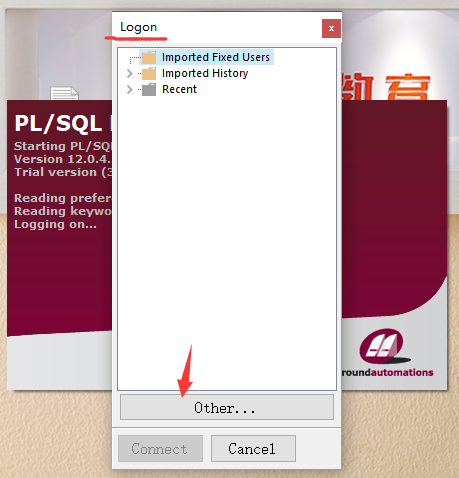


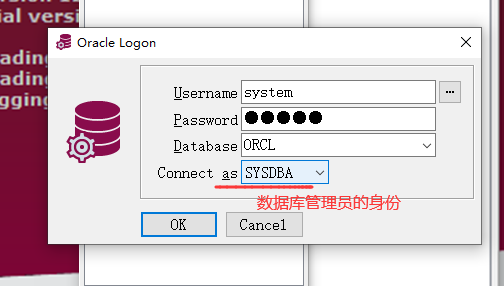


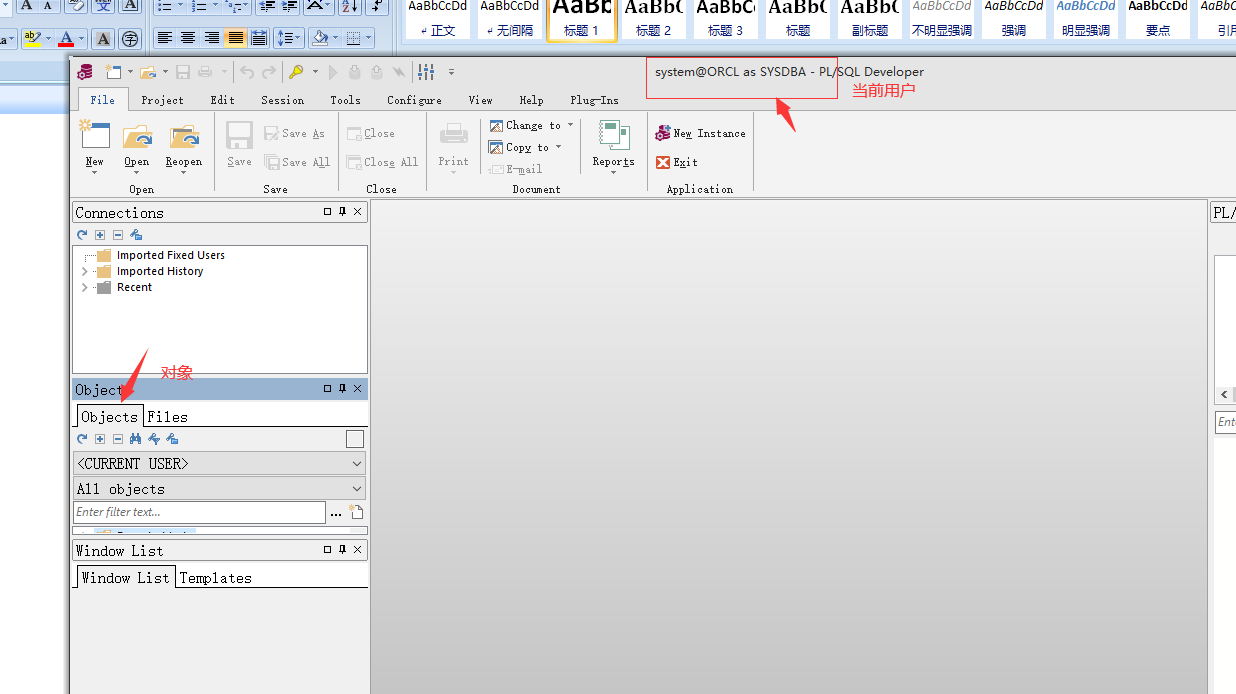


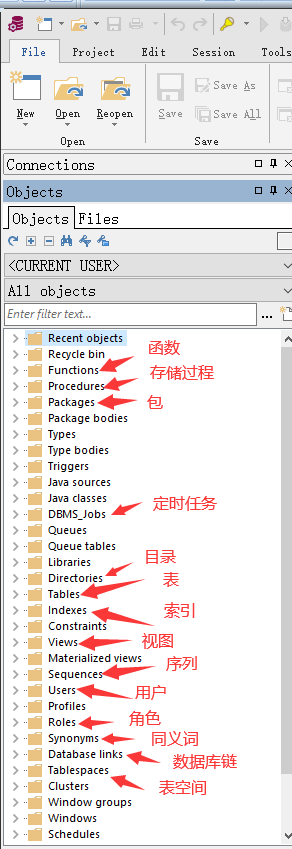




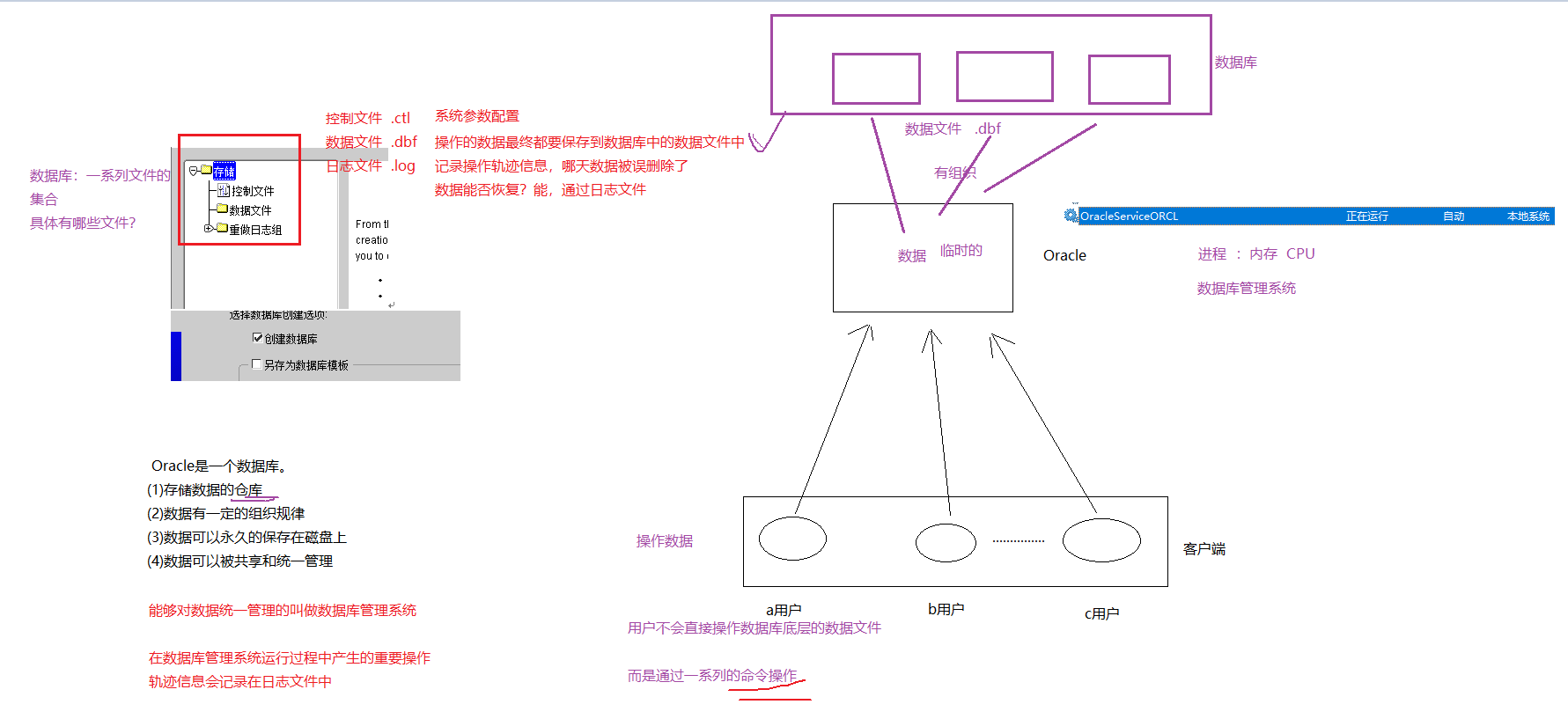




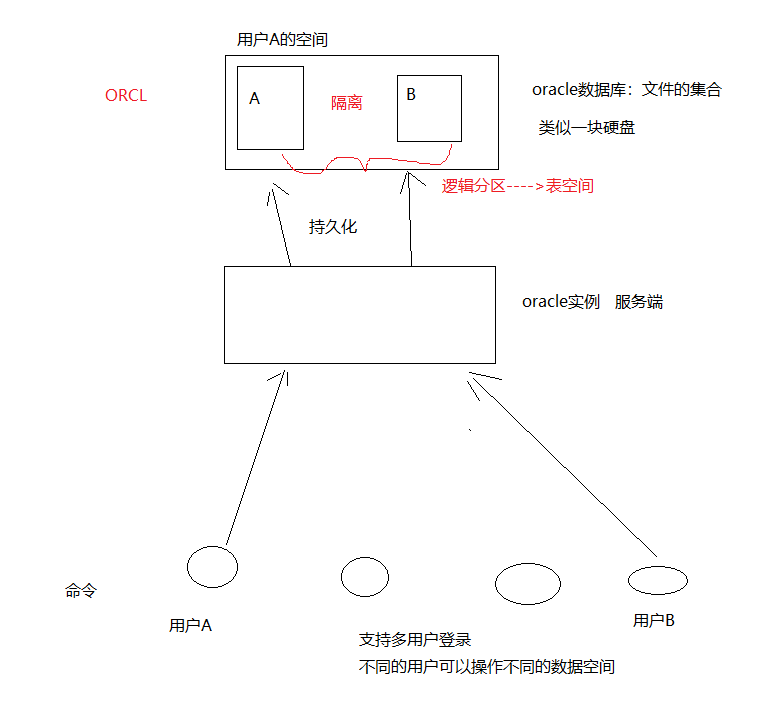




# 2.数据库的概念



# 3.用户、表空间、权限、角色



## 3.1 表空间

是对数据库的逻辑分区，起到的作用

(1)分类管理

(2)降低故障，一个表空间出现故障了，不影响其他的表空间

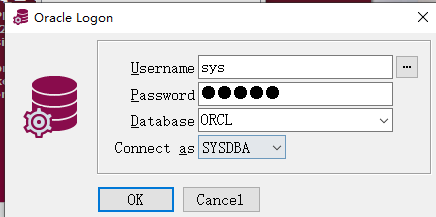
(3)动态扩容，如果某个表空间存储不够，按需扩容

如何创建表空间

需要高级用户，才有权限创建

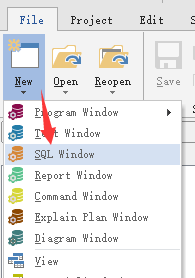
改为sys用户

(1)使用管理员用户sys登录到oracle上



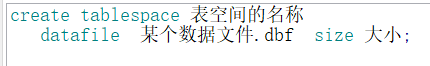
(2)创建表空间

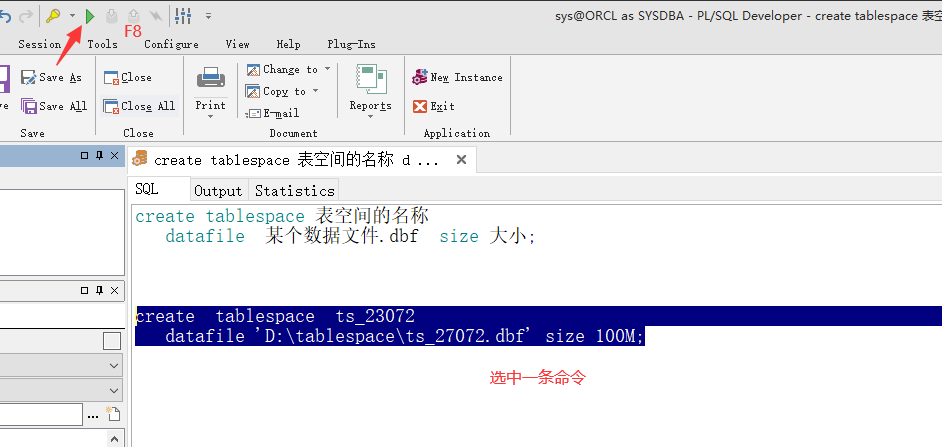
使用相关的命令



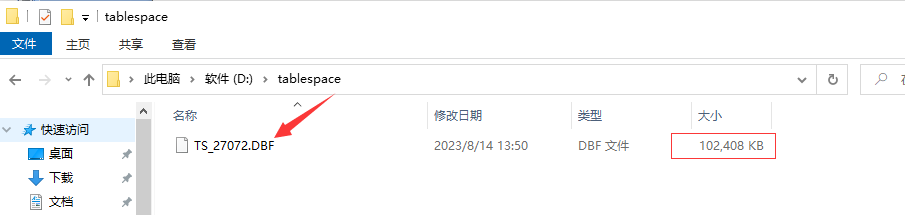
SQL Window :结构化查询语言的窗口

语法：





检查表空间

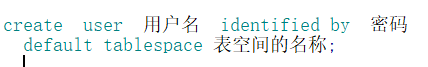


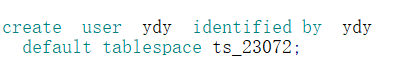
## 3.2 用户

oracle支持多用户

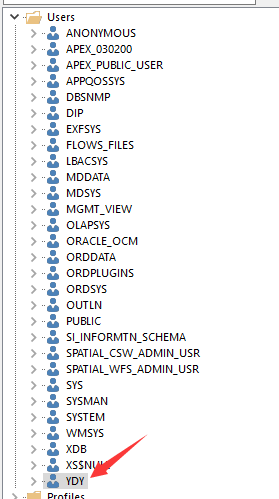
每个用户一般使用不同的表空间

创建用户的语法

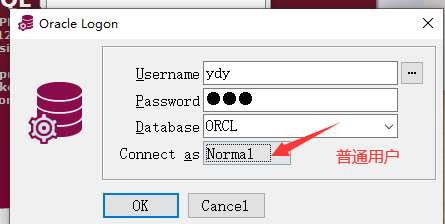


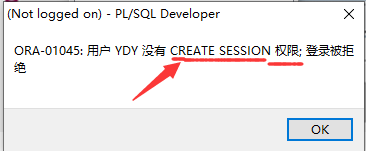


检查ydy用户



切换成普通用户登录





刚创建好的用户，连登录权限都没有，赋权限

## 3.3 权限

oracle中有很多个权限，高权限用户可以为低权限用户赋权限。

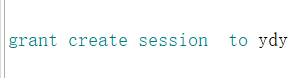
语法

grant 权限名 to 用户

grant 权限名1 , 权限名2, 权限名3……权限名N to 用户

(1)切换高级用户

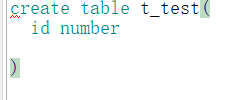
(2)赋权



(3)切换普通用户



(4)创建表



权限不足

为普通用户赋建表的权限

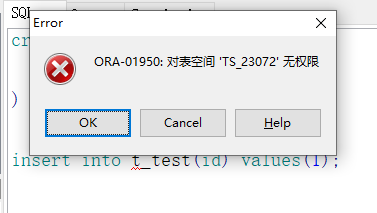
回到了高级用户，



切换普通用户

建表成功

给表中插入一条数据



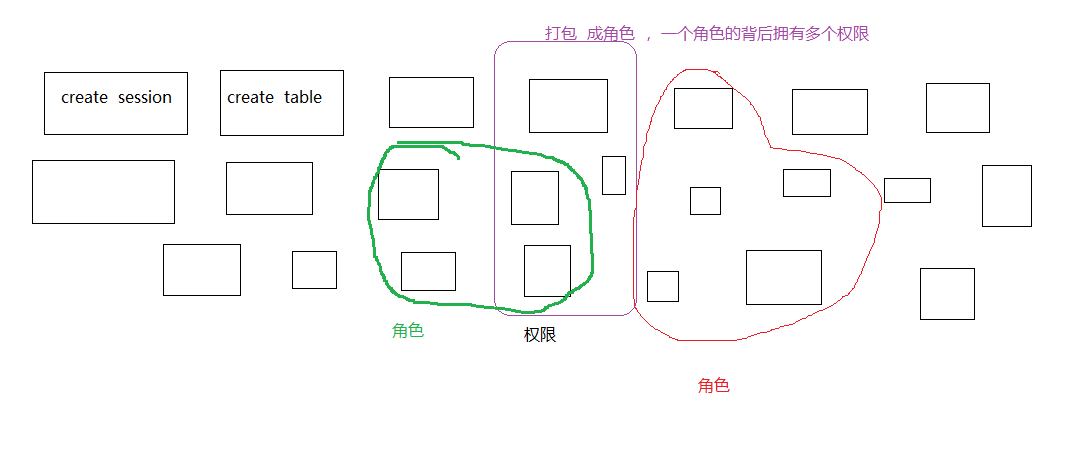
想象：普通用户在一个表空间中操作，处处受到权限的限制

频繁的使用高级用户grant 权限 ，不舒服。

简化赋权，使用角色代替。

## 3.4 角色

角色：就是权限的打包

赋角色

grant 角色1，角色2,……角色N to 用户

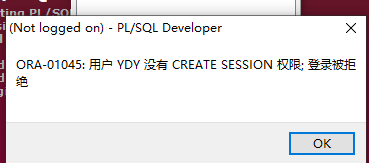
实验

(1)使用高级用户，回收权限

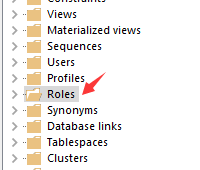
语法：revoke 权限1 from 用户

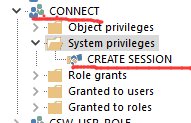
或者：revoke 权限1 ,权限2,….权限N from 用户

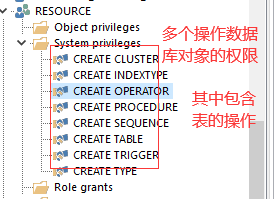


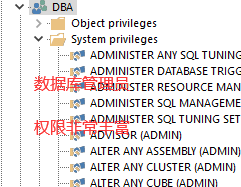


(2)使用角色



登录角色



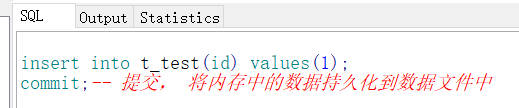


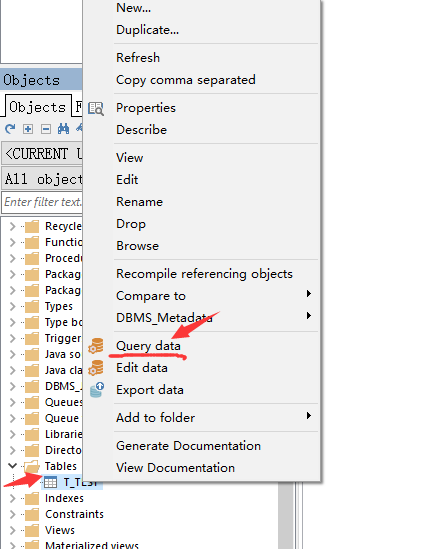
一般人是不会拥有dba权限的，除非专业的数据库管理员。



切换到普通用户

登录？插入数据？



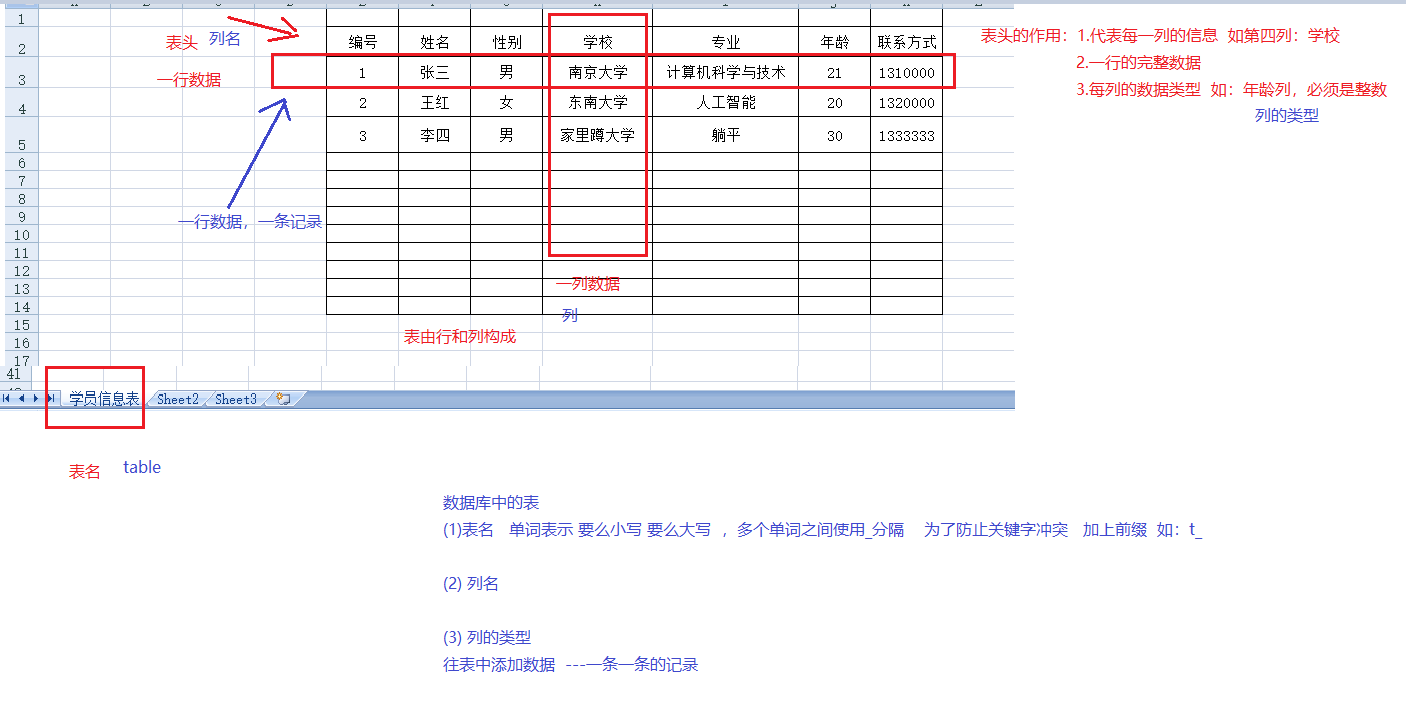


# 4.表

## 4.1 表的概念

是存储数据的载体。

现实生活中，经常创建excel表格



语法

create table 表名(

列1 类型1,

列2 类型2,

列3 类型3,

………….

列N 类型N

);

## 4.2 数据类型

oracle中有很多类型，基本和Java类型类似，但是真正使用到的类型很少

(1)数字类型 NUMBER/number（数据库中，大小写不分）

number:一个数字 很少用

number(10):是一个10位的整数

number(3,2):三位的数字，整数部分是1位，小数部分是2位

(2)字符类型

定长：长度固定char(位数) 学号、手机号码、身份证号码、社保卡号

不定长：长度不固定varchar(位数) 姓名、家庭住址、专业名称

面试题：char和varchar的区别？

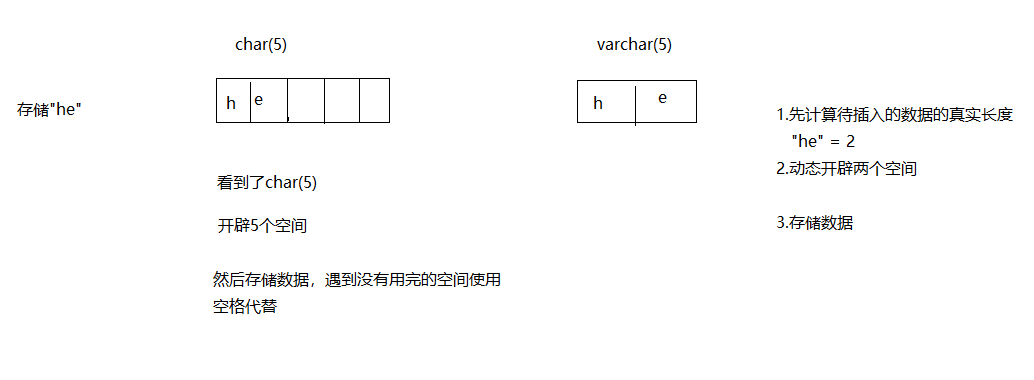
共性：都是字符串的表示方式

不同点：

a.char是定长，varchar是不定长

b.char有空间的浪费，效率高

varchar有时间上的消耗，节约空间，oracle高版本中对varchar的性能做了大大的提升，重新整出一个类型 varchar2

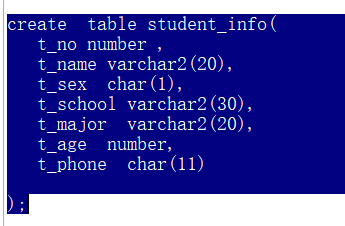


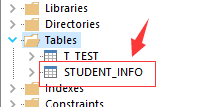
结论：

正确的用法，定长的就用char,不定长的使用varchar2

(3)日期类型 date

(4)布尔类型 boolean

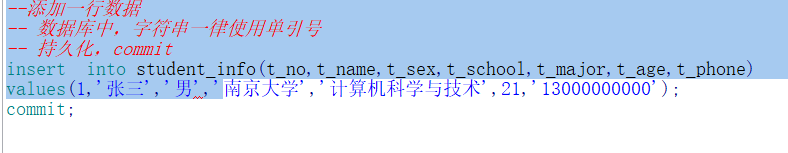


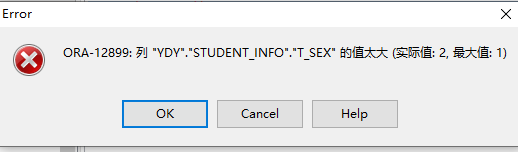


## 4.3 CRUD操作

CRUD：增删改查

插入数据：使用insert命令



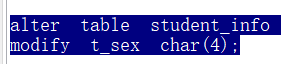


中文字符，占两个长度

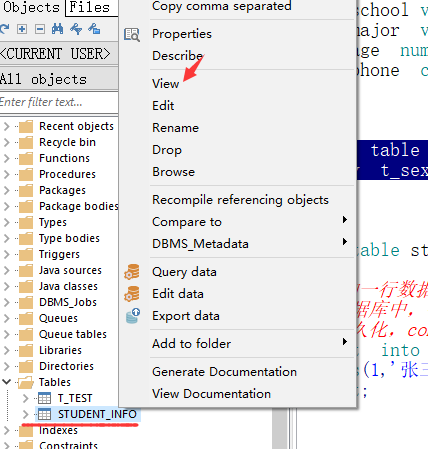
修改表的结构：

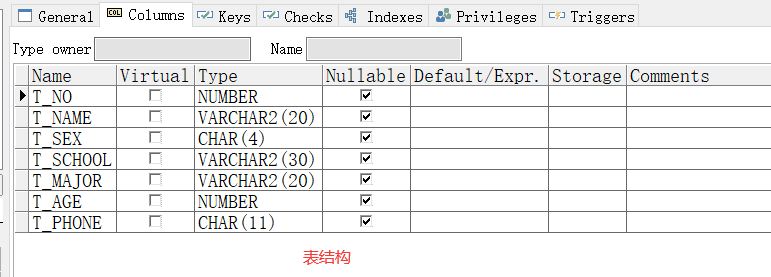
两种方法

a.使用alter命令修改字段

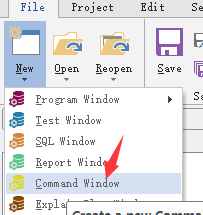


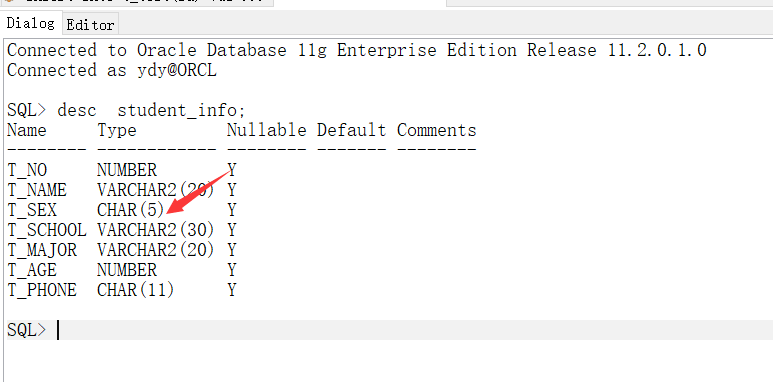
如何检查？





或者使用命令查询当前的表结构：





删除字段

alter table 表名

drop 列;

新增一个字段 ------------笔试考

alter table 表名

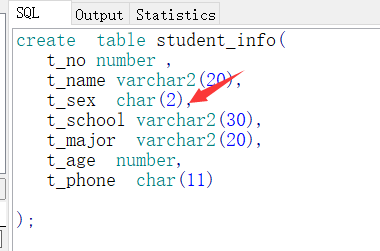
add 字段 类型;

b.删除表，重建表

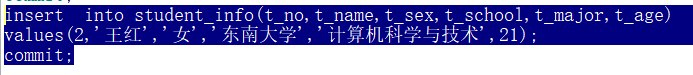
drop table 表名

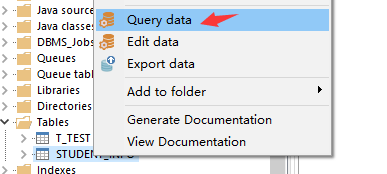


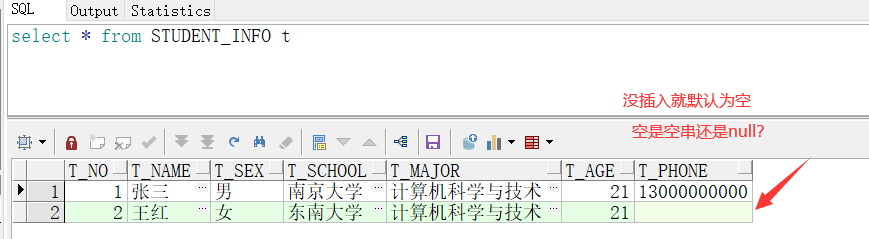
再重新create



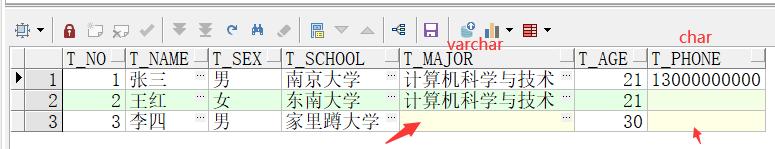
如果只插入部分数据





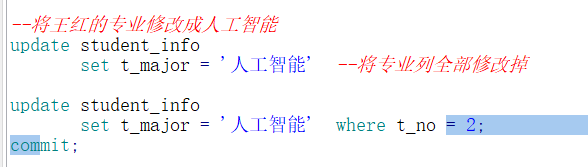






暂时不知道，等select语句说完之后。

修改数据：



同时修改多个列

update 表名

set 列1 = 值1,

列2= 值2,

….

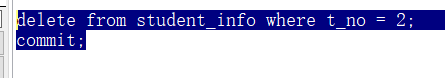
列n= 值n where 条件

删除数据：

办法：

(1)drop table 表 暴力，表都删除了，当然没有数据

(2)delete ,对表结构没有影响



删除所有

delete from 表

(3)truncate table 表 清空表数据 对表结构也没有影响。



面试题：

问，删除数据的几种方式？drop 、delete、truncate

问，delete和truncate的区别？

delete删除数据会写日志文件，可恢复，是安全的；性能不高；

truncate删除数据不会写日志文件，不可恢复的，是不安全的；性能高；

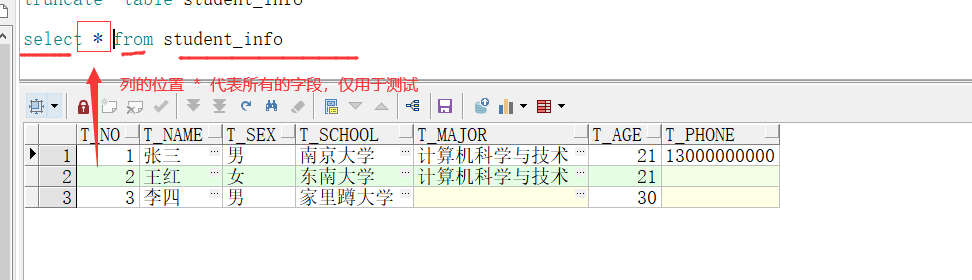
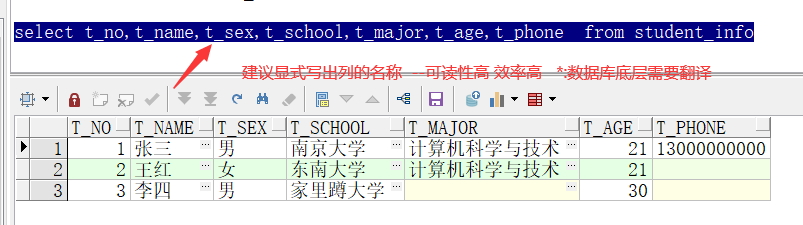
查询数据

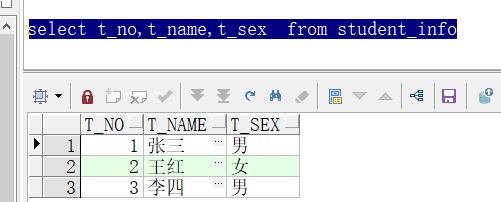
select 字段 from 表

select 字段1, 字段2 …… from 表

select \* from 表

select



# 5.范式

如何设计一张表 基本= 如何设计一个类

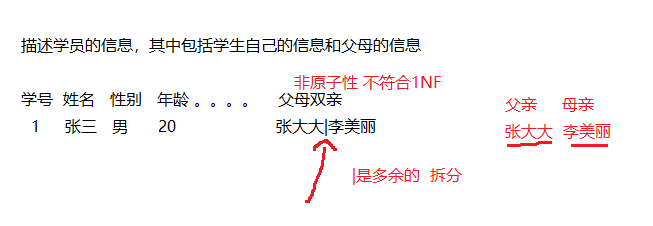
遵守设计范式

范式就是规范，目标：降低数据的冗余，不同的范式呈递次规范。

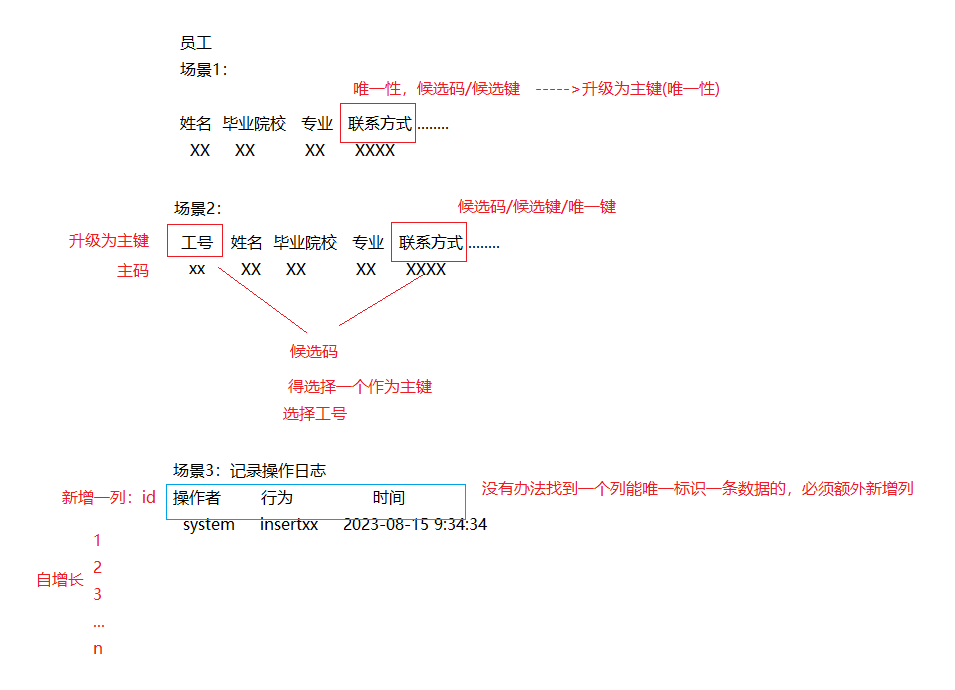
具体的范式有6种：1NF，2NF，3NF，BCNF(巴斯-科德范式)，4F，5F

实际生产过程一般要求数据库的表达到3NF即可。

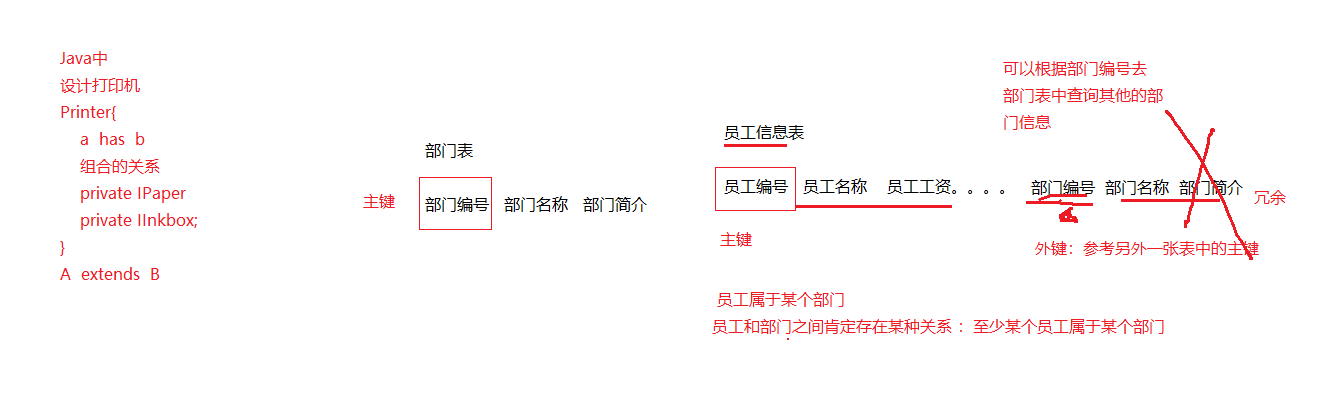
1NF：表中的每一项必须是原子性，不可拆分的，最低要求。



2NF：每个表都必须拥有主键



3NF：一个表中的外键必须是另外一张表的主键。



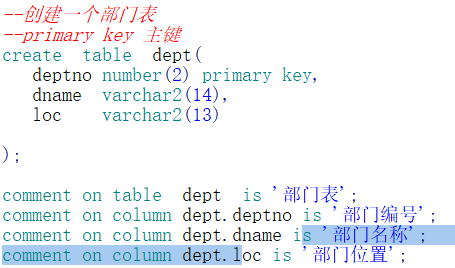
# 6.表的练习

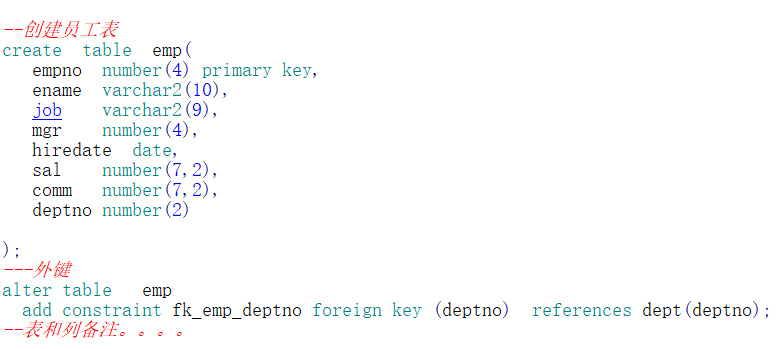
需求：

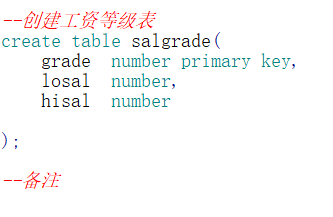












# 7.SQL语言

## 7.1 SQL分类

SQL ： structured query language 结构化查询语言。

结构化:固定模式，每个数据库都遵循SQL的规范。

查询：误导 只能查，实际上不仅仅可以查数据。

SQL的分类

(1)数据定义语言 DDL语言

create user

create tablespace

create table

alter

drop

(2)数据操纵语言 DML语言

CRUD

insert update delete truncate

查询，可以简单查询，也可以复杂查询。

(3)数据控制语言 DCL语言

grant

revoke

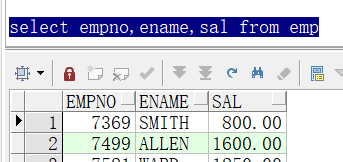
## 7.2 查询

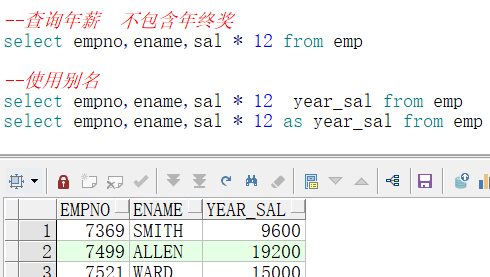
select语句

### 7.2.1 from子句

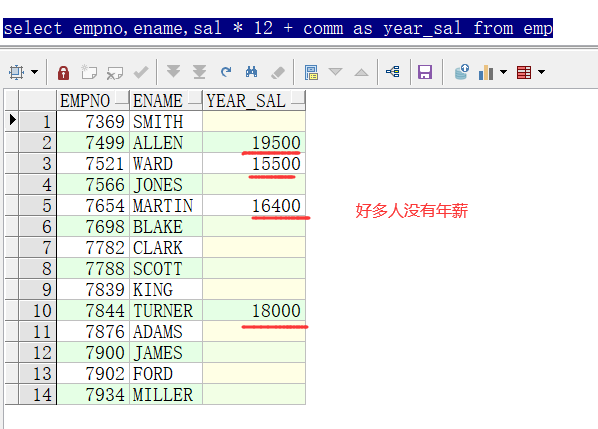
select 字段 from 表名 (数据的来源)

查询员工的年薪(不包含年终奖)

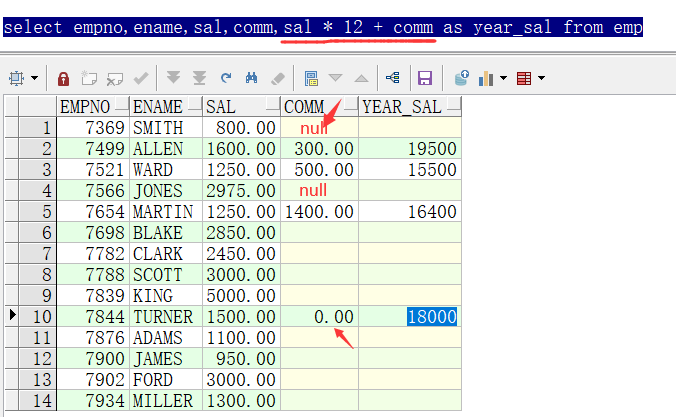




查询员工的年薪，包含年终奖



分析：null参与计算，结果为null

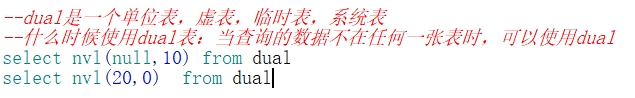


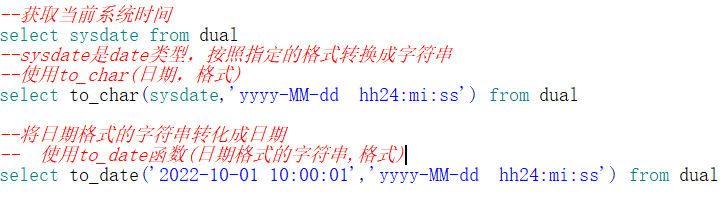
如果comm为null，则按照0计算，否则正常计算。

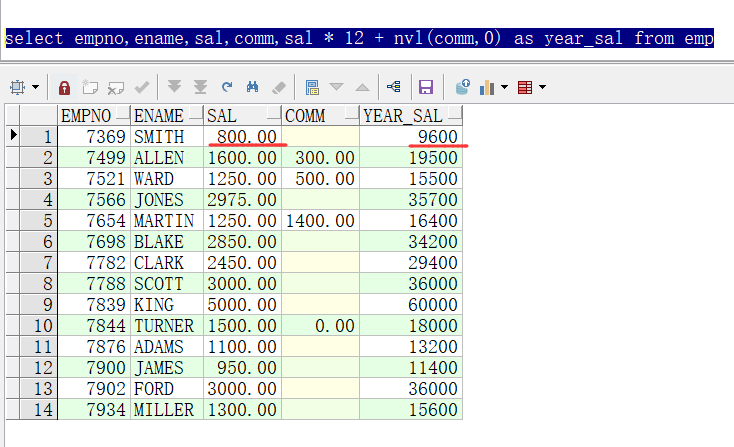
oracle中提供一个函数(相当于Java中的方法)

nvl(字段,值)：如果字段为null，则使用后面的值代替，如果不为null，使用字段本身

测试函数：







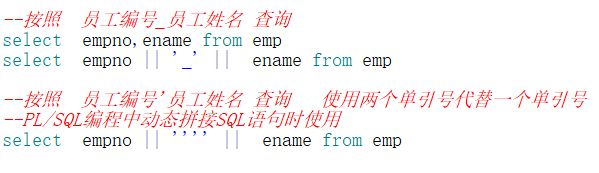
说明：关于别名问题

字段的别名

表的别名



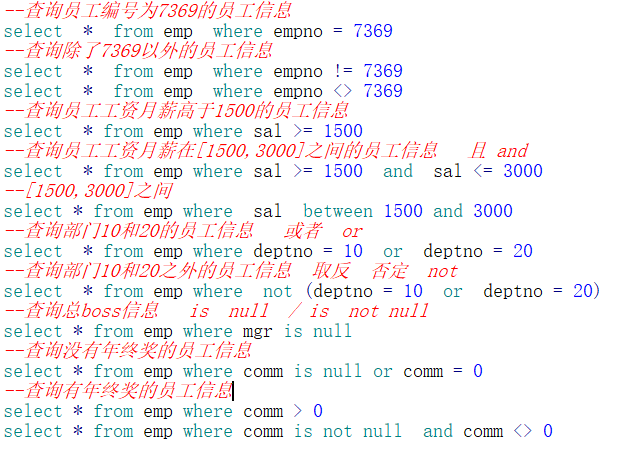
### 7.2.2 字符串拼接

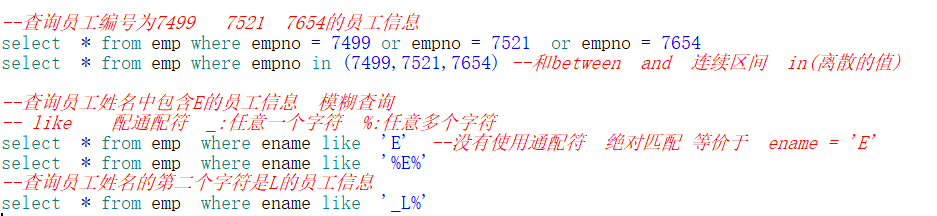


### 7.2.3 where子句

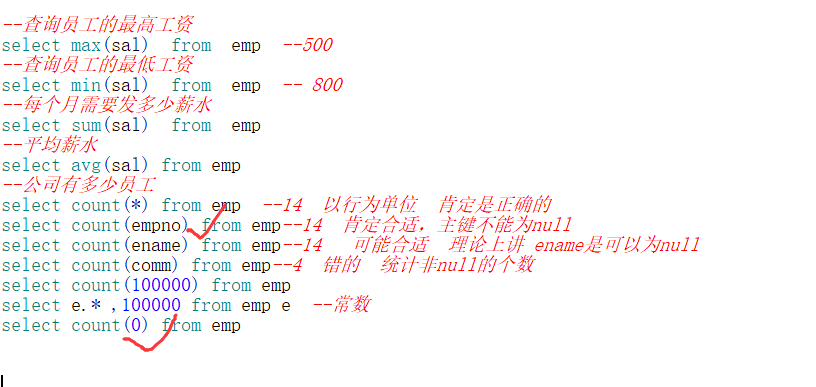
过滤数据

select 字段 from 表 where 过滤条件



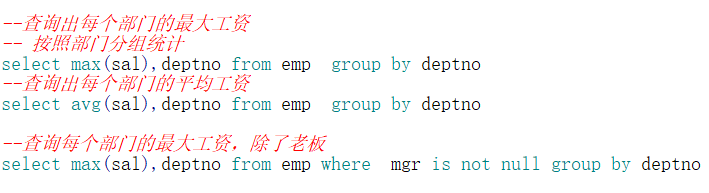


### 7.2.4 group by 子句



总结：

max、min、sum、avg、count都是统计函数，共同的特征：统计的结果只有一个数据



说明：分组中经常会出现的错误

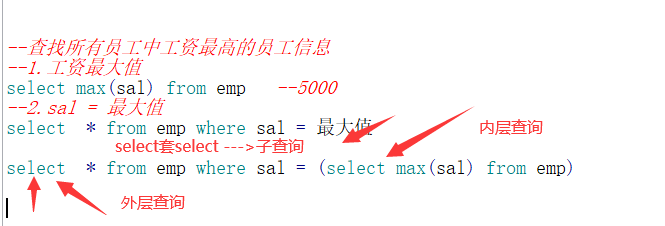


分析：group by:分组

max:统计 一个组中只有一个最大值，但是一个组中可能有多个empno

### 7.2.5 子查询

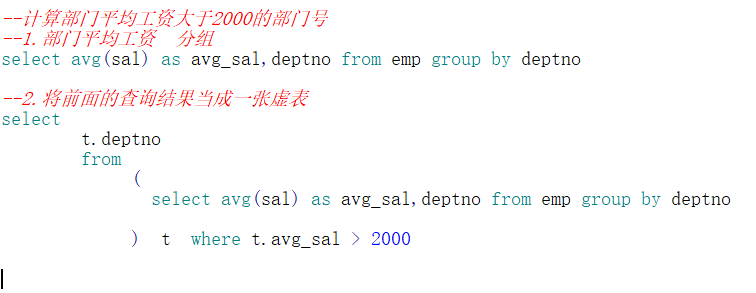
查找所有员工中工资最高的员工信息



执行时，先执行内层查询，再执行外层

什么时候会使用子查询？问题拆分时，通常伴随复杂问题。

计算部门平均工资大于2000的部门号



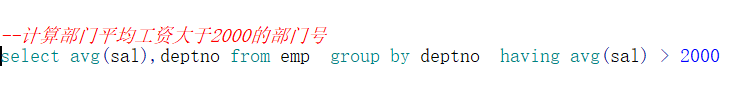
### 7.2.6 having子句

分析：也是一个子查询，

先内层查询：分组

外层查询：对分组后的数据再过滤

数据库中，针对先分组，再对分组后的数据过滤，使用having表示。



having是专门为“针对先分组，再对分组后的数据过滤”量身定做的，简化查询语句。

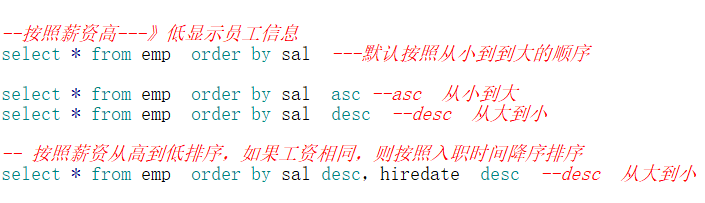
where和having的区别？

都能过滤数据

where：针对的是原始数据的过滤

having是分组后的数据过滤，having更加的有局限性。

### 7.2.7 order by子句



### 7.2.8 完整的SELECT语句结构

select ….. from 表

where 条件

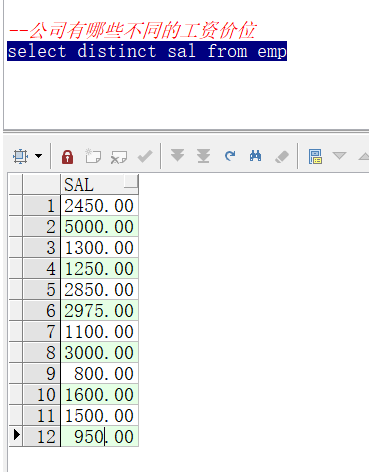
group by 分组

having 再过滤

order by

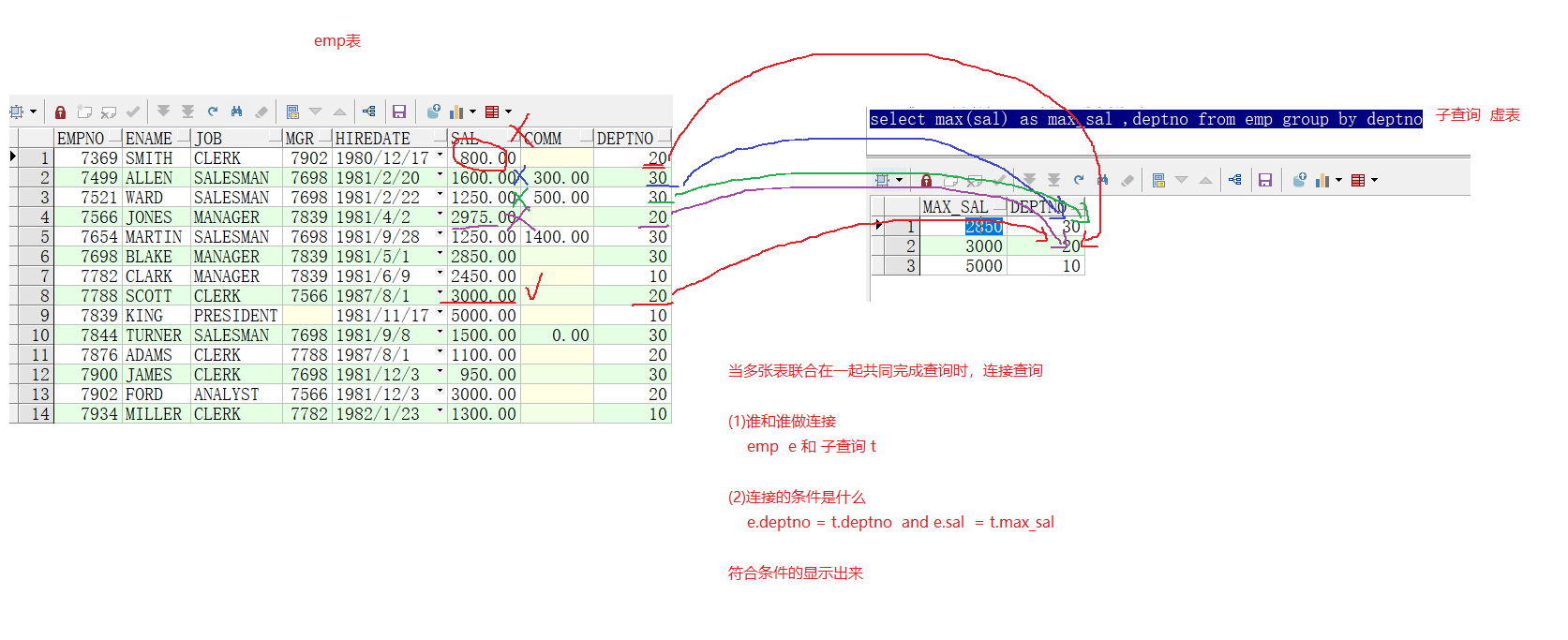
### 7.2.9 去重

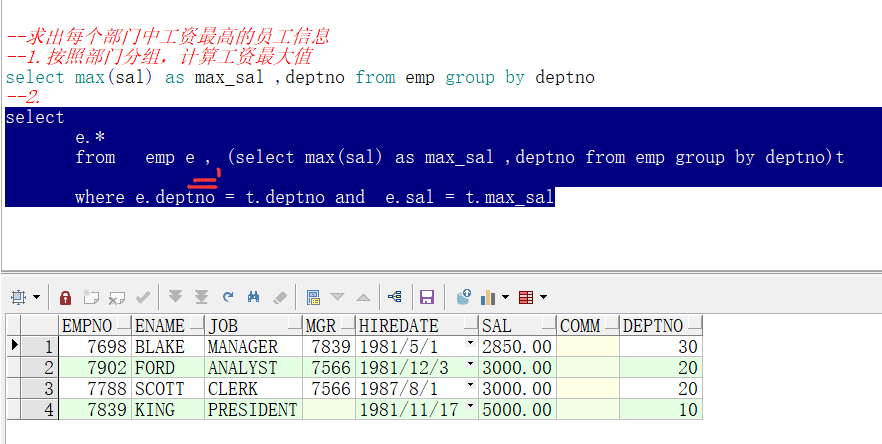
单词不会写。



### 7.2.10 表的连接

求出每个部门中工资最高的员工信息





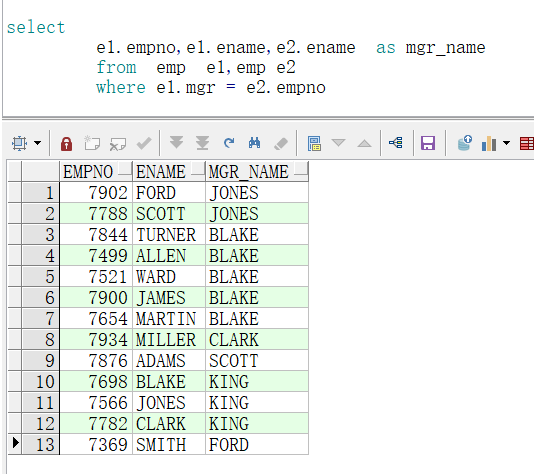
语法：

select 字段 from 表1,表2,….表N

where 连接条件

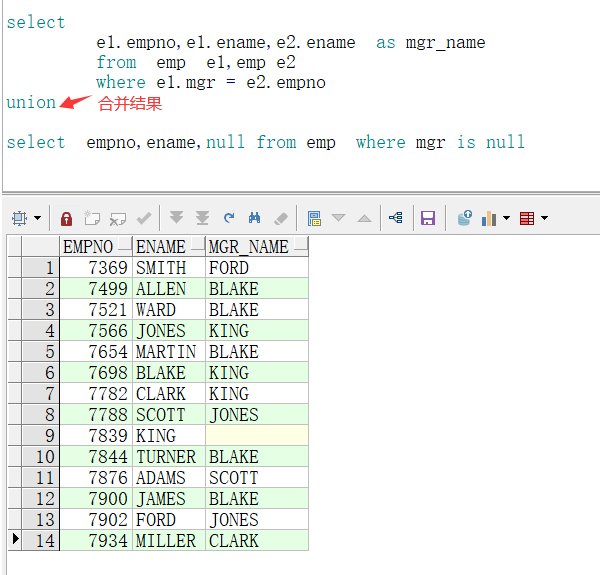
需求：查询员工的编号、姓名、领导的名称





分析：少了一条数据，KING，没有一个员工编号 = null

补上KING数据



说明：当查询一张表时，如果查询条件牵涉到本表之间的关联，多多的想到表的自关联。

技巧：使用别名构造多张表。

思考：

语法

select 字段 from 表1,表2,….表N

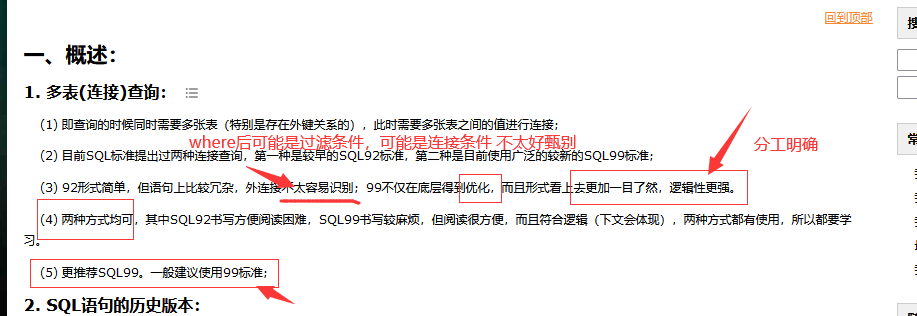
where 连接条件

有什么不足？

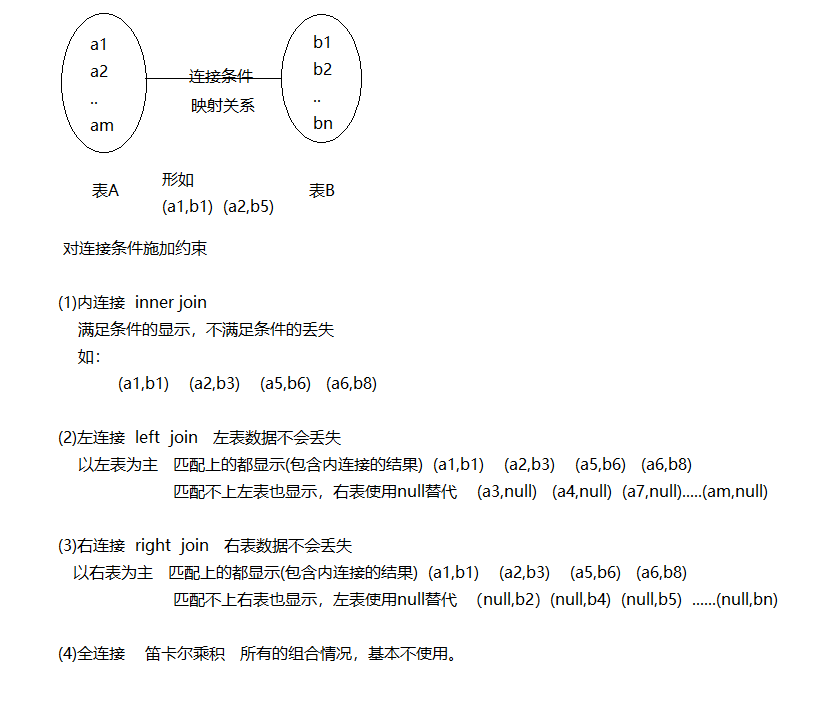
(1)会丢失数据，采用union补

(2)where的用法是过滤数据，而在这里作为连接条件，造成了where用法很模糊

于是出现了新的连接查询：



SQL99标准中的连接



面试：数据库有几种连接查询？

SQL92标准： select …..from a,b,c where ….

SQL99标准： join查询 分为 inner left right…..

可能追问？92和99有什么不同？

也可能会问 inner join和left join的不同？

也可能会问，什么时候会使用left join？内连接丢失数据的时候。

语法：

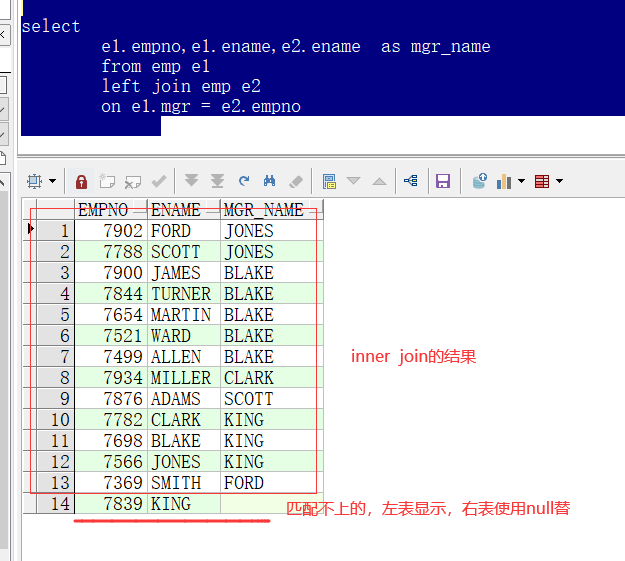
select 字段

from 表A

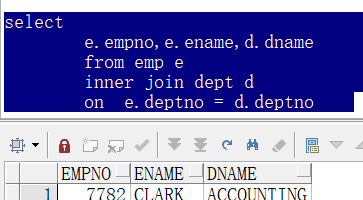
inner join/left join/right join 表B

on 连接条件

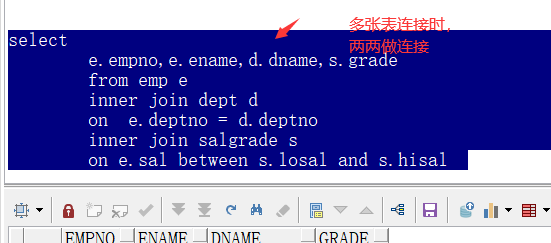
where 过滤条件



需求：查询员工的编号、名称、所属的部门名称。



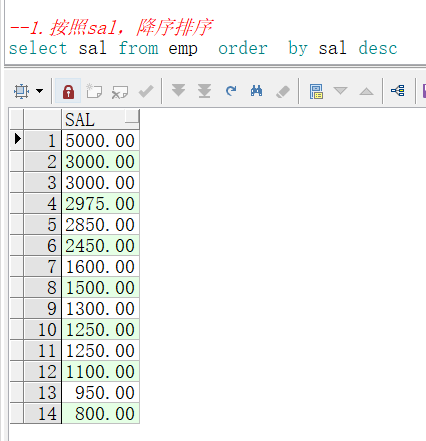
需求：查询员工的编号、名称、所属的部门名称，以及工资的等级。



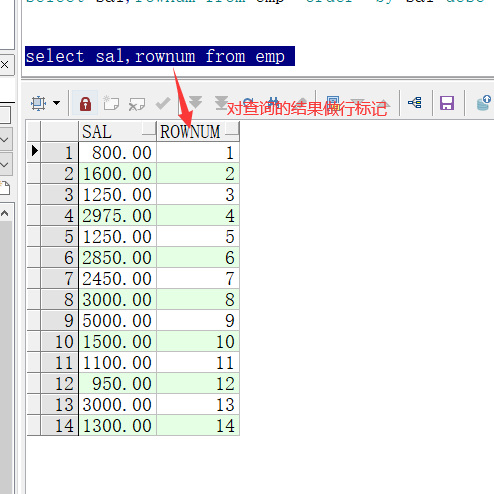
## 7.3 SQL笔试题

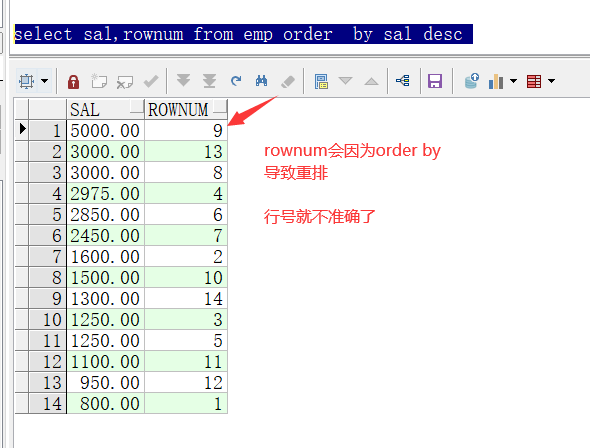
笔试题一、不使用max函数，求员工的最高工资，请写出通用的SQL语句。

方法一、

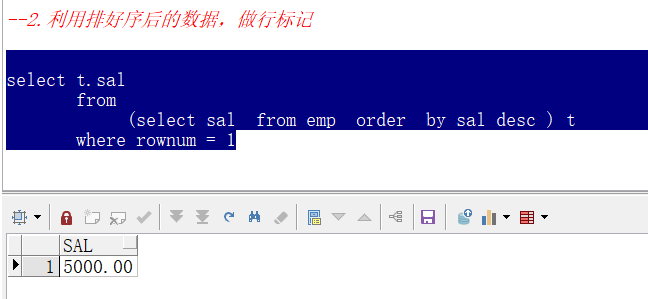


取得第一条数据。获取行号 利用oracle中的关键字 rownum



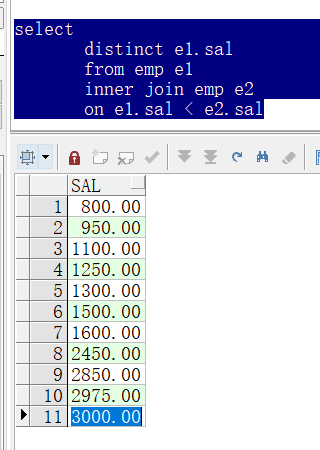


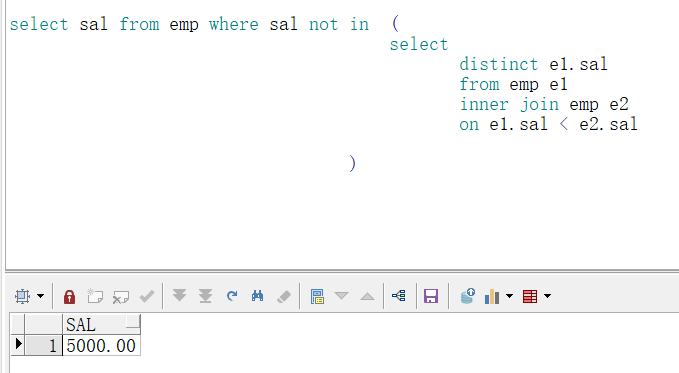
避免rownum参与排序：先排序，把排序后的结果采用rownum标记



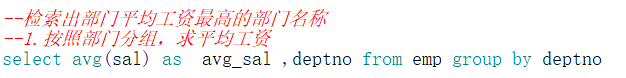
but: rownum只有oracle有，mysql就没有，方法一不通用

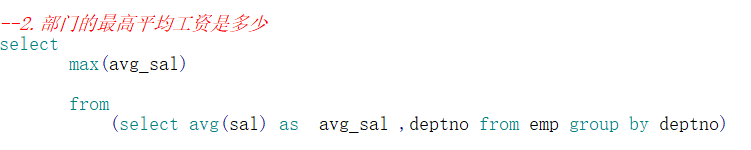
方法二、利用的是的表的自关联

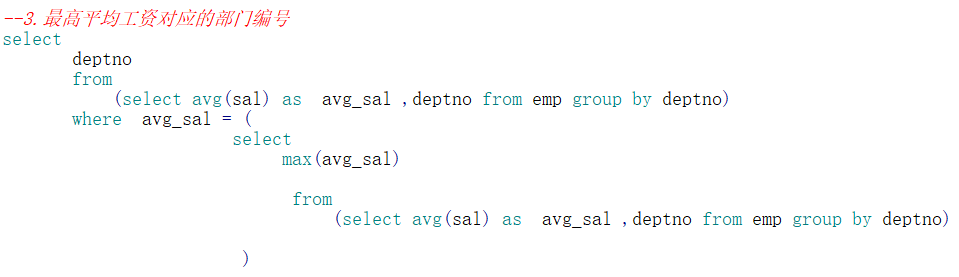


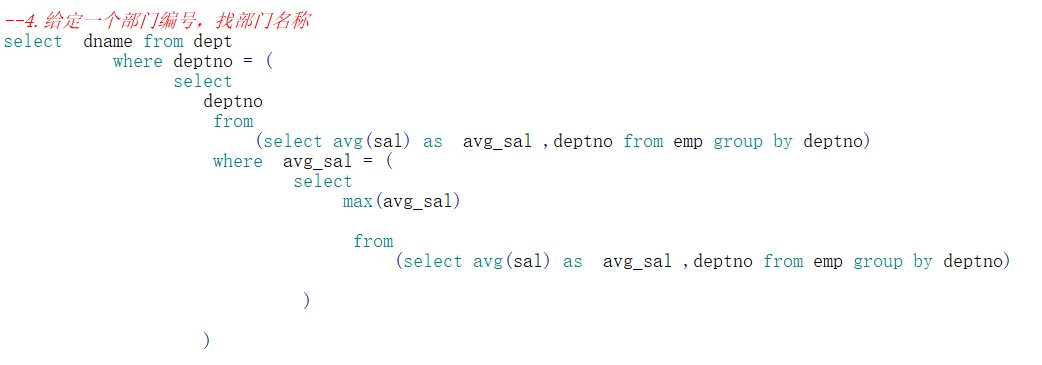


笔试题二、检索出部门平均工资最高的部门名称

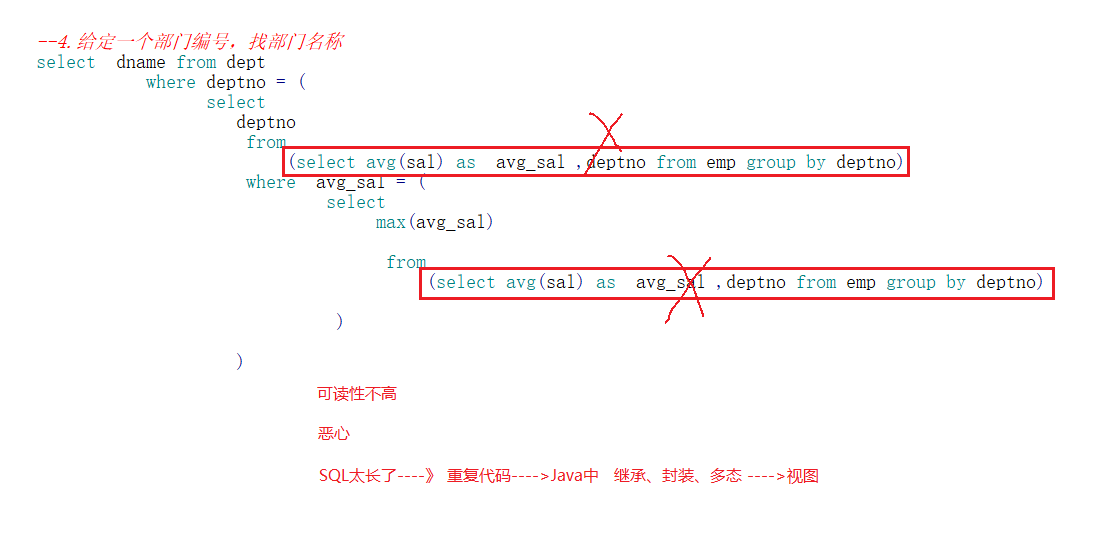








问题得解.



# 8.视图

视图：其实就是一个子查询，在oracle中，以View对象存在的。

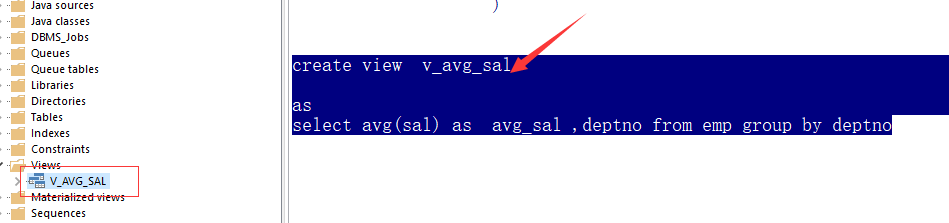
如何创建一个视图

(1)得拥有创建视图得权限

使用sys用户赋创建视图的权限

grant create view to ydy

(2)创建视图



语法

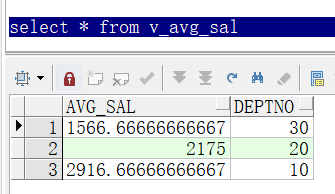
create view 视图名称

as

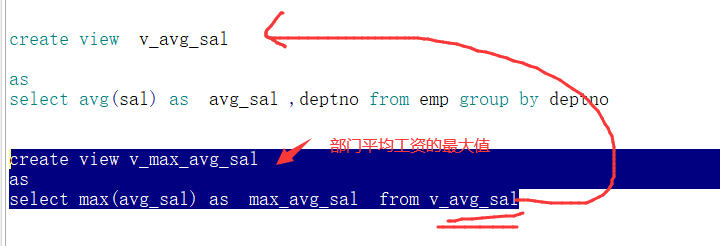
子查询

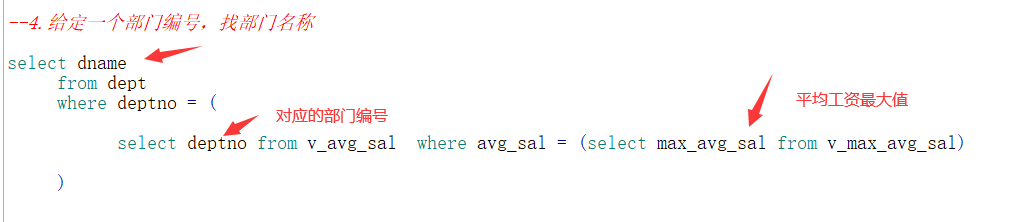
(3)使用视图

和使用表一模一样(子查询)



优化上述的SQL语句

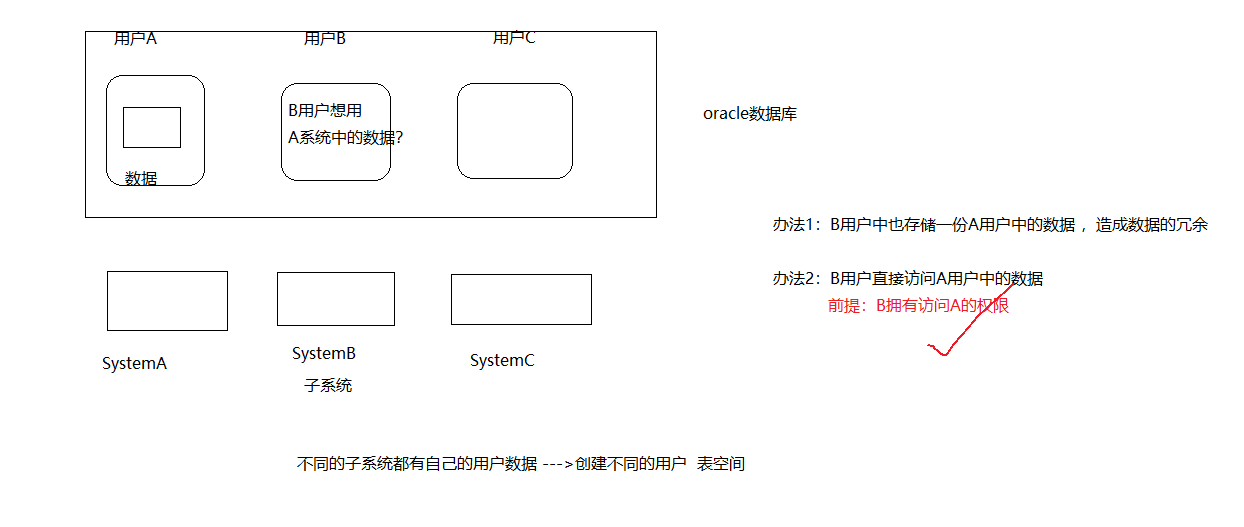




视图的作用：

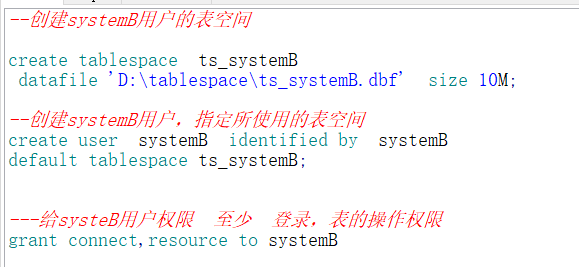
(1)简化查询。

(2)保护数据



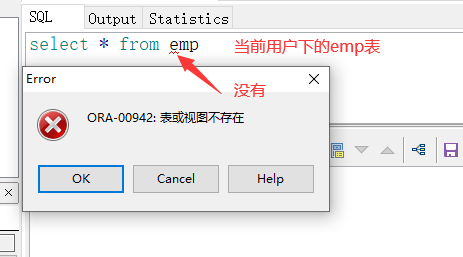
目前只有一个用户ydy

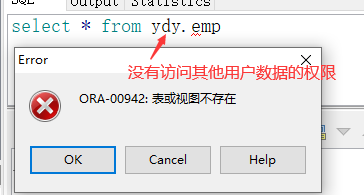
模拟新用户，systemB



切换systemB用户登录

访问ydy用户的emp数据，直接访问

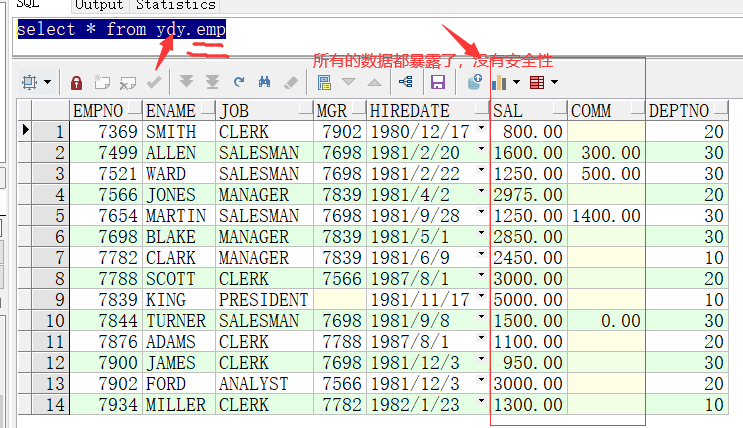




切换到ydy用户下，给systemB用户赋访问emp的权限



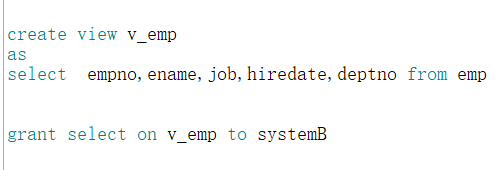
使用systemB访问数据



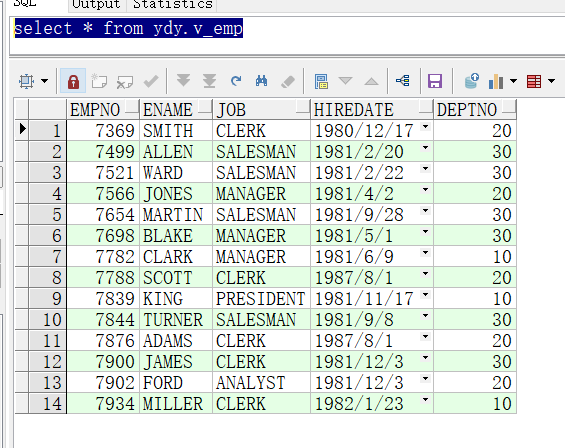
切换到ydy用户下，回收对emp查询的权限，改成视图访问



将数据封装到视图中

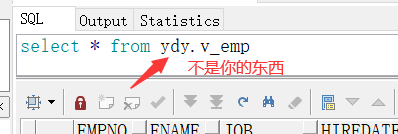


切换到systemB用户中

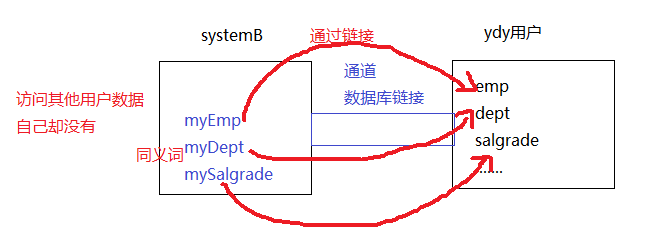


# 9.数据库链接、同义词

在不同的用户之间访问数据时，如果需要对数据保护，则一般使用视图的方式，开放访问权限，访问者需要在访问的时候携带用户名

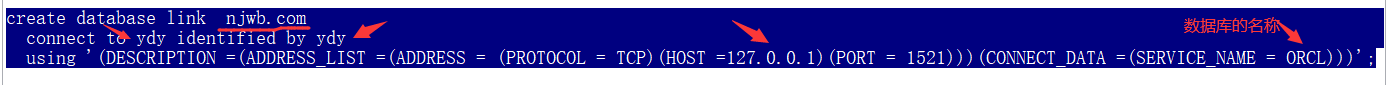


如果对数据安全性没有要求(内部系统，多个子系统，多个用户，不存在数据安全性问题)，希望直接访问(不想带用户名，感觉就是自己的)，得使用数据库链接和同义词。



创建链接







创建同义词







systemB下的同义词SN\_EMP和ydy.emp是一样的，称为同义词。

