**Oracle 笔记 电子档**

**参考：**[**Oracle 下的单词**](英文的释义及在命令中的含义/Oracle下的单词.docx)

**第一部分：概念**

1. 如何选择数据库
   1. 负载
   2. 成本
   3. 安全性
2. Oracle基本概念
   1. 关系数据模型：是关于数据库系统所采取的数据方式
   2. 字段：即表头
   3. 实体：所谓【实体】就是E-R模型
   4. 超键(super key)：在关系中能唯一标识元组的属性集称为关系模式的超键
   5. 候选键(candidate key)：不含有多余属性的超键称为候选键
   6. 主键(primary key)：用户选作元组标识的一个候选键程序主键
3. 范式理论（点击：[前三范式的讲解](http://www.cnblogs.com/GISerYang/archive/2012/05/09/2491996.html)）
   1. 第一范式 主要特点：实体的属性不能再分
   2. 第二范式 每个非全属性完全函数依赖于任一候选键
   3. 第三范式 所有的非主属性对任何候选键都不存在传递性

四、体系结构

点击：[数据库体系结构详解](http://blog.chinaunix.net/uid-28255964-id-3551144.html)

以及翻阅笔记本上的笔记

五、RAC 技术 ：两个实例连接一个数据库 （详见笔记本）

六、用户名不区分大小写，密码区分大小写

**第二部分：命令**

用户：sys（可创建数据库）

数据库的管理员，都可以管理数据库

system （不可创建数据库）

登陆时 sys后面必须加 as sysdba sys as sysdba

system后面可加as sysoper也可不加 sys as sysoper

更改密码：alter user 用户名 identified by 密码

注意：必须是管理员用户或具有更改密码权限的用户才可操作

更改自己本身的密码：password

提交：commit

本机登陆：conn 用户名/密码

如果是sys管理员登陆必须在后面加上 as sysdba

system可加可不加 as sysoper

解锁：alter user 用户名 account unlock；

查看当前用户：show user；

查看emp表：

在（Scott）下查看自己用户下的emp表：select \* from emp;

在sys或system下查看Scott下的emp表：select \* from scott.emp;

查看所有用户的名字：select username from all\_users;

断开当前用户连接：disc 用户名；

设置显示行的宽度，默认是80个字符：linesize

show linesize 显示当前的宽度

set linesize 设置宽度

每页显示的行数目：pagesize

建立txt 记录步骤：

spool c:/笔记.txt （开始）

spool off （结束）

清屏：host cls；

查询自己的emp表 ：select \* from emp；

查询scott（其他用户）的emp表：select \* from scott.emp；

set timing on/off （on表示开启Oracle中语句执行时间的计算）

“/”执行上一条语句

“--”单行注释

/\*\*/多行注释

创建用户： 一般是具有dba（数据库管理员）的权限才能使用

1、管理员登陆

2、创建

create user 用户名 identified by 密码；

3、给用户授权

grant 权限名（角色名） to 用户名；

4、回收权限

revoke 权限名（角色名） from 用户名；

5、删除用户

drop user 用户 （可恢复）

查询用户具有的权限

select \* from dba\_role\_privs; 授予用户和其他角色的角色  
select \* from dba\_sys\_privs;  授予用户和其他角色的系统权限  
select \* from dba\_tab\_privs;  数据库中对象的所有授权

给用户增删改查的权限

grant select on scott.emp to 用户名

select on scott.emp——给 用户名 查看scott 的emp表的权限

insert 增 delete 删

update 改 select 查

all 所有关于emp的操作权限（增删改查…）

级联（系统权限）

1、管理员下建立几个用户

2、给第一个用户登陆权限，以及级联下放系统权限的权限

grant connect to 用户名 with admin option ；

\* connect 具有登陆权限的角色

\* with admin option 给用户级联下放系统权限的权限

3、依次下放权限

4、用过管理员回收任一用户的登陆权限

——结论：当回收任一用户的登录权限（系统权限）后，级联下放的用户不受影响

级联（对象权限）

1、管理员下建立几个用户

2、给第一个用户查看scott的emp的对象权限，以及级联下放对象权限的权限

grant select on scott.emp to 用户名 with grant option ；

\* select on scott.emp 可以查看scott的emp的对象权限

\* with grant option 给用户级联下放对象权限的权限

3、依次下放权限

4、用过管理员回收任一用户的对象权限（查看scott 的emp的权限）

——结论：当回收任一用户的对象权限后，会级联影响后面的用户

角色——权利即权限的集合 （预定义角色，自定义）

1、预定义角色——系统自己含有的角色

**①**、connect

包含权限：create session 登陆权限

②、resource

包含权限：create table

create type

create indextype

create procedure

create cluster

create sequence

create trigger

③、dba

包含权限：具有所有系统权限，但是给了其他用户dba后，不具有启用和关闭数据 库的权限，并且不具有删除sys system等系统用户的权限，被授予dba的用 户并不是完全具有管理员的权限

2、自定义角色——自己创建的角色，并对其授权

①、建立角色

语法：create role 角色名；

②、给角色授权

语法：grant 权限名1，权限名2，权限名3… to 角色名；

删除角色：drop role 角色名；

显示所有角色：select \* from dba\_roles；

显示角色具有的系统权限：

select privilege，admin\_option from role\_sys\_privs where role =’角色名 ’;

显示角色具有的对象权限：

通过查询数据字典视图 role\_tab\_privs 可以查看角色的对象权限或是列的权限

select privilege from role\_tab\_privs where role =’角色名 ’;

删除用户时，显示无法删除当前连接的用户的处理方法

1、查看当前连接状况：

select username,sid,serial# from v$session;

2、找到要删除的用户sid和SERIAL#对应的值执行以下语句：

alter system kill session'SID, SERIAL#';（kill session：断开连接的意思）

3、当显示System altered（用户已更改）后执行删除用户的命令

drop user 用户名；

Oracle的存储结构

角色、段、区、块

查询Oracle中所有的表空间

select \* from dba\_tablespaces；

创建表空间

1、创建简单的表空间

create tablespace test datafile ‘ d:\ test\_data.dbf ’size 20M；

2、指定数据文件的最大尺寸以及自动扩展

create tablespace test datafile ‘ d:\test\_data.dbf ’size 20M autoextend  
 On Next 5M Maxsize 500M；

3、查看表空间是否创建成功（查询所有的数据文件）

select file\_name，tablespace\_name from dba\_data\_files order by file\_name；

4、为一个表空间创建多个数据文件

create tablespace test datafile （创建一个名称叫test的表空间）

‘ d:\ test01\_data.dbf ’size 20M autoextend On Next 5M Maxsize 500M，

‘ d:\ test02\_data.dbf ’size 20M autoextend On Next 5M Maxsize 500M，

‘ d:\ test03\_data.dbf ’size 20M autoextend On Next 5M Maxsize 500M；

在已有的表空间中新增数据文件

alter tablespace test add datafile ‘ d:\test04\_data.dbf ’size 20M autoextend On Next 5M Maxsize 500M ;

修改表空间大小

(解释：为什么修改表空间大小而改的是数据文件的大小，因为表空间其实就是数据文件的集合，在创建的时候，只不过给了一个空间，用来在数据库中存放数据文件，而且设置大小设置的也是数据文件的大小，所以修改表空间的大小就是修改数据文件的大小)

alter database datafile ‘d:\test\_data.dbf’resize 100m [autoextend On Next 5M Maxsize 500M];

使用表空间 （unlimited 无限配额）

\* 创建用户时使用

create user 用户名 identified by 密码 default tablespace 表空间名 quota 分配 的空间大小 on 表空间名 ；

\*给已存在的用户指定表空间，也就是创建用户后使用

alter user username default tablespace 表空间名

\* 创建用户后给表空间分配配额

alter user 用户名 quota 分配的空间大小 on 表空间名；

修改默认的表空间

1、查看用户所使用的表空间

select user\_id,username,default\_tablespace from dba\_users;

2、修改默认的表空间

alter user scott（用户名） default tablespace test01（原表空间名） quota 30M on test02（修改为，新表空间名）;

查看用户所使用的表空间

select user\_id,username,default\_tablespace from dba\_users;

删除表空间

drop tablespace 表空间名 ；

——（表空间内没有建表和修改，也就是说空的）

drop tablespace 表空间名 including contents and datafiles ；

——（删除表空间，包括内容和文件数据）

修改表空间名称

alter tablespace 现在的名称 rename to 改为什么名称 ；

注：修改表空间名称，不影响数据文件

改变表空间的状态

a、表空间有两种状态

①、联机（online）状态

②、脱机（offline）状态

1、使表空间脱机

alter tablespace 表空间名 offline ；

2、使表空间联机

alter tablespace 表空间名 online ；

表空间的读写权限修改

1、只读权限（表空间只能读，不能写）

alter tablespace 表空间名 read only ；

2、可读可写（表空间既能读，也能写）

alter tablespace 表空间名 read write ；

移动数据文件

1、

2、

常用的数据类型 （size 字符长度）

1、字符类型

char（size）  
 ——固定字符长度类型 如：性别，身份证号码等。

特点：查询速度快，但不省空间

varchar2（size）  
 ——变长字符类型，只有Oracle有这个类型

特点：查询速度稍慢，但省空间

2、数字类型

如果是整数 ：number或number（size）

如果是小数 ：number（m，n）该数的有效位数为m位，小数后n位

number（m，-n）

例：number（6,2）表示有效位数为6位，小数后2位

3、日期类型

date 要根据日期的格式写

更改日期格式 alter session set nls\_date\_format =’yyyy-mm-dd’

4、大数据类型

clob 大字符型，最大4G 放很长的字符的，主要存放：小说，论文等

blob 二进制数据，最大4G 主要存放：视频，图片等

建表（步骤不是唯一的，这只是其中的一种）

1、创建表空间

create tablespace stu datafile ‘d:\stu\_data.dbf’size 30M autoextend on next 5M maxsize 500M;

2、创建用户并把表空间和配额分配给用户

create user gulang identified by 123 default tablespace stu quota unlimited on stu;

3、给用户授权

grant connect,resource to gulang;

4、建立学生信息表（建表）

create table stu(

s\_id number(5), ——逗号表示没有结束

s\_name varchar2(8),

s\_sex char(2),

s\_idcard char(18),

s\_age number,

birthday date ——结束时不能写逗号

); ——创建完成，结束用分号

查看表结构

desc 表名；

查看表内容

select \* from 表名；

不想破坏原表，把表复制一个来使用（这里以复制scott下的scott.emp为例）

create table emp1 as select \* form scott.emp；

删除表

drop table 表名； （可恢复）

truncate table 表名；（不可恢复）

修改表的结构

1、追加新列

alter table 表名 add 字段名 字段类型；

2、修改列的数据类型

alter table 表名 modify 字段名 字段类型；

——注意：当表内有内容时，谨慎使用

3、删除列

alter table 表名 drop column 字段名；

4、重命名列

alter table 表名 rename column 原字段名 to 新字段名；

查询当前用户所创建的所有表

select table\_name from user\_tables;

创建约束：

not null 不能为空

unique 唯一的

直接在字段类型后边加就好了

default 默认

sex char（2）default ‘ 男 ’ 无论什么数据库都用default‘ ’  
 如果该字段不写内容，则以默认的值为准，写了以写的内容为准

check 在一定范围内

age number check（age>=18 and age<=25）——年龄只能在18和25之间

sex char（2）check（sex=‘男’or sex=‘女’）——只能填写男或者女

sex char（2）default ‘ 男 ’check（sex=‘男’or sex=‘女’）

当default与check同时使用时，default放在check的前面，在SQL中可以放 前面也可以放后面，但是Oracle中必须要在前面

alter table tablename add [constraint name] check（字段名 in（‘ ’，‘ ’，‘ ’））;

在有很多的范围的时候，不用or，用in方便些，直接在in后面用逗号隔开所 有范围

主外键约束：（\*主键只能有一个，外键可以有多个\*）

主键约束（primary key）

特点：1、能够唯一识别某个实体的属性

2、内容不允许重复，也就是唯一的

3、不允许为空

语法：

创建表的时候直接在后面加 [constraint pkname] primary key

alter table 表名 add constraint pkname primary key(主键字段名)

外键约束（foreign key）

特点：1、必须是别的表中的主键

2、主键表中主键字段不存在的数据，外键表中不能输入

3、外键可以为空

语法：（在创建表的时候使用）

1、外键字段名 字段类型 [foreign key] references 基于哪张表的表名（将那张 表的哪个字段作为外键，一般为那张表的主键，名）on delete cascade

2、外键字段名 字段类型 [foreign key] references 基于哪张表的表名（将那张 表的哪个字段作为外键，一般为那张表的主键，名）on update cascade

例：id number [foreign key] references class(id) on delete | update cascade；

（创建之后使用）

alter table tablename add constraint fkname foreign key (外键字段名) references 主键表名（主键字段名）on delete cascade;

增加约束

alter table tablename modify 字段名 约束名;

alter table tablename add [constraint name] check（字段名 in（‘ ’，‘ ’，‘ ’））;

在有很多的范围的时候，不用or，用in方便些，直接在in后面用逗号隔开所 有范围

增加主键约束

alter table tablename add constraint pkname primary key（主键字段名）；

增加外键约束

alter table tablename add constraint fkname foreign key (外键字段名) references 主键表名（主键字段名）on delete cascade;

查看约束信息

select table\_name,constraint\_name,constraint\_type,status from user\_constraints where table\_name='大写的表名'

删除约束

alter table tablename drop constraint 约束名；（一般都是用这个删除 ）

alter table tablename drop primary key （主键名）[cascade]

**增、删、改、查 （字符和日期必须加单引号）**

添加数据

insert into tablename [(字段1、字段2…)] values （字段1的内容、字段2的内容…）

如果表中的所有字段都要插入内容，tablename后的字段可以不写，后面的内容按 原表的字段顺序写

也可以只插入几个内容，就把插入的字段写在tablename后面，内容按照写的字段 顺序写（有些字段类型是非空的，所以必须注意）

只要tablename后面写的有字段，就按照写的字段顺序写

**\* 插入空值**

**不指定或insert into tablename values（null）**

查询表内容

select \* from tablename；（在自己用户下查看）

Oracle中的日期格式（指的是输入字段类型为日期时，输入的格式）

默认格式 dd-mm-yyyy ‘6-1月-2016’

更改日期格式 alter session set nls\_date\_format =’yyyy-mm-dd’

修改内容（update）:按照英文意思也可以叫做 更新内容

update tablename set 修改的内容 where 修改的条件；

例：

update stu set sex = ‘女’（如果不加条件，更改的是所有的sex，将所有性别修 改为女）

update stu set sex = ‘女’where id = 1 （只将id = 1 的性别改为女）

update stu set sex = ‘女’where id = 1 and job = 2 （同时满足两个条件，修改 为女）

update stu set sex = ‘女’where id = 1 or job = 2 （只要满足其中一个条件，就 修改）

update stu set sex = ‘女’，job=2 where id = 1 （修改id=1的sex和job两个字 段的内容）

**\* 修改空值 （空值必须写成 is null）**

**~~update tablename set sex = ‘男’where sex = null ；（这是错误的）~~**

**update tablename set sex = ‘男’where sex is null ；**

**将 sex字段下的空值都改为 男**

删除内容

delete from tablename where 删除的条件；

注意：删除带有英文的条件时，区分大小写

复制 emp 表

create table emp1（新建一个空表的名字） as select \* from emp（原表名）；

查询内容

按字段查询

语法

select 字段名1、字段名2… from 方案.表名 [where 过滤的条件]

给字段取别名

select 原字段名 as “别名”，原字段名 别名 from emp；

例如：select ename as “员工姓名”，deptho 部门号 from emp；

**\*** **as 和 双引号 可以省略 在 sql server 中通用**

字段的计算

查看所有员工12个月的基本工资总和，也就是年薪

select ename 姓名，sal\*12 年基本工资 from emp ；

计算员工月薪（月基本工资+奖金）

select ename 姓名 ，sal+comm 月薪 from emp；

处理空值的函数（Oracle独有）

**\* 空值和任何值运算，结果都是空**

语法：nvl （空值的字段名，想要将空值更改为什么值）

例：将 奖金 的空值全部更改为0用作计算

select ename 姓名，sal+nvl（comm ， 0） 月薪 from emp；

**\* SQL中处理空值的函数：is null（空值的字段名，想要将空值更改为什么值）**

使用连接字符串（ || ）

select ename ||‘ 的工资是： ’||sal from emp ；

查询结果

员工姓名的工资是：工资

过滤多余的重复记录只保留一条（distinct）

select distinct deptno from emp；

查询在某个范围 between…and…

查询工资在2000到3000之间的员工信息

select \* from emp where sal >=2000 and sal <= 3000 ;

等价于

select \* from emp where sal between 2000 and 3000 ;

查询工资不在2000 到 3000 之间的员工信息

select \* from emp where sal <=2000 and sal >= 3000 ;

等价于（between 前加not 否定即可）

slect \* from emp where sal not between 2000 and 3000 ;

属于若干个孤立的值 in

查询部门号为 10 和 30 的员工信息

select \* from emp where deptho = 10 or deptho = 30 ；

等价于

select \* from emp where deptho in （10,30）；

查询没有奖金的员工信息 即查询空值 null （用到 is null）

select \* from emp where comm is null ；

以某个字段排序 oder by 所有语句中，都是写在语句最后

**\* asc 升序 可以省略不写，默认是升序**

**\* desc 降序**

查询工资由高到低排序

select \* from emp order by sal desc ；

按部门由低到高排，工资由高到低排

select \* from emp order by deptho ，sal desc ；

查询系统日期

select sysdate from dual ；

模糊查询

语法：select … from 表名 where 字段名 like 匹配条件 ；

% （百分号） 表示任意0个或多个字符

\_ （下划线） 表示任意单个字符

例：查找员工姓名以S开头的员工

select \* from emp where enama like ‘ S% ’；

查找员工姓名含有S的员工

select \* form emp where ename like ‘ %S% ’；

查找员工姓名第二个字母是S的员工

select \* from emp where ename like ‘ \_S% ’；

分组函数（聚合函数）

求和 sum（）

求工资的总和：select sum（sal）工资总和 from emp ；

求最大和最小值 max（） min（）

求工资的最大和最小值：select max（sal） 最高工资，min（sal）from emp ；

求平均值 avg（）

求平均工资：select avg（sal）平均工资 from emp；

计数 count（）

查询员工总数

select count（\*） from emp ； \*表示所有，就相当于把所有的员工行数就 查询出来了

select count（empno）from emp； 一般都是查询主键的字段名，唯一性，不 可能为空，查询的比较准确

select count（ename）from emp； 一般不会直接查询员工姓名，不准确吧， 反正感觉不顺眼

统计有多少部门（部门必定有重复所以要加distinct）

select count（distinct deptho）from emp；

分组（相当于将数据分组后进行查询，应该是将多个重复的变为几个组）

语法：group by 和 having 子句

group by 用于对查词的结果分组统计（就是分组）

having 用于限制分组显示结果（就是对分组进行条件过滤，只用于分组，和 where不同，而且，当不需要过滤条件时，可省略不写）

例：

求每个部门的平均工资

select avg（sal）平均工资，deptho from emp group by deptho ；

求部门平均工资大于2000的部门号及平均工资

select avg（sal）平均工资，deptho from emp group by deptho having avg （sal）>2000；

**\* 只有在having中，分组函数才能跟数字作比较**

单表查询的语法

select 字段（查询内容）

from 表名

where 整体过滤的条件

group by 分组字段

having 分组过滤的条件

order by 字段名（排序）

多表查询

原理：

select \* from emp ，dept

连接方法一：

select \* from emp ，dept where emp.deptno = dept.deptno ;

连接方法二：

select \* from emp join dept on emp.deptno = dept.deptno where 其他过滤条件

注意：两张表相连至少需要一个连接条件；n张表相连，至少需要n-1个条件。

多表查询时，给表取别名（与给字段取别名的方法一样，as “别名” 或者 直接写别名）

select \* from emp e ，dept d；

或者

select \* from emp as “e”，dept as “d”；

emp表与salgrades表相连

select \* from emp e，salgrades s

where e.sal between s.losal and s.hisal ；

三表相连

连接条件：n张表相连，需要n-1个条件

select \* from emp e，dept d, salgrades s

where e.deptno = d.deptno and e.sal between s.losal and s.hisal ;

自连接 → 直接上代码，一看就懂

查询SIMTH的上级

select e1.ename , e2.ename

from emp emp1, emp emp2

where e1.mgr = e2.empno and e1.ename = ‘SMITH’;

子查询

\* 单列子查询

查询薪资比SIMTH高的员工信息

select \* from emp where sal > (select sal from emp where ename = ‘SIMTH’);

\* 多列子查询

查询与SIMTH部门和职位完全相同的所有员工信息

select \*

from emp

where (deptno , job) = (select deptno, job from emp where ename = ‘SMITH’);

**\* 注：黄色部分的前后顺序必须一致**

不等于在Oracle中的表示方法

有两种：!= <>

在from子句中使用子查询

显示高于自己部门平均工资的员工信息

select \*

from emp e , (select deptno , avg(sal) a\_avg from emp group by deptno) a

where e.deptno = a.deptno and e.sal > a.a\_avg ；

**\* 注：黄色部分为什么要加别名**

**如果不加，后面判断工资高于部门平均工资就会是 e.sal > a.avg (sal) 这样执行是 不会通过的，不要问为什么反正不通过的，就要加个别名就好了**

子查询在create、update中的应用

在create中的应用

复制表的结构及内容

复制所有内容及结构：

create table myemp as select \* from emp；

复制某些字段的内容及结构：

create table myemp（id，name…）as select empno，ename from emp；

仅复制表的结构，不复制表的内容

法一：

直接复制表所有内容和结构，然后用delete删除内容

法二：

create table myemp as select \* from emp where 1=2；

在update中的应用

希望更新员工scott 的岗位、工资、补助与员工Smith的一样

update emp

set （job，sal，comm） = （select job，sal，comm from emp where ename = ‘smith’）where ename = ‘scott’；

内连接和外连接

之前的查询都是内连接

外连接的概念或者作用（外连接与内连接的区别）

内连接：查询并显示仅满足过滤条件的内容

外连接：查询并显示所有的内容，主要体现在空值

外连接分为：左外、右外、完全连接

**①左外连接 left join** (如果左侧的表完全显示我们就说是左外连接）

**②右外连接 right join** (如果右侧的表完全显示我们就说是右外连接)

**③完全外连接** **full join** (完全显示两个表，没有匹配的记录置为空)

**④相等连接 inner join** (？不太清楚)

语法一：

select \*

from emp e

left（right）join dept d

on e.deptno = d.deptno

[where ];

语法二：

select \* from emp e, dept d where e.deptno(+) = d. deptno ;

**\* “+”号加在匹配的表上，狭义一点可以理解为，需要全部显示的表不加“（+）”**

**反之加了“（+）”的不完全显示，这个时候就是外连接，无法区分是左外和右外， 没有什么标准区分。**

分页查询

select \*

from (select a.\*，rownum rn from （select \* from emp order by sal desc）a) e

where e.rn between 4 and 9 ；

递归查询

查询编号为7499的上级的上级的上级的……所有信息

select \*

from emp

start with empno = 7499 connect by prior mgr = empno；

函数

字符函数

1、lower（char）

将字符串表达式char中的所有大写字母转换为小写

select lower（ename） from emp；

2、upper（char）

将字符串表达式char中的所有小写字母转换为大写

select upper（‘dog’） from dual；

3、initcap（char）

首字母转换为大写

select initcap（ename）from emp；

4、substr（char，start，length）

返回字符串表达式char中从第start位开始的length个字符

select substr（ename，1,3）from emp；

表示：返回从名字的第一位开始到第三个字符，也就是取姓名的前三个字符

5、length（char）

返回字符串表达式char的长度

select length（ename）from emp；

扩展：以首字母小写显示员工姓名

select lower(substr(ename,1,1)) || substr(ename, 2,length(ename)-1) from emp;

6、select （char，search\_string, replace\_string）

替换某个字段，也就相当于给那个字段取别名，无意义

select （ename，‘smith’，‘dog’）form emp；

7、select instr（父串，子串，取第几个）from dual；

取子串在父串中的位置，

select instr（‘big big dog’，‘big’，2）from dual；

父串为‘big big dog’而子串‘big’取第二个big也就是‘big big dog’的 位置

数学函数（主要用于银行和结账等系统）

1、round（n,[m]） 四舍五入

用于执行四舍五入，如果省掉m，则四舍五入到整数，

如果m是正数，则四舍五入到小数点m位后，

如果是负数，则四舍五入到小数点m位前

select round （avg（sal）） from emp； 四舍五入取整

select round （avg（sal），1） from emp； 四舍五入取小数点后一位

select round （avg（sal），-1） from emp；四舍五入取小数点左(前)一位

2、trunc（n,[m]） 截取

用于截取数字，如果省掉m，就截去小数部分

如果是正数，截取小数点m位后，

如果是负数，截取小数点m位前

语法与round 一样

区别（round 与 trunc）

①、round需要四舍五入

②、trunc直接截取，不需要四舍五入

3、mob（m，n） 取余

select mob（5,3）from dual；

取5/3的余数

4、floor（n） 返回小于或等于n的最大整数

5、ceil（n） 返回大于或等于n的最小整数

日期函数

1、sysdate 返回系统时间

select ceil（sysdate - hiredate） 入职天数 from emp；

2、add\_months(d,n) 将日期加上特定月份

select add\_months(sysdate,2) from dual;

在系统时间的基础上加上两个月

扩展：查询入职时间超过8个月的员工

select \*

from emp

where add\_months (hiredate,8) < sysdate;

或者 add\_months (hiredate,-8) > sysdate;

3、last\_day(d) 返回指定日期所在月份的最后一天

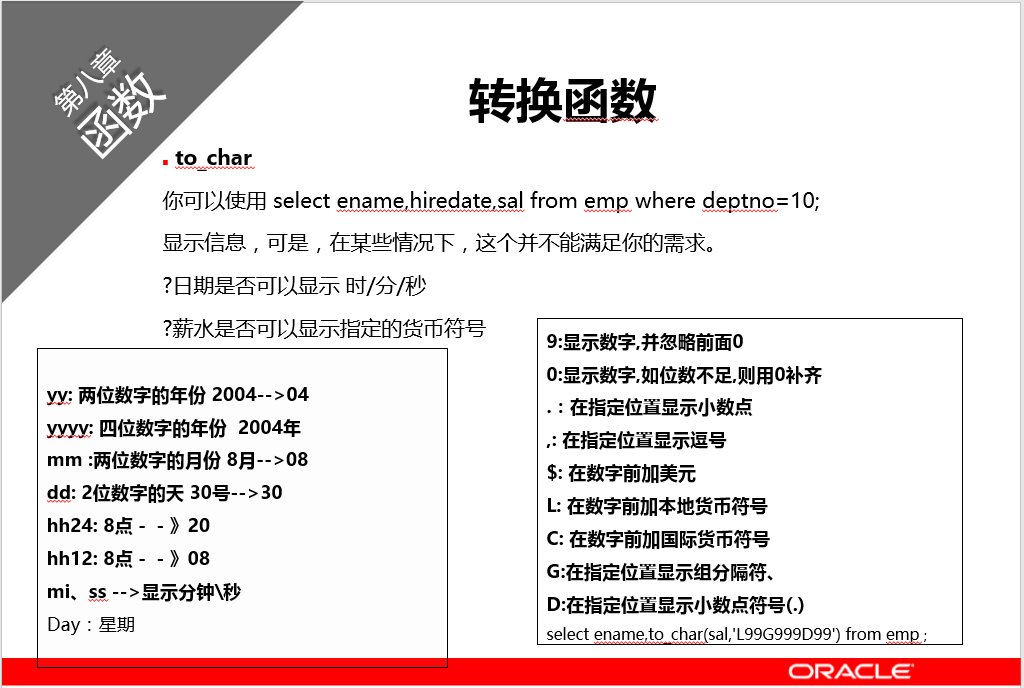
select last\_day（sysdate） from dual；

4、months\_between(n,m) 返回n和m之间的月份数

select months\_between (sysdate,‘1-5月-2016’) from dual；

转换函数

to\_char



to\_date 用于将字符串转换为date类型的数据

临时转换日期格式

insert into emp (empno,hiredate)

values (1000,to\_date(‘2016-6-1’，‘yyyy-mm-dd’))；

视图： 一张假表

先给建视图的权限 create view

语法：create [or replace] view 视图名 as （select查询语句）[with read only]；

create [or replace] view 视图名 as （select查询语句）[with check option]；

对视图进行DML操作（增删改）

删除视图： drop view 视图名

索引

创建索引

单列：create index 索引名 on emp（deptno）；

用emp的deptno创建索引

多列：create index 索引名 on emp（deptno，job）

（deptno，job）顺序影响查询速度，前面的速度快，先查了deptno才会查job， 所以经常查询的放在前面，次要的放在后面

行地址：select rowed，dname from dept；

删除索引：drop index 索引名

序列（序列一般用在主键中）

创建序列（权限）： create sequence

create sequence stu\_seq 序列的名字

start with 1 从哪个数字开始

increment by 1 以1增长，每次加1

maxvalue 2000 最大值

minvalue 1 最小值 = 开始值

nocycle 不循环，cycle循环

cache 10； 缓存1-10，预先分配的序列号，如果这一项什么都不 写，默认为 cache 20

使用序列说明

nextval（下一个值）和 currval（当前值）

第一次使用时先使用stu\_seq.nextval 因为必须要先下一个值啊，不然都没有激活 当前值是多少？那是没有的。别问为什么，就是这么规定的

查询下一个值

select stu\_seq.nextval from dual;

查询当前值

select stu\_seq.currval from dual;

使用序列方法（语法）

insert into stu values(stu\_seq.nextval,‘AA’)；

跳号问题的解决方法

法一：

create sequence stu\_seq

start with 0 改为0开始就可以避免跳号

increment by 1

法二：

Oracle中有一个延迟段，可以关闭延迟段，解决跳号问题，在未使用之前就使用这 个命令，只需要一次，就会永远关闭，并解决跳号问题，也可以不在序列中直接设 置……

alter system set deferred\_segment\_creation = false；

删除序列

drop sequence 序列名；

更新序列（一般不用，直接删了，重新建一个就好了）

alter sequence stu\_seq maxvalue 5000 cycle；

🡪还可以更改其他的数值，加上就好了

同义词 （私有同义词，公有同义词）

相当于给对象取别名，没多大意义，直接加上对象查询就ok的

私有同义词：只能当前用户使用，也就是创建同义词的那个用户

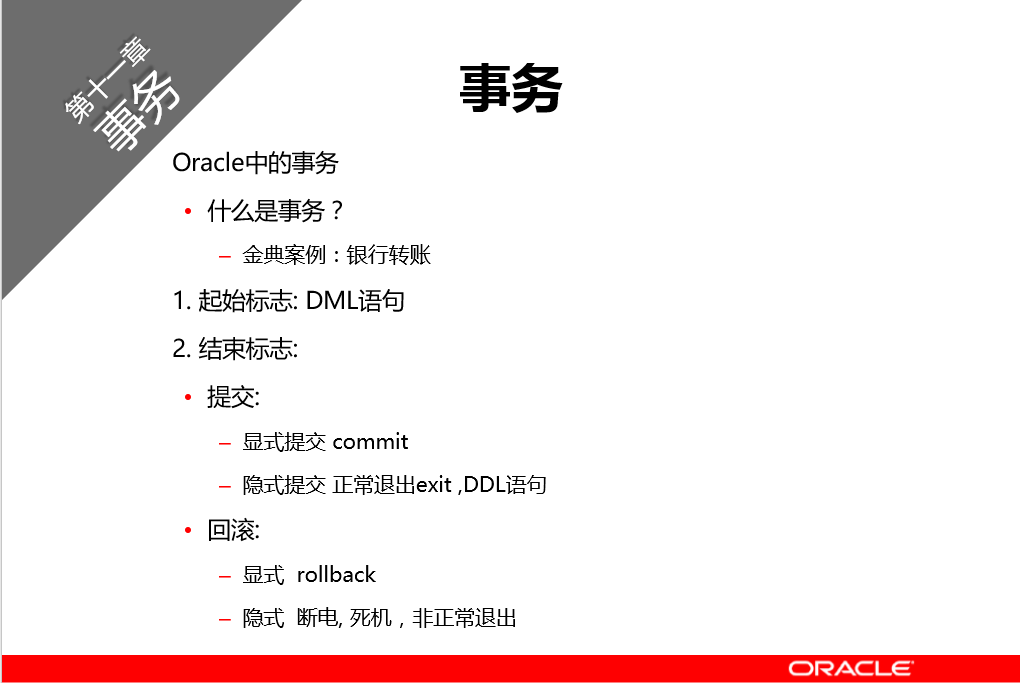
create synonym semp for scott.emp；

公有同义词：所有用户都能使用

create public synonym semp for scott.emp；

删除同义词：drop synonym 同义词名；

事务（经典案例：银行转账）



创建保存点： savepoint 保存点名；

事务回滚： rollback to 保存点名；

提交事务： commit

sql\*plus 设置自动提交： set autocommit on/off；

set serveroutput on 打开服务

**补全**

给其他用户查询所有表的权限

grant select any table to user;