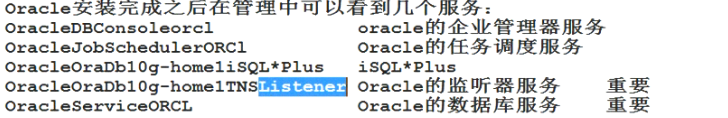
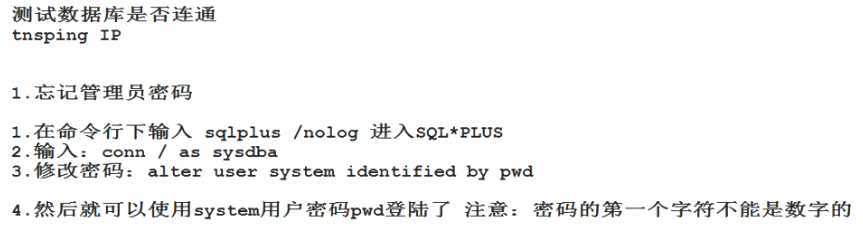
# oracle的安装

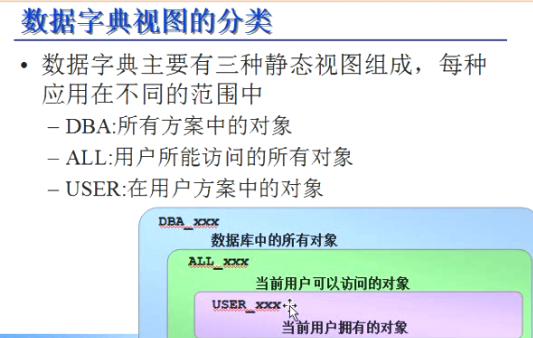
oracle安装之后可以看到的几个服务



测试oracle数据库是否连通

：

Oracle数据字典的视图分类



## Grant授权详解:

|  |
| --- |
| --select \* from dba\_users; 查询数据库中的一切用户  　　--alter user TEST\_SELECT account lock; 锁住用户  　　--alter user TEST\_SELECT account unlock; 给用户解锁  　　--create user xujin identified by xujin; 树立用户  　　--grant create tablespace to xujin; 授权  　　--grant select on tabel1 to xujin; 授权查询  　　--grant update on table1 to xujin;  　　--grant execute on procedure1 to xujin 授权存储流程  　　--grant update on table1 to xujin with grant option; 授权更新权限转移给xujin用户，许进用户能够继续授权；  　　--索回权限  　　--revoke select on table1 from xujin1; 索回查询select表的权限；  　　--revoke all on table1 from xujin;  　　/\*grant connect to xujin;  　　revoke connect from xujin  　　grant select on xezf.cfg\_alarm to xujin;  　　revoke select on xezf.cfg\_alarm from xujin;\*/  　　--select table\_name,privilege from dba\_tab\_privs where grantee='xujin' 查询一个用户拥有的对象权限  　　--select \* from dba\_sys\_privs where grantee='xujin' 查询一个用户拥有的系统权限  　　--select \* from session\_privs --当钱会话有效的系统权限  　　--角色  　　--create role xujin1;--树立xujin1角  　--grant insert on xezf.cfg\_alarm to xujin1; 将插入表的信息  　　--revoke insert on xezf.cfg\_alarm from xujin1; 索回xujin1角色的权限  　　--grant xujin1 to xujin ; 将角色的权限授权给xujin;  　　-- create role xujin2;  　　--grant xujin1 to xujin2; 将角色xujin1授权给xujin2;  　　--alter user xujin default xujin1,xujin2; 修正用户默许角色  　　-- DROP ROLE xujin1;删除角色1；  　　--select \* from role\_sys\_privs where role=xujin1;  　　--检查许进1角色下有什么系统权限；  　　--select granted\_role,admin\_option from role\_role\_privs where role='xujin2';  　--检查xujin1角色下面有什么角色权限  　　-select \* from role\_sys\_privs where role='xujin2';  　　--select table\_name,privilege from role\_tab\_privs where role='xujin1';  　　--select \* from dba\_role\_privs where grantee='xujin' --检查用户下面有多少个角色； |

## Oracle建表语句:

|  |
| --- |
| 建表一般来说是个挺简单的事情，但是Oracle的建表语句有很多可选的参数，有些我们可能平时不太用，用的时候又不知道怎么用，这里就写一个较完整的建表的例子：  CREATE TABLE banping  (id NUMBER(5)  CONSTRAINT banping\_id\_pk PRIMARY KEY,  last\_name VARCHAR2(10)  CONSTRAINT banping\_last\_name\_nn NOT NULL,  first\_name VARCHAR2(10) NOT NULL UNIQUE,  userid VARCHAR2(8)  CONSTRAINT banping\_userid\_uk UNIQUE,  start\_date DATE DEFAULT SYSDATE,  title VARCHAR2(10),  dept\_id NUMBER(7)  CONSTRAINT banping\_dept\_id\_fk REFERENCES dept(id),  salary NUMBER(11,2),  user\_type VARCHAR2(4)  CONSTRAINT banping\_user\_type\_ck CHECK  (user\_type IN(‘IN’,'OUT’)),  CONSTRAINT banping\_uk\_title UNIQUE (title,salary)  )  INITRANS 1 MAXTRANS 255  PCTFREE 20 PCTUSED 50  STORAGE( INITIAL 1024K NEXT 1024K PCTINCREASE 0 MINEXTENTS 1 MAXEXTENTS 5)  TABLESPACE data  ;  解释说明如下：  语法  CREATE TABLE [schema.]table  (column datatype [, column datatype] … )  [TABLESPACE tablespace]  [PCTFREE integer]  [PCTUSED integer]  [INITRANS integer]  [MAXTRANS integer]  [STORAGE storage-clause]  [LOGGING | NOLOGGING]  [CACHE | NOCACHE] ];  Schema：表的所有者  Table：表名  Column：字段名  Datatype：字段的数据类型  Tablespace：表所在的表空间  Pctfree：为了行长度增长而在每个块中保留的空间的量（以占整个空间减去块头部后所剩余空间的百分比形式表示），当剩余空间不足pctfree时，不再向该块中增加新行。  Pctused：在块剩余空间不足pctfree后，块已使用空间百分比必须小于pctused后，才能向该块中增加新行。  INITRANS：在块中预先分配的事务项数，缺省值为1  MAXTRANS：限定可以分配给每个块的最大事务项数，缺省值为255  STORAGE：标识决定如何将区分配给表的存储子句  LOGGING：指定表的创建将记录到重做日志文件中。它还指定所有针对该表的后续操作都将被记录下来。这是缺省设置。  NOLOGGING：指定表的创建将不被记录到重做日志文件中。  CACHE：指定即使在执行全表扫描时，为该表检索的块也将放置在缓冲区高速缓存的LRU列表最近使用的一端。  NOCACHE：指定在执行全表扫描时，为该表检索的块将放置在缓冲区高速缓存的LRU列表最近未使用的一端。  STORAGE子句：  INITIAL：初始区的大小  NEXT：下一个区的大小  PCTINCREASE：以后每个区空间增长的百分比  MINEXTENTS：段中初始区的数量  MAXEXTENTS：最大能扩展的区数  如果已为表空间定义了MINIMUM EXTENT，则表的区大小将向上舍入为MINIMUM EXTENT值的下一个较高的倍数。  外键关联的表dept的id列必须是唯一的或者是自身的主键，如不是可以用以下语句填加：  alter table dept add constraint dept\_id\_pk primary key(id)  块空间使用参数可用来控制对数据段和索引段空间的使用：  控制并发性参数：  INITRANS和MAXTRANS指定初始的和最大的事务位置数，这些事务位置在索引块或者数据块内创建。事务位置用来存储在某一事件点上正在对块进行更改的事务的信息。一个事务只占用一个事务位置，即使它正在更改多行或者多个索引条目。 INITRANS对数据段的缺省值为1，对索引段的缺省值为2，以保证最低程度的并发。例如，如果INITRANS设为3，则保证至少3个事务可以同时对块进行更改。如果需要，也可以从块空闲空间内分配其它事务位置，以允许更多的事务并发修改块内的行。 MAXTRANS的缺省值为255，它设置可更改数据块或者索引块的并发事务数的限制。设置后，该值限制事务位置对空间的使用，从而保证块内有足够的空间供行或者索引数据使用。  控制数据空间使用的参数：  数据段的PCTFREE指定每个数据块中保留空间的百分比，用于因更新块内的行而导致的增长。PCTFREE的缺省值为10%。 数据段的PCTUSED代表Oracle服务器试图为表内的每个数据块维持的已用空间的最低百分比。如果一个块的已用空间低于PCTUSED，则将这块放回到空闲列表中。段的空闲列表示容纳将来所插入内容的可选择块的列表。根据缺省，每个段在创建时都有一个空闲列表。PCTUSED的缺省值为40%。 PCTFREE和PCTUSED都按可用数据空间百分比来计算，可用数据空间是从整个块大小减去块头空间后剩余的块空间。 块空间使用参数只能针对段指定，而不能在表空间级别设置。  下面步骤介绍对PCTFREE=20且PCTUSED=40的数据段如何管理块内空间：  1.向块中插入行，直到块内的空闲空间小等于20%。当行所占用的块内数据空间达到80%（100-PCTFREE）或者更多后，即无法再向该块进行插入。  2.剩余的20%可在行大小增长时使用。例如，更新初始为NULL的列并分配一个值。这样，由于更新，块使用率可能超过80%。  3.如果由于更新，删除了块内的行或者行大小减少，块使用率可能跌至80%以下。但是，仍然无法向块中插入，直到块使用率跌至PCTUSED以下，在本例中PCTUSED为40%。  4.当块使用率跌至PCTUSED以下后，该块可用于插入。随着向块内插入行，块使用率增长，重复从步骤1开始的循环。 |

## 表分区:

|  |
| --- |
| -----------为表进行分区。  create table sales(  product\_id varchar2(50),  sales\_date date not null,  sales\_cost number(10)  )  partition by range(sales\_cost)(  partition p1 values less than(1000),-----按照销量进行分区  partition p2 values less than(2000),  partition p3 values less than(3000)  );  insert into sales values('v001','02-5月-2007',900);  insert into sales values('v002','03-5月-2007',1900);  insert into sales values('v003','04-5月-2007',2900);  insert into sales values('v004','04-5月-2007',3900);-----出错  -------查看用户的表和分区  select table\_name,partition\_name from user\_tab\_partitions;  ----查看分区的值  select \* from sales partition(p4);  -------删除分区  alter table sales drop partition p1;  ------添加分区  alter table sales add partition p4 values less than (4000);  -------自动创建分区  create table employee(  emp\_id number(4),  emp\_name varchar2(15),  emp\_address varchar2(15),  depariment varchar2(10)  )  partition by hash (depariment) partitions 4;  ---------授权同义词给用户  grant create synonym to test\_rom; |