oracle本身不区分大小写 （会把关键字全部转为大写再执行），但是对引号里的字符区分大小写

序列名区分大小写。

## 视图：

视图不存储数据 其数据来源于查询原表；

1.简化操作，提高工作效率

2.数据安全性

缺点

1)性能差

　 2)修改有点限制

CREATE VIEW 视图名字 AS select语句

简单视图（单表视图）可以增删改查

## 序列：

序列: Sequence 是oracle提供的用于产生一系列唯一数字的数据库对象。由于oracle中没有设置自增列的方法，

所以我们在oracle数据库中主要用序列来实现主键自增的功能。

创建序列：

CREATE SEQUENCE sequence //创建序列名称

[INCREMENT BY n] //递增的序列值是 n 如果 n 是正数就递增,如果是负数就递减 默认是 1

[START WITH n] //开始的值,递增默认是 minvalue 递减是 maxvalue

[{MAXVALUE n | NOMAXVALUE}] //最大值

[{MINVALUE n | NOMINVALUE}] //最小值

[{CYCLE | NOCYCLE}] //循环/不循环

[{CACHE n | NOCACHE}];//分配并存入到内存中

//序列调用 产生一个新的序列

select seq\_test.nextval from dual

//查看当前序列的值

select seq\_test.currval from dual

dual 表

dual是一个虚拟表，用来构成select的语法规则，oracle保证dual里面永远只有一条记录。我们可以用它来做很多事情，如下：

　　1、查看当前用户，可以在 SQL Plus中执行下面语句 select user from dual;

　　2、用来调用系统函数

　　select to\_char(sysdate,'yyyy-mm-dd hh24:mi:ss') from dual;--获得当前系统时间

　　select SYS\_CONTEXT('USERENV','TERMINAL') from dual;--获得主机名

　　select SYS\_CONTEXT('USERENV','language') from dual;--获得当前 locale

　　select dbms\_random.random from dual;--获得一个随机数

　　3、得到序列的下一个值或当前值，用下面语句

　　select your\_sequence.nextval from dual;--获得序列your\_sequence的下一个值

　　select your\_sequence.currval from dual;--获得序列your\_sequence的当前值

　　4、可以用做计算器 select 7\*9 from dual;

​

## 索引：

将表和索引部署到不同硬盘中的表更空间中 能提高访问性能，因为oracle能够并行读取不同硬盘的数据，这样还可以避免io冲突

基数低的列不适合B树索引 ，适合位图索引

对于性别，可取值的范围只有'男','女'，并且男和女可能各站该表的50%的数据，这时添加B树索引还是需要取出一半的数据， 因此完全没有必要。相反，如果某个字段的取值范围很广，几乎没有重复，比如身份证号，此时使用B树索引较为合适。事实上，当取出的行数据占用表中大部分的数据时，即使添加了B树索引，数据库如oracle、mysql也不会使用B树索引，很有可能还是一行行全部扫描。

如果某列是自增的 要建反向键索引 它可以将添加的数据随机分散到索引中

如果某个表的主键是自增 ，由于oracle会为主键建立默认索引（B树索引），添加数据会偏向某一边的索引 最终导致某一边的叶子节点被大量争夺

位图索引sql

Create bitmap Index 索引名字 On 表名字(列名);

反向键索引sql

Create bitmap Index 索引名字 On 表名字(列名) reverse ;

默认索引 修改为方向键索引sql

Alter index 索引名 rebuild reverse；

基于函数的索引：其是常规的B树索引 但它存放的数据是函数后得到的，而不是存放表中数据本身

Create bitmap Index 索引名字 On 表名字(函数名（列名）);

注意事项：

索引的开销大 每当插入，删除，更新 oracle都必须对索引进行更新

oracle会自动给表的主键创建索引

建设原则:

　1、索引应该经常建在Where 子句经常用到的列上。如果某个大表经常使用某个字段进行查询，并且检索行数小于总表行数的5%。则应该考虑。

　2、对于两表连接的字段，应该建立索引。如果经常在某表的一个字段进行Order By 则也经过进行索引。

　3、不应该在小表上建设索引。

创建索引:

　单一索引:Create Index 索引名字 On 表名字(Column\_Name);

　复合索引:Create Index 索引名字 On emp(deptno,job); —>在emp表的deptno、job列建立索引。

select \* from emp where deptno=66 and job='sals' ->走索引。

　　 select \* from emp where deptno=66 OR job='sals' ->将进行全表扫描。不走索引

　 　 select \* from emp where deptno=66 ->走索引。

　　 select \* from emp where job='sals' ->进行全表扫描、不走索引。

## 存储过程和存储函数

指存储在数据库中供所有用户程序调用的子程序叫存储过程、存储函数。

存储过程、存储函数的区别在于，前者没有返回值，后者有返回值

## PL/SQL

PL/SQL是Oracle对标准数据库语言SQL的过程化扩充，它将数据库技术和过程化程序设计语言联系起来，是一种应用开发语言

## 数据导出导入

导出数据

exp 用户名/密码@127.0.0.1/ORCL owner=用户名 grants=y file=D:\backup\OSYUNWEI.dmp log=D:\backup\OSYUNWEI.log

添加用户

alter system set DB\_CREATE\_FILE\_DEST = 'E:\oracle\app\oracle';

create tablespace SCF\_DATA2;

CREATE USER zylin

IDENTIFIED BY 111111

DEFAULT TABLESPACE SCF\_DATA2;

GRANT CONNECT TO zylin;

GRANT RESOURCE TO zylin;

GRANT CREATE VIEW TO zylin;

GRANT SELECT ANY DICTIONARY TO zylin;

GRANT UNLIMITED TABLESPACE TO zylin;

GRANT dba TO zylin;

oracle数据导入

imp full=y file=D:\zylin.dmp

删除用户

drop user 用户名字 cascade

## 分区技术

范围分区

散列分区（hash分区）

用户和模式是 一 一 对应的 并且名字相同

Profile文件 管理密码

## 事务

设置回退点

Savepoint 回退点名称

回退

Rallback to 回退点名称

系统再事务执行期间发送错误，死锁和中止等情况，系统会自动隐式回滚

锁 oracle得锁管理和分配都自动完成。不需要用户干预

当然 也提供了手工加锁的命令

Oracle的行级锁