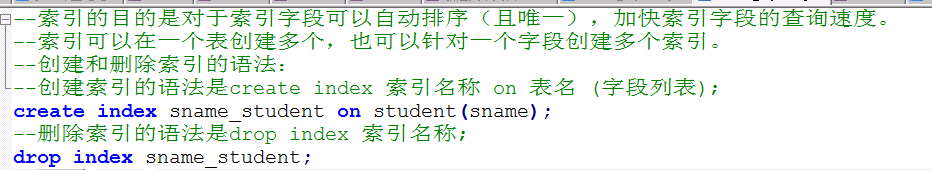
## DQL多表查询

1. 索引和视图

索引：即数据和储存位置的对应，是表的一个概念部分，用以快速定位数据和排序数据，提高检索效率。对某数据创建索引后查询该数据时系统会使用索引以降低查询速度，否则系统会先“全表扫面”找到符合条件的再显示出来，需要进行大量I/O操作，会影响性能。

创建索引： CREATE INDEX *索引名(格式一般为col\_table)* ON *table*(*column*);

删除索引： DROP INDEX 索引名;



优点：筛选数据，简化查询语句(让你编写方便而不是提高检索效率)

SET autotrace trace explain; 启用查询计划

→select语句→可以查看该语句执行的开销

缺点：额外占用磁盘空间，还需要维护(随数据表改变而改变，定期重构索引是必要的)

分类：单列索引和复合索引、唯一索引(如主索引)和不唯一索引、标准索引(B树索引ROOT+BRANCH NODE+LEAF NODE)、位图索引、函数索引

视图：一个内容由查询定义的虚表。本质和SELECT语句、子查询没区别，只是被别名化的方式不同而已。

作用：筛选数据→安全原因(限制筛选)、简化查询语句→易于复杂查询的理解和使用

提供数据的 安全性 易用性(易查询性)

视图的定义可以来源于多个库的多个表或者其它视图

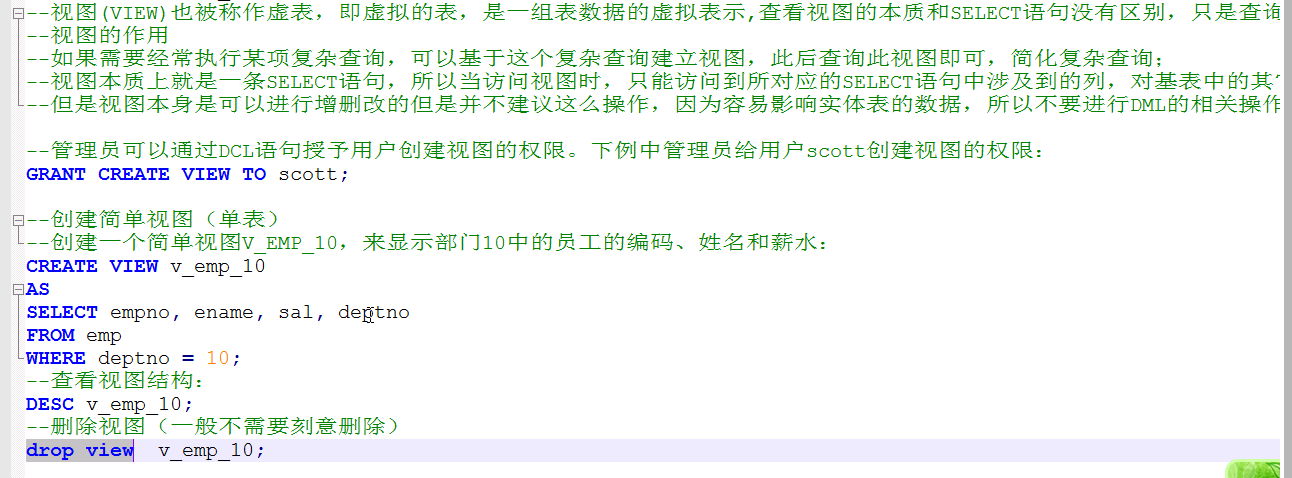
虚表像实表支持INSERT、DELETE和UPDATE，不过它不以数据值集形式储存在数据库里

视图可以给查询提供支持(快捷和限制)，如select emp.\*, rm.rownum from (子查询) rm；→ 把子查询创建为一个视图shitu1 → select emp.\*, shitu.rownum from shitu1;

创建视图： CREATE VIEW 视图名 AS 查询SELECT语句;

删除视图： DROP VIEW 视图名;

查看视图： DESC 视图名;



1. 聚合函数

avg、sum、min、max、count、

1. 类型转换函数

字符串、数字→日期： TO\_DATE(‘值’/字段,’格式’)、

日期、数字→字符串： TO\_CHAR(‘date值’/字段，’fmt’)

fmt可含YYYY\_MM\_DD HH:MI:SS如TO\_CHAR(date,’YYYY’)， TO\_number(CHAR)

1. ORACLE自带的虚拟表dual，用于以字段名、字段值的方式进行一些简单的信息获取如计算、系统时间、
2. 取整函数： ROUND(x[,y]) TRUNC(x[,y])

保留到小数后y位(个位为0位)，则从y+1位进行四舍五入/截掉

TRUNC逻辑拓展 TRUNC(SYSDATE,’MM’) ： 2019-08-01

TRUNC(SYSDATE,’MI’) ： 2019-08-30 9:56:00

只能精确到分’MI’，SS是最小时间刻度，TRUNC(SYSDATE,’SS’)，在 时-分-秒 后没有内容可截取而报错精度说明符错误

1. date可以加减运算：如

SELECT emp.\*,SYSDATE-hiredate jrts FROM emp;

nvl(comm,0)， 注意子查询中的空值结果带来的影响：需要在子查询前用nvl

in(..,null,..) →等价为 = null

SYSDATE

1. 字符函数

upper(x) 把x全大写、lower(x) 把x全小写、initcap(x) 把x首字母大写



1. 替换查询：decode(col，val，替换值[，val，替换值....]，其他替换值)

最后一段可省略，默认为空

decode(col, val1，替换后的值，val2，替换后的值2，其他替换后的值)

1. sign函数。大于0返回1，等于0返回0，小于0返回-1.

SELECT ename, sal, decode(sign(sal - 3000), 1, ‘有钱人’, -1, ‘穷人’, 0, ‘普通人’) FROM emp;

1. rownum 按SELECT后的顺序编号( ORDER BY 在SELECT 后执行)

rownum在where子句中被限制时rownum只能从1开始：<5, =1, >=1

ORDER BY注意排序。默认ASC →解决→别名化

1. 分组查询：group by 分组， having 对聚合函数作限制
2. 多表组合查询：

SELECT \* FROM 表1,表2 WHERE 连接条件;

——(表1,表2 表示两表记录笛卡尔乘积组合)可多表(>2)查询

连接条件： 等值查询↔两表时：连接查询

非等值查询：非等值的查询条件

自连接查询：给一张表两个别名来作等值条件来查询

1. 连接查询(两表组合查询)：内连接查询、左外连接查询、右外连接查询、全外连接查询

INNER JOIN、LEFT/RIGHT/FULL OUTER JOIN on

内连接：显示符合条件的数据记录组合；等效于组合查询的等值查询

内连接： SELECT \* FROM 表1 INNER JOIN 表2 ON 等值连接条件 SELECT \* FROM 表1,表2 WHERE 等值连接条件

外连接：内连接的基础上把某边连接条件值为空的记录也显示出来(全显示)

——OUTER可省略

左外连接： 左表 LEFT OUTER JOIN 右表 ON 等值条件 ——左连接左表全显

左表,右表 where 等值条件右+ ——(+表显示满足等值条件的记录)

右外连接： 左表RIGHT OUTER JOIN 右表 ON 等值条件

左表,右表 where 等值条件左+

全外连接： 左表FULL OUTER JOIN 右表 ON 等值条件 ——（不能两边加+）