-- 结构

select distinct 字段1,字段2 from 表名 where 条件 group by 字段 having 条件 order by 字段;

-- 含义

select -> 查询|显示的内容

from ->数据来源

where -> 条件， 筛选每行的条件

group by -> 按照什么标准分组

having -> 条件，筛选每组的条件

order by -> 对要显示的结果排序

-- 执行顺序

1) from -> 所有的数据

2) where -> 留下符合条件的单行记录

3) group by -> 具有共同特性一组一组的记录

4) having -> 留下的符合条件的组

5) select -> 显示记录

6) order by ->对要显示的记录进行排序

DDL数据定义语言: create:创建；drop:删除；alter:修改；rename:重命名;truncate:截断

DML数据管理(操作)语言: insert:插入；delete:删除；update:更新；select:查询

DCL 数据库控制语言: grant:授权；revoke:回收权利；commit:提交事务；rollback:回滚事务

查询行(记录): where 过滤行记录条件 ,条件有

a)、= 、 >、 =、 <=、 !=、 <>、 between and

b)、and 、or、 not、 union、 union all、 intersect 、minus

c)、null :is null、 is not null、 not is null

d)、like :模糊查询 % \_ escape('单个字符')

e)、in 、 exists(难点) 及子查询

集合操作

Union、Union All、Intersect、Minus

λ Union，并集(去重) 对两个结果集进行并集操作，不包括重复行同时进行默认规则的排序； λ Union All，全集(不去重) 对两个结果集进行并集操作，包括重复行，不进行排序 ；

λ Intersect，交集(找出重复) 对两个结果集进行交集操作，不包括重复行，同时进行默认规 则的排序；

λ Minus，差集(减去重复) 对两个结果集进行差操作，不包括重复行，同时进行默认规则的排 序

like :模糊查询

模糊查询，使用通配符:

λ %:零个及以上(任意个数的)的字符

λ \_:一个字符

λ 遇到内容中包含 % \_ 使用escape('单个字符')指定转义符

排序

使用 ORDER BY 排序，排序不是真实改变存储结构的顺序，而是获取的集合的 顺序。

λ 顺序 :asc（默认） desc

λ 多字段: 在前面字段相等时，使用后面的字段排序

λ 空排序: 降序为 desc，注意 null 为最后nulls first, nulls last

# 函数—单行函数

## 日期函数

λ sysdate/current\_date 以date类型返回当前的日期

λ add\_months(d,x) 返回加上x月后的日期d的值

λ LAST\_DAY(d) 返回的所在月份的最后一天

λ months\_between(date1,date2) 返回date1和date2之间月的数目

λ next\_day(sysdate,'星期一') 下周星期一

## 转换函数（重点\*\*\*）

to\_date(c,m) ◊ 字符串以指定格式转换为日期

select \* from emp where hiredate

between to\_date('1982-01-01', 'yyyy-mm-dd') and

to\_date('1982-12-31', 'yyyy-mm-dd');

to\_char(d,m) ◊ 日期以指定格式转换为字符串

select to\_char(sysdate,'yyyy"年"mm"月"dd"日"') from dual;

## 其他函数 (保证类型兼容)

--1)、nvl nvl(string1,string2) ◊ 如果string1为null,则结果为string2的值

select ename, nvl(null,0) from emp; select ename, nvl(to\_char(comm),'hello') from emp;

--2)、decode decode(condition,case1,express1,case2 , express2,….casen , expressn, expressionm) select ename,decode(deptno, 10,'十',20,'二十') from emp;

--3)、case when then else end

--给所有的员工 涨薪,10-->10% 20-->8% 30 -->15% 其他-->20%

--decode

select ename, sal, deptno, decode(deptno, 10, sal \* 1.1, 20, sal \* 1.08,

30, sal \* 1.15, sal \* 1.2) raisesal from emp;

--case when then else end

select ename, sal, deptno, (case deptno when 10 then sal \* 1.1

when 20 then sal \* 1.08

when 30 then sal \* 1.15 else sal \* 1.2 end)

raisesal from emp;

# 组函数

组函数|多行函数|聚合函数 即多条记录 返回一个结果。我们需要掌握如下几个组函数： avg 、sum、 min、 max、 count

1)、count :统计记录数 count() -->\* 或一个列名

2)、max min: 最大值 最小值

3)、sum：求和

4)、avg:平均值

注意: 1、组函数仅在选择列表和Having子句中有效

2、出现组函数，select 只能有组函数或分组字段

说明：

λ 组信息 与单条记录不能同时查询

--查询 最高薪水的员工名称 及薪水

~~select max(sal), ename, sal from emp;~~ ◊ 错误

select ename, sal from emp where sal=(select max(sal) from emp );

λ 组函数 不能用在 where中，能使用的地方 select having

λ null 不参与运算

# 分组

分组: group by , 将符合条件的记录 进一步的分组

过滤组:having , 过滤组信息 ，表达式 同 where 一致

## group by :分组

1)、select 出现分组函数,就不能使用 非分组信息，可以使用 group by 字段

2)、group by字段 可以不出现 select 中 ，反之select 除组函数外的，其他字段必 须出现在group by 中

## 过滤组 having

where :过滤行记录，不能使用组函数， having:过滤组 可以使用组函数

# rowid 和 rownum

## rowed

实现重复记录的删除

## rownum

1、必须排序

2、不能直接取大于 1 的数

# 表连接（92）

当我们获取的数据不是来自于同一张表而是来自于多张表时就需要使用到表连接

select \* from emp;

select \* from dept;

注意:同名列 非\* 必须区分 数据源 、关系列、 过滤条件、字段

## 笛卡尔积

--非\* 必须区分 使用表名 或别名.区分

select \* from emp , dept;

select ename , dname from emp , dept;

select ename, dname, e.deptno from emp e, dept d;

## 等值连接(在笛卡尔积基础上 取条件列相同的值)

--找出30部门的员工名称及部门名称:先关联后过滤

select ename, dname, e.deptno from emp e, dept d where e.deptno=d.deptno and e.deptno=30;

--记录很多时 :先过滤后关联

-- 数据来源： emp (select \* from emp where deptno=30) e , dept(select \* from dept where deptno=30) d

select \* from emp where deptno=30;

select \* from dept where deptno=30;

-- 查询的字段：ename, dname, e.deptno

-- 条件：e.deptno=d.deptno , deptno=30

select ename, dname, e.deptno from (select \* from emp where deptno=30) e ,(select \* from dept where deptno=30) d where e.deptno=d.deptno;

## 非等值连接 > < != <>between and

--查询员工姓名，工资及等级

-- 数据源： emp e, salgrade s

-- 字段： ename, grade, sal -- sal between losal and hisal

select ename, grade, sal from salgrade s, emp e where sal between losal and hisal;

## 自连接: 特殊的等值连接 (来自于同一张表)

--找出 存在上级的员工姓名 及上级名称

-- 数据来源： emp e, emp m

-- 字段： e.ename, m.ename

-- 条件：e.mgr=m.empno

select e.ename, m.ename from emp e, emp m where e.mgr=m.empno;

## 外连接

--找出 所有的员工姓名 及上级名称 --找出 所有部门的员工数 及部门名称

select dname, nu from dept d, (select count(1) nu, deptno from emp group by deptno) e where d.deptno(+)=e.deptno;

看+和, 主表在,的左边就叫左外连接 主表在,的右边叫右连接

# 99 连接

λ 交叉连接 cross join --->笛卡尔积

select \* from emp cross join dept;

λ 自然连接(主外键、同名列) natural join -->等值连接

select \* from emp natural join dept;

--在指定列过程中同名列归共同所有(\*除外)

select deptno,e.ename,d.dname from emp e natural join dept d;

λ join using连接(同名列) -->等值连接

select deptno,e.ename,d.dname from emp e join dept d using(deptno);

λ [inner]join on 连接 -->等值连接 非等值 自连接 (解决一切) 关系列必须区分

--on select ename, dname from emp join dept on emp.deptno = dept.deptno

where emp.deptno = 30;

on 非等值连接 、自连接 --部门编号为30的员工名称 工资等级

select ename, grade from emp e join salgrade s on e.sal between s.losal and s.hisal where deptno=30;

--部门编号为30的员工名称 上级名称

select e.ename,m.ename mname from emp e join emp m on e.mgr =m.empno where e.deptno =30;

--部门编号为30的员工 员工名称 部门名称 工资等级 上级名称

select e.ename, dname, grade, m.ename mname from emp e

join dept d on e.deptno = d.deptno

join salgrade s on e.sal between s.losal and s.hisal

join emp m on e.mgr = m.empno

where e.deptno = 30;

λ left|right [outer] join on|using -->外连接

--所有部门的 部门名称，员工数 --左外 select dname, n from dept d left outer join (select deptno, count(1) n from emp group by deptno) i on d.deptno = i.deptno;

select dname, n from dept d left outer join (select deptno, count(1) n from emp group by deptno) i using (deptno); --右外

select dname, n from (select deptno, count(1) n from emp group by deptno) i right outer join dept d on d.deptno = i.deptno;

λ full join on|using -->全连接 满足直接匹配，不满足 相互补充null ，确保 所有表的记录 都至少 出现一次

select \* from (select 1 no, 'a' "name" from dual union select 2 no, 'b' "name" from dual) a full join (select 1 no, 'c' "name" from dual union select 3 no, 'd' "name" from dual) b on a.no=b.no;

select \* from (select 1 no, 'a' "name" from dual union select 2 no, 'b' "name" from dual) a full join (select 1 no, 'c' "name" from dual union select 3 no, 'd' "name" from dual) b using(no);

# 视图

视图:建立在表|结果集|视图上的虚拟表，有以下作用

1、简化:select 查询语句

2、重用:封装select语句 命名

3、隐藏:内部细节

4、区分:相同数据不同查询

不是所有的用户都有创建视图的权限

1、前提： create view -->组 connect resource dba

2、授权: -->sqlplus /nolog

a)、sys登录 conn sys/123456@orcl as sysdba

b)、授权: grant dba to scott;

回收: revoke dba from scott;

c)、重新登录

create or replace view 视图名 as select语句 [with read only];

要求:所有列必须存在名称。

对视图的删除不会删除原有表的数据

drop view 视图名;