### Mysql select（个人版）

--查询： select \* from 表名

--算术： +，-，\* ，/，--（）可以改变优先级

--空值： 不参与运算，如要进行运算则需要先将空值转换成0

--别名： 列名 别名， 列名 AS 别名

--去重： distinct， 举例：select distinct job,deptno from emp;

--限定和排序： 限定条件--where , 排序--order by; (where 子句中的列可以不在select子句中出现 （不推荐），，-- where子句不能使用列的别名）；

--限定行号： limit (写在查询语句最后，限定显示哪些行数据），limit m,n 表示显示从第m行开始的n行数据，第一行的行号是0，第二行是1，，一般显示前几行数据的就是 limit N;

--比较运算符： <= , >= , < , > , = , != , <> ,

--其他运算符between...and...(两值之间），in(列表匹配，同时匹配多个值），like (模糊匹配）、%匹配零个或者多个字符，\_ 表示匹配任意一个字符，isnull匹配空值.(is not null)

--逻辑运算符： and , or ,not

--排序： order by , --asc 升序（默认可以不写），--desc （降序排列）可以使用别名排序 order by ‘nianxin’ desc -- 多列排序: ORDER BY deptno,sal DESC; -- 排序的列有先后顺序，请务必注意. order by 只会出现在limit之前，在其他所有子句之后；

--数字函数：round(m,n)将数字m精确到小数点后n位（支持四舍五入）,truncate(m,n)将数字m精确到小数点后n位（不支持四舍五入）;floop(x)向下取整，返回小于等于x的最大整数（不支持四舍五入）;

--字符函数： -- LOWER(x) 将字符串x转换成小写；-- UPPER(x) 将字符串x转换成大写；

--字符处理函数： --concat(x1,x2,x3.....xn)连接多个字符串；length(x)返回字符串的长度；--mid(x,m,n)截取字符串X第m个字符开始，截取长度为n的字符，n不写默认所有；left(x,m);right(x,m)

--日期函数：now()返回当前日期和时间 ; curdate()返回当前日期 ; cuitime()返回当前时间 ; 日期函数不能直接进行加减运算，需要用其他函数处理 ; YEAR(date) 返回当前日期所在的年份 ; month(date)返回当前日期所在的月份 ; day(date)返回当前日期所在的天 ;

--判断条件 ：if(条件判断，value1,value2)如果条件判断为真，则返回value1，如果条件判断为假，则返回value2,例如IF(sal>3000,'高富帅','穷屌丝')，if的嵌套例如IF(sal>3000,'高富帅',IF(sal>=1500, '中产','穷屌丝'))；ifnull(value1,value2)判断value1的值，如果该值为非空，则返回自己；如果为空则返回value2

CASE

/\*

CASE 表达式

WHEN value1 then result1

when value2 THEN result2

WHEN value3 THEN result3

...

WHEN valueN THEN resultN

ELSE resultN+1

END

\*/

-- 公司依据不同的职位进行涨薪

SELECT \* FROM emp;

CLERK +200

SALESMAN +300

MANAGER +500

其他人员 不涨薪

SELECT ename,sal,job,

(

CASE job

WHEN 'CLERK' THEN sal+200

WHEN 'SALESMAN' THEN sal+300

WHEN 'MANAGER' THEN sal+500

ELSE sal

END

) AS addsal

FROM emp;

--聚合函数不可以和聚合函数嵌套使用

--在WHERE子句中不允许使用聚合函数进行限定

--HAVING执行的顺序是在SELECT子句之后，所以可以使用前面提到的分组函数(使用在group by 后面）

--在使用HAVING时必须要有GROUP BY ，但使用GROUP BY可以没有HAVING

--在HAVING中出现的分组函数，未必要在SELECT子句中出现，主要还是看最终想查询的结果

--having 中出现的聚合函数可以不在select中出现 or（在SELECT子句的后面可以不包含聚合函数，而在GROUP BY 和HAVING中依然可以使用），，，比如：

查询最高工资不低于3000的部门的人数：

SELECT deptno,COUNT(deptno) FROM emp

GROUP BY deptno

HAVING MAX(sal)>=3000;（select中没有出现max(sal) ）

等同于

SELECT deptno,MAX(sal),COUNT(1) 'bumenrenshu' FROM emp

GROUP BY deptno

HAVING MAX(sal)>=3000；（select中出现max(sal) ）

--where 子句是限定普通条件；-- 也可以把普通的限定条件放在HAVING子句中

例如：

SELECT job, SUM(sal) FROM emp

WHERE job NOT LIKE 'SALES%'

GROUP BY job

HAVING SUM(sal)>5000

ORDER BY SUM(sal) DESC;

等同于

SELECT job, SUM(sal) FROM emp

GROUP BY job

HAVING SUM(sal)>5000 AND job NOT LIKE 'SALES%'

ORDER BY SUM(sal) DESC;

--多表查询时，如果输出项没有某一个表的项目，则select中可以不出现，在限定条件中可以出现，，但是在from 后面列出此表；

例如：

--查询每个工资等级所对应的人数

SELECT grade,COUNT(1) FROM emp e, salgrade s

WHERE e.sal BETWEEN s.losal AND s.hisal

GROUP BY grade;

-- in 可以使不同的属性分别对应 例如：

-- 查询同ALLEN部门编号和职位都相同的员工：

SELECT ename FROM emp

WHERE deptno=(

SELECT deptno FROM emp AND 连接

WHERE ename='allen') AND job=(

SELECT job FROM emp

WHERE ename='allen')

等同于：

SELECT ename FROM emp

WHERE (deptno,job) IN (SELECT deptno,job FROM emp IN 连接

WHERE ename='allen')

--如何区分什么时候选择使用多表查询，什么时候选择子查询？

-- 当最终的查询结果来自一张表的时候，只是用了其他表中的数据作为限定条件，多表查询和子查询都可以实现，推荐使用子查询（效率更高）；

-- 当最终的查询结果来自多找表的时候，只能用多表查询；

-- 子查询非常适合查询条件基于未知数据 ；

-- 子查询中可以使用函数

-- 子查询返回的结果有多个，使用多行比较运算符 IN,ANY（任意）,ALL（所有）

举例：

--查询职位比CLERK最高薪水还少（小于最大值）的其他职位的员工信息

SELECT \* FROM emp

WHERE sal<(

SELECT MAX(sal) FROM emp 用小于最大值的方法

WHERE job='clerk')

AND job != 'clerk'

等同于：

SELECT \* FROM emp

WHERE sal< ANY (SELECT sal FROM emp 用any的方法

WHERE job='clerk')

AND job !='clerk'

--把子查询的结果看做是一张新表

SELECT d.dname,d.loc, c.ct FROM dept d,

(SELECT deptno,COUNT(empno) ct FROM emp GROUP BY deptno) c -- 如果子查询的select子句中有聚合函数，请使用别名转换成普通的列

WHERE d.deptno=c.deptno

AND c.ct>5;

---列出所有部门的相应信息和部门人数，没有员工的部门则部门人数显示为0

SELECT d.\*,IFNULL(b.ab,0)FROM dept d

LEFT OUTER JOIN

(SELECT deptno,COUNT(1) ab FROM emp

GROUP BY deptno) b

ON d.deptno=b.deptno

---列出所有部门的相应信息和部门人数，没有员工的部门则部门人数显示为0

SELECT d.\*,IFNULL(ct.c,0) FROM dept d

LEFT OUTER JOIN

(SELECT deptno,COUNT(empno) AS c FROM emp GROUP BY deptno) ct -- 把子查询当做一张表来处理

ON d.deptno=ct.deptno

这个题嘛，要好好看哈，好好学哈，值得拥有啊。

---查询高于自己部门平均工资的员工名字,部门号,工资,平均工资（保留2位小数）

SELECT deptno,AVG(sal) FROM emp GROUP BY deptno -- 每个部门的平均工资

SELECT e.ename,e.deptno,e.sal,a.asal FROM emp e,

(SELECT deptno,AVG(sal) AS asal FROM emp GROUP BY deptno) a

WHERE e.deptno=a.deptno

AND e.sal>a.asal;

这个题嘛，也要好好看哈，好好学哈，值得拥有啊。

\*\*9.查询入职日期排名后5位的员工信息，且同一天入职员工要求并列展示 （这道题错点很多很多，要注意）

正确答题：

SELECT \* FROM emp

WHERE hiredate IN( ------- 最后用where 子句限定 in ()

SELECT hiredate FROM ------ 再从表a 中取出hiredate 数据

(SELECT DISTINCT hiredate FROM emp

ORDER BY hiredate DESC ------给子查询表取别名 a

LIMIT 5) a);

错误答题：

SELECT \* FROM emp

WHERE hiredate IN(

SELECT DISTINCT hiredate FROM emp ----去重，日期排序，limit限定，取其前面5个值，

ORDER BY hiredate DESC 这是对的，但是limit限定后显示的是前5个值的表

LIMIT 5); 意思就是说，这只是一个表，而不是数据

错误答题

SELECT hiredate FROM emp

ORDER BY hiredate DESC ----错的离谱，完全没考虑重复的情况 也就是同一天入职的

LIMIT 5 情况。