

MySQL数据库第三部分讲义

重点掌握DQL数据查询语言

一 单表查询主要内容

(一) 动手做(TODO 共44个任务)

单表查询语法基本格式:

1. 查询所有/指定字段
2. 查询列表中使用算术表达式
3. 定义别名
4. 去掉重复行DISTINCT (对于查询结果)
5. 通过WHERE子句过滤查询指定记录
 - (1). 使用算术比较运算符比较筛选
 - (2)使用逻辑运算符 AND和OR筛选
 - (3) 在WHERE子句中使用特殊关键字
 - 1). 带 BETWEEN AND 的范围查询
 - 2). 带 IN 关键字的查询
 - 3). 带 LIKE 的字符匹配查询
 - 4) 查询空值 (IS NULL)
6. 排序ORDER BY
7. 分组GROUP BY子句和分组聚合函数
9. HAVING子句
10. LIMIT 关键字的使用及分页查询

(二) 理解并口述 (技术点和面试点)

1. 分组函数使用注意事项
2. COUNT(*) 与COUNT(列名)及COUNT(DISTINCT 列名)的区别:
3. WHERE子句与HAVING子句的区别:
4. SELECT语句书写顺序与实际执行顺序

(三) 课后任务 (TODO 23个任务)

二 常用函数 (TODO 17个任务)

1. 数学函数
2. 字符串函数
- 3.日期和时间函数
4. 条件函数
5. 系统信息函数
6. 加密函数

MySQL数据库第三部分讲义

重点掌握DQL数据查询语言

一 单表查询主要内容

(一) 动手做(TODO 共44个任务)

单表查询语法基本格式:

```
SELECT [DISTINCT] <column1 [as new name] ,columns2,...> FROM <table1_name> [WHERE  
<search_condition>] [GROUP BY <group_by_expression>] [HAVING <group_condition>] [ORDER BY  
<column_list> [ASC|DESC]] ;
```

1. 查询所有/指定字段

- ☒ 任务1：查看部门表的所有字段信息

```
SELECT * FROM dept;
```

- ☒ 任务2：查看部门里部门号和部门名称的字段信息内容

```
SELECT deptno,dname FROM dept;
```

- ☒ 任务3：查看员工表的所有员工的姓名，工资，奖金，部门号。

```
SELECT ename,sal,comm,deptno FROM emp;
```

面试考点 在SQL查询语句中使用 * 的缺点：查出不必要的列，消耗系统时间，效率低。使用指定列的好处，不仅可以提高查询效率，还可以改变列在查询结果中的默认显示顺序。

2. 查询列表中使用算术表达式

- ☒ 任务4、查询显示员工姓名，工资和涨300元后的工资的信息。

```
SELECT ename,sal,sal+300 FROM emp;
```

- ☒ 任务5、查询显示员工工资降10%后的年薪信息

```
SELECT ename,sal, (sal-sal*0.1)*12 FROM emp;
```

3. 定义别名

```
-- 3.1 定义列的别名
-- 任务1 查询显示员工工资降10%后的年薪信息
SELECT (sal*0.9)*12 AS yearsal FROM emp;
-- 或者as可以省略
SELECT (sal*0.9)*12 yearsal FROM emp;
-- 英文别名要区分大小写或有特殊符号需要加双引号
SELECT (sal*0.9)*12 AS "YearSal" FROM emp;
-- 或者中文别名
SELECT (sal*0.9)*12 年薪 FROM emp;
-- 中文中间有空格或有特殊符号需要加双引号
SELECT (sal*0.9)*12 "年 薪" FROM emp;
-- 3.2 定义表的别名
-- 任务 给员工表起一个简单别名,查询所有员工的姓名，职位。
SELECT e.ename,e.job FROM emp e;
```

- ☒ 任务6：查询部门表dept中deptno字段，dname字段，loc字段，使用中文作为别名。（loc字段表示部门所在地址）。

```
SELECT deptno 部门编号,dname 部门名称,loc 地址 FROM dept;
```

- ☑ 任务7: 查询工资等级表salgrade 的所有信息，给工资等级表取一个别名。

```
SELECT * FROM salgrade 工资表;
```

注意:

别名中有特殊字符（如：空格，大小写）时必须加双引号，例如“Salary”, "年薪"

4. 去掉重复行DISTINCT (对于查询结果)

DISTINCT 的作用是去重。它可用来限制在查询结果显示不重复的记录，重复的记录就去掉了。DISTINCT的作用范围是后面所有字段的组合。

- ☑ 任务8 显示emp表中的job(职务) 列，要求显示的“职务”记录不重复。

```
SELECT DISTINCT job FROM emp;
```

- ☑ 任务9 查询所有员工所在的部门号，要求不重复。

```
SELECT DISTINCT deptno FROM emp;
```

- ☑ 任务10 显示部门与薪水不重复的员工信息。

```
SELECT DISTINCT deptno,sal FROM emp;
```

**** 面试要点**** DISTINCT的作用范围是后面所有字段的组合。

5. 通过WHERE子句过滤查询指定记录

在SELECT 语句中使用WHERE子句可以实现对数据行的筛选操作，只有满足WHERE子句判断条件的行才会显示在结果集中，而那些不满足WHERE子句判断条件的行则不包括在结果集中。 **注意：WHERE子句中不可以用列的别名。**

查询条件	符号或关键字
比较	=、<、<=、>、>=、!=、<>
指定范围	between and、not between and
指定集合	in、not in
匹配字符	like、not like
是否为空	is null、is not null
多个查询条件	and、or

(1). 使用算术比较运算符比较筛选

- ☑ 任务11 查询员工SMITH的姓名，工资和部门号

```
SELECT ename,sal,deptno
FROM emp
WHERE ename='SMITH';
```

- ☒ 任务12 检索出工资大于等于1500的所有员工的员工姓名和工资。

```
SELECT ename,sal
FROM emp
WHERE sal >= 1500;
```

- ☒ 任务13 检索出工资不等于3000的所有员工的员工姓名和工资。

```
SELECT ename,sal
FROM emp
WHERE sal <> 3000;
```

(2)使用逻辑运算符 AND和OR筛选

- ☒ 任务14 检索10号部门职位是'CLERK' 工资大于1000的员工的姓名和工资及部门号。

```
SELECT ename,sal,deptno
FROM emp
WHERE deptno=10 AND job = 'CLERK' AND sal>1000 ;
```

- ☒ 任务15 查询部门在30号并且工资大于1000的员工或者工资大于2000的员工的姓名，工资和部门号。

```
SELECT ename,sal,deptno
FROM emp
WHERE deptno=30 AND sal>1000 OR sal>2000;
```

- ☒ 任务16 检索不在30号部门并且工资不在1500到3000的员工的姓名，工资和部门号。

```
SELECT ename,sal,deptno
FROM emp
WHERE deptno<>30 AND (sal<1500 OR sal>3000);
```

- ☒ 任务17 查询出不在10号部门和20号部门的员工姓名和部门号。

```
SELECT ename,deptno
FROM emp
WHERE deptno<>10 AND deptno<>20;
```

(3) 在WHERE子句中使用特殊关键字

1). 带 BETWEEN AND 的范围查询

- ☒ 任务18: 查询薪水在1000到2000元的部门的员工名单。

```
SELECT ename FROM emp WHERE sal BETWEEN 1000 AND 2000;
```

- ☒ 任务19: 查询工资不在2000到3000之间的员工的姓名和工资。

```
SELECT ename,sal,deptno FROM emp WHERE sal NOT BETWEEN 2000 AND 3000;
```

- ☒ 任务20 检索不在30号部门并且工资不在1500到3000的员工的姓名，工资和部门号。

```
SELECT ename,sal,deptno FROM emp WHERE deptno<>30 AND sal NOT BETWEEN 1500 AND 3000;
```

2). 带 IN 关键字的查询

- ☒ 任务21 查询10部门，20部门和50部门的员工的薪水。

```
SELECT ename,sal,deptno FROM emp WHERE deptno IN(10,20,50);
```

- ☒ 任务22 查询出不在10号部门和20号部门的员工姓名和部门号。

```
SELECT ename,deptno FROM emp WHERE deptno NOT IN(10,20);
```

3). 带 LIKE 的字符匹配查询

- ☒ 任务23 查询名字第二个字母是'A'的员工的员工姓名，工资和部门号。

```
SELECT ename,sal,deptno  
FROM emp  
WHERE ename LIKE '_A%';
```

- ☒ 任务24 查询名字中带有'_'下划线的员工的员工姓名和工资 (可以自己插入一条数据做测试)

```
SELECT ename,sal FROM emp WHERE ename LIKE '%\_%' ESCAPE '\\';
```

注意： % 表示零或多个字符 _表示一个字符 使用ESCAPE 转义， \表示转义符

4) 查询空值 (IS NULL)

- ☒ 任务25 查询没有奖金的员工的员工姓名，工资和部门号。

```
SELECT ename,sal,deptno,comm FROM emp WHERE comm IS NULL OR comm=0;
```

- ☒ 任务26 查询有奖金的员工的员工号，姓名，工资，奖金。

```
SELECT ename ,sal,comm,deptno  
FROM emp  
WHERE comm IS NOT NULL AND comm != 0;
```

空值NULL的使用要注意:

1、NULL是未知的、不存在、不确定、不可用、未分配的值。2、空值不等于零或空格，NULL与空字符串不同，因为空值是不存在的值，而空字符串是长度为0的字符串。3、任意类型都可以支持空值。4、因为空值代表未知的值，所以并不是所有的空值都相等。所以不可以用“=”等号运算符来检查空值。5、数据库中，任何值和NULL做算术运算结果都为NULL，MySQL数据库中，NULL与字符串连接结果也是NULL。但是Oracle数据库，NULL与字符串连接结果是字符串。

6. 排序ORDER BY

- ☒ 任务27 查询出部门表的部门编号，部门名称，并按着部门编号升序排序。

```
SELECT deptno,dname FROM dept ORDER BY deptno ASC;
```

注意：ASC可以省略，表示默认升序排列。

- ☒ 任务28 查询每个员工的姓名，工资，年薪（别名ysal），并按年薪降序排序

```
SELECT ename,sal,sal*12 ysal FROM emp ORDER BY ysal DESC;
```

注意：ORDER BY中可以使用列别名，但是WHERE子句中不能使用列别名。

- ☒ 任务29 查询出每个员工的薪水与部门，并按着部门编号降序排序,工资降序。

```
SELECT ename,deptno,sal
FROM emp
ORDER BY deptno DESC,sal DESC;
```

- ☒ 任务30 查询出员工的姓名，职位，入职时间和部门号，按着入职时间升序排序和部门号排序降序排序。

```
SELECT ename,job,hiredate,deptno FROM emp ORDER BY hiredate,deptno DESC;
```

注意：DESC的作用域是作用在离它最近的列上。

7. 分组GROUP BY子句和分组聚合函数

- ☒ 任务31 查询各个部门的员工的平均工资。

```
SELECT deptno,AVG(sal) avgsal
FROM emp
GROUP BY deptno;
```

- ☒ 任务32 查询各个部门各个职位的平均工资和最大工资。并按照平均工资升序排序。

```
SELECT deptno,job,AVG(sal) avgsal,MAX(sal)
FROM emp
GROUP BY deptno,job
ORDER BY avgsal ASC;
```

- ☒ 任务33 统计10, 20, 30 号部门的员工人数。

```
SELECT deptno,COUNT(empno)
FROM emp
WHERE deptno IN(10,20,30)
GROUP BY deptno;
```

- ☒ 任务34 统计各个职位的员工人数和平均工资。

```
SELECT job,COUNT(empno),AVG(sal)
FROM emp
GROUP BY job;
```

- ☒ 任务35 查询各个部门各个岗位的平均工资。

```
SELECT deptno,job,AVG(sal)
FROM emp
GROUP BY deptno,job;
```

- ☒ 任务36 统计员工表中有奖金（包括奖金不能为0）的员工人数

```
SELECT COUNT(*)
FROM emp
WHERE comm!=0 AND comm is not null;
-- 或者
SELECT COUNT(comm)
FROM emp
WHERE comm!=0 ;
```

- ☒ 任务37 统计员工表所有员工的人数

```
SELECT COUNT(*) FROM emp;
```

- ☒ 任务38 统计职位的个数

```
SELECT COUNT(DISTINCT job) FROM emp;
```

注意 (1) COUNT() 与COUNT(列名)的区别: COUNT()将返回表格中所有存在的行的总数包括值为NULL的行, 然而COUNT(列名)将返回表格中除去NULL以外的所有行的总数(有默认值的列也会被计入), COUNT(DISTINCT 列名),得到的结果将是除去值为NULL和重复数据后的结果。(2) 分组函数用于统计表数据。MAX() 求最大值, 返回表达式中所有值的最大值。适用任何数据类型 MIN() 求最小值, 返回表达式中所有值的最小值。适用任何数据类型 AVG() 求平均值, 返回表达式中所有值的平均值, 只能用于数字 SUM() 求和, 返回表达式中所有值的和, 只能用于数字求和 COUNT() 统计计算总行数, 返回整数。

9. HAVING子句

HAVING子句:对分组后的结果进行过滤筛选。

- ☒ 任务39 列出最低薪金大于1500的各种工作及从事此工作的全部雇员人数

```
SELECT job,MIN(sal),COUNT(empno)
FROM emp
GROUP BY job
HAVING MIN(sal)>1500;
```

- ☒ 任务40 列出最高薪水小于3500的各个部门人数。

```
SELECT deptno, MAX(sal), COUNT(*)
FROM emp
GROUP BY deptno
HAVING MAX(sal)<3500;
```

- ☒ 任务41 查询各个部门平均工资小于2500的员工的平均工资和最高工资。

```
SELECT deptno,AVG(sal),MAX(sal)
FROM emp
GROUP BY deptno
HAVING AVG(sal)<2500;
```

- ☒ 任务42 列出至少有5个员工的所有部门号和所在部门的员工人数

```
SELECT deptno,COUNT(empno)
FROM emp
GROUP BY deptno
HAVING COUNT(empno)>=5;
```

10. LIMIT 关键字的使用及分页查询

LIMIT 后面的数字代表限制的条数，若条数小于总记录数，则显示限制的数量，若大于总记录数，则显示全部数据。LIMIT 接受一个或两个数字参数。参数必须是一个整数常量。如果给定两个参数，第一个参数指定第一个返回记录行的偏移量，第二个参数指定返回记录行的最大数目。初始记录行的偏移量是 0(而不是 1)。LIMIT 后面可以跟一个参数，例如 LIMIT 3，也可以跟两个参数，例如LIMIT 0,3

- ☒ 任务43 查询员工表前3名的员工的编号和姓名。

```
SELECT empno,ename FROM emp LIMIT 3;
```

--或者

```
SELECT empno,ename FROM emp LIMIT 0,3;
```

- ☒ 任务44 查询员工表第三页的员工信息信息（假设每页显示4条，总共有14条）

```
SELECT empno,ename FROM emp LIMIT 8,4;
```

注意:

MySQL的分页语句中 LIMIT后面的 第一个参数是 (page-1)*pagesize 第二个参数是pagesize 其中page表示页码，pagesize表示每一页显示的最大记录数

（二）理解并口述（技术点和面试点）

1. 分组函数使用注意事项

1) 分组函数只能出现在SELECT选择列表、HAVING子句、ORDER BY子句中。2) 在查询列表中出现的列没有在分组函数中出现的话, 就必须在GROUP BY中出现 (Oracle严格要求)。3) GROUP BY子句的位置在WHERE子句的后面, ORDER BY的前面4) 分组结果限制只能使用HAVING。GROUP BY语法规定: 首尾呼应、前后一致1) 只有出现在GROUP BY里的字段, 才能出现在SELECT后面和ORDER BY的子句中; 2) 没有出现在GROUP BY里的字段, 只有配合分组函数才能出现在SELECT和ORDER BY里面;

3) 如果在GROUP BY里的字段应用了单行函数, 那么在SELECT后面和ORDER BY子句中也要用同样的单行函数
分组查询在面试题中常常出现, 要注意使用分组函数的注意事项。

2. COUNT(*) 与COUNT(列名)及COUNT(DISTINCT 列名)的区别:

COUNT(*)将返回表格中所有存在的行的总数包括值为NULL的行 COUNT(列名)将返回表格中除去NULL以外的所有行的总数(有默认值的列也会被计入) COUNT(DISTINCT 列名),得到的结果将是除去值为NULL和重复数据后的结果

3. WHERE子句与HAVING子句的区别:

(1) WHERE 搜索条件在进行分组操作之前应用。而 HAVING 搜索条件在进行分组操作之后应用。(2) WHERE子句过滤的是行(记录), HAVING子句过滤的是分组(组标识、每组数据的聚合结果)(3) WHERE子句包含单行函数, HAVING子句只能包含GROUP BY后面的表达式和组函数(4) WHERE子句执行在前, HAVING子句执行在后(5) WHERE子句不允许用列别名, HAVING子句后面可以使用列别名

4. SELECT语句书写顺序与实际执行顺序

查询中用到的关键词主要包含六个, 并且SQL语句书写顺序依次为: SELECT--FROM--WHERE--GROUP BY--HAVING--ORDER BY 其中SELECT和FROM是必须的, 其他关键词是可选的, 这六个关键词的执行顺序与SQL语句的书写顺序并不是一样的, 而是按照下面的顺序来执行 FROM--WHERE--GROUP BY--HAVING--SELECT--ORDER BY

FROM:需要从哪个数据表检索数据 WHERE:过滤表中数据的条件 GROUP BY:如何将上面过滤出的数据分组 HAVING:对上面已经分组的数据进行过滤的条件 SELECT:查看结果集中的哪个列, 或列的计算结果 ORDER BY:按照什么样的顺序来查看返回的数据

还有FROM后面的表关联, 是自右向左解析的 而WHERE条件的解析顺序是自右向左的。也就是说, 在写SQL的时候, 尽量把数据量小的表放在最右边来进行关联(用小表去匹配大表)。WHERE后面条件AND 右边应尽量写为假的条件, OR右边应尽量写为真的条件, 这样SQL语句执行效率会高一些。

(三) 课后任务 (TODO 23个任务)

1. 任务一: 从雇员表中选出部门编号为30的员工信息
2. 任务二: 检索出工资大于2000小于5000的雇员信息
3. 任务三: 在条件中使用IN运算, 查询在部门编码为10,30的部门工作的员工。
4. 任务四: 显示姓名中有“ %N%”的雇员
5. 任务五: 显示姓名以S开头的的雇员
6. 任务六: 显示姓的第二个字母为“A”的的雇员
7. 任务七: 显示姓名中没有'L'字的员工的详细信息或含有'SM'字的员工信息
8. 任务八: 查出奖金为空的员工信息
9. 任务九: 查询部门编号为20, 并且工资少于1000的员工信息
10. 任务十: 查询姓名为“ALLEN”, 或者工作为“ANALYST”
11. 任务十一: 查出部门编号不为空的员工信息
12. 任务十二 查询职位(JOB)为'PRESIDENT'的员工的工资
13. 任务十三 查询佣金(COMM)为0或为NULL的员工信息

14. 任务十四 显示10号部门的所有经理('MANAGER')和20号部门的所有职员('CLERK')的详细信息
15. 任务十五 检索部门号及其本部门的最低工资，并按着最低工资降序排序。
16. 任务十六 列出最高薪水大于2500的各个部门人数
17. 任务十七 求1001号课成绩大于80分的学生的学号及成绩，并按成绩由高到低列出。
18. 任务十八 查询成绩在70~80分之间的学生选课得分情况
19. 任务十九 列出每门选修课的平均成绩。按着每门课平均成绩降序排列。
20. 任务二十 列出成绩不为空值的学生的学号和课号。
21. 任务二十一 统计每门课及格的人数
22. 任务二十二 查询至少有两门课成绩都大于80的学生号，统计课程数
23. 任务二十三 分页查询学生表第三页的学生记录，假设每页显示5条数据，总共有18条数据。

二 常用函数 (TODO 17个任务)

除了上面讲的分组聚合函数外，MySQL中还有很多内置的常用函数。MySQL中函数可以直接使用下面的语法：SELECT 函数名称(); ## Oracle不可以这样用

1. 数学函数

函数	作用
ABS(x)	返回 x 的绝对值
CEIL(x),CEILING(x)	返回大于或等于 x 的最小整数
FLOOR	返回小于或等于 x 的最大整数
RAND()	返回 0~1 的随机数
RAND(X)	返回 0~1 的随机数, x 值相同时返回的随机数相同
SIGN(X)	返回 x 的符号。x 是负数、0、正数分别返回-1,0 和 1
PI()	返回圆周率 (3.141593)
TRUNCATE(x,y)	返回数值 x 保留到小数点后 y 位的值
ROUND(x)	返回离 x 最近的整数
ROUND(x,y)	保留 x 小数点后 y 位的值, 但截断时要进行四舍五入
POW(x,y),POWER(x,y)	返回 x 的 y 次方(x^y)
SQRT(x)	返回 x 的平方根
EXP(x)	返回 e 的 x 次方(e^x)
MOD(x,y)	返回 x 除以 y 的余数
LOG(x)	返回自然对数(以 e 为底的对数)
LOG10(x)	返回以 10 为底的对数
RADIANS(x)	将角度转换成弧度
DEGREES(x)	将弧度转换成角度
SIN(x)	求正弦值
ASIN(x)	求反正弦值
COS(x)	求余弦值
ACOS(x)	求反余弦值
TAN(x)	求正切值
ATAN(x),ATAN2(x,y)	求反正切值
COT(x)	求余切值

☒ 任务1 对4.67四舍五入

```
SELECT ROUND(4.67) ;
```

注意: ROUND函数四舍五入。

☒ 任务2 取余操作 (取模)

```
SELECT MOD(17,8) ;
SELECT MOD(-17,8) ;
SELECT MOD(17,-8) ;
SELECT MOD(-17,-8) ;
```

注意：余数的结果的符号和被除数相同。

2. 字符串函数

函数	作用
ASCII(x)	返回 x 的 <u>ascii</u> 值
LENGTH(s)	返回字符串 s 的长度
CONCAT(s1,s2,...)	将字符串 s1,s2 等多个字符串合并为一个字符串
UPPER(s)	将字符串 s 的所有字母都变成大写字母
LOWER(s)	将字符串 s 的所有字母都变成小写字母
LEFT(s,n)	返回字符串 s 的前 n 个字符
RIGHT(s,n)	返回字符串 s 的后 n 个字符
LTRIM(s)	去掉字符串 s 开始处的空格
RTRIM(s)	去掉字符串 s 结尾处的空格
SUBSTRING(x,y,z)	取字符串 x 中 y 处开始的 z 个字符
TRIM(s)	去掉字符串两端的空格
REPLACE(s,s1,s2)	用字符串 s2 替代字符串 s 中的字符串 s1
STRCMP()	比较字符串的大小
FORMAT	将指定字符串按照格式和地区进行格式化

☑ 任务1 统计字符串hello world的长度

```
SELECT LENGTH('hello world');
```

☑ 任务2 统计员工姓名的长度。

```
SELECT ename,LENGTH(ename) FROM emp;
```

☑ 任务3 去掉字符串"hello world"两边的空格

```
SELECT TRIM('hello world');
```

☑ 任务4 将字符串"hello world"和字符串"MySQL"连接起来。

```
SELECT CONCAT('hello world','MySQL') FROM dual; --dual表示系统中的虚表
```

注意：MySQL中空值NULL连接字符串仍然是空值，例如：SELECT CONCAT(NULL,'MySQL');但是Oracle中空值NULL连接字符串结果为字符串，例如：SELECT CONCAT(NULL,'Oracle') FROM dual; 或者 SELECT NULL||'Oracle' FROM dual;

☑ 任务5 从字符串"hello world"中截取出字符串"world"

```
SELECT SUBSTRING('hello world',7,5);
```

注意：SUBSTRING函数截取从第7个字符开始截取后面的5个字符，索引从1开始。

3.日期和时间函数

函数	作用
NOW()	返回当前系统的日期时间
MONTH(d)	返回日期 d 中的月份值，范围是 1~12
DAYOFMONTH(date)	返回 date 的月份中的日期，在 1 到 31 范围内。
YEAR(d)	返回日期 d 中的年份值
WEEKDAY(date)	返回 date 的星期索引 (0=星期一，1=星期二，……6= 星期天)
DATE_ADD(date,INTERVAL <u>expr</u> type)	进行日期增加的操作，可以精确到秒
DATE_SUB(date,INTERVAL <u>expr</u> type)	进行日期减少的操作，可以精确到秒
CURRENT_TIME()	以 'HH:MM:SS' 或 HHMMSS 格式返回当前系统时间值
CURRENT_DATE()	返回当前系统日期
DAY(d)	返回日期 d 中获取日期的天

- ☒ 任务1 获取当前的系统时间

```
SELECT SYSDATE();  
或者  
SELECT NOW();
```

- ☒ 任务2 返回当前日期的年份，月份，日

```
SELECT YEAR(NOW()),MONTH(NOW()),DAY(NOW()) FROM dual;
```

- ☒ 任务3 查询当前日期的一个月以后的日期。

```
SELECT DATE_ADD(NOW(),INTERVAL 1 month);
```

4. 条件函数

IF(value,t f)	如果value是真, 返回t, 否则返回f
IFNULL(value1,value 2)	如果value1不为空返回value1, 否则返回value2
CASE WHEN [value1] THEN[result1]... ELSE[default] END	如果value1是真, 返回result1, 否则返回default
CASE [expr]WHEN[value1] THEN[result1]... ELSE[default] END	如果expr等于value1, 返回result1, 否则返回default

☑ 任务1 判断条件如果1小于2, 就输出yes,否则输出no。

```
SELECT IF(1<2,'yes ','no');
```

☑ 任务2 如果员工表为空,输出comm,,否则输出0。

```
SELECT comm,IFNULL(comm,0) FROM emp;
```

注意 IFNULL可以用于空值处理。这个与Oracle的NVL(exp1,exp2)类似。

☑ 任务3 查询所有员工的姓名，工资，年收入

```
SELECT ename,sal,sal*12+IFNULL(comm,0) FROM emp;
```

☑ 任务4 查询员工表员工涨前与涨后工资情况，根据职位涨工资，职位为PRESIDENT的涨1000，职位是经理的涨800，其他涨400。

```
SELECT empno,ename,job,sal 涨前,
CASE  WHEN job='PRESIDENT' THEN sal+1000
      WHEN job='MANAGER'   THEN sal+800
      ELSE sal+400
      END  涨后
FROM emp;
```

5. 系统信息函数

函数	作用
CONVERT(type,data)	用来转换数据类型
DATALength(data)	返回用于指定表示的字节数
HOST_NAME()	返回当前用户所登录的计算机名
SYSTEM_USER()	返回当前用户
CURRENT_USER()	
USER_NAME(1)	从任意数据中返回 “ <u>dbo</u> ”

☑ 任务1 查看当前用户

```
SELECT CURRENT_USER();
--或者
SELECT SYSTEM_USER();
```

6. 加密函数

函数	作用
PASSWORD(str)	实现对字符串 str 进行加密,这个加密过程是不可逆转的
MD5(str)	实现MD5加密，此加密算法不可逆
ENCODE(str,key)	使用 key 做为密钥加密
DECODE(str,key)	使用 key 做为密钥解密

☑ 任务1 对字符串进行MD5加密。(MD5加密不可逆)

```
SELECT MD5("root");
```

☒ 任务2 对字符串"hello"进行加密后再解密。

```
SELECT DECODE(ENCODE("hello","password"),"password");
```