MySQL day01课堂笔记

1、什么是数据库？什么是数据库管理系统？什么是SQL？他们之间的关系是什么？

数据库：

英文单词Database，简称DB。按照一定格式存储数据的一些文件的组合。
顾名思义：存储数据的仓库，实际上就是一堆文件。这些文件中存储了具有特定格式的数据。

数据库管理系统：

Database Management System，简称DBMS。
数据库管理系统是专门用来管理数据库中数据的，数据库管理系统可以对数据库当中的数据进行增删改查。

常见的数据库管理系统：

MySQL、 Oracle、 MS SqlServer、 DB2、 sybase等....

SQL：结构化查询语言:

程序员需要学习SQL语句，程序员通过编写SQL语句，然后DBMS负责执行SQL语句，最终来完成数据库中数据的增删改查操作。

SQL是一套标准，程序员主要学习的就是SQL语句，这个SQL在mysql中可以使用，同时在Oracle中也可以使用，在DB2中也可以使用。

三者之间的关系？

先安装数据库管理系统MySQL，然后学习SQL语句怎么写，编写SQL语句之后，DBMS对SQL语句进行执行，最终来完成数据库的数据管理。

2、安装MySQL数据库管理系统。

第一步：先安装，选择“经典版”
第二步：需要进行MySQL数据库实例配置。

注意：一路下一步就行了！！！！！

需要注意的事项?

- *端口号：*

端口号port是任何一个软件/应用都会有的，端口号是应用的唯一代表。
端口号通常和IP地址在一块，IP地址用来定位计算机的，端口号port是用来定位计算机上某个服务的/某个应用的！在同一台计算机上，端口号不能重复。具有唯一性。

mysql*数据库启动的时候，这个服务占有的默认端口号是* `3306`
这是大家都知道的事儿。记住。

- *字符编码方式?*

*设置*mysql*数据库的字符编码方式为* UTF8
*一定要注意：先选中第*3*个单选按钮，然后再选择*utf8*字符集。*

- *服务名称?*

*默认是：*MySQL
不用改。

- *选择配置环境变量*path*：*

如果没有选择怎么办？你可以手动配置
path=*其它路径*; `C:\Program Files (x86)\MySQL\MySQL Server 5.5\bin`

- mysql*超级管理员用户名不能改，一定是：*root

*你需要设置*mysql*数据库超级管理员的密码。*
*我们设置为*123456

*设置密码的同时，可以激活*root*账户远程访问。*
*激活：表示*root*账号可以在外地登录。*
*不激活：表示*root*账号只能在本机上使用。*
我这里选择激活了！

3、MySQL数据库的完美卸载！

第一步：双击安装包进行卸载删除。
第二步：删除目录：
把 `C:\ProgramData` 下面的MySQL目录干掉。
把 `C:\Program Files (x86)` 下面的MySQL目录干掉。

这样就卸载结束了！

4、看一下计算机上的服务，找一找MySQL的服务在哪里？

计算机-->右键-->管理-->服务和应用程序-->服务-->找mysql服务
MySQL的服务，默认是“启动”的状态，只有启动了mysql才能用。
默认情况下是“自动”启动，自动启动表示下一次重启操作系统的时候自动启动该服务。

可以在服务上点击右键：
启动
重启服务
停止服务
...

还可以改变服务的默认配置：
服务上点击右键，属性，然后可以选择启动方式：
自动（延迟启动）
自动
手动
禁用

5、在windows操作系统当中，怎么使用命令来启动和关闭mysql服务呢？

语法：
`net stop 服务名称`;
`net start 服务名称`;

其它服务的启停都可以采用以上的命令。

6、mysql安装了，服务启动了，怎么使用客户端登录mysql数据库呢？

使用bin目录下的mysql.exe命令来连接mysql数据库服务器

- *本地登录（显示编写密码的形式）：*

```
C:\Users\Administrator>mysql -uroot -p123456
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.5.36 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql>
```

本地登录（隐藏密码的形式）：

```
C:\Users\Administrator>mysql -uroot -p
Enter password: ******
```

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.5.36 MySQL Community Server (GPL)

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql>
```

7、mysql常用命令：

- 退出mysql： `exit`

- 查看mysql中有哪些数据库？

`show databases;`

注意：以分号结尾，分号是英文的分号。

```
mysql> show databases;
    +--------------------+
    | Database           |
    +--------------------+
    | information_schema |
    | mysql              |
    | performance_schema |
    | test               |
    +--------------------+
```

mysql默认自带了4个数据库。

- 怎么选择使用某个数据库呢？

```
mysql> use test;
    Database changed
```

表示正在使用一个名字叫做test的数据库。

- 怎么创建数据库呢？

```
 mysql> create database bjpowernode;
     Query OK, 1 row affected (0.00 sec)
 mysql> show databases;
     +--------------------+
     | Database           |
     +--------------------+
     | information_schema |
     | bjpowernode        |
     | mysql              |
     | performance_schema |
     | test               |
     +--------------------+
```

- *查看某个数据库下有哪些表？*

```
mysql> show tables;
```

注意：以上的命令不区分大小写，都行。

- *查看mysql数据库的版本号：*

```
mysql> select version();
    +-----------+
    | version() |
    +-----------+
    | 5.5.36    |
    +-----------+
```

- *查看当前使用的是哪个数据库？*

```
 mysql> select database();
     +-------------+
     | database()  |
     +-------------+
     | bjpowernode |
     +-------------+

mysql> show
    -> databases
    -> ;

     +--------------------+
     | Database           |
     +--------------------+
     | information_schema |
     | bjpowernode        |
     | mysql              |
     | performance_schema |
     | test               |
     +--------------------+
```

*注意：*mysql*是不见“;”不执行，“;”表示结束！*

```
mysql> show
    ->
    ->
    ->
    ->
    ->
    ->
    ->
    ->
    -> \c
    -- 上面的'->'为多次输入回车的结果
mysql>
```

`\c` 用来终止一条命令的输入。

8、数据库当中最基本的单元是表：table

- 什么是表table？为什么用表来存储数据呢？

姓名性别年龄(列：字段)

张三 男 20　　------->行（记录）
李四 女 21　　------->行（记录）
王五 男 22　　------->行（记录）

数据库当中是以表格的形式表示数据的。
因为表比较直观。

- 任何一张表都有行和列：

行（row）：被称为数据/记录。
列（column）：被称为字段。

姓名字段、性别字段、年龄字段。

- 了解一下：

每一个字段都有：字段名、数据类型、约束等属性。
字段名可以理解，是一个普通的名字，见名知意就行。

- 数据类型：字符串，数字，日期等，后期讲。

- 约束：约束也有很多，其中一个叫做唯一性约束，

这种约束添加之后，该字段中的数据不能重复。

9、关于SQL语句的分类？

SQL语句有很多，最好进行分门别类，这样更容易记忆。
分为：

- DQL：

数据查询语言（凡是带有select关键字的都是查询语句）
`select` ...

- DML：

数据操作语言（凡是对表当中的数据进行增删改的都是DML）
`insert` `delete` `update`
`insert` 增
`delete` 删
`update` 改

这个主要是操作表中的数据data。

- DDL：

数据定义语言
凡是带有 `create` 、 `drop` 、 `alter` 的都是DDL。
DDL主要操作的是表的结构。不是表中的数据。
`create` ：新建，等同于增
`drop` ：删除
`alter` ：修改
这个增删改和DML不同，这个主要是对表结构进行操作。

- TCL：

不是王牌电视。
是事务控制语言
包括：
事务提交： `commit` ;
事务回滚： `rollback` ;

- DCL：

是数据控制语言。
例如：授权 `grant` 、撤销权限 `revoke` ....

10、导入一下提前准备好的数据：

bjpowernode.sql 这个文件中是我提前为大家练习准备的数据库表。
怎么将sql文件中的数据导入呢？

```
mysql> source D:\course\03-MySQL\document\bjpowernode.sql
```

注意：路径中不要有中文！！！！

11、关于导入的这几张表？

```
mysql> show tables;
+----------------------+
| Tables_in_bjpowernode |
+----------------------+
| dept                 |
| emp                  |
| salgrade             |
+----------------------+
```

dept*是部门表*
emp*是员工表*
salgrade *是工资等级表*

怎么查看表中的数据呢？

```
select * from 表名;
-- 表名一般为英文格式，如为中文需要加''
-- 新版中默认中文为字符串，可以不加引号，但视频中安装的版本需要
```

*//统一执行这个*SQL*语句。*

```
mysql> select * from emp; -- 从emp表查询所有数据。
    +-------+--------+-----------+------+------------+---------+---------+---
-----+
    | EMPNO | ENAME  | JOB       | MGR  | HIREDATE   | SAL     | COMM    |
DEPTNO |
    +-------+--------+-----------+------+------------+---------+---------+---
-----+
    |  7369 | SMITH  | CLERK     | 7902 | 1980-12-17 |  800.00 |    NULL |
  20 |
    |  7499 | ALLEN  | SALESMAN  | 7698 | 1981-02-20 | 1600.00 |  300.00 |
  30 |
    |  7521 | WARD   | SALESMAN  | 7698 | 1981-02-22 | 1250.00 |  500.00 |
  30 |
    |  7566 | JONES  | MANAGER   | 7839 | 1981-04-02 | 2975.00 |    NULL |
  20 |
    |  7654 | MARTIN | SALESMAN  | 7698 | 1981-09-28 | 1250.00 | 1400.00 |
  30 |
```

```
|  7698 | BLAKE  | MANAGER   | 7839 | 1981-05-01 | 2850.00 |    NULL |
    30 |
|  7782 | CLARK  | MANAGER   | 7839 | 1981-06-09 | 2450.00 |    NULL |
    10 |
|  7788 | SCOTT  | ANALYST   | 7566 | 1987-04-19 | 3000.00 |    NULL |
    20 |
|  7839 | KING   | PRESIDENT | NULL | 1981-11-17 | 5000.00 |    NULL |
    10 |
|  7844 | TURNER | SALESMAN  | 7698 | 1981-09-08 | 1500.00 |    0.00 |
    30 |
|  7876 | ADAMS  | CLERK     | 7788 | 1987-05-23 | 1100.00 |    NULL |
    20 |
|  7900 | JAMES  | CLERK     | 7698 | 1981-12-03 |  950.00 |    NULL |
    30 |
|  7902 | FORD   | ANALYST   | 7566 | 1981-12-03 | 3000.00 |    NULL |
    20 |
|  7934 | MILLER | CLERK     | 7782 | 1982-01-23 | 1300.00 |    NULL |
    10 |
+-------+--------+-----------+------+------------+---------+---------+---
-----+

mysql> select * from dept;
+--------+------------+----------+
| DEPTNO | DNAME      | LOC      |
+--------+------------+----------+
|     10 | ACCOUNTING | NEW YORK |
|     20 | RESEARCH   | DALLAS   |
|     30 | SALES      | CHICAGO  |
|     40 | OPERATIONS | BOSTON   |
+--------+------------+----------+

mysql> select * from salgrade;
+-------+-------+-------+
| GRADE | LOSAL | HISAL |
+-------+-------+-------+
|     1 |   700 |  1200 |
|     2 |  1201 |  1400 |
|     3 |  1401 |  2000 |
|     4 |  2001 |  3000 |
|     5 |  3001 |  9999 |
+-------+-------+-------+
```

12、不看表中的数据，只看表的结构，有一个命令：

```
desc 表名;
```

```
mysql> desc dept;
```

```
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| DEPTNO | int(2)      | NO   | PRI | NULL    |       |部门编号
| DNAME  | varchar(14) | YES  |     | NULL    |       |部门名字
| LOC    | varchar(13) | YES  |     | NULL    |       |地理位置
+--------+-------------+------+-----+---------+-------+
mysql> desc emp;
+----------+-------------+------+-----+---------+-------+
| Field    | Type        | Null | Key | Default | Extra |
+----------+-------------+------+-----+---------+-------+
| EMPNO    | int(4)      | NO   | PRI | NULL    |       |员工编号
| ENAME    | varchar(10) | YES  |     | NULL    |       |员工姓名
| JOB      | varchar(9)  | YES  |     | NULL    |       |工作岗位
| MGR      | int(4)      | YES  |     | NULL    |       |上级编号
| HIREDATE | date        | YES  |     | NULL    |       |入职日期
| SAL      | double(7,2) | YES  |     | NULL    |       |工资
| COMM     | double(7,2) | YES  |     | NULL    |       |补助
| DEPTNO   | int(2)      | YES  |     | NULL    |       |部门编号
+----------+-------------+------+-----+---------+-------+
mysql> desc salgrade;
+-------+---------+------+-----+---------+-------+
| Field | Type    | Null | Key | Default | Extra |
+-------+---------+------+-----+---------+-------+
| GRADE | int(11) | YES  |     | NULL    |       |工资等级
| LOSAL | int(11) | YES  |     | NULL    |       |最低工资
| HISAL | int(11) | YES  |     | NULL    |       |最高工资
+-------+---------+------+-----+---------+-------+

-- describe缩写为: desc
mysql> describe dept;
+--------+-------------+------+-----+---------+-------+
| Field  | Type        | Null | Key | Default | Extra |
+--------+-------------+------+-----+---------+-------+
| DEPTNO | int(2)      | NO   | PRI | NULL    |       |
| DNAME  | varchar(14) | YES  |     | NULL    |       |
| LOC    | varchar(13) | YES  |     | NULL    |       |
+--------+-------------+------+-----+---------+-------+
```

13、简单查询

13.1、查询一个字段?

```
select 字段名 from 表名;
```

- *查询时需要注意：*

`select` *和* `from` *都是关键字。*
字段名和表名都是标识符。

- *强调：*

对于SQL语句来说，是通用的，
所有的SQL语句以“;”结尾。
另外SQL语句不区分大小写，都行。

- *查询部门名字？*

```
mysql> select dname from dept;
    +------------+
    | dname      |
    +------------+
    | ACCOUNTING |
    | RESEARCH   |
    | SALES      |
    | OPERATIONS |
    +------------+
    4 rows in set (0.00 sec)

mysql> SELECT DNAME FROM DEPT;
    +------------+
    | DNAME      |
    +------------+
    | ACCOUNTING |
    | RESEARCH   |
    | SALES      |
    | OPERATIONS |
    +------------+
    4 rows in set (0.00 sec)
```

13.2、查询两个字段，或者多个字段怎么办？

使用逗号隔开“,”

- *查询部门编号和部门名？*

```
select deptno,dname from dept;
    +--------+------------+
    | deptno | dname      |
    +--------+------------+
    |     10 | ACCOUNTING |
    |     20 | RESEARCH   |
    |     30 | SALES      |
    |     40 | OPERATIONS |
    +--------+------------+
```

13.3、查询所有字段怎么办？

第一种方式：可以把每个字段都写上

```
select a,b,c,d,e,f... from tablename;
```

第二种方式：可以使用*

```
select * from dept;
+--------+------------+----------+
| DEPTNO | DNAME      | LOC      |
+--------+------------+----------+
|     10 | ACCOUNTING | NEW YORK |
|     20 | RESEARCH   | DALLAS   |
|     30 | SALES      | CHICAGO  |
|     40 | OPERATIONS | BOSTON   |
+--------+------------+----------+
```

这种方式的缺点：

1、*效率低*

2、*可读性差。*

在实际开发中不建议，可以自己玩没问题。

你可以在DOS命令窗口中想快速的看一看全表数据可以采用这种方式。

13.4、给查询的列起别名？

```
mysql> select deptno,dname as deptname from dept;
    +--------+------------+
    | deptno | deptname   |
    +--------+------------+
    |     10 | ACCOUNTING |
    |     20 | RESEARCH   |
    |     30 | SALES      |
    |     40 | OPERATIONS |
    +--------+------------+
```

*使用*as*关键字起别名。*
*注意：只是将显示的查询结果列名显示为*deptname*，原表列名还是叫：*dname
*记住：*select*语句是永远都不会进行修改操作的。（因为只负责查询）*

as*关键字可以省略吗？可以的*

```
mysql> select deptno,dname deptname from dept;
```

假设起别名的时候，别名里面有空格，怎么办？

```
mysql> select deptno,dname dept name from dept;
```

DBMS*看到这样的语句，进行*SQL*语句的编译，不符合语法，编译报错。*
怎么解决？

```
 select deptno,dname 'dept name' from dept; //加单引号
 select deptno,dname "dept name" from dept; //加双引号
    +--------+------------+
    | deptno | dept name  |
    +--------+------------+
    |     10 | ACCOUNTING |
    |     20 | RESEARCH   |
    |     30 | SALES      |
    |     40 | OPERATIONS |
    +--------+------------+
```

注意：在所有的数据库当中，字符串统一使用单引号括起来，
*单引号是标准，双引号在*oracle*数据库中用不了。但是在*mysql
中可以使用。

再次强调：数据库中的字符串都是采用单引号括起来。这是标准的。
双引号不标准。

- **13.5、*计算员工年薪？* sal * 12**

```
mysql> select ename,sal from emp;
    +--------+---------+
    | ename  | sal     |
    +--------+---------+
    | SMITH  |  800.00 |
    | ALLEN  | 1600.00 |
    | WARD   | 1250.00 |
    | JONES  | 2975.00 |
    | MARTIN | 1250.00 |
    | BLAKE  | 2850.00 |
    | CLARK  | 2450.00 |
    | SCOTT  | 3000.00 |
    | KING   | 5000.00 |
    | TURNER | 1500.00 |
    | ADAMS  | 1100.00 |
    | JAMES  |  950.00 |
    | FORD   | 3000.00 |
    | MILLER | 1300.00 |
    +--------+---------+
mysql> select ename,sal*12 from emp; // 结论：字段可以使用数学表达式!
    +--------+----------+
    | ename  | sal*12   |
    +--------+----------+
    | SMITH  |  9600.00 |
    | ALLEN  | 19200.00 |
    | WARD   | 15000.00 |
    | JONES  | 35700.00 |
    | MARTIN | 15000.00 |
    | BLAKE  | 34200.00 |
    | CLARK  | 29400.00 |
    | SCOTT  | 36000.00 |
    | KING   | 60000.00 |
    | TURNER | 18000.00 |
    | ADAMS  | 13200.00 |
    | JAMES  | 11400.00 |
    | FORD   | 36000.00 |
    | MILLER | 15600.00 |
    +--------+----------+

mysql> select ename,sal*12 as yearsal from emp; //起别名
    +--------+----------+
    | ename  | yearsal  |
    +--------+----------+
    | SMITH  |  9600.00 |
    | ALLEN  | 19200.00 |
    | WARD   | 15000.00 |
    | JONES  | 35700.00 |
    | MARTIN | 15000.00 |
    | BLAKE  | 34200.00 |
```

```
| CLARK  | 29400.00 |
| SCOTT  | 36000.00 |
| KING   | 60000.00 |
| TURNER | 18000.00 |
| ADAMS  | 13200.00 |
| JAMES  | 11400.00 |
| FORD   | 36000.00 |
| MILLER | 15600.00 |
+--------+----------+

mysql> select ename,sal*12 as '年薪' from emp; //别名是中文，用单引号括起来。
+--------+----------+
| ename  | 年薪     |
+--------+----------+
| SMITH  |  9600.00 |
| ALLEN  | 19200.00 |
| WARD   | 15000.00 |
| JONES  | 35700.00 |
| MARTIN | 15000.00 |
| BLAKE  | 34200.00 |
| CLARK  | 29400.00 |
| SCOTT  | 36000.00 |
| KING   | 60000.00 |
| TURNER | 18000.00 |
| ADAMS  | 13200.00 |
| JAMES  | 11400.00 |
| FORD   | 36000.00 |
| MILLER | 15600.00 |
+--------+----------+
```

14、条件查询

14.1、什么是条件查询?

不是将表中所有数据都查出来。是查询出来符合条件的。

语法格式：

```
select
字段1,字段2,字段3....
from
表名
where
条件;
```

14.2、都有哪些条件？

= 等于

查询薪资等于800的员工姓名和编号？

```
select empno,ename from emp where sal = 800;
```

查询SMITH的编号和薪资？

```
select empno,sal from emp where ename = 'SMITH'; //字符串使用单引号
```

<>或!= 不等于

查询薪资不等于800的员工姓名和编号？

select empno,ename from emp where sal != 800;

select empno,ename from emp where sal <> 800; // *小于号和大于号组成的不等号*

< 小于

查询薪资小于2000的员工姓名和编号？

```
mysql> select empno,ename,sal from emp where sal < 2000;
    +-------+--------+---------+
    | empno | ename  | sal     |
    +-------+--------+---------+
    |  7369 | SMITH  |  800.00 |
    |  7499 | ALLEN  | 1600.00 |
    |  7521 | WARD   | 1250.00 |
    |  7654 | MARTIN | 1250.00 |
    |  7844 | TURNER | 1500.00 |
    |  7876 | ADAMS  | 1100.00 |
    |  7900 | JAMES  |  950.00 |
    |  7934 | MILLER | 1300.00 |
    +-------+--------+---------+
```

<= 小于等于

查询薪资小于等于3000的员工姓名和编号？

```
select empno,ename,sal from emp where sal <= 3000;
```

- >大于

查询薪资大于3000的员工姓名和编号?

```
select empno,ename,sal from emp where sal > 3000;
```

- >= 大于等于

查询薪资大于等于3000的员工姓名和编号?

```
select empno,ename,sal from emp where sal >= 3000;
```

- `between ... and ....` 两个值之间,等同于 >= and <=

查询薪资在2450和3000之间的员工信息?包括2450和3000
第一种方式: >= and <= （ `and` *是并且的意思。*）

```
select empno,ename,sal from emp where sal >= 2450 and sal <= 3000;
    +-------+-------+---------+
    | empno | ename | sal     |
    +-------+-------+---------+
    |  7566 | JONES | 2975.00 |
    |  7698 | BLAKE | 2850.00 |
    |  7782 | CLARK | 2450.00 |
    |  7788 | SCOTT | 3000.00 |
    |  7902 | FORD  | 3000.00 |
    +-------+-------+---------+
```

第二种方式: `between ... and ...`

```
select
empno,ename,sal
from
emp
where
sal between 2450 and 3000;
```

注意:
*使用*between and*的时候,必须遵循左小右大。*
between and*是闭区间,包括两端的值。*

- is null 为 null（is not null 不为空）

查询哪些员工的津贴/补助为null?

```
mysql> select empno,ename,sal,comm from emp where comm = null;
    Empty set (0.00 sec)

mysql> select empno,ename,sal,comm from emp where comm is null;
    +-------+--------+---------+------+
    | empno | ename  | sal     | comm |
    +-------+--------+---------+------+
    |  7369 | SMITH  |  800.00 | NULL |
    |  7566 | JONES  | 2975.00 | NULL |
    |  7698 | BLAKE  | 2850.00 | NULL |
    |  7782 | CLARK  | 2450.00 | NULL |
    |  7788 | SCOTT  | 3000.00 | NULL |
    |  7839 | KING   | 5000.00 | NULL |
    |  7876 | ADAMS  | 1100.00 | NULL |
    |  7900 | JAMES  |  950.00 | NULL |
    |  7902 | FORD   | 3000.00 | NULL |
    |  7934 | MILLER | 1300.00 | NULL |
    +-------+--------+---------+------+
    10 rows in set (0.00 sec)
```

注意：在数据库当中null不能使用等号进行衡量。需要使用is null
因为数据库中的null代表什么也没有，它不是一个值，所以不能使用
等号衡量。

查询哪些员工的津贴/补助不为null?

```
select empno,ename,sal,comm from emp where comm is not null;
    +-------+--------+---------+---------+
    | empno | ename  | sal     | comm    |
    +-------+--------+---------+---------+
    |  7499 | ALLEN  | 1600.00 |  300.00 |
    |  7521 | WARD   | 1250.00 |  500.00 |
    |  7654 | MARTIN | 1250.00 | 1400.00 |
    |  7844 | TURNER | 1500.00 |    0.00 |
    +-------+--------+---------+---------+
```

- and 并且

查询工作岗位是MANAGER并且工资大于2500的员工信息?

```
select
empno,ename,job,sal
from
emp
where
job = 'MANAGER' and sal > 2500;

    +-------+-------+---------+---------+
    | empno | ename | job     | sal     |
    +-------+-------+---------+---------+
    |  7566 | JONES | MANAGER | 2975.00 |
    |  7698 | BLAKE | MANAGER | 2850.00 |
    +-------+-------+---------+---------+
```

- or *或者*

*查询工作岗位是*MANAGER*和*SALESMAN*的员工?*

```
select empno,ename,job from emp where job = 'MANAGER';
select empno,ename,job from emp where job = 'SALESMAN';

select
empno,ename,job
from
emp
where
job = 'MANAGER' or job = 'SALESMAN';

    +-------+--------+----------+
    | empno | ename  | job      |
    +-------+--------+----------+
    |  7499 | ALLEN  | SALESMAN |
    |  7521 | WARD   | SALESMAN |
    |  7566 | JONES  | MANAGER  |
    |  7654 | MARTIN | SALESMAN |
    |  7698 | BLAKE  | MANAGER  |
    |  7782 | CLARK  | MANAGER  |
    |  7844 | TURNER | SALESMAN |
    +-------+--------+----------+
```

- and和or同时出现的话，*有优先级问题吗?*

*查询工资大于*2500*，并且部门编号为*10*或*20*部门的员工?*

```
select
*
from
emp
where
sal > 2500 and deptno = 10 or deptno = 20;
```

分析以上语句的问题?

and*优先级比*or*高。*

*以上语句会先执行*and*，然后执行*or*。*

以上这个语句表示什么含义?

*找出工资大于*2500*并且部门编号为*10*的员工，或者*20*部门所有员工找出来。*

```
select
*
from
emp
where
sal > 2500 and (deptno = 10 or deptno = 20);
```

and*和*or*同时出现，*and*优先级较高。如果想让*or*先执行，需要加“小括号”*

以后在开发中，如果不确定优先级，就加小括号就行了。

- in 包含，相当于多个 or （not in 不在这个范围中）

*查询工作岗位是*MANAGER*和*SALESMAN*的员工?*

```
 select empno,ename,job from emp where job = 'MANAGER' or job = 'SALESMAN';
 select empno,ename,job from emp where job in('MANAGER', 'SALESMAN');
     +-------+--------+----------+
     | empno | ename  | job      |
     +-------+--------+----------+
     |  7499 | ALLEN  | SALESMAN |
     |  7521 | WARD   | SALESMAN |
     |  7566 | JONES  | MANAGER  |
     |  7654 | MARTIN | SALESMAN |
     |  7698 | BLAKE  | MANAGER  |
     |  7782 | CLARK  | MANAGER  |
     |  7844 | TURNER | SALESMAN |
     +-------+--------+----------+
```

*注意：*in*不是一个区间。*in*后面跟的是具体的值。*

- *查询薪资是800和5000的员工信息？*

```
select ename,sal from emp where sal = 800 or sal = 5000;
select ename,sal from emp where sal in(800, 5000); //这个不是表示800到5000都找出来。
+-------+---------+
| ename | sal     |
+-------+---------+
| SMITH |  800.00 |
| KING  | 5000.00 |
+-------+---------+
select ename,sal from emp where sal in(800, 5000, 3000);

-- not in 表示不在这几个值当中的数据。

select ename,sal from emp where sal not in(800, 5000, 3000);
+--------+---------+
| ename  | sal     |
+--------+---------+
| ALLEN  | 1600.00 |
| WARD   | 1250.00 |
| JONES  | 2975.00 |
| MARTIN | 1250.00 |
| BLAKE  | 2850.00 |
| CLARK  | 2450.00 |
| TURNER | 1500.00 |
| ADAMS  | 1100.00 |
| JAMES  |  950.00 |
| MILLER | 1300.00 |
+--------+---------+
```

not *可以取非，主要用在* is *或* in *中*

`is null`

`is not null`

`in`

`not in`

- **like** *称为模糊查询*

支持%或下划线匹配

%匹配任意多个字符

下划线：任意一个字符。

（%是一个特殊的符号，_也是一个特殊符号）

找出名字中含有O的?

```
mysql> select ename from emp where ename like '%O%';
    +-------+
    | ename |
    +-------+
    | JONES |
    | SCOTT |
    | FORD  |
    +-------+
```

找出名字以T结尾的?

```
select ename from emp where ename like '%T';
```

找出名字以K开始的?

```
select ename from emp where ename like 'K%';
```

找出第二个字每是A的?

```
select ename from emp where ename like '_A%';
```

找出第三个字母是R的?

```
select ename from emp where ename like '__R%';
```

```
# t_student学生表

name

zhangsan
lisi
wangwu
zhaoliu
jack_son
```

找出名字中有“

```
select name from t_student where name like '%_%'; //这样不行。

mysql> select name from t_student where name like '%\_%'; // \转义字符。
    +----------+
    | name     |
    +----------+
    | jack_son |
    +----------+
```

15、*排序*

- **15.1、*查询所有员工薪资，排序?***

```
select
ename,sal
from
emp
order by
sal; // 默认是升序!!!

    +--------+---------+
    | ename  | sal     |
    +--------+---------+
    | SMITH  |  800.00 |
    | JAMES  |  950.00 |
    | ADAMS  | 1100.00 |
    | WARD   | 1250.00 |
    | MARTIN | 1250.00 |
    | MILLER | 1300.00 |
    | TURNER | 1500.00 |
    | ALLEN  | 1600.00 |
    | CLARK  | 2450.00 |
    | BLAKE  | 2850.00 |
    | JONES  | 2975.00 |
    | FORD   | 3000.00 |
    | SCOTT  | 3000.00 |
    | KING   | 5000.00 |
    +--------+---------+
```

- **15.2、*怎么降序?***

- *指定降序：*

```
select
ename,sal
from
emp
order by
sal desc;

    +--------+---------+
    | ename  | sal     |
    +--------+---------+
    | KING   | 5000.00 |
    | SCOTT  | 3000.00 |
    | FORD   | 3000.00 |
    | JONES  | 2975.00 |
    | BLAKE  | 2850.00 |
    | CLARK  | 2450.00 |
    | ALLEN  | 1600.00 |
    | TURNER | 1500.00 |
    | MILLER | 1300.00 |
    | MARTIN | 1250.00 |
    | WARD   | 1250.00 |
    | ADAMS  | 1100.00 |
    | JAMES  |  950.00 |
    | SMITH  |  800.00 |
    +--------+---------+
```

- *指定升序?*

```
select
ename,sal
from
emp
order by
sal asc;

    +--------+---------+
    | ename  | sal     |
    +--------+---------+
    | SMITH  |  800.00 |
    | JAMES  |  950.00 |
    | ADAMS  | 1100.00 |
    | WARD   | 1250.00 |
    | MARTIN | 1250.00 |
    | MILLER | 1300.00 |
    | TURNER | 1500.00 |
    | ALLEN  | 1600.00 |
    | CLARK  | 2450.00 |
```

```
| BLAKE  | 2850.00 |
| JONES  | 2975.00 |
| FORD   | 3000.00 |
| SCOTT  | 3000.00 |
| KING   | 5000.00 |
+--------+---------+
```

• 15.3、可以两个字段排序吗？或者说按照多个字段排序？

查询员工名字和薪资，要求按照薪资升序，如果薪资一样的话，
再按照名字升序排列。

```
select
ename,sal
from
emp
order by
sal asc, ename asc; // sal在前，起主导，只有sal相等的时候，才会考虑启用ename排序。

    +--------+---------+
    | ename  | sal     |
    +--------+---------+
    | SMITH  |  800.00 |
    | JAMES  |  950.00 |
    | ADAMS  | 1100.00 |
    | MARTIN | 1250.00 |
    | WARD   | 1250.00 |
    | MILLER | 1300.00 |
    | TURNER | 1500.00 |
    | ALLEN  | 1600.00 |
    | CLARK  | 2450.00 |
    | BLAKE  | 2850.00 |
    | JONES  | 2975.00 |
    | FORD   | 3000.00 |
    | SCOTT  | 3000.00 |
    | KING   | 5000.00 |
    +--------+---------+
```

• 15.4、了解：根据字段的位置也可以排序

```
select ename,sal from emp order by 2; -- 2表示第二列。第二列是sal
```

按照查询结果的第2列sal排序。

了解一下，不建议在开发中这样写，因为不健壮。
因为列的顺序很容易发生改变，列顺序修改之后，2就废了。

16、综合一点的案例：

找出工资在1250到3000之间的员工信息，要求按照薪资降序排列。

```
select
ename,sal
from
emp
where
sal between 1250 and 3000
order by
sal desc;

    +--------+---------+
    | ename  | sal     |
    +--------+---------+
    | FORD   | 3000.00 |
    | SCOTT  | 3000.00 |
    | JONES  | 2975.00 |
    | BLAKE  | 2850.00 |
    | CLARK  | 2450.00 |
    | ALLEN  | 1600.00 |
    | TURNER | 1500.00 |
    | MILLER | 1300.00 |
    | MARTIN | 1250.00 |
    | WARD   | 1250.00 |
    +--------+---------+
```

关键字顺序不能变：

```
select
...
from
...
where
...
order by
...;
```

以上语句的执行顺序必须掌握：

*第一步：*from

*第二步：*where

*第三步：*select

*第四步：*order by（*排序总是在最后执行！*）

17、数据处理函数

17.1、数据处理函数又被称为单行处理函数

单行处理函数的特点：一个输入对应一个输出。

和单行处理函数相对的是：多行处理函数。（多行处理函数特点：多个输入，对应1个输出！）

17.2、单行处理函数常见的有哪些？

lower 转换小写

```
mysql> select lower(ename) as ename from emp;
    +--------+
    | ename  |
    +--------+
    | smith  |
    | allen  |
    | ward   |
    | jones  |
    | martin |
    | blake  |
    | clark  |
    | scott  |
    | king   |
    | turner |
    | adams  |
    | james  |
    | ford   |
    | miller |
    +--------+
```

14个输入，最后还是14个输出。这是单行处理函数的特点。

upper 转换大写

```
mysql> select * from t_student;
    +----------+
    | name     |
    +----------+
    | zhangsan |
    | lisi     |
    | wangwu   |
    | jack_son |
    +----------+

mysql> select upper(name) as name from t_student;
```

```
+----------+
| name     |
+----------+
| ZHANGSAN |
| LISI     |
| WANGWU   |
| JACK_SON |
+----------+
```

- substr 取子串（substr(被截取的字符串,起始下标,截取的长度)）

```
select substr(ename, 1, 1) as ename from emp;
+-------+
| ename |
+-------+
| S     |
| A     |
| W     |
| J     |
| M     |
| B     |
| C     |
| S     |
| K     |
| T     |
| A     |
| J     |
| F     |
| M     |
+-------+`
```

注意：起始下标从1开始，没有0.

- 找出员工名字第一个字母是A的员工信息？

第一种方式：模糊查询

```
select ename from emp where ename like 'A%';
```

第二种方式：substr函数

```
select
ename
from
emp
where
substr(ename,1,1) = 'A';
```

首字母大写?

```
select name from t_student; -- 取出名字进行观察
select upper(substr(name,1,1)) from t_student; -- 只保留首字母，并将其转为大写
select substr(name,2,length(name) - 1) from t_student; -- 只保留首字母以外的字符串
select concat(upper(substr(name,1,1)),substr(name,2,length(name) - 1)) as result from t_student;
-- 两段字符串合为一段

    +----------+
    | result   |
    +----------+
    | Zhangsan |
    | Lisi     |
    | Wangwu   |
    | Jack_son |
    +----------+
```

- **concat*函数进行字符串的拼接***

```
select concat(empno,ename) from emp;
    +---------------------+
    | concat(empno,ename) |
    +---------------------+
    | 7369SMITH           |
    | 7499ALLEN           |
    | 7521WARD            |
    | 7566JONES           |
    | 7654MARTIN          |
    | 7698BLAKE           |
    | 7782CLARK           |
    | 7788SCOTT           |
    | 7839KING            |
    | 7844TURNER          |
    | 7876ADAMS           |
    | 7900JAMES           |
    | 7902FORD            |
    | 7934MILLER          |
    +---------------------+
```

- length 取长度

```
select length(ename) enamelength from emp;
    +-------------+
    | enamelength |
    +-------------+
    |           5 |
    |           5 |
    |           4 |
    |           5 |
    |           6 |
    |           5 |
    |           5 |
    |           5 |
    |           4 |
    |           6 |
    |           5 |
    |           5 |
    |           4 |
    |           6 |
    +-------------+
```

- trim 去空格

```
mysql> select * from emp where ename = '  KING';
    Empty set (0.00 sec)

mysql> select * from emp where ename = trim('   KING');
    +-------+-------+-----------+------+------------+---------+------+--------
+
    | EMPNO | ENAME | JOB       | MGR  | HIREDATE   | SAL     | COMM | DEPTNO
|
    +-------+-------+-----------+------+------------+---------+------+--------
+
    |  7839 | KING  | PRESIDENT | NULL | 1981-11-17 | 5000.00 | NULL |     10
|
    +-------+-------+-----------+------+------------+---------+------+--------
+
```

`str_to_date` *将字符串转换成日期*
`date_format` *格式化日期*
`format` *设置千分位*

- case..when..then..when..then..else..end

*当员工的工作岗位是*MANAGER*的时候，工资上调*10%*，当工作岗位是*SALESMAN*的时候，工资上调*50%*,其它正常。*

（注意：不修改数据库，只是将查询结果显示为工资上调）

```
select
ename,
job,
sal as oldsal,
(case job when 'MANAGER' then sal*1.1 when 'SALESMAN' then sal*1.5 else sal
end) as newsal
from
emp;

    +--------+-----------+---------+---------+
    | ename  | job       | oldsal  | newsal  |
    +--------+-----------+---------+---------+
    | SMITH  | CLERK     |  800.00 |  800.00 |
    | ALLEN  | SALESMAN  | 1600.00 | 2400.00 |
    | WARD   | SALESMAN  | 1250.00 | 1875.00 |
    | JONES  | MANAGER   | 2975.00 | 3272.50 |
    | MARTIN | SALESMAN  | 1250.00 | 1875.00 |
    | BLAKE  | MANAGER   | 2850.00 | 3135.00 |
    | CLARK  | MANAGER   | 2450.00 | 2695.00 |
    | SCOTT  | ANALYST   | 3000.00 | 3000.00 |
    | KING   | PRESIDENT | 5000.00 | 5000.00 |
    | TURNER | SALESMAN  | 1500.00 | 2250.00 |
    | ADAMS  | CLERK     | 1100.00 | 1100.00 |
    | JAMES  | CLERK     |  950.00 |  950.00 |
    | FORD   | ANALYST   | 3000.00 | 3000.00 |
    | MILLER | CLERK     | 1300.00 | 1300.00 |
    +--------+-----------+---------+---------+
```

- round *四舍五入*

> “
>
> *引入一种之间表示数字的方式，便于讲解* `round` *函数。*

```
select 字段 from 表名;
```

```
select ename from emp;
select 'abc' from emp; // select后面直接跟“字面量/字面值”

mysql> select 'abc' as bieming from emp;
    +---------+
```

```
| bieming |
+---------+
| abc     |
| abc     |
| abc     |
| abc     |
| abc     |
| abc     |
| abc     |
| abc     |
| abc     |
| abc     |
| abc     |
| abc     |
| abc     |
| abc     |
+---------+
```

```
mysql> select abc from emp;
    ERROR 1054 (42S22): Unknown column 'abc' in 'field list'
```

*这样肯定报错，因为会把*abc*当做一个字段的名字，去*emp*表中找*abc*字段去了。*

```
select 1000 as num from emp; // 1000 也是被当做一个字面量/字面值。
    +------+
    | num  |
    +------+
    | 1000 |
    | 1000 |
    | 1000 |
    | 1000 |
    | 1000 |
    | 1000 |
    | 1000 |
    | 1000 |
    | 1000 |
    | 1000 |
    | 1000 |
    | 1000 |
    | 1000 |
    | 1000 |
    | 1000 |
    +------+
```

*结论：*select*后面可以跟某个表的字段名（可以等同看做变量名），也可以跟字面量/字面值（数据）。其中，表的行数与原表行数相同。*

```
select 21000 as num from dept;
```

```
    +-------+
    | num   |
    +-------+
    | 21000 |
    | 21000 |
    | 21000 |
    | 21000 |
    +-------+

mysql> select round(1236.567, 0) as result from emp; //保留整数位。
    +--------+
    | result |
    +--------+
    |   1237 |
    |   1237 |
    |   1237 |
    |   1237 |
    |   1237 |
    |   1237 |
    |   1237 |
    |   1237 |
    |   1237 |
    |   1237 |
    |   1237 |
    |   1237 |
    |   1237 |
    |   1237 |
    +--------+

select round(1236.567, 1) as result from emp; //保留1个小数
select round(1236.567, 2) as result from emp; //保留2个小数
select round(1236.567, -1) as result from emp; // 保留到十位。
    +--------+
    | result |
    +--------+
    |   1240 |
    |   1240 |
    |   1240 |
    |   1240 |
    |   1240 |
    |   1240 |
    |   1240 |
    |   1240 |
    |   1240 |
    |   1240 |
    |   1240 |
    |   1240 |
    |   1240 |
    |   1240 |
    +--------+

select round(1236.567, -2) as result from emp;
```

```
+--------+
| result |
+--------+
|   1200 |
|   1200 |
|   1200 |
|   1200 |
|   1200 |
|   1200 |
|   1200 |
|   1200 |
|   1200 |
|   1200 |
|   1200 |
|   1200 |
|   1200 |
|   1200 |
+--------+
```

rand() *生成随机数*

与round配合使用可控制位数。

```
mysql> select rand() from emp; -- 无round函数时的情况
    +---------------------+
    | rand()              |
    +---------------------+
    |  0.4808966177337771 |
    | 0.26691885224256556 |
    |  0.8919044387451415 |
    |  0.6587656677316555 |
    |   0.618115053156498 |
    | 0.11427829332116832 |
    |  0.7170463378699928 |
    | 0.24239684012010398 |
    | 0.06084521772418657 |
    |  0.5770355989942658 |
    |  0.7026423725324146 |
    |  0.7821050964129205 |
    |  0.8025983952010035 |
    |  0.6666767174999013 |
    +---------------------+

mysql> select round(rand()*100,0) from emp; // 100以内的随机数
+---------------------+
| round(rand()*100,0) |
    +---------------------+
    |                  76 |
    |                  29 |
    |                  15 |
```

```
|                   88 |
|                   95 |
|                    9 |
|                   63 |
|                   89 |
|                   54 |
|                    3 |
|                   54 |
|                   61 |
|                   42 |
|                   28 |
+----------------------+
```

- **ifnull *可以将* null *转换成一个具体值***

ifnull*是空处理函数。专门处理空的。*

*在所有数据库当中，只要有*NULL*参与的数学运算，最终结果就是*NULL*。*

```
mysql> select ename, sal + comm as salcomm from emp;
+--------+---------+
| ename  | salcomm |
+--------+---------+
| SMITH  |    NULL |
| ALLEN  | 1900.00 |
| WARD   | 1750.00 |
| JONES  |    NULL |
| MARTIN | 2650.00 |
| BLAKE  |    NULL |
| CLARK  |    NULL |
| SCOTT  |    NULL |
| KING   |    NULL |
| TURNER | 1500.00 |
| ADAMS  |    NULL |
| JAMES  |    NULL |
| FORD   |    NULL |
| MILLER |    NULL |
+--------+---------+
```

计算每个员工的年薪?

年薪 = (*月薪* + *月补助*) * 12

```
select ename, (sal + comm) * 12 as yearsal from emp;
+--------+----------+
| ename  | yearsal  |
+--------+----------+
| SMITH  |     NULL |
| ALLEN  | 22800.00 |
| WARD   | 21000.00 |
```

```
| JONES  |     NULL |
| MARTIN | 31800.00 |
| BLAKE  |     NULL |
| CLARK  |     NULL |
| SCOTT  |     NULL |
| KING   |     NULL |
| TURNER | 18000.00 |
| ADAMS  |     NULL |
| JAMES  |     NULL |
| FORD   |     NULL |
| MILLER |     NULL |
+--------+----------+
```

*注意：*NULL*只要参与运算，最终结果一定是*NULL*。为了避免这个现象，需要使用*ifnull*函数。*

ifnull*函数用法：*ifnull(*数据, 被当做哪个值*)

*如果“数据”为*NULL*的时候，把这个数据结构当做哪个值。*

*补助为*NULL*的时候，将补助当做*0

```
select ename, (sal + ifnull(comm, 0)) * 12 as yearsal from emp;
+--------+----------+
| ename  | yearsal  |
+--------+----------+
| SMITH  |  9600.00 |
| ALLEN  | 22800.00 |
| WARD   | 21000.00 |
| JONES  | 35700.00 |
| MARTIN | 31800.00 |
| BLAKE  | 34200.00 |
| CLARK  | 29400.00 |
| SCOTT  | 36000.00 |
| KING   | 60000.00 |
| TURNER | 18000.00 |
| ADAMS  | 13200.00 |
| JAMES  | 11400.00 |
| FORD   | 36000.00 |
| MILLER | 15600.00 |
+--------+----------+
```

18、*分组函数（多行处理函数）*

多行处理函数的特点：输入多行，最终输出一行。

5*个：*

count	*计数*
sum	*求和*
avg	*平均值*
max	*最大值*
min	*最小值*

注意：
分组函数在使用的时候必须先进行分组，然后才能用。
如果你没有对数据进行分组，整张表默认为一组。

找出最高工资?

```
mysql> select max(sal) from emp;
    +----------+
    | max(sal) |
    +----------+
    |  5000.00 |
    +----------+
```

找出最低工资?

```
mysql> select min(sal) from emp;
    +----------+
    | min(sal) |
    +----------+
    |   800.00 |
    +----------+
```

计算工资和：

```
mysql> select sum(sal) from emp;
    +----------+
    | sum(sal) |
    +----------+
    | 29025.00 |
    +----------+
```

计算平均工资：

```
mysql> select avg(sal) from emp;
    +-------------+
    | avg(sal)    |
    +-------------+
    | 2073.214286 |
    +-------------+
```

14个工资全部加起来，然后除以14。

计算员工数量？

```
mysql> select count(ename) from emp;
    +--------------+
    | count(ename) |
    +--------------+
    |           14 |
    +--------------+
```

- 分组函数在使用的时候需要注意哪些？

第一点：分组函数自动忽略NULL，你不需要提前对NULL进行处理。

```
 mysql> select sum(comm) from emp;
     +-----------+
     | sum(comm) |
     +-----------+
     |   2200.00 |
     +-----------+

 mysql> select count(comm) from emp;
     +-------------+
     | count(comm) |
     +-------------+
     |           4 |
     +-------------+

 mysql> select avg(comm) from emp;
     +------------+
     | avg(comm)  |
     +------------+
     | 550.000000 |
     +------------+
```

第二点：分组函数中count(

```
mysql> select count(*) from emp;
    +----------+
    | count(*) |
    +----------+
    |       14 |
    +----------+

mysql> select count(comm) from emp;
    +-------------+
    | count(comm) |
    +-------------+
    |           4 |
    +-------------+
```

count(*具体字段*)：*表示统计该字段下所有不为*NULL*的元素的总数。*
count(*)：*统计表当中的总行数。（只要有一行数据*count*则++）*
*因为每一行记录不可能都为*NULL*，一行数据中有一列不为*NULL*，则这行数据就是有效的。*

*第三点：分组函数不能够直接使用在*where*子句中。*
找出比最低工资高的员工信息。

```
select ename,sal from emp where sal > min(sal);
-- 表面上没问题，运行一下?
    ERROR 1111 (HY000): Invalid use of group function
```

说完分组查询(group by)*之后就明白了了。*

第四点：所有的分组函数可以组合起来一起用。

```
select sum(sal),min(sal),max(sal),avg(sal),count(*) from emp;
    +----------+----------+----------+-------------+----------+
    | sum(sal) | min(sal) | max(sal) | avg(sal)    | count(*) |
    +----------+----------+----------+-------------+----------+
    | 29025.00 |   800.00 |  5000.00 | 2073.214286 |       14 |
    +----------+----------+----------+-------------+----------+
```

19、分组查询（非常重要：五颗星*）

19.1、什么是分组查询?

在实际的应用中，可能有这样的需求，需要先进行分组，然后对每一组的数据进行操作。
这个时候我们需要使用分组查询，怎么进行分组查询呢?

```
select
...
from
...
group by
...
```

计算每个部门的工资和?
计算每个工作岗位的平均薪资?
找出每个工作岗位的最高薪资?
....

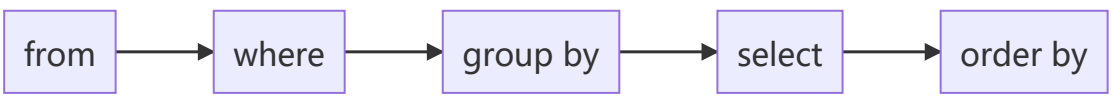
19.2、将之前的关键字全部组合在一起，来看一下他们的执行顺序?

```
select
...
from
...
where
...
group by
...
order by
...
```

以上关键字的顺序不能颠倒，需要记忆。

执行顺序是什么?

1. from
2. where
3. group by
4. select
5. order by



- 为什么分组函数不能直接使用在where后面？

```
select ename,sal from emp where sal > min(sal);-- 运行会报错
```

因为分组函数在使用的时候必须先分组之后才能使用。
where执行的时候，还没有分组。所以where后面不能出现分组函数。

```
select sum(sal) from emp;
```

这个没有分组，为啥sum()函数可以用呢？
因为select在group by之后执行。

19.3、找出每个工作岗位的工资和？

实现思路：按照工作岗位分组，然后对工资求和。

```
select
job,sum(sal)
from
emp
group by
job;

    +-----------+----------+
    | job       | sum(sal) |
    +-----------+----------+
    | ANALYST   |  6000.00 |
    | CLERK     |  4150.00 |
    | MANAGER   |  8275.00 |
    | PRESIDENT |  5000.00 |
    | SALESMAN  |  5600.00 |
    +-----------+----------+
```

以上这个语句的执行顺序？
先从emp表中查询数据。
根据job字段进行分组。
然后对每一组的数据进行sum(sal)

```
select ename,job,sum(sal) from emp group by job;
    +-------+-----------+----------+
    | ename | job       | sum(sal) |
    +-------+-----------+----------+
    | SCOTT | ANALYST   |  6000.00 |
    | SMITH | CLERK     |  4150.00 |
    | JONES | MANAGER   |  8275.00 |
    | KING  | PRESIDENT |  5000.00 |
    | ALLEN | SALESMAN  |  5600.00 |
    +-------+-----------+----------+
```

*以上语句在*mysql*中可以执行，但是毫无意义。*
*以上语句在*oracle*或者新版*sql*中执行报错。*
oracle*的语法比*mysql*的语法严格。（*mysql*的语法相对来说松散一些！）*

重点结论：
*在一条*select*语句当中，如果有*group by*语句的话，*
select*后面只能跟：参加分组的字段，以及分组函数。*
其它的一律不能跟。

19.4、找出每个部门的最高薪资

实现思路是什么？
按照部门编号分组，求每一组的最大值。

select*后面添加*ename*字段没有意义，另外*oracle*会报错。*

```
mysql> select ename,deptno,max(sal) from emp group by deptno;
+-------+--------+----------+
| ename | deptno | max(sal) |
+-------+--------+----------+
| CLARK |     10 |  5000.00 |
| SMITH |     20 |  3000.00 |
| ALLEN |     30 |  2850.00 |
+-------+--------+----------+
-- 此时max(sal)对应的ename也不为上表所示，古此方法没有意义，只会展示其表的该组的第一项ename

mysql> select deptno,max(sal) from emp group by deptno;
+--------+----------+
| deptno | max(sal) |
+--------+----------+
|     10 |  5000.00 |
|     20 |  3000.00 |
|     30 |  2850.00 |
+--------+----------+
```

- **19.5、*找出“每个部门，不同工作岗位”的最高薪资？***

```
+--------+-----------+---------+--------+
| ename  | job       | sal     | deptno |
+--------+-----------+---------+--------+
| MILLER | CLERK     | 1300.00 |     10 |
| KING   | PRESIDENT | 5000.00 |     10 |
| CLARK  | MANAGER   | 2450.00 |     10 |

| FORD   | ANALYST   | 3000.00 |     20 |
| ADAMS  | CLERK     | 1100.00 |     20 |
| SCOTT  | ANALYST   | 3000.00 |     20 |
| JONES  | MANAGER   | 2975.00 |     20 |
| SMITH  | CLERK     |  800.00 |     20 |

| BLAKE  | MANAGER   | 2850.00 |     30 |
| MARTIN | SALESMAN  | 1250.00 |     30 |
| ALLEN  | SALESMAN  | 1600.00 |     30 |
| TURNER | SALESMAN  | 1500.00 |     30 |
| WARD   | SALESMAN  | 1250.00 |     30 |
| JAMES  | CLERK     |  950.00 |     30 |
+--------+-----------+---------+--------+
```

技巧：两个字段联合成1个字段看。（两个字段联合分组）

```
select
deptno, job, max(sal)
from
emp
group by
deptno, job;
    +--------+-----------+----------+
    | deptno | job       | max(sal) |
    +--------+-----------+----------+
    |     10 | CLERK     |  1300.00 |
    |     10 | MANAGER   |  2450.00 |
    |     10 | PRESIDENT |  5000.00 |
    |     20 | ANALYST   |  3000.00 |
    |     20 | CLERK     |  1100.00 |
    |     20 | MANAGER   |  2975.00 |
    |     30 | CLERK     |   950.00 |
    |     30 | MANAGER   |  2850.00 |
    |     30 | SALESMAN  |  1600.00 |
    +--------+-----------+----------+
```

19.6、使用having可以对分完组之后的数据进一步过滤。

having不能单独使用，having不能代替where，having必须和group by联合使用。

找出每个部门最高薪资，要求显示最高薪资大于3000的？

第一步：找出每个部门最高薪资
按照部门编号分组，求每一组最大值。

```
select deptno,max(sal) from emp group by deptno;
    +--------+----------+
    | deptno | max(sal) |
    +--------+----------+
    |     10 |  5000.00 |
    |     20 |  3000.00 |
    |     30 |  2850.00 |
    +--------+----------+
```

第二步：要求显示最高薪资大于3000

```
select
deptno,max(sal)
from
emp
group by
deptno
having
max(sal) > 3000;
    +--------+----------+
    | deptno | max(sal) |
    +--------+----------+
    |     10 |  5000.00 |
    +--------+----------+
```

思考一个问题：以上的sql语句执行效率是不是低？
比较低，实际上可以这样考虑：先将大于3000的都找出来，然后再分组。

```
select
deptno,max(sal)
from
emp
where
sal > 3000
group by
deptno;
    +--------+----------+
    | deptno | max(sal) |
    +--------+----------+
    |     10 |  5000.00 |
    +--------+----------+
```

优化策略：

where*和*having，*优先选择*where，where*实在完成不了了，再选择*having。

19.7、where没办法的？？？？

*找出每个部门平均薪资，要求显示平均薪资高于*2500*的。*

第一步：找出每个部门平均薪资

```
select deptno,avg(sal) from emp group by deptno;
    +--------+-------------+
    | deptno | avg(sal)    |
    +--------+-------------+
    |     10 | 2916.666667 |
    |     20 | 2175.000000 |
    |     30 | 1566.666667 |
    +--------+-------------+
```

*第二步：要求显示平均薪资高于*2500*的*

```
select
deptno,avg(sal)
from
emp
group by
deptno
having
avg(sal) > 2500;

    +--------+-------------+
    | deptno | avg(sal)    |
    +--------+-------------+
```

```
|     10 | 2916.666667 |
+--------+-------------+
```

20、*大总结（单表的查询学完了）*

```
select
...
from
...
where
...
group by
...
having
...
order by
...
```

以上关键字只能按照这个顺序来，不能颠倒。

执行顺序?

1. from
2. where
3. group by
4. having
5. select
6. order by

从某张表中查询数据，
*先经过*where*条件筛选出有价值的数据。*
对这些有价值的数据进行分组。
*分组之后可以使用*having*继续筛选。*
select*查询出来。*
最后排序输出！

> *找出每个岗位的平均薪资，要求显示平均薪资大于*1500*的，除*MANAGER*岗位之外，要求按照平均薪资降序排。*

```
select
job, avg(sal) as avgsal
from
emp
where
job <> 'MANAGER'
group by
job
having
avg(sal) > 1500
order by
avgsal desc;

    +-----------+-------------+
    | job       | avgsal      |
    +-----------+-------------+
    | PRESIDENT | 5000.000000 |
    | ANALYST   | 3000.000000 |
    +-----------+-------------+
```