# MySQL

## 1、数据库

### 1、数据库的优势

1、持久化数据到本地

2、可以实现结构话查询，方便管理。

### 2、数据库的相关概念

DB(database):存储数据的长裤，它保存了一系列有组织的数据。

DBMS(Database Management System)数据库管理系统。数据库是通过DBMS创建和操作的容器。（操作DB）

常见的数据库库管理系统：

Oracle（神喻）：甲骨文（最高！）；（价格贵）

DB2:IBM；（处理海量的数据）

SQL Server:微软；（只能安装到wwindows）

Sybase：赛尔斯；

MySQL:甲骨文；

关系型数据库主要有 MySQL、 Oracle、 DB2, Infomix、 SQL Server 等数据库。

非关系型数据库主要有 NoSQL、 VoltDB 等数据库。

SQL(Structure Query Language)结构化查询语言：专门用来和数据库通信的语言。

1. 不是某个特定数据库供应商转悠的语言，几乎所有的DBMS都支持SQL。

### 3、数据库的特点

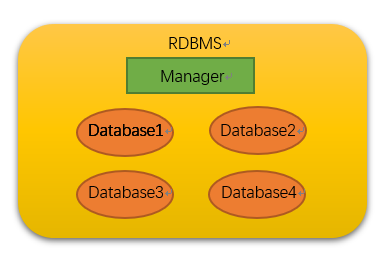
1、将数据放到表中，表在放到库中。

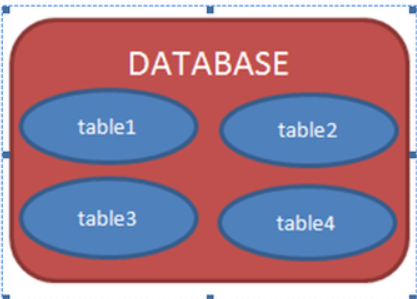
2、一个数据库中可以有多个表，每个表都有一个名字，用来表示自己。表名具有唯一性。

3、表具有一些特性，这些特性定义了数据在表中如何存储，类似java中“类”的设计。

4、表由列组成，也称之为字段。所有表都是由一个或多个列组成，每一列类似java中的属性。

5、表中的数据是按行存储，每一行雷士java中的“对象”。





## 2、MySQL的版本

### 1、MySQL的类别

MySQL Community Server：社区版完全免费， 但是官方不提供技术支持．

MySQL Server：企业版能够为企业提供高性能的数据库应用，以及高稳定性的数据库系统，提供完整的数据库提交、回滚以及锁机制等功能，但是该版本收费，官方只提供电话支持。

MySQL Cluster 主要用于建立数据库集群服务器，需要在以上两个版本的基础上使用。

### 2、版本命名规则

MySQL 的命名机制由 3 个数字组成， 例如： MySQL-5.6.15

第1个数字5是主版本号，用于描述文件格式，表示所有版本5的发行版都有相同的文件格式。

第二个数字6是发行级别，他与主版本号组合在一起构成了发行序列号。

第3个数字15是发行系列的版本号

## 3、Mysql的文件的安装和配置

### Mysql的安装

注意点：安装的文件路径不能有中文。

### 2、卸载mysql

1、停止MySQL。

2、添加删除程序中卸载MySQL

3、到安装目录删除MySQL

删除：C:\Documents and Settings\All Users\Application Data\MySQL

C：\ProgramData\MySQL

4、查看注册表：

cmd输入regedit

HKEY LOCAL MACHINE\SYSTEM\CurrentControlSet\Services

HKEY LOCAL MACHINE\SYSTEM\ControlSet001\Services

HKEY LOCAL MACHINE\SYSTEM\ControlSet002\Services

### 2、安装后的文件

### 1.MysQL安装成功后会在两个目录中存储文件：

>D:\Program Filea\MySQL\MySQL Server 5.1:DBMS管理程序

>C：\ProgramData\MysQL \MysQL Server 5.1\data:DBMs数据库文件（卸载MysQL时不会删除这个目录，需要自己手动删除）

2.MysQL重要文件

>D:\Program File3\MySQL\MySQIL Server 5.1\bin\myaql.exe:客户端程序，用来操作服务器。但必须保证服务器已开启才能连接上！

>D:\Program Files\MySQL\MySQL Server 5.1\bin\mysqld.exe:服务器程序，必须先启动它，客户端才能连接上服务器。

>D:\Program Files\MySQL\MySQL Server 5.1\bin\my.ini:服务器配置文件（里面的默认端口为3306，如没有必要不要修改）

3.C:PTOgramData\MySQL\MysQL Server 5:1\dataL（隐藏目录，都是数据库，一个文件夹都是一个数据库，里面的frm是表。该目录下的每个目录表示一个数据库，例如该目录下有一个mySQL目录，那么说明你的DBMS中有一个名为mySQL的database。

>在某个数据库目录下会有0~N个扩展名为frm的文件，每个frm文件表示一个table。你不要用文本编辑器打开它，它是由DBMS来读写的！

4.my.ini，MysQL最为重要的配置文件配置MySQL的端口：默认为3306，没有必要去修改它；

>配置字符编码：

\*[client]下配置客户端编码：default-character-set=gbk

\*[mySQLd]下配置服务器编码：character-set-server=utf8配置二进制数据大小上限：

\*在[mySQLd]下配置：maxallowed packet=8M

C：\ProgramData\MysQL\MysQLServer 5.5\data目录下：有一个目录就有一个数据库！！！每个数据库目录下，有一个frm文件，就说明有一张表！

5.如果在安装时不不小心退出了配置，可以在

## 4、MYSQL启动和登陆

客户端可以通过命令行或者图形化界面登陆数据库。

### 1、服务器操作（该操作必须在管理员模式下）

1.开启服务器（必须保证mysql为windows服务）：net start mysql。

>查看进程表中是否存在：mysql.exe进程（存在）

例如：net start +服务器名 net start mysql80

2.关闭服务器（必须保证mysql为windows服务）：net stop mysql。

>查看进程表中是否存在：mysql.exe进程不存在客户登操作cmd

2、登录服务器：mysql -uroot -p123 - hlocalhost -P 3306

mysql 【-h主机名 -P端口号】 -u用户名 -p密码（括号可选）

>-u：后面跟随用户名

>-p：后面跟随密码

>-h：后面跟随IP 127.0.01 给出IP地址

>-P :端口号

通过MySQL文件下的MySQL 8.0 Command Line Client 登录，但该方式只支持root用户。

3、退出服务器：exit或quit或者ctrl+c

## 5、SQL语言

### 1.SQL：结构化查询语言（Structured Query Language)

### 2.SQL的作用：客户端使用SQL来操作服务器。

>启动mysql.exe，连接服务器后，就可以使用SQL来操作服务器了。

>将来会使用Java程序连接服务器，然后使用SQL来操作服务器。

### 3.SQL标准（例如SQL99，即1999年制定的标准）。

>由国际标准化组织（ISO）制定的，对DBMS的统一操作方式（例如相同的语句可以操作：mySQL、oracle等）。

### 4.SQL方言

某种DEMS不只会支持SQL标准，而且还会有一些自己独有的语法，这就称之为方言！例如limit语句只在MySQL中可以使用。

### 5、SQL语法规则

1.sql语句可以在单行或多行书写，以分号结尾。

2.可使用空格和缩进来增强语句的可读性。

3.MySQL不区别大小写，但建议关键字大写，表名，列名小写。

4.注释：

单行注释：#注释文字

单行注释：-- 注释文字

多行注释：/\* 注释文字 \*/

5、sql中是不区分字符和字符串的。

6、查看mysql的版本：

SELECT version();

mysql –-version

mysql -V

# 2、SQL语句分类

1、DDL（Data Definition Language):数据定义语言，用来定义数据库对象：库、表、列等。

>创建、删除、修改：库、表结构！

2、DML（Data Manipulation Language):数据操作语言，用来定义数据库记录（数据）：地等节录>增、删、改：

3、DCL（Data Control Language)：数据控制语言，用来定义访问权限和安全级别：

4、DQLData Query Language）：数据查询语言，用来查询记录（数据）。

5事物查询语言。

DDL:数据库或表的结构操作

DML：对表的记录进行更新（增、删、改）

DQL:对包的记录的查询 难点

### DCL:对用户的创建，及授权。

## 2.数据类型（创建表时指定列类型）

int：整型

double：浮点型，例如:double(5，2）表示最多5位，其中必须有2位小数，即最大值为999.99；

decimal：浮点型，在表单钱方面使用该类型，因为不会出现精度缺失问题；

char：固定长度字符串类型；char(255) 数据不做 255，则会补足到255长度，如果超过255长度，则会报错。

varchar：可变长度字符串类型；varchar(65535)，zhangSan 原来数据的长度。

但是要专门一个字节要存储的多长。如果长度大于一个字节，要增加到两个字节。

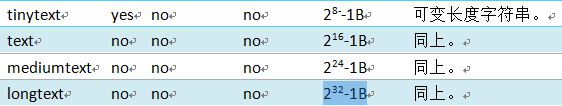
text(clob):字符串类型：mysql独有的类型，可以存储超大的字符数据。Clob为官方提供

I

>小

>中

>大



b1ob：字节类型：大的二进制字符（大字符）

>很小

>小

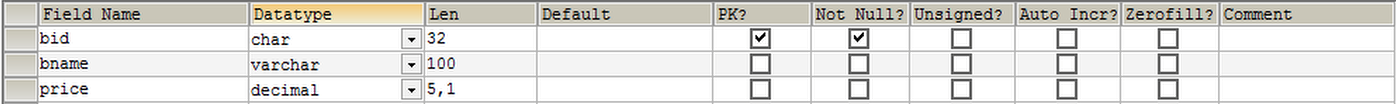
>中



date：日期类型，格式为：yyyy-04-dd；年月日

time：时间类型，格式为：hh:mm:sg 时分秒

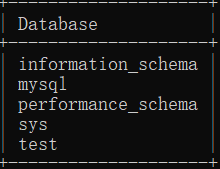
timestamp：时间戳类型：年月日时分秒



### 1、DDL

#### 1.数据库

\*查看所有数据库：SHOW DATABASES 1064 表示语法错误



Mysql:保存入库信息

Information\_schema:源数据信息

Performance\_schema:性能信息

test：测试数据库，没有表

\*切换（选择要操作的）数据库：USE数据库名

成功显示：Database changed

注：操作前要先选择库。

\*创建数据库：

CREATE DATABASE [IE NOT EXISTS ] database\_name [CHARSET=utf8]

[IF NOT EXISTS] :数据库不存时创建

[CHARSET=utf8] ：(编码方式)不提供编码时默认为utf8 安装时指定为utf8 括号中的内容是可选的

\*删除数据库：DROP DATABASE [IF EXISTS] database\_name；

\*修改数据库编码：ALTER DATABASE 数据库名 CHARACTER SET utf8（编码格式）

#### 3、表

\*创建表：（创建表前要指定库）

CREATE TABLE [IF NOT EXISTS] TABLE\_NAME（

列名 列类型，（逗号分隔）

列名 列类型，

列名 列类型，

…

列名 列类型

);

例如：CREATE TABLE st\_stu1(

Number char(11),

Name varchar(50),

Age int,

Gender varchar(10)

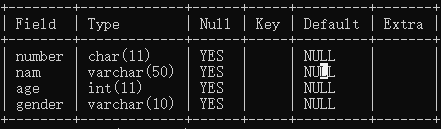
);

\*查看当前数据库中所有表名称：SHOW TABLES；

\*在当前的数据库查看别的数据库的表：SHOW TABLES FORM mysql

\*查看指定表的创建语句：SHOW CREATETABLE表名（了解）；

查看表结构：DESC表名；(description)



删除表：DROP TABLE表名；

\*修改表：前缀：ALTERTABLE表名

>修改之添加列：

ALTER TABLE表名ADD(

列名 列类型，

列名 列类型，

列名 列类型，

….

）

>修改之修改列类型（如果被修改的列已存在数据，那么新的类型可能会影响到已存在数据）：ALTER TABLE表名 MODIFY列名列类型；

>修改之修改列名（字段名）：

ALTER TABLE 表名 CHANGE原列名 新列名 新列类型；

>修改之删除列：ALTER TABLE表名DROP列名；

>修改表名称：ALTER TABLE原表名 RENAME TO新表名；

>删除数据表:drop table [if exists] table\_name1,table\_name2….;

DQL:SELECT \*FROM 表名 查询表记录

Emp：

Dept：部门表

Salgrade：

### 2、DML（数据操作语言，它是对表记录的操作（增、删、改）！

>插入数据

INSERT INTO表 名（列名1，列名2，…）VALUE（列值1，列值2，...）；

数据中所有的字符串类型，必须使用单引，不能使用双引！日期类型也要用单引

例如：INSERT INTO STU(

Number，name，age，gender

);

VALUES(

‘ITCAST\_0001’,’zhangsan’,28,’male’

);

查看是否添加成功

Select\*from table\_name;

注：1在表名后给出要插入的列名，其他没有指定的列等同与插入null值。所以插入记录总是插入一行，不可能是半行。

2、在VALUES后给出列值，值的顺序和个数必须与前面指定的列对应。

INTERT INTO 表名 VALUES（列值1，列值2）

>没有给出要插入的列，那么表示插入所有列。

>值的个数必须是该表列的个数。

>值的顺序，必须与表创建时给出的列的顺序相同。

例如：INSERT INTO STU VALUES(

‘ITCAST\_0001’,’zhangsan’,28,’male’

);

>修改数据

\* UPDATE 表名SET列名1=列值1，列名2=列值2，..[WHERE条件]  
 例如：UPDATE student SET age=18 WHERE name=‘libin’ or id=‘11’；

UPDATE student SET name=’John’ where name=’Smith’

条件（条件可选的）：

>条件必须是一个boolean类型的值或表达式：

UPDATE person SET gender=’男’，age=age+1 WHERE sid='1’；

>运算符：

=、l=、<>( 不等于)、>、<、>=、<=、BETWEEN...AND、IN(...）、IS NULL、NOT、OR、AND

BETWEEN...AND

例如:UPDATE student SET age=18 WHERE BETWEEN 18 and 40;//修改在18-40之间。或者是

UPDATE student SET age=18 WHERE age>= 18 and age<= 40;

IN(…)

UPDATE student SET age=18 WHERE name IN(‘zhangsan ’ ,’lisi’);条件为在in中里面的。

Not

UPDATE student SET age=18 WHERE not age>= 18 and age<= 40;

满足相反条件的。

IS Null

UPDATE student SET age=18 WHERE age=NULL;//错误的表示方法，=返回值为false

UPDATE student SET age=18 WHERE age IS NULL;//age的值为null都会修改

>删除数据（记录）

DELETE FROM 表名[WHERE条件]；

\*TRUNCATE TABLE 表名：TRUNCATE是DDL语句，它是先删除drop该表，再create该表。而且无法回滚！！！

# Dcl

一个项目创建一个用户，一个项目对应的数据库只有一个。

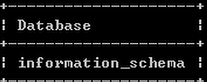
这个只能对这个数据库有权限，其他的数据库无法操作。

DCL（理解）

1.创建用户

\*CREATE USER用户名@ip地址 IDENTIFIED BY，‘密码’；

>用户只能在指定的IP地址上登录



创建后只能看见一个库。其他库得授权

\*CREATE USER用户名@‘%’ IDENTIFIED BY ’密码‘；

>用户可以在任意IP地址上登录。

2.给用户授权

授权格式：grant 权限 on 数据库.\* to 用户名@登录主机 identified by "密码";

>权限、用户、数据库

，给用户分派在指定的数据库上的指定的权限

>例如：GRANT CREATE，ALTER，DROP,INSERT,UPDATE,DELETE,SELECT ON mydb1.\* TO user1@localhost:

\*给user1用户分派在mydb1数据库上的create、alter、drop、insert、update、delete、select权限

\*GRANT ALL ON数据库.\*TO用户名IP地址；

>给用户分派指定数据库上的所有权限

例如：　　授权格式：grant 权限 on 数据库.\* to 用户名@登录主机 identified by "密码";

**2.1 登录MYSQL（有ROOT权限），这里以ROOT身份登录：**

　　@>mysql -u root -p

　　@>密码

**2.2 首先为用户创建一个数据库(testDB)：**

　　mysql>create database testDB;

**2.3 授权test用户拥有testDB数据库的所有权限（某个数据库的所有权限）：**

　　 mysql>grant all privileges on testDB.\* to test@localhost identified by '1234';

 　　mysql>flush privileges;//刷新系统权限表

　　格式：grant 权限 on 数据库.\* to 用户名@登录主机 identified by "密码";

**2.4 如果想指定部分权限给一用户，可以这样来写:**

　　mysql>grant select,update on testDB.\* to test@localhost identified by '1234';

　　mysql>flush privileges; //刷新系统权限表

**2.5 授权test用户拥有所有数据库的某些权限：**

　　mysql>grant select,delete,update,create,drop on \*.\* to test@"%" identified by "1234";

     //test用户对所有数据库都有select,delete,update,create,drop 权限。

　 //@"%" 表示对所有非本地主机授权，不包括localhost。（localhost地址设为127.0.0.1，如果设为真实的本地地址，不知道是否可以，没有验证。）

　//对localhost授权：加上一句grant all privileges on testDB.\* to test@localhost identified by '1234';即可。

3.撤销授权

\*REVOKE权限1，…，权限n oN数据库.\*FROM用户名自rB地址：

>撤消指定用户在指定数据库上的指定权限

>例如：REVOKE CREATE,ALTER，DROP ON mydb1.\* FROM user1@localhost:

\*撤消user1用户在mydb1数据库上的create、alter、arop权限

4.查看权限

\*SHOW GRANTS FOR用户名@ip地址

>查看指定用户的权限

5.删除用户\*DROP USER用户名@ip地址

　@>mysql -u root -p

　　@>密码

 　　mysql>Delete FROM user Where User='test' and Host='localhost';

 　　mysql>flush privileges;

 　　mysql>drop database testDB; //删除用户的数据库

删除账户及权限：>drop user 用户名@'%';

　　　　　　　　>drop user 用户名@ localhost;

6、修改用户密码

@>mysql -u root -p

  　　@>密码

  　　mysql>update mysql.user set password=password('新密码') where User="test" and Host="localhost";

  　　mysql>flush privileges;

## dql

查询不会对数据库修改

## 1、字段（列）控制或者属性

语法：select 查询列表 from 表名。

查询的内容：表中的字段（属性）、常量、表达式和函数。

查询的结果：是一个 虚拟的表格，不会对表的内容进行修改。

注意点：1、操作数据之前要先打开相应得库。

2、` 符号，数字1旁边的，使用他来表示字段，例如`name` 。目的是有些字段可能与sql 的关键字相同。

3、

### 查询单个字段

select last\_name from employees;

### 查询表中的多个字段

SELECT last\_name,salary,email FROM employees;

注：字段之间用逗号隔开，且数量和属性不受限制。

1. 查询表中的所有字段

SELECT last\_name,salary,email ,……FROM employees;

SELECT \* FROM employees;（\*表示所有字段，顺序为数据库存储的顺序。）

1. 查询常量值

SELECT 100;

SELECT ‘john’

1. 查询表达式

SELECT 100%98;

1. 查询函数

SELECT VERSION();调用该方法并得到它的返回值。

7、起别名

1）查询所有列

SELECT\*FROM表名；

SELECT \*FROM emp；

-->其中\*表示查询所有列

2）查询指定列

SELECT列1[，列2，..列N] FROM表名；（至少一列）

SELECT empno，ename，salcomm FROM表名；（只看姓名和工资）

3）完全重复的记录只一次

当查询结果的的多行记录一模一样时，只显示一行。一般查询所有列时很少会有这种情况，但只查询一列（或几列）时，这种可能就大了！

SELECT DISTINCT\*|列1，列2，...列N]FRoM表名；

SELECT DISTINCT sal FROM emp;

-->保查询员工表的工资，如果存在相同的工资只显示一次！

4）列运算

I数量类型的列可以做加、减、乘、除运算（不可以使用+来连接字符串）

SELECT sal\*1.5 FROM emp；无法转换成整数的都当成0，例如 name+100=100

SELECT sal+comm FROM emp；任何结果和null相加都为null

SELECT sal\*IFNULL<comm,0> FROM emp 判断是否为null，如果是，则用0代替。

II字符串类型可以做连续运算

SELECT CONCAT('$'，sa1)FRoM emp；可以将$符号与sal连接起来

II 转换NUll值

有时需要把ULL转换成其它值，例如com+1000时，如果com列存在NUL值，那么NULL+1000还是NULI，而我们这时希望把NULL当前o来运算。

SELECT IENULI(Comm,0)+1000 FROM emp；

-->TENULI(comm,0):如果comm中存在NULL值，那么当成0来运算。

v 给列起别名

你也许已经注意到了，当使用列运算后，查询出的结果集中的列名称很不好看，这时我们需要给列名起个别名，这样在结果集中列名就显示别名了

SELECT IFNULL(comm,0)+1000 As 奖金FROM emp；

->其中As可以省略

2.条件控制

蔡件查询II

与前面介绍的UPDATE和DELETE语句一样，SELECr语句也可以使用WHERE子句来控制记录。

\*SELECT empno,ename,sal,comm FROM emp WHERE sal>10000 AND comm IS NOT NULL;

\*SELECI empno,ename,sal FROM emp WHERE sal BETWEEN 20000 AND 30000；

\*SELECI empno,ename,job FROM ep WHERE job IN(1经理，，董事长）；

2）模糊查询

当你想查询姓张，并且姓名一共两个字的员工时，这时就可以使用模糊查询

\*SELECT\*FROM emp WHERE ename LIKE ’张\_’；匹配的字数为两个人

->模糊查询需要使用运算符：LIKE，其中\_匹配一个任意字符，注意，只匹配一个字符而不是多个。

的品工

->上面语句查询的是姓张，名字由两个字组成的

\*SELECT \* FROM emP WHERE ename LIKE: 姓名由3个字组成的员工

如果我们想查询姓张，名字几个字可以的员工时就要使用“%”了。

SELECT\*FROM emp WHERE ename LIKE‘张%’；

-一>其中匹配0~N个任意字符，所以上面语句查询的是姓张的所有员工。

SELECT \*FROM emp WHERE ename LIKE1s阿s！；

-->千万不要认为上面语句是在查询姓名中间带有阿字的员工，因为a匹配0~N个字符，所以姓名以阿开头和结尾的员工也都会查询到。

SELECT\*FROM emp WHERE ename LIKE's'；

-->这个条件等同与不存在，但如果姓名为NULL的查询不出来！

排序

二、排序

1）升序

SELECT \* FROM WHERE emp ORDER BY sal ASC（排序列） ;//升序

-->按sal排序，升序！

-->其中ASC是可以省略的

2）降序

SELECT \* FROM WHERE emp ORDER BY sal DESC;

-->按comm排序，降序！

-->其中DESC不能省略

3）使用多列作为排序条件(排序列存在相同值)

SELECT \* FROM WHERE emp ORDER BY sal ASC ，comm DESC:sal升序，当然sal相同时，comm使用降序排

练习： 1. 查询出部门编号为30的所有员工

SELECT \*

FROM emp

WHERE deptno=30;

2. 查询所有销售员的姓名、编号和部门编号。

SELECT ename, empno, deptno

FROM emp

WHERE job='销售员';

3. 找出奖金高于工资的员工。

SELECT \*

FROM emp

WHERE comm > sal

4. 找出奖金高于工资60%的员工。

SELECT \*

FROM emp

WHERE comm > sal\*0.6;

5. 找出部门编号为10中所有经理，和部门编号为20中所有销售员的详细资料。

SELECT \*

FROM emp

WHERE (deptno=10 AND job='经理') OR (deptno=20 AND job='销售员')

6. 找出部门编号为10中所有经理，部门编号为20中所有销售员，还有即不是经理又不是销售员但其工资大或等于20000的所有员工详细资料。

SELECT \*

FROM emp

WHERE (deptno=10 AND job='经理') OR (deptno=20 AND job='销售员') OR (job NOT IN ('经理', '销售员') AND sal >= 20000)

8. 无奖金或奖金低于1000的员工。

SELECT \*

FROM emp

WHERE comm IS NULL OR comm < 1000

9. 查询名字由三个字组成的员工。

SELECT \*

FROM emp

WHERE ename LIKE '\_\_\_'

10.查询2000年入职的员工。

SELECT \*

FROM emp

WHERE hiredate LIKE '2000-%' 模糊查询

11. 查询所有员工详细信息，用编号升序排序

SELECT \*

FROM emp

ORDER BY empno

12. 查询所有员工详细信息，用工资降序排序，如果工资相同使用入职日期升序排序

SELECT \*

FROM emp

ORDER BY sal DESC, hiredate ASC

13.查询每个部门的平均工资

SELECT deptno,AVG(3al)

FROM emp

GROUP BY deptno

14.查询每个部门的底员数量。

SELECT deptno,COUNT(\*)

FROM emp

GROUP BY deptno

15.查询每种工作的最高工资、最低工资、人数

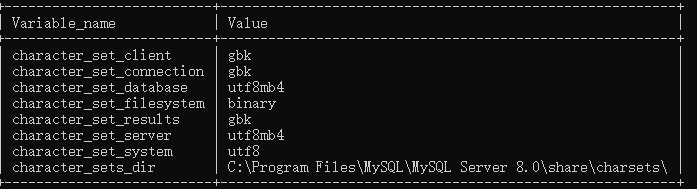
SELECT job,MAX(3al),MIN(sal),COUNT(\*)

FROM emp

GROUP BY job

1. 查看MYSQL数据库的编码

SHOW VARIABLES LIKE 'char%';



character\_set\_client=utf8,无论客户端发送的是什么编码的数据，mysq1都当成是utf8的数据！

>若客卢端发送的是GBK

>服务器会当成utf8对待

>总结：必然乱码！

处理问题的手段有两种；

>让客户端发送ut8的数据（行不通）

>把character\_set\_client修改为gbk

set character setclient=gbk；-->只在当前窗口内有效，也就是说，关闭窗口后，再打开，又回到utf8了。

character set results=utf8,把数据用什么编码发送给客户端！

>若服务器发送给客户端的是utf8的数据

>客户端会把它当成gbk，因为我们的小黑屏，只能显示gbk

>总结：必然乱码！

处理问题的手段有两种：

>让服务器发送gbk的数据：set character\_set\_results=gbk

>让小黑屏使用utf8来解读（行不通）

my.ini

在总配置文件中进行配置，可以一劳永逸

[client]

port=3306

[mysq1]

default-character-set=gbk/\*它可以一劳永逸！它可以修改三个变量：client、results、condection\*/

mysql数据的备份和恢复

1.数据库导出SQL脚本 （不用登陆数据库）

>myaqldump -u用户名-p密码数据库名生成的脚本文件路径

例如：mysqldump-uroot-p123mydb1>c:/mydb1.sg1(与mysgl.exe和mysqld.exe一样，、都在bin目录下）

>注意，不要打分号，不要登录mysq1，直接在cmd下运行

>注意，生成的脚本文件中不包含create database语句

2、恢复数据库内容，如果将数据库删掉，可以将数据库的内容恢复，无法恢复数据库

2.执行sQ脚本

第一种方式

>myaq1-u用户名-p密码数据库<脚本文件路径

>例如：

\*先删除mydb1库，再重新创建mydb1库

\*mysq1-uroot-p123 mydb1<C:\mydbl.sql

>注意，不要打分号，不要登录mysq1，直接在cmd下运行

第二种方式I

>登录mysq1

>source SQL脚本路径

>例如：

\*先删除mydb1库，再重新创建mydb1库

\*切换到mydb1库

\*source c:\\mydb1.sq1

mysq1dump-uroot-p123 mydb3>c:/a.sq1-->备份

mysq1-uroot-p123 mydb3<c:/a.sql-->恢复

source c:/a.sql-->恢复

约束：

1.主键约束（唯一标识 例如身份证）

\*\*\*\*非空\*\*\*表示不能卫龙

\*\*\*\*唯一\*\*\* 表示不能相同

\*\*\*\*被引用\*\*\*\* 外键引用 不是主键不能被引用，不具有唯一

\*当表的某一列被指定为主键后，该列就不能为空，不能有重复值出现。

\*创建表时指定主键的两种方式：

CREATE TABLE stu(

sid CHAR(6） PRIMARY KEY，//在指定列后加PRIMARY KEY 该列为主键列

sname VARCHAR(20)，

age INT，

gender VARCHAR(10)

）；

例如：

Insert into stu values（1，‘hello’）；

Insert into stu values（1，‘hi‘’）；//dumlicate 在唯一的字段插入重复的值。且该值不能为null。

指定sid列为主键列，即为sid列添加主键约束

且创建的主键的值不能重复。

另一种创建的方法

CREATE TABLE stu(

sid CHAR(6），

sname VARCHAR(20)，

ageINT，

gender VARCHAR(10)，

PRIMARY KEY(sid) //主键是sid

）；

指定sid列为主键列，即为sid列添加主键约束

\*修改表时指定主键：

’ALTER TABLE stu

’ADD PRIMARY KEY(sid)；

例如：

\*删除主键：ALTER TABLE Stu DROP PRIMARY KEY； 不用指定主键名，因为只有一个主键

2.主键自增长

\*因为主键列的特性是：必须唯一、不能为空，所以我们通常会指定主键类为整型，然后设置其自动增长，这样可以保证在插入数据时主键列的唯一和非空特性。

\*创建表时指定主键自增长

CREATE TABLE stu(

sid INT PRIMARY KEY AUTO INCREMENT,

sname VARCHAR(20)，

age

INT，

gender VARCHAR(10）

\*修改表时设置主键自增长：ALTER TABLE 3tu CHANGE sid sid INT AUTO INCREMENT；

\*修改表时删除主键自增长：ALTER TABLE Stu CHANGE sid 3id INT；

测试主键自增长：

>INSERT INTO Stu VALUES(NULL,'zhangSan',23,'male')；

>INSERT INTO stu(sname,age,gender)VALUES('zhangSan',23,'male');

3.非空约束

\*因为某些列不能设置为NULl值，所以可以对列添加非空约束。

\*例如：

CREATE TABLE stu（

sid INT PRIMARY EY AUTO INCREMENT，

sname VARCHAR（20）NOT NULL，//加个not null

ageINT，gender VARCHAR（10）

）；

\*对sname列设置了非空约束

4.唯一约束

\*数据库某些列不能设置重复的值，所以可以对列添加唯一约束。

\*例如：

CREATE TABLE stu（

sid INT PRIMARY KEY AUTO INCREMENT，

Bnamel VARCHAR（20）NOT NULL UNIQUE，非空和唯一同时加到统一字段

ageINT，gender VARCHAR（10）

）；

\*对name列设置了非空约束

5、概念模型

Is-a

has a（关联）

>1对1

>1对多

>多对多

use a

当我们要完成一个软件系统时，需要把系统中的实体抽取出来，形成概念模型。

例如部门、员工都是系统中的实体

。概念模型中的实体最终会成为Java中的类、数据库中表。

实体之间还存在着关系，关系有三种：

\*1对多：例如每个员工都从属一个部门，而一个部门可以有多个员工，其中员工是多方，而部门是一方。\*1对1：例如老公和老婆就是一对一的关系，一个老公只能有一个老婆，而一个老婆只能有一个老公。

\*多对多：老师与学生的关系就是多对多，一个老师可以有多个学生，一个学生可以有多个老师。

概念模型在Java中成为实体类（javaBean）

类就使用成员变量来完成关系，一般都是双向关联！多对一双问中关联，即员工关联部门，部门也关联员工

class Employee{//多方关联一方

private Department department；

}

class Department{//一方关联多方

private List<Employee>employeeslI

class Husband{//d一对一关系

private Wife wife；

}

class Wife{//一对一关系

private Husband class Student

}

Class Student{

private List<Teacher> teachers

}

class Teacher{//多对多关系

private List<Student>students；

}

6.外键约束

.外键必须是另一表的主键的值（外键要引用主键！）

\*外键可以重复

\*外键可以为空

CREATE TABLE emp(

empno INT PRIVMARY KEY AUTO\_INCREMENT,

ename VARCHAR(50),

dno INT,

CONSTRAINT fk\_emp\_dept FOREIGN KEY(dno)REFERENCES dept(deptno)

7、建立关系模型中的三种关系

1-1

主表：

CREATE TABLE hushand(

hid INT PRIMARY KEY AUTO\_INCREMENT,

hname VARCHAR(50)

);

INSERT INTO hasband VALUES（NULL，'刘备’）；

INSERT INTO hasband VALUES（NULL，'关羽’）；

INSERT INTO hasband VALUES（NULL，'张飞'）；

从表：

CREATE TABLE wife(

Wif INT PRIMARY KEY AUTO\_INCREMENT,

wname VARCHAR(50),

CONSTRAINT fk\_wife hasband FOREIGN KEY(wif)REFERENCES hasband (hid)

);

wid：非空，唯一，引用hid

INSERT INTO wife VALUES（1，'杨贵妃’）；

INSERT INTO wife VALUES（2，'妲己’）；

从表的主键即是外键

多对多

CREATE TABLE student

（ sid INT PRIMARY KEY AUTO\_INCREMENT,主键

sname VARCHAR(50)

)

CREATE TABLE teacher(

tid INT PRIMARY KEY AUTO\_INCREMENT,主键

tname VARCHAR(50)

)

CREATE TABLE stu\_tea(

sid INT,

tid INT,

CONSTRAINT fk student FOREIGN KEY(sid)REFERENCES atudent(收、id),

CONSTRAINT fk teacher FOREIGN KEY(tid) REFERENCES teacher(tid)

);

INSERT INTO student VALUES（NULL，'刘德华）；

INSERT INTO student VALUES（NULL，'梁期伟’）；

INSERT INTO student VALUES（NULL，黄日华）；

INSERT INTO student VALUES（NULL，'首侨伟’）；

INSERT INTO student VALUES（NULL，'汤朕业）；

INSERT INTO teacher VALUES（NULL，'崔老师）；

INSERT INTO teacher VALUES（NULL，'刘老师’）；

INSERT INTO teacher VALUES（NULL，'石老师1）；

INSERT INTO stu\_ tea VALUES(1,2);

INSERT INTO stu\_tea VALUES(2,2);

INSERT INTO stu\_tea VALUES(3,2);

## 多表查询

### 合并结果集（合并两张表格）

\*要求被合并的表中，列的类型和列数相同（实质为查询的两个结果集相同）

\*UNION，去除重复行

UNION ALL 保留重复行

SELECT \*FROM ab

UNION ALL

SELECT \*FROM cd;

SELECT \*FROM ab

UNION //去掉all后，完全相同的结果集会被去掉

SELECT \*FROM cd;

\*UNIONALL，不去除重复行

### 连接查询：一次性查询多个表

子查询：查询里面包查询

# 1、.内连接

## \*方言：SELECT\*FROM表1别名1，表2别名2WHERE别名1.xx=别名2.xx

\*标准：SELECT\*FROM表1别名1NNER JOIN表2别名2 ON别名1.xx=别名2.xx

\*自然：SETECT\*FROM表1别名1NATURAL JOIN表2别名2

\*内连接查询出的所有记录都满足条件

附录：SQLyog中的快捷键

F12 对编写的命令格式化