# MySQL基础知识

瑞典MySQL AB公司开发，

2008年1月16日，Sun宣布已经与MySQL AB达成协议，以大约10亿美元收购MySQL AB。

2009年04月20日，甲骨文宣布，该公司将以每股9.5美元的价格收购Sun。该交易价值约为74亿美元。

MySQL是一种关联数据库管理系统 由于其体积小、速度快、总体拥有成本低，一般中小型网站的开发都选择 MySQL 作为网站数据库。

## MySQL版本及下载

MySQL数据库版本相对比较繁杂。常见的有：Community社区版、Enterprise企业版。

Community版是开源免费的，这也是我们通常用的MySQL的版本。可以满足绝大多数用户需求。

Enterprise版，官方指出提供30天免费试用期。可进一步划分为MySQL标准版、MySQL企业版、MySQL集群版。官方提供付费服务。

其中Community Server 可以直接从mysql 的官网下载。但Enterprice Edition只能从Oracle edelivery上下载，而Edelivery有时会屏蔽中国IP。

1. MySQL Community Server 社区版本，开源免费，但不提供官方技术支持。
2. MySQL Enterprise Edition 企业版本，需付费，可以试用30天。
3. MySQL Cluster 集群版，开源免费。可将几个MySQL Server封装成一个Server。
4. MySQL Cluster CGE 高级集群版，需付费。
5. MySQL Workbench（GUI TOOL）一款专为MySQL设计的ER/数据库建模工具。它是著名的数据库设计工具DBDesigner4的继任者。MySQL Workbench又分为两个版本，分别是社区版（MySQL Workbench OSS）、商用版（MySQL Workbench SE）。

下载mysql时注意区分版本细节及所应用的系统平台：linux(32/64) 、win(32/64)

# MySQL的安装以及服务启动

详情请看《附件1：MySQL5.7安装文档.docx》

# MySQL的基本操作

重启MySQL服务

[root @localhost ~]# systemctl restart mysqld

查看MySQL服务状态

[root@localhost ~]# systemctl status mysqld

停止MySQL服务

[root@localhost ~]# systemctl stop mysqld

启动MySQL服务

[root@localhost ~]# systemctl start mysqld

## 数据库CURD

对数据库进行增(Create)、改(Update) 、查(Retrieve) 、删(Delete)操作。

### 创建数据库

**CREATE {DATABASE | SCHEMA} [IF NOT EXISTS] db\_name**  
**[create\_specification [, create\_specification] ...]**

* IF NOT EXISTS 表示只有数据库不存在的时候才创建，如果存在同名就不再执行该语句
* Create specification是建库的一些选项

1. 创建一个名称为mydb1的数据库。

create database mydb1;

1. 创建一个使用utf-8字符集的mydb2数据库。

create database mydb2 character set utf8;

1. 创建一个使用utf-8字符集，并带校对规则的mydb3数据库。会对存入的数据进行检查。

create database mydb3 character set utf8 collate utf8\_general\_ci;

### 查看数据库

显示所有数据库

show databases;

显示创建数据库的语句信息

show create database mydb2;

“ ` ”（ESC键 下面的按键），表示反引号，默认情况下，反引号括起来的字符串，区分大小写。

show create database mydb1;

注意 ：mysql默认语言集是latin1，每次在创建数据库的时候应指定字符集。Oracle是在安装时，即指定了字符集。

### 修改数据库

修改mydb1的字符集为utf8(不能修改数据库名)

alter database mydb1 character set utf8;

### 删除数据库

删除数据库mydb3

drop database mydb3;

## 表的CURD

对表本身进行操作：创建，查看，修改，删除

### 创建表

create table t1 (id int, name varchar(20))

但此时会报错误：ERROR 1046 (3D000): No database selected。注意，在mysql中对表操作前，必须先选择所使用的数据库。

use mydb2;

查看当前选择的数据库中的表：

show tables;

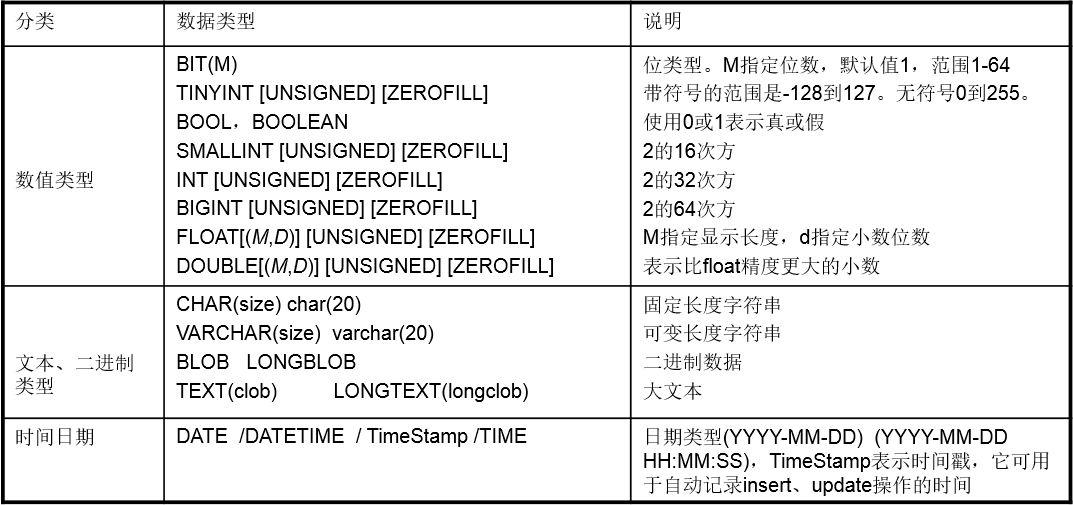
查看表结构：

在Mysql中显示多行数据应该在查询语句结尾处添加 \G来替换结束标记“；”

查看创建表的语法：

show create table t1; ENGINE=InnoDB 默认指定的存储引擎 innoDB。

mysql中的数据类型：



* bit 1位　 可以指定位数，如：bit(3)
* int 2字节 可以指定最大位数，如：int<4>　最大为4位的整数
* float 2个字节　可以指定最大的位数和最大的小数位数，如：float<5,2> 最大为一个5位的数，小数位最多2位
* double　4个字节　可以指定最大的位数和最大的小数位数，如：float<6,4> 最大为一个6位的数，小数位最多4位
* char　 必须指定字符数,如char(5) 为不可变字符　即使存储的内容为'ab',也是用5个字符的空间存储这个数据
* varchar　必须指定字符数,如varchar(5) 为可变字符　如果存储的内容为'ab',占用2个字符的空间；如果为'abc',则占用3个字符的空间
* text: 大文本(大字符串)
* blob：二进制大数据　如图片，音频文件，视频文件
* date: 日期　如：'1921-01-02'
* datetime: 日期+时间　如：'1921-01-02 12:23:43'
* timeStamp: 时间戳，自动赋值为当前日期时间

创建一个员工表：

create table employee(empno int, ename varchar(20), sal int);

### 查看表

查看所有的表：

show tables;

查看指定表的创建语句

show create table employee;

注意，mysql表名称区分大小写, 对列名不区分大小写

显示指定表的结构：

desc employee;

### 修改表

更改表名： rename table employee to worker;

增加一个字段： alter table employee add column height double; （column关键字在Oracle中，添加则语法错误）

修改一个字段：alter table employee modify column height float;

删除一个字段：alter table employee drop column height;

修改表的字符集:alter table employee character set gbk;

### 删除表

删除employee表

drop table employee; (MySQL中不能使用purge，添加会出现语法错误)

## 表数据的CURD

### create数据

创建一个员工表，新建employee表并向表中添加一些记录：

create table employee(

id int,

name varchar(20),

sex int,

birthday date,

salary double,

entry\_date date,

resume text

);

insert into employee values(1,'张三',1,'1983-04-27',15000,'2012-06-24','一个大牛');

insert into employee(id,name,sex,birthday,salary,entry\_date,resume) values(2,'李四',1,'1984-02-22',10000,'2012-07-24','一个中牛');

insert into employee(id,name,sex,birthday,salary,entry\_date,resume) values(3,'王五',0,'1985-08-28',7000,'2012-08-24','一个小虾');

### update数据

将所有员工薪水都增加500元。

update employee set salary=salary+500;

将王五的员工薪水修改为10000元，resume改为也是一个中牛

update employee set salary=10000, resume='也是一个中牛' where name='王五';

### delete数据

删除表中姓名为王五的记录。

delete from employee where name='王五'; 【注意from不能省略】

删除表中所有记录。

delete from employee;

使用truncate删除表中记录。

truncate table employee;--无条件 效率高

### Retrieve数据

查询员工的年收入：

select id, name as "名字", salary "月薪", salary\*12 年薪 from employee where id >=2;

### 综合案例

创建一个学生表，插入数据

create table student(

id int,

name varchar(20),

chinese int,

english int,

math int

);

insert into student(id,name,chinese,english,math) values(1, '范建',80,85,90);

insert into student(id,name,chinese,english,math) values(2,'罗况',90,95,95);

insert into student(id,name,chinese,english,math) values(3,'杜子腾',80,96,96);

insert into student(id,name,chinese,english,math) values(4,'范冰',81,97,85);

insert into student(id,name,chinese,english,math) values(5,'申晶冰',85,84,90);

insert into student(id,name,chinese,english,math) values(6,'郝丽海',92,85,87);

insert into student(id,name,chinese,english,math) values(7,'郭迪辉',75,81,80);

insert into student(id,name,chinese,english,math) values(8,'拎壶冲',77,80,79);

insert into student(id,name,chinese,english,math) values(9,'任我行',95,85,85);

insert into student(id,name,chinese,english,math) values(10,'史泰香',94,85,84);

#### 基础SQL

* 查询表中所有学生的信息。

select \* from student;

* 查询表中所有学生的姓名和对应的英语成绩。

select name,english from student;

* 过滤表中重复数据。

select english from student;

select DISTINCT english from student;

select DISTINCT english,name from student;

select english+chinese+math from student;

select english+chinese+math as 总分 from student;

select name,english+chinese+math as 总分 from student;

* 在所有学生英语分数上加10分特长分。

select name,english+10 from student;

* 统计每个学生的总分。

select english+chinese+math from student;

* 使用别名表示学生分数

select name,english+chinese+math as 总分 from student;

select name,english+chinese+math 总分 from student;

* 查询姓名为何东的学生成绩

select \* from student where name='何东';

* 查询英语成绩大于90分的同学

select \* from student where english>90;

* 查询总分大于250分的所有同学

select \* from student where english+chinese+math>250;

* 查询英语分数在 85－95之间的同学。

select \* from student where english>=85 and english<=95;

select \* from student where english between 85 and 95;

* 查询数学分数为84,90,91的同学。

select \* from student where math=84 or math=90 or math=91;

select \* from student where math in(84,90,91);

* 查询所有姓何的学生成绩。

select \* from student where name like '何%';

* 查询数学分>85，语文分>90的同学。

select \* from student where math>85 and chinese>90;

* 对数学成绩排序后输出。

select \* from student order by math;

* 对总分排序后输出，然后再按从高到低的顺序输出

select \* from student order by math+chinese+english desc;

* 对姓何的学生成绩排序输出

select \* from student where name like '何%' order by math+chinese+english desc;

select name, math+chinese+english from student where name like '何%' order by math+chinese+english desc;

* 统计一个班级共有多少学生？

select count(\*) from student;

* 统计数学成绩大于90的学生有多少个？

select count(\*) from student where math>90;

* 统计总分大于250的人数有多少？

select count(\*) from student where math+chinese+english>250;

* 统计一个班级数学总成绩？

select sum(math) from student;

* 统计一个班级语文、英语、数学各科的总成绩

select sum(math), sum(chinese), sum(english) from student;

* 统计一个班级语文、英语、数学的成绩总和

select sum(math+chinese+english)from student;

select sum(math)+sum(chinese)+sum(english) from student;

* 求一个班级数学平均分？

select avg(math) from student;

* 求一个班级总分平均分

select avg(math+chinese+english)from student;

select avg(math)+avg(chinese)+avg(english) from student;

* 求班级最高分和最低分

select max(math+chinese+english),min(math+chinese+english) from student;

#### 分组数据

为学生表，增加一个班级列，练习分组查询。

alter table student add column class\_id int;

注意语法：Oracle中不能有“column”关键字，MySQL中有没有“column”都可以执行。

更新表：

update student set class\_id=1 where id<=5;

update student set class\_id=2 where id>5;

(update student set class\_id=2 where id between 6 and 10;)

查出各个班的总分，最高分。

求各个班级 英语的平均分：

select classid, avg(english)

from student

group by class\_id

如根据组函数的语法要求，将select后增加name列，而不加至group by 之后：

select name, classid, avg(english)

from student

group by classid;

会发现MySQL检查不出错误。相比Oracle数据库，MySQL分组检查不严格。

select sum(math+chinese+english),max(math+chinese+english) from student group by class\_id;

查询出班级总分大于1300分的班级ID

select class\_id from student group by class\_id having sum(math+chinese+english)>1300;

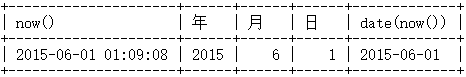
select class\_id from student where sum(math+chinese+english)>1300 group by class\_id ;

对于组函数的应用与Oracle类似，可以应用于Having中，但不能用于where子句中。

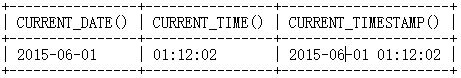
## 日期时间函数

MySQL里面时间分为三类：时间、日期、时间戳(含有时分秒的sysdate)。

如执行：select now(), year(now()) 年, month(now()) 月, day(now()) 日, date(now());



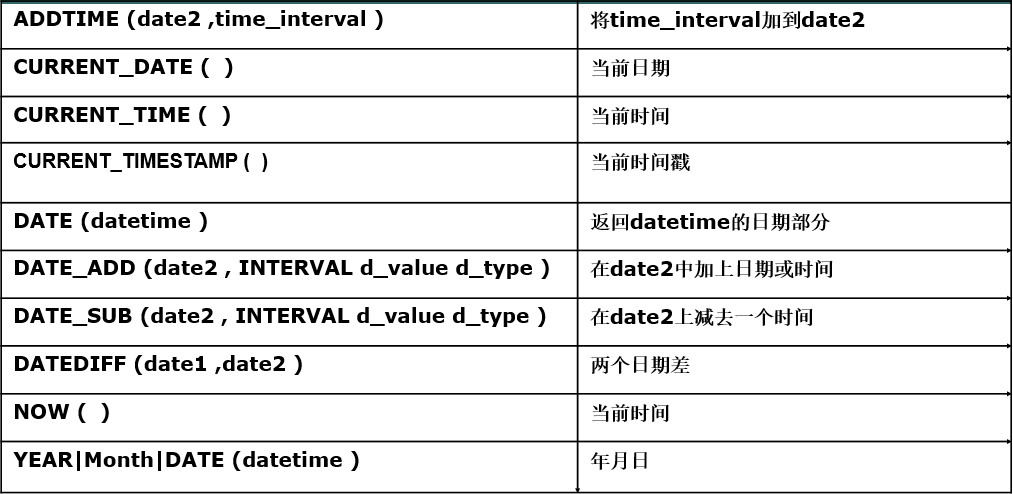
select CURRENT\_DATE() , CURRENT\_TIME(), CURRENT\_TIMESTAMP() from dual;



昨天、今天、明天：

select now()-1 昨天, now() 今天, now()+1 明天 from dual;

发现与Oracle中的日期加减操作有所不同。



select date\_add(now(), INTERVAL 2 year) from dual; //增加两年

select date\_add(now(), INTERVAL -1 day) 昨天, now() 今天, date\_add(now(), INTERVAL +1 day) 明天;

## 字符串相关函数

select concat('hello ', 'mysql ', 'haha ', 'hehe ') from dual;

Oracle默认只能拼两个，如需多个拼接可以使用嵌套。

select 'hello ' || 'mysql ' from dual; ‘||’ 在 MySQL不可以使用。



日期转字符串：

在MySQL中没有to\_date函数，进行日期转换需使用date\_format()来代替。

select date\_format(now(), 'yyyy-mm-dd') from dual; 在Oracle中的‘yyyy-mm-dd’MySQL下不支持。

select date\_format(now(), '%Y-%m-%d') from dual; y和Y不一样。

select date\_format(now(), '%Y-%c-%d %h:%i:%s') from dual; c和m、M不一样

所以yyyy-mm-dd hh24:mi:ss格式在MySQL中对应'%Y-%c-%d %h:%i:%s'

字符串转日期：

select str\_to\_date('2013-6-04 05:14:15' , '%Y-%c-%d %h:%i:%s') from dual;

MySQL的格式字符串格式如下：

|  |  |
| --- | --- |
| **格式** | **描述** |
| %a | 缩写星期名 |
| %b | 缩写月名 |
| %c | 月，数值 |
| %D | 带有英文前缀的月中的天 |
| %d | 月的天，数值(00-31) |
| %e | 月的天，数值(0-31) |
| %f | 微秒 |
| %H | 小时 (00-23) |
| %h | 小时 (01-12) |
| %I | 小时 (01-12) |
| %i | 分钟，数值(00-59) |
| %j | 年的天 (001-366) |
| %k | 小时 (0-23) |
| %l | 小时 (1-12) |
| %M | 月名 |
| %m | 月，数值(00-12) |
| %p | AM 或 PM |
| %r | 时间，12-小时（hh:mm:ss AM 或 PM） |
| %S | 秒(00-59) |
| %s | 秒(00-59) |
| %T | 时间, 24-小时 (hh:mm:ss) |
| %U | 周 (00-53) 星期日是一周的第一天 |
| %u | 周 (00-53) 星期一是一周的第一天 |
| %V | 周 (01-53) 星期日是一周的第一天，与 %X 使用 |
| %v | 周 (01-53) 星期一是一周的第一天，与 %x 使用 |
| %W | 星期名 |
| %w | 周的天 （0=星期日, 6=星期六） |
| %X | 年，其中的星期日是周的第一天，4 位，与 %V 使用 |
| %x | 年，其中的星期一是周的第一天，4 位，与 %v 使用 |
| %Y | 年，4 位 |
| %y | 年，2 位 |

## 数学相关函数



mysql> select abs(-1) from dual;

mysql> select bin(255) from dual;

mysql> select ceiling(5.9) from dual;

mysql> select conv(255, 10, 8) from dual;

mysql> select floor(5.8) from dual;

mysql> select format(3.1415926, 3) from dual;

mysql> select hex(255) from dual;

mysql> select least(3, 4, 5, -1 , 8, 0) from dual;

mysql> select mod(10, 3) from dual;

mysql> select rand() from dual;

## 多表查询

创建多表查询案例——MySQL建表\_仿照oracle建表脚本.sql 【mysql> source 绝对路径/脚本名】

Oracle中连接方法：

等值连接

不等值连接

外连接

自连接

MySQL 使用SQL99标准的连接查询（JOIN..ON..）

### 交叉连接：

叉集，即笛卡尔集

select e.\*, d.\*

from emp e cross join dept d

无连接条件

### 内连接

只返回满足连接条件的数据（两边都有的才显示）。 对应等值连接

select e.\*, d.\*

from emp e inner join dept d

on e.deptno=d.deptno

也可以省略inner关键字。

对应Oracle写法：

select e.\*, d.\*

from emp e , dept d

where e.deptno=d.deptno

### 左外连接

左边有值才显示。

select e.\*, d.\*

from emp e left outer join dept d

on e.deptno=d.deptno

也可以省略outer关键字

### 右外连接

右边有值才显示。

select e.\*, d.\*

from emp e right outer join dept d

on e.deptno=d.deptno

也可以省略outer关键字

【注意】SQL99中，外链接取值与关系表达式=号左右位置无关。取值跟from后表的书写顺序有关。

“xxx left outer join yyy” 则为取出xxx的内容。

“xxx right outer join yyy”则为取出yyy的内容

### 全外联接

任一边有值就会显示。

select e.\*, d.\*

from emp e full outer join dept d

on e.deptno=d.deptno

也可以省略outer关键字

### 对比练习

建立scott数据库，对比oracle进行练习：

#### 题目1：

查询员工信息,员工号,姓名,月薪,部门名称

select ...

from emp e, dept d

where e.deptno = d.deptno;

Oracle实现：

select e.deptno, e.ename, e.sal, d.dname

from emp e, dept d

where e.deptno = d.deptno

SQL99实现：

select e.deptno, e.ename, e.sal, d.dname

from emp e inner join dept d

on e.deptno = d.deptno

对比记忆规律：

“,” → [inner] join

where → on

对比结论：mysql能识别Oracle中使用 = 连接的书写方法。

#### 题目2：

统计各个部门员工总人数

分析：部门包括10/20/30 → 分组

各部门人数 → 多表

select ...

from emp e, dept d

where d.deptno = e.deptno

group by ...

(注意：group by后面出现的子集应在select下进行检索)

实现为：

select d.deptno, d.dname, count(e.deptno)

from dept d, emp e

where d.deptno = e.deptno

group by d.deptno, d.dname

查询发现没有40号部门。此时应使用外链接保存一侧结果。

oracle实现：

select d.deptno, d.dname , count(e.deptno)

from dept d, emp e

where d.deptno = e.deptno (+)

group by d.deptno, d.dname

SQL99实现：

select d.deptno, d.dname , count(e.deptno)

from dept d left outer join emp e

on d.deptno = e.deptno

group by d.deptno, d.dname

对比记忆规律：

“,”→ left**/**right outer join (左右要看from后面表出现的顺序)

where → on

结论：oracle的语法(+) mysql不支持

### 自连接

查询员工、老板信息，显示: xxx的老板是xxx

分析：将一张emp表当成两张表看待：员工表、老板表（员工表的老板 是 老板表的员工）

1. 先按照oracle语法写

select e.ename, b.ename

from emp e, emp b

where e.mgr = b.empno

1. 完善显示格式concat

select concat( e.ename, ' 的老板是 ', b.ename )

from emp e, emp b

where e.mgr = b.empno

1. 显示king的老板

select concat( e.ename, ' 的老板是 ', b.ename )

from emp e, emp b

where e.mgr = b.empno (+)

1. 改用MySQL支持的SQL99语法

select concat( e.ename, ' 的老板是 ', b.ename )

from emp e left outer join emp b

on e.mgr = b.empno ;

1. 滤空修正nvl

select concat( e.ename, ' 的老板是 ', nvl(b.ename, '他自己' ) )

from emp e left outer join emp b

on e.mgr = b.empno ;

结论 nvl 在mysql下不能使用： ERROR 1305 (42000): FUNCTION mydb61.nvl does not exist

1. 滤空修正 ifnull

select concat( e.ename, ' 的老板是 ', ifnull(b.ename, '他自己' ) )

from emp e left outer join emp b

on e.mgr = b.empno ;

注意：

Oracle中有一个通用函数，与MYSQL中的ifnull函数名字相近：

nullif：如nullif(a, b) 当 a = b 时返回null, 不相等的时候返回a值。nullif('L9,999.99', 'L9,999.99')

mysql中nullif（）函数也存在。

## 表的约束

\*定义主键约束　primary key: 不允许为空，不允许重复

\*定义主键自动增长　auto\_increment

\*定义唯一约束　unique

\*定义非空约束　not null

\*定义外键约束　constraint ordersid\_FK foreign key(ordersid) references orders(id)

\*删除主键：alter table tablename drop primary key ;

MySQL中约束举例：

create table myclass (

id INT(11) primary key auto\_increment,

name varchar(20) unique

);

create table student (

id INT(11) primary key auto\_increment,

name varchar(20) unique,

passwd varchar(15) not null,

classid INT(11),

constraint stu\_classid\_FK foreign key(classid) references myclass(id)

);

check约束在MySQL中语法保留，但没有效果。

# mysql中文乱码问题

两层因素：

因素1： MySQL自身的设计

【实验步骤1】：

mysql> show variables like 'character%'; 查看所有应用的字符集

【实验步骤2】：

$ mysql -uroot -p123456 --default\_character\_set=gbk 指定字符集登录数据库

mysql> show variables like 'character%';

影响了与客户端相关联的 3处 (最外层)

在这种状态下执行use mydb2;

mysql> select \* from employee;

查看输出，会出现乱码。

原来的三条数据，是以utf8的形式存储到数据库中，当使用gbk连接以后，数据库仍按照utf8的形式将数据返回，出错。

【实验步骤3】：

在该环境下插入带有中文的一行数据。

mysql> insert into employee(id,name,sex,birthday,salary,entry\_date,resume) values(10,'张三疯',1,'1983-09-21',15000,'2012-06-24','一个老牛');

ERROR 1366 (HY000): Incorrect string value: '\x80\xE4\xB8\xAA\xE8\x80...' for column 'resume' at row 1

因素2：操作系统的语言集

linux操作系统 是一个 多用户的操作

[root@localhost ~]# cat /etc/sysconfig/i18n

LANG="zh\_CN.UTF-8"

操作系统的菜单按照zh\_CN显示, 文件存储按照utf8

linux操作系统语言环境 和 用户的配置的语言环境LANG 相互影响

[mysql01@localhost ~]$ echo $LANG

zh\_CN.UTF-8

【实验步骤4】：

修改用户下的.bash\_profile 中的LANG，屏蔽操作系统的LANG设置。再查数据库

mysql> select \* from employee;

结论： 用户的LANG设置，影响了应用程序的语言环境,导致myql的语言环境发生了改变：

mysql> show variables like 'character%';

在此环境下，检索中文会出现乱码。

【实验步骤5】：在上述环境之下，向数据库中插入中文。

insert into employee(id,name,sex,birthday,salary,entry\_date,resume) values(5,'张三疯',1,'1987-05-21',15000,'2014-06-24','一个老牛');

数据能插入到数据库中，没 有 报 任 何 错 误！但显示不正确。