◆ **mysql 介绍**

(1)mysql 数据库是 瑞典 AB 开发

(2)mysql->sun->oracle

(3)mysql 数据库的特点

1. 开源
2. 免费
3. 跨平台(windows / linux /unix /苹果)
4. 处理并发性 (13000 个)/安全/稳定
5. 该数据库 轻(对资源要求不高.) 安装文件 37.7m ,而且对 cpu / 内存要求不高.

◆ **安装和配置**

1. 一般说，一台机器装一个 mysql.
2. 安装和配置过程见 mysql 安装图解

◆ **mysql 使用**

1. 可以使用 window dos 登录到 mysql 数据库.

基本语法

在 dos 下输入

mysql –u 用户名 -p 密码

特别说明: -p 后面的密码 不要有空格 / 使用该命令的时候，需要配置一下环境变量.

2. mysql 数据库也提供图形化界面来登录 mysql 数据库

演示:

3. 如何在 mysql 中创建数据库

基本语法

create database 数据库名

- 创建一个名称为 mydb1 的数据库。

create database mydb1; 【sql】

- 创建一个使用 utf-8 字符集的 mydb2 数据库。

create database mydb2 character set utf8

- 创建一个使用 utf-8 字符集，并带校对规则的 mydb3 数据库

create database mydb3 character set utf8 collate utf8_general_ci;

◆ mysql 的一些常用指令

①如何查看，创建数据库的指令:

show create databse 数据库名;

② 显示数据库

指令 show databases;

③查看创建数据库的指令

show create database 数据名;

④删除数据库:

drop database 数据库名

⑤ 如何指定使用某个数据库

use 数据库名;

⑥ 如何备份和恢复数据库.

mysqldump –u 用户名 –p 密码 数据名 > 存放路径

该指令，需要在 dos 控制台下直接执行

恢复数据库:

1.创建一个数据库 mydb2 ,但是这个数据库目前是空.

2.use 数据名

3.在 mysql 控制台下 使用 source 备份文件路径

◆ 创建表

基本语法

```
create table 表名 (
列名 列的数据类型,
....
) character set 字符集名称 collate 校对规则
```

案例 :

创建一张用户表

create table users (

id int ,

name varchar(64),

pwd varchar(64),

birthday date)

◆ **mysql 数据类型(重点)**

① 数值型:

1. bit(m)　m 默认为 1 最大 64

案例

create table test1 (id bit(1)); ---//这里显示乱码?

2. tinyint [unsigned] 如果是有符号则表示 -128 到 127 ，如果是无符号 0-255

案例

create table test3(num tinyint) -- -128 到　127

create table test4(num tinyint unsigned)　0 --- 255

4. smallint

samllint 是两个字节表示的.

带符号是　负的 2 的 15 次方 到 2 的 15 次方-1 ,无符号 2 的 16 方 -1

其它的数值类型，见下图即可

类型	字节	最小值	最大值
		(带符号的/无符号的)	(带符号的/无符号的)
TINYINT	1	-128	127
		0	255
SMALLINT	2	-32768	32767
		0	65535
MEDIUMINT	3	-8388608	8388607
		0	16777215
INT	4	-2147483648	2147483647
		0	4294967295
BIGINT	8	-9223372036854775808	9223372036854775807
		0	18446744073709551615

6． float

FLOAT[(*M*,*D*)] [UNSIGNED]　是定长

m ：　表示有效位

d: 表示小数点有几位

案例:

create table test5(num float);

create table test6(num float(5,1));

7. double

其用法和 float 类似，只是表示的范围更大,也是定长

8. numeric(m,d)

用于表示小数，或者整数

create table test7 (num numeric); //这样其实就是可以存放整数.

create table test8 (num numeric(5,2));//这样就可以表示 有效为 5,小数点有两位的数

② 字串类型

一览图:

列类型	存储需求
CHAR(*M*)	*M*个字节，0 <= *M* <= 255
VARCHAR(*M*)	*L*+1个字节，其中*L* <= *M* 且0 <= *M* <= 65535(参见下面的注释)
BINARY(*M*)	*M*个字节，0 <= *M* <= 255
VARBINARY(*M*)	*L*+1个字节，其中*L* <= *M* 且0 <= *M* <= 255
TINYBLOB, TINYTEXT	*L*+1个字节，其中*L* < 2^8
BLOB, TEXT	*L*+2个字节，其中*L* < 2^{16}
MEDIUMBLOB, MEDIUMTEXT	*L*+3个字节，其中*L* < 2^{24}
LONGBLOB, LONGTEXT	*L*+4个字节，其中*L* < 2^{32}
ENUM('*value1*','*value2*',...)	1或2个字节，取决于枚举值的个数(最多65,535个值)
SET('*value1*','*value2*',...)	1、2、3、4或者8个字节，取决于set成员的数目(最多64个成员)

常用的有

(1)char(m)

m 范围是 0-255, 定长.

char(20) 如果你存放 ‘abc’ 字串，实际在表 ‘abc ’;

案例:

create table test11 (name char(20));

☞ 小技巧:

mysql 自带的 client 默认支持 utf8 码，所有我们在添加中文的时候，需要设置让 client 支持 gbk

* show variables like ‘char%’; //显示关于字符的设置参数

* set character_set_client=gbk; //可以存中文

* set character_set_results=gbk; //可以看中文

(2) varchar(m)

m 表示大小 ,范围 0-65535， 变长

varchar(20) 如果你存放 ‘abc’ 字串，实际在表 ‘abc’;

案例 省略...

建议: 如果表的某列长度固定，比如 产品编号..学号. .. 而且在 255 内，我们应当使用 char

,如果长度不能取得，或者长度 大于 255 小于 65535 则使用 varchar

(3) text

该类型，可以表示更大的字串.

③ 日期类型

(1) date

日期 (年-月-日)

create table test12(birthday date);

对于 date 只保存 年-月-日

(2) datetime

日期时间类型

create table test13(hiredate datetime);

(3)timestamp

邮戳： 该类型可以保存 年-月-日 ： 时:分:秒

它和 datetime 最大的区别是，当你 update 某条记录的时候，该列值，最自动更新

create table test14 (name varchar(64) , sal float, hiredate1 timestamp, hiredate2 datetime);

```
mysql> select * from test14;
+------+------+---------------------+---------------------+
| name | sal  | hiredate1           | hiredate2           |
+------+------+---------------------+---------------------+
| aaa  | 56.7 | 2011-03-28 10:41:53 | 2011-03-28 10:41:53 |
+------+------+---------------------+---------------------+
1 row in set (0.00 sec)

mysql> update test14 set sal=sal+10 where name='aaa';
Query OK, 1 row affected (0.05 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from test14;
+------+------+---------------------+---------------------+
| name | sal  | hiredate1           | hiredate2           |
+------+------+---------------------+---------------------+
| aaa  | 66.7 | 2011-03-28 10:42:27 | 2011-03-28 10:41:53 |
+------+------+---------------------+---------------------+
```

建议: 如果不知道该不该用 timestamp ， 就不要用.

◆ 创建表综合案例

字段	属性
Id	整形
name	字符型
sex	字符型或 bit 型
brithday	日期型

Entry_date	日期型
job	字符型
Salary	小数型
resume	大文本型

create table emp(

id int,

name varchar(64),

sex char(2),

birthday date,

Entry_date date,

job varchar(32),

salary float,

resume text)

◆ 修改表结构

```
--添加新的列
alter table  表名  addd  列名  数据类型
--修改列(列的类型和大小)
alter table  表名  modify  列名  新的数据类型
--删除某列
alter table  表名  drop   列名
rename table  原表名  to  新表名
alter table  表名  character set  字符集名;
alter table user change column name username varchar(20);
```

案例:

- 在上面员工表的基本上增加一个 image 列。

alter table emp add image blob;

- 修改 job 列，使其长度为 60。

alter table emp modify job varchar(60);

- 删除 sex 列。

alter table emp drop sex;

- 表名改为 user。

rename table emp to user;

- 修改表的字符集为 utf-8

alter table user character set utf8;

- 列名 name 修改为 username

alter table user change column name username varchar(30);

如何显示创建表的指令:

show create table 表名;

◆ insert 语句

基本语法:

```
insert into  表名  [列名.....] values (值....);
```

● 插入的数据应与字段的数据类型相同。

比如:

create table test15 (name varchar(64));

insert into test15 (name) values(‘aaa’);

insert into test15 (name) values(34);

create table test16 (age int);

insert into test16 (age) values(34);

insert into test16 (age) values(‘aaa’);(错)

insert into test16 (age) values(‘111’);(虽然 ok,但是不是好的写法.)

- 数据的大小应在列的规定范围内，例如：不能将一个长度为 80 的字符串加入到长度为 40 的列中。

- 在 values 中列出的数据位置必须与被加入的列的排列位置相对应。

create table test17 (id int ,name varchar(64));

insert into test17 (id,name) values(3,’aaa’);

insert into test17 (name,id) values(’aaa’,3);

- **字符和日期型数据应包含在单引号中。**
- 插入空值，不指定或 insert into table value(null)

◆ update 语法

基本语法:

update 表名 set 列名=表达式 ... where 条件
说明: 如果 where 后面没有条件，则相当于对整个表进行操作。

- UPDATE 语法可以用新值更新原有表行中的各列。
- SET 子句指示要修改哪些列和要给予哪些值。
- WHERE 子句指定应更新哪些行。如没有 WHERE 子句，则更新所有的行。

将所有员工薪水修改为 5000 元。

update employee set sal=5000;

- 将姓名为’zs’的员工薪水修改为 3000 元;

update employee set sal=3000 where name=’zs’;

- 将 wu 的薪水在原有基础上增加 1000 元

update employee set sal=sa+1000 where name=’wu’;

◆ delete 语句

基本语法

```
delele from  表名  where  条件;
```

注意:

- 如果不使用 where 子句，将删除表中所有数据。
  所有要小心使用.
- Delete 语句不能删除某一列的值（可使用 update）

- 使用 delete 语句仅删除记录，不删除表本身。如要删除表，使用 drop table 语句。
- 同 insert 和 update 一样，从一个表中删除记录将引起其它表的参

照完整性问题，在修改数据库数据时，头脑中应该始终不要忘记这个潜在的问题。

- 删除表中数据也可使用 TRUNCATE TABLE 语句，它和 delete 有所不同，参看 mysql 文档。

  truncate table 表名，可以删除表的记录，速度快，但不能回滚..

在 mysql 中事务的特殊说明:

(1) mysql 控制台是默认自动提交事务(dml)

(2) 如果我们要在控制台使用事务，应该这样

- set autocommit=false;
- savepoint 保存点
- //操作...
- rollback to 保存点.

◆ **select 语句**

基本语法

```
select 列名...., 列（可以运行） from 表名 where 条件;
```

注意事项:

- Select 指定查询哪些列的数据。
- column 指定列名。
- *号代表查询所有列。

select * from 表名;

- From 指定查询哪张表。
- DISTINCT 可选，指显示结果时，是否剔除重复数据

select distinct * from 表名

- 练习：
    - 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 distinct * from 表名

- 练习
    - 在所有学生分数上加 10 分特长分(即查询所有学生总分再加 10 分)。

select english+math+chinese+10 , name from student;

    - 统计每个学生的总分。

    - 使用别名表示学生分数。

select english as ‘英语’, math as 数学 , chinese from student;

- 使用 where 子句，进行过滤查询。练习：
    - 查询姓名为 wu 的学生成绩

select english ,name from student where name = ‘wu’;

- 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 200 分的所有同学

select * from student where (math+english+chinese)>200;

◆ where 子句如何使用

- 在where子句中经常使用的运算符

比较运算符	> < <= >= = <> !=	大于、小于、大于(小于)等于、不等于
	BETWEEN ...AND...	显示在某一区间的值
	IN(set)	显示在in列表中的值，例：in(100,200)
	LIKE ‘张%’	模糊查询
	IS NULL[is not null]	判断是否为空
逻辑运算符	and	多个条件同时成立
	or	多个条件任一成立
	not	不成立，例：where not(salary>100);

Like语句中，% 代表零个或多个任意字符，_ 代表一个字符，例first_name like ‘_a%’;

案例:

- 查询英语分数在 80－90 之间的同学。

select * from student where english>=80 and english<=90;

- 查询数学分数为 89,90,91 的同学

select * from student where math in (89,90,91);

- 查询所有姓李的学生成绩。

select * from student where name lik ‘李%’;

- 查询数学分>80，语文分>80 的同学。

select * from student where matn>80 and chinese>80;

◆ order by 子句

- 练习：
    - 对数学成绩排序后输出。

select name,math 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 math+english+chinese as allfen , name from student order by allfen;

    - 对姓李的学生成绩排序输出

select (math+english+chinese) as allfen,name from student where name like ‘李%’ order by allfen;

◆ count

◆ 练习：

- 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+english+chinese)>250;

◆ sum 的用法

练习：

■ 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(english),sum(chinese) from student;

- 统计一个班级语文、英语、数学的成绩总和

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

- 统计一个班级语文成绩平均分

select sum(chinese)/count(*) from student;

◆ avg 的用法

◆ 练习：

- 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+english+chinese) from student;

◆ group by 用法

练习：对订单表中商品归类后，显示每一类商品的总价

select product , sum(price) from orders group by product;

◆ having 用法

练习：查询购买了几类商品，并且每类总价大于 100 的商品

select product , sum(price) from orders group by product having sum(price)>100

◆ 日期和时间函数

CURRENT_DATE ( )	**当前日期**

CURRENT_TIME ( )	**当前时间**
CURRENT_TIMESTAMP ( )	**当前时间戳**
DATE (datetime)	**返回 datetime 的日期部分**
DATE_ADD (date2 , INTERVAL d_value d_type)	**在 date2 中加上日期或时间**
DATE_SUB (date2 , INTERVAL d_value d_type)	**在 date2 上减去一个时间**
DATEDIFF (date1 ,date2)	**两个日期差(结果是天)**
TIMEDIFF(date1,date2)	**两个时间差(多少小时多少分钟多少秒)**
NOW ( )	**当前时间**
YEAR|Month|DATE (datetime)	**年月日**

案例:

select current_date() from dual ； 得到当前日期

select current_time() from dual ； 得到请求时间;

date_add() date_sub（）的用法

说： 有一个留言表

create table message(id int , title varchar(64), publishdate datetime);

请查询出，两个小时内，发布的消息:

select * from message where date_add(publishdate, interval 2 hour) >=

now();

```
mysql> select * from message where  date_add(publishdate, interval 2 hour) >= now();
+------+--------+---------------------+
| id   | title  | publishdate         |
+------+--------+---------------------+
|    1 | hello1 | 2011-03-28 14:56:29 |
+------+--------+---------------------+
1 row in set (0.00 sec)

mysql> select * from message where  date_add(publishdate, interval 2 hour) <= now();
+------+--------+---------------------+
| id   | title  | publishdate         |
+------+--------+---------------------+
|    2 | hello2 | 2010-11-11 14:23:23 |
+------+--------+---------------------+
```

特别说明

date_add(日期/date/datetime/timestamp, interval　数　type)

type 可以使用如下值:

type 值
MICROSECOND
SECOND
MINUTE
HOUR
DAY
WEEK
MONTH
QUARTER
YEAR

字符函数:

，常用函数一览图:

CHARSET(str)	**返回字串字符集**
CONCAT (string2　[,...])	**连接字串**
INSTR (string ,substring)	**返回 substring 在 string 中出现的位置,没有返回 0**
UCASE (string2)	**转换成大写**
LCASE (string2)	**转换成小写**
LEFT (string2 ,length)	**从　string2　中的左边起取**

	length 个字符
LENGTH (string)	**string 长度**
REPLACE (str ,search_str ,replace_str)	**在 str 中用 replace_str 替换 search_str**
STRCMP (string1 ,string2)	**逐字符比较两字串大小,**
SUBSTRING (str ,position [,length])	**从 str 的 position 开始,取 length 个字符**
LTRIM (string2) RTRIM (string2) trim	**去除前端空格或后端空格**

把 ename 列 的 smiTh 第一个字母大写，其它全部小写，怎么办?

select **UCASE(SUBSTRING (LCASE (‘**smiTh**’), 1,1)) from dual;**

//首先把‘smiTh’的首字母取出，转成大写

ucase(substring(‘smiTh’,1,1))

//把‘smiTh’去掉首字母后，余下的部分取出,//转成小写

lcase(substring(‘smiTh’,2,length(‘smiTh’)-1))

//最后拼接

select

concat(**ucase(substring(‘smiTh’,1,1)),**

lcase(substring(‘smiTh’,2,length(‘smiTh’)-1))) from dual;

结果:

select concat(lcase(substring('smiTh',1,1)), ucase(substring('smiTh',2,length('smiTh')-1))) from dual;

◆ 数学函数!!!

◆ mysql 的常见约束

① primary key (主键)

特点: 主键是用于唯一标识一条记录的约束，一张表，最多只能有一个主键，主键不能为 null,也不能重复

create table user1 (id int primary key, name varchar(32));

② auto_increment

可以自增长.

举例:

create table user2 (id int primary key auto_increment , name varchar(32));

③ unique （唯一）

特点: 表的某列的值，不能重复, 可以为 null (可以有多个 null), 一张表中可以有多个 unique.

create table user4(id int unique,name varchar(32));

④ not null (非空)

mysql 的表的列，默认情况下可以为 null, 如果不允许某列为空 ，则

可使用 not null 说明

create table user5(id int primary key, name varchar(32) not null);

⑤ 外键 foreign key

从理论上说明，我们先建立主表，再建从表

--部门表

create table dept(id int primary key,

name varchar(64));

insert into dept values(1,’财务部’);

--雇员表

create table emp(id int primary key,

name varchar(32),

deptid int references dept(id));

上面的建立外键的写法是错误的。

应该这样.(表级定义)

create table emp(id int primary key,

name varchar(32),

deptid int ,

constraint emp_fk foreign key (deptid) references dept(id)

);

小结外键:

（1）外键只能指向 主表的主键列，或者 unique

（2）外键的数据类型和它指向的列的数据类型一样.

（3）外键的值，要么为空，要么是指向的那列中存在值.

(4)外键可以指向本表的主键列，或者 unique

产品分类

```
create table producttye(
id int primary key,
catagory varchar(32),
parentId int,
constraint type_fk foreign key (parentid) references producttye(id));

insert into producttye values(1,’电器’,null);
insert into producttye values(2,’电冰箱’,1);
insert into producttye values(3,’电视机’,1);

insert into emp values(1,’张三’,2);
```

◆ check

```
create table user7 (age int check (age>12));
```

补充讲解 mysql 分页查询:

返回第 4 条 ----第 7 条记录

```
select * from student limit 3,4;
```

基本语法

select * from 表名 where 条件 ... limit 从第几条取，取出几条

从第几条取 ： 这里 mysql 从 0 开始编号.

- 安装语文成绩排序，查询出第 3 名到 第 5 名

select * from student order by chinese desc limit 2,3

扩展, 分页: pageNow, pageSize

select * from 表名 where 条件 [group by .. having .. order by ..] limit (pageNow-1)*pagesize, pageSize;