* MYSQL数据库
* SQL语句

## MYSQL数据库:

### 什么是数据库:

数据库就是一个文件系统,但是这个文件访问需要通过标准SQL语句.

### 什么是关系型数据库:

关系型的数据库通常保存的是实体之间**关系**.

### 常见的关系型数据库:

MYSQL :免费的小型的数据库,现在已经被Oracle收购了.

SQLServer :微软公司研发的数据库产品.收费的中型数据库.

Oracle :Oracle公司研发的数据库产品.收费的大型的数据库.

DB2 :IBM公司研发的数据库产品.收费的大型的数据库.

SyBase :数据库已经退出历史舞台.留下PowerDesigner:数据建模工具.

SQLite :嵌入式数据库.Android系统中使用.

## MYSQL的安装和卸载:

### MYSQL卸载:

1. 找到MYSQL的安装路径:

\* D:\Program Files (x86)\MySQL\MySQL Server 5.5\my.ini

1. 找到两个路径:

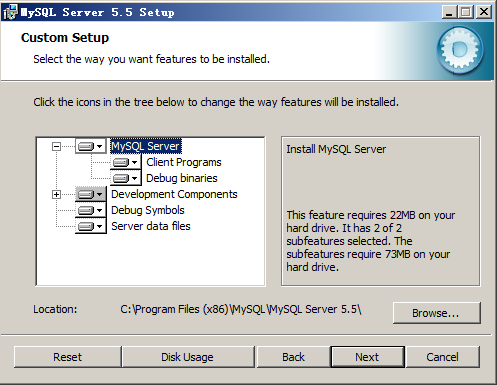
\* basedir="D:/Program Files (x86)/MySQL/MySQL Server 5.5/"

\* datadir="D:/ProgramData/MySQL/MySQL Server 5.5/Data/"

1. 在控制面板中卸载MYSQL:
2. 需要将记录的路径中的残留的文件删除:

### MYSQL安装:

1. 自定义安装:



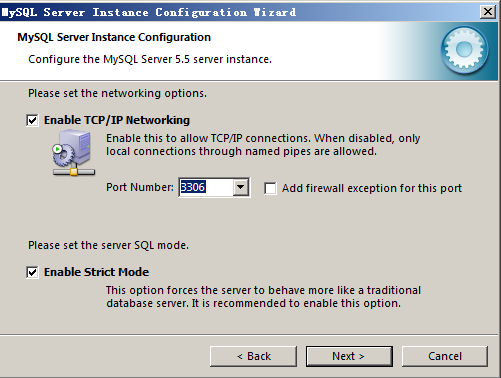
\* 服务器安装路径:

D:\Program Files (x86)\MySQL\MySQL Server 5.5\

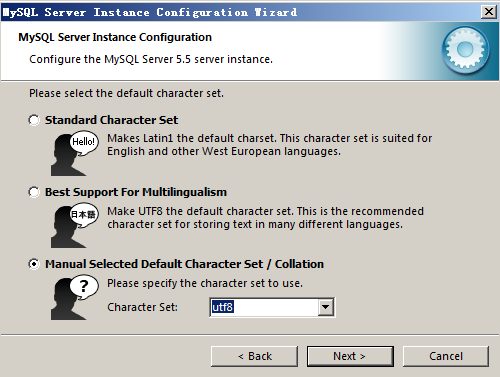
\* 数据存放的路径:

D:\ProgramData\MySQL\MySQL Server 5.5\

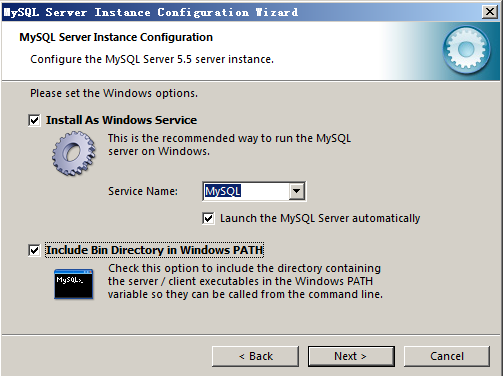
1. MYSQL默认端口号:



1. 设置MYSQL数据库字符集:



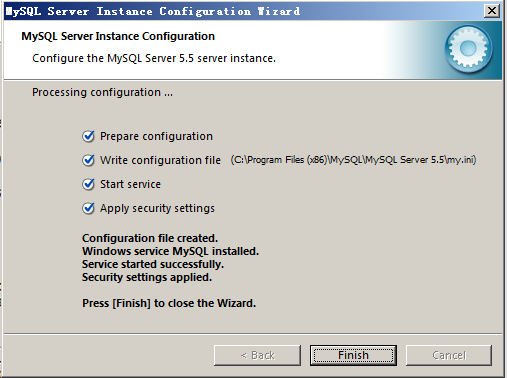
1. 勾选：Include Bin Directory in Windows PATH.



1. 设置MYSQL数据库最高管理员root的密码:



1. 四个对勾全部选中的情况下说明安装成功:



1. 测试MYSQL是否安装成功:

\* 在cmd的命令下输入:

\* mysql -u root -p



### 重置root密码:

1. 停止mysql服务:

\* services.msc进入服务界面:

1. 打开cmd的窗口:

\* 输入:mysqld --skip-grant-tables 回车.(相当于启动了一个MYSQL的服务,跳过权限认证)

1. 重新打开一个cmd的窗口:

\* mysql -u root -p

\* 输入密码: 直接回车

1. 使用数据库:

\* use mysql;

1. 输入命令:

\* update user set password = password('root123') where user='root';

1. 关闭两个cmd的窗口:
2. 在任务管理器中结束一个进程:mysqld.
3. 进入到服务界面:重启mysql服务.

### MYSQL内部存储结构:

一个软件对应一个数据库,在数据库内部可以存在多张表:一个实体类对应一张表.表中可以有多条记录的.一般情况下：一个类的实例对应一条记录.

## SQL语句:

### SQL的概述:

#### 什么是SQL:

SQL:Structured Query Language.结构化查询语言.

#### SQL的特性:

非过程性语言:

\* 过程性语言:一条语句的执行需要依赖其他的语句.

示例：

\* if(i < 3){

语句;

}

\* 非过程性语言:一条语句的执行,就会有一个结果.不需要依赖其他语句.

为了增强SQL语言能力,各个数据库生产商增强了SQL语言.

\* Oracle数据库中提供了PL/SQL

\* SQL Server:T-SQL.

#### SQL语法的分类:

**\* DDL :Data Definition Language.数据定义语言.**

**\* create 、alter、drop ...**

**\* DML :Data Manipulation Language.数据处理语言.**

**\* update 、 insert 、 delete...**

**\* DCL :Data Control Language.数据控制语言.**

**\* if 、grant...**

**\* DQL :Data Query Language.数据查询语言.**

**\* select**

### SQL对数据库进行操作:(CRUD)

#### 创建数据库:

语法:

\* create database 数据库名 [character set 字符集 collate 字符集校对规则];

**练习:**

**创建一个数据库名称为mydb1数据库.**

**\* create database mydb1;**

**创建一个使用utf8字符集的mydb2数据库.**

**\* create database mydb2 character set utf8;**

**创建一个使用utf8字符集,并带有校对规则的mydb3数据库.**

**\* create database mydb3 character set utf8 collate utf8\_bin;**

#### 查看数据库：

查看数据库服务器中有哪些数据库:

\* 语法:

\* show databases;

查看数据库的定义的细节:

\* 语法:

\* show create database 数据库名;

**练习:**

**\* 查看当前数据库服务器中有哪些数据库 ：**

**\* show databases;**

**\* 查看前面创建的mydb2数据库定义信息:**

**\* show create database mydb2;**

#### 删除数据库:

删除某个数据库 ：

语法:

\* drop database 数据库名;

**练习:**

**\* 删除前面创建的mydb1数据库:**

**\* drop database mydb1;**

#### 修改数据库:

修改数据库:

\* 语法:

\* alter database 数据库名 character set 字符集 collate 校对规则;

**练习:**

**\* 修改之前创建的mydb2数据库,将字符集改为gbk.**

**\* alter database mydb2 character set gbk;**

#### 切换数据库:

切换数据库 ：

\* 语法:

\* use 数据库名;

#### 显示当前正在使用的数据库:

查看当前正在使用的数据库 ：

\* 语法:

\* select database();

---------------------------------------------------以上属于DDL语句------------------------------------------------------------

### SQL语句对数据库中表进行操作:(CRUD)

#### 创建表:

语法:

\* create table 表名 (字段名 字段类型(长度) 约束,字段名 字段类型(长度) 约束..)[character set 字符集 collate 校对规则];

一般情况下有一个实体,一般就会创建一个表.

\* public class User{

private int id;

private String username;

private String password;

private double salary;

private int age;

}

MYSQL常用的数据类型:

Java MYSQL

String char/varchar

**\* char和varchar的区别:**

**\* char :代表的是固定长度的字符串.定义数据库中某个字段类型为char(8).存入一个hello.那么在hello字符串后面添加3个空格进行补齐.**

**\* varchar:代表的是可变长度的字符串.定义数据库中某个字段类型为varchar(8).存入一个hello.那么存入数据库就是hello这个字符串.**

byte TINYINT

short SMALLINT

int INT

long BIGINT

float FLOAT

double DOUBLE

boolean BIT

Date DATE/TIME/DATETIME/TIMESTAMP

**\* datetime与timestamp区别?**

**\* datetime :既有日期又有时间的日期类型.必须手动传入一个日期.**

**\* timestamp :既有日期又有时间的日期类型.如果没有给这个列设置值,采用系统的当前的时间存入到这个列上.**

**File BLOB/TEXT**

**\* BLOB :存放一个二进制文件.**

**\* TEXT :存放一个文件文件.**

**\*\*\*\*\* 在Oracle数据库中用BLOB和CLOB**

**练习:创建一个员工表:employee.**

**\* 在创建表之前一定要选择一个数据库:**

**\* create database web015;**

**\* use web015;**

**\* create table employee(**

**id int,**

**name varchar(20),**

**gender varchar(10),**

**birthday date,**

**entry\_date date,**

**job varchar(50),**

**salary double,**

**resume text**

**);**

**\*\*\*\*\* 字符串长度必须写,其他类型都有默认的长度.int默认长度11.**

**建表约束:(单表约束)---为了保证数据完整性.**

**\* 主键约束 :primary key.可以唯一确定表中一条记录的字段.(非空唯一).主键是整形可以使用auto\_increment.**

**\* 唯一约束 :unique.创建表中这个字段的时候,这个字段不允许出现重复的值.**

**\* 非空约束 :not null.代表创建表中的这个字段的时候,不可以向这个字段插入空值.**

**练习:创建一个带有约束的表:**

**create table employee2(**

**id int primary key auto\_increment,**

**name varchar(20) unique,**

**gender varchar(20) not null,**

**birthday date not null,**

**entry\_date date not null,**

**job varchar(50) not null,**

**salary double not null,**

**resume text not null**

**);**

#### 查看表:

查看数据库中有哪些表:

\* 语法:

\* show tables;

查看表的结构:

\* 语法:

\* desc 表名;

#### 修改表:

语法:

\* alter table 表名 add 字段名 类型(长度) 约束; ---修改表添加字段.

\* alter table 表名 modify 字段名 类型(长度) 约束; ---修改表修改字段的类型长度和约束.

\* alter table 表名 drop 字段名; ---修改表删除表中这个字段.

\* alter table 表名 change 旧的字段名 新的字段名 类型(长度) 约束; ---修改表的字段名

\* rename table 旧表名 to 新表名; ---修改表名.

\* alter table 表名 character set 字符集; ---修改表的字符集.

\* alter table 表名 change 旧字段 新字段 类型 after 某个字段 ---将某个字段放到某个字段的后面

\* alter table 表名 change 旧字段 新字段 类型 first ---将某个字段放到第一位

**练习:**

**\* 在上面员工表中添加一个image列.**

**\* alter table employee add image varchar(20);**

**\* 修改job列,使其长度变为60.**

**\* alter table employee modify job varchar(60);**

**\* 删除gender列.**

**\* alter table employee drop gender;**

**\* 表名改为user**

**\* rename table employee to user;**

**\* 修改表的字符集utf8**

**\* alter table user character set utf8;**

**\* 修改列名name修改为username.**

**\* alter table user change name username varchar(20);**

#### 删除表:

删除表语法:

\* drop table 表名;

**练习:**

**\* 删除user表:**

**\* drop table user;**

----------------------------------------以上属于DDL语句------------------------------------------------------------------------.

### SQL对数据库中表中的记录(数据)进行操作:(CRUD)

#### 添加记录:

语法:

\* 语法一:向表中插入部分字段.

\* insert into 表名 (字段名1,字段名2,字段名3...) values (值1,值2,值3...);

\* 语法二:向表中插入所有字段.

\* insert into 表名 values (值1,值2,值3...);

**\*\*\*\*\* 注意:**

**\* 1.添加的过程中,字段的个数与类型与值的个数与类型一致!**

**\* 2.添加字段的值的时候,值的长度不能超过字段设置最大长度!**

**\* 3.添加字段的值的类型如果是字符串或者日期类型,那么在插入的值的时候,就需要使用单引号引起来.**

**练习:**

**insert　into　employee　values　(2,'bbb','woman','1992-01-04','2013-02-03','WORKER',5600,'bbb');**

**查看插入的信息： select \* from 表名;**

**向数据库中添加3条记录:**

**\* 添加部分字段:**

**\* insert into employee (id,name,gender,job,salary) values (1,'aaa','man','HR',4500);**

**\* 添加所有字段:**

**\* insert into employee values(2,’bbb’,’woman’,’1992-01-03’,’2013-02-03’,’WORKER’,**

**5600,’bbb’);**

**\* 添加带有中文的数据:**

**\* insert into employee values (3,'张三','男','1994-01-04','2012-04-08','CEO',10000,'我是张三');**

\* 报错:Incorrect string value: '\xD5\xC5\xC8\xFD' for column 'name'

\* MYSQL数据库共有6个与字符集相关的参数!!!

\* 其中有3个与mysql的客户端相关的参数，3个与mysql服务器相关参数.

\*\*\*\*\* 解决插入中文乱码问题:

\* 找到mysql的安装路径：my.ini文件.

\* [mysql]

default-character-set=utf8

\* [mysqld]

character-set-server=utf8

\* 通过show variables like '%character%';---6个与字符集相关参数.

\* 将my.ini中与客户端相关的字符集的地方修改为gbk.

\* 重启mysql服务.

#### 修改记录:

语法:

\* update 表名 set 字段名 = 值,字段名=值 [where 条件];

**练习:**

**\* 将所有员工的工资改为6000元.**

**\* update employee set salary = 6000;**

**\* 将姓名为张三的员工的工资改为3000元.**

**\* update employee set salary = 3000 where name = '张三';**

**\* 将姓名为bbb的员工的工资改为9000元,将job改为Manager.**

**\* update employee set salary = 9000 , job='Manager' where name = 'bbb';**

**\* 将姓名为aaa的员工的工资在原有基础上加1000元.**

**\* update employee set salary = salary + 1000 where name = 'aaa';**

#### 删除记录:

语法:

\* delete from 表名 [where 条件];

**\*\*\*\*\* 注意:**

**\* 删除的是一条记录,不能删除一条记录的某个字段(修改).**

**\* 删除的是表的记录,而不是整个表.删除整个表 drop table;**

**\* 删除的时候一旦有外键的关联,删除表中记录的时候,先解除外键关系.**

**\* 删除表中的记录truncate table 表名; 删除表中所有记录:delete from 表名;**

**练习:**

**\* 删除表中姓名为张三记录:**

**\* delete from employee where name = '张三';**

**\* 删除表中的所有记录:**

**\* delete from employee ;**

**\* 使用truncate table删除表中所有记录:**

**\* truncate table employee;**

**\*\*\*\*\* truncate table 和delete from 区别?**

**\* delete属于DML.事务可以作用在DML语句上.事务进行回滚!!!delete是一条记录一条记录的删除.**

**\* truncate属于DDL.事务不能作用DDL上.truncate table 删除表中所有记录:直接将整个表进行删除.再创建一个新的表.**

**示例：**

**开启事务：start transaction**

**使用delete删除记录**

**回滚；rollback;**

------------------------------属于DML语句(insert,update,delete)--------------------------------------------------------------

#### 查询记录:(\*\*\*\*\*)

创建一个学生考试成绩表:

create table exam(

id int primary key auto\_increment,

name varchar(20),

math int,

english int,

chinese int

);

insert into exam values (null,'张三',62,83,79);

insert into exam values (null,'李四',73,92,81);

insert into exam values (null,'王五',81,83,65);

insert into exam values (null,'赵六',73,67,84);

insert into exam values (null,'孙七',93,75,93);

insert into exam values (null,'张八',72,69,95);

1. 基本查询语句:

\* 语法:

\* select [distinct] \*|[列名,列名...] from 表名;

\* distinct:去掉查询记录的中重复的值.

**\* 练习:**

**\* 查询表中所有学生信息:**

**\* select \* from exam;**

**\* 查询表中所有学生的姓名与英语成绩的信息:**

**\* select name,english from exam;**

**\* 过滤表中的重复数据:(过滤英语成绩信息)**

**\* select distinct english from exam;**

1. 执行运算及起别名:

\* 语法:

\* select 表达式 from 表;

\* select 列名 as 别名,列名 as 别名 from 表名; ----as关键字可以省略的.

**\* 练习:**

**\* 在所有学生的分数上加10分特长分:**

**\* select name ,math + 10,english + 10 ,chinese + 10 from exam;**

**\* 统计每个学生的总分:**

**\* select name ,math+english+chinese from exam;**

**\* 使用别名的方式表示学生的总分:**

**\* select name ,math+english+chinese as sum from exam;**

**\* select name ,math+english+chinese sum from exam;**

1. select语句的条件过滤:

语法:

\* select \*|列名,列名 from 表名 where 条件;

**练习:**

**\* 查询姓名为张三的学生成绩信息:**

**\* select \* from exam where name ='张三';**

**\* 查询英语成绩大于90分的学生信息:**

**\* select \* from exam where english > 90;**

**\* 查询总分大于240分的所有学生信息:**

**\* select name, math + english + chinese from exam where math + english + chinese > 240;**

**where子句后的运算符:**

**\* > ,< ,>= ,<= ,= ,<>**

**\* in :一组值,在某组数据之间.**

**\* like :模糊查询.**

**\* 使用占位符: \_和%。**

**\* \_代表的是一个字符: where name like '张\_';**

**\* %代表的是任意个字符: where name like '张%';**

**\* 张% :以张开头.**

**\* %张 :以张结尾.**

**\* %张%:包含张即可.**

**\* is null**

**\* and , or ,not**

**练习:**

**查询英语分数在80-90分的学生的信息:**

**\* select \* from exam where english > 80 and english < 90;**

**查询数学分数为62,81,93的学生的信息.**

**\* select \* from exam where math in (62,81,93);**

**查询所有姓张的学生信息:**

**\* select \* from exam where name like '张%';**

**查询数学分数>80,并且语文>80的学生信息:**

**\* select \* from exam where math > 80 and chinese > 80;**

1. 查询记录的排序

\* 语法:

\* select \* from 表 where 条件 order by 列名 asc/desc; ---asc:代表排序的时候按照升序排序.desc:代表排序的时候按照降序排序.默认值:asc:升序.

**\* 练习:**

**\* 对数学成绩进行排序后输出:**

**\* select \* from exam order by math;**

**\* 对总成绩按照从高到低顺序排序输出:**

**\* select name,math + english + chinese from exam order by math + english + chinese desc;**

**\* 对学生的数学成绩进行降序排序，如果数学成绩相同,那么参照英语成绩进行降序.**

**\* select \* from exam order by math desc,english desc;**

**\* 对姓张的同学进行总成绩的排序输出:**

**\* select name,math+english+chinese from exam where name like '张%' order by math+english+chinese ;**

1. 聚合函数:(分组函数).

\* count() :统计表中记录的个数.

\* 语法:

\* select count(\*|列名) from 表名;

**\* 练习:**

**\* 统计exam中有多少个学生:**

**\* select count(\*) from exam;**

**\* 统计数学成绩大于80分学生个数:**

**\* select count(\*) from exam where math > 80;**

**\* 统计总成绩大于220分学生个数:**

**\* select count(\*) from exam where math+english+chinese > 220;**

\* sum() :求和.

\* 语法:

\* select sum(列名) from exam;

**\* 练习:**

**\* 统计一个班级的数学总成绩:**

**\* select sum(math) from 表名;**

**\* 统计一个班级数学，英语，语文各科总成绩:**

**\* select sum(math),sum(english),sum(chinese) from exam;**

**\* 统计一个班级语文，数学，英语的成绩总和:**

**\* select sum(math)+sum(english)+sum(chinese) from exam;**

**\* select sum(math+english+chinese) from exam;**

**\* ifnull() :ifnull(math,0)如果math这列的值为null 那么就使用0代表.**

**\* select sum(ifnull(math,0)+english+chinese) from exam;**

**\* 统计一个班级的语文成绩的平均分:**

**\* select sum(chinese)/count(\*) from exam;**

\* avg() :求平均值.

\* 语法:

\* select avg(列名) from 表名;

**\* 练习:**

**\* 求一个班级的语文成绩的平均分:**

**\* select avg(chinese) from exam;**

**\* 求一个班级总分的平均分:**

**\* select avg(ifnull(math,0)+english+chinese) from exam;**

\* max() :求最大值.

语法:

\* select max(列名) from 表名;

**练习：**

**\* 查询语文成绩的最高分是多少分:**

**\* select max(chinese) from exam;**

\* min() :求最小值.

语法:

\* select min(列名) from 表名;

**练习:**

**\* 查询英语成绩的最低分是多少分:**

**\* select min(english) from exam;**

1. 分组统计:

\* 语法:

\* select \* from 表 where 条件 **group by 列名;**

\* 创建表:

create table orders(

id int,

product varchar(20),

price float

);

insert into orders values (1,'电视',800);

insert into orders values (2,'洗衣机',1200);

insert into orders values (3,'冰箱',2300);

insert into orders values (4,'电视',800);

insert into orders values (5,'洗衣机',1200);

**\* 练习:**

**对订单表中的商品归类后,显示每类商品的总价格:**

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

**对订单表中的商品归类后,显示每类商品的数量:**

**\* select product,count(\*) from orders group by product;**

**查询购买了几类商品,显示的是总价格大于2000的商品.**

**\* select product,sum(price) from orders where sum(price) > 2000 group by product; ---错误的!!!**

**\* select product,sum(price) from orders group by product where sum(price) > 2000 ; ---错误的!!!**

**\*\*\*\*\* where后面的条件不能使用聚集函数.**

**\*\*\*\*\* 使用另外一个关键字:having:----分组后的条件过滤.用于group by 进行条件过滤.having后面可以跟聚集函数.**

**(当对分组后的结果仍然不满意，需要添加其他条件，并且需要使用聚合函数时，可以使用having，having必须和group by一起使用，不能单独使用)**

**\*\*\*\*\* select product ,sum(price) from orders group by product having sum(price) > 2000;**

1. **查询语句的总结:**

**select**

**xxx**

**from**

**xxx**

**where**

**xxx**

**group by**

**xxx**

**having**

**xxx**

**order by**

**xxx**

**以上关键字的顺序不能改变，必须严格遵守**

**以上语句的执行顺序：**

**1)from 将硬盘上的表文件加载到内存.**

**2)where 将符合条件的数据行 摘取出来，生成一张新的零时表.**

**3)group by 根据列表中的数据种类，将当前临时表划分为若干个新的临时表.**

**4)having 可以过滤掉group by 生成的不符合条件的临时表.**

**5)select 对当前临时表进行整列读取.**

**6)order by 对select生成的临时表，进行重新排序，生成新的临时表.**

**7)limit 对最终生成的临时表的数据行，进行截取.**

\* S...F...W...G...H...O...:

**\* select \*|列名 from 表名 where 条件 group by 列名 having 分组条件过滤 order by 列名;**

-------------------------------------以上属于DQL---------------------------------------------------------------------------------

## 数据库的备份和还原:

### 数据库的备份:

数据库的备份:

\* 语法:

\* cmd --> mysqldump -u root -p 数据库名称 > d:/mysql171128.sql(不能加分号结尾)

### 数据库的还原:

数据库还原:

\* 要还原数据库：首先必须有一个空的数据库:

\* 语法

\* cmd --> mysql -u root -p 空数据库名称 < d:/mysql171128.sql(不能加分号结尾)

\* 进入到mysql数据库中进入到空数据库:

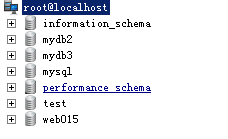
\* source d:/mysql171128.sql

## 可视化工具的使用:

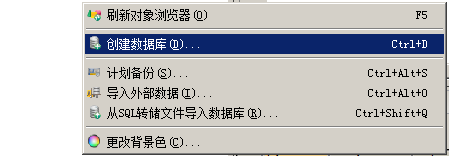
### 第三方的可视化工具的使用:

以SQLyog为例来学习可视化界面:

\* 连接后的界面:



\* 创建数据库 ：



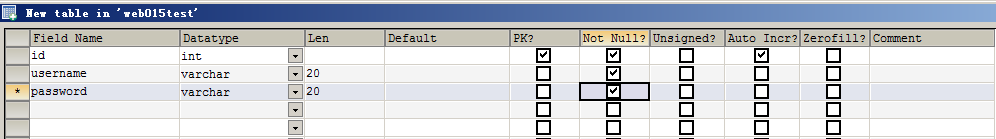
\* 创建成功后的界面:



\* 创建表:



\* 输入字段:



\*