

# 今天内容：MySQL数据库和SQL语言-拓薪教育

## 数据库的简介

1. 什么是数据库：数据仓库。访问必须只能用SQL语句来访问。数据库也是一个文件的系统。
2. 数据库的作用：存储数据的作用。开发任何的应用，都有数据库。
3. 关系型的数据库：数据库中保存的都是实体与实体之间的关系。
4. 常见的数据库
  - \* Java开发，必用的两个数据库Oracle和MySQL
    - \* Oracle数据库（甲骨文） 大型的数据库，收费的。
    - \* MySQL数据库 小型的数据库，免费开源的。被Oracle收购了（在6.x版本下开始收费了）
    - \* SQLServer 微软的数据库
    - \* DB2 IBM公司产品，大型的数据库，收费的。
    - \* SyBASE 退出了历史的舞台。PowerDigener（数据库的设计的工具）

## MySQL数据库的安装和卸载

1. MySQL数据库的卸载
  - \* 先找到MySQL的安装路径，找到my.ini配置文件。
  - \* basedir="C:/Program Files (x86)/MySQL/MySQL Server 5.5/" -- MySQL安装路径（my.ini没有删除）
  - \* datadir="C:/ProgramData/MySQL/MySQL Server 5.5/Data/" -- MySQL数据存放位置（手动删除）
  - \* 直接通过控制面板卸载程序。
2. 安装MySQL
  - \* 安装的路径中不能有中文和空格。
3. 进行测试
  - \* cmd -- 输入mysql -u root -p -- 回车 -- 输入密码 -- 进入MySQL的服务器。

## MySQL数据库概念

1. 总结：一个数据库的服务器中包含多个数据库，一个数据库中有多张表，一个表中包含多个字段（字段和JavaBean的属性是对应），表中存放是数据，一行数据和一个JavaBean实体对象是对应的。

## SQL语言（操作数据库）

1. Structured Query Language，结构化查询语言
2. SQL非过程性的语言
  - \* 过程性的语言：依赖上一条或者上几条语句执行。
  - \* 非过程性的语言：一条语言，就对应一个返回的结果。
3. SQL语言是基础
  - \* 在Oracle使用自己的语言，PL/SQL只能在Oracle来说使用。

## SQL的分类

- 1.DDL 数据定义语言
  - \* 创建数据库 创建表 创建视图 创建索引 修改数据库 删除数据库 修改表 删除表
  - \* create -- 创建 alter -- 修改 drop -- 删除
- 2.DML 数据操作语言
  - \* 操作数据 插入数据(insert) 修改数据(update) 删除数据(delete)
- 3.DCL 数据控制语言
  - \* if else while
- 4.DQL 数据查询语言
  - \* 从表中查询数据(select)

## 数据库的操作 ( CURD ) 创建数据库 ( 重点 )

- 1.创建数据库的语法
    - \* 基本的语法 : create database 数据库名称;
    - \* 正宗的语法 : create database 数据库名称 character set 编码 collate 校对规则;
  - 2.校对规则 ( 了解 ) : 决定当前数据库的属性。
- 创建一个名称为mydb1的数据库。
- \* 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,并带校对规则的mydb3数据库。
- \* create database mydb3 character set 'utf8' collate 'utf8\_bin';

## 查看数据库 ( 重点 )

- |                               |                 |
|-------------------------------|-----------------|
| 1.show databases;             | -- 查看所有数据库      |
| 2.use 数据库名称; (****)           | -- 使用数据库        |
| 3.show create database 数据库名称; | -- 查询数据库的创建的信息  |
| 4.select database();          | -- 查询当前正在使用的数据库 |

## 删除数据库 ( 重点 )

- |                        |          |
|------------------------|----------|
| 1.drop database 数据库名称; | -- 删除数据库 |
|------------------------|----------|

- 查看当前数据库服务器中的所有数据库
- \* show databases;
- 查看前面创建的mydb2数据库的定义信息
- \* show create database mydb2;
- 删除前面创建的mydb1数据库
- \* drop database mydb1;

## 修改数据库

- 1.语法 : alter database 数据库名称 character set 'gbk' collate '校对规则';

## 表结构操作 ( CURD )

### 创建表

### 1.语法：

```
create table 表名称(
    字段1 类型(长度) 约束,
    字段2 类型(长度) 约束,
    字段3 类型(长度) 约束
);
```

### 2.注意：

- \* 创建表的时候，后面用小括号，后面分号。
- \* 编写字段，字段与字段之间使用逗号，最后一个子段不能使用逗号。
- \* 如果声明字符串数据的类型，长度是必须指定的。
- \* 如果不指定数据的长度，有默认值的。int类型的默认长度是11

### 3.创建一张表结构（员工表练习）

```
create table employee(
    id int,
    name varchar(30),
    gender char(5),
    birthday date,
    entry_date date,
    job varchar(50),
    salary double,
    resume text
);
```

### 4.执行SQL语句

- \* 查询当前正在使用的数据库 `select database();`
- \* 选择你要使用的数据库 `use mydb2;`
- \* 执行创建表的SQL语句。

### 5.使用desc employee;查询表的信息

Field	Type	Null	Key	Default	Extra
id	int(11)	YES		NULL	
name	varchar(30)	YES		NULL	
gender	char(5)	YES		NULL	
birthday	date	YES		NULL	
entry_date	date	YES		NULL	
job	varchar(50)	YES		NULL	
salary	double	YES		NULL	
resume	text	YES		NULL	

## 数据库的数据类型（重点）

### 字符串型（重点）

VARCHAR（用的比较多）：长度是可变的。 例子：name varchar(8)，存入数据hello，存入进去之后，name字段长度自动变成了5。

CHAR：长度是不可变的。 例子：name char(8) 存入数据hello，用空格来补全剩余的位置。

### 大数据类型（不常用）

BLOB：字节（电影 mp3）

TEXT : 字符 ( 文本的内容 )

#### 数值型 ( 重点 )

TINYINT 、 SMALLINT、 INT、 BIGINT、 FLOAT、 DOUBLE

#### 逻辑性

BIT

在Java中是true或者false

在数据库bit类型 ( 1或者0 )

#### 日期型 ( 重点 )

DATE : 只包含日期 ( 年月日 )

TIME : 只包含时间 ( 时分秒 )

DATETIME : 包含日期和时间。如果插入数据的时候，字符值为空，字段的值就是空了。

TIMESTAMP : 包含日期和时间。如果插入数据的时候，设置字段的值为空，默认获取当前的系统的时间，把时间保存到字段中。

### 单表的约束 ( 了解 )

- 1.约束的好处：保证数据的完整性。
- 2.主键约束 ( 重要 ) 代表记录的唯一标识。
  - \* 关键字：primary key 通过该关键字声明某一列为主键。
  - \* 唯一 值就不能相同
  - \* 非空 值也不能为空
  - \* 被引用 ( 和外键一起来使用 )
- 3.唯一约束
  - \* 声明字段值是唯一的。使用关键字 unique
- 4.非空约束
  - \* 声明字段的值是不能空的。not null

### 删除和查看表

- 1.删除表语法：drop table 表名；
- 2.查看标签
  - \* desc 表名； -- 查询表的信息
  - \* show tables; -- 查看当前数据库中所有的标签
  - \* show create table 表名; -- 查看表的创建的信息

### 修改表

- 1.语法
  - \* alter table 表名 add 新列名 类型(长度) 约束; -- 添加列
  - \* alter table 表名 drop 列名; -- 删除列
  - \* alter table 表名 modify 列名 类型(长度) 约束; -- 修改列的类型或者约束
  - \* alter table 表名 change 旧列名 新列名 类型(长度) 约束; -- 修改列名
  - \* rename table 表名 to 新表名; -- 修改表的名称
  - \* alter table 表名 character set utf8; -- 修改表的字符集

在上面员工表的基本上增加一个image列。

```
alter table employee add image varchar(50);
```

修改job列，使其长度为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;
```

修改表的字符集为utf8

```
alter table user character set utf8;
```

列名name修改为username

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

## 数据的操作 ( CRUD ) ( 重点 )

### 插入数据 ( insert )

1. 插入数据的语法：

```
* insert into 表名 (字段1,字段2,字段3) values (值1,值2,值3);
```

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

2. 注意事项

- \* 插入的数据与字段类型必须是相同的。

- \* 数据的大小范围在字段范围内

- \* 值与字段一一对应

- \* 字符串或者日期类型数据需要使用单引号

```
insert into user values (1,'meimei','1956-1-1','1957-1-1','HR',5000,'meimeimei','xx');
insert into user values (2,'小凤','1996-1-1','2013-1-1','BOSS',15000,'mei','xx');
insert into user values (3,'聪聪','1993-11-11','2015-09-10','WORKER',500.0,'chou','yy');
insert into user values (4,'如花','1994-1-1','2013-1-1','BOSS',25000,'mei','xx');
insert into user values (5,'小苍','1991-1-1','2014-1-1','BOSS',15000,'mei','xx');
insert into user values (6,'小泽','1986-1-1','2013-1-1','BOSS',15000,'mei','xx');
```

character_set_client	utf8
character_set_connection	utf8
character_set_database	utf8
character_set_filesystem	binary
character_set_results	utf8
character_set_server	utf8
character_set_system	utf8

```
character_set_client=utf8      -- 客户端向MySQL服务器端发送内容
```

```
character_set_results=utf8    -- MySQL服务器端向客户端发送内容
```

## MySQL插入中文数据乱码

1. 先把MySQL服务停止。
2. 找到MySQL安装文件的my.ini的配置文件

```
[client]
port=3306
[mysql]
default-character-set=gbk
```
3. 重启MySQL服务

## 修改数据 ( update )

1. 语法 : update 表名 set 字段1=值, 字段2=值 where 条件; `where username = 'meimei';`
2. 如果没有where条件语句, 默认更新所有的数据。
3. 如果有where条件, 默认更新符合条件的记录。

将所有员工薪水修改为5000元。

```
update user set salary = 5000;
```

将姓名为'聪聪'的员工薪水修改为3000元。

```
update user set salary = 3000 where username = '聪聪';
```

将姓名为'小凤'的员工薪水修改为4000元, job改为ccc。

```
update user set salary = 4000, job = 'ccc' where username = '小凤';
```

将如花的薪水在原有基础上增加1000元。

```
update user set salary = salary+1000 where username = '如花';
```

## 删除数据 ( delete )

1. 语法 : delete from 表名 where 条件;
2. 如果没有where条件, 默认删除所有的数据。
3. truncate 表名; 删除表中所有的数据。delete from 表名; 也可以删除所有数据。
  - \* 区别 : truncate先把你整个表删除掉, 默默创建一个空的表 ( 和原来的表结构是一样的 )。
  - \* delete from 表名 一行一行的删除。 ( 使用它 )
  - \* 事物的概念 : 事物提交和事物回滚。

删除表中名称为'聪聪'的记录。

```
delete from user where username = '聪聪';
```

删除表中所有记录。

```
delete from user;
```

使用truncate删除表中记录。

## 查询数据 ( select ) ( 重点 )

### 基本的select语句

1. 语法
  - \* select \* from 表名; `-- 查询所有列的记录`
  - \* select 字段1, 字段2, 字段3 from 表名; `-- 查询字段123的记录`
  - \* DISTINCT -- 去除重复的数据 ( 面试 )

```
select distinct english from stu;
```

### 练习

```
create database day15;
use day15;
create table stu(
    id int,
    name varchar(30),
    math int,
    english int,
    chinese int
);

insert into stu values (1,'美美',78,93,56);
insert into stu values (2,'聪聪',18,13,16);
insert into stu values (3,'小凤',98,96,89);
insert into stu values (4,'如花',90,100,46);
insert into stu values (5,'欧阳锋',74,93,56);
insert into stu values (6,'吴彦祖',37,11,89);
insert into stu values (7,'聪大',88,77,66);
insert into stu values (8,'聪二',55,44,33);
```

## 查询语句中使用运算和别名

在所有学生分数上加10分特长分。

```
select name,(math+10) m,(english+10) e,(chinese+10) c from stu;
```

统计每个学生的总分。

```
select name,(math+english+chinese) 总分 from stu;
```

使用别名表示学生分数

```
select name,(math+english+chinese) 总分 from stu;
```

## 使用where条件过滤

查询姓名为聪聪的学生成绩

```
select name,math,chinese from stu where name = '聪聪';
```

查询英语成绩大于90分的同学

```
select name,english from stu where english > 20;
```

查询总分大于200分的所有同学

```
select name,math+english+chinese from stu where (math+english+chinese) > 200;
```

## where子句中出现的运算

1. > < <= >= = <> 大于、小于、大于(小于)等于、不等于

2. in 表示范围。

```
* select * from stu where math = 18;
```

查询出一条数据

```
* select * from stu where math in (78,18,99);
```

3. like 模糊查询 -- 符合模糊的条件

```
* select * from stu where name like '张_';
```

姓张的名称（只有两个）的记录

```
* select * from stu where name like '张%';
```

姓张的名称（张飞 张翼德 张是是冠希）的记录。

```
* select * from stu where name like '%张';
```

末尾是张（聪聪张 XSD张）

```
* select * from stu where name like '%张%';
```

只要名称中包含张。

4.is null 判断某一个字段记录是否为空

5.and与 or或者 not非

查询英语分数在 80 - 90之间的同学。

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

查询数学分数为89,90,91的同学。

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

查询所有姓小的学生成绩。

```
select * from stu where name like '小%';
```

查询数学分>80，语文分>80的同学。

```
select * from stu where math > 80 or chinese > 80;
```

总结：select 列名（运算） from 表名（别名） where 条件（运算的符号）；

## order by 对查询的结果进行排序

### 1. 排序的语法

```
* select * from 表名 where 条件 order by 列名 升序/降序;
```

### 2. 升序和降序

```
* order by 列名 asc; (升序, 默认值)
```

```
* order by 列名 desc; (降序)
```

### 3. order by 子句必须出现在select语句的末尾。

对数学成绩排序后输出。

```
select name,math from stu order by math desc;
```

对总分排序按从高到低的顺序输出

```
select name,(math+english+chinese) as total from stu order by total desc;
```

对学生成绩按照英语进行降序排序，英语相同学员按照数学降序

```
select name,english,math from stu order by english desc,math desc;
```

对姓聪的学生成绩排序输出

```
select name,(math+english+chinese) as total from stu where name like '聪%' order by total desc;
```

## 聚集函数

1. 聚集函数：总计某一列数据总和。一列的个数。一列的平均数。一列中最大值和最小值。

2. 聚集函数来操作列的。

### 3. 聚集函数

```
* count      -- 计数
```

```
* sum        -- 求和
```

```
* ifnull     判断是否为空：语法：ifnul(xxx,0)    如果xxx为null，替换成0
```

```
* avg        -- 平均值
```

```
* max        -- 最大值
```

```
* min        -- 最小值
```

练习：

统计一个班级共有多少学生？

```
select count(name) from stu;
```

统计数学成绩大于90的学生有多少个？

```
select count(math) from stu where math >= 90;
```

统计总分大于220的人数有多少？

```
select count(*) from stu where math + english+chinese > 200;
```



统计一个班级数学总成绩？

```
select sum(math) from stu;
```

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 stu;
```

统计一个班级语文、英语、数学的成绩总和

```
select sum(ifnull(math,0)+english+chinese) from stu;
```

```
select sum(math) + sum(english) + sum(chinese) from stu;
```

\* 编写一条更新语句：update stu set math = null where id = 2;

统计一个班级语文成绩平均分