



黑马程序员™
www.itheima.com

传智播客旗下
高端IT教育品牌

MySQL 基础

目录 Contents

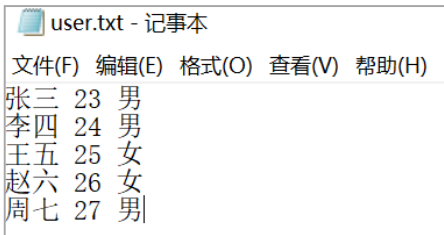
- ◆ 数据库的基本概念
- ◆ DDL 操作数据库和数据表
- ◆ DML 表数据的增删改
- ◆ DQL 表数据的查询
- ◆ 约束



数据库的基本概念

为什么要学习数据库?

- 更加方便的对开发中的数据进行管理!
- 我们之前是如何对数据进行管理的呢? IO 流。



- ① 使用字符缓冲流每次读取一行数据
- ② 将读取到的数据封装为 User 对象
- ③ 将多个 User 对象保存到集合中
- ④ 遍历集合, 判断对象名称是否是王五
- ⑤ 是王五, 将其年龄修改为35
- ⑥ 使用字符缓冲流将集合数据写回文件



数据库的基本概念

为什么要学习数据库?

USER		
name	age	gender
张三	23	男
李四	24	男
王五	25	女
赵六	26	女
周七	27	男

```
UPDATE USER SET age=35 WHERE NAME='王五';
```



为什么要学习数据库?

USER		
name	age	gender
张三	23	男
李四	24	男
王五	35	女
赵六	26	女
周七	27	男








```
UPDATE USER SET age=35 WHERE NAME='王五';
```



数据库的基本概念

数据库的介绍

- 用于存储和管理数据的仓库。
- 英文单词为 DataBase。简称 DB!
- 它的存储空间很大，可以存放百万条、千万条、上亿条数据。
- 使用一种统一的方式操作数据库 —— SQL








Rank			DBMS	Database Model	Score		
Sep 2019	Aug 2019	Sep 2018			Sep 2019	Aug 2019	Sep 2018
1.	1.	1.	Oracle +	Relational, Multi-model 	1346.66	+7.18	+37.54
2.	2.	2.	MySQL +	Relational, Multi-model 	1279.07	+25.39	+98.60
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model 	1085.06	-8.12	+33.78
4.	4.	4.	PostgreSQL +	Relational, Multi-model 	482.25	+0.91	+75.82
5.	5.	5.	MongoDB +	Document	410.06	+5.50	+51.27
6.	6.	6.	IBM Db2 +	Relational, Multi-model 	171.56	-1.39	-9.50
7.	7.	7.	Elasticsearch +	Search engine, Multi-model 	149.27	+0.19	+6.67
8.	8.	8.	Redis +	Key-value, Multi-model 	141.90	-2.18	+0.96
9.	9.	9.	Microsoft Access	Relational	132.71	-2.63	-0.69
10.	10.	10.	Cassandra +	Wide column	123.40	-1.81	+3.85



数据库的基本概念

数据库的介绍

- 用于存储和管理数据的仓库。
- 英文单词为 DataBase。简称 DB！
- 它的存储空间很大，可以存放百万条、千万条、上亿条数据。
- 使用一种统一的方式操作数据库 —— SQL

Rank			DBMS	Database Model	Score		
Sep 2019	Aug 2019	Sep 2018			Sep 2019	Aug 2019	Sep 2018
1.	1.	1.	Oracle +	Relational, Multi-model 	1346.66	+7.18	+37.54
2.	2.	2.	MySQL +	Relational, Multi-model 	1279.07	+25.39	+98.60
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model 	1085.06	-8.12	+33.78
4.	4.	4.	PostgreSQL +	Relational, Multi-model 	482.25	+0.91	+75.82
5.	5.	5.	MongoDB +	Document	410.06	+5.50	+51.27
6.	6.	6.	IBM Db2 +	Relational, Multi-model 	171.56	-1.39	-9.50
7.	7.	7.	Elasticsearch +	Search engine, Multi-model 	149.27	+0.19	+6.67
8.	8.	8.	Redis +	Key-value, Multi-model 	141.90	-2.18	+0.96
9.	9.	9.	Microsoft Access	Relational	132.71	-2.63	-0.69
10.	10.	10.	Cassandra +	Wide column	123.40	-1.81	+3.85



数据库的基本概念

数据库的介绍

- MySQL 是一个最流行的关系型数据库管理系统之一。由瑞典 MySQL AB 公司开发，后被 Oracle 公司收购。
- 关系型数据库是将数据保存在不同的数据表中，而不是将所有数据放在一个大仓库内，而且表与表之间还可以有关联关系。这样就提高了访问速度以及提高了灵活性。
- MySQL 所使用的 SQL 语句是用于访问数据库最常用的标准化语言。
- 免费(6 版本之前)。



数据库的基本概念

数据库的安装

- 为了能够更好的体现真实开发环境，我们将 MySQL 安装在 Linux 系统上，以此来模拟公司的数据库服务器。
- 安装及配置过程：参考文档。

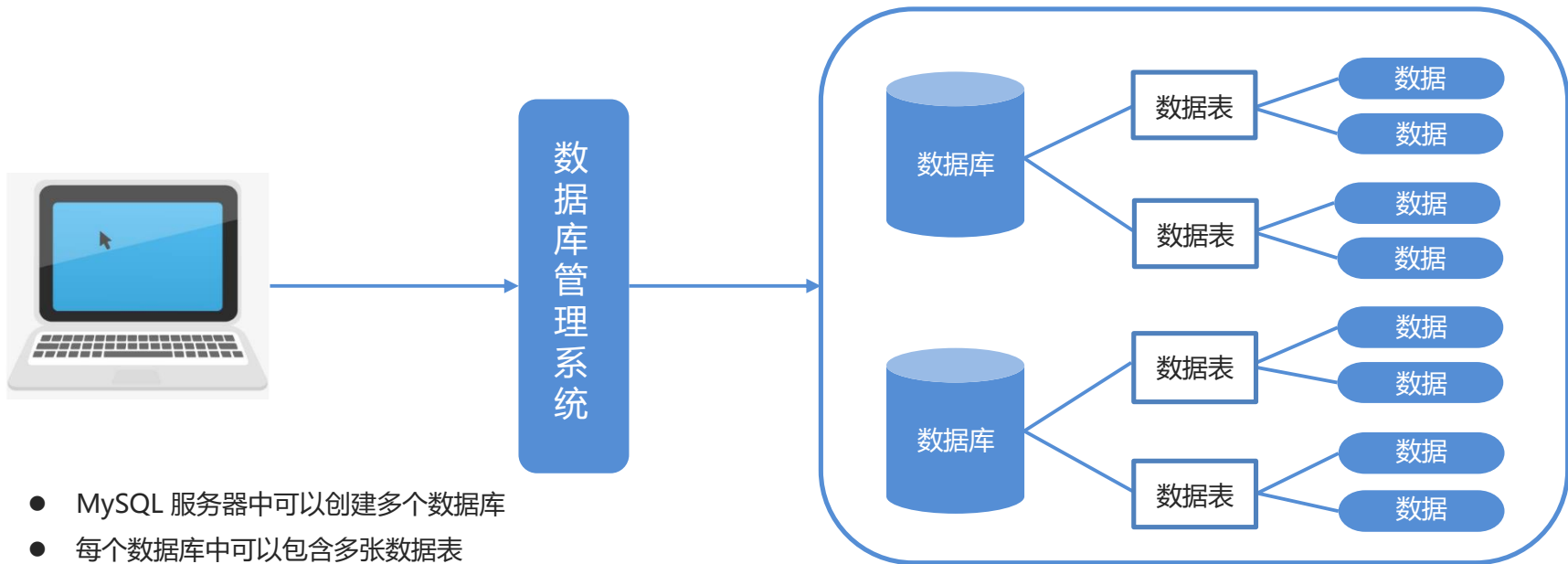
目录 Contents

- ◆ 数据库的基本概念
- ◆ DDL 操作数据库和数据表
- ◆ DML 表数据的增删改
- ◆ DQL 表数据的查询
- ◆ 约束



DDL 操作数据库和数据表

数据库、数据表、数据的关系



- MySQL 服务器中可以创建多个数据库
- 每个数据库中 can 包含多张数据表
- 每个数据表中可以存储多条数据记录
- 客户端通过数据库管理系统来操作 MySQL 数据库



SQL 的介绍

- SQL(Structured Query Language): 结构化查询语言。其实就是定义了操作所有关系型数据库的一种规则。

- 通用语法规则

SQL 语句可以单行或多行书写, 以分号结尾

可使用空格和缩进来增强语句的可读性

MySQL 数据库的 SQL 语句不区分大小写, 关键字建议使用大写

单行注释: -- 注释内容 #注释内容(MySQL特有)

多行注释: /* 注释内容 */

- SQL 分类

DDL(Data Definition Language): 数据定义语言。用来操作数据库, 表, 列等。

DML(Data Manipulation Language): 数据操作语言。用来对数据库中表的数据进行增删改。

DQL(Data Query Language): 数据查询语言。用来查询数据库中表的记录(数据)。

DCL(Data Control Language): 数据控制语言。用来定义数据库的访问权限和安全级别, 及创建用户。



DDL 操作数据库

- CRUD

C(Create): 创建

R(Retrieve): 查询

U(Update): 修改

D>Delete): 删除



DDL 查询数据库

- 查询所有数据库

```
SHOW DATABASES;
```

- 查询数据库的创建语句

```
SHOW CREATE DATABASE 数据库名称;
```



DDL 创建数据库

- 创建数据库

```
CREATE DATABASE 数据库名称;
```

- 创建数据库(判断, 如果不存在则创建)

```
CREATE DATABASE IF NOT EXISTS 数据库名称;
```

- 创建数据库(指定字符集)

```
CREATE DATABASE 数据库名称 CHARACTER SET 字符集名称;
```



DDL 修改、删除、使用数据库

- 修改数据库(修改字符集)

```
ALTER DATABASE 数据库名称 CHARACTER SET 字符集名称;
```

- 删除数据库

```
DROP DATABASE 数据库名称;
```

- 删除数据库(判断, 如果存在则删除)

```
DROP DATABASE IF EXISTS 数据库名称;
```

- 使用数据库

```
USE 数据库名称;
```

- 查看当前使用的数据库

```
SELECT DATABASE();
```




DDL 查询数据表

- 查询所有的数据表

```
SHOW TABLES;
```

- 查询表结构

```
DESC 表名;
```

- 查询表字符集

```
SHOW TABLE STATUS FROM 库名 LIKE '表名';
```



DDL 创建数据表

- 创建数据表

```
CREATE TABLE 表名(  
    列名 数据类型 约束,  
    列名 数据类型 约束,  
    ...  
    列名 数据类型 约束  
);
```

- 数据类型

int: 整数类型

double: 小数类型

date: 日期类型。包含年月日, 格式 yyyy-MM-dd

datetime: 日期类型。包含年月日时分秒, 格式 yyyy-MM-dd HH:mm:ss

timestamp: 时间戳类型。包含年月日时分秒, 格式 yyyy-MM-dd HH:mm:ss

* 如果不给该列赋值、或赋值为null, 则默认使用当前系统时间自动赋值

varchar(长度): 字符串类型



DDL 操作数据库和数据表

DDL 创建数据表

- 复制数据表

```
CREATE TABLE 表名 LIKE 被复制的表名;
```



DDL 修改数据表

- 修改表名

```
ALTER TABLE 表名 RENAME TO 新表名;
```

- 修改表的字符集

```
ALTER TABLE 表名 CHARACTER SET 字符集名称;
```

- 单独添加一列

```
ALTER TABLE 表名 ADD 列名 数据类型;
```

- 修改某列的数据类型

```
ALTER TABLE 表名 MODIFY 列名 新数据类型;
```

- 修改列名和数据类型

```
ALTER TABLE 表名 CHANGE 列名 新列名 新数据类型;
```

- 删除某一列

```
ALTER TABLE 表名 DROP 列名;
```



DDL 删除数据表

- 删除数据表

```
DROP TABLE 表名;
```

- 删除数据表(判断, 如果存在则删除)

```
DROP TABLE IF EXISTS 表名;
```

目录 Contents

- ◆ 数据库的基本概念
- ◆ DDL 操作数据库和数据表
- ◆ DML 表数据的增删改
- ◆ DQL 表数据的查询
- ◆ 约束



DML 表数据的增删改

DML 新增表数据

- 给指定列添加数据

```
INSERT INTO 表名(列名1,列名2,...) VALUES (值1,值2,...);
```

- 给全部列添加数据

```
INSERT INTO 表名 VALUES (值1,值2,...);
```

- 批量添加数据

```
INSERT INTO 表名(列名1,列名2,...) VALUES (值1,值2,...), (值1,值2,...),...;
```

```
INSERT INTO 表名 VALUES (值1,值2,...), (值1,值2,...),...;
```

列名和值的数量以及数据类型要对应，除了数字类型，其他数据类型的数据都需要加引号(单引双引都行，推荐单引)。



DML 表数据的增删改

DML 修改表数据

- 修改表中的数据

```
UPDATE 表名 SET 列名1=值1,列名2=值2,... [WHERE 条件];
```

修改语句中必须加条件，如果不加条件，则会将所有数据都修改。



DML 表数据的增删改

DML 删除表数据

- 删除表中的数据

```
DELETE FROM 表名 [WHERE 条件];
```

删除语句中必须加条件，如果不加条件，则会将所有数据都删除。

目录 Contents

- ◆ 数据库的基本概念
- ◆ DDL 操作数据库和数据表
- ◆ DML 表数据的增删改
- ◆ DQL 表数据的查询
- ◆ 约束



DQL 表数据查询 —— 数据准备

```
-- 创建db1数据库
CREATE DATABASE db1;

-- 使用db1数据库
USE db1;

-- 创建数据表
CREATE TABLE product(
    id INT,          -- 商品编号
    NAME VARCHAR(20), -- 商品名称
    price DOUBLE,    -- 商品价格
    brand VARCHAR(10), -- 商品品牌
    stock INT,       -- 商品库存
    insert_time DATE  -- 添加时间
);

-- 添加数据
INSERT INTO product VALUES (1,'华为手机',3999,'华为',23,'2088-03-10'),
(2,'小米手机',2999,'小米',30,'2088-05-15'),
(3,'苹果手机',5999,'苹果',18,'2088-08-20'),
(4,'华为电脑',6999,'华为',14,'2088-06-16'),
(5,'小米电脑',4999,'小米',26,'2088-07-08'),
(6,'苹果电脑',8999,'苹果',15,'2088-10-25'),
(7,'联想电脑',7999,'联想',NULL,'2088-11-11');
```



DQL 表数据查询 —— 查询语法

```
SELECT      字段列表
FROM        表名列表
WHERE       条件列表
GROUP BY    分组字段
HAVING      分组后的过滤条件
ORDER BY    排序
LIMIT      分页
```



DQL 表数据查询 —— 查询全部

- 查询全部的表数据

```
SELECT * FROM 表名;
```

- 查询指定字段的表数据

```
SELECT 列名1,列名2,... FROM 表名;
```

- 去除重复查询

```
SELECT DISTINCT 列名1,列名2,... FROM 表名;
```

- 计算列的值(四则运算)

```
SELECT 列名1 运算符(+ - * /) 列名2 FROM 表名;
```

- 起别名查询

```
SELECT 列名 AS 别名 FROM 表名;
```



DQL 表数据查询 —— 条件查询

● 查询条件分类

符号	功能
>	大于
<	小于
>=	大于等于
<=	小于等于
=	等于
<> 或 !=	不等于
BETWEEN ... AND ...	在某个范围之内(都包含)
IN(...)	多选一
LIKE 占位符	模糊查询 _单个任意字符 %多个任意字符
IS NULL	是NULL
IS NOT NULL	不是NULL
AND 或 &&	并且
OR 或	或者
NOT 或 !	非, 不是

● 条件查询语法

```
SELECT 列名列表 FROM 表名 WHERE 条件;
```



DQL 表数据查询 —— 聚合函数查询

- 聚合函数的介绍

将一列数据作为一个整体，进行纵向的计算。

- 聚合函数的分类

函数名	功能
count(列名)	统计数量(一般选用不为null的列)
max(列名)	最大值
min(列名)	最小值
sum(列名)	求和
avg(列名)	平均值

- 聚合函数查询语法

```
SELECT 函数名(列名) FROM 表名 [WHERE 条件];
```



DQL 表数据查询 —— 排序查询

- 排序查询语法

```
SELECT 列名列表 FROM 表名 [WHERE 条件] ORDER BY 列名 排序方式,列名 排序方式,...;
```

排序方式：ASC-升序，DESC-降序

如果有多个排序条件，只有当前边的条件值一样时，才会判断第二条件。



DQL 表数据查询 —— 分组查询

- 分组查询语法

```
SELECT 列名列表 FROM 表名 [WHERE 条件] GROUP BY 分组列名  
[HAVING 分组后的条件过滤]  
[ORDER BY 排序列名 排序方式];
```



DQL 表数据查询 —— 分页查询

- 分页查询语法

```
SELECT 列名列表 FROM 表名  
[WHERE 条件]  
[GROUP BY 分组列名]  
[HAVING 分组后的条件过滤]  
[ORDER BY 排序列名 排序方式]  
LIMIT 当前页数,每页显示的条数;
```

当前页数: $(\text{当前页数} - 1) * \text{每页显示的条数}$

目录 Contents

- ◆ 数据库的基本概念
- ◆ DDL 操作数据库和数据表
- ◆ DML 表数据的增删改
- ◆ DQL 表数据的查询
- ◆ 约束

约束的介绍

- 什么是约束

对表中的数据进行限定，保证数据的正确性、有效性、完整性！
- 约束的分类

约束	作用
PRIMARY KEY	主键约束
PRIMARY KEY AUTO_INCREMENT	主键自增
UNIQUE	唯一约束
NOT NULL	非空约束
FOREIGN KEY	外键约束
FOREIGN KEY ON UPDATE CASCADE	外键级联更新
FOREIGN KEY ON DELETE CASCADE	外键级联删除



主键约束

- 主键约束的特点

主键约束默认包含非空和唯一两个功能。

一张表只能有一个主键。

主键一般用于表中数据的唯一标识。

- 建表时添加主键约束

```
CREATE TABLE 表名(  
    列名 数据类型 PRIMARY KEY,  
    ...  
    列名 数据类型 约束  
);
```

- 删除主键约束

```
ALTER TABLE 表名 DROP PRIMARY KEY;
```

- 建表后单独添加主键约束

```
ALTER TABLE 表名 MODIFY 列名 数据类型 PRIMARY KEY;
```

主键自增约束

- 建表时添加主键自增约束

```
CREATE TABLE 表名(  
    列名 数据类型 PRIMARY KEY AUTO_INCREMENT,  
    ...  
    列名 数据类型 约束  
);
```

- 删除主键自增约束

```
ALTER TABLE 表名 MODIFY 列名 数据类型;
```

- 建表后单独添加主键自增约束

```
ALTER TABLE 表名 MODIFY 列名 数据类型 AUTO_INCREMENT;
```

MySQL 中的自增约束，必须配合键的约束一起使用！

唯一约束

- 建表时添加唯一约束

```
CREATE TABLE 表名(  
    列名 数据类型 UNIQUE,  
    ...  
    列名 数据类型 约束  
);
```

- 删除唯一约束

```
ALTER TABLE 表名 DROP INDEX 列名;
```

- 建表后单独添加唯一约束

```
ALTER TABLE 表名 MODIFY 列名 数据类型 UNIQUE;
```

非空约束

- 建表时添加非空约束

```
CREATE TABLE 表名(  
    列名 数据类型 NOT NULL,  
    ...  
    列名 数据类型 约束  
);
```

- 删除非空约束

```
ALTER TABLE 表名 MODIFY 列名 数据类型;
```

- 建表后单独添加非空约束

```
ALTER TABLE 表名 MODIFY 列名 数据类型 NOT NULL;
```




传智播客旗下高端IT教育品牌