

1、初识MySQL

笔记本: Java17&18班上课笔记

创建时间: 2021/3/6 8:22

更新时间: 2021/3/6 11:59

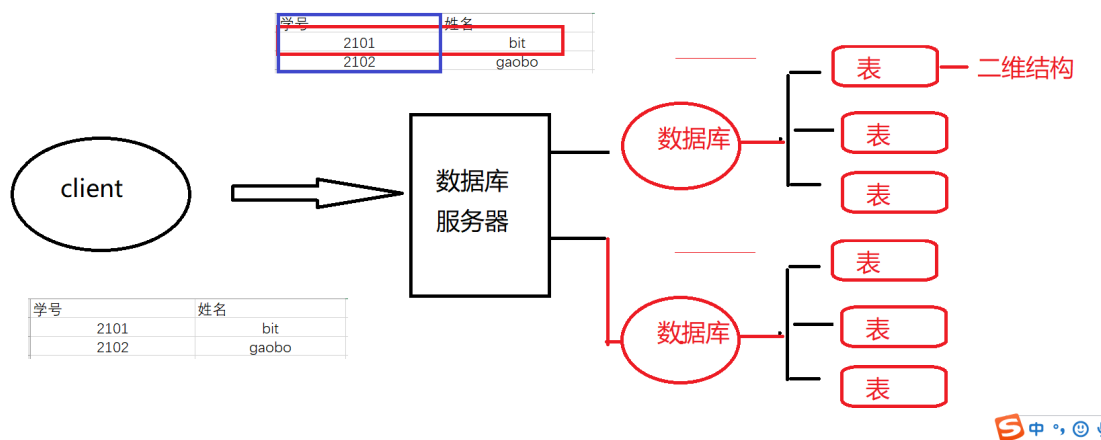
作者: 1262913815@qq.com

初识MySQL

1、到底什么是MySQL?

MySQL数据库。数据库，存储数据的一种服务器。

发展史:



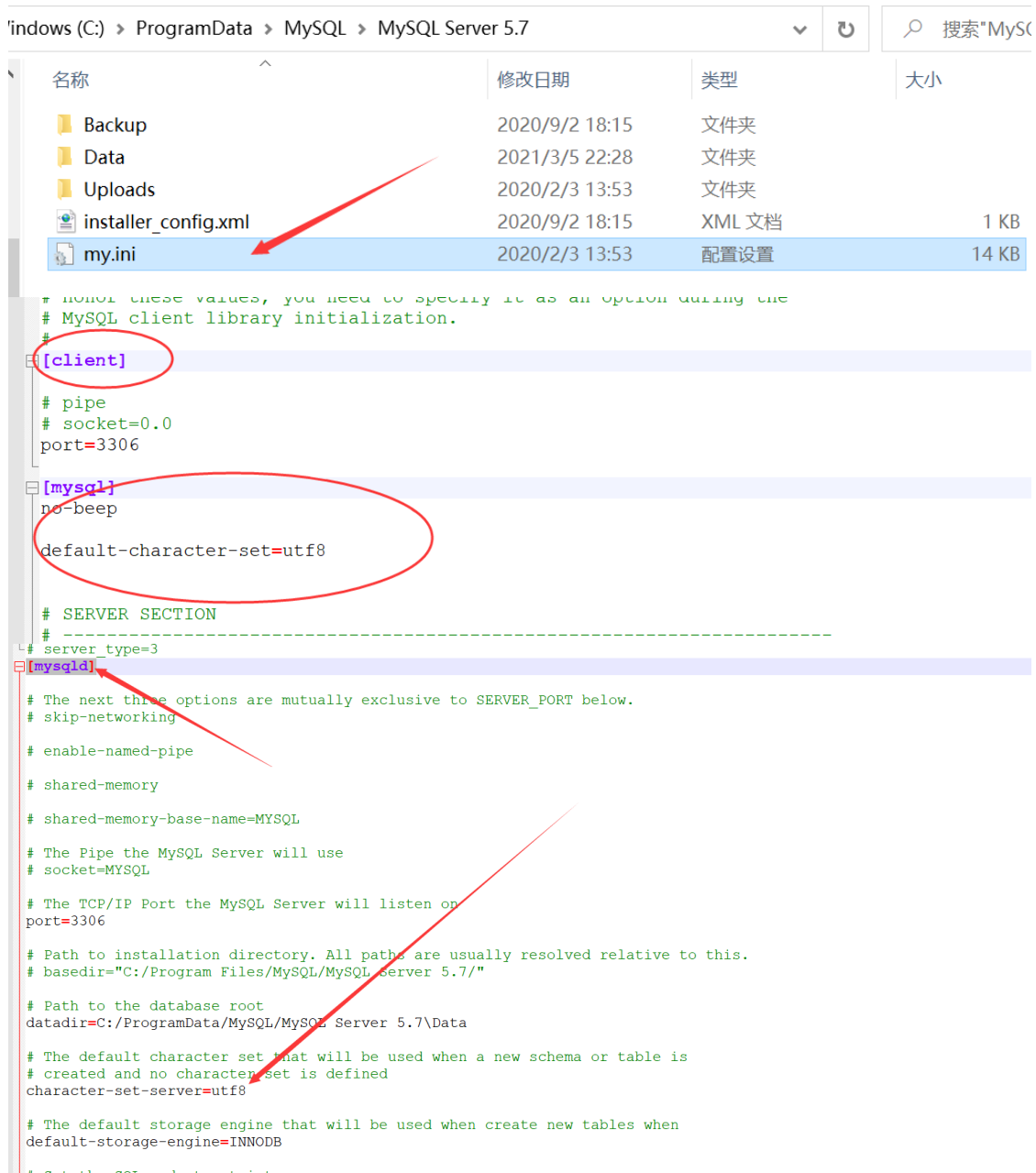
client:

- 1、MySQL自带的
- 2、cmd 配置环境变量
- 3、工具 Navicat

2、安装:

- 1、MySQL服务器
- 2、客户端
- 3、编码问题

C:\ProgramData\MySQL\MySQL Server 5.7



以上：就是安装MySQL 和 mysql的一些原理知识。

3、数据库 和 数据结构 有什么区别？

数据结构一门学科

数据库是一个软件，数据库底层也用到了数据结构，且很依赖于这个数据结构。后期会讲高级数据结构。B树 B+Tree

4、数据库的优点

文件保存数据有以下几个缺点：

- 文件的安全性问题
- 文件不利于数据查询和管理
- 文件不利于存储海量数据
- 文件在程序中控制不方便

文件存储的缺点就是数据库的优点

5、数据库的分类

1、关系型数据库(二维表格)

MySQL(开源, 免费的)、oracle(最好的数据库)、
SQL server

2、非关系型数据库(key-value)

Redis hbase

MariaDB

但是MySQL不适用于 大型的复杂问题

5、链接数据库：

mysql -u root -p111111

```
C:\Users\GaoBo>mysql -u root -p111111
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.30-log MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

```
-- 使用mysql数据库
use mysql;
```

```
-- 更新用户表的root账户，设置为任意ip都可以访问，
密码修改为123456
update user set
host="%",authentication_string=password('123456')
where
user="root";
-- 刷新权限
flush privileges;
```

6、SQL语言

SQL语言也是一种编程语言。SQL是用来操作数据库的数据的。

DDL(刚进公司的我们，也不会用到，用在项目开始的时候)

DML(使用频率最高的，CURD)

DCL(运维，对于我们来说使用频率不高)

7、什么是存储引擎？

存储引擎是：数据库管理系统如何存储数据、如何为存储的数据建立索引和如何更新、查询数据等技术的实现方法。

```
mysql> show engines
-> ;
+-----+-----+-----+-----+
| Engine | Support | Transactions |
+-----+-----+-----+-----+
| InnoDB | DEFAULT | Supports transactions, row-level locking, and foreign keys | YES | YES | |
| MRG_MYISAM | YES | Collection of identical MyISAM tables | NO | NO | NO |
| MEMORY | YES | Hash based, stored in memory, useful for temporary tables | NO | NO |
| BLACKHOLE | YES | /dev/null storage engine (anything you write to it disappears) | NO | NO |
| MyISAM | YES | MyISAM storage engine | NO | NO |
| CSV | YES | CSV storage engine | NO | NO |
```

```

NO |
| ARCHIVE | YES | Archive storage | NO | NO |
engine
NO |
| PERFORMANCE_SCHEMA | YES | Performance | NO | NO |
Schema
NO |
| FEDERATED | NO | Federated MySQL storage | NULL | NULL | NULL |
engine
+-----+-----+-----+-----+
+-----+-----+-----+-----+
9 rows in set (0.01 sec)

```

InnoDB:

MyISAM:

问题：InnoDB和 MyISAM 引擎的区别？？？

Transactions：事务

8、如何学号MySQL?

- 1、多上机
- 2、多编写SQL语句
- 3、学会使用编程语言来操作数据库

9、关于库的操作

1、显示数据库：show databases;

```

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| cash |
| eetest |
| examination_system |
| musicserver |
| mybatistest |
| mysql |
| performance_schema |
| problem |
| sakila |
| ssm |
| sys |
| test |
| test0301 |
| testbook |
| usermanger |
| world |
+-----+
17 rows in set (0.00 sec)

```

2、选中数据库

当你在使用某个数据库的时候，一定得先选中：use 数据库名；

```
mysql> use test0301;  
Database changed
```

3、查看当前选中的数据库下的表：show tables;

```
mysql> show tables;  
+-----+  
| Tables_in_test0301 |  
+-----+  
| account            |  
| classes            |  
| course             |  
| exam_result        |  
| score              |  
| student            |  
| test_user          |  
| user               |  
+-----+  
8 rows in set (0.00 sec)
```

4、删除数据库

```
mysql> drop database test;  
Query OK, 0 rows affected (0.06 sec)
```

```
mysql> show databases;  
+-----+  
| Database           |  
+-----+  
| information_schema |  
| cash               |  
| eetest             |  
| examination_system |  
| musicserver        |  
| mybatistest        |  
| mysql              |  
| performance_schema |  
| problem            |  
| sakila             |  
| ssm                |  
| sys                |  
| test0301           |  
| testbook           |  
| usermanger         |  
| world              |  
+-----+  
16 rows in set (0.00 sec)
```

5、创建数据库

```
CREATE DATABASE [IF NOT EXISTS] db_name [create_specification [,  
create_specification] ...]
```

create_specification:

[DEFAULT] CHARACTER SET charset_name -- 字符编码

[DEFAULT] COLLATE collation_name -- 校验规则 字符串的比较？

```
mysql> create database if not exists testJava17_18;  
Query OK, 1 row affected (0.00 sec)
```

```
mysql> show databases;
```

Database
information_schema
cash
eetest
examination_system
musicserver
mybatistest
mysql
performance_schema
problem
sakila
ssm
sys
test0301
testbook
testjava17_18
usermanger
world

17 rows in set (0.00 sec)

utf8mb4其实在mysql当中 就是utf-8

6、常用的数据类型,在创建表的时候,表中的字段的数据类型。

数值类型:

整型 :

BIT (M) bit BIT(3)

INT

浮点型:

DECIMAL(M,D): 小数

M:代表有效长度。13.45 的长度是4

D: 小数点后几位?

例如: DECIMAL(4,2) 能否存储这样的数据: 123.4 不可以。实际上会被解析为: 123.40 你的M变为了5

字符串类型:

VARCHAR(SIZE): varchar(4): 4个字节 String

TEXT 65535

日期类型:

DATETIME: 8个字节 表示的时间范围: 1000-9999年 缺点: 不会进行时区 的检索和转换。

TIMESTAMP: 4个字节。可以进行时区 的检索和转换 , 缺点: 1970-2038年。

7、得先回表的操作:

1、创建表

```
mysql> drop table if exists stu_test;  
Query OK, 0 rows affected, 1 warning (0.01 sec)
```

```
mysql> create table stu_test(  
-> ID INT,  
-> NAME VARCHAR(10),  
-> password VARCHAR(10),  
-> age INT,  
-> sex VARCHAR(1),  
-> birthday timestamp,  
-> amout decimal(6,2),  
-> resume TEXT  
-> );  
Query OK, 0 rows affected (0.08 sec)
```

```
mysql> insert into stu_test  
values(1,'bit','111',10,'男','2021-03-06',12.4,'今天  
很开心,学了mysql,博哥教 的');  
Query OK, 1 row affected (0.00 sec)
```



```
mysql> select * from stu_test;
```

ID	NAME	password	age	sex	birthday	amount	resume
1	bit	111	10	男	2021-03-06 00:00:00	12.40	今天很开心, 学了mysql, 博哥教的

```
1 row in set (0.00 sec)
```

```
mysql> insert into stu_test
values(2,'bit2','1112',102,'男','2021-03-06
11:40:42',12.4,'今天很开心, 学了mysql, 博哥教的');
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from stu_test;
```

ID	NAME	password	age	sex	birthday	amount	resume
1	bit	111	10	男	2021-03-06 00:00:00	12.40	今天很开心, 学了mysql, 博哥教的
2	bit2	1112	102	男	2021-03-06 11:40:42	12.40	今天很开心, 学了mysql, 博哥教的

```
2 rows in set (0.00 sec)
```

2、显示表

```
mysql> show tables;
```

Database	Table
mysql	tables

```
| Tables_in_testjava17_18 |
+-----+
| stu_test                  |
+-----+
1 row in set (0.00 sec)
```

3、删除表

```
mysql> drop table if exists stu_test;
Query OK, 0 rows affected (0.02 sec)
```

4、将数据类型放到表上

练习：

有一个商店的数据，记录客户及购物情况，有以下三个表组成：

商品goods(商品编号goods_id, 商品名goods_name, 单价unitprice, 商品类别category, 供应商provider)

客户customer(客户号customer_id, 姓名name, 住址address, 邮箱email, 性别sex, 身份证card_id)

购买purchase(购买订单号order_id, 客户号customer_id, 商品号goods_id, 购买数量nums)

```
mysql> use testjava17_18;
Database changed
mysql> drop table if exists goods;
Query OK, 0 rows affected, 1 warning (0.00 sec)
```

```
mysql> create table goods(
-> goods_id int,
-> goods_name varchar(20),
-> unitprice decimal(11,4),
-> category varchar(20),
```

```

-> provider varchar(20)
-> );
Query OK, 0 rows affected (0.04 sec)

mysql>
mysql>
mysql> drop table if exists customer;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> create table customer(
-> customer_id int,
-> name varchar(20),
-> address varchar(20),
-> email varchar(20),
-> sex varchar(1),
-> card_id varchar(20)
-> );
Query OK, 0 rows affected (0.04 sec)

mysql>
mysql>
mysql> drop table if exists purchase;
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql> create table purchase(
-> order_id int,
-> customer_id int,
-> goods_id int,
-> nums int
-> );
Query OK, 0 rows affected (0.05 sec)

mysql> show tables;
+-----+
| Tables_in_testjava17_18 |
+-----+
| customer                  |
| goods                    |
| purchase                  |
| stu_test                 |
+-----+
4 rows in set (0.00 sec)

```

8、操作表（CURD）。

增、删、**查 (难)** 改

准备数据:

```
drop table if exists student;
create table student(
    id INT,
    sn INT,
    name VARCHAR(20),
    qq_mail VARCHAR(20)
);
```

1、插入:

```
-- 单行全列插入
insert into student
values(1,1001,'bit','12345@qq.com');
insert into student
values(2,1002,'bit2','123452@qq.com');
```

每次只能插入一行, 且每一个字段要和数据库对应

```
mysql> select * from student;
+-----+-----+-----+-----+
| id    | sn    | name  | qq_mail      |
+-----+-----+-----+-----+
| 1     | 1001  | bit   | 12345@qq.com |
| 2     | 1002  | bit2  | 123452@qq.com |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

每一行就是一个学生, 每一列就是一个学生的某个属性。

对应类来说:

每一行就是一个学生对象, 每一列就是一个学生的某个属性。

```
class student {
    private int id;
    private int sn;
```

```

        private String name;
        private String qq_mail;
    }

```

等后期学到mybatis，框架的时候。

指定列，多行插入

```

mysql> insert into student(id,sn,name)
values(3,1003,'gaobo'),(4,1004,'gaobo2');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

```

```

mysql> select * from student;
+-----+-----+-----+-----+
| id    | sn    | name  | qq_mail      |
+-----+-----+-----+-----+
| 1     | 1001  | bit   | 12345@qq.com |
| 2     | 1002  | bit2  | 123452@qq.com|
| 3     | 1003  | gaobo | NULL          |
| 4     | 1004  | gaobo2| NULL          |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

```

如果没有指定某个字段的值，那么默认是NULL.但是注意，后期会讲约束：

查询：

1、全列查询：select * from studnt; *:当前表当中的所有的子段

```

mysql> select * from student;
+-----+-----+-----+-----+
| id    | sn    | name  | qq_mail      |
+-----+-----+-----+-----+
| 100   | 10000 | 唐三藏 | NULL          |
| 101   | 10001 | 孙悟空 | 11111         |
| 102   | 20001 | 曹孟德 | NULL          |
| 103   | 20002 | 孙仲谋 | NULL          |
+-----+-----+-----+-----+

```

4 rows in set (0.00 sec)

准备数据表:

```
DROP TABLE IF EXISTS exam_result;
CREATE TABLE exam_result (
    id INT,
    name VARCHAR(20),
    chinese DECIMAL(3,1),
    math DECIMAL(3,1),
    english DECIMAL(3,1)
);
INSERT INTO exam_result (id,name, chinese, math,
english) VALUES
(1,'唐三藏', 67, 98, 56),
(2,'孙悟空', 87.5, 78, 77),
(3,'猪悟能', 88, 98, 90),
(4,'曹孟德', 82, 84, 67),
(5,'刘玄德', 55.5, 85, 45),
(6,'孙权', 70, 73, 78.5),
(7,'宋公明', 75, 65, 30);
```

全列查询

```
mysql> select * from exam_result;
+-----+-----+-----+-----+-----+
| id    | name   | chinese | math  | english |
+-----+-----+-----+-----+-----+
| 1     | 唐三藏 | 67.0    | 98.0  | 56.0    |
| 2     | 孙悟空 | 87.5    | 78.0  | 77.0    |
| 3     | 猪悟能 | 88.0    | 98.0  | 90.0    |
| 4     | 曹孟德 | 82.0    | 84.0  | 67.0    |
| 5     | 刘玄德 | 55.5    | 85.0  | 45.0    |
| 6     | 孙权   | 70.0    | 73.0  | 78.5    |
| 7     | 宋公明 | 75.0    | 65.0  | 30.0    |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

指定列查询:

select **id,name** from exam_result;

```
mysql> select id,name from exam_result;
```

id	name
1	唐三藏
2	孙悟空
3	猪悟能
4	曹孟德
5	刘玄德
6	孙权
7	宋公明

查询字段为表达式

```
mysql> select id,name,10 from exam_result;
```

id	name	10
1	唐三藏	10
2	孙悟空	10
3	猪悟能	10
4	曹孟德	10
5	刘玄德	10
6	孙权	10
7	宋公明	10

7 rows in set (0.00 sec)

将所有的数学成绩+10

```
mysql> select id,name,math+10 from exam_result;
```

id	name	math+10
1	唐三藏	108.0
2	孙悟空	88.0
3	猪悟能	108.0
4	曹孟德	94.0
5	刘玄德	95.0
6	孙权	83.0
7	宋公明	75.0

7 rows in set (0.01 sec)

查询总分

```
mysql> select id,name,math+chinese+english from exam_result;
```

id	name	math+chinese+english
1	唐三藏	221.0
2	孙悟空	242.5
3	猪悟能	276.0
4	曹孟德	233.0
5	刘玄德	185.5
6	孙权	221.5
7	宋公明	170.0

7 rows in set (0.00 sec)

起别名: as 起别名 : XXX as 别名

```
mysql> select id,name,math+chinese+english as 总分 from exam_result;
```

id	name	总分
1	唐三藏	221.0
2	孙悟空	242.5
3	猪悟能	276.0
4	曹孟德	233.0
5	刘玄德	185.5
6	孙权	221.5
7	宋公明	170.0

7 rows in set (0.00 sec)

```
mysql> select id,name,math as 数学成绩 from exam_result;
```

id	name	数学成绩
1	唐三藏	98.0
2	孙悟空	78.0
3	猪悟能	98.0
4	曹孟德	84.0

5	刘玄德	85.0
6	孙权	73.0
7	宋公明	65.0

7 rows in set (0.00 sec)

也可以给表起别名。

```
mysql> select id,name,math as 数学成绩 from
exam_result as exam;
```

id	name	数学成绩
1	唐三藏	98.0
2	孙悟空	78.0
3	猪悟能	98.0
4	曹孟德	84.0
5	刘玄德	85.0
6	孙权	73.0
7	宋公明	65.0

7 rows in set (0.00 sec)

起别名的作用：就是当，列名 或者 表名 太复杂了 比如太长了 其此时 就可以用别名代替！！！！

