---------------------------------------------------

计划课程安排：

FUNDAMENTAL(7天)         该阶段已结束

OOP(11天)           该阶段已结束

JAVASE01(10天) 该阶段已结束

JAVASE02(12天) 该阶段已结束

WEBBASIC(11天) 该阶段已结束

JQUERY(2天) 该阶段已结束

DATABASE(6天) ←当前为该阶段第1天

JDBC(3天)

SERVLETJSP(12天)

SPRINGMYBATIS01(8天)

AJAX(2天)

SPRINGMYBATIS02(12天)

LINUX(2天)

NGINX(2天)

REDIS(2天)

T-STORE(5天)

预计合计：107天

预计剩余：53天

注：上述信息随时可能发生更改，仅供参考。

---------------------------------------------------

**2019年5月16日**

**Java就业班-第64天**

**8:30实训室2-早自习**

由于投影仪又出故障了，晨讲没有办法演示，所以今天跳过了。

……

快要上课了，保全的人还是没有来，准点上课估计够呛。

不过也不影响，我今天还是先自学吧。

**9:03实训室2-正课**

还是赶在上课之前修好了。

讲师还是在讲JQ，我还是继续自学JDBC吧。

复习Statement:

• Statement：用于执行不带参数的简单SQL语句；

• PreparedStatement（从 Statement 继承）：用于执行带或不带参数的预编译SQL语句；

• CallableStatement（从PreparedStatement 继承）：用于执行数据库存储过程的调用。

关于JDBC事务的概念。

事务指的是，一组要么同时执行成功，要么同时执行失败的SQL语句。是数据库操作的一个执行单元。

·事务的四大特点（ACID）

-atomicity(原子性)

表示一个事务内所有的操作是一个整体，要么全部成功，要么全部失败。

-consistency(一致性)

表示一个事务内有一个操作失败时，所有的更改过的数据必须回滚到修改前的状态；

-isolation(隔离性)

事务查看数据时数据所处的状态，要么是另一并发事务修改他之前的状态，要么是另一事物修改它之后的状态，事务不会查看中间状态的数据。

-durability(持久性)

持久性事务完成后，他对于系统的影响是永久性的。

测试代码：

package jdbc;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.SQLException;

/\*\*

\* 测试事务

\* @author Administrator

\*

\*/

public class demo06 {

public static void main(String[] args) {

Connection cn = null;

PreparedStatement ps1 = null ;

PreparedStatement ps2 = null ;

try {

//加载驱动类

Class.forName("com.mysql.jdbc.Driver");

cn = DriverManager.getConnection("jdbc:mysql://localhost:3306/kamuisdatabase","root","123456");

System.out.println("数据库已连接...");

cn.setAutoCommit(true);//JDBC中默认是true自动提交

ps1 = cn.prepareStatement("insert into user (name,password,age) values (?,?,?)");

ps1.setObject(1, "kamui");

ps1.setObject(2, "kam123");

ps1.setObject(3, 25);

ps1.execute();

System.out.println("ps1:插入一个用户。");

try {

Thread.sleep(3000);

} catch (InterruptedException e) {

e.printStackTrace();

}

ps2 = cn.prepareStatement("insert into user (name,password) values (?,?,?)");

ps2.setObject(1, "kamui");

ps2.setObject(2, "kam123");

ps2.execute();

System.out.println("ps2:插入一个用户。");

} catch (SQLException e) {

e.printStackTrace();

} catch (ClassNotFoundException e) {

e.printStackTrace();

try {

cn.rollback();//回滚操作

} catch (Exception e2) {

}

}finally {//按照后开先关的顺序，关闭数据库

//注意一定要将异常捕获分开写，否则会导致抛出异常的时候其他链接无法没有执行关闭方法

if(ps1!=null) {

try {

ps1.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(ps2!=null) {

try {

ps1.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

if(cn!=null) {

try {

cn.close();

} catch (SQLException e) {

e.printStackTrace();

}

}

}

}

}

**9:30实训室2-正课**

JQ的课程好像是结束了？

讲师也开始讲数据库的课程了，有点出乎意料，先听一下看看讲师讲的效果怎么样吧。

首先介绍了一下行业相关知识，着重讲解了一下开闭源，和市面上流行的几款数据库。

由于讲师还是讲前端的这个人，他的习惯还是最后写笔记，所以课程笔记还是最后补上。

**10:20实训室2-正课**

讲师公布了新的课程计划，数据库内容会讲5天，然后JDBC会讲3天左右。

…………

接着就是讲解并通过控制台练习了SQL的基本执行语句。

…………

有点困，看样子昨晚没睡好，我得休息一下了。

**11:50实训室2-午休时间**

基本上一上午就是对数据库进行详细的介绍。

下午的具体课程应该就是练习MySQL数据库指令了。

**14:35实训室2-正课**

练习：

1. 创建2个数据库分别是mydb1和mydb2
2. 在mydb1里面创建员工表emp 姓名 年龄 工龄salary
3. 在mydb2里面创建英雄表hero 姓名 类型type 价格price
4. 删除两个表然后再删除两个数据库

代码：

//启动控制台

win+R → cmd.exe

//登录

mysql -h127.0.0.1 -uroot -p123456

//创建数据库mydb1

create database mydb1 ;

//使用mydb1

use mydb1 ;

//创建员工表emp

create table emp (

name varchar(10) ,

age int ,

salary int

) ;

//创建数据库mydb2

create database mydb2 ;

//使用mydb2

use mydb2 ;

//创建英雄表

create table hero (

name varchar(10) ,

type varchar(2) ,

price int

) ;

//删除表hero

drop table hero ;

//切换至mydb1数据库；

use mydb1;

//删除表emp

drop table emp ;

//删除数据库

drop database mydb1 ;

drop database mydb2 ;

以上

**15:14实训室2-正课**

继续上课。

//修改表名-格式：rename table 原名 to 新名 ：

rename table t2 to tdb1 ;

//修改引擎和字符集格式：alter table 表名 engine=myisam/innodb charset=utf8/gbk ;

alter table tdb1 engine=innodb charset=utf8 ;

//添加表字段-格式：

//最后面格式：alter table 表名 add 字段名 字段类型；

//最前面格式：alter table 表名 add 字段名 字段类型 first ;

//某字段后面：alter table 表名 add 字段名 字段类型 after 字段名 ;

alter table tdb1 add gender varchar(5);

alter table tdb1 add id int first;

alter table tdb1 add salary int after name;

//删除表字段-格式：alter table 表名 drop 字段名 ；

alter table tdb1 drop salary ;

//修改字段名和类型-格式：alter table 表名 change 原名 新名 新类型；

alter table tdb1 change id pid int ;

//修改字段类型和位置-格式：alter table 表名 modify 字段名 新类型 first/after xxx ;

alter table tdb1 modify gender int first;

alter table tdb1 modify gender int after name;

练习：

1. 创建数据库mydb3 指定字符集utf8
2. 创建table\_emp表 只有id字段 指定引擎myisam 字符集为gbk;
3. 修改表名为t\_emp
4. 修改表引擎为innodb 字符集为utf8
5. 在最后面添加name字段
6. 在name前面添加age字段
7. 在age后面添加salary字段
8. 修改salary字段名为sal
9. 修改age字段到最后面
10. 删除sal字段
11. 删除表
12. 删除mydb3

代码：

//启动控制台

win+R → cmd.exe

//登录

mysql -h127.0.0.1 -uroot -p123456

//创建数据库

create database mydb3 charset=utf8 ;

//使用数据库

use mydb3 ;

//创建表

create table table\_emp (id int) engine=myisam charset=gbk ;

//修改表名

rename table table\_emp to t\_emp ;

//修改引擎和字符集

alter table t\_emp engine=innodb charset=utf8 ;

//在最后面添加name字段

alter table t\_emp add name varchar(10) ;

//在name前面添加age字段

alter table t\_emp add age int after id ;

//在age后面添加salary字段

alter table t\_emp add salary int after age ;

//修改salary字段名为sal

alter table t\_emp change salary sal int ;

//修改age字段到最后面

alter table t\_emp modify age int after name ;

//删除sal字段

alter table t\_emp drop sal ;

//删除表

drop table t\_emp ;

//删除数据库

drop database mydb3 ;

以上。

**16:23实训室2-正课**

继续上课。

//启动控制台

win+R → cmd.exe

//登录

mysql -h127.0.0.1 -uroot -p123456

drop database db1 ;

create database db1 character set utf8 ;

use db1 ;

create table person(name varchar(10),age int) ;

//插入数据-全表插入格式：insert into 表名 values();

insert into person values('Kamui',25);

//插入数据-指定字段格式：insert into 表名(字段名1，字段名2)values(值1，值2)；

insert into person(name) values('Subaru');

//插入数据-批量插入格式：

insert into person values('Luna',17),('Tsukasa',15),('Lux',16);

insert into person (name)values('Sakura'),('Naruto'),('Sasuke');

//插入中文数据：

insert into person values('卡穆依',25);

※如果报错，执行以下命令：

set names gbk;

//查询数据-格式：select 字段信息 from 表名 where 条件;

select \* from person;

select name from person;

select \* from person where age<30 ;

//修改数据-格式：update 表名 set 字段名=值 where;

update person set age=17 where name='Kamui';

练习：修改17岁的人名为Unknow。

代码：update person set name='Unknow' where age=17;

//删除数据-格式：delete from 表名 where 条件;

delete from person where name='Unknow';

delete from person where age>20;

delete from person;

练习：

1. 创建表hero表 id 名字 name 类型 type 价格 money;
2. 插入以下数据

1 Kamui 圣骑 25000

2 Subaru 火法 25000

3 Luna 冰法 25000

4 Tsukasa 勇者 16000

5 Clearlove 盲仔 77777

6 Boy♂Next♂Door Deep♂Dark♂Fantasy 6969696

1. 修改所有25000为16000
2. 修改火法和冰法为法爷
3. 删除价格为77777的英雄
4. 删除价格大于16000的英雄
5. 添加性别gender字段在name的后面
6. 修改所有英雄的性别为女
7. 删除所有数据
8. 删除表

代码：

//创建表

create table hero(

id int,

name varchar(20) ,

type varchar(20) ,

money int

);

//插入数据

insert into hero values

(1, 'Kamui' , '圣骑', 25000),

(2, 'Subaru' , '火法', 25000),

(3, 'Luna', '冰法', 25000),

(4, 'Tsukasa', '勇者', 16000),

(5, 'Clearlove', '盲仔', 77777),

(6, 'Boy♂Next♂Door', 'Deep♂Dark♂Fantasy', 6969696);

//修改数据

update hero set money=16000 where money=25000;

update hero set type='法爷' where type='火法';

update hero set type='法爷' where type='冰法';

//删除数据

delete from hero where money=77777;

delete from hero where money>16000;

//添加数据字段

alter table hero add gender varchar(5) after name;

//修改数据

update hero set gender='女';

//删除全部

delete from hero;

drop table hero;

**17:50实训室2-正课**

这个讲师讲课特别慢，但同时也是边讲边练习，比起讲前端的时候要稍微好一点点了，如果学员很用心的话，应该是可以学的比较熟练吧。只不过知识层面始终还是停留在表层。

今天的课似乎到这里就结束了，最后把笔记补一下就收拾东西回宿舍了。

今日课程笔记：

###数据库

- 之前学习的通过IO流操作文件的形式对数据进行增删改查 存在很多弊端

1. 效率低

2. 一般只能保存小量数据

3. 只能保存文本数据

- 什么是DBMS： DataBaseManagementSystem，数据库管理系统（数据库管理软件），作用就是负责对数据进行增删改查的软件，常见的DBMS： MySQL、Oracle、DB2、SQLServer、SQLite等

- 数据库分类：

1. 关系型数据库： 以表为单位保存数据，经过数学理论验证可以保存现实生活中存在的任何关系

2. 非关系型数据库： 以键值对形式保存数据，一般用于解决特殊场景，如数据缓存。

- 开源和闭源

1. 开源：公开源代码， 免费 ， 盈利方式：靠卖服务 ， 开源有大拿无偿维护升级。

2. 闭源：不公开源代码，收费， 盈利方式：靠卖产品+卖服务 ，闭源有大拿攻击 但是公司会花钱养一群人维护升级。

- 主流数据库软件介绍

1. MySQL： Oracle公司产品， 08年被Sun公司收购，09Sun被Oracle 拉里.埃里森

MariaDB 市场占有率第一

2. Oracle： Oracle公司产品 市场占有率第二 ，性能最高 价格最贵的数据库

3. SQLServer： 微软公司产品 排第三 ， 主要应用在微软整套解决方案中

4. DB2： IBM公司产品 主要应用在IBM整套解决方案中

5. sqlite： 轻量级数据库，只具备基础的增删改查操作

###SQL

- Structured Query Language：结构化查询语言，用户程序员和数据库软件进行交流的语言

###如何连接数据

mysql -uroot -p

###数据库相关

1. 查看所有数据库

- show databases;

2. 创建数据库

- 格式： create database 数据库名;

create database db1;

3. 查看数据库详情

- 格式： show create database 数据库名;

show create database db1;

4. 创建数据库 指定字符集

- 格式： create database 数据库名 character set utf8/gbk;

create database db2 character set gbk;

5. 删除数据库

- 格式： drop database 数据库名;

drop database db2;

6. 使用数据库

- 格式： use 数据库名;

use db1;

###表相关

- 前提一定使用了数据库，才能执行表相关的SQL

1. 创建表

- 格式：create table 表名(字段1名 字段1类型,字段2名 字段2类型,.....);

create table person(name varchar(5),age int);

- 练习：创建学生表(student) 字段： 学号id 姓名name 语文chinese 数学math 英语english

create table student(id int,name varchar(10),chinese int,math int,english int);

2. 查看所有表

- 格式： show tables;

show tables;

3. 查看表详情

- 格式： show create table 表名;

show create table person;

- 引擎：

1. innodb：默认，支持事物、外键等高级操作

2. myisam： 只支持基础的增删改查操作，不支持事物、外键等高级操作

###回顾

- 数据库相关

1. show databases;

2. create database db1;

3. show create database db1;

4. drop database db1;

5. use db1;

- 表相关

1. create table t1(name varchar(10),age int);

2. show tables;

3. show create table t1;

- 练习：

1. 创建2个数据库分别是mydb1 和 mydb2（字符集为gbk）

create database mydb1;

create database mydb2 character set gbk;

2. 在mydb1里面添加员工表emp 里面有姓名 年龄 工资sal

use mydb1;

create table emp(name varchar(10),age int,sal int);

3. 在mydb2里面添加英雄表hero 字段有姓名 年龄 英雄类型type

use mydb2;

create table hero(name varchar(10),age int,type varchar(10));

4. 删除两个数据库

drop database mydb1;

drop database mydb2;

###表相关 续

1. 查询表字段信息

- 格式： desc 表名;

desc person;

2. 删除表

- 格式： drop table 表名;

drop table person;

3. 创建表指定引擎和字符集

- 格式： create table 表名(字段1名 字段1类型,字段2名 字段2类型,.....) engine=innodb/myisam charset=gbk/utf8;

create table person(name varchar(10),age int)engine=myisam charset=gbk;

4. 修改表名

- 格式： rename table 原名 to 新名;

rename table person to t\_person;

5. 修改表引擎和字符集

- 格式： alter table 表名 engine=myisam/innodb charset=utf8/gbk;

alter table t\_person engine=innodb charset=gbk;

6. 添加表字段

- 格式： alter table 表名 add 字段名 字段类型; //最后

- 格式： alter table 表名 add 字段名 字段类型 first;//最前面

- 格式： alter table 表名 add 字段名 字段类型 after xxx;在xxx后面

alter table t\_person add sal int;

alter table t\_person add id int first;

alter table t\_person add gender varchar(5) after name;

7. 删除表字段

- 格式： alter table 表名 drop 字段名;

alter table t\_person drop gender;

8. 修改表名和字段类型

- 格式： alter table 表名 change 原名 新名 新类型;

alter table t\_person change sal salary int;

9. 修改字段类型和位置

- 格式： alter table 表名 modify 字段名 新类型 first/ after xxx;

alter table t\_person modify name varchar(5) after age;

- 练习：

1. 创建数据库mydb3 指定字符utf8 并使用

create database mydb3 character set utf8;

use mydb3;

2. 创建temp表 只有id字段 指定引擎为myisam 字符集为gbk

create table temp(id int) engine=myisam charset=gbk;

3. 修改表名为t\_emp ;

rename table temp to t\_emp;

4. 修改引擎为innodb 修改字符集为utf8

alter table t\_emp engine=innodb charset=utf8;

5. 在最后面添加name字段

alter table t\_emp add name varchar(10);

6. 在name前面添加age字段

alter table t\_emp add age int after id;

7. 在age后面添加工资sal字段

alter table t\_emp add sal int after age;

8. 修改sal字段名称为salary

alter table t\_emp change sal salary int;

9. 修改age字段到最后面

alter table t\_emp modify age int after name;

10. 删除salary字段

alter table t\_emp drop salary;

11. 删除表

drop table t\_emp;

12. 删除数据库

drop database mydb3;

###数据相关

create database db1 character set utf8;

use db1;

create table person(id int,name varchar(10),age int);

1. 添加数据

- 全表插入格式： insert into 表名 values (值1,值2,值3); 要求值得数量和顺序必须和表字段的数量、顺序一致

insert into person values(1,'Tom',18);

- 指定字段插入格式： insert into 表名 (字段1名,字段2名) values (值1,值2); 要求值得数量和顺序和前面指定的一致

insert into person (id,name) values(2,'Jerry');

- 批量插入数据

insert into person values(3,'aaa',10),(4,'bbb',11);

insert into person (id,name) values(5,'ccc'),(6,'ddd');

2. 查询数据

- 格式： select 字段信息 from 表名 where 条件;

select name from person;

select name,age from person where id<3;

3. 修改数据

- 格式： update 表名 set 字段名=值 where 条件;

update person set age=50 where id=2;

4. 删除数据

- 格式： delete from 表名 where 条件;

delete from person where name='Tom';

###乱码问题

insert into person values (10,'Kamui',17);

- 如果执行包含中文SQL 报错时 执行以下指令解决

set names gbk;

- 如果执行插入中文没有报错 但是查询出现乱码，此时把数据库删除 重新创建确保数据库字符集为utf8 表的字符集为utf8再次测试，如果还是解决不了问题，重新安装数据库或联系项目经理帮忙解决

###课程回顾

1. 数据库相关

- show databases;

- create database db1 character set utf8/gbk;

- show create database db1;

- drop database db1;

- use db1;

2. 表相关

- create table t1(name varchar(10),age int)engine=myisam/innodb charset=utf8/gbk;

- show tables;

- show create table t1;

- desc t1;

- drop table t1;

- rename table t1 to t2;

- alter table t1 engine=mysiam/innodb charset=utf8/gbk;

- alter table t1 add 字段名 类型 first/after xxx;

- alter table t1 drop 字段名;

- alter table t1 change 原名 新名 新类型;

- alter table t1 modify 字段名 新类型 first/after xxx;

3. 数据相关

- insert into t1 (name,age) values('Kamui',28),('Subaru',38);

- select name,age from t1 where id<10;

- update t1 set age=50 where id=5;

- delete from t1 where age<30;

**18:30宿舍-休息**

比起之前的前端课程，今天算是上了一天比较正常的课了。

由于我也才自学到JDBC的内容，而这边已经开始将数据库了的话，就还是跟着课程走吧，毕竟学费不能白交嘛。

昨天的评论区里也是有很多老哥给出了许多建议，也多亏了这些建议让我对现在的行情和主流技术有了更多的认识，在此表示感谢，一直以来都是拜众Acer老哥所赐，自认为少走了不少弯路，谢谢各位。

由于最近也有老哥们私信我关于培训班的问题，我才注意到我尽是专注学习和摸鱼，完全忘记记录机构情况的目的了，抱歉……我这就补上……

今天利用课余时间重新推算了一遍课程时间，如果项目经理没有说谎的话，那么下个月中旬确实就应该会有企业过来机构捞人了，而且很有可能是那种大批招人搬砖的企业，入职之后可能还会有培训，或者是以实习生的身份入职。但应该也只是开始捞人而已，课还是会继续上，从机构角度来看，无非就是尽早把学员推出去就业，尽快完成合约，这样后续的问题就可以交给企业去处理了。

具体的捞人流程关键应该还是看企业的面试，从我在其他班级的抽屉里发现的试卷和即将毕业的学员的陈述来看，如果是大批招人的企业，面试环节就是试卷考试。考过了就可以入职，只不过待遇堪忧。而真正意义上单独面试的企业则更偏向于从培优班或大数据班捞人，所以就业班=垃圾班的说法也不是毫无道理的。

哎~毕竟是商业行为，机构和企业要这么处理也无可厚非。

从目前了解到的情况来看，入学前销售所说的内容只是片面属实。提到的东西和流程确实都有，但完全不能抱以期望。至于所谓的就业班月薪6000+，培优班8000+，大数据1W+什么的都是特例，甚至于机构的关系并不大，而且所公布的数据真实性非常可疑。从机构毕业出去的90%都是搬砖或转行的说法一点都不为过。

其实，从个人的角度来说，想学编程不需要报班，遇到不懂的问题，百度和A站老哥比培训机构老师靠谱。

今日感想总结：数据库还是蛮有意思的。

今日评分：8/10分

今天就到这里，明天继续投稿。

感谢支持！

================================================================

**Acer程序员群：790482850**

**up主的java学习日记word文件及全部代码（随日记同步更新）：**

Adress：github.com/wt62635