

《数据库概论》 实验二 用户自定义完整性约束及使用高级程序设计语言访问数据库 实验报告

姓名：吴紫航 学号：171860659 联系方式：邮箱 401986905@qq.com

实验环境

[一句话介绍你使用的操作系统、软件版本]

Windows10

Mysql workbench8.0CE

IntelliJ IDEA 2019.2.4 x64

实验过程

[实验的详细过程，必须包含所有 SQL 语句和要求的实验截图]

1.

```
Create Table if not exists `Worker`(  
  `name` char(30) not null,  
  `id` int not null,  
  `age` int,  
  `salary` int,  
  `dep_id` int,  
  primary key (`id`)  
);
```

```
Create Table if not exists `department`(  
  `dep_name` char(30) not null,  
  `dep_id` int not null,  
  `leader_id` int,  
  primary key (`dep_id`)  
);
```

```
Create Table if not exists `project`(  
  `pro_name` char(30) not null,  
  `pro_id` int not null,  
  `dep_id` int,  
  primary key (`pro_id`)  
);
```

```
Create Table if not exists `job`(  
  `worker_id` int not null,  
  `pro_id` int not null,
```

```
`work_time` int,  
primary key(`worker_id`,`pro_id`)  
);
```

```
INSERT INTO department  
(`dep_name`,`dep_id`,`leader_id`)  
VALUES  
(`a_dep`,10001,171800010),  
(`b_dep`,10002,171800020),  
(`c_dep`,10003,171800030),  
(`d_dep`,10004,171800040)  
;  
Select * from department;
```

```
INSERT INTO job  
(`worker_id`,`pro_id`,`work_time`)  
VALUES  
(`171800011`,1,14),  
(`171800021`,2,23),  
(`171800031`,3,12),  
(`171800041`,4,35),  
(`171800012`,1,44),  
(`171800022`,2,35),  
(`171800032`,3,32),  
(`171800042`,4,23),  
(`171800013`,1,44),  
(`171800023`,2,27),  
(`171800033`,3,32),  
(`171800043`,4,33),  
(`171800010`,1,20),  
(`171800020`,2,17),  
(`171800030`,3,32),  
(`171800040`,4,5)  
;  
Select * from job;
```

```
INSERT INTO project  
(`pro_name`,`pro_id`,`dep_id`)  
VALUES  
(`a_pro`,1,10001),  
(`b_pro`,2,10002),  
(`c_pro`,3,10003),  
(`d_pro`,4,10004)  
;
```

Select * from project;

INSERT INTO worker

(`name`,`id`,`age`,`salary`,`dep_id`)

VALUES

('mike',171800011,25,10000,10001),
('kate',171800021,45,14000,10002),
('har',171800031,33,17000,10003),
('pete',171800041,36,21000,10004),
('woozie',171800012,31,23000,10001),
('kart',171800022,44,13000,10002),
('gedy',171800032,37,12000,10003),
('Dare',171800042,22,12000,10004),
('Ann',171800013,26,15000,10001),
('Belly',171800023,25,12000,10002),
('Carr',171800033,25,12000,10003),
('Dade',171800043,29,13000,10004),
('Andy',171800010,31,20000,10001),
('Bob',171800020,45,18000,10002),
('Carl',171800030,33,32000,10003),
('David',171800040,36,22000,10004)

;

Select * from worker;

Department

	dep_name	dep_id	leader_id
▶	a_dep	10001	171800010
	b_dep	10002	171800020
	c_dep	10003	171800030
	d_dep	10004	171800040
*	NULL	NULL	NULL

Job

	worker_id	pro_id	work_time
▶	171800010	1	20
	171800011	1	14
	171800012	1	44
	171800013	1	44
	171800020	2	17
	171800021	2	23
	171800022	2	35
	171800023	2	27
	171800030	3	32
	171800031	3	12
	171800032	3	32
	171800033	3	32
	171800040	4	5
	171800041	4	35
	171800042	4	23
	171800043	4	33
*	NULL	NULL	NULL

Project

	pro_name	pro_id	dep_id
▶	a_pro	1	10001
	b_pro	2	10002
	c_pro	3	10003
	d_pro	4	10004
*	NULL	NULL	NULL

Worker

	name	id	age	salary	dep_id
▶	Andy	171800010	31	20000	10001
	mike	171800011	25	10000	10001
	woozie	171800012	31	23000	10001
	Ann	171800013	26	15000	10001
	Bob	171800020	45	18000	10002
	kate	171800021	45	14000	10002
	kart	171800022	44	13000	10002
	Belly	171800023	25	12000	10002
	Carl	171800030	33	32000	10003
	har	171800031	33	17000	10003
	gedy	171800032	37	12000	10003
	Carr	171800033	25	12000	10003
	David	171800040	36	22000	10004
	pete	171800041	36	21000	10004
	Dare	171800042	22	12000	10004
	Dade	171800043	29	13000	10004
*	NAME	ID	AGE	SALARY	DEP_ID

2.

在建表时已经创建了非空约束和主键约束，先考虑外键约束即可
考虑到外键约束有：

- (1) department.leader_id 引用 worker.id
- (2) job.worker_id 和 job.pro_id 引用 worker.id 和 project.pro_id
- (3) project.dep_id 引用 department.dep_id
- (4) worker.dep_id 引用 department.dep_id

因此添加外键约束代码如下

```
alter table department add constraint constrain1 foreign key(leader_id) references worker(id)
on delete restrict on update cascade;
```

```
alter table job add constraint constrain2 foreign key(worker_id) references worker(id) on
delete restrict on update cascade;
```

```
alter table job add constraint constrain3 foreign key(pro_id) references project(pro_id) on
delete restrict on update cascade;
```

```
alter table project add constraint constrain4 foreign key(dep_id) references
department(dep_id) on delete restrict on update cascade;
```

```
alter table worker add constraint constrain5 foreign key(dep_id) references
department(dep_id) on delete restrict on update cascade;
```

✓	33	15:18:37	alter table department add constraint constrain1 foreign key(leader_id) references worker(id) on delete restrict on ...	4 row(s) affected	Records: 4 Duplicates: 0 Warnings: 0
✓	34	15:18:37	alter table job add constraint constrain2 foreign key(worker_id) references worker(id) on delete restrict on update...	16 row(s) affected	Records: 16 Duplicates: 0 Warnings: 0
✓	35	15:18:38	alter table job add constraint constrain3 foreign key(pro_id) references project(pro_id) on delete restrict on updat...	16 row(s) affected	Records: 16 Duplicates: 0 Warnings: 0
✓	36	15:18:38	alter table project add constraint constrain4 foreign key(dep_id) references department(dep_id) on delete restrict ...	4 row(s) affected	Records: 4 Duplicates: 0 Warnings: 0
✓	37	15:18:38	alter table worker add constraint constrain5 foreign key(dep_id) references department(dep_id) on delete restrict ...	16 row(s) affected	Records: 16 Duplicates: 0 Warnings: 0

3.

(1)违反非空约束(部门名非空)

```
INSERT INTO department
(dep_name`,`dep_id`,`leader_id`)
VALUES
(null,10005,171800010)
;
```

39 15:20:26 INSERT INTO department (dep_name`,`dep_id`,`leader_id`) VALUES (null,10005,171800010)

Error Code: 1048. Column 'dep_name' cannot be null

(2)违反主键约束 (工作的员工-项目有主键唯一性)

```
INSERT INTO job
(worker_id`,`pro_id`,`work_time`)
VALUES
('171800011',1,17)
;
```

40 15:21:08 INSERT INTO job (worker_id`,`pro_id`,`work_time`) VALUES ('171800011',1,17)

Error Code: 1062. Duplicate entry '171800011-1' for key 'PRIMARY'

(3)违反外键约束(含有主键值 pro_id 作为 job 表外键的 project 行不能被 delete)

```
delete from project
where pro_id=1;
```

41 15:21:31 delete from project where pro_id=1

Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('exp2`.`job`, CONSTRAINT...

(4)违反外键约束(含有主键值 id 作为 job 表和 department 表外键的 worker 行不能被 delete)

```
delete from worker
where id=171800010;
```

42 15:21:56 delete from worker where id=171800010

Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('exp2`.`department`, CONSTRAINT...

(5)违反外键约束(含有主键值 dep_id 作为 project 表和 worker 表外键的 department 行不能被 delete)

```
delete from department where dep_id=10001;
```

43 15:22:39 delete from department where dep_id=10001

Error Code: 1451. Cannot delete or update a parent row: a foreign key constraint fails ('exp2`.`project`, CONSTRAINT...

4.

(a)

delimiter \$\$

create trigger t1

before insert ON job

for each row

begin

if new.work_time>24

then set new.work_time=24;

```

        end if;
    end
    $$
delimiter ;

(b)
delimiter $$
create trigger t2
before insert ON job
for each row
begin
    update worker
    set salary=salary*1.05
    where new.worker_id=worker.id and worker.id Not in(
        select leader_id
        from department
    );

    update worker
    set salary=salary*1.08
    where new.worker_id=worker.id and worker.id in(
        select leader_id
        from department
    );
end
$$
delimiter ;

```

测试

```

• INSERT INTO job (`worker_id`,`pro_id`,`work_time`)
VALUES
('171800010',4,33),
('171800011',4,14),
('171800012',4,25);

```

	worker_id	pro_id	work_time
▶	171800010	1	20
	171800010	4	24 ←
	171800011	1	14
	171800011	4	14 ←
	171800012	1	44
	171800012	4	24 ←

工作时间大于 24 自动设置为 24

	name	id	age	salary	dep_id
▶	Andy	171800010	31	21600	10001
	mike	171800011	25	10500	10001
	woozie	171800012	31	24150	10001

薪水增加 5%，如果是部门负责人则再增加 3%（即总共 8%）

注：插入前的薪水见第一问的截图

5.

【创建用户和授权】

create user 'worker'@'localhost' identified by '991204';

grant select on exp2.worker to 'worker'@'localhost';

grant update(age) on exp2.worker to 'worker'@'localhost';

✓	4	21:48:05	create user 'worker'@'localhost' identified by '991204'	0 row(s) affected
✓	5	21:48:05	grant select on exp2.worker to 'worker'@'localhost'	0 row(s) affected
✓	6	21:48:05	grant update(age) on exp2.worker to 'worker'@'localhost'	0 row(s) affected

【登陆 worker 执行修改操作】

修改前

	name	id	age	salary	dep_id
▶	Andy	171800010	31	21600	10001
	mike	171800011	25	10500	10001
	woozie	171800012	31	24150	10001
	Ann	171800013	26	15000	10001
	Bob	171800020	45	18000	10002
	kate	171800021	45	14000	10002
	kart	171800022	44	13000	10002
	Belly	171800023	25	12000	10002
	Carl	171800030	33	32000	10003
	har	171800031	33	17000	10003
	gedy	171800032	37	12000	10003
	Carr	171800033	25	12000	10003
	David	171800040	36	22000	10004
	pete	171800041	36	21000	10004
	Dare	171800042	22	12000	10004
	Dade	171800043	29	13000	10004
•	NULL	NULL	NULL	NULL	NULL

update worker

set age=age+1;

update worker

set salary=salary+1000;

select * from worker;

修改职工的年龄和年薪

年龄有修改权限，但是薪水没有修改权限，因此修改结果如下图

✓	4	21:37:14	update worker set age=age+1 where id>0	16 row(s) affected Rows matched: 16 Changed: 16 Warnings: 0
✗	5	21:37:17	update worker set salary=salary+1000 where id>0	Error Code: 1143. UPDATE command denied to user 'worker'@'localhost' for column 'salary' in table 'worker'

	name	id	age	salary	dep_id
▶	Andy	171800010	32	21600	10001
	mike	171800011	26	10500	10001
	woozie	171800012	32	24150	10001
	Ann	171800013	27	15000	10001
	Bob	171800020	46	18000	10002
	kate	171800021	46	14000	10002
	kart	171800022	45	13000	10002
	Belly	171800023	26	12000	10002
	Carl	171800030	34	32000	10003
	har	171800031	34	17000	10003
	gedy	171800032	38	12000	10003
	Carr	171800033	26	12000	10003
	David	171800040	37	22000	10004
	pete	171800041	37	21000	10004
	Dare	171800042	23	12000	10004
	Dade	171800043	30	13000	10004
•	NULL	NULL	NULL	NULL	NULL

【删除】

revoke select on exp2.worker from 'worker'@'localhost';

revoke update(age) on exp2.worker from 'worker'@'localhost';

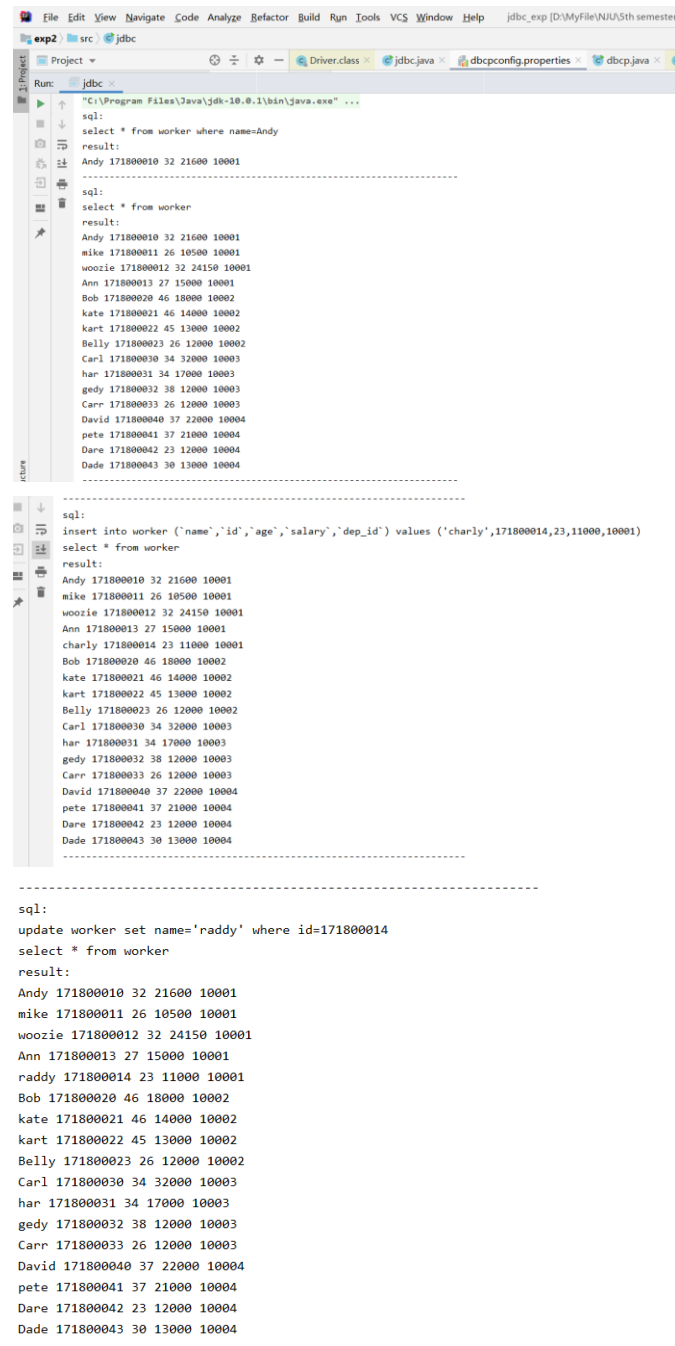
drop user 'worker'@'localhost' ;

✓	1	21:43:21	revoke select on exp2.worker from 'worker'@'localhost'	0 row(s) affected
✓	2	21:43:31	revoke update(age) on exp2.worker from 'worker'@'localhost'	0 row(s) affected
✓	3	21:45:45	drop user 'worker'@'localhost'	0 row(s) affected

6.

Jdbc: 一共八条 sql, 其中第一条为动态 sql 语句 (详见源代码 jdbc.java)

效果图如下



```
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help jdbc_exp [D:\MyFile\NJU\5th semester]
exp2 src jdbc
Project Driver.class jdbc.java dbcpconfig.properties dbcp.java
Run: jdbc
"C:\Program Files\Java\jdk-10.0.1\bin\java.exe" ...
sql:
select * from worker where name=Andy
result:
Andy 171800010 32 21600 10001
-----
sql:
select * from worker
result:
Andy 171800010 32 21600 10001
mike 171800011 26 10500 10001
woozie 171800012 32 24150 10001
Ann 171800013 27 15000 10001
Bob 171800020 46 18000 10002
kate 171800021 46 14000 10002
kart 171800022 45 13000 10002
Belly 171800023 26 12000 10002
Carl 171800030 34 32000 10003
har 171800031 34 17000 10003
gedy 171800032 38 12000 10003
Carr 171800033 26 12000 10003
David 171800040 37 22000 10004
pete 171800041 37 21000 10004
Dare 171800042 23 12000 10004
Dade 171800043 30 13000 10004
-----
sql:
insert into worker ('name','id','age','salary','dep_id') values ('charly',171800014,23,11000,10001)
select * from worker
result:
Andy 171800010 32 21600 10001
mike 171800011 26 10500 10001
woozie 171800012 32 24150 10001
Ann 171800013 27 15000 10001
charly 171800014 23 11000 10001
Bob 171800020 46 18000 10002
kate 171800021 46 14000 10002
kart 171800022 45 13000 10002
Belly 171800023 26 12000 10002
Carl 171800030 34 32000 10003
har 171800031 34 17000 10003
gedy 171800032 38 12000 10003
Carr 171800033 26 12000 10003
David 171800040 37 22000 10004
pete 171800041 37 21000 10004
Dare 171800042 23 12000 10004
Dade 171800043 30 13000 10004
-----
sql:
update worker set name='raddy' where id=171800014
select * from worker
result:
Andy 171800010 32 21600 10001
mike 171800011 26 10500 10001
woozie 171800012 32 24150 10001
Ann 171800013 27 15000 10001
raddy 171800014 23 11000 10001
Bob 171800020 46 18000 10002
kate 171800021 46 14000 10002
kart 171800022 45 13000 10002
Belly 171800023 26 12000 10002
Carl 171800030 34 32000 10003
har 171800031 34 17000 10003
gedy 171800032 38 12000 10003
Carr 171800033 26 12000 10003
David 171800040 37 22000 10004
pete 171800041 37 21000 10004
Dare 171800042 23 12000 10004
Dade 171800043 30 13000 10004
-----
```



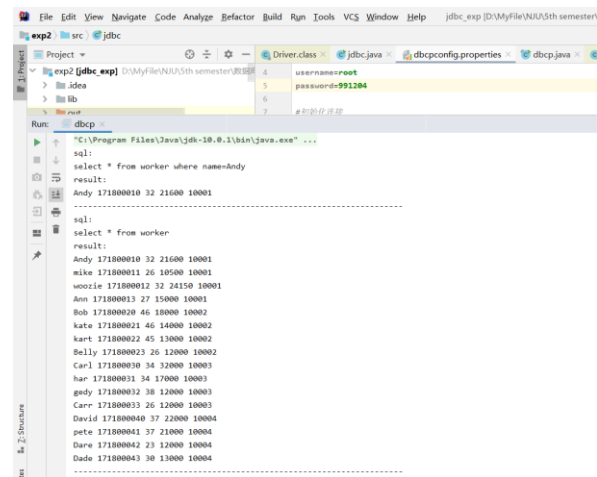
```

-----
sql:
delete from worker where id=171800014
select * from worker
result:
Andy 171800010 32 21600 10001
mike 171800011 26 10500 10001
woozie 171800012 32 24150 10001
Ann 171800013 27 15000 10001
Bob 171800020 46 18000 10002
kate 171800021 46 14000 10002
kart 171800022 45 13000 10002
Belly 171800023 26 12000 10002
Carl 171800030 34 32000 10003
har 171800031 34 17000 10003
gedy 171800032 38 12000 10003
Carr 171800033 26 12000 10003
David 171800040 37 22000 10004
pete 171800041 37 21000 10004
Dare 171800042 23 12000 10004
Dade 171800043 30 13000 10004
-----

```

Process finished with exit code 0

Dhcp: 与 jdbc 方式的 sql 代码相同(详见源代码 dhcp.java)
效果图如下,



```

-----
sql:
insert into worker ('name','id','age','salary','dep_id') values ('charly',171800014,23,11000,10001)
select * from worker
result:
Andy 171800010 32 21600 10001
mike 171800011 26 10500 10001
woozie 171800012 32 24150 10001
Ann 171800013 27 15000 10001
charly 171800014 23 11000 10001
Bob 171800020 46 18000 10002
kate 171800021 46 14000 10002
kart 171800022 45 13000 10002
Belly 171800023 26 12000 10002
Carl 171800030 34 32000 10003
har 171800031 34 17000 10003
gedy 171800032 38 12000 10003
Carr 171800033 26 12000 10003
David 171800040 37 22000 10004
pete 171800041 37 21000 10004
Dare 171800042 23 12000 10004
Dade 171800043 30 13000 10004
-----

```

```

-----
sql:
update worker set name='raddy' where id=171800014
select * from worker
result:
Andy 171800010 32 21600 10001
mike 171800011 26 10500 10001
woozie 171800012 32 24150 10001
Ann 171800013 27 15000 10001
raddy 171800014 23 11000 10001
Bob 171800020 46 18000 10002
kate 171800021 46 14000 10002
kart 171800022 45 13000 10002
Belly 171800023 26 12000 10002
Carl 171800030 34 32000 10003
har 171800031 34 17000 10003
gedy 171800032 38 12000 10003
Carr 171800033 26 12000 10003
David 171800040 37 22000 10004
pete 171800041 37 21000 10004
Dare 171800042 23 12000 10004
Dade 171800043 30 13000 10004
-----

```

```

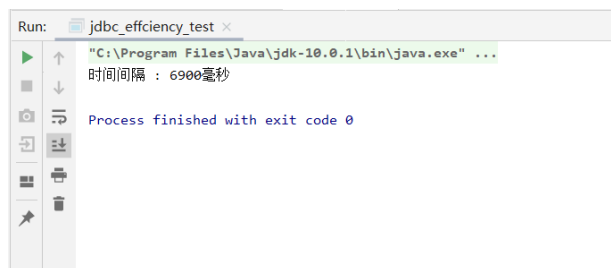
-----
sql:
delete from worker where id=171800014
select * from worker
result:
Andy 171800010 32 21600 10001
mike 171800011 26 10500 10001
woozie 171800012 32 24150 10001
Ann 171800013 27 15000 10001
Bob 171800020 46 18000 10002
kate 171800021 46 14000 10002
kart 171800022 45 13000 10002
Belly 171800023 26 12000 10002
Carl 171800030 34 32000 10003
har 171800031 34 17000 10003
gedy 171800032 38 12000 10003
Carr 171800033 26 12000 10003
David 171800040 37 22000 10004
pete 171800041 37 21000 10004
Dare 171800042 23 12000 10004
Dade 171800043 30 13000 10004
-----

```

Jdbc 和 dbcp 的比较

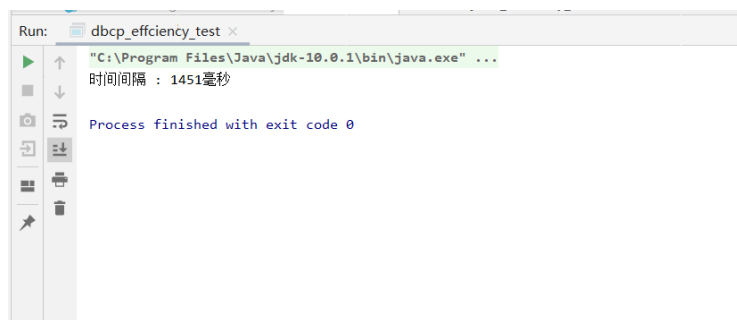
1. jdbc 测试流程 (详见 jdbc_efficiency_test.java)

- (1) 记录起始时间, 用 jdbc 方式连接数据库
- (2) 执行 **select * from worker where name= 'Andy'**
- (3) 关闭连接
- (4) 循环(1)–(3) 步骤 2000 次
- (5) 计算总时间间隔, 输出截图如下



2. dbcp 测试流程 (详见 dbcp_efficiency_test.java)

- (1) 记录起始时间, 用 dbcp 方式连接数据库
- (2) 执行 **select * from worker where name= 'Andy'**
- (3) 关闭连接
- (4) 循环(1)–(3) 步骤 2000 次
- (5) 计算总时间间隔, 输出截图如下



实验中遇到的困难及解决办法

[详细说明你认为本次实验中比较困难的地方，也可以对实验设计提出建议]

(1) 第四问触发器之前没有理解 delimiter 更换语句结束执行符号的作用

导致编译无法通过，后来通过查阅资料理解了 mysql 是单个语句编译执行，如果遇到默认结束符合“;”则会立刻执行当前语句，但是由于块只有 begin 没有 end，因此编译错误

(2) 第四问的条件块 if 忘记加上 endif，分析：写 c++ 写习惯了，忽略了需要加上 endif，后来不断尝试发现

(3) 第五问用 sql 语句新建 worker 用户后，不知道如何切换（使用 mysqlworkbench 平台）网上资料都是关于命令行的切换用户方式，后与同学陈振宇讨论知道，需要右键-edit connection 后新建 connection 然后登陆 worker 用户即可

(4) 不知道 idea 如何创建资源 resources 文件

解决方法：直接新建文件夹，然后右键-设置为 resources 文件

(5) 不知道如何导入 jar

解决方法：新建 lib 文件夹后直接放入即可

参考文献及致谢

[如果你参考了任何书籍、网页，或与他人进行了讨论，请在此注明]

参考网站：

https://blog.csdn.net/w_linux/article/details/79655073 关于 SQL 约束

<https://blog.csdn.net/nangeali/article/details/74999574> 外键约束

<https://www.cnblogs.com/qzhc/p/11190942.html> JDBC 教程

<https://www.cnblogs.com/fightingtong/p/8401898.html> 导入 jar 包方法

<https://www.cnblogs.com/sunseine/p/5947448.html> DBCP 教程

致谢：陈振宇

关于 MYSQL workbench 切换用户的方法的讨论