

## 实验步骤:

(1) 插入 department 数据:

```
1. INSERT INTO `course`.`department` (`d_no`, `d_name`) VALUES ('001', '信自学院');
2. INSERT INTO `course`.`department` (`d_no`, `d_name`) VALUES ('002', '管理学院');
3. INSERT INTO `course`.`department` (`d_no`, `d_name`) VALUES ('003', '机电学院');
4. INSERT INTO `course`.`department` (`d_no`, `d_name`) VALUES ('004', '外文学院');
5. INSERT INTO `course`.`department` (`d_no`, `d_name`) VALUES ('005', '法学院');
6. INSERT INTO `course`.`department` (`d_no`, `d_name`) VALUES ('006', '理学院');
```

(2) 插入 student 数据:

```
1. INSERT INTO `course`.`student` (`S_no`, `S_name`, `gender`, `d_no`, `phone`, `birthday`, `address`) VALUES ('201510101101', '刘晓东', '男', '001', '11000000001', '1999-5-10', '昆明');
2. INSERT INTO `course`.`student` (`S_no`, `S_name`, `gender`, `d_no`, `phone`, `birthday`, `address`) VALUES ('201510101102', '林慧', '女', '001', '11000000002', '1999-12-15', '上海');
3. INSERT INTO `course`.`student` (`S_no`, `S_name`, `gender`, `d_no`, `phone`, `birthday`, `address`) VALUES ('201511101103', '李远鹏', '男', '002', '11000000003', '1998-10-25', '北京');
4. INSERT INTO `course`.`student` (`S_no`, `S_name`, `gender`, `d_no`, `phone`, `birthday`, `address`) VALUES ('201511101104', '吴娜文', '女', '002', '11000000004', '1999-8-10', '昆明');
5. INSERT INTO `course`.`student` (`S_no`, `S_name`, `gender`, `d_no`, `phone`, `birthday`, `address`) VALUES ('201511101105', '刘智', '男', '003', '11000000005', '1999-5-8', '北京');
6. INSERT INTO `course`.`student` (`S_no`, `S_name`, `gender`, `d_no`, `phone`, `birthday`, `address`) VALUES ('201511101106', '赵立民', '男', '003', '11000000006', '1999-2-25', '上海');
7. INSERT INTO `course`.`student` (`S_no`, `S_name`, `gender`, `d_no`, `phone`, `birthday`, `address`) VALUES ('201513101111', '张亮亮', '男', '004', '11000000007', '1998-9-5', '上海');
8. INSERT INTO `course`.`student` (`S_no`, `S_name`, `gender`, `d_no`, `phone`, `birthday`, `address`) VALUES ('201513101116', '王丽萍', '女', '005', '11000000008', '1998-6-3', '重庆');
```

(3) 插入 teacher 数据:

```

1. INSERT INTO `course`.`teacher` (`t_no`, `t_name`, `d_no`) VALUES ('2001', '张华', '001');
2. INSERT INTO `course`.`teacher` (`t_no`, `t_name`, `d_no`) VALUES ('2002', '王明', '002');
3. INSERT INTO `course`.`teacher` (`t_no`, `t_name`, `d_no`) VALUES ('2003', '李萍', '003');
4. INSERT INTO `course`.`teacher` (`t_no`, `t_name`, `d_no`) VALUES ('2004', '田野', '004');
5. INSERT INTO `course`.`teacher` (`t_no`, `t_name`, `d_no`) VALUES ('2005', '赵瑾', '005');
6. INSERT INTO `course`.`teacher` (`t_no`, `t_name`, `d_no`) VALUES ('2006', '胡一民', '006');

```

(4) 插入 course 数据:

```

1. INSERT INTO `course`.`course` (`c_no`, `c_name`, `period`, `credit`, `c_type`, `t_no`) VALUES ('1001', '高等数学', '90', '6', '必修', '2006');
2. INSERT INTO `course`.`course` (`c_no`, `c_name`, `period`, `credit`, `c_type`, `t_no`) VALUES ('1002', '英语', '90', '6', '必修', '2004');
3. INSERT INTO `course`.`course` (`c_no`, `c_name`, `period`, `credit`, `c_type`, `t_no`) VALUES ('1003', '计算机基础', '70', '4', '必修', '2001');
4. INSERT INTO `course`.`course` (`c_no`, `c_name`, `period`, `credit`, `c_type`, `t_no`) VALUES ('1004', '数据库应用', '60', '4', '必修', '2001');
5. INSERT INTO `course`.`course` (`c_no`, `c_name`, `period`, `credit`, `c_type`, `t_no`) VALUES ('1005', '会计学', '100', '6', '必修', '2002');
6. INSERT INTO `course`.`course` (`c_no`, `c_name`, `period`, `credit`, `c_type`, `t_no`) VALUES ('1006', '经济学', '80', '5', '必修', '2003');
7. INSERT INTO `course`.`course` (`c_no`, `c_name`, `period`, `credit`, `c_type`, `t_no`) VALUES ('2001', '网页设计', '32', '1', '必修', '2001');

```

(5) 插入 choose 数据:

```

1. SET FOREIGN_KEY_CHECKS = 0;
2. INSERT INTO `course`.`choose` (`id`, `s_no`, `c_no`, `score`) VALUES ('1', '201510101101', '1001', '50');
3. INSERT INTO `course`.`choose` (`id`, `s_no`, `c_no`, `score`) VALUES ('2', '201510101101', '1002', '55');
4. INSERT INTO `course`.`choose` (`id`, `s_no`, `c_no`, `score`) VALUES ('3', '201510101102', '1001', '60');
5. INSERT INTO `course`.`choose` (`id`, `s_no`, `c_no`, `score`) VALUES ('4', '201510101102', '1002', '65');
6. INSERT INTO `course`.`choose` (`id`, `s_no`, `c_no`, `score`) VALUES ('5', '201510101103', '1001', '67');
7. INSERT INTO `course`.`choose` (`id`, `s_no`, `c_no`, `score`) VALUES ('6', '201510101103', '1005', '70');

```

```

8. INSERT INTO `course`.`choose` (`id`, `s_no`, `c_no`, `score`) VALUES ('7', '201510101104', '1005', '78');
9. INSERT INTO `course`.`choose` (`id`, `s_no`, `c_no`, `score`) VALUES ('8', '201510101105', '1006', '82');
10. INSERT INTO `course`.`choose` (`id`, `s_no`, `c_no`, `score`) VALUES ('9', '201513101116', '2001', '0');
11. INSERT INTO `course`.`choose` (`id`, `s_no`, `c_no`, `score`) VALUES ('10', '201513101111', '1004', '0');
12. SET FOREIGN_KEY_CHECKS = 1;

```

(疑问：此处 score 内容是 tinyint 类型 那么有符号的范围是-128~ 127 无符号的范围是 0 ~ 255 感觉最后两条插入的数值不能为 null 类型)

#### TINYINT 数据类型

TINYINT 数据类型用于存储需要 1 个存储字节的无符号整数。

##### 语法

[ UNSIGNED ] TINYINT

##### 注释

TINYINT 数据类型是一种精确数字数据类型：其精度在算术运算后不变。

可以将 TINYINT 显式指定为 UNSIGNED，但是该类型始终是无符号的，因此 UNSIGNED 修饰符无效。

TINYINT 的值的范围是 0 到  $2^8 - 1$ ，即 0 到 255。

在嵌入式 SQL 中，不应将 TINYINT 列值指定为 char 或无符号 char 的变量中，因为这样做的结果是系统会尝试将列的值转换为字符串，然后将第一个字节指派给程序中的变量。相反，应该将 TINYINT 列读入 2 个字节或 4 个字节的整数列。此外，要从使用 C 语言编写的应用程序向数据库发送 TINYINT 值，C 变量的类型应该为 unsigned char。

在将字符串转换为 TINYINT 时，前导空格和零会被忽略。如果前导字符是 '+'，会被忽略。如果前导字符是 '-'，则后面的位会被解释为负数。前导的 '0' 字符将跳过，其余字符将转换为整数。如果值超出目标数据类型的有效范围，字符串包含非法字符，或者字符串无法解码为整数，则会返回错误。

## 实验内容：

(1)：使用 insert 语句向 student 表中插入一条记录。记录内容为自己的个人信息。

```

1. insert into course.student (S_no,S_name,gender,d_no,phone,birthday,address) values('18408220','孙敬垚','男','001','15092524952','2001-01-29','山东')

```

使用 select \* from student; 语句查看结果。

The screenshot shows the MySQL Workbench interface. On the left, the 'Schemas' pane shows the 'course' database with tables 'choose', 'course', 'department', 'student', 'teacher', 'indexes', 'Foreign Keys', and 'Triggers'. The 'student' table is selected. The main window displays the 'student' table data in a grid. A red arrow points to the first row of data. Below the grid, the 'Output' pane shows the execution of the insert statement: 'insert into course.student (S\_no,S\_name,gender,d\_no,phone,birthday,address) values('18408220','孙敬垚','男','001','15092524952','2001-01-29','山东')'. The output shows that 1 row(s) were affected.

S_no	S_name	gender	d_no	phone	birthday	address
18408220	孙敬垚	男	001	15092524952	2001-01-29	山东
20151010101	李强东	男	001	11000000001	1999-05-10	昆明
20151010102	林慧	女	001	11000000002	1999-12-15	上海
20151101103	李达德	男	002	11000000003	1998-10-25	北京
20151101104	吴博文	女	002	11000000004	1998-08-10	昆明
20151101105	刘馨	男	003	11000000005	1999-05-08	北京
20151101106	赵立民	男	003	11000000006	1999-02-25	上海
201513101111	徐亮亮	男	004	11000000007	1998-09-05	上海
201513101116	王福萍	女	005	11000000008	1998-06-03	重庆

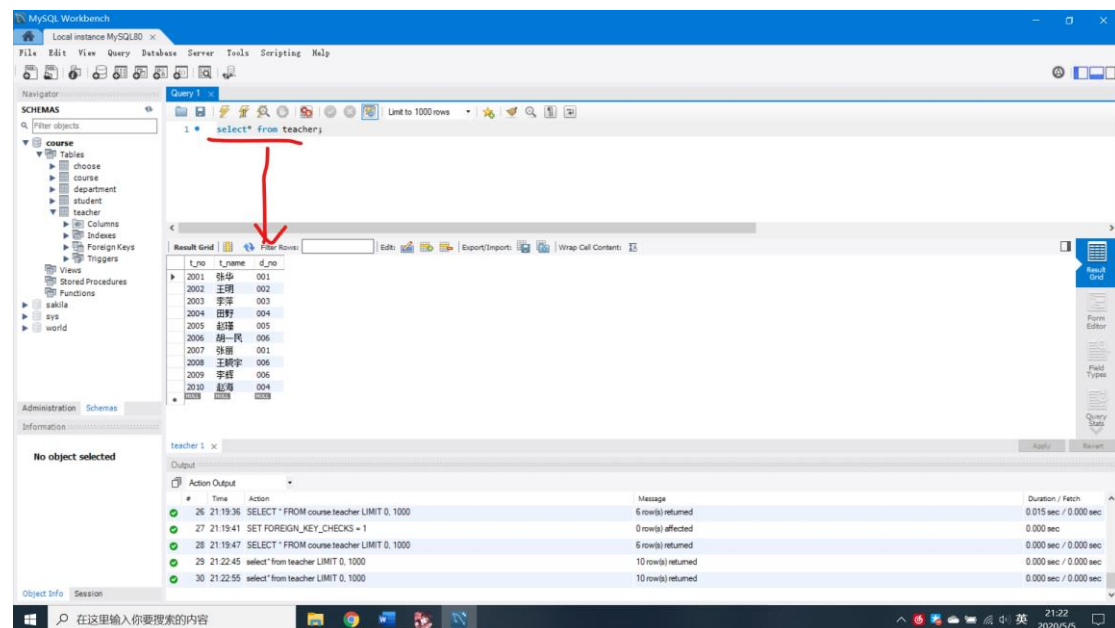
(2)：使用 insert 语句批量向 teacher 表中插入如下表中的记录。

```

1. SET FOREIGN_KEY_CHECKS = 0;
2. insert into course.teacher (t_no,t_name,d_no)values(2007,'张丽',001);
3. insert into course.teacher (t_no,t_name,d_no)values(2008,'王晓宇',006);
4. insert into course.teacher (t_no,t_name,d_no)values(2009,'李辉',006);
5. insert into course.teacher (t_no,t_name,d_no)values(2010,'赵海',004);
6. SET FOREIGN_KEY_CHECKS = 1;

```

使用 select \* from teacher;语句查看结果。



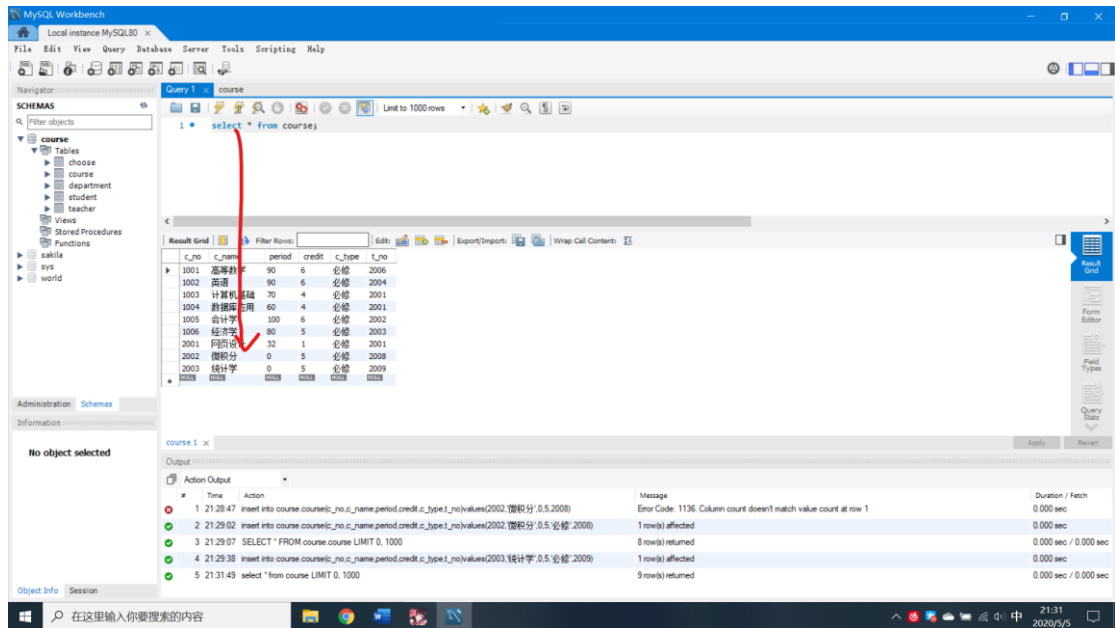
(3): 使用 insert 语句向 course 表中指定字段插入如下信息。

```

1. insert into course.course(c_no,c_name,period,credit,c_type,t_no)values(2002,
  '微积分',0,5,'必修',2008);
2. insert into course.course(c_no,c_name,period,credit,c_type,t_no)values(2003,
  '统计学',0,5,'必修',2009);

```

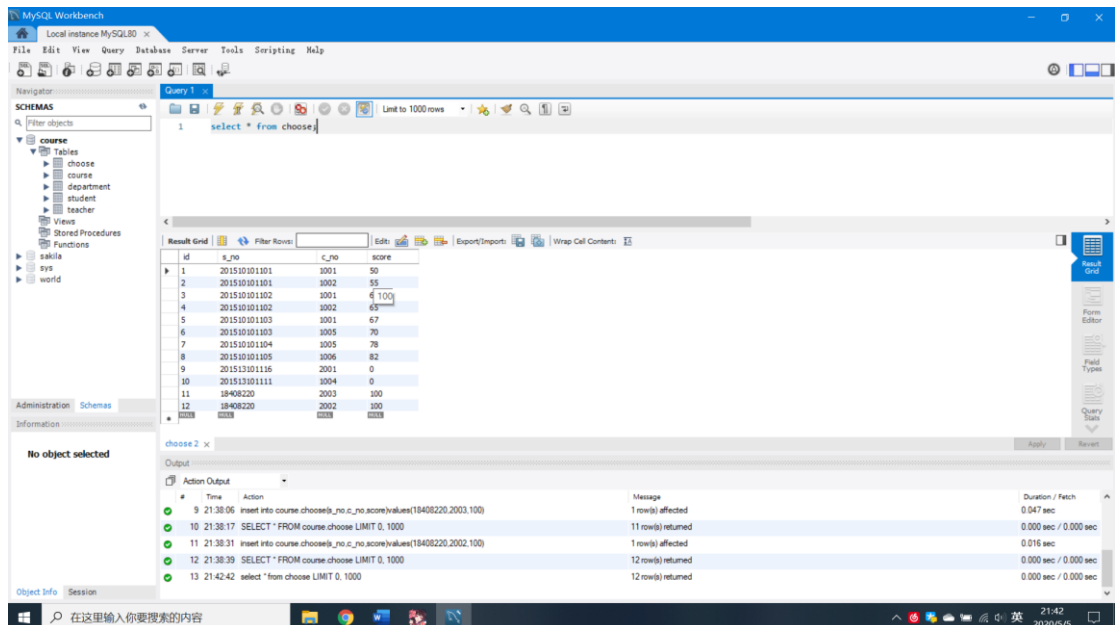
使用 select \* from course; 语句查看结果。



(4): 使用 insert 语句向 choose 表中插入两条选课记录。要求将自己所选的 微积分、统计学的信息插入该表，其中，id 字段自动生成下一个标号，score 字段自己模拟两个成绩。

1. `insert into course.choose(s_no,c_no,score)values(18408220,2002,100);`
2. `insert into course.choose(s_no,c_no,score)values(18408220,2003,100);`

使用 select \* from choose; 语句查看结果。



(5): 使用 update 语句，将 course 表中课程号为“2002”的教师编号更改为“2010”。

1. `UPDATE course.course SET t_no = '2010' WHERE (c_no = '1005');`

Result Grid

	c_no	c_name	period	credit	c_type	t_no
▶	1001	高等数学	90	6	必修	2006
	1002	英语	90	6	必修	2004
	1003	计算机基础	70	4	必修	2001
	1004	数据库应用	60	4	必修	2001
	1005	会计学	100	6	必修	2002
	1006	经济学	80	5	必修	2003
	2001	网页设计	32	1	必修	2001
	2002	微积分	0	5	必修	2008
	2003	统计学	0	5	必修	2009
*	NULL	NULL	NULL	NULL	NULL	NULL

使用 `select * from choose;` 语句查看结果。

1 `select * from course;`

Limit to 1000 rows

Result Grid

	c_no	c_name	period	credit	c_type	t_no
	1001	高等数学	90	6	必修	2006
	1002	英语	90	6	必修	2004
	1003	计算机基础	70	4	必修	2001
	1004	数据库应用	60	4	必修	2001
▶	1005	会计学	100	6	必修	2010
	1006	经济学	80	5	必修	2003
	2001	网页设计	32	1	必修	2001
	2002	微积分	0	5	必修	2008
	2003	统计学	0	5	必修	2009
*	NULL	NULL	NULL	NULL	NULL	NULL

(6): 删除教师表中教师编号为“2008”的教师。

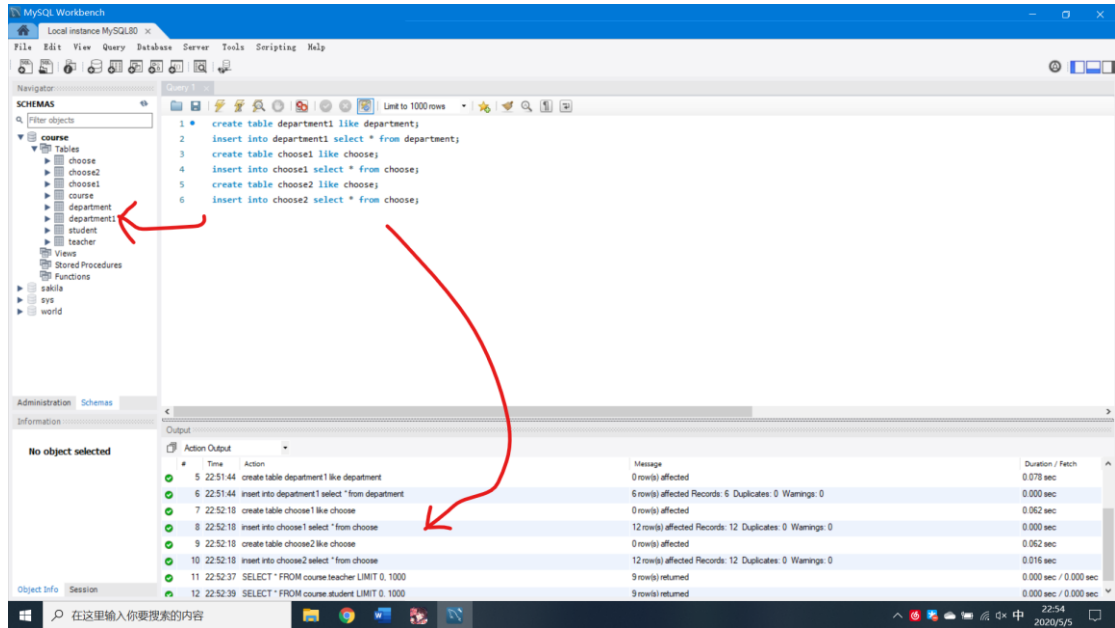
	t_no	t_name	d_no
▶	2001	张华	001
	2002	王明	002
	2003	李萍	003
	2004	田野	004
	2005	赵瑾	005
	2006	胡一民	006
	2007	张丽	001
	2008	王晓宇	006
	2009	李辉	006
	2010	赵海	004
•	NULL	NULL	NULL

1. `SET FOREIGN_KEY_CHECKS = 0;`
2. `delete from teacher where t_no = 2008;`
3. `SET FOREIGN_KEY_CHECKS = 1;`

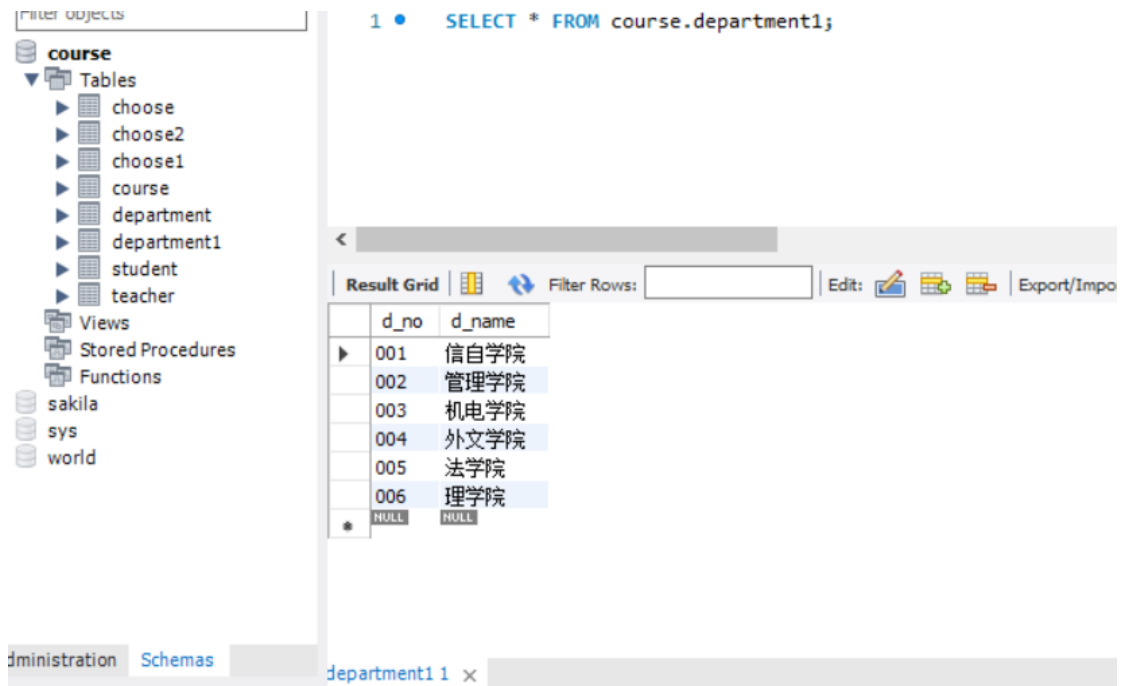
使用 `select * from teacher;` 语句查看结果。

	t_no	t_name	d_no
▶	2001	张华	001
	2002	王明	002
	2003	李萍	003
	2004	田野	004
	2005	赵瑾	005
	2006	胡一民	006
	2007	张丽	001
	2008	王晓宇	006
	2009	李辉	006
•	NULL	NULL	NULL

(7): 使用以下三条语句创建三张新的数据表。然后, 使用 `insert into` 语句将 `department` 记录以及 `choose` 记录插入到新建的数据表中。

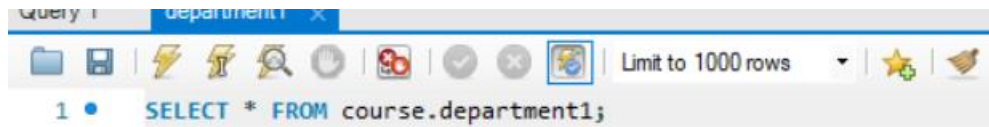


(8): 使用 replace into 命令向 department1 表中插入如下一条记录, 查看执行 后系统返回所影响行数, 并查看 department1 表中的内容。



1. replace into department1(d\_no,d\_name)values(004,'英语系');





Result Grid

Filter Rows:

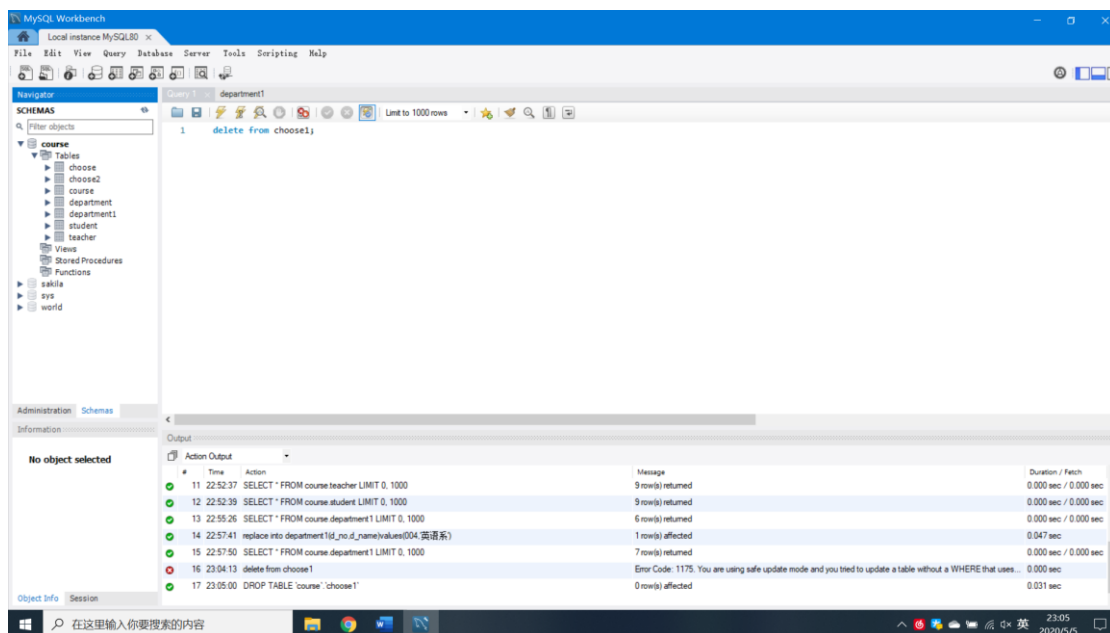
Edit: Export/Import:

	d_no	d_name
▶	001	信自学院
	002	管理学院
	003	机电学院
	004	外文学院
	005	法学院
	006	理学院
	4	英语系
✱	NULL	NULL

思考: replace into 会在插入时, 如果已经存在, 就会删除该记录 然后插入  
而改成 insert 会检测是否存在, 如果存在的话 将会插入失败

(9): 使用 delete 语句删除 choose1 表中的所有记录。

1. delete from choose1;



(10): 向 choose1 表中任意插入两条记录查看 id 字段的内容。

原 ID 字段:

1 • SELECT \* FROM course.choose1;

<

Result Grid

Filter Rows:

Edit: Export/Import: Wrap

	id	s_no	c_no	score
▶	1	201510101101	1001	50
	2	201510101101	1002	55
	3	201510101102	1001	60
	4	201510101102	1002	65
	5	201510101103	1001	67
	6	201510101103	1005	70
	7	201510101104	1005	78
	8	201510101105	1006	82
	9	201513101116	2001	0
	10	201513101111	1004	0
	11	18408220	2003	100
	12	18408220	2002	100
✱	NULL	NULL	NULL	NULL

choose1 1 ×

添加后:

Query 1 choose1 choose1 x

```
1 insert into choose1(s_no,c_no,score)values(18408203,2003,60);
2 • insert into choose1(s_no,c_no,score)values(18408223,2003,65);
```

Result Grid

	id	s_no	c_no	score
▶	1	201510101101	1001	50
	2	201510101101	1002	55
	3	201510101102	1001	60
	4	201510101102	1002	65
	5	201510101103	1001	67
	6	201510101103	1005	70
	7	201510101104	1005	78
	8	201510101105	1006	82
	9	201513101116	2001	0
	10	201513101111	1004	0
	11	18408220	2003	100
	12	18408220	2002	100
	13	18408203	2003	60
	14	18408223	2003	65

choose1 x

ID 字段是自增变化的。

(11): 使用 truncate 语句清空 choose1 表中的所有记录。使用 show create table choose1 语句查看表结构。任意插入一条语句查看 id 字段的内容。

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

course

Tables

- choose
- choose1
- choose2
- course
- department
- department1
- student
- teacher

Views

Stored Procedures

Functions

sakila

sys

world

Administration Schemas

Information

No object selected

Query 1

```
1 truncate table choose1;
2 • select *from choose1;
```

Result Grid

	id	s_no	c_no	score
--	----	------	------	-------

choose1 x

Output

Action Output

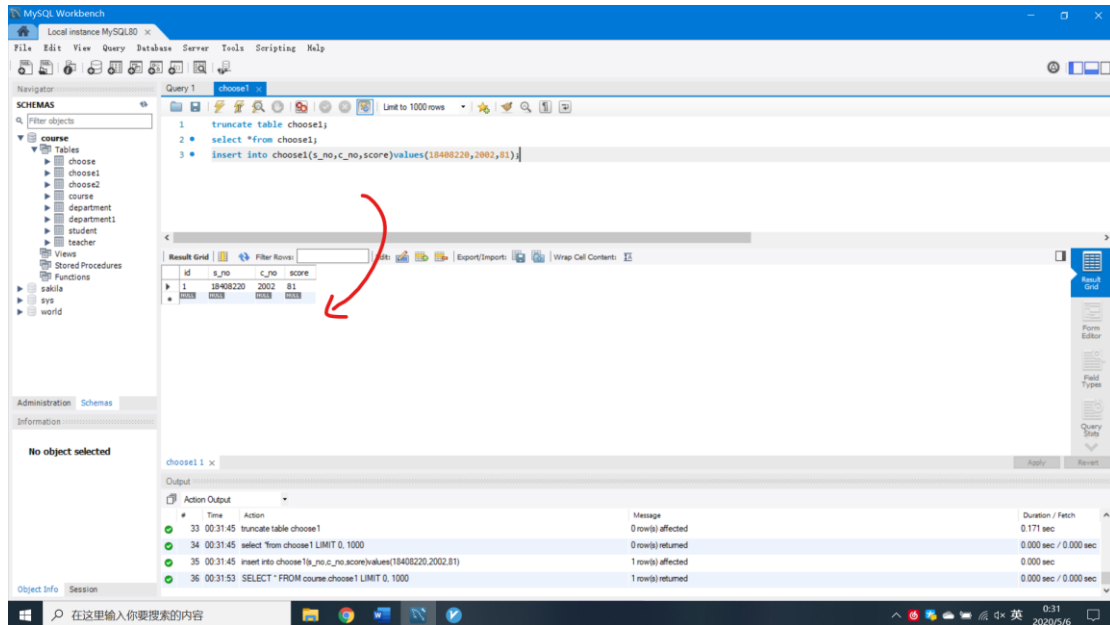
#	Time	Action	Message	Duration / Fetch
23	00:22:52	SELECT * FROM course choose1 LIMIT 0, 1000	14 row(s) returned	0.000 sec / 0.000 sec
24	00:28:34	truncate table choose1	0 row(s) affected	0.219 sec
25	00:29:52	truncate table choose1	0 row(s) affected	0.110 sec
26	00:29:53	select from choose1 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

Object Info Session

在这里输入你要搜索的内容

0:29 2020/5/6

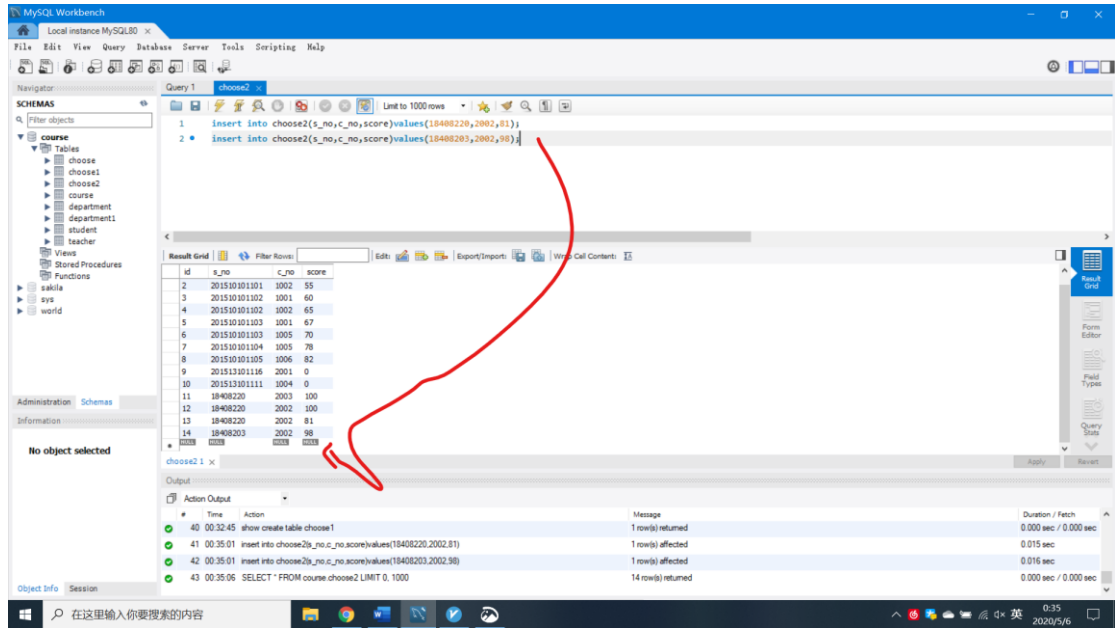
1. `truncate table choose1;`
2. `select *from choose1;`
3. `insert into choose1(s_no,c_no,score)values(18408220,2002,81);`
4. `show create table choose1;`



Column	Type	Default Value	Nullable	Character Set	Collation	Privileges	Extra	Comments
c_no	char(4)		NO	gbk	gbk_chinese_ci	select,insert,update,references		课程号
id	int		NO			select,insert,update,references	auto_increment	auto_increment
s_no	char(12)		NO	gbk	gbk_chinese_ci	select,insert,update,references		学号
score	tinyint		NO			select,insert,update,references		成绩

(12): 向 choose2 表中任意插入两条记录查看 id 字段的内容

1. `insert into choose2(s_no,c_no,score)values(18408220,2002,81);`
2. `insert into choose2(s_no,c_no,score)values(18408203,2002,98);`
3. `select *from choose2;`



修改后的 ID 是按照自增的顺序来排列的。  
从之前的 ID 值+1 转换成当前 ID。