2020/08/24_数据库_第3课_MySql语法_数据库相关的操作

笔记本： 数据库
创建时间： 2020/8/24 星期一 10:37
作者： ileemi

MySQL

下载安装 MySQL Database、MySQL Workbench。

> 命令行启动MySQL：**net start mysql**

MySQL 默认端口号为：3306

数据库中学号（数据）的类型，一般都给字符串：方便查询
VARCHAR -- 变长字符串存储
CHAR -- 定长字符串存储（浪费空间）

外键需要已存在的两张表，表创建完成，才能建立关系。

接口：
使用API，统计数据库数据

SQL语言( )

SQL语法

数据库操作

SQL语法 -- 解释型，不分大小写，-- 注释，数据库中默认字符集为UTF-8。

> 创建数据库：

```
CREATE DATABASE mysqlwork_student
```

> 删除数据库：

```
DROP DATABASE mysqlwork_student
```

> 修改数据库:

```
ALTER DATABASE mysqlwork_student CHARACTER SET = UTF8
```

> 使用数据库

```
use mysqlwork_student
```

数据库表的操作

> 创建数据库表:

```
-- 创建班级表
CREATE TABLE T_CLASS (
        CLASS_ID VARCHAR ( 255 ),
        CLASS_NAME VARCHAR ( 255 ),
PRIMARY KEY ( CLASS_ID ));

-- 创建学生表 -- 插入外键(班级ID) CONSTRAINT -- 约束
CREATE TABLE T_STUDENT (
        STU_ID VARCHAR ( 255 ),
        STU_NAME VARCHAR ( 255 ) NOT NULL,
        CLASS_ID VARCHAR ( 255 ) NOT NULL,
        PRIMARY KEY ( STU_ID ),
CONSTRAINT FK1 FOREIGN KEY ( CLASS_ID ) REFERENCES T_CLASS ( CLASS_ID
));

-- 创建课程表
CREATE TABLE T_COURSE (
        COURSE_ID VARCHAR ( 255 ),
        COURSE_NAME VARCHAR ( 255 ),
PRIMARY KEY ( COURSE_ID ));

-- 创建选课表 -- 符合组建PRIMARY KEY(STU_ID, COURSE_ID)
-- 添加外键的错误写法 STU_ID VARCHAR ( 255 ) REFERENCES
T_STUDENT(STU_ID),
CREATE TABLE T_SELECT (
        STU_ID VARCHAR ( 255 ),
        COURSE_ID VARCHAR ( 255 ),
```

```
        SCORE FLOAT NOT NULL,
        PRIMARY KEY ( STU_ID, COURSE_ID ),
        CONSTRAINT FK2 FOREIGN KEY ( STU_ID ) REFERENCES T_STUDENT ( STU_ID ),
CONSTRAINT FK3 FOREIGN KEY ( COURSE_ID ) REFERENCES T_COURSE ( COURSE_ID ));
```

REFERENCES -- 引用
CONSTRAINT -- 约束（添加外键）
有空格使用 '' 单引号
复合主键设置外键 -- 在最后一起设置

> 删除表

```
-- 删除表
DROP TABLE T_CLASS;
DROP TABLE T_STUDENT;
DROP TABLE T_COURSE;
DROP TABLE T_SELECT;
```

> 修改表

```
-- 修改表
ALTER TABLE T_CLASS;

-- 添加列
ALTER TABLE add age int;
```

表数据的操作

-- 表创建完成后,增加表中的数据(增加, 删除, 修改, 查询)
-- 增加(insert)
-- 删除(delete)
-- 修改(update)
-- 查询(select)

> 插入数据

```
INSERT INTO T_CLASS(CLASS_ID, CLASS_NAME) VALUES('001', '科锐1班');
INSERT INTO T_CLASS(CLASS_ID) VALUES('002');
INSERT INTO T_CLASS VALUES('003', '科锐2班');
```

> 修改（条件修改）

```
-- 没有指定要修改的项,表中的数据将全部修改
UPDATE T_CLASS SET CLASS_NAME = '科锐3班';

-- 条件修改
UPDATE T_CLASS SET CLASS_NAME = '科锐1班' WHERE CLASS_;
UPDATE T_CLASS SET CLASS_NAME = '科锐2班' WHERE CLASS_;
UPDATE T_CLASS SET CLASS_NAME = '科锐3班' WHERE CLASS_;
```

> 删除，条件删除

```
-- 删除数据
DELETE FROM T_CLASS;                -- 删除表中所有的数据
-- 条件删除 删除班级表中班级ID为 '003'的项
DELETE FROM T_CLASS WHERE CLASS_;                    -- 删除表中所有的数据
```

> **查询**:

-- 查询 T_CLASS 班级表中所有的列的数据
SELECT *FROM T_CLASS;

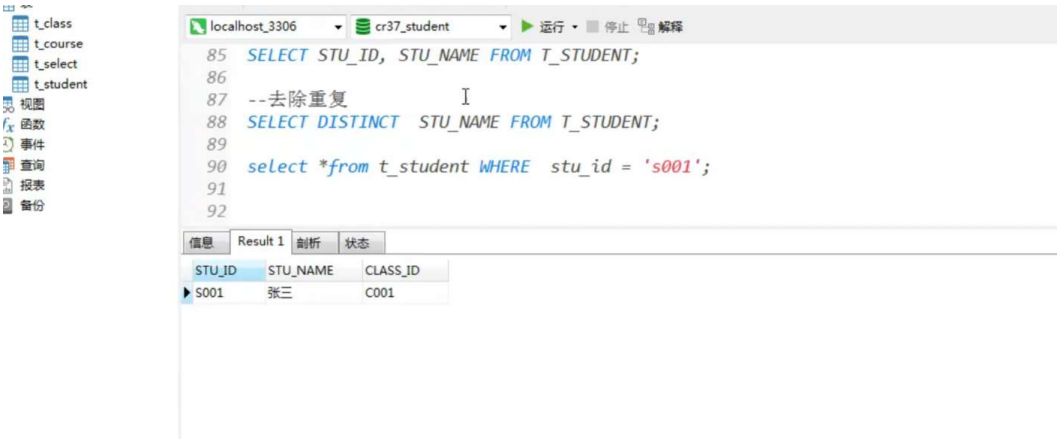
' * ' -- 代表全部

```
-- 查询student表中id列的数据
select id as '学号' from student;
```

去除重复数据 -- SELECT后 加 DISTINCT

[into_option] -- 将查询结果写入到文件中

```
-- 去除重复
SELECT DISTINCT FROM T_STUDENT;
SELECT * FROM T_STUDENT WHERE STU_;
```



WHERE 条件

```
select *from t_student WHERE  stu_id = 's001' or class_id = 'c001';

select *from t_student WHERE  NOT stu_id='s001'
select *from t_student WHERE  stu_id <> 's001'
select *from t_student WHERE  stu_id != 's001'
```

信息 | Result 1 | 剖析 | 状态

STU_ID	STU_NAME	CLASS_ID
S002	李四	C001
S003	王五	C002
S004	韩梅梅	C002
S005	万事达	C003
S006	万事达	C003

```
-- 存在
select *from t_student WHERE class_id in ('c002', 'c003');

-- 范围
select *from t_student WHERE class_id BETWEEN 'C001' AND 'C002'
```

信息 | Result 1 | 剖析 | 状态

STU_ID	STU_NAME	CLASS_ID
S001	张三	C001
S002	李四	C001
S003	王五	C002
S004	韩梅梅	C002

-- 范围查询

'xxx' AND 'xxx'

```
SELECT * FROM T_CLASS WHERE CLASS_ID BETWEEN 'CR001' and 'CR003';
SELECT * FROM T_CLASS WHERE CLASS_ID in('CR001', 'CR003');
SELECT * FROM T_CLASS WHERE CLASS_ID not in('CR001', 'CR003');
```

-- 模糊查询

LIKE -- 支持无线嵌套

_ -- 任意字符

```
-- 模糊查询
select *from t_student WHERE stu_id LIKE '___1'
```

信息 | Result 1 | 剖析 | 状态

STU_ID	STU_NAME	CLASS_ID
S001	张三	C001

```
-- 模糊查询
select *from t_student WHERE stu_id LIKE '___1'
select *from t_student WHERE stu_id LIKE 'S___'
select *from t_student WHERE stu_id LIKE 'S%'
select *from t_student WHERE stu_id LIKE '%4%'
select *from t_student WHERE stu_id LIKE '%4'
select *from t_student WHERE stu_name LIKE '张%'
```

信息 | Result 1 | 剖析 | 状态

STU_ID	STU_NAME	CLASS_ID
S001	张三	C001

-- 分组 聚合函数 分组查询指定的数据

GROUP BY

```
115  -- 分组 聚合函数
116  select stu_id from t_select GROUP BY stu_id
117  select stu_id, sum(score) from t_select GROUP BY stu_id -- 学生总分
118  select stu_id, avg(score) from t_select GROUP BY stu_id
119  select stu_id, max(score) from t_select GROUP BY stu_id
120  select stu_id, min(score) from t_select GROUP BY stu_id
121  select sum(score) from t_select -- 总分
122
123  --班级总分
124  select course_id, sum(score) from t_select GROUP BY course_id
125
126
```

信息 | Result 1 | 剖析 | 状态

course_id	sum(score)
L001	164.5
L002	180
L003	99.5

-- 显示前三个

LIMIT

ORDER BY -- 以指定的方式进行排序

```
126  --  显示前3个
127  select * from t_student limit 3;
128
129  --求最高成绩的学生
130  select stu_id, max(score) from  t_select;
131
132  -- 排序
133  select * from  t_select ORDER BY score ASC ;
134  select * from  t_select ORDER BY score desc ;
```

信息 | Result 1 | 剖析 | 状态

STU_ID	COURSE_ID	SCORE
S002	L002	100
S002	L003	99.5
S002	L001	99
S001	L002	80
S001	L001	60.5
S003	L001	5

排序后显示一个数据最高分：

```
132  -- 排序
133  select * from  t_select ORDER BY score ASC ;
134  select * from  t_select ORDER BY score desc ;
135  select * from  t_select ORDER BY score desc limit 1;
136
137
```

信息 | Result 1 | 剖析 | 状态

STU_ID	COURSE_ID	SCORE
S002	L002	100

-- 子查询

```
137  -- 子查询 显示科锐1班的学生
138  select * from t_student where CLASS_ID = (select class_id from t_class where
     CLASS_NAME = '科锐1班')
139
```

信息 | Result 1 | 剖析 | 状态

STU_ID	STU_NAME	CLASS_ID
S001	张三	C001
S002	李四	C001

```
140  select * from t_student where CLASS_ID in (select class_id from t_class where
     CLASS_NAME = '科锐1班' or CLASS_NAME = '科锐2班')
141
142
143
```

信息 | Result 1 | 剖析 | 状态

STU_ID	STU_NAME	CLASS_ID
S001	张三	C001
S002	李四	C001
S003	王五	C002
S004	韩梅梅	C002

-- 多表查询 -- 将多个表进行拼接（要以数据关系进行）

```
143  -- 多表查询
144  select * from t_student, t_class -- 笛卡尔面积
145
```

信息 | Result 1 | 剖析 | 状态

STU_ID	STU_NAME	CLASS_ID	CLASS_ID(1)	CLASS_NAME
S001	张三	C001	C001	科锐1班
S001	张三	C001	C002	科锐2班
S001	张三	C001	C003	科锐3班
S002	李四	C001	C001	科锐1班
S002	李四	C001	C002	科锐2班
S002	李四	C001	C003	科锐3班
S003	王五	C002	C001	科锐1班
S003	王五	C002	C002	科锐2班
S003	王五	C002	C003	科锐3班
S004	韩梅梅	C002	C001	科锐1班
S004	韩梅梅	C002	C002	科锐2班

```
143  -- 多表查询 笛卡尔面积
144  select stu_id, stu_name, class_name from t_student, t_class where t_student.
     class_id = t_class.class_id  wher
145
```

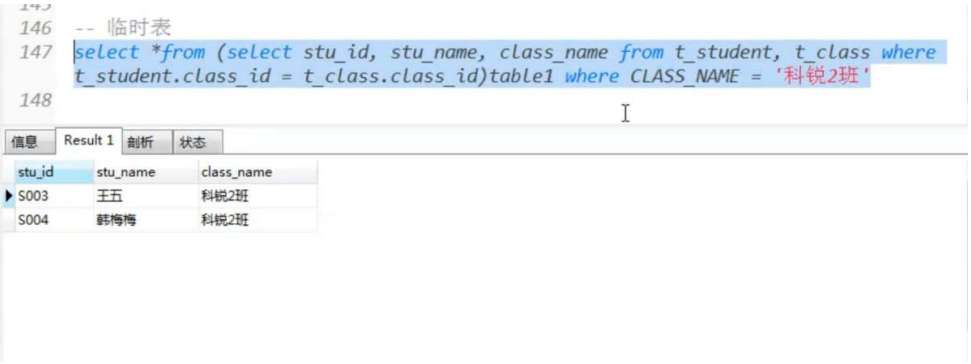
WHERE

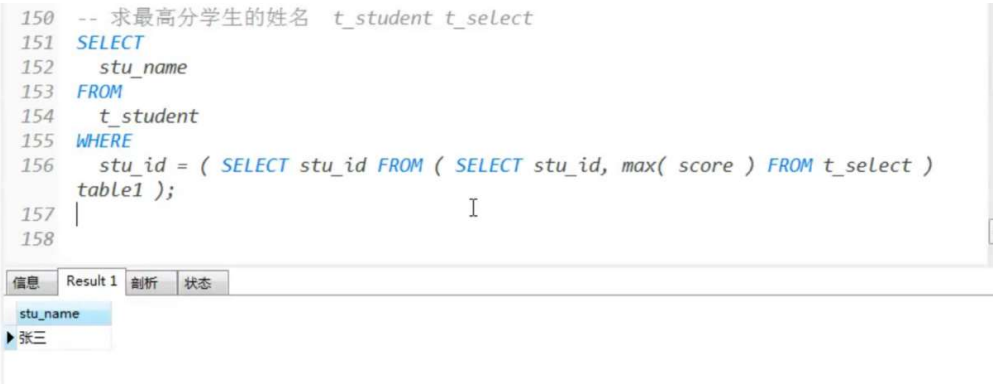
信息 | Result 1 | 剖析 | 状态

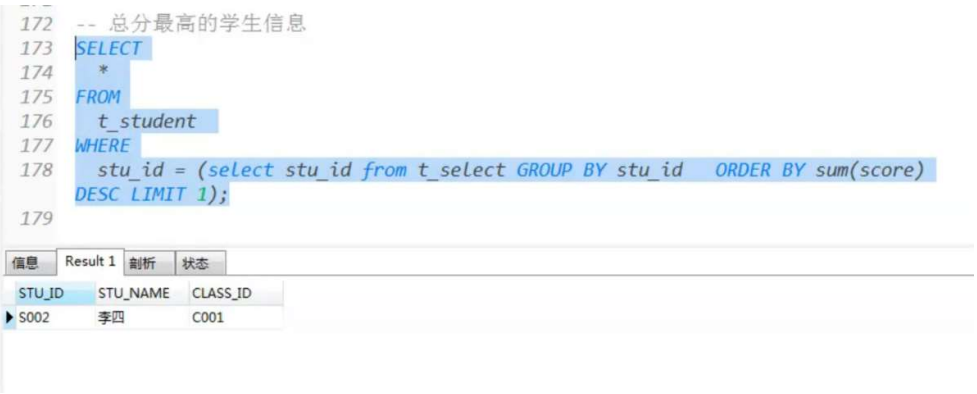
stu_id	stu_name	class_name
S001	张三	科锐1班
S002	李四	科锐1班
S003	王五	科锐2班
S004	韩梅梅	科锐2班
S005	万事达	科锐3班
S006	万事达	科锐3班

不可以将表组合后进行查询，组合后的表并不存在。

-- 临时表（在数据库中不存在） -- 将组合表当作临时表







代码示例：

```
-- 创建数据库 -- 增加 删除 修改 查询
CREATE DATABASE mysqlwork_student
DROP DATABASE mysqlwork_student
ALTER DATABASE mysqlwork_student CHARACTER SET = UTF8

-- 使用数据库
use mysqlwork_student

-- 创建班级表
CREATE TABLE T_CLASS (
  CLASS_ID VARCHAR ( 255 ),
  CLASS_NAME VARCHAR ( 255 ),
PRIMARY KEY ( CLASS_ID ));

-- 创建学生表 -- 插入外键(班级ID) CONSTRAINT -- 约束
CREATE TABLE T_STUDENT (
  STU_ID VARCHAR ( 255 ),
```

```
  STU_NAME VARCHAR ( 255 ) NOT NULL,
  CLASS_ID VARCHAR ( 255 ) NOT NULL,
  PRIMARY KEY ( STU_ID ),
CONSTRAINT FK1 FOREIGN KEY ( CLASS_ID ) REFERENCES T_CLASS ( CLASS_ID
));

-- 创建课程表
CREATE TABLE T_COURSE (
  COURSE_ID VARCHAR ( 255 ),
  COURSE_NAME VARCHAR ( 255 ),
PRIMARY KEY ( COURSE_ID ));


-- 创建选课表 -- 符合组建PRIMARY KEY(STU_ID, COURSE_ID)
-- 添加外键的错误写法 STU_ID VARCHAR ( 255 ) REFERENCES
T_STUDENT(STU_ID),
CREATE TABLE T_SELECT (
  STU_ID VARCHAR ( 255 ),
  COURSE_ID VARCHAR ( 255 ),
  SCORE FLOAT NOT NULL,
  PRIMARY KEY ( STU_ID, COURSE_ID ),
  CONSTRAINT FK2 FOREIGN KEY ( STU_ID ) REFERENCES T_STUDENT ( STU_ID ),
CONSTRAINT FK3 FOREIGN KEY ( COURSE_ID ) REFERENCES T_COURSE ( COURSE_ID
));

-- 删除表
DROP TABLE T_CLASS;
DROP TABLE T_STUDENT;
DROP TABLE T_COURSE;
DROP TABLE T_SELECT;

-- 修改表
ALTER TABLE T_CLASS;


-- 表创建完成后,增加表中的数据(增加, 删除, 修改, 查询)
-- 增加(insert)
-- 删除(delete)
-- 修改(update)
-- 查询(select)

-- 表中插入数据
INSERT INTO T_CLASS(CLASS_ID, CLASS_NAME) VALUES('CR001', '科锐1班');
-- INSERT INTO T_CLASS(CLASS_ID) VALUES('CR002');
INSERT INTO T_CLASS VALUES('CR002', '科锐2班');
INSERT INTO T_CLASS VALUES('CR003', '科锐3班');
```

```
INSERT INTO T_STUDENT VALUES('S01', '李四', 'CR001');
INSERT INTO T_STUDENT VALUES('S02', '王五', 'CR002');
INSERT INTO T_STUDENT VALUES('S03', '张三', 'CR002');
INSERT INTO T_STUDENT VALUES('S04', '小明', 'CR003');

INSERT INTO T_COURSE VALUES('CR001', 'C语言');
INSERT INTO T_COURSE VALUES('CR002', 'C++');
INSERT INTO T_COURSE VALUES('CR003', '数据结构');


INSERT INTO T_SELECT VALUES('S01', 'CR001', 60);
INSERT INTO T_SELECT VALUES('S02', 'CR001', 88);
INSERT INTO T_SELECT VALUES('S02', 'CR002', 99);
INSERT INTO T_SELECT VALUES('S02', 'CR003', 97);
INSERT INTO T_SELECT VALUES('S03', 'CR001', 100);
INSERT INTO T_SELECT VALUES('S03', 'CR002', 56);


-- 修改表中的字段

-- 没有指定要修改的项,表中的数据将全部修改
UPDATE T_CLASS SET CLASS_NAME = '科锐3班';

-- 条件修改
UPDATE T_CLASS SET CLASS_NAME = '科锐1班' WHERE CLASS_;
UPDATE T_CLASS SET CLASS_NAME = '科锐2班' WHERE CLASS_;
UPDATE T_CLASS SET CLASS_NAME = '科锐3班' WHERE CLASS_;

-- 删除数据
DELETE FROM T_CLASS;    -- 删除表中所有的数据
-- 条件删除 删除班级表中班级ID为 '003'的项
DELETE FROM T_CLASS WHERE CLASS_;    -- 删除表中所有的数据


-- 查询班级所有数据
SELECT *FROM T_CLASS;

 -- 查询
SELECT DISTINCT FROM T_STUDENT;
SELECT * FROM T_STUDENT WHERE STU_;

SELECT CLASS_ID FROM T_CLASS;
SELECT CLASS_ID AS '班级编号', CLASS_NAME AS '班级名称' FROM T_CLASS;

-- 条件查询
SELECT * from T_CLASS WHERE CLASS_NAME is null;
SELECT * from T_CLASS WHERE CLASS_NAME is not null;
```

```
-- = >= > <= < <> !=
-- or and not
SELECT * FROM T_CLASS WHERE CLASS_ID <> 'CR001';
SELECT * FROM T_CLASS WHERE CLASS_and CLASS_NAME = '科锐1班';
SELECT * FROM T_CLASS WHERE CLASS_or CLASS_NAME = '科锐2班';
SELECT * FROM T_CLASS WHERE 1 = 1 and 1 = 1;


-- 范围
SELECT * FROM T_CLASS WHERE CLASS_ID BETWEEN 'CR001' and 'CR003';
SELECT * FROM T_CLASS WHERE CLASS_ID in('CR001', 'CR003');
SELECT * FROM T_CLASS WHERE CLASS_ID not in('CR001', 'CR003');

-- 排序
SELECT * FROM T_SELECT order by SCORE asc;  -- 升序
SELECT * FROM T_SELECT order by SCORE desc;  -- 降序
SELECT * FROM T_SELECT order by SCORE desc LIMIT 1;    -- 显示一行

-- 去除重复
select DISTINCT CLASS_NAME FROM T_CLASS;


-- 子查询
-- 显示科锐1班的所有同学
SELECT
  *
FROM
  T_STUDENT
WHERE
  CLASS_ID IN ( SELECT CLASS_ID FROM T_CLASS WHERE CLASS_NAME = '科锐1
班' );

-- 显示科锐1班或者科锐2班的所有同学
SELECT
  *
FROM
  T_STUDENT
WHERE
  CLASS_ID IN ( SELECT CLASS_ID FROM T_CLASS WHERE CLASS_NAME = '科锐1
班' OR CLASS_NAME = '科锐2班' );

-- 张三班级名称是什么
SELECT
  CLASS_NAME
FROM
  T_CLASS
WHERE
```

```
  CLASS_ID = ( SELECT CLASS_ID FROM T_STUDENT WHERE STU_NAME = '张三' );

SELECT * FROM T_COURSE;
SELECT * FROM T_SELECT;

-- 聚合函数
-- 计算总分
SELECT sum(score) FROM T_SELECT;
SELECT count(score) FROM T_SELECT;

-- 计算班级总分以及班级平均分
select STU_ID, sum(score) FROM T_SELECT group by STU_ID;
select COURSE_ID, avg(score) FROM T_SELECT group by COURSE_ID;


-- 多表组合(笛卡尔面积) 意义不大
SELECT
  T_SELECT.STU_ID,
  T_STUDENT.STU_NAME,
  T_SELECT.COURSE_ID,
  T_COURSE.COURSE_NAME,
  score
FROM
  T_SELECT,
  T_STUDENT,
  T_COURSE
WHERE
  T_STUDENT.STU_ID = T_STUDENT.STU_ID
  AND T_COURSE.COURSE_ID = T_SELECT.COURSE_ID;


-- 临时表
SELECT
  *
FROM
  (
  SELECT
    T_SELECT.STU_ID,
    T_STUDENT.STU_NAME,
    T_SELECT.COURSE_ID,
    T_COURSE.COURSE_NAME,
    score
  FROM
    T_SELECT,
    T_STUDENT,
    T_COURSE
  WHERE
```

```
    T_STUDENT.STU_ID = T_SELECT.STU_ID
    AND T_COURSE.COURSE_ID = T_SELECT.COURSE_ID
  ) table1
WHERE
  STU_NAME = '李四';

-- 临时字段
SELECT
  *,
  ( SELECT STU_NAME FROM T_STUDENT WHERE T_STUDENT.STU_ID =
T_SELECT.STU_ID ) AS 'student_name'
FROM
  T_SELECT;

-- 谁的总分大于60 HAVING
SELECT
  STU_ID,
  sum( score )
FROM
  T_SELECT
GROUP BY
  STU_ID
HAVING
  sum( score ) > 60;
SELECT
  STU_ID,
  sum( score )
FROM
  T_SELECT
GROUP BY
  STU_ID;

-- GROUP BY 分组
-- exists == in    > 3 -- 谁选了哪些课程
SELECT
  *
FROM
  T_STUDENT
WHERE
  STU_ID IN ( SELECT STU_ID FROM T_SELECT );
SELECT
  *
FROM
  T_STUDENT
WHERE
  EXISTS ( SELECT STU_ID FROM T_SELECT WHERE T_SELECT.STU_ID =
T_STUDENT.STU_ID );
```