数据完整性

1. 主键约束(**primary key**)
    - 确保字段值不重复不为NULL
2. 唯一约束(**unique**)
    - 确保字段值不重复
3. 外键约束(**foreign key**)
    - 确保字段值必须来自于指定表
4. 检查约束(**check**)
    - 确保字段值的取值范围
5. 缺省约束(**default**)
    - 给相应字段提供默认值

SQL语法

数据库操作

- 创建数据库

```
-- 创建students数据库
create database students;
```

- 删除数据库

```
-- 删除students数据库
drop database students;
```

表操作

- 创建数据库表

```
-- 创建class表
create table class (id integer primary key not null autoincrement, name
varchar(50) not null);

-- 创建student表
create table student (id integer primary key not null autoincrement, name
varchar(50) not null, class_id integer, constraint fk_1 foreign
key(class_id) refences class(id));
```

- 删除数据库表

```
-- 删除student表
drop table student;
```

- 修改数据库表

```
-- 添加列
alter student add age int;
```

数据操作

- 插入数据

```
-- 插入数据
insert into student (id, name) values (1, 'aaa');
insert into student (id, name) values (2, 'bbb'),(3, 'ccc'),(4, 'ddd');
```

- 删除数据

```
-- 删除student中id为2的项
delete from student where id = 2;
-- 删除student中name为'ccc'的项
delete from student where name = 'ccc';
```

- 查询数据

```
-- 查询student表中所有的列的数据
select * from student;

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

- 修改数据

```
-- 修改学号为1的学生的姓名为'abab'
update student set name = 'abab' where id = 1;
```

例子

```
create database student;

-- 创建班级表
create table t_class (
class_id varchar(45),
class_name varchar(45),
primary key (class_id)
```

```
);

-- 创建学生表
create table t_student (
student_id varchar(45),
student_name varchar(45),
class_id varchar(45),
primary key (student_id),
foreign key (class_id) references t_class(class_id)
);

-- 课程表
create table t_course (
course_id varchar(45),
course_name varchar(45),
primary key (course_id)
);


-- 选课表
create table t_select(
student_id varchar(45),
course_id varchar(45),
score  float,
primary key (student_id, course_id),
foreign key (student_id) references t_student(student_id),
foreign key (course_id) references t_course(course_id)
);

show tables;


-- 插入数据
insert into  t_class (class_id, class_name) values('cr01', '科锐1班');
insert into  t_class (class_id, class_name) values('cr001', '科锐001班');
insert into  t_class  values('cr02', '科锐2班');
insert into  t_class  (class_id) values('cr03');
insert into  t_class (class_id, class_name) values('cr04', '科锐1班');

insert into  t_student values('s01', '张三', 'cr01');
insert into  t_student values('s02', '李四', 'cr01');
insert into  t_student values('s03', '王五', 'cr02');
insert into  t_student values('s04', '韩梅梅', 'cr04');

insert into  t_course values('c01', 'c语言');
insert into  t_course values('c02', 'c++');
insert into  t_course values('c03', '数据结构');

insert into  t_select values('s01', 'c01', 40);
insert into  t_select values('s02', 'c01', 90);
insert into  t_select values('s02', 'c02', 95);
insert into  t_select values('s02', 'c03', 100);
insert into  t_select values('s03', 'c01', 70);
insert into  t_select values('s03', 'c02', 80);

-- 删除表数据
delete from t_class;
```

```sql
-- 条件删除
delete from t_class where class_id = 'cr03';

-- 修改
update t_class set class_name = '科锐3版';
update t_class set class_name = '科锐3班' where class_id = 'cr03';


-- 查询
select 1;
select * from t_class;
SELECT class_id from t_class;
SELECT class_id as '班级编号', class_name as '班级名称' from t_class;

SELECT class_id, class_name = 'a'  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 <> 'cr01';
select * from t_class where class_id = 'cr01' and class_name = '科锐1班';
select * from t_class where class_id = 'cr01' 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 'cr01' and 'cr03';
select * from t_class where class_id in('cr01', 'cr03');
select * from t_class where class_id not in('cr01', 'cr03');

-- 排序
select * from t_class order by class_id asc;
select * from t_class order by class_id desc;
select *  from t_class order by class_id desc LIMIT 2;

-- 去除重复
select DISTINCT class_name from t_class;



-- 子查询  =   in  exists

-- 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'
);

-- 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 student_name = '张三'
) ;

select *from t_course;
select *from t_select;
```

```sql
-- 张三的总分多少?
-- c语言的总分多少?
-- 聚合函数
select sum(score) from t_select;
select count(score) from t_select;

select student_id, sum(score) from t_select group by student_id;
select course_id, avg(score) from t_select group by course_id;

-- 多表组合 笛卡尔面积
select t_select.student_id,
t_student.student_name,
t_select.course_id,
t_course.course_name,
score
 from t_select, t_student, t_course
where t_student.student_id = t_select.student_id and
t_course.course_id = t_select.course_id;


-- 临时表
select *from (
select t_select.student_id,
t_student.student_name,
t_select.course_id,
t_course.course_name,
score
 from t_select, t_student, t_course
where t_student.student_id = t_select.student_id and
t_course.course_id = t_select.course_id
)table1 where student_name = '张三';

-- 临时字段
select *,
(select student_name from t_student where t_student.student_id =
t_select.student_id)  as 'student_name'
from t_select;

-- 谁的总分大于60 HAVING
select student_id, sum(score) from t_select GROUP BY student_id HAVING
sum(score)  > 60;

select student_id, sum(score) from t_select GROUP BY student_id;

-- GROUP BY 分组

-- exists == in   > 3
-- 谁选了课程
select *from t_student where student_id in (select student_id from t_select);


select *from t_student where exists (
select student_id from t_select where t_select.student_id =
t_student.student_id);
```