CREATE DATABASE xk\_system;

USE xk\_system;

CREATE TABLE department (

dept\_id CHAR(2) PRIMARY KEY, -- 院系号

dept\_name VARCHAR(100) NOT NULL -- 院系名称

);

-- 插入一些初始数据

INSERT INTO department (dept\_id, dept\_name) VALUES

('01', '计算机科学与技术'),

('02', '电子工程'),

('03', '机械工程');

CREATE TABLE course (

course\_id CHAR(8) PRIMARY KEY, -- 课程号

course\_name VARCHAR(100) NOT NULL, -- 课程名称

semester VARCHAR(20) NOT NULL, -- 学期

credit INT NOT NULL -- 学分

);

-- 插入一些初始数据

INSERT INTO course (course\_id, course\_name, semester, credit) VALUES

('08301001', '离散数学', '2023秋季', 4),

('08305001', '数据库原理', '2023秋季', 4),

('08305002', '数据结构', '2023秋季', 4),

('08306001', '机械设计基础', '2023春季', 3);

CREATE TABLE student (

student\_id CHAR(10) PRIMARY KEY, -- 学号

name VARCHAR(100) NOT NULL, -- 姓名

gender ENUM('男', '女') NOT NULL, -- 性别

age INT NOT NULL, -- 年龄

dept\_id CHAR(2), -- 院系号

FOREIGN KEY (dept\_id) REFERENCES department(dept\_id) -- 外键，引用院系表

);

-- 插入一些初始数据

INSERT INTO student (student\_id, name, gender, age, dept\_id) VALUES

('S202101001', '李明', '男', 21, '01'),

('S202102002', '王强', '男', 22, '02'),

('S202301003', '张丽', '女', 20, '01'),

('S202103004', '赵子杰', '男', 23, '03'),

('S202202005', '孙怡婷', '女', 21, '02'),

('S202003006', '陈宝华', '男', 22, '03'),

('S202001007', '刘洋', '男', 22, '01'),

('S202101008', '杨红', '女', 21, '01'),

('S202002009', '王晓明', '男', 24, '02'),

('S202203010', '李雪', '女', 20, '03');

CREATE TABLE teacher (

teacher\_id CHAR(10) PRIMARY KEY, -- 工号

name VARCHAR(100) NOT NULL, -- 姓名

dept\_id CHAR(2), -- 院系号

FOREIGN KEY (dept\_id) REFERENCES department(dept\_id) -- 外键，引用院系表

);

-- 插入一些初始数据

INSERT INTO teacher (teacher\_id, name, dept\_id) VALUES

('T201501001', '王晓萍', '01'),

('T201502002', '李超盛', '02'),

('T201503003', '张国栋', '03'),

('T201501004', '陈燕', '01'),

('T201502005', '赵佳婷', '02'),

('T201503006', '孙勇', '03'),

('T201501007', '杨丽华', '01'),

('T201502008', '刘开成', '02'),

('T201503009', '李平', '03'),

('T201501010', '黄士高', '01');

CREATE TABLE admin (

admin\_id CHAR(10) PRIMARY KEY, -- 管理员号

name VARCHAR(100) NOT NULL -- 姓名

);

-- 插入一些初始数据

INSERT INTO admin (admin\_id, name) VALUES

('A201001001', '张京'),

('A201101002', '李华'),

('A201201003', '王丁宁');

CREATE TABLE course\_selection (

student\_id CHAR(10) NOT NULL, -- 学生学号

course\_id CHAR(8) NOT NULL, -- 课程号

grade INT, -- 成绩

FOREIGN KEY (student\_id) REFERENCES student(student\_id), -- 外键，关联学生表

FOREIGN KEY (course\_id) REFERENCES course(course\_id) -- 外键，关联课程表

);

-- 插入一些初始数据

INSERT INTO course\_selection (student\_id, course\_id, grade) VALUES

('S202101001', '08301001', 85),

('S202003006', '08305001', 90),

('S202101008', '08305002', 88);

CREATE TABLE class (

teacher\_id CHAR(10) NOT NULL, -- 教师工号

course\_id CHAR(8) NOT NULL , -- 课程号

class\_time VARCHAR(20) NOT NULL,

PRIMARY KEY (teacher\_id, course\_id) ,

FOREIGN KEY (teacher\_id) REFERENCES teacher(teacher\_id) -- 外键，关联教师表

);

-- 插入一些初始数据

INSERT INTO class (teacher\_id, course\_id, class\_time) VALUES

('T201501001', '08301001', '星期一1-4节'), -- 王晓萍教授开设离散数学，星期一1-4节

('T201502002', '08305001', '星期二1-4节'), -- 李超盛教师开设数据库原理，星期二1-4节

('T201503003', '08306001', '星期三1-4节'), -- 张国栋教师开设机械设计基础 ，星期三1-4节

('T201501004', '08301001', '星期四1-4节'), -- 陈燕教授开设离散数学，星期四1-4节

('T201502005', '08305001', '星期五1-4节'), -- 赵佳婷教师开设数据库原理，星期五1-4节

('T201503006', '08305002', '星期一5-8节'); -- 孙勇教师开设数据结构，星期一5-8节

CREATE TABLE user (

id CHAR(10) PRIMARY KEY, -- 用户 ID

username VARCHAR(100) NOT NULL, -- 用户名

password VARCHAR(100) NOT NULL -- 密码

);

-- 插入一些初始数据（对应学生、教师、管理员）

-- 学生数据

INSERT INTO user (id, username, password) VALUES

('S202101001', '李明', '123456'),

('S202102002', '王强', '123456'),

('S202301003', '张丽', '123456'),

('S202103004', '赵子杰', '123456'),

('S202202005', '孙怡婷', '123456'),

('S202003006', '陈宝华', '123456'),

('S202001007', '刘洋', '123456'),

('S202101008', '杨红', '123456'),

('S202002009', '王晓明', '123456'),

('S202203010', '李雪', '123456');

-- 教师数据

INSERT INTO user (id, username, password) VALUES

('T201501001', '王晓萍', '123456'),

('T201502002', '李超盛', '123456'),

('T201503003', '张国栋', '123456'),

('T201501004', '陈燕', '123456'),

('T201502005', '赵佳婷', '123456'),

('T201503006', '孙勇', '123456'),

('T201501007', '杨丽华', '123456'),

('T201502008', '刘开成', '123456'),

('T201503009', '李平', '123456'),

('T201501010', '黄士高', '123456');

-- 管理员数据

INSERT INTO user (id, username, password) VALUES

('A201001001', '张京', '123456'),

('A201101002', '李华', '123456'),

('A201201003', '王丁宁', '123456');

DELIMITER $$

CREATE TRIGGER check\_user\_id\_before\_insert

BEFORE INSERT ON user

FOR EACH ROW

BEGIN

-- 验证新插入的 ID 是否在 student、teacher 或 admin 表中存在

IF NOT EXISTS (SELECT 1 FROM student WHERE student\_id = NEW.id) AND

NOT EXISTS (SELECT 1 FROM teacher WHERE teacher\_id = NEW.id) AND

NOT EXISTS (SELECT 1 FROM admin WHERE admin\_id = NEW.id) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'ID must exist in student, teacher, or admin table';

END IF;

END$$

CREATE TRIGGER check\_user\_id\_before\_update

BEFORE UPDATE ON user

FOR EACH ROW

BEGIN

-- 验证更新后的 ID 是否在 student、teacher 或 admin 表中存在

IF NOT EXISTS (SELECT 1 FROM student WHERE student\_id = NEW.id) AND

NOT EXISTS (SELECT 1 FROM teacher WHERE teacher\_id = NEW.id) AND

NOT EXISTS (SELECT 1 FROM admin WHERE admin\_id = NEW.id) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'ID must exist in student, teacher, or admin table';

END IF;

END$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER prevent\_student\_delete

BEFORE DELETE ON student

FOR EACH ROW

BEGIN

IF EXISTS (SELECT 1 FROM user WHERE id = OLD.student\_id) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Cannot delete student; referenced in user table';

END IF;

END$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER prevent\_student\_update

BEFORE UPDATE ON student

FOR EACH ROW

BEGIN

IF EXISTS (SELECT 1 FROM user WHERE id = OLD.student\_id) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Cannot update student\_id; referenced in user table';

END IF;

END$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER prevent\_teacher\_delete

BEFORE DELETE ON teacher

FOR EACH ROW

BEGIN

IF EXISTS (SELECT 1 FROM user WHERE id = OLD.teacher\_id) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Cannot delete teacher; referenced in user table';

END IF;

END$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER prevent\_teacher\_update

BEFORE UPDATE ON teacher

FOR EACH ROW

BEGIN

IF EXISTS (SELECT 1 FROM user WHERE id = OLD.teacher\_id) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Cannot update teacher\_id; referenced in user table';

END IF;

END$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER prevent\_admin\_delete

BEFORE DELETE ON admin

FOR EACH ROW

BEGIN

IF EXISTS (SELECT 1 FROM user WHERE id = OLD.admin\_id) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Cannot delete admin; referenced in user table';

END IF;

END$$

DELIMITER ;

DELIMITER $$

CREATE TRIGGER prevent\_admin\_update

BEFORE UPDATE ON admin

FOR EACH ROW

BEGIN

IF EXISTS (SELECT 1 FROM user WHERE id = OLD.admin\_id) THEN

SIGNAL SQLSTATE '45000'

SET MESSAGE\_TEXT = 'Cannot update admin\_id; referenced in user table';

END IF;

END$$

DELIMITER ;

ALTER TABLE course\_selection

ADD COLUMN teacher\_id CHAR(10) NOT NULL;

ALTER TABLE course\_selection

DROP FOREIGN KEY course\_selection\_ibfk\_2;

ALTER TABLE class

ADD CONSTRAINT unique\_course\_teacher UNIQUE (course\_id, teacher\_id);

UPDATE course\_selection

SET teacher\_id = 'T201501001'

WHERE student\_id = 'S202101001' AND course\_id = '08301001';

UPDATE course\_selection

SET teacher\_id = 'T201502002'

WHERE student\_id = 'S202003006' AND course\_id = '08305001';

UPDATE course\_selection

SET teacher\_id = 'T201503006'

WHERE student\_id = 'S202101008' AND course\_id = '08305002';

ALTER TABLE course\_selection

ADD CONSTRAINT FOREIGN KEY (course\_id, teacher\_id) REFERENCES class(course\_id, teacher\_id);