SET NAMES utf8mb4;

SET FOREIGN\_KEY\_CHECKS = 0;

CREATE DATABASE IF NOT EXISTS ROOM;

USE ROOM;

-- ----------------------------

-- room\_info表的结构

-- ----------------------------

DROP TABLE IF EXISTS `room\_info`;

CREATE TABLE `room\_info` (

`room\_id` int(0) NOT NULL,

`type\_id` int(0) NOT NULL,

`room\_number` varchar(20) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

`status` varchar(10) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL DEFAULT 'Vacant',

PRIMARY KEY (`room\_id`) USING BTREE,

UNIQUE INDEX `room\_number`(`room\_number`) USING BTREE,

INDEX `type\_id`(`type\_id`) USING BTREE,

INDEX `idx\_room\_info\_room\_number`(`room\_number`) USING BTREE,

CONSTRAINT `room\_info\_ibfk\_1` FOREIGN KEY (`type\_id`) REFERENCES `room\_type` (`type\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT

) ENGINE = InnoDB CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = Dynamic;

-- ----------------------------

-- room\_info表的记录

-- ----------------------------

INSERT INTO `room\_info` VALUES (1, 1, '101', 'Vacant');

INSERT INTO `room\_info` VALUES (2, 2, '102', 'Vacant');

INSERT INTO `room\_info` VALUES (3, 3, '103', 'Vacant');

INSERT INTO `room\_info` VALUES (4, 4, '104', 'Vacant');

INSERT INTO `room\_info` VALUES (5, 5, '105', 'Vacant');

INSERT INTO `room\_info` VALUES (6, 6, '201', 'Vacant');

INSERT INTO `room\_info` VALUES (7, 7, '202', 'Vacant');

INSERT INTO `room\_info` VALUES (8, 8, '203', 'Vacant');

-- ----------------------------

-- customer\_info表的结构

-- ----------------------------

DROP TABLE IF EXISTS `customer\_info`;

CREATE TABLE `customer\_info` (

`customer\_id` int(0) NOT NULL,

`room\_id` int(0) NOT NULL,

`check\_in\_date` date NOT NULL,

`check\_out\_date` date NULL DEFAULT NULL,

`total\_cost` decimal(10, 2) NULL DEFAULT NULL,

`username` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

PRIMARY KEY (`customer\_id`) USING BTREE,

INDEX `room\_id`(`room\_id`) USING BTREE,

CONSTRAINT `customer\_info\_ibfk\_1` FOREIGN KEY (`room\_id`) REFERENCES `room\_info` (`room\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT

) ENGINE = InnoDB CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = Dynamic;

-- ----------------------------

-- customer\_info表的记录

-- ----------------------------

INSERT INTO `customer\_info` VALUES (1, 1, '2023-06-21', '2023-06-23', 200.00, '张三');

INSERT INTO `customer\_info` VALUES (2, 2, '2023-06-21', '2023-06-25', 300.00, '王五');

INSERT INTO `customer\_info` VALUES (3, 3, '2023-06-22', '2023-06-26', 400.00, '刘麻子');

INSERT INTO `customer\_info` VALUES (4, 4, '2023-06-23', '2023-06-30', 900.00, '小猫');

INSERT INTO `customer\_info` VALUES (5, 5, '2023-06-24', '2023-06-29', 750.00, '小月月');

INSERT INTO `customer\_info` VALUES (6, 6, '2023-06-25', '2023-06-28', 540.00, '洋洋');

INSERT INTO `customer\_info` VALUES (7, 7, '2023-06-26', '2023-06-27', 90.00, '沸羊羊');

INSERT INTO `customer\_info` VALUES (8, 8, '2023-06-27', '2023-06-29', 240.00, '舔狗');

-- ----------------------------

-- room\_price表的结构

-- ----------------------------

DROP TABLE IF EXISTS `room\_price`;

CREATE TABLE `room\_price` (

`price\_id` int(0) NOT NULL,

`type\_id` int(0) NOT NULL,

`start\_date` date NOT NULL,

`end\_date` date NOT NULL,

`price` decimal(10, 2) NOT NULL,

PRIMARY KEY (`price\_id`) USING BTREE,

INDEX `type\_id`(`type\_id`) USING BTREE,

CONSTRAINT `room\_price\_ibfk\_1` FOREIGN KEY (`type\_id`) REFERENCES `room\_type` (`type\_id`) ON DELETE RESTRICT ON UPDATE RESTRICT

) ENGINE = InnoDB CHARACTER SET = utf8mb4 COLLATE = utf8mb4\_0900\_ai\_ci ROW\_FORMAT = Dynamic;

-- ----------------------------

-- room\_price表的记录

-- ----------------------------

INSERT INTO `room\_price` VALUES (1, 1, '2023-06-21', '2023-06-30', 100.00);

INSERT INTO `room\_price` VALUES (2, 2, '2023-06-21', '2023-06-30', 150.00);

INSERT INTO `room\_price` VALUES (3, 3, '2023-06-21', '2023-06-30', 200.00);

INSERT INTO `room\_price` VALUES (4, 4, '2023-06-21', '2023-06-30', 300.00);

INSERT INTO `room\_price` VALUES (5, 5, '2023-06-21', '2023-06-30', 250.00);

INSERT INTO `room\_price` VALUES (6, 6, '2023-06-21', '2023-06-30', 180.00);

INSERT INTO `room\_price` VALUES (7, 7, '2023-06-21', '2023-06-30', 90.00);

INSERT INTO `room\_price` VALUES (8, 8, '2023-06-21', '2023-06-30', 120.00);

-- ----------------------------

-- vacant\_rooms\_view视图的结构

-- ----------------------------

DROP VIEW IF EXISTS `vacant\_rooms\_view`;

CREATE ALGORITHM = UNDEFINED SQL SECURITY DEFINER VIEW `vacant\_rooms\_view` AS select `ri`.`room\_id` AS `room\_id`,`ri`.`room\_number` AS `room\_number`,`rt`.`type\_name` AS `type\_name` from (`room\_info` `ri` join `room\_type` `rt` on((`ri`.`type\_id` = `rt`.`type\_id`))) where (`ri`.`status` = 'Vacant');

-- 触发器名称：check\_in\_trigger

-- 触发器功能：在插入顾客信息后，更新对应房间的状态为“Occupied”（已入住）

-- 触发时机：AFTER INSERT

DROP TRIGGER IF EXISTS `check\_in\_trigger`;

delimiter ;;

CREATE TRIGGER `check\_in\_trigger` AFTER INSERT ON `customer\_info` FOR EACH ROW BEGIN

UPDATE room\_info SET status = 'Occupied' WHERE room\_id = NEW.room\_id;

END

;;

delimiter ;

-- 触发器名称：check\_out\_trigger

-- 触发器功能：在更新顾客信息后，如果顾客的退房日期不为空，更新对应房间的状态为“Vacant”（空闲）

-- 触发时机：AFTER UPDATE

DROP TRIGGER IF EXISTS `check\_out\_trigger`;

delimiter ;;

CREATE TRIGGER `check\_out\_trigger` AFTER UPDATE ON `customer\_info` FOR EACH ROW BEGIN

IF NEW.check\_out\_date IS NOT NULL THEN

UPDATE room\_info SET status = 'Vacant' WHERE room\_id = NEW.room\_id;

END IF;

END

;;

delimiter ;

SET FOREIGN\_KEY\_CHECKS = 1;

-- 单表查询设计：

-- 查询表：customer\_info

-- 查询功能：根据顾客ID查询顾客信息

-- SQL 语句示例：

SELECT \* FROM customer\_info WHERE customer\_id = 1;

-- 多表查询：

-- 查询表：customer\_info、room\_info、room\_type、room\_price

-- 查询功能：查询顾客入住信息及对应房间类型和价格

SELECT ci.customer\_id, ci.check\_in\_date, ci.check\_out\_date, ri.room\_number, rt.type\_name, rp.price

FROM customer\_info ci

JOIN room\_info ri ON ci.room\_id = ri.room\_id

JOIN room\_type rt ON ri.type\_id = rt.type\_id

JOIN room\_price rp ON rt.type\_id = rp.type\_id;

-- 带参数的存储过程：

-- 存储过程名称：calculate\_cost

-- 存储过程功能：根据指定的起始日期和结束日期计算入住顾客的总天数和总费用

-- SQL 语句示例：

CREATE PROCEDURE calculate\_cost(IN start\_date DATE, IN end\_date DATE)

BEGIN

SELECT rt.type\_name, SUM(DATEDIFF(ci.check\_out\_date, ci.check\_in\_date)) AS total\_days, SUM(rp.price \* DATEDIFF(ci.check\_out\_date, ci.check\_in\_date)) AS total\_cost

FROM customer\_info ci

JOIN room\_info ri ON ci.room\_id = ri.room\_id

JOIN room\_type rt ON ri.type\_id = rt.type\_id

JOIN room\_price rp ON rt.type\_id = rp.type\_id AND ci.check\_in\_date >= rp.start\_date AND ci.check\_out\_date <= rp.end\_date

WHERE (ci.check\_in\_date BETWEEN start\_date AND end\_date) OR (ci.check\_out\_date BETWEEN start\_date AND end\_date)

GROUP BY rt.type\_name;

END;