数据库设计文档

[1. 引言 1](#_Toc127799138)

[1.1 编写目的 1](#_Toc127799139)

[1.2 项目来源 1](#_Toc127799140)

[1.3 文档约定 1](#_Toc127799141)

[1.4 预期读者和阅读建议 1](#_Toc127799142)

[1.5 参考资料 2](#_Toc127799143)

[2. 数据库命名规则 2](#_Toc127799144)

[3. 数据库设计说明 2](#_Toc127799145)

[3.1 数据库逻辑设计 2](#_Toc127799146)

[3.2 数据库分布 2](#_Toc127799148)

[3.3 基表设计 3](#_Toc127799149)

[3.4 视图设计 5](#_Toc127799150)

[3.5 索引设计 5](#_Toc127799151)

[4. 词汇表 5](#_Toc127799157)

[5. 历史数据处理 6](#_Toc127799158)

[附录 7](#_Toc127799158)

# 引言

本数据库设计说明书是关于“网络预约挂号”（Online Hospital Registration Booking）项目的数据库的设计，主要包括数据逻辑结构设计、数据字典以及运行环境、安全设计等。

## 编写目的

编写此数据库设计说明书的目的是展示“网络预约挂号”项目中各项功能实现时所需要的数据库，并可以此数据为据实现具体的数据存储。同时为项目开发人员、数据库管理人员等提供设计依据，了解本项目的数据库设计思路、数据库整体架构以及各种详细信息。

此文档的主要读者是：软件系统的开发人员、项目经理、测试人员、文档编写人员。

## 项目来源

* 任务提出者：指导老师
* 软件开发者：常馨悦
* 产品使用者：大众

## 文档约定

|  |  |
| --- | --- |
| **约定** | 描述 |
| 大写字母 | 首字母缩写词、缩写、特定命令名称以及键盘上的键名 |
| **粗体** | 重要操作 |
| 斜体 | 其他文档引用 |
| 等宽字体 | 示例命令行、程序代码、Web URL、文件名以及程序输出 |

## 预期读者和阅读建议

* 开发人员；
* 项目经理；
* 测试人员；
* 文档编写人员。

阅读建议：读者应通过该文档可以了解相关数据库设计的思路、数据库整体架构以及各种详细信息。

## 参考资料

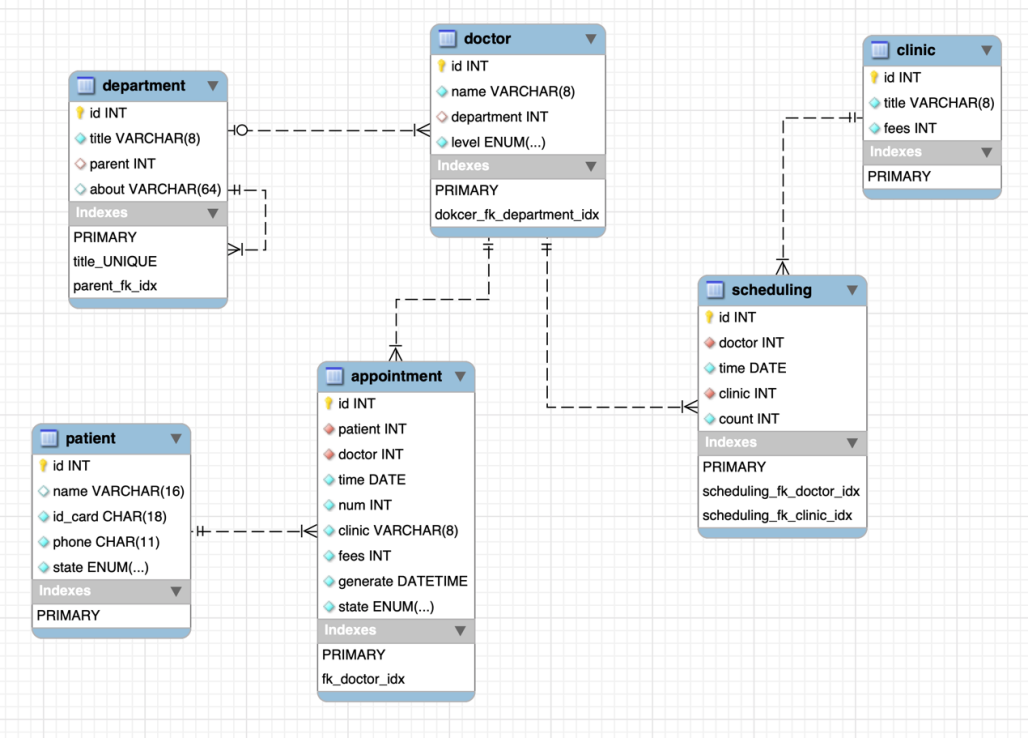
* “线上预约挂号”项目合同书；
* “线上预约挂号”项目计划任务书；
* 项目需求分析报告；
* 项目概要设计报告；
* 项目详细设计报告；
* 用户界面风格指导；
* 开发项目标难；
* 系统规格需求说明；
* 使用实例文档；

# 数据库命名规则

* 表名、字段名使用英文小写；
* 多个单词使用下划线分隔，如 id\_card；
* 非外键字段不要加表名前缀。

# 数据库设计说明

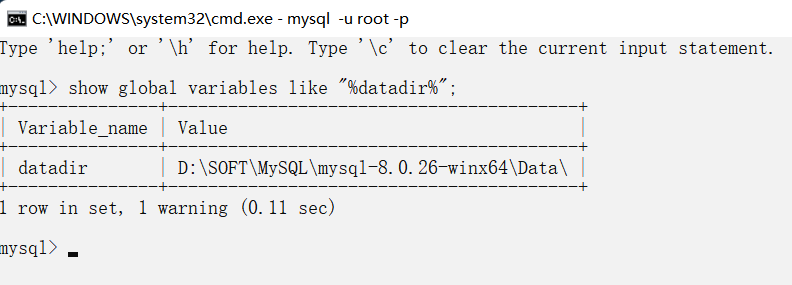
## 数据库逻辑设计



## 数据库分布

数据库分布采用一张表格进行描述，其格式如下：

|  |  |  |
| --- | --- | --- |
| 数据库  英文名称 | 数据库  中文名称 | 数据库  安装  物理位置 |
| Works | 工作表 | D:\SOFT\MySQL\mysql-8.0.26-winx64\Data\ |



## 基表设计

1. 科室表：department

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 是否空 | 默认值 | 键 | 描述 |
| id | int | 非空 | NULL | 主键 | 科室编号 |
| title | varchar(8) | 非空 | NULL |  | 科室名称 |
| parent | int | 空 | NULL | 外键 | 上一级科室 |
| about | varchar(64) | 空 | NULL |  | 科室信息 |

1. 门诊表：clinic

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 是否空 | 默认值 | 键 | 描述 |
| id | int | 非空 | NULL | 主键 | 门诊编号 |
| title | varchar(8) | 非空 | NULL |  | 门诊类型 |
| fees | int | 非空 | NULL |  | 门诊收费 |

1. 医生表：doctor

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 是否空 | 默认值 | 键 | 描述 |
| id | int | 非空 | NULL | 主键 | 医生编号 |
| name | varchar(8) | 非空 | NULL |  | 医生姓名 |
| department | int | 非空 | NULL | 外键 | 医生科室 |
| level | enum | 非空 | 住院医师 |  | 职称：  住院医师  主治医师  副主任医师  主任医师 |

1. 排班表：scheduling

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 是否空 | 默认值 | 键 | 描述 |
| id | int | 非空 | NULL | 主键 | 排班编号 |
| doctor | int | 非空 | NULL | 外键 | 医师编号 |
| clinic | int | 非空 | NULL | 外键 | 门诊类型编号 |
| time | date | 非空 | NULL |  | 出诊日期 |
| count | int | 非空 | 0 |  | 预约人数：不超过30 |

1. 就诊人表：patient

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 是否空 | 默认值 | 键 | 描述 |
| id | int | 非空 | NULL | 主键 | 就诊人编号 |
| name | varchar(16) | 非空 | NULL |  | 就诊人姓名 |
| id\_card | char(18) | 非空 | NULL |  | 身份证号码 |
| phone | char(11) | 非空 | NULL |  | 手机号 |
| state | enum | 非空 | 正常 |  | 状态：  正常  注销 |

1. 预约挂号表：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 类型 | 是否空 | 默认值 | 键 | 描述 |
| id | int | 非空 | NULL | 主键 | 预约编号 |
| patient | int | 非空 | NULL | 外键 | 就诊人编号 |
| doctor | int | 非空 | NULL | 外键 | 医生编号 |
| time | date | 非空 | NULL |  | 门诊日期 |
| num | int | 非空 | NULL |  | 预约号 |
| clinic | varchar(8) | 非空 | NULL |  | 诊室 |
| fees | int | 非空 | NULL |  | 挂号费 |
| generate | datetime | 非空 | now() |  | 预约时间 |
| state | enum | 预约 | 预约 |  | 状态：  预约  取消 |

## 视图设计

每个视图采用一张表格进行描述，其格式如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段编号 | 英文字段名 | 中文字段名 | 字段类型 | 字段源 |
| 1 | id | 编号 | int | scheduling.id |
| 2 | did | 医生编号 | int | doctor.id |
| 3 | doctor | 医生姓名 | varchar | doctor.name |
| 4 | time | 预约时间 | date | schedulinh.time |
| 5 | clinic | 诊室 | varchar | clinic.title |
| 6 | fees | 挂号费 | int | scheduling.fees |
| 7 | pid | 科室编号 | int | department.id |
| 8 | department | 科室名称 | varchar | department.title |
| 9 | count | 预约人数 | int | scheduling.count |
| 10 | level | 职称 | varchar | doctor.level |

## 索引设计

每个数据库的所有采用一张表格进行描述，其格式如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据库编号：1 | | | | |
| 索引  编号 | 基表名称 | 索引名称 | 字段集名称 | 备注 |
| 1 | appointment | fk\_doctor\_idx | `doctor` | BTREE |
| 2 | appointment | fk\_patient | `patient` | BTREE |
| 3 | department | parent\_fk\_idx | `parent` | BTREE |
| 4 | doctor | dokcer\_fk\_department\_idx | `department` | BTREE |
| 5 | scheduling | scheduling\_fk\_doctor\_idx | `doctor` | BTREE |
| 6 | scheduling | scheduling\_fk\_clinic\_idx | `clinic` | BTREE |

# 表名目录

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 表名（英文） | 描述 | 备注 |
| 1 | department | 医院科室 | 基础信息 |
| 2 | clinic | 科室门诊：普通、副主任、主任、专家等 | 基础信息 |
| 3 | doctor | 医生 | 基础信息 |
| 4 | patient | 就诊人 | 基础信息 |
| 5 | scheduling | 排班：医生出诊时间及门诊类型 | 操作 |
| 6 | appointment | 预约挂号 | 操作 |

# 历史数据处理

在创建新的数据库时，同时创建独立的历史表空间，所述历史表空间用于存储历史版本数据和事务信息。

附录：

1. 表结构： ddl.sql

-- MySQL Script generated by MySQL Workbench

-- Fri Jun 24 14:15:58 2022

-- Model: New Model Version: 1.0

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

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

-- Schema works

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

DROP SCHEMA IF EXISTS `works` ;

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

-- Schema works

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

CREATE SCHEMA IF NOT EXISTS `works` DEFAULT CHARACTER SET utf8mb4 ;

USE `works` ;

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

-- Table `works`.`department`

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

DROP TABLE IF EXISTS `works`.`department` ;

CREATE TABLE IF NOT EXISTS `works`.`department` (

`id` INT NOT NULL AUTO\_INCREMENT,

`title` VARCHAR(8) NOT NULL,

`parent` INT NULL,

`about` VARCHAR(64) NULL,

PRIMARY KEY (`id`),

INDEX `parent\_fk\_idx` (`parent` ASC) VISIBLE,

CONSTRAINT `parent\_fk`

FOREIGN KEY (`parent`)

REFERENCES `works`.`department` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `works`.`doctor`

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

DROP TABLE IF EXISTS `works`.`doctor` ;

CREATE TABLE IF NOT EXISTS `works`.`doctor` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(8) NOT NULL,

`department` INT NULL,

`level` ENUM('住院医师', '主治医师', '副主任医师', '主任医师') NOT NULL DEFAULT '住院医师',

PRIMARY KEY (`id`),

INDEX `dokcer\_fk\_department\_idx` (`department` ASC) VISIBLE,

CONSTRAINT `dokcer\_fk\_department`

FOREIGN KEY (`department`)

REFERENCES `works`.`department` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `works`.`patient`

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

DROP TABLE IF EXISTS `works`.`patient` ;

CREATE TABLE IF NOT EXISTS `works`.`patient` (

`id` INT NOT NULL AUTO\_INCREMENT,

`name` VARCHAR(16) NULL,

`id\_card` CHAR(18) NOT NULL,

`phone` CHAR(11) NOT NULL,

`state` ENUM('正常', '注销') NOT NULL DEFAULT '正常',

PRIMARY KEY (`id`))

ENGINE = InnoDB;

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

-- Table `works`.`clinic`

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

DROP TABLE IF EXISTS `works`.`clinic` ;

CREATE TABLE IF NOT EXISTS `works`.`clinic` (

`id` INT NOT NULL AUTO\_INCREMENT,

`title` VARCHAR(8) NOT NULL,

`fees` INT NOT NULL,

PRIMARY KEY (`id`))

ENGINE = InnoDB;

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

-- Table `works`.`scheduling`

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

DROP TABLE IF EXISTS `works`.`scheduling` ;

CREATE TABLE IF NOT EXISTS `works`.`scheduling` (

`id` INT NOT NULL AUTO\_INCREMENT,

`doctor` INT NOT NULL,

`time` DATE NOT NULL,

`clinic` INT NOT NULL,

`count` INT NOT NULL DEFAULT 0,

PRIMARY KEY (`id`),

INDEX `scheduling\_fk\_doctor\_idx` (`doctor` ASC) VISIBLE,

INDEX `scheduling\_fk\_clinic\_idx` (`clinic` ASC) VISIBLE,

CONSTRAINT `scheduling\_fk\_doctor`

FOREIGN KEY (`doctor`)

REFERENCES `works`.`doctor` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `scheduling\_fk\_clinic`

FOREIGN KEY (`clinic`)

REFERENCES `works`.`clinic` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

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

-- Table `works`.`appointment`

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

DROP TABLE IF EXISTS `works`.`appointment` ;

CREATE TABLE IF NOT EXISTS `works`.`appointment` (

`id` INT NOT NULL AUTO\_INCREMENT,

`patient` INT NOT NULL,

`doctor` INT NOT NULL,

`time` DATE NOT NULL,

`num` INT NOT NULL,

`clinic` VARCHAR(8) NOT NULL,

`fees` INT NOT NULL,

`generate` DATETIME NOT NULL DEFAULT now(),

`state` ENUM('预约', '取消') NOT NULL DEFAULT '预约',

PRIMARY KEY (`id`),

INDEX `fk\_doctor\_idx` (`doctor` ASC) VISIBLE,

CONSTRAINT `fk\_patient`

FOREIGN KEY (`patient`)

REFERENCES `works`.`patient` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_doctor`

FOREIGN KEY (`doctor`)

REFERENCES `works`.`doctor` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

1. 基础数据：data.sql

insert into department values(1,'内科',null,'');

insert into department values(2,'外科',null,'');

insert into department values(3,'妇产科',null,'');

insert into department values(4,'儿科',null,'');

insert into department values(null,'神经内科',1,'');

insert into department values(null,'呼吸内科',1,'');

insert into department values(null,'消化内科',1,'');

insert into department values(null,'心血管内科',1,'');

insert into department values(null,'普通外科',2,'');

insert into department values(null,'骨科',2,'');

insert into department values(null,'神经外科',2,'');

insert into department values(null,'妇科',3,'');

insert into department values(null,'产科',3,'');

insert into department values(null,'儿科',4,'');

insert into department values(null,'新生儿科',4,'');

insert into clinic values(null,'普通门诊',10);

insert into clinic values(null,'副主任门诊',25);

insert into clinic values(null,'主任门诊',35);

insert into clinic values(null,'知名专家',80);

insert into clinic values(null,'特需门诊',300);

insert into clinic values(null,'国际门诊',600);

insert into clinic values(null,'急症门诊',10);

insert into doctor(name,department,level) values('扁鹊',5,'主任医师');

insert into doctor(name,department,level) values('华佗',10,'主任医师');

insert into doctor(name,department,level) values('张仲景',6,'副主任医师');

insert into doctor(name,department,level) values('孙思邈',7,'主治医师');

insert into doctor(name,department,level) values('李时珍',8,'主治医师');

insert into doctor(name,department,level) values('东方宇辉',5,'主任医师');

insert into doctor(name,department,level) values('东方土锤',10,'副主任医师');

insert into doctor(name,department,level) values('东方呦呦',11,'副主任医师');

insert into doctor(name,department,level) values('东方顿顿',12,'副主任医师');

insert into doctor(name,department,level) values('东方七七',9,'副主任医师');

insert into doctor(name,department,level) values('东方蓓蓓',14,'主治医师');

insert into doctor(name,department,level) values('东方东东',9,'主治医师');

insert into doctor(name,department,level) values('东方杰西',11,'主治医师');

insert into doctor(name,department,level) values('东方大琪',13,'主治医师');

insert into doctor(name,department,level) values('东方老鱼',7,'住院医师');

insert into patient(name,id\_card,phone) values('赛文','430103200104020042','13100000000');

insert into patient(name,id\_card,phone) values('赛罗','430103200104020043','15011111111');

insert into patient(name,id\_card,phone) values('雷欧','430103200104020044','13122222222');

insert into patient(name,id\_card,phone) values('迪伽','430103200104020045','18633333333');

insert into patient(name,id\_card,phone) values('赛伽','430103200104020046','15044444444');

insert into patient(name,id\_card,phone) values('捷徳','430103200104020047','13755555555');

insert into patient(name,id\_card,phone) values('贝利亚','430103200104020048','13966666666');

insert into scheduling(doctor,time,clinic) values(1,'2022/7/4',1);

insert into scheduling(doctor,time,clinic) values(1,'2022/7/5',3);

insert into scheduling(doctor,time,clinic) values(1,'2022/7/6',3);

insert into scheduling(doctor,time,clinic) values(1,'2022/7/7',4);

insert into scheduling(doctor,time,clinic) values(1,'2022/7/8',4);

insert into scheduling(doctor,time,clinic) values(1,'2022/7/9',5);

insert into scheduling(doctor,time,clinic) values(2,'2022/7/4',4);

insert into scheduling(doctor,time,clinic) values(2,'2022/7/5',4);

insert into scheduling(doctor,time,clinic) values(2,'2022/7/6',4);

insert into scheduling(doctor,time,clinic) values(2,'2022/7/7',5);

insert into scheduling(doctor,time,clinic) values(2,'2022/7/8',3);

insert into scheduling(doctor,time,clinic) values(2,'2022/7/9',1);

insert into scheduling(doctor,time,clinic) values(3,'2022/7/4',2);

insert into scheduling(doctor,time,clinic) values(3,'2022/7/5',4);

insert into scheduling(doctor,time,clinic) values(3,'2022/7/6',2);

insert into scheduling(doctor,time,clinic) values(3,'2022/7/7',2);

insert into scheduling(doctor,time,clinic) values(3,'2022/7/8',2);

insert into scheduling(doctor,time,clinic) values(3,'2022/7/9',2);

insert into scheduling(doctor,time,clinic) values(4,'2022/7/4',1);

insert into scheduling(doctor,time,clinic) values(4,'2022/7/5',1);

insert into scheduling(doctor,time,clinic) values(4,'2022/7/6',1);

insert into scheduling(doctor,time,clinic) values(4,'2022/7/7',1);

insert into scheduling(doctor,time,clinic) values(4,'2022/7/8',1);

insert into scheduling(doctor,time,clinic) values(4,'2022/7/9',1);

insert into scheduling(doctor,time,clinic) values(5,'2022/7/4',1);

insert into scheduling(doctor,time,clinic) values(5,'2022/7/5',1);

insert into scheduling(doctor,time,clinic) values(5,'2022/7/6',1);

insert into scheduling(doctor,time,clinic) values(5,'2022/7/7',1);

insert into scheduling(doctor,time,clinic) values(5,'2022/7/8',1);

insert into scheduling(doctor,time,clinic) values(5,'2022/7/9',1);

insert into scheduling(doctor,time,clinic) values(6,'2022/7/4',1);

insert into scheduling(doctor,time,clinic) values(6,'2022/7/5',3);

insert into scheduling(doctor,time,clinic) values(6,'2022/7/6',4);

insert into scheduling(doctor,time,clinic) values(6,'2022/7/7',3);

insert into scheduling(doctor,time,clinic) values(6,'2022/7/8',5);

insert into scheduling(doctor,time,clinic) values(6,'2022/7/9',6);

insert into scheduling(doctor,time,clinic) values(7,'2022/7/4',2);

insert into scheduling(doctor,time,clinic) values(7,'2022/7/5',7);

insert into scheduling(doctor,time,clinic) values(7,'2022/7/6',2);

insert into scheduling(doctor,time,clinic) values(7,'2022/7/7',2);

insert into scheduling(doctor,time,clinic) values(7,'2022/7/8',4);

insert into scheduling(doctor,time,clinic) values(7,'2022/7/9',2);

insert into scheduling(doctor,time,clinic) values(8,'2022/7/4',2);

insert into scheduling(doctor,time,clinic) values(8,'2022/7/5',4);

insert into scheduling(doctor,time,clinic) values(8,'2022/7/6',7);

insert into scheduling(doctor,time,clinic) values(8,'2022/7/7',2);

insert into scheduling(doctor,time,clinic) values(8,'2022/7/8',2);

insert into scheduling(doctor,time,clinic) values(8,'2022/7/9',2);

insert into scheduling(doctor,time,clinic) values(9,'2022/7/4',2);

insert into scheduling(doctor,time,clinic) values(9,'2022/7/5',2);

insert into scheduling(doctor,time,clinic) values(9,'2022/7/6',4);

insert into scheduling(doctor,time,clinic) values(9,'2022/7/7',7);

insert into scheduling(doctor,time,clinic) values(9,'2022/7/8',2);

insert into scheduling(doctor,time,clinic) values(9,'2022/7/9',2);

insert into scheduling(doctor,time,clinic) values(10,'2022/7/4',2);

insert into scheduling(doctor,time,clinic) values(10,'2022/7/5',2);

insert into scheduling(doctor,time,clinic) values(10,'2022/7/6',2);

insert into scheduling(doctor,time,clinic) values(10,'2022/7/7',4);

insert into scheduling(doctor,time,clinic) values(10,'2022/7/8',7);

insert into scheduling(doctor,time,clinic) values(10,'2022/7/9',2);

insert into scheduling(doctor,time,clinic) values(11,'2022/7/4',7);

insert into scheduling(doctor,time,clinic) values(11,'2022/7/5',1);

insert into scheduling(doctor,time,clinic) values(11,'2022/7/6',1);

insert into scheduling(doctor,time,clinic) values(11,'2022/7/7',1);

insert into scheduling(doctor,time,clinic) values(11,'2022/7/8',1);

insert into scheduling(doctor,time,clinic) values(11,'2022/7/9',1);

insert into scheduling(doctor,time,clinic) values(12,'2022/7/4',1);

insert into scheduling(doctor,time,clinic) values(12,'2022/7/5',7);

insert into scheduling(doctor,time,clinic) values(12,'2022/7/6',1);

insert into scheduling(doctor,time,clinic) values(12,'2022/7/7',1);

insert into scheduling(doctor,time,clinic) values(12,'2022/7/8',1);

insert into scheduling(doctor,time,clinic) values(12,'2022/7/9',1);

insert into scheduling(doctor,time,clinic) values(13,'2022/7/4',1);

insert into scheduling(doctor,time,clinic) values(13,'2022/7/5',1);

insert into scheduling(doctor,time,clinic) values(13,'2022/7/6',7);

insert into scheduling(doctor,time,clinic) values(13,'2022/7/7',1);

insert into scheduling(doctor,time,clinic) values(13,'2022/7/8',1);

insert into scheduling(doctor,time,clinic) values(13,'2022/7/9',1);

insert into scheduling(doctor,time,clinic) values(14,'2022/7/4',1);

insert into scheduling(doctor,time,clinic) values(14,'2022/7/5',1);

insert into scheduling(doctor,time,clinic) values(14,'2022/7/6',1);

insert into scheduling(doctor,time,clinic) values(14,'2022/7/7',7);

insert into scheduling(doctor,time,clinic) values(14,'2022/7/8',1);

insert into scheduling(doctor,time,clinic) values(14,'2022/7/9',1);、

1. v\_scheduling视图建立

create view v\_scheduling

as

select s.id id,d.id did, d.name doctor,time,c.title clinic,fees,p.id pid,p.title department, count

from doctor d

join scheduling s

on s.doctor=d.id

join clinic c

on s.clinic=c.id

join department p

on d.department=p.id;