# 数据库设计

## 数据库信息：meeting

用户名：meeting

密码：meeting

## User表：

Head-varchar（45）（头像路径）

Name-varchar（25）

Password-varchar(存为md5加密后的密码25)

Gender-varchar（10）

Depart-varchar（25）

Email-varchar(前台对用户输入的email地址进行验证)（45）

Phone-int(前台对用户输入的密码进行验证)（15）

ID-int(假定为4位数字，从1001开始，9999结束)(4)

Working-boolean(true表示工作，false表示离职)

Remark-varchar(备注，留出的一列，一般为空)(25)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 字段含义 | 主键 | 字段类型 | 是否为空 |
| id | 用户工号 | 是 | int | 否 |
| name | 用户名 |  | varchar(25) | 否 |
| password | 密码 |  | varchar(25) | 否 |
| head | 头像路径 |  | varchar(45) | 否 |
| gender | 性别 |  | varchar(25) | 否 |
| depart | 部门 |  | varchar(25) | 否 |
| email | 邮箱 |  | varchar(25) | 否 |
| phone | 电话 |  | varchar(25) | 否 |
| working | 工作状态 |  | boolean | 否 |
| remark | 备注 |  | varchar(25) | 是 |

CREATE TABLE IF NOT EXISTS `user` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(25) NOT NULL,

`password` varchar(25) NOT NULL,

`head` varchar(45) NOT NULL,

`gender` varchar(25) NOT NULL,

`dapart` varchar(25) NOT NULL,

`email` varchar(25) NOT NULL,

`phone` varchar(25) NOT NULL,

`working` tinyint(1) NOT NULL,

`remark` varchar(25) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=1 ;

## Admin表：

Head-varchar（45）（头像路径）

Super-boolean（表示是否为超级管理员）

Name-varchar（25）

Password-varchar(存为md5加密后的密码25)

Gender-varchar（10）

Depart-varchar（25）

Email-varchar(前台对用户输入的email地址进行验证)（45）

Phone-int(前台对用户输入的密码进行验证)（15）

ID-int(假定为4位数字，从1001开始，9999结束)(4)

Working-boolean(true表示工作，false表示离职)

Remark-varchar(备注，留出的一列，一般为空)(25)

初始化一名超级管理员，ID1000

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 字段含义 | 主键 | 字段类型 | 是否为空 |
| id | 用户工号 | 是 | int | 否 |
| name | 用户名 |  | varchar(25) | 否 |
| password | 密码 |  | varchar(25) | 否 |
| head | 头像路径 |  | varchar(45) | 否 |
| gender | 性别 |  | varchar(25) | 否 |
| depart | 部门 |  | varchar(25) | 否 |
| email | 邮箱 |  | varchar(25) | 否 |
| phone | 电话 |  | varchar(25) | 否 |
| working | 工作状态 |  | boolean | 否 |
| super | 超级管理员 |  | boolean | 否 |
| remark | 备注 |  | varchar(25) | 是 |

CREATE TABLE IF NOT EXISTS `admin` (

`id` int(11) NOT NULL AUTO\_INCREMENT,

`name` varchar(25) NOT NULL,

`password` varchar(25) NOT NULL,

`head` varchar(45) NOT NULL,

`gender` varchar(25) NOT NULL,

`depart` varchar(25) NOT NULL,

`email` varchar(25) NOT NULL,

`phone` varchar(25) NOT NULL,

`working` tinyint(1) NOT NULL,

`super` tinyint(1) NOT NULL,

`remark` varchar(25) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO\_INCREMENT=1;

## preUser表：

Head-varchar（45）（头像路径）

Name-varchar（25）

Psd-varchar(存为md5加密后的密码25)

Gender-varchar（10）

Depart-varchar（25）

Email-varchar(前台对用户输入的email地址进行验证)（45）

Phone-int(前台对用户输入的密码进行验证)（15）

ID-int(假定为4位数字，从1001开始，9999结束)(4)

Remark-varchar(备注，留出的一列，一般为空)(25)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 字段含义 | 主键 | 字段类型 | 是否为空 |
| id | 用户工号 | 是 | int | 否 |
| name | 用户名 |  | varchar(25) | 否 |
| password | 密码 |  | varchar(25) | 否 |
| head | 头像路径 |  | varchar(45) | 否 |
| gender | 性别 |  | varchar(25) | 否 |
| depart | 部门 |  | varchar(25) | 否 |
| email | 邮箱 |  | varchar(25) | 否 |
| phone | 电话 |  | varchar(25) | 否 |
| remark | 备注 |  | varchar(25) | 是 |

CREATE TABLE IF NOT EXISTS `preuser` (

`id` int(11) NOT NULL,

`name` varchar(25) NOT NULL,

`password` varchar(25) NOT NULL,

`head` varchar(45) NOT NULL,

`gender` varchar(25) NOT NULL,

`depart` varchar(25) NOT NULL,

`email` varchar(25) NOT NULL,

`phone` varchar(25) NOT NULL,

`remark` varchar(25) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

## MeetingRoom表

ID-int（会议室编号）

housenumber-int（门牌号）

Limit-int（人数上限）

State（状态（开放、维护、永久关闭））

（再议）Schedule-varchar（存实例化后的对象的路径，对象为3天内的时间安排）

Remark-varchar(备注，留出的一列，一般为空)(25)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 字段含义 | 主键 | 字段类型 | 是否为空 |
| id | 会议室id | 是 | int | 否 |
| housenumber | 门牌号 |  | int | 否 |
| limit | 容纳人数 |  | int | 否 |
| state | 状态 |  | varchar(25) | 否 |
| schedule | 时间表路径 |  | varchar(25) | 否 |
| remark | 备注 |  | varchar(25) | 是 |

CREATE TABLE IF NOT EXISTS `meetingroom` (

`id` int(11) NOT NULL,

`housenumber` int(11) NOT NULL,

`limit` int(11) NOT NULL,

`state` varchar(25) NOT NULL,

`schedule` varchar(25) NOT NULL,

`remark` varchar(25) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

## MeetingApply表：

Meeting-ID(会议编号)

Title-varchar（会议名称）

Date-date申请日期）

Description-varchar（会议介绍）

During-int（会议时长）

TotalNumber-int（人数）

People(参会人员)

Sponsor（发起者）

State-varchar（存储申请表当前的状态（未审核，审核未通过，审核通过））

Remark-varchar（备注）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 字段含义 | 主键 | 字段类型 | 是否为空 |
| id | 会议编号 | 是 | int | 否 |
| title | 会议名称 |  | varchar(25) | 否 |
| date | 会议日期 |  | date | 否 |
| during | 会议时长 |  | int | 否 |
| totalnumber | 参与人数 |  | int | 否 |
| people | 参与人员 |  | varchar(100) | 否 |
| sponsor | 发起者 |  | varchar(25) | 否 |
| state | 状态 |  | varchar(25) | 否 |
| description | 会议简介 |  | varchar(100) | 否 |
| remark | 备注 |  | varchar(25) | 是 |

CREATE TABLE IF NOT EXISTS `meetingapply` (

`id` int(11) NOT NULL,

`title` varchar(25) NOT NULL,

`date` date NOT NULL,

`during` int(11) NOT NULL,

`totalnumber` int(11) NOT NULL,

`people` varchar(100) NOT NULL,

`sponsor` varchar(25) NOT NULL,

`state` varchar(25) NOT NULL,

`description` varchar(100) NOT NULL,

`remark` varchar(25) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

## Meeting表

（储存通过审核并选择了会议室的会议）：

Meeting-ID(会议编号)

RoomID(会议室编号)

Date-varchar（申请日期）

Title-varchar（会议名称）

Description-varchar（会议介绍）

Start-Time（会议开始时间）

During-int（会议时长）

TotalNumber-int（人数）

People(参会人员)

Sponsor（发起者）

State-int（存储会议当前的状态（已申请会议室还没开始、会议中、已结束））

Remark-varchar（备注）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 字段含义 | 主键 | 字段类型 | 是否为空 |
| id | 会议编号 | 是 | int | 否 |
| title | 会议名称 |  | varchar(25) | 否 |
| date | 会议日期 |  | date | 否 |
| start | 开始时刻 |  | time | 否 |
| during | 会议时长 |  | int | 否 |
| housenumber | 会议室编号 |  | int | 否 |
| totalnumber | 参与人数 |  | int | 否 |
| people | 参与人员 |  | varchar(100) | 否 |
| sponsor | 发起者 |  | varchar(25) | 否 |
| state | 状态 |  | varchar(25) | 否 |
| description | 会议简介 |  | varchar(100) | 否 |
| remark | 备注 |  | varchar(25) | 是 |

CREATE TABLE `meeting` (

`id` int(11) NOT NULL,

`title` varchar(25) NOT NULL,

`date` date NOT NULL,

`start` time NOT NULL,

`during` int(11) NOT NULL,

`room` int(11) NOT NULL,

`totalnumber` int(11) NOT NULL,

`people` varchar(100) NOT NULL,

`sponsor` varchar(25) NOT NULL,

`state` varchar(25) NOT NULL,

`description` varchar(100) NOT NULL,

`remark` varchar(25) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

## Notice表：

Notice-id

Date-date（通知日期）

State- Varchar（已查看、未查看）

Type-varchar（通知类型：会议邀请、审查状况。。。）

Information(通知的内容)

From-int（来源）

Remark-varchar(备注，留出的一列，一般为空)(25)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 字段含义 | 主键 | 字段类型 | 是否为空 |
| id | 通知编号 | 是 | int | 否 |
| information | 通知内容 |  | varchar(25) | 否 |
| date | 通知日期 |  | date | 否 |
| type | 通知类型 |  | varchar(25) | 否 |
| from | 通知来源 |  | varchar(25) | 否 |
| state | 通知状态 |  | varchar(25 | 否 |
| remark | 备注 |  | varchar(25) | 是 |

CREATE TABLE `notice` (

`id` int(11) NOT NULL,

`information` varchar(45) NOT NULL,

`date` date NOT NULL,

`type` varchar(25) NOT NULL,

`from` varchar(25) NOT NULL,

`state` varchar(25) NOT NULL,

`remark` varchar(25) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

**Depart表：**

Id-int（部门编号）

Name-varchar（15）

TotalNumber-int（人数）

People（部门人员）

State-boolean（true表示存在，false表示不存在）

Remark-varchar(备注，留出的一列，一般为空)(25)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段 | 字段含义 | 主键 | 字段类型 | 是否为空 |
| id | 通知编号 | 是 | int | 否 |
| name | 部门名称 |  | varchar(25) | 否 |
| totalnumber | 总人数 |  | Int | 否 |
| people | 成员 |  | varchar(100) | 是 |
| state | 状态 |  | boolean | 否 |
| remark | 备注 |  | varchar(25) | 是 |

CREATE TABLE `depart` (

`id` int(11) NOT NULL,

`name` varchar(25) NOT NULL,

`totalnumber` int(11) NOT NULL,

`people` varchar(100) DEFAULT NULL,

`state` tinyint(4) NOT NULL,

`remark` varchar(25) DEFAULT NULL,

PRIMARY KEY (`id`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;