

自助工时管理系统项目数据库设计报告



**系别：信息技术系**

**班级：15级计算机科学与技术1班**

**数据库设计：12号，黄丽华**

目录

[ 1](#_Toc533263473)

[自助工时管理系统项目数据库设计报告 1](#_Toc533263474)

[1 需求分析 3](#_Toc533263475)

[2 概念分析 3](#_Toc533263476)

[1职员实体及属性 3](#_Toc533263477)

[2团队实体及属性 3](#_Toc533263478)

[3子团队实体及属性 4](#_Toc533263479)

[4工时实体及其属性 4](#_Toc533263480)

[3 逻辑分析 4](#_Toc533263481)

[4 逻辑设计阶段 5](#_Toc533263482)

[1. 关系模型的转化 5](#_Toc533263483)

[2. 模式设计 5](#_Toc533263484)

[3. 数据结构设计 5](#_Toc533263485)

[5 物理设计阶段 8](#_Toc533263486)

[1. 物理设计阶段的任务 8](#_Toc533263487)

[6 系统实现 8](#_Toc533263488)

[6.1 建立数据库 8](#_Toc533263489)

[6.2 建立数据表 8](#_Toc533263490)

[6.3 关系图 10](#_Toc533263491)

[7 主要代码 10](#_Toc533263492)

[7.1 分页查询 10](#_Toc533263493)

[7.2 统计图查询（统计图，散点图，饼图） 10](#_Toc533263495)

[7.3 id查询 12](#_Toc533263498)

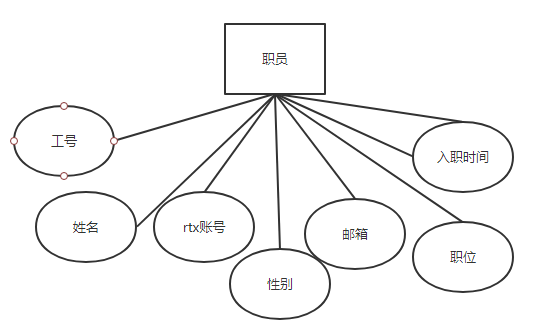
[8 总结 12](#_Toc533263499)

1. 需求分析

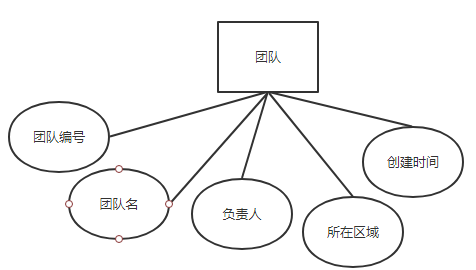
见《需求分析报告》

1. 概念分析

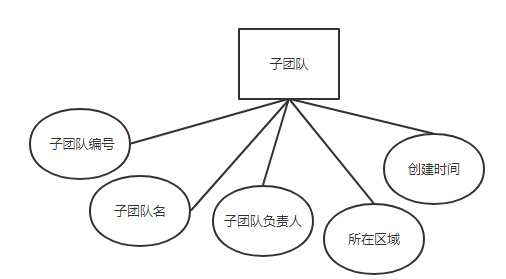
## 1职员实体及属性



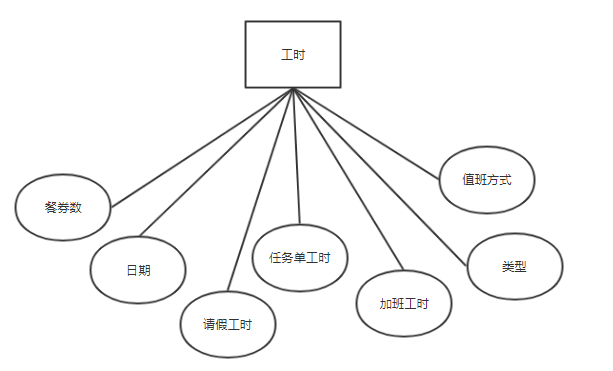
## 2团队实体及属性



## 3子团队实体及属性

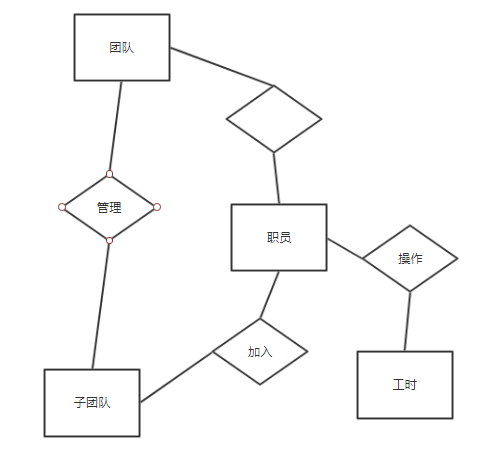


## 4工时实体及其属性



1. 逻辑分析

E-R图



1. 逻辑设计阶段
   1. 关系模型的转化

职员基本信息（职员编号，名字，性别，邮箱，rtx账号，职位，入职时间）

团队基本信息（团队编号，名字，团队负责人，所在区域，创建时间）

团队基本信息（子团队编号，子团队名字，子团队负责人，所在区域，创建时间）

工时基本信息（职员编号，团队编号，餐券数，日期，请假工时，任务单工时，加班工时，值班方式，类别）

* 1. 模式设计

1. 数据库设计

Hez=(staff,team,subteam,staffteam,overtime)

1. 表设计

staff=(id,name,password,jobnumber,gender,entrytime,phone,email,rtxaccount,not ificationway, notes,position,merit,hard,points)

team=( id, code,name,responsible,place,createTime,notes)

staffteam=(id,staff\_id,team\_id)

teamsub=(id,team\_id,name,code,responsible,place,createTime,notes) overtime=(id, staff\_id, team\_id, date, staffteam\_id, mealcoupon, dayoff, assignment, startime, endtime, duration, type, dutymode, notes)

* 1. 数据结构设计

|  |  |  |  |
| --- | --- | --- | --- |
| 职员信息表（staff） | | | |
| **字段** | **类型** | **约束** | **字段含义** |
| id | int | 主键（PK\_user）自动增长 | 用户id |
| name | varchar(8) | Not noll | 名字 |
| password | int | Not null | 密码 |
| jobnumber | VARCHAR(10) | Not null | 用户密码 |
| gender | varchar(2) |  | 性别 |
| entrytime | datetime |  | 进入时间(yyyy-MM-dd) |
| email | varchar(10) |  | 邮箱 |
| phone | varchar(11) |  | 性别 |
| rtxaccount | varchar(12) |  | rtx账号 |
| notificationway | varchar(4) |  | 通知方式 |
| notes | varchar(MAX) |  | 注释 |
| position | varchar(10) |  | 职位 |
| merite | int |  | 功绩 |
| hard | int |  | 苦劳 |
| points | int |  | 积分 |
|  |  |  |  |
| 团队表(team) |  |  |  |
| **字段** | **类型** | **约束** | **字段含义** |
| id | int | 主键(PK\_resume)/自动增长 | resume的id |
| code | varchar(10) | Not null | 团队编号 |
| name | varchar(8) | Not null | 名字 |
| responsible | varchar(8) |  | 负责人 |
| palce | varchar(50) |  | 所在区域 |
| createTime | datetime |  | 创建时间 |
| notes | varchar(50) |  |  |
| 子团队表（teamSub） |  |  |  |
| **字段** | **类型** | **约束** | **字段含义** |
| id | int | 主键(PK\_teamSub) | 子团队id |
| team\_id | int | 外键（PK\_team\_teamSub） | 团队id |
| code | varchar(10) | Not null | 团队编号 |
| name | varchar(10 | Not null | 名字 |
| responsible | varchar(8) |  | 负责人 |
| palce | varchar(50) |  | 所在区域 |
| createTime | datetime |  | 创建时间 |
| notes | varchar(50) |  |  |
| 中间表staffteam |  |  |  |
| **字段** | **类型** | **含义** | **字段含义** |
| id | int | 主键（PK\_staffteam）/自动增长 | 职位id |
| staff\_id | int | 外键(FK\_staff\_staffteam) | 员工表id |
| team\_id | int | 外键(FK\_team\_staffteam) | 团队表id |
|  |  |  |  |
| 工时表(overtime) |  |  |  |
| 字段 | 类型 | 备注 | **字段描述** |
| id | int | 主键(FK\_overtime) | 公司id |
| staff\_id | int | 外键(FK\_staff\_overtime) | 职位id |
| team\_id | int | 外键(FK\_team\_overtime) | 团队id |
| staffteam\_id | int | 外键(FK\_staffteam\_overtime) | 中间表id |
| date | datetime |  | 日期 |
| mealcoupon | int |  | 餐券数 |
| dayoff | int |  | 请假工时时长 |
| assignment | int |  | 任务单工时时长 |
| starttime | varchar(5) |  | 加班工时的开始时间 |
| endtime | varchar(5) |  | 加班工时的结束时间 |
| duration | int |  | 加班工时时长 |
| type | varchar(50) |  | 类型 |
| dutymode | varchar(4) |  | 值班方式 |
| notes | varchar(50) |  | 注释 |

1. 物理设计阶段
   1. 物理设计阶段的任务
2. 确定数据库的物理结构,在关系数据库中主要是存取方法和存储结构;
3. 物理结构进行评价，评价的重点是时间和空间效率。
4. 系统实现
   1. 建立数据库

create database hez;

* 1. 建立数据表

Overtime表建立

CREATE TABLE [dbo].[overtime](

[id] [int] IDENTITY(1,1) NOT NULL,

[staffteam\_id] [int] NOT NULL,

[team\_id] [int] NOT NULL,

[staff\_id] [int] NOT NULL,

[date] [datetime] NULL,

[mealcoupon] [int] NULL,

[dayoff] [int] NULL,

[assignment] [int] NULL,

[startime] [varchar](5) NULL,

[endtime] [varchar](5) NULL,

[duration] [int] NULL,

[type] [varchar](50) NULL,

[dutymode] [varchar](4) NULL,

[notes] [varchar](50) NULL,

CONSTRAINT [PK\_overtime] PRIMARY KEY CLUSTERED

)

Staff表建立

CREATE TABLE [dbo].[staff](

[id] [int] IDENTITY(1,1) NOT NULL,

[name] [varchar](8) NULL,

[password] [varchar](32) NULL,

[jobnumber] [varchar](10) NULL,

[gender] [varchar](2) NULL,

[entrytime] [datetime] NULL,

[email] [varchar](32) NULL,

[phone] [varchar](11) NULL,

[rtxaccount] [varchar](4) NULL,

[notificationway] [varchar](32) NULL,

[notes] [varchar](max) NULL,

[position] [varchar](10) NULL,

[merit] [int] NULL,

[hard] [int] NULL,

[points] [int] NULL,

CONSTRAINT [PK\_staff] PRIMARY KEY CLUSTERED

)

Staffteam表建立

CREATE TABLE [dbo].[staffteam](

[id] [int] IDENTITY(1,1) NOT NULL,

[staff\_id] [int] NOT NULL,

[team\_id] [int] NOT NULL,

CONSTRAINT [PK\_staffteam] PRIMARY KEY CLUSTERED

)

CREATE TABLE [dbo].[team](

[id] [int] IDENTITY(1,1) NOT NULL,

[code] [varchar](10) NULL,

[name] [varchar](8) NULL,

[responsible] [varchar](8) NULL,

[place] [varchar](50) NULL,

[createTime] [datetime] NULL,

[notes] [varchar](50) NULL CONSTRAINT [DF\_team\_notes] DEFAULT ('无'),

CONSTRAINT [PK\_team] PRIMARY KEY CLUSTERED

)

teamSub表建立

CREATE TABLE [dbo].[teamSub](

[id] [int] IDENTITY(1,1) NOT NULL,

[team\_id] [int] NULL,

[name] [varchar](10) NULL,

[code] [varchar](50) NULL,

[responsible] [varchar](8) NULL,

[place] [varchar](50) NULL,

[createTime] [datetime] NULL,

[notes] [varchar](50) NULL CONSTRAINT [DF\_teamSub\_notes] DEFAULT ('无'),

CONSTRAINT [PK\_teamSub] PRIMARY KEY CLUSTERED

)

* 1. 关系

overtime.staff\_id=staff.id

ovetime.team\_id=team.id

overtime.staffteam\_id=staffteam.id

subTeam\_team\_id= team.id

1. 主要代码
   1. 分页查询：显示从~14行,改变rownumber>?，决定接下来的行记录的起始位置。

select top 14 o.\* from (select row\_number() over(order by id asc) as rownumber,

\* from(select \* from [overtime]) as ot ) as o

where rownumber>0;

* 1. 统计图查询（直方图，散点图，饼图，排序图）：

近三个月加班工时

select convert(varchar,month(date))+'月' as monthDate ,team\_id,sum(duration) as totalduration

from overtime

where staff\_id=1 and date>(select dateadd(month,-3,getdate()))

group by month(date), team\_id

order by monthDate,team\_id

我的加班工时，请假工时，任务单工时

select convert(varchar,year(date))+'年'+convert(varchar,month(date))+'月' as date,team\_id,sum(duration) as totalduration

from overtime

where staff\_id=1

group by year(date),month(date), team\_id

order by year(date),month(date)

select convert(varchar,year(date))+'年'+convert(varchar,month(date))+'月' as date,team\_id,sum(dayoff) as dayoff

from overtime

where staff\_id=1

group by year(date),month(date), team\_id

order by year(date),month(date)

select convert(varchar,year(date))+'年'+convert(varchar,month(date))+'月' as year\_month,team\_id,sum(assignment) as assignment

from overtime

where staff\_id=1

group by year(date),month(date), team\_id

order by year(date),month(date)

饼图

select team\_id,sum(duration)as totalduration

from overtime

where date>='2018-5-11' and date<='2019-1-1'

group by team\_id

散点图

select staff\_id,sum(duration)as totalduration ,count(duration) as times

from overtime

group by staff\_id

排序图

select staff\_id,sum(duration)as totalduration,count(duration) as times

from overtime

group by staff\_id

order by totalduration asc, times desc

* 1. id查询

select id from staff where name='刘喜云'

select id from staff where jobnumber='2015911112'

select id from team where name='\*\*\*'

staffteam id

select \* from staffteam where staff\_id =1

select \* from staffteam where team\_id = 1

找符合条件的overtime记录

select \* from overtime where staffteam\_id in (2,3)

1. 总结

数据库，**是存放数据的地**方，一个好的数据库才能更好为我们的操作提供访问，存储的能力，使我们对数据的存取更方便。所以在设计时，应从范式，扩展性，约束，访问权限等方面全面考虑周全。范式的设计，可以减少数据冗余从而节约存储空间提高查询效率，同时也使得数据一致性容易得到维护；非空约束，唯一性约束都是常见的约束。它们的存在避免了系统发生空异常，并且使得数据更好维护。良好的数据库设计，节省数据的存储空间，能够保证数据的完整性。

从需求分析阶段开始，数据库共有逻辑设计，物理设计和实现维护阶段四个生命周期。数据库的需求分析，在“需求分析报告”中详细介绍，本报告从逻辑设计阶段对数据的设计进行了介绍。需求分析阶段的分析是否正确，决定了数据库设计的准确性。我使用ER图创建概念数据模型图，展示所有数据以及数据间关系。在物理设计阶段，根据特定数据库管理系统所提供的多种存储结构和存取方法等依赖于具体计算机结构的各项物理设计措施，对具体的应用任务选定最合适的物理存储结构。当设计完成之后，使用数据库管理系统（DBMS）中的数据定义语言（DDL）来创建数据结构。数据库创建完成后，通过sql server实现该数据库。数据库开始运行后，对其性能进行监视。并针对功能做出设计与修改。