# 数据库设计部分

课程设计题目：实现职业分类、职业信息、用人单位信息管理；实现求职者信息、职业匹配管理；实现费用管理

要求：

1. 实现职业分类、职业信息管理（职业号、职业类型号、用人单位、需求人数、已招聘人数、备注）
2. 实现用人单位、求职者信息管理、求职者信息设有聘用标志，默认值为未聘用
3. 实现求职者和职业匹配管理（职业号、求职人员编号）
4. 实现费用管理，包括用人单位交费情况、求职人员交费情况
5. 创建存储查询各种职业的需求数和已招聘人数
6. 建立表间关系

数据表以及设计

1. 职业分类表Classification（职业类型号，职业类型名）

|  |  |
| --- | --- |
| 职业类型号 | 职业类型名 |
| 1 | 系统架构师 |
| 2 | 会计师 |
| 3 | 创意设计师 |

1. 用人单位表Employer（单位编号，单位名称，职业类型号）

|  |  |  |
| --- | --- | --- |
| 单位编号 | 单位名称 | 职业类型号 |
| 1 | 广告公司 | 3 |
| 2 | 电视台 | 1 |
| 3 | 教育培训 | 2 |
| 4 | 软件公司 | 1 |
| 5 | 网络公司 | 3 |
| 6 | 集团公司 | 2 |

1. 职业信息表

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 职业号 | 职业名称 | 单位编号 | 职业类型号 | 专业要求 | 已聘人数 | 需求人数 | 工资 |
| 1 | 平面设计 | 1 | 3 | 教育经验 | 3 | 5 | 5000 |
| 2 | 媒体广告 | 5 | 3 | 广告学 | 1 | 3 | 6000 |
| 3 | 会计 | 3 | 2 | 财务管理 | 4 | 10 | 4500 |
| 4 | 财务助理 | 6 | 2 | 不限 | 2 | 4 | 4000 |
| 5 | 软件工程师 | 4 | 1 | 软件工程 | 5 | 7 | 6500 |
| 6 | 网络安全 | 2 | 1 | 计算机科学 | 2 | 3 | 5500 |

1. 介绍人员表

|  |  |  |
| --- | --- | --- |
| 介绍人员编号 | 姓名 | 电话 |
| 1 | 李四 | 136388888 |
| 2 | 张三 | 138384333 |

1. 求职者信息表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 求职者编号 | 姓名 | 性别 | 攻读专业 | 是否聘用 | 月薪要求 |
| 1 | 孙华 | 男 | 财务管理 |  | 4500 |
| 2 | 陈茜 | 女 | 教育专业 |  | 5000 |
| 3 | 孙琦 | 女 | 广告学 |  | 5500 |
| 4 | 刘富文 | 男 | 安全工程 | 是 | 6000 |
| 5 | 王小杨 | 女 | 英语 |  | 4000 |
| 6 | 李华 | 男 | 软件工程 |  | 6500 |
| 7 | 余慧 | 男 | 通信工程 | 是 | 4000 |
| 8 | 张琪 | 男 | 计算机科学 |  | 5500 |

1. 根据表格画出ER图

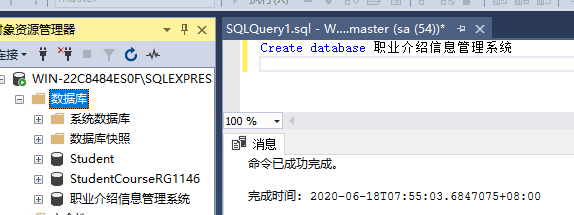
白色的地图

描述已自动生成

根据ER图写出关系模型：

* 职业分类（职业类型号、职业类型名）
* 用人单位（单位编号、单位名称、职业类型号）
* 职业信息（职业号、职业名、单位编号、职业类型号、专业要求、已聘人数、需求人数、工资）
* 求职者信息（求职者编号、姓名、性别、攻读专业、是否聘用、月薪要求）
* 介绍人员（介绍人员编号、姓名、电话）
* 费用管理（求职者编号、用人单位编号、介绍人员编号、是否缴费）
* 匹配（求职者编号、职业号、数量）

创建数据库



创建表

职业分类表

CREATE TABLE [dbo].[职业分类表] (

[职业类型号] INT IDENTITY (1, 1) NOT NULL,

[职业类型名] CHAR (10) NOT NULL,

PRIMARY KEY CLUSTERED ([职业类型号] ASC)

);

用人单位表

CREATE TABLE [dbo].[用人单位表] (

[单位编号] INT IDENTITY (1, 1) NOT NULL,

[单位名] CHAR (10) NOT NULL,

[职业类型号] INT NOT NULL,

PRIMARY KEY CLUSTERED ([单位编号] ASC),

CONSTRAINT [FK\_用人单位表\_职业分类表] FOREIGN KEY ([职业类型号]) REFERENCES [dbo].[职业分类表] ([职业类型号])

);

职业信息表

CREATE TABLE [dbo].[职业信息表] (

[职业号] INT IDENTITY (1, 1) NOT NULL,

[职业名称] char (10) NOT NULL,

[单位编号] INT NOT NULL,

[职业类型号] INT NOT NULL,

[专业要求] NVARCHAR(50) NULL,

[已聘人数] int DEFAULT ((0)) NOT NULL,

[需求人数] int NOT NULL,

[工资] INT NOT NULL,

PRIMARY KEY CLUSTERED ([职业号] ASC),

CONSTRAINT [FK\_职业信息表\_职业分类表] FOREIGN KEY ([职业类型号]) REFERENCES [dbo].[职业分类表] ([职业类型号]),

CONSTRAINT [FK\_职业信息表\_用人单位表] FOREIGN KEY ([单位编号]) REFERENCES [dbo].[用人单位表] ([单位编号])

);

求职者信息表

CREATE TABLE [dbo].[求职者信息表]

(

[求职者编号] INT identity NOT NULL PRIMARY KEY,

姓名 char(10) NOT NULL,

性别 char(10) NOT NULL,

攻读专业 char(10) NOT NULL,

是否聘用 char(10) NOT NULL DEFAULT '无',

月薪要求 int NOT NULL

)

介绍人员表

CREATE TABLE [dbo].[介绍人员表]

(

[介绍人员编号] INT identity NOT NULL PRIMARY KEY,

姓名 char(10) NOT NULL,

电话 char(11) NOT NULL,

CONSTRAINT [CK\_介绍人员表\_电话] CHECK (电话 LIKE '[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9][0-9][0-9][0-9]'

   or 电话 LIKE '[0-9][0-9][0-9][0-9]-[0-9][0-9][0-9][0-9][0-9][0-9][0-9]'

   or 电话 LIKE      '[0-9][0-9][0-9]-[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'

   or 电话 LIKE '[0-9][0-9][0-9][0-9]-[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'

)

)

费用管理表

CREATE TABLE [dbo].[费用管理表]

(

[求职者编号] INT identity NOT NULL PRIMARY KEY,

用人单位编号 int NOT NULL,

介绍人员编号 int NOT NULL,

是否缴费 char(10) NOT NULL DEFAULT '否',

CONSTRAINT [FK\_费用管理表\_介绍人员表] FOREIGN KEY ([介绍人员编号]) REFERENCES [介绍人员表]([介绍人员编号]),

CONSTRAINT [FK\_费用管理表\_用人单位表] FOREIGN KEY ([用人单位编号]) REFERENCES [用人单位表]([单位编号])

)

匹配管理表

CREATE TABLE [dbo].[匹配管理表]

(

求职者编号 int NOT NULL ,

职业号 int NOT NULL ,

数量 int NOT NULL,

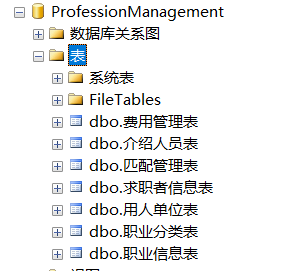
CONSTRAINT [FK\_匹配管理表\_求职者信息表] FOREIGN KEY ([求职者编号]) REFERENCES [求职者信息表]([求职者编号]),

CONSTRAINT [FK\_匹配管理表\_职业信息表] FOREIGN KEY ([职业号]) REFERENCES [职业信息表]([职业号]),

CONSTRAINT [PK\_匹配管理表] PRIMARY KEY ([求职者编号])

)

表创建完之后如下图：



向数据库中写入部分数据



