

大学生求职招聘系统数据库设计

鲍海洋 曾 丽

成都锦城学院计算机与软件学院 四川 成都 611731

【摘要】为了避免人力的浪费，提升面试通过率以及发布更多兼职信息，本求职招聘管理系统力求以更高效的方式完成企业对人才的录用工作，以及帮助大学生更高效的寻找到合适的工作岗位。本文从需求分析、数据库设计、数据库实现几个方面阐述了大学生求职招聘系统的设计过程。

【关键词】 数据库；求职；招聘

1 项目概述

在如今的社会需求下出现了很多新的职业与岗位且每年都会有一批又一批的学生涌入社会求职，但是大多数由于公司招聘人才的信息不全面以及对员工信息了解又不足的情况下很多时候采取的是面试，但这样往往浪费了很多的时间。

本项目的用户是专科生、本科生、各个中小型公司。是针对求职招聘的需求设计的，可以完成求职、招聘、兼职等主要功能。

2 需求分析

本系统包括学生求职管理、个人岗位发布和公司招聘管理。面向的用户分为学生、个人和公司。在以学生的身份去使用时，学生可以查询各种类型的工作信息可以在不同的时间选择不同的兼职，同时可以选择对应的

公司或者个体但是在选择专职类型工作时只能提供学生去应聘和面试的地点，学生只能进行对自己信息的注册、填写、修改等操作达到学生求职的功能。对于公司来说，公司首先要进行注册之后填写完信息之后才可以发布专职或者兼职类型的工作，公司可以看到还有哪些学生没有工作，也可以查看自己发布多少个工作，选择了自己工作的学生的部分信息为了更好的实现对接。但是每个公司只能对自己公司的信息进行修改也可以对自己发布的专职工作信息进行修改。对于个人，个人能进行填写修改信息，只能发布兼职工作信息和修改删除自己的发布的兼职工作信息。

3 数据库规划与设计

3.1 概念模型

首先通过需求分析建立起 E-R 实体联系图如下：

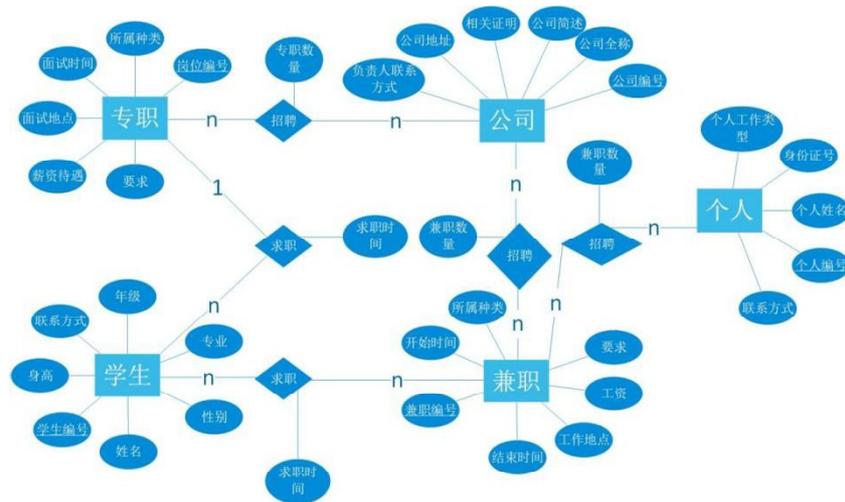


图1 求职系统 ER 图

3.2 关系模型

根据已经建立好的 E-R 模型图表达出每一个关系模式如下：

学生表（学生编号，姓名，性别，身高，联系方式，年级，专业）其中主键是学生编号

专职岗位表（岗位编号，所属种类，面试时间，面试地点，薪资待遇，要求）其中主键是岗位编号

兼职岗位表（兼职编号，所属种类，开始时间，结束时间，工作地点，工资，要求）其中主键是兼职编号

公司招聘表（公司编号，公司全称，公司简述，公司地址，负责人联系方式）其中主键是公司编号

个人招聘表（个人编号，个人姓名，身份证号，个人工作类型，个人联系方式）其中主键是个人编号

学生工作求职表（学生编号，岗位编号，求职时间）主键是学生编号岗位编号，外键是学生编号和岗位编号

学生兼职求职表（学生编号，兼职编号，求职时间）主键是学生编号兼职编号，外键是学生编号和兼职编号

公司专职招聘表（公司编号，岗位编号，专职数量）

主键是公司编号岗位编号，外键是公司编号和岗位编号
 公司兼职招聘表（公司编号，兼职编号，兼职数量）
 主键是公司编号兼职编号，外键是公司编号和兼职编号
 个人兼职招聘表（个人编号，岗位编号，兼职数量）
 主键是个人编号岗位编号，外键是个人编号和岗位编号

4 数据库实现

4.1 表的实现

在通过需求分析以及数据库规划之后进行了对数据库表的实现，先创建了一个叫 WORK 的数据库，在该数据库中一共创建了 10 个表，这些表的作用主要是存储用户的信息以及用户和实体之间的联系 student 表、works 表、timejob 表、companys 表、personals 表主要是用来存储信息的，其余的表都是这些表之间的联系。部分代码展示如下：

```
create table personals(id char(6) primary
key, -- 个人编号
name nvarchar(6) not null, -- 个人姓名
NOID char(17) NOT NULL, -- 身份证号
type nvarchar(20) not null, -- 个人工作类型
telephone char(11) NOT NULL -- 电话)
```

4.2 视图

视图不仅可以简化用户对数据的理解也可以简化他们的操作。那些被经常使用的查询可以被定义为视图从而使用户不必为以后的操作每次指定全部的条件。它大大简化了用户对数据的操作。因为在定义视图时，若视图本身就是一个复杂查询的结果集，这样在每一次执行相同的查询时，不必重新写这些复杂的查询语句，只要一条简单的查询视图语句即可。可见视图向用户隐藏了表与表之间的复杂的连接操作。在通过表的实现之后，我们创建了视图 v_sw_work，视图是面向用户面向学生的，它的好处可以避免用户对数据进行的破坏，所以此视图将面向学生姓名为张三的学生，该学生可以通过此视图找到自己选择的所有专职记录并输出其专职编号和专职类型。也可以通过视图 v_sw 筛选出推销专职类型的学生的全部信息，最后可以通过视图统计一下所有公司发布兼职工作的个数，以上场景部分代码展示如下：

(1) 用户张三用过视图查询到自己参与的所有专职类型以及编号 create view v_sw_work as select d.worksid, d.worktype, e.jobid, e.jobtype from student a join s_w b on a.sid=b.id join s_j c on a.sid=c.id join works d on d.worksid=b.worksid join timejob e on e.jobid=c.jobid where a.sname='张三'；

(2) 公司可以查询选择了推销专职类型的学生信息，这里选择创建了查询推销专职的学生的代码 create view v_sw as select b.* from works a join s_w on s_w.worksid=a.worksid join student b on b.sid=s_w.id where a.worktype='推销'。

4.3 存储过程

数据库创建过程中一定少不了存储过程的实现，存储过程的创建有利于再数量庞大的数据库中更快有效的进行访问，现对其需求功能进行模拟，如今我们要对数据库进行维护或者公司需要扩招需要统计这个数据库中

没有工作的学生的数量。其次学生在选择工作之后又因为工作信息不完整，或者有其他需求需要联系自己选择的工作的负责人，并要求展示出工作负责人的联系电话工作类型。最后学生由于其他原因不想留有个人信息在其数据库中，可以通过编号删除，并且提示。

查询某个人选择的专职工作类型和公司负责人联系电话的存储过程代码展示如下：

```
create proc 查询电话 @b char(6) As declare @x
nvarchar(20) declare @c nvarchar(20) select @x=b.
telephone, @c=a.worktype from c_w join companys
b on c_w.id=b.id join works a on a.worksid=c_
w.worksid join s_w d on d.worksid=a.worksid join
student e on e.sid=d.id where e.sid=@b print @
b+' 选择的专职工作类型为: '+@c+' 负责人联系电话
为: '+ltrim(@x)
```

4.4 触发器

触发器是一种特殊的存储过程，它分为三类 DML 触发器、DDL 触发器和登录触发器，本文使用的是 DML 触发器。它不是被调用触发不是由程序调用，也不是手工启动，而是由事件来触发，当对一个表进行操作（insert, delete, update）时就会激活它执行。而且它一般是建立在表内的所以它有及其便利的优势使得数据库更能高效的进行数据的流动。当学生已经选择了专职类型工作之后，想再选择其他工作时就会提醒学生选择失败，如果公司没有进行完善信息就发布工作信息也会发布失败。

(1) 当学生以及选择了专职且还在职时，不许已有专职的学生选择工作的触发器实现：

```
create trigger 限制学生 on s_w after insert as
declare @a nvarchar(10) declare @b nvarchar(10)
declare @c nvarchar(10) if exists(select b.sid
from s_w a join student b on a.id=b.sid) select @
a=a.id, @b=b.worktype, @c=b.worksid from inserted a
join works b on a.worksid=b.worksid
```

```
where a.id in (select b.sid from s_w a join
student b on a.id=b.sid) print '编号为: '+ltrim(@
a)+' 学生已经选择了编号为: '+@c+' 编号类型为: '+@
b+' 的工作，所以选择工作失败' rollback
```

(2) 不许信息未完善的公司发布工作的触发器实现：

```
create trigger 限制公司 on c_w after insert as
declare @a nvarchar(10) declare @b nvarchar(20)
if exists(select * from companys a where
(a.name is null) or (a.sketch is null) or
(a.address is null) or (a.telephone is null))
select @a=a.id, @b=b.name from inserted a join
companys b on a.id=b.id where a.id in(select a.id
from companys a where (a.name is null) or (a.sketch
is null) or
```

```
(a.address is null) or (a.telephone is null))
print '编号为: '+ltrim(@a)+' 公司名字为: '+ltrim(@
b)+' 该公司因为信息不完整所以不能发布工作信息'
rollback
```

5 测试

5.1 输入测试

1. 向系统输入公司的信息(公司编号, 公司全称, 公司简述, 公司地址, 负责人联系方式) INSERT INTO companys values(1, '阿里巴巴集团控股有限公司', '网购', '中国浙江省杭州市余杭区文一西路969号', 1008611)

INSERT INTO companys values(2, '腾讯', '腾讯以技术丰富互联网用户的生活', '深圳市南山区海天二路33号腾讯滨海大厦', 86013388)

5.2 功能测试

功能测试主要是对该数据库实现的对用户需求的视图存储过程触发器等一些特定语句功能的测试, 测试模拟实际运用中需要用到的语句查验是否成功。

(1) 学生用户通过视图统计每个公司发布兼职信息的个数的测试语句如下: select * from v_cj_number

(2) 当有学生不想继续在本系统找寻工作, 又不想本人信息泄露时, 就使用此语句实现快速清除自己的信息, 测试语句如下: Exec 删除学生 1;

(3) 当学生选择了某个专职工作之后, 想找到自己专职工作的类型信息和工作的负责人联系方式, 就可以通过该指令实现起快速查询, 测试语句如下: exec 查询电话 '1';

(4) 当公司想要发布一些工作岗位时, 可以首先通查询未工作学生的学生数量, 从而发布工作。此时便可

以通过该指令代码实现未选择工作的学生数量测试语句如下: exec 统计未工作学生;

(5) 当有信息不全面的公司发布工作时, 系统会自动触发提醒, 不能发布该工作, 应为公司信息不完整, 只能在信息填写完成之后才能发布工作。以此我们创建一个只有编号的公司, 在由该公司发布工作信息, 测试语句如下: INSERT INTO companys values (3) INSERT INTO c_w values (3, 2, 10);

(6) 当学生已经选择了一个专职工作之后, 就不能再继续选择其他的专职工作, 由此, 测试一个已经选择了专职工作的学生再去选择另一个专职工作, 测试语句如下: INSERT INTO s_w values (1, 2, '2021.6.13')。

6 总结

在大数据横行的时代, 通过人才招聘公司或者中介找寻工作的方法终究会被数据所替代, 而且通过公开透明的数据能极大的减少了找寻工作所浪费的时间, 且如今大学生大多无社会经验, 也为了更多的大学生能通过大学闲余时间去找个兼职, 帮助更好的提升自己。以此, 本文的探究或有增益。

【参考文献】

[1] 何玉洁. 数据库系统教程(第2版), ISBN: 978-7-115-40245-5, 北京: 人民邮电出版社. 2010.

[2] 杨冬青, 李红燕等译. 数据库系统概念(第6版), 北京: 机械工业出版社, 2012.