Database Documentation

Role Models

Interviews of Women in Technology

Loren Paulsen
5/31/2007

Overview

The Role Models website will serve two distinct functions. For the public, it will serve as a repository of interview information. For employees, it will serve as a complete project management system. My involvement in the project involves creating the website and database.

Public Facing Website

The public web site, to be hosted at a yet-to-be-determined domain, will host information about all of the interviews we have taken as well as other online resources. The prototype has been built in PHP, utilizing MySQL as its backend database. The final web site will built in ASP.NET 2.0 with a Microsoft SQL Server database.

Project Management System

Staff will use the internal project management system to update the data on the public website and manage all aspects of interviews in development. Typically, employees will be adding interviews, people, orders, and monitoring the editing process.

Managers have additional functionality available to them. It will be necessary to sum all of the hours worked (multiplied by the appropriate rates) on a particular project, ensuring it is under budget. It will also be necessary to sum all of the hours a particular employee has worked across all of the interviews (multiplied by their rate) where they have not been paid to determine how much they are owed.

ER Diagram

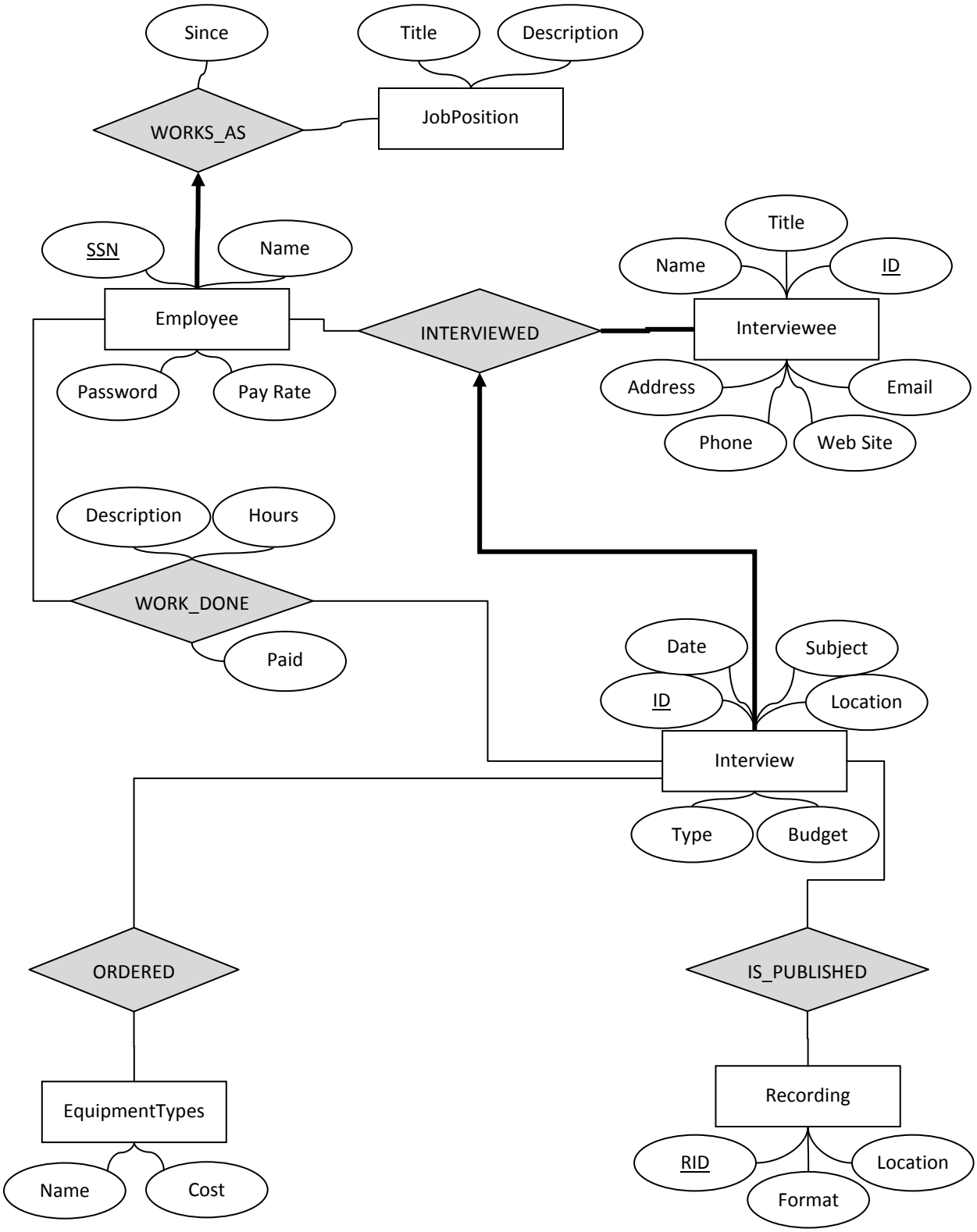
Since
Title
Description
WORKS_AS
JobPosition
SSN
Name
Employee
INTERVIEWED
Title
Name
ID
Interviewee
Password
Pay Rate
Address
Email
Phone
Web Site
Description
Hours
WORK_DONE
Paid
Date
Subject
ID
Location
Interview
Type
Budget
ORDERED
IS_PUBLISHED
EquipmentTypes
Recording
Name
Cost
RID
Location
Format

Table Definitions

```
CREATE TABLE Employee (
        eid VARCHAR(9) NOT NULL,
        first_name VARCHAR(25) NOT NULL,
        last_name VARCHAR(25) NOT NULL,
        password VARCHAR(20) NOT NULL,
        pay_rate DECIMAL(10,2) NOT NULL,
        works_as VARCHAR(20),
        PRIMARY KEY (eid),
        FOREIGN KEY (works_as) REFERENCES
JobPosition (title)
                ON UPDATE CASCADE
);
CREATE TABLE JobPosition (
        title VARCHAR(20) NOT NULL,
        description VARCHAR(50) NOT NULL,
        PRIMARY KEY (title)
);
CREATE TABLE Interviewee (
        id INT NOT NULL AUTO_INCREMENT,
        name VARCHAR(20) NOT NULL,
        title VARCHAR(20),
        address1 VARCHAR(50),
        address2 VARCHAR(50),
        city VARCHAR(50),
        state VARCHAR(50),
        zip VARCHAR(50),
        phone VARCHAR(10),
        website VARCHAR(255),
        email VARCHAR(50),
        PRIMARY KEY (id)
);
CREATE TABLE Interview (
        id INT NOT NULL AUTO_INCREMENT,
        interviewee_id INT NOT NULL,
        date_taken DATE NOT NULL,
        subject VARCHAR(20) NOT NULL,
        location VARCHAR(20) NOT NULL,
        type VARCHAR(10) NOT NULL,
        budget DECIMAL(10,2),
        PRIMARY KEY (id),
        FOREIGN KEY (interviewee_id) REFERENCES
Interviewee (Id)
);
CREATE TABLE Tasks (
        name VARCHAR(25) NOT NULL,
        est_hours INT NOT NULL,
        PRIMARY KEY (name)
);
CREATE TABLE EquipmentType (
        name VARCHAR(25) NOT NULL,
        cost DECIMAL(10,2) NOT NULL,
        PRIMARY KEY(name)
);
CREATE TABLE Recording (
        rid INT NOT NULL AUTO_INCREMENT,
        format VARCHAR(50) NOT NULL,
        location VARCHAR(500) NOT NULL,
        interview_id INT NOT NULL,
        PRIMARY KEY (rid),
        FOREIGN KEY (interview_id) REFERENCES
Interview(id)
);

CREATE TABLE WORK_DONE (
        eid VARCHAR(9) NOT NULL,
        task_name VARCHAR(25) NOT NULL,
        interview_id INT NOT NULL,
        comment VARCHAR(50),
        hours INT NOT NULL,
        paid BOOL,
        PRIMARY KEY (eid, task_name,
interview_id),
        FOREIGN KEY (eid) REFERENCES Employee,
        FOREIGN KEY (task_name) REFERENCES Tasks
(name)
                ON UPDATE CASCADE,
        FOREIGN KEY (interview_id) REFERENCES
Interview(id)
);
CREATE TABLE ORDERED (
        equipment_name VARCHAR(25) NOT NULL,
        interview_id INT NOT NULL,
        PRIMARY KEY (interview_id,
equipment_name),
        FOREIGN KEY (equipment_name) REFERENCES
EquipmentType (name)
                ON UPDATE CASCADE,
        FOREIGN KEY (interview_id) REFERENCES
Interview (id)
);
```