

Overview of database:

candidate – This table is used keep track of the candidates that will be entered into the database.

Columns:

candidate_id – Use an int as a primary key

firstname & surname – Uses varchar (45)

address – Uses varchar (100) as the address is a lot longer than the names.

phone – Uses an int to take the phone number, could have used varchar(10) but though an int would be more appropriate

candidate skills – This table is used to keep track of the skills each candidate has.

Columns:

candidate_id – Uses candidate_id from candidates table as Foreign Key

skills_id – Uses skill_id from skill table as Foreign Key

Both columns are used as a composite key so the to prevent duplicate entries.

Constraints – both foreign keys are set to restrict on Update and Delete. This assumes once a candidate has a skill, they will always have that skill.

hired – This is just an extra table I put in to complete this database. It was made because I assumed that not everyone that was offered a job interview would accept it.

Columns:

employee_id – This is just a generic employee id added when someone is hired

interview_id – This is a foreign to the interview table. This assumes that no one can be hired for 2 jobs based on an interview so the interview column is set to unique. This could also be used to link table to the interview table to get a employees candidate_id.

Constraints: Set to restrict the interview id key as once an interview_id is assigned it should not be changed or deleted so update and delete are set to restricted.

hospitals – used to enter a hospital into the database

Columns:

hospital_id – unique identifier given to each hospital

hospital_name - uses varchar(45)

hospital_address - uses varchar(100)

hospital_phone – uses int because I thought it would be better than varchar

Interviews– used to keep track of which candidates went for what positions and when

Columns:

interview_id – unique id given to each interview

date_of_interview – uses date as input

candidate_id – uses Foreign Key candidate_id from candidate table

hospital_id - uses Foreign Key hospital_id from candidate table

job_offer – uses Varchar (3) so yes or no can be entered

position_advertised – uses Foreign Key from position available table. This references the position available table as opposed to the positions tables because the position should only be in the interview table if the position is vacant.

Constraints: candidate_id and hospital_id are set to restricted on update and delete as we don't want them removed from the other tables. Position_advertised is set to cascade on delete and update as the position can be removed or updated if its filled.

Position– used to keep track of all the jobs in the database

Columns:

Job_id - unique Id given to each job

Position_type – gives name to each position

Positions available– keeps track of the positions that are need to be filled in each hospital

Columns:

positions_id – Foreign Key taken from positions table

hospital_id – Foreign Key taken from hospitals table

number_of_positions – used to track number of each position needed in each hospital

Both id columns are used as a composite key so the to prevent duplicate entries.

Constraints: Both are again set to restrict as I don't want to lose or change a position or hospital as it would affect other tables

Position skills – used to assign a skills to a position

Columns:

position_id – Foreign key taken from position table

position_skills – Foreign key taken from skill table

Both columns are used as a composite key so the to prevent duplicate entries.

Constraints: Both are again set to restrict as I don't want to lose or change a position or hospital as it would affect other tables

Skill table – used to track the skills in the database

Columns:

Skill_id – uses int to assign id to a skill

Skill_name – gives name to a skill in the database

Summary of database:

There are 9 tables which might seem excessive but every is useful and attempts to ensure no information can be duplicated and any table can be linked back to the name of the candidates, the name of the hospital and the name jobs to make information clear. The skills are linked to candidates using a many to many table **candidate_skills**. Skills also link with the position via **position_skills** which is also a many to many relationship. The interview links 3 tables **hospital**, **candidate** and **positions_available** so the entire database can be connected.

Assumptions and Additions:

- I added the number of position available for each job in each hospital in **Positions_available**. That is making the assumption that a hospital needs to keep track of the number of positions for each job are required.
- There is a job offer column in the **interviews** which was added. This was added because I didn't want to put in a new table for job offers and hired. In this database if the interview goes well the candidate may be offered a position but that candidate might have also been offered a job at another hospital and they can only accept one. This is why the hired table was added so candidates could go for multiple interview and get multiple job offers but only accept one, like in the real world.
- Assumes that the data is entered into the table in a logical order (ie, a candidate can't be registered for an interview if they aren't in the system yet). The hospital, candidate, position and skill table must always have their information entered first.

Operating system used: Windows 10

ER Diagram of database

