

# COMP20070 – Databases and Information Systems

## MySQL Assignment

Elvin Jiby

### Description of Database

My database is comprised of seven tables, which are defined as follows:

- *candidate* is a table that contains information about the candidate's personal details
- *candidateskills* is a table that stores information on the skills that a candidate may have
- *position* is a table that contains details on the type of job offered, as well as the hospital in which it is provided at
- *positionskills*, like *candidateskills*, stores information on the skills that a position may require
- *skills* contain information on all the skills that can be referenced by the *candidateskills* and *positionskills* tables
- *hospital* stores basic details of a hospital
- *interview* consists of details of a candidate being interviewed for a particular position

There are seven stored procedures that allow the insertion of a tuple of information into one of the tables chosen by the user. They are in the format ***insert\_X***

Eleven queries are also present, which carry out the queries specified in the assignment specifications. They are in the format ***STEP4\_Query\_x***, which are ordered in the same way as the assignment outline.

Some things to note when inserting new information:

- The identifiers *candidateID*, *hospitalID*, *interviewID* & *positionID* have a varying character length of 30. *skillID*, however, only accepts integers.
- *positionOffered* in the *interview* table is of type tinyint, which only takes a Boolean value or either 1 or 0.
- *interviewDate* in the *interview* table takes a date as input
- Any other constraints have a varying character length of 30.

## Assumptions/Additions

- I assumed a many-to-many relationship between *skills* and *candidate* (connected by the *candidateskills* table), *skills* and *position* (connected by the *positionskills* table) & *position* and *candidate* (connected by the *interview* table)
- I assumed that candidates could possess multiple skills, so I created the *candidateskills* table for that purpose. It is useful for managing the skill set of each candidate without too much hassle
- Like *candidateskills*, I also assumed that a position can require multiple skills. This is found in the *positionskills* table, where a position can require one, many or even no skills
- I presumed that one candidate can be interviewed more than once for a particular position
- A hospital can have lots of interviews for a position
- A hospital can hire many candidates for a position
- The *interview* table also contains information on the date of the interview, as well as a Boolean value that indicates whether a candidate received the job or not (1 = offered, 0 = not offered)

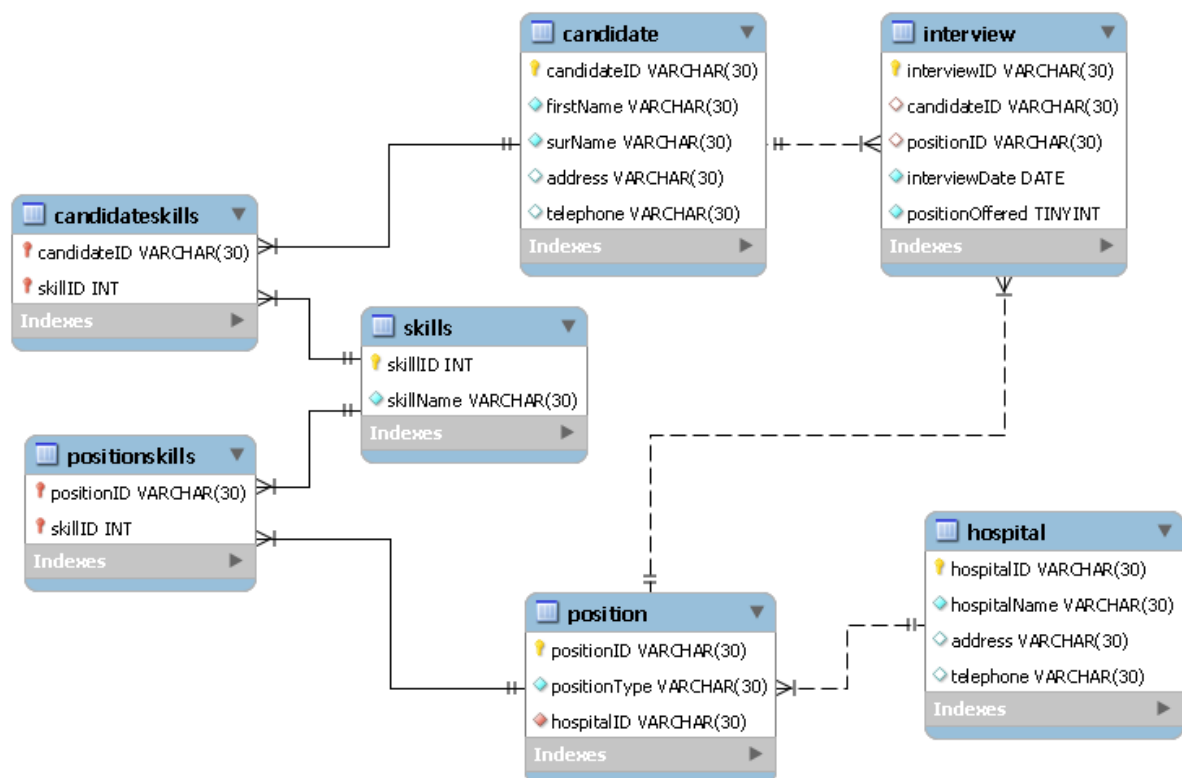
## Reaction Policies Used

- *candidateID* and *positionID* in the *interview* table have the **ON DELETE SET NULL** policy. I chose this since I assume most businesses would prefer to keep records of their past interviews, even if the candidate or position is no longer associated with that hospital. This does mean that an interview can have an unknown candidate and position associated with it.
- *skillID* in the *candidateskills* and *positionskills* table have the **ON DELETE RESTRICT** policy, to ensure that skills associated with candidates and positions are not lost. For example, if the nursing skill was deleted, that does not mean that any candidate with a nursing skill should just lose that skill for no reason
- *candidateID* and *positionID*, in the *candidateskills* and *positionskills* tables respectively, have a **ON DELETE CASCADE** policy because if a candidate or position is deleted, there is no reason to keep information

of their skill set. The same policy is applied to *hospitalID* in the *position* table

- Apart from that, the only other foreign policy used is **ON UPDATE CASCADE**, which is handy in making global changes to an attribute of a table. Taking an example, if a name of a candidate was changed, the cascade policy can help make the same changes to any other table that contains that same candidate to keep everything up-to-date.

## Entity-Relationship Diagram



## Operating System Used

Windows 11