Comp20240 (assignment 1)

Name (student id): Cian Fitzgerald (20206773)
Operating system used: Mac

A short clear description of your database.

My database is comprised of 6 tables in total, these tables are as follows:

- Hospital details (id, name, address, telephone)
- Position details (id, type, hospital id)
- Candidate details (id, first name, last name, address, telephone)
- Interview details (id, candidate id, position id, date, offered)
- Candidate skills – skills possessed by the candidate (candidate id, skills)
- Required skills – skills required for the position (position id, skills)

The first four tables listed are the main tables described in step two of the assignment, the last two tables listed were created in order to map the relationship of candidates and positions to their corresponding skills. Due to the fact that each candidate can have many skills and each position can require many skills, the creation of these tables was necessary in order to create the one-to-many relationships associated with these elements of the database. There are six one to many relationships between the tables, these are between:

- Hospital id of the hospital details table (parent) and of the position details table (child).
- The position id of the position details table (parent) and of the interview details table (child).
- The candidate id of the candidate details table (parent) and the interview details table (child).
- The position id of the position details table and the required skills table (required_skills).
- The candidate id of the candidate details table and the candidate skills table (has_skills)

There is a total of 17 stored procedures. Six of which are to add new rows to each of the tables, when using these procedures, it must be taken into consideration for the candidate skills and required skills that a new child row cannot be created due to foreign key restrictions, and therefore only new rows can be added for already existing candidates in the parent row. This is similar for the new-row stored procedures for Position Details and Interview Details, a new candidate or position cannot be added to these tables until they have been added to the parent tables due to the foreign key constraint. Therefore, for these two tables, rows can only be added for already existing candidates or hospitals in the database. The rest of the stored procedures are a combination of queries and parametric queries as listed in the details of this assignment.

Assumptions:

- That each interview had an id number, this was used as the primary key of this table.
- That the candidates were either offered the job or not, therefore this data entry was of type VARCHAR (20), with entry of yes or no.
- It was assumed that interviews only occurred on particular dates for each candidate.
- For stored procedure 8 "Find the positions sorted according to the hospitals who are advertising them." It was assumed that this meant to sorted them in alphabetical order.
- For stored procedure 10, it was assumed that the 'specified date' was to be given by the user through a parametric query.

Additions:

- The main additions made were the tables labelled has_skills and required_skills. these tables were used to link the one to many relationship of the candidates and the skills they possessed, and the positions and the skills required. The has_skills table had two columns (candidate id, skills), foreign keys were then created to link these entries back to the parent records. The required_skills table had two columns (position id, skills), foreign keys were then created to link these entries back to the parent records.
- Three other foreign keys were added to link the one to many relationships which I have listed above.
- Within the interview details table a date column of datatype DATE that displayed on which date the interview took place
- Within the interview details table a position offered column was added with Data entries 'yes' or 'no'.
- For every table, a stored procedure was created including a parametric query that allows you to insert a new row in to that table.

Reaction policies:

- Hospital id (hospital details and position details):
  on update – cascade to ensure that if a hospital id is changed then the id in the position details will change also.
  on delete – restrict to ensure that all hospital details are kept on record.

- position id (position details and interview details):
  on update – cascade to ensure that if a position id is changed then the id in the interview details will too.
  on delete – restrict, candidates may apply to other positions in the future so it is important to keep their records.

- candidate id (candidate details and interview details):

on update – cascade to ensure that if a candidate id is changed then the id in the interview details will too.
on delete – restrict for similar reasons to above, it is good to keep these details on record as they may apply again in the future.

- candidate id (candidate details and has_skills):
  on update – cascade to ensure that if a candidate id is changed then the id in the has_skills table will to.
  on delete – restrict, as we want to keep all candidate id records.

- position id (position details and required_skills):
  on update – cascade to ensure that if a candidate id is changed then the id in the position details will to
  on delete – restrict as it is important to keep on file who has interviewed for positions and whether the position was offered etc. Even if the position is filled, we still want to keep these records on file for future reference.

Entity Relationship Diagram: