COMP20070 Final Assignment – Applicant Tracking System

Samuel Dornan – 20385853

Database represents the following scenario: “Hospitals advertise positions which require specific skills (e.g., nursing, administrative, etc.). Candidates may be invited to interviews for the positions”.

Applicant Tracking System for a hospital or health service, allowing applicants to apply for jobs, determine which jobs they may be a good fit for based on their skills, and allowing hiring managers to see which jobs an applicant may be a good fit for based on their skills provided.

This system relies heavily on the use of ID numbers for indexing to increase the likelihood of a user finding the results they are looking for, and assumes the user has some familiarity with MySQL.

Key constraints defined in step 2 are primarily handled through the interviews table, or else are handled through link tables such as “position_skills” and “candidate_skills”/

Developed on Windows 11 Pro

Data Lineage

Table Name	Column Name	Type	Description
`candidate_details`	`candidate_address`	TEXT	Free text to allow users to input longer addresses
`candidate_details`	`candidate_firstname`	TEXT	Free text to allow for different symbols used in different names
`candidate_details`	`candidate_id`	INT-PRIMARY KEY	Primary key to allow consistent indexing
`candidate_details`	`candidate_surname`	TEXT	Free text to allow for different symbols used in different names
`candidate_details`	`candidate_telephone`	INT	Integer to assure that telephone numbers remain as telephone numbers, increasing the likelihood that they are useful
`candidate_skills`	`candidate_details-id`	INT-PRIMARY KEY	Primary key for consistent indexing and relational integrity
`candidate_skills`	`skills_details-id`	INT-PRIMARY KEY	Primary key for consistent indexing and relational integrity
`hospital_details`	`hospital_address`	TEXT	Free text to allow users to input longer addresses

`hospital_details`	`hospital_id`	INT	Primary key for consistent indexing and relational integrity
`hospital_details`	`hospital_name`	TEXT	Free text as hospital names may be more than one word or contain symbols such as apostrophes
`hospital_details`	`hospital_telephone`	INT	Integer to assure that telephone numbers remain as telephone numbers, increasing the likelihood that they are useful
`interview_details`	`candidate-hired`	TINYINT	Tinyint used as MySQL treats these as Boolean operators; 1 = True, 0 = False
`interview_details`	`interview-candidate_id`	INT	Used to create a relationship between the candidate details and interview table
`interview_details`	`interview-position_id`	INT	Used to create a relationship between the position details and the interview table
`interview_details`	`interview_date`	DATE	ISO 8601 Format
`interview_details`	`interview_id`	INT-PRIMARY KEY	Primary key for consistent indexing and relational integrity
`position_details`	`hospital_details-id`	INT	Used to create a relationship between the position details and the hospital details table
`position_details`	`position_id`	INT-PRIMARY KEY	Primary key for consistent indexing and relational integrity
`position_details`	`position_name`	TEXT	Used to allow position name to be entered in plain language
`position_details`	`position_type`	TEXT	Used to allow position type to be entered in plain language
`position_skills`	`position_details-id`	INT-PRIMARY KEY	Used to create a relationship between the position details table and skills table
`position_skills`	`skills_details-id`	INT-PRIMARY KEY	Used to create a relationship between the position details table and skills table
`skills_details`	`skill_id`	INT-PRIMARY KEY	Primary key for consistent indexing and relational integrity
`skills_details`	`skill_name`	TEXT	Used to allow skills to be entered in plain language

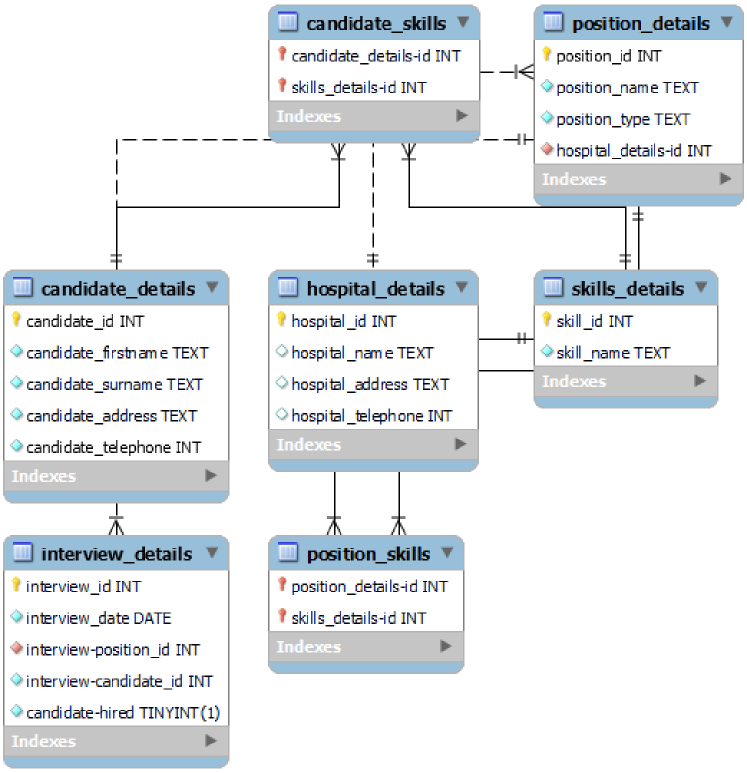
Assumptions & additions

- Addition of link tables for the creation of many-many relationships
- Task 4.6: Requires to search for position ID rather than position name
- Task 4.8: This doesn't explicitly say which positions need to be found; assumed all positions would need to be found for a given hospital ID
- Task 4.9: Dates are in ISO 8601 format
- Task 4.10: Returns unique candidates who were interviewed on a specific date. Note: does not account for candidates who may have been interviewed on dates other than the one selected; does not show if a candidate was interviewed more than once on the selected date. (This is because it was assumed a hiring manager who wanted to know which candidates were interviewed more than once on a specific date would ask for that dammit.)

Reaction policies

- Heavy reliance on indexing by ID values to ensure that valid results are returned after queries
- Use of automatically generated integers for ID values where possible to reduce the risk of users forgetting to input these or else attempting to input values that already exist.
- Constraining connections by ID to ensure that referential integrity is maintained

ER Diagram


candidate_skills
candidate_details-id INT
skills_details-id INT
Indexes
position_details
position_id INT
position_name TEXT
position_type TEXT
hospital_details-id INT
Indexes
candidate_details
candidate_id INT
candidate_firstname TEXT
candidate_surname TEXT
candidate_address TEXT
candidate_telephone INT
Indexes
hospital_details
hospital_id INT
hospital_name TEXT
hospital_address TEXT
hospital_telephone INT
Indexes
skills_details
skill_id INT
skill_name TEXT
Indexes
interview_details
interview_id INT
interview_date DATE
interview-position_id INT
interview-candidate_id INT
candidate-hired TINYINT(1)
Indexes
position_skills
position_details-id INT
skills_details-id INT
Indexes