**Job Seeker Applications View:**

/\*

The jobSeekerApplications view displays a summary of all the job applications for a given jobSeeker based off the jobSeekerId, in this case I have used 5.

The view displays all relevant information for the jobSeeker to see such as their name, application id, application status, job title, job description and employer.

\*/

CREATE VIEW jobSeekerApplicationsView AS SELECT

jobSeekers.jobSeekerId,

jobSeekers.forename AS `Forename`,

jobSeekers.surname AS `Surname`,

applications.applicationId AS `Application ID`,

applicationStatus.status AS `Application Status`,

jobListings.title AS `Job Title`,

jobListings.description AS `Job Description`,

employers.name AS `Employer`

FROM

applications

INNER JOIN jobSeekers ON applications.jobSeekerId = jobSeekers.jobSeekerId

INNER JOIN applicationStatus ON applicationStatus.statusId = applications.statusId

INNER JOIN jobListings ON jobListings.jobId = applications.jobId

INNER JOIN employers ON employers.employerId = jobListings.employerId

WHERE

jobSeekers.jobSeekerId = 11;

Output

A screenshot of a computer

Description automatically generated

**Upcoming Interviews View:**

/\*

The upcomingInterviews view displays the relevant details of each jobSeeker who has an upcoming interview. It also displays the employer and job they are being interviewed for along with a field to mark the interview as complete. This will allow a business to get the information of an upcomng interviewee.

\*/

CREATE VIEW upcomingInterviews AS SELECT

js.jobSeekerId AS `Job Seeker ID`,

js.forename AS `Forename`,

js.surname AS `Surname`,

js.dob AS `DOB`,

js.email AS `Email`,

js.phoneNo AS `Phone Number`,

inter.date AS `Interview Date`,

ap.applicationId AS `Application ID`,

jl.title AS `Job Title`,

e.name AS `Employer`,

inter.complete AS `Mark as Complete`

FROM

jobSeekers js

INNER JOIN applications ap ON ap.jobSeekerId = js.jobSeekerId

INNER JOIN applicationStatus aps ON aps.statusId = ap.statusId

INNER JOIN interviews inter ON inter.applicationId = ap.applicationId

INNER JOIN jobListings j ON jl.jobId = ap.jobId

INNER JOIN employers e ON e.employerId = jl.employerId

WHERE

aps.statusId = 5 AND inter.complete = 0

ORDER BY inter.date;

A screenshot of a computer

Description automatically generatedOutput

**Completed Interviews View:**

/\*

The completedInterviews view displays all interviews that have been completed, allowing the user to add an interview score, update the application status, add comments, strengths, and weaknesses. This will allow businesses to easily update an application to inform applicants of their outcome, and how they can do better.

\*/

CREATE VIEW completedInterviews AS SELECT

js.jobSeekerId AS `Job Seeker ID`,

js.forename AS `Forename`,

js.surname AS `Surname`,

js.dob AS `DOB`,

js.email AS `Email`,

js.phoneNo AS `Phone Number`,

inter.date AS `Interview Date`,

jl.title AS `Job Title`,

e.name AS `Employer`,

inter.score AS `Score`,

ap.statusId AS `Update Status`,

inter.comments AS `Comments`,

inter.improvements AS `Improvements`,

inter.strengths AS `Strengths`

FROM

jobSeekers js

INNER JOIN applications ap ON ap.jobSeekerId = js.jobSeekerId

INNER JOIN applicationStatus aps ON aps.statusId = ap.statusId

INNER JOIN interviews inter ON inter.applicationId = ap.applicationId

INNER JOIN jobListings jl ON jl.jobId = ap.jobId

INNER JOIN employers e ON e.employerId = jl.employerId

WHERE

inter.complete = 1

ORDER BY inter.date DESC;

Output**A screenshot of a computer

Description automatically generated**

**Employer Hire and Rejection Stats View:**

/\*

The employerHireRejectionStats view displays statistics such as total interviews, total hired, total rejected, hire rate and rejection rate. This helps businesses to see how many people they have rejected compared to hires, allowing them to assess the performance of their recruiters.

\*/

CREATE VIEW employerHireRejectionStats AS SELECT

e.employerId AS `Employer ID`,

e.name AS Employer,

COUNT(DISTINCT inter.interviewId) AS `Total Completed Interviews`,

SUM(CASE WHEN ci.`Update Status` = 6 THEN 1 ELSE 0 END) AS `Total Hired`,

SUM(CASE WHEN ci.`Update Status` = 7 THEN 1 ELSE 0 END) AS `Total Rejected`,

ROUND((SUM(CASE WHEN `ci`.`Update Status` = 6 THEN 1 ELSE 0 END) / COUNT(DISTINCT inter.interviewId)) \* 100, 1) AS `Hiring Rate`,

ROUND((SUM(CASE WHEN `ci`.`Update Status` = 7 THEN 1 ELSE 0 END) / COUNT(DISTINCT inter.interviewId)) \* 100, 1) AS `Rejection Rate`

FROM

employers e

INNER JOIN jobListings jl ON jl.employerId = e.employerId

INNER JOIN applications ap ON ap.jobId = jl.jobId

INNER JOIN jobSeekers js ON js.jobSeekerId = ap.jobSeekerId

LEFT JOIN completedInterviews ci ON `ci`.`Job Seeker ID` = ap.jobSeekerId

LEFT JOIN interviews inter ON inter.applicationId = ap.applicationId

WHERE

ap.statusId IN (6, 7) AND inter.complete = 1

GROUP BY

e.employerId, e.name;

Output

A screenshot of a computer

Description automatically generated

**Jobs Closing Next Week View:**

/\*

This view lists all the job listings that have a closing date in the next week. This helps jobSeekers to apply for jobs before it is too late

\*/

CREATE VIEW jobsClosingNextWeek AS SELECT

\*

FROM

jobListings

WHERE

closingDate <= DATE\_ADD(CURDATE(), INTERVAL 7 DAY) AND closingDate >= CURDATE()

Output

A screenshot of a computer

Description automatically generated