**View 1 – Jobs suitable for a specific applicant.**

/\* Applicants should be able to find suitable employment efficiently. Thus, this view allows applicants to search for opportunities in their preferred sector that fit their salary and time requirements \*/

CREATE VIEW jobs\_in\_trades AS

SELECT

    job.job\_id,

    job.job\_title,

    sector.sector\_title,

    employer.company\_name,

    employer.email\_address,

    job.hourly\_salary,

    job.on\_site,

    job.city,

    job.country,

    job.date\_posted,

    job.application\_deadline

FROM job

INNER JOIN employer ON employer.employer\_id = job.employer\_id

INNER JOIN sector ON sector.sector\_id = job.sector\_id

#Hard-coded values refer to customer with ID 61, would be changed in real-time depending on who was logged in and viewing jobs

WHERE job.hourly\_salary >= '12' AND job.full\_time = 1 AND job.sector\_id = '4' AND application\_deadline < CURRENT\_DATE() + 1

ORDER BY job.job\_id;

A screenshot of a phone

Description automatically generated

**View 2 – Job opportunities closing within the next week.**

/\* Job seekers may need to find a job within a specific timeframe or with some urgency. Viewing jobs with soon approaching deadlines can result in quicker response times, aiding in efficient job searching by prioritising applications that will be more quickly processed \*/

CREATE VIEW jobs\_closing\_this\_week AS

SELECT

    job.job\_id,

    job.job\_title,

    sector.sector\_title,

    employer.company\_name,

    employer.email\_address,

    job.hourly\_salary,

    job.on\_site, job.city,

    job.country,

    job.date\_posted,

    job.application\_deadline

FROM job

INNER JOIN employer ON employer.employer\_id = job.employer\_id

INNER JOIN sector ON sector.sector\_id = job.sector\_id

WHERE job.application\_deadline BETWEEN CURRENT\_DATE() AND CURRENT\_DATE() + 7

ORDER BY job.job\_id;

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated



**View 3 – All applications for a specific job.**

/\* Employers should be able to hire the most qualified candidate for a role. Thus, this view displays all applicants for a vacant position and their qualifications to allow an employer to select the most desired candidate \*/

CREATE VIEW job\_applications AS

SELECT

    job\_seeker.job\_seeker\_id,

    job\_seeker.first\_name,

    job\_seeker.last\_name,

    job\_seeker.email\_address, job\_seeker.phone\_num,

    #Group all qualifications into a single field for aesthetic purposes

    GROUP\_CONCAT(certificate.qualification\_title) AS certificates\_in,

    application.date\_submitted,

    application\_status.status

FROM application

INNER JOIN job\_seeker ON job\_seeker.job\_seeker\_id = application.job\_seeker\_id

LEFT JOIN application\_status ON application\_status.application\_status\_id = application.application\_status\_id

LEFT JOIN certificate ON certificate.job\_seeker\_id = application.job\_seeker\_id

#56 is an example id for a job, would change depending on which job is viewed

WHERE application.job\_id = '56'

GROUP BY job\_seeker.job\_seeker\_id

ORDER BY job\_seeker.job\_seeker\_id;

A screenshot of a computer

Description automatically generated

**View 4 – Insights about wages and job demand within a specific sector.**

/\* This view summarises how well paid and in demand jobs in each sector are, which can influence and speed up job searching. For employers, this can be used to determine how much competition is present and in which sectors advertising should be prioritised \*/

CREATE VIEW sector\_statistics AS

SELECT

    sector.sector\_title,

    ROUND(AVG(job.hourly\_salary), 2) AS average\_salary,

    COUNT(application.job\_id) AS applications\_made,

    COUNT(\*) AS number\_of\_jobs,

    #IF statement ensures dividing by 0 is avoided

    IF(

        COUNT(\*)=0, 0, ROUND(COUNT(application.job\_id)/COUNT(job.sector\_id), 2)

    ) AS applications\_per\_job

FROM job

INNER JOIN sector ON sector.sector\_id = job.sector\_id

LEFT JOIN application ON application.job\_id = job.job\_id

GROUP BY sector.sector\_title

ORDER BY average\_salary DESC;

**A screenshot of a computer

Description automatically generated**

**Python – Connecting to View 2**

Connection and authentication

A screen shot of a computer

Description automatically generated

Retrieving records and formatting them nicely

A computer screen shot of a program

Description automatically generated

A computer screen with text

Description automatically generated

Output

A black screen with white text

Description automatically generated

A screen shot of a computer

Description automatically generated

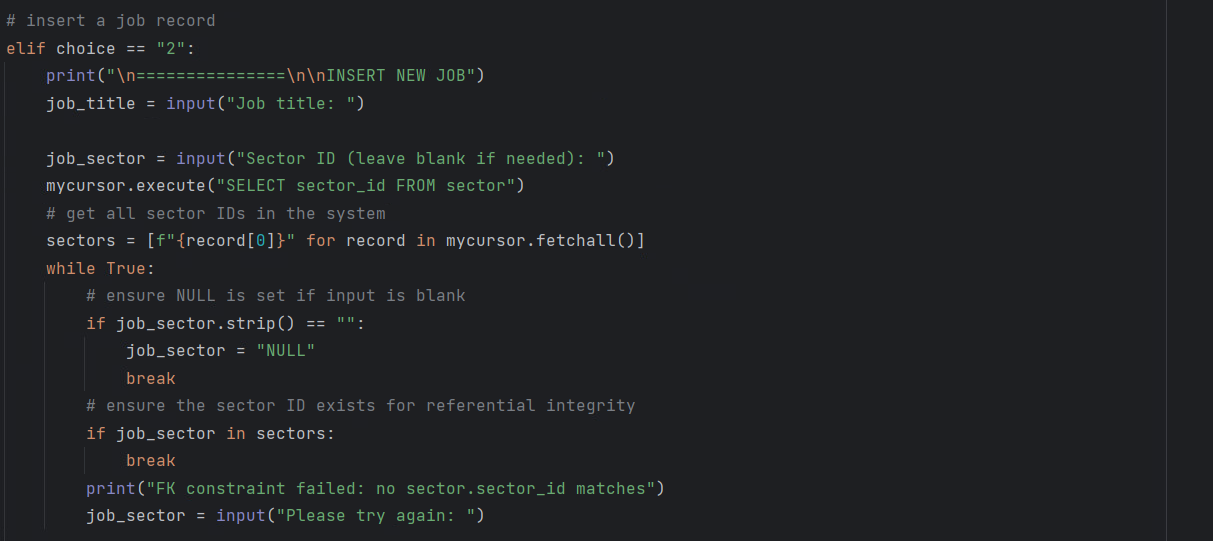
**Python – Inserting a new job record**

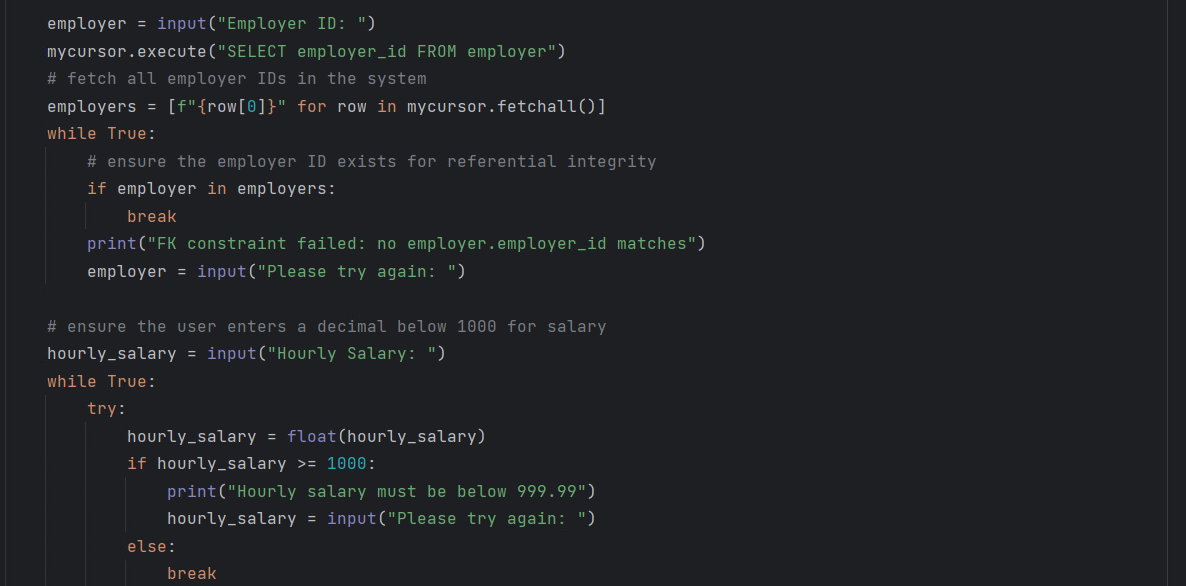
Connection and authentication

A screen shot of a computer

Description automatically generated

Data entry and validation





A computer screen with green and white text

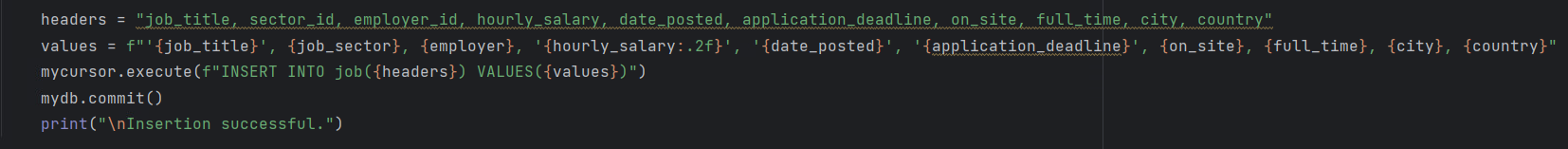
Description automatically generated

A computer screen shot of a program code

Description automatically generated



Formatting data and inserting into the job table



Job table before insertion

A screenshot of a computer

Description automatically generated

Input

A computer screen shot of a black background with white text

Description automatically generated

Output



Job table after insertion

**A screenshot of a computer

Description automatically generated**