Searching for Documents:

Assumptions:

- User’s username is stored in variable **user**
- Optionally, some user-defined document type is stored in variable **type**
- Optionally, user specific a specific file name in variable **filename**
- Optionally, user defines name of application in variable **applicationName**

- Select all documents for applicant

```
SELECT      DocFileName, DocType, Description
FROM        DOCUMENT
WHERE       ApplicantUsername = user;
```

- Select all documents of a certain type

```
SELECT      DocFileName, DocType, Description
FROM        DOCUMENT
WHERE       ApplicantUsername = user
AND         DocType = type;
```

- Select document with a specific file name

```
SELECT      DocFileName, DocType, Description
FROM        DOCUMENT
WHERE       ApplicantUsername = user
AND         DocFileName = filename;
```

- Select documents submitted with a specific application

```
SELECT      D.DocFileName, D.DocType, D.Description
FROM        DOCUMENT AS D
WHERE       D.ApplicantUsername = user
AND         D.DocFileName IN    (SELECT     S.DocFile
                                 FROM       SUBMIT_WITH AS S
                                 WHERE      S.ApplicationName = applicationName
                                 AND        S.ApplicantUsername = user);
```

—

Searching for Applications:

Assumptions:

- User’s username is stored in variable **user**
- Optionally, user defines specific date to look for applications in variable **date**
- Optionally, user defines date that they want to use as upper limit on submissions, in variable **beforeDate**
- Optionally, user defines date that they want to use as lower limit on submissions, in variable **afterDate**
- Optionally, user defines specific status of applications, in variable **status**

- Optionally, user defines specific category of applications, in variable **category**
- Optionally, user defines specific application name, in variable **applicationName**
- Optionally, user defines specific company, in variable **company**
- Optionally, user defines specific job position, in variable **position**
- Optionally, user defines specific date and time of an interview, in variable **interviewDatetime**
- Optionally, user specific a specific file name in variable **filename**

- Select all applications for applicant, sort by status

```
SELECT      AName, Notes, DateSubmitted, Status
FROM        APPLICATION
WHERE       ApplicantUsername = user
ORDER BY    Status;
```

- Select all applications for applicant, sort by date submitted

```
SELECT      AName, Notes, DateSubmitted, Status
FROM        APPLICATION
WHERE       ApplicantUsername = user
ORDER BY    DateSubmitted;
```

- Select all applications for applicant, sort by days since submission

```
SELECT      AName, Notes, DATEDIFF(CURDATE(), DateSubmitted), Status
FROM        APPLICATION
WHERE       ApplicantUsername = user
ORDER BY    DATEDIFF(CURDATE(), DateSubmitted);
```

- Select applications with a specific status

```
SELECT      AName, Notes, DateSubmitted, Status
FROM        APPLICATION
WHERE       ApplicantUsername = user
AND         Status = status;
```

- Select all applications submitted on a specific date

```
SELECT      AName, Notes, DateSubmitted, Status
FROM        APPLICATION
WHERE       ApplicantUsername = user
AND         DateSubmitted = date;
```

- Select all applications submitted before a specific date

```
SELECT      AName, Notes, DateSubmitted, Status
FROM        APPLICATION
WHERE       ApplicantUsername = user
AND         DateSubmitted <= beforeDate;
```

- Select all applications submitted after a specific date

```
SELECT      AName, Notes, DateSubmitted, Status
FROM        APPLICATION
WHERE       ApplicantUsername = user
AND         DateSubmitted >= afterDate;
```

- Select all applications submitted within some date range

```
SELECT      AName, Notes, DateSubmitted, Status
FROM        APPLICATION
WHERE       ApplicantUsername = user
AND         DateSubmitted BETWEEN beforeDate AND afterDate;
```

- Select all applications from a specific category

```
SELECT      A.AName, A.Notes, A.DateSubmitted, A.Status
FROM        APPLICATION AS A, APPL_CATEGORY AS C
WHERE       A.ApplicantUsername = user
AND         C.ApplicantUsername = A.ApplicantUsername
AND         A.AName = C.AName
AND         C.Category = category;
```

- Select all info for a specific application

```
SELECT      AName, Notes, DateSubmitted, Status
FROM        APPLICATION
WHERE       ApplicantUsername = user
AND         AName = applicationName;
```

- Select all URLs for a specific application

```
SELECT      RelevantURL
FROM        APPL_RELEVANT_URL
WHERE       ApplicantUsername = user
AND         AName = applicationName;
```

- Select all categories for a specific application

```
SELECT      Category
FROM        APPL_CATEGORY
WHERE       ApplicantUsername = user
AND         AName = applicationName;
```

- Find total number of applications for a user

```
SELECT      COUNT(AName)
FROM        FROM APPLICATION
WHERE       ApplicantUsername = user;
```

- Select application corresponding to a specific job

```
SELECT      A.AName, A.Notes, A.DateSubmitted, A.Status, C.JobPostURL
FROM        APPLICATION AS A
WHERE       A.ApplicantUsername = user
AND         A.AName IN          (SELECT     C.ApplicationName
                                 FROM       CORRESPONDS_TO AS C
                                 WHERE      C.ApplicantUsername = user
                                 AND        C.CompName = company
                                 AND        C.PosName = position);
```

- Select application related to a specific interview, identifying interview by specific date and time

```
SELECT      A.AName, A.Notes, A.DateSubmitted, A.Status
FROM        APPLICATION AS A
WHERE       A.ApplicantUsername = user
AND         A.AName IN          (SELECT     W.ApplicationName
                                 FROM       INTERVIEW AS W
                                 WHERE      W.ApplicantUsername = user
                                 AND        W.DateTime = interviewDatetime);
```

- Select all applications that include a particular document

```
SELECT      A.AName, A.Notes, A.DateSubmitted, A.Status
FROM        APPLICATION AS A
WHERE       A.ApplicantUsername = user
AND         A.AName IN          (SELECT     S.ApplicationName
                                 FROM       SUBMIT_WITH AS S
                                 WHERE      S.ApplicantUsername = user
                                 AND        S.DocFile = filename);
```

—

<u>Searching for Job Positions:</u>

Assumptions:

- User's username is stored in variable **user**
- Optionally, user specific a specific salary in variable **salary**
- Optionally, user defines specific company, in variable **company**
- Optionally, user defines specific job position, in variable **position**

- Select all jobs that an applicant has applied to, sort by position type

```
SELECT      J.*
FROM        JOB AS J, CORRESPONDS_TO AS C
WHERE       C.ApplicantUsername = user
AND         C.CompName = J.CompName
AND         C.PosName = J.PositionName
ORDER BY    J.PositionType;
```

- Select all jobs that an applicant has applied to, sort by salary highest to lowest

```
SELECT      J.*
FROM        JOB AS J, CORRESPONDS_TO AS C
WHERE       C.ApplicantUsername = user
AND         C.CompName = J.CompName
AND         C.PosName = J.PositionName
ORDER BY    J.Salary DESC;
```

- Select all jobs that an applicant has applied to with a salary above some minimum, sort by salary highest to lowest

```
SELECT      J.*
FROM        JOB AS J, CORRESPONDS_TO AS C
WHERE       J.Salary >= salary
AND         C.ApplicantUsername = user
AND         C.CompName = J.CompName
AND         C.PosName = J.PositionName
ORDER BY    J.Salary DESC;
```

- Show all jobs that an applicant has applied to from a specific company, sort by salary highest to lowest

```
SELECT      J.*
FROM        JOB AS J, CORRESPONDS_TO AS C
WHERE       C.CompName = company
AND         C.ApplicantUsername = user
AND         C.CompName = J.CompName
AND         C.PosName = J.PositionName
ORDER BY    J.Salary DESC;
```

- Show all jobs for which an applicant has received an offer

```
SELECT      O.ResponseDeadline, J.*
FROM        OFFER AS O, JOB AS J
WHERE       O.ApplicantUsername = user
AND         O.CompName = J.CompName
AND         O.PosName = J.PositionName
ORDER BY    O.ResponseDeadline;
```

- Show all jobs for which an applicant has received a referral, and from who

```
SELECT      C.Fname, C.Lname, J.*
FROM        CONTACT AS C, REFERRAL AS R, JOB AS J
WHERE       R.ApplicantUsername = user
AND         R.ContID = C.ContactID
AND         R.CompName = J.CompName
AND         R.PosName = J.PositionName
```

```
ORDER BY    C.Lname, C.Fname;
```

- Show all jobs for which an applicant has an interview, and when

```
SELECT      W.DateTime, J.*
FROM        JOB AS J, INTERVIEW AS W, MENTIONS AS M
WHERE       M.ApplicantUsername = user
AND         M.ApplicantUsername = W.ApplicantUsername
AND         M.InterID = W.InterviewID
AND         M.CompName = J.CompName
AND         M.PosName = J.PositionName
ORDER BY    W.DateTime;
```

- Show all jobs that an applicant tracks, ordered by DateToApply then by ApplicationDeadline

```
SELECT      T.DateToApply, T.Notes, J.*
FROM        TRACKS AS T, JOB AS J
WHERE       T.ApplicantUsername = user
AND         T.CompName = J.CompName
AND         T.PosName = J.PositionName
ORDER BY    T.DateToApply, J.ApplicationDeadline;
```

- Show all jobs that an applicant tracks, ordered by ApplicationDeadline

```
SELECT      T.DateToApply, T.Notes, J.*
FROM        TRACKS AS T, JOB AS J
WHERE       T.ApplicantUsername = user
AND         T.CompName = J.CompName
AND         T.PosName = J.PositionName
ORDER BY    J.ApplicationDeadline;
```

- Show all jobs that an applicant tracks from a specific company, ordered by DateToApply then by ApplicationDeadline

```
SELECT      T.DateToApply, T.Notes, J.*
FROM        TRACKS AS T, JOB AS J
WHERE       T.ApplicantUsername = user
AND         T.CompName = company
AND         T.CompName = J.CompName
AND         T.PosName = J.PositionName
ORDER BY    T.DateToApply, J.ApplicationDeadline;
```

- Show all info for a specific job

```
SELECT      J.Description, J.JobPostFile, J.PositionType, J.ApplicationDeadline, J.Salary
FROM        JOB AS J, CORRESPONDS_TO AS C
WHERE       J.CompName = company
AND         J.PositionName = position
AND         C.ApplicantUsername = user
AND         C.CompName = J.CompName
AND         C.PosName = J.PositionName;
```

- Show all qualifications for a specific job

```
SELECT      Qualification
FROM        JOB_QUALIFICATION
WHERE       CompanyName = company
AND         PositionName = position;
```

- Show all responsibilities for a specific job

```
SELECT      Responsibility
FROM        JOB_RESPONSIBILITY
WHERE       CompanyName = company
AND         PositionName = position;
```

—

<u>Searching for Interviews:</u>

Assumptions:

- User’s username is stored in variable **user**
- Optionally, user defines specific company, in variable **company**
- Optionally, user defines specific job position, in variable **position**
- Optionally, user defines specific date and time of an interview, in variable **interviewDatetime**

- Show all interviews for an applicant, sorted by DateTime

```
SELECT      Stage, DateTime, ApplicationName
FROM        INTERVIEW
WHERE       ApplicantUsername = user
ORDER BY    DateTime;
```

- Show interview details related to a specific job

```
SELECT      W.Stage, W.DateTime, W.ApplicationName
FROM        INTERVIEW AS W, MENTIONS AS M
WHERE       M.ApplicantUsername = user
AND         M.CompName = company
AND         M.PosName = position
AND         M.ApplicantUsername = W.ApplicantUsername
AND         M.InterID = W.InterviewID;
```

- Show all jobs that an interview is related to, specifying interview by DateTime

```
SELECT      J.*
```

```
FROM        JOB AS J, INTERVIEW AS W, MENTIONS AS M
WHERE       M.ApplicantUsername = user
AND         W.DateTime = interviewDatetime
AND         M.ApplicantUsername = W.ApplicantUsername
AND         M.InterID = W.InterviewID;
```

- Show all contacts that will be attending an interview and the applicant’s relationship to the contact(s), specifying interview by DateTime

```
SELECT      C.LinkedInURL, C.Fname, C.Lname, K.Relationship, K.Notes,
            K.LastContactDate
FROM        CONTACT AS C, ATTENDS AS A, KNOWS AS K, INTERVIEW AS W
WHERE       W.DateTime = interviewDatetime
AND         W.ApplicantUsername = user
AND         K.ApplicantUsername = user
AND         A.ContID = C.ContactID
AND         A.ApplicantUsername = W.ApplicantUsername
AND         A.InterID = W.InterviewID
AND         K.ContID = C.ContactID;
```

—

Searching for Contacts:

Assumptions:

- User’s username is stored in variable **user**
- Optionally, user defines a contact’s first name in variable **firstName**
- Optionally, user defines a contact’s last name in variable **lastName**
- Optionally, user defines a specific company in variable **company**
- Optionally, user defines a specific relationship to a contact in variable **relation**

- Show all contacts for an applicant and the applicant’s relationship to them

```
SELECT      C.LinkedInURL, C.Fname, C.Lname, K.Relationship, K.Notes,
            K.LastContactDate
FROM        CONTACT AS C, KNOWS AS K
WHERE       K.ApplicantUsername = user
AND         K.ContID = C.ContactID;
```

- Show relationship and other relevant info for a specific contact, identified by first and last name

```
SELECT      C.LinkedInURL, K.Relationship, K.Notes, K.LastContactDate
FROM        CONTACT AS C, KNOWS AS K
WHERE       K.ApplicantUsername = user
AND         C.Fname = firstName
AND         C.Lname = lastName
AND         K.ContID = C.ContactID;
```

- Show all emails for a specific contact, identified by first and last name

```
SELECT      Email
FROM        CONTACT_EMAIL NATURAL JOIN CONTACT
WHERE       Fname = firstName
AND         Lname = lastName;
```

- Show all phone numbers for a specific contact, identified by first and last name

```
SELECT      Phone
FROM        CONTACT_PHONE NATURAL JOIN CONTACT
WHERE       Fname = firstName
AND         Lname = lastName;
```

- Show all contacts with a specific relationship to the applicant

```
SELECT      C.LinkedInURL, C.Fname, C.Lname, K.Notes, K.LastContactDate
FROM        CONTACT AS C, KNOWS AS K
WHERE       K.ApplicantUsername = user
AND         K.Relationship = relation
AND         K.ContID = C.ContactID;
```

- Show all contacts who work at a specific company

```
SELECT      C.LinkedInURL, C.Fname, C.Lname, W.Role, K.Relationship, K.Notes,
            K.LastContactDate
FROM        CONTACT AS C, KNOWS AS K, WORKS_AT AS W
WHERE       K.ApplicantUsername = user
AND         W.CompName = company
AND         K.ContID = C.ContactID
AND         W.ContID = C.ContactID;
```

- Show all contacts who have provided a referral, and for which job

```
SELECT      C.LinkedInURL, C.Fname, C.Lname, R.Date, R.Notes, R.CompName,
            R.PosName
FROM        CONTACT AS C, REFERRAL AS R
WHERE       R.ApplicantUsername = user
AND         R.ContID = C.ContactID;
```

—

Searching for Job Offers:

Assumptions:

- User’s username is stored in variable **user**
- Optionally, user defines an offer file name in variable **filename**

- Show all job offers for an applicant, sorted by ResponseDeadline

```
SELECT      OfferFileName, ResponseDeadline, Compensation, StartDate, Notes,
            CompName, PosName
FROM        OFFER
WHERE       ApplicantUsername = user
ORDER BY    ResponseDeadline;
```

- Show all job offers for an applicant, sorted by Compensation

```
SELECT      OfferFileName, ResponseDeadline, Compensation, StartDate, Notes,
            CompName, PosName
FROM        OFFER
WHERE       ApplicantUsername = user
ORDER BY    Compensation;
```

- Show details of the job that a job offer corresponds to, if the applicant only knows the OfferFileName but not the name of the job

```
SELECT      O.ResponseDeadline, O.Compensation, O.StartDate, O.Notes, J.CompName,
            J.PositionName, J.Description, J.JobPostFile, J.PositionType
FROM        OFFER AS O, JOB AS J
WHERE       O.OfferFileName = filename
AND         O.ApplicantUsername = user
AND         O.CompName = J.CompName
AND         O.PosName = J.PositionName;
```

—

<u>Searching for Referrals:</u>

Assumptions:

- User’s username is stored in variable **user**
- Optionally, user defines specific company, in variable **company**
- Optionally, user defines specific job position, in variable **position**
- Optionally, user defines a contact’s first name in variable **firstName**
- Optionally, user defines a contact’s last name in variable **lastName**
- Optionally, user defines a target date in variable **targetDate**

- Show all referrals for an applicant and who provided the referral

```
SELECT      Date, Notes, CompName, PosName, Fname, Lname, LinkedInURL
FROM        REFERRAL, CONTACT
WHERE       ApplicantUsername = user
AND         ContID = ContactID;
```

- Show all referrals provided for a specific job, and by whom

```
SELECT      Date, Notes, Fname, Lname, LinkedInURL
FROM        REFERRAL, CONTACT
WHERE       ApplicantUsername = user
```

```
AND         CompName = company
AND         PosName = position
AND         ContID = ContactID;
```

- Show all referrals provided for a specific company, and by whom

```
SELECT      Date, Notes, PosName, Fname, Lname, LinkedInURL
FROM        REFERRAL, CONTACT
WHERE       ApplicantUsername = user
AND         CompName = company
AND         ContID = ContactID;
```

- Show all referrals provided by a specific contact, who is identified by first and last name

```
SELECT      Date, Notes, CompName, PosName
FROM        REFERRAL, CONTACT
WHERE       ApplicantUsername = user
AND         Fname = firstName
AND         Lname = lastName
AND         ContID = ContactID;
```

- Show all referrals provided after a target date, and by whom, sorted by Date

```
SELECT      Date, Notes, CompName, PosName, Fname, Lname, LinkedInURL
FROM        REFERRAL, CONTACT
WHERE       ApplicantUsername = user
AND         Date >= targetDate
AND         ContID = ContactID;
```

- For data manipulation queries, retrieve the ID of a contact who is identified by first and last name

```
SELECT      ContactID
FROM        CONTACT
WHERE       Fname = firstName
AND         Lname = lastName;
```

—

Searching for Companies:

Assumptions:

- User’s username is stored in variable **user**
- Optionally, user defines a particular industry in variable **industry**
- Optionally, user defines specific company, in variable **company**

- Show all companies, and their details, that are relevant to an applicant, that is, all companies whose jobs an applicant has applied to or is tracking

```
(SELECT     Y.*
FROM        COMPANY AS Y
WHERE       Y.CompanyName IN  (SELECT     DISTINCT C.CompName
                              FROM        CORRESPONDS_TO AS C
                              WHERE       C.ApplicantUsername = user))
UNION
(SELECT     Y.*
FROM        COMPANY AS Y
WHERE       Y.CompanyName IN  (SELECT     DISTINCT T.CompName
                              FROM        TRACKS AS T
                              WHERE       T.ApplicantUsername = user));
```

- Show all companies, and their details, that are relevant to an applicant, from a specific industry

```
(SELECT     Y.*
FROM        COMPANY AS Y
WHERE       Y.CompanyName IN  (SELECT     DISTINCT C.CompName
                              FROM        CORRESPONDS_TO AS C
                              WHERE       C.ApplicantUsername = user)
AND         Y.Industry = industry)
UNION
(SELECT     Y.*
FROM        COMPANY AS Y
WHERE       Y.CompanyName IN  (SELECT     DISTINCT T.CompName
                              FROM        TRACKS AS T
                              WHERE       T.ApplicantUsername = user)
AND         Y.Industry = industry);
```

- Show all companies where an applicant has contacts

```
SELECT      Y.*
FROM        COMPANY AS Y
WHERE       Y.CompanyName IN  (SELECT     DISTINCT W.CompName
                              FROM        WORKS_AT AS W, KNOWS AS K
                              WHERE       K.ApplicantUsername = user
                              AND         W.ContID = K.ContID);
```

- Show all info for a specific company

```
SELECT      Industry, HomePageURL, Description
FROM        COMPANY
WHERE       CompanyName = company;
```

—

Searching for Applicant Details:

Assumptions:

- User’s username is stored in variable **user**

- Show basic information about applicant

```
SELECT      Fname, Lname, Education, IsActive
FROM        APPLICANT NATURAL JOIN USER
WHERE       Username = user;
```

- Show all active applicants

```
SELECT      Fname, Lname, Education
FROM        APPLICANT NATURAL JOIN USER
WHERE       IsActive = 1;
```

- Show all deactivated applicants

```
SELECT      Fname, Lname, Education
FROM        APPLICANT NATURAL JOIN USER
WHERE       IsActive = 0;
```

- Show all experiences for an applicant

```
SELECT      Experience, ExperienceDesc
FROM        APPLICANT_EXPERIENCE
WHERE       ApplicantUsername = user;
```

- Show all projects for an applicant

```
SELECT      Project, ProjectDesc
FROM        APPLICANT_PROJECT
WHERE       ApplicantUsername = user;
```

- Show all certifications for an applicant

```
SELECT      Certification
FROM        APPLICANT_CERTIFICATION
WHERE       ApplicantUsername = user;
```

- Show all skills for an applicant

```
SELECT      Skill
FROM        APPLICANT_SKILL
WHERE       ApplicantUsername = user;
```

- Show all competitions for an applicant

```
SELECT      Competition
FROM        APPLICANT_COMPETITION
WHERE       ApplicantUsername = user;
```

—

<u>Searching for Admin Details (to display when an admin logs in):</u>

Assumptions:

- User’s username is stored in variable **user**

- Show basic information about admin

```
SELECT      Fname, Lname, IF(DeveloperFlag = 1, DeveloperType, “Not a developer”),
            IsActive
FROM        ADMIN NATURAL JOIN USER
WHERE       Username = user;
```

- Show all active admins

```
SELECT      Fname, Lname, IF(DeveloperFlag = 1, DeveloperType, “Not a developer”),
FROM        ADMIN NATURAL JOIN USER
WHERE       IsActive = 1;
```

- Show all deactivated admins

```
SELECT      Fname, Lname, IF(DeveloperFlag = 1, DeveloperType, “Not a developer”),
FROM        ADMIN NATURAL JOIN USER
WHERE       IsActive = 0;
```

- Show all responsibilities for admin

```
SELECT      Responsibility
FROM        ADMIN_RESPONSIBILITY
WHERE       AdminUsername = user;
```

- Show all specializations for a developer

```
SELECT      Specialization
FROM        DEV_SPECIALIZATION, ADMIN
WHERE       Username = user
AND         DeveloperFlag = 1
AND         Username = AdminUsername;
```

- Show all permission abilities that an admin has

```
SELECT      Ability
FROM        PERMISSION_ABILITY, ADMIN_HAS_PERM
WHERE       AdminUsername = user
AND         PermLevel = PermissionLevel;
```