**PROJECT REPORT**

**DATABASE FOR UNIVERSITIES RECRUITMENT MANAGEMENT**

**Oracle Account:**

**Alekhya Vanga**

* User Name: **av118**
* Password: **xt63hQDd**

**1)DESIGN PHASE:**

**A)DESCRIPTION OF THE SYSTEM:**

* **ENTERPRISE DESCRIPTION:** This is the database which is maintained by the database team of the company. Here company comes to a particular university and hire the students based on the required skills, if the student is eligible for all the skills described for a particular job, then he can attend the interview and if he clears the interview he gets the job. Here many companies can visit many universities.
* **SYSTEM PERFORMANCE:** This system must provide all the information about the student and his skills to the company as well as university who are eligible for the job and who are going to attend the interview.
* **END USERS:**

1) Company recruiters (who interview the student)

2)Student Applicant

* **DATA OBSOLESCENCE:** The university recruitment process is always conducted on large scale, and the data maintained by them is very huge. As the recruitments are done every year on a large basis they need to maintain large data. To keep track of this data, they need to backup the data and delete the unwanted data from the tables. The main benefit or advantage of this process is that the company can keep the track of the data. They can also perform analysis of data that how many have hired every time they conduct the recruitments in universities.
* **IDEA FOR THE PROJECT:** I was doing my undergrad and when I was in my final year, we are told to apply for the on campus jobs by consulting our faculty members and enquiring them about each and every job and its description, and we students found it as a difficult task to go and talk about each job description, So I have implemented that idea, so that it becomes easy for every student if he can access the database and apply for jobs and prefer different companies depending upon their interests.

**B) ENTITY RELATIONSHIP DESIGN:**

* **ENTITIES:**

**1) Company** (C\_Id, Cname, Clocation, Cemail, Cphone)

**Description:** In this entity, a company visits university, it posts some job requirements and based on the skills of the students it conducts the interview.

* C\_Id is the company unique ID.
* Cname is the name of the company.
* Clocation is the place where company is located.
* Cemail is the email of the company.
* Cphone is the phone number or contact number of the company.

**2) University** (U\_Id, Uniname, Unilocation, Uniemail, Uniphone)

**Description:** This entity university holds all the university details like name, address, mail, phone number.

* U\_Id is the unique ID for university
* Uniname is the university name.
* Unilocation is the university location.
* Uniemail is the email of the university.
* Uniphone is the phone number of the university.

**3) Interview** (I\_Id, Iname, Ilocation, Interviewdate)

**Description:** In this entity, it holds the details of the interviewer who is going to take the interview and interview details like place and date. The interview is attended by the student who has the skills required by the company.

* I\_Id is the unique ID for the interviewer who is going to take the particular interview.
* Iname is the interviewer name who is taking the interview.
* Ilocation is the location at which interview is conducted.
* Interviewdate is the date on which interview is conducted or held.

**4) Job\_Req** (R\_Id, Rname, Rlocation, Intern\_Experience)

**Description:** This entity describes the job requirement details like name, place where the person is required to go if he or she is selected for the job.

* R\_Id is the unique id for the particular job requirement.
* R\_name is the requirement name.
* R\_location is the place where the position is available,for example if the job position is available in boston, then Rlocation indicates that.
* Intern\_experience is the requirement telling that the student must have internship done before and its further requirement like example: six months of intern or one year of intern based on the companies wish.

**5) Student\_ Applicant** (S\_Id, Sname, Saddress, Sbranch, Semail, Sphone)

**Description:** This entity gives the information about the students who are applying for jobs.

* S\_Id is the unique id for the students who are applying for jobs.
* Sname is the name of the student who is applying for a job.
* Saddress is the students address.
* Sbranch is the branch name of the students,for example there are many branches like computer science, electronics etc.
* Semail is the email address of the student.
* Sphone is the phone number of the students.

**6) Skillset** (Ss\_Id, Ssname)

**Description:** This entity describes about a particular skill.

* Ss\_Id is the unique ID for the skill.
* Ssname is the name of a particular skill.
* **RELATIONSHIPS:**

**1) Company/University (visits)** 🡪**Many to Many**

As many companies visit many universities, the relation here is given as many to many relationship.

**2) Company/Interview (conducts)🡪One to Many**

This is one to many relationship because one company can conduct many interviews.

**3) Student \_Applicant/Interview (attends)🡪one to many**

This is one to many relationship because one student can apply and attend many interviews.

**4) Student\_Applicant/Job\_Requirement (Applied\_by\_student)🡪Many to Many**

This is a many to many relationship because, many students apply for many jobs based on the job requirements and skills required.

**5) Company/Job\_Requirement (posts)🡪One to Many**

This is a one to many relationship as one company can post many job requirements.

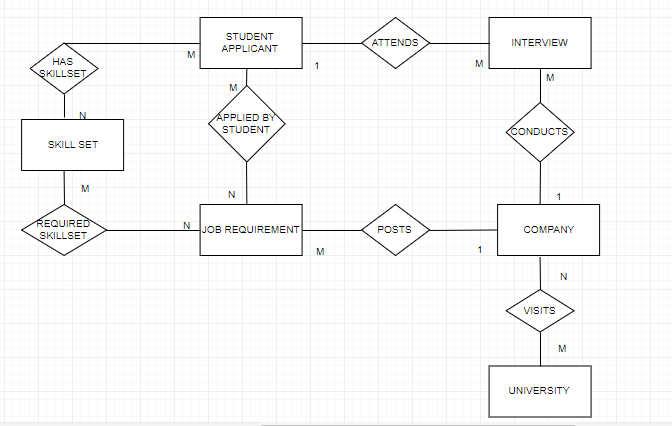
**6) Job\_Requirement/Skillset (Required\_skillset)🡪Many to Many**

This is a many to many relationship because many job requirements need many skills in order to be selected for a job.

**7) Skillset/Student\_applicant (has\_skillset)🡪Many to Many**

This is many to many relationship because many students have many skills, based on the skills required they attend the interview.

**ER-DIAGRAM:**



**C) CONCEPTUAL LEVEL:**

All the tables below are in third normal form (3NF).

Here all the primary keys are written in **BOLD** and all the foreign keys are underlined.

* **University(U\_Id,**Uniname,Unilocation,Uniemail,Uniphone)

**1) Primary key:** U\_Id is the primary key in this table.

**2) Column domain:**U\_Id number(3),Uniname char(10),Unilocation char(15),Uniemail varchar(10),Uniphone number(10).

**3) Domain Integrity checks:**

U\_Id is the primary key in this table and this is unique and not null.

Uniname is a character and not null.

Unilocation is a character and not null.

Uniemail is a variable & character (varchar) and not null.

Uniphone is a number and not null.

**4) Functional Dependencies:**

U\_Id🡪Uniname,Unilocation,Uniemail,Uniphone.

* **Company**(**C\_Id**,Cname,Clocation,Cemail,Cphone)

**1) Primary key:C\_Id** is the primary key in this table.

**2) Column domain:**C\_Id number(3),Cname char(10),Clocation char(10),Cemail varchar(10),Cphone number(10).

**3) Domain Integrity checks:**

C\_Id is the primary key in this table and this is unique and not null.

Cname is a character and not null.

Clocation is a character and not null.

Cemail is a variable & character (varchar) and not null.

Cphone is a number and not null.

**4) Functional Dependencies:**

C\_Id🡪Cname,Clocation,CemaiL,Cphone.

* **Student\_applicant**(**S\_Id,**Sname,Saddress,Sbranch,Semail,Sphone)

**1) Primary key: S\_Id** is the primary key in this table.

**2) Column domain:**S\_Id number(3),Sname char(10),Saddress char(10),Sbranch char(10),Semail varchar(10),Sphone number(10).

**3) Domain Integrity checks:**

S\_Id is the primary key in this table and this is unique and not null.

Sname is a character and not null.

Saddress is a character and not null.

Sbranch is a character and not null.

Semail is a variable & character (varchar) and not null.

Sphone is a number and not null.

**4) Functional Dependencies:**

S\_Id🡪Sname,Saddress,Sbranch,Semail,Sphone.

* **Interview**(**I\_Id,**Iname,Ilocation,Interviewdate,C\_Id,S\_Id)

**1) Primary key:S\_Id** is the primary key in this table.

**2) Foreign Keys:**C\_Id is referencing company table and S\_Id is referencing Student\_Applicant table.

**3) Column domain:**I\_Id number(3),Iname char(10),Ilocation char(10),Interviewdate date,C\_Id number(3),S\_Id number(3).

**4) Domain Integrity checks:**

I\_Id is the primary key in this table and this is unique and not null.

Iname is a character and not null.

Ilocation is a character and not null.

Interviewdate is date and not null.

C\_Id is a number and not null.

S\_Id is a number and not null.

**5) Functional Dependencies:**

I\_Id🡪Iname,Ilocation,Interviewdate,C\_Id,S\_Id.

* **Job\_Requirement**(**R\_Id**,Rname,Rlocation,Intern\_Experience,C\_Id)

**1) Primary key:R\_Id** is the primary key in this table.

**2) Foreign key:**C\_Id is the foreign key referenced from company table.

**3) Column domain:**R\_Id number(2),Rname char(20),Rlocation char(10),Intern\_Experience varchar(10),C\_Id number(3).

**4) Domain Integrity checks:**

R\_Id is the primary key in this table and this is unique and not null.

Rname is a character and not null.

Rlocation is a character and not null.

Intern\_experience is a variable & character and not null.

C\_Id is a number and not null.

**5) Functional Dependencies:**

R\_Id🡪Rname,Rlocation,Intern\_Experience,C\_Id.

* **Skillset**(**Ss\_Id,**Ssname)

**1) Primary key:Ss\_Id** is the primary key in this table.

**2) Column domain:**Ss\_Idnumber(2),Ssname char(10).

**3) Domain Integrity checks:**

Ss\_Id is the primary key in this table and this is unique and not null.

Ssname is a character and not null.

**4) Functional Dependencies:**

Ss\_Id🡪Ssname

* **Visits**(**U\_Id,C\_Id)**

**1) Primary key:U\_Id,C\_Id** is the primary keys in this table.

**2) Foreign key:**U\_Id,C\_Id.

**3) Column domain:**U\_Id number(3),C\_Id number(3).

**4) Domain Integrity checks:**

U\_Id is the primary key in this table and this is unique and not null and U\_Id is also the foreign key referencing from university table.

C\_Id is a primary key and not null and also foreign key referencing from company table.

* **Required\_Skillset**(**R\_Id,Ss\_Id**)

**1) Primary key:R\_Id,Ss\_Id** are the primary keys in this table.

**2) Foreign key:**R\_Id,Ss\_Id.

**3) Column domain:**R\_Id number(2),Ss\_Id number(3).

**4) Domain Integrity checks:**

R\_Id is the primary key in this table and not null and R\_Id is also the foreign key referencing from Job\_Requirement table.

Ss\_Id is the primary key in this table and not null and Ss\_Id is also the foreign key

Referencing from skillset table.

* **Applied\_By\_Student**(**R\_Id,S\_Id**)

**1) Primary key:R\_Id,S\_Id** are the primary keys in this table.

**2) Foreign key:**R\_Id,S\_Id are the foreign keys.

**3) Column domain:**R\_Id number(2),S\_Id number(3).

**4) Domain Integrity checks:**

R\_Id is the primary key in this table and not null and R\_Id is also the foreign key referencing from Job\_Requirement table.

S\_Id is the primary key in this table and not null and S\_Id is also the foreign key

Referencing from Student\_Applicant table.

* **Have\_skillset**(**S\_Id,Ss\_Id**)

**1) Primary key:S\_Id,Ss\_Id** are the primary keys in this table.

**2) Foreign key:**S\_Id,Ss\_Id are the foreign keys.

**3) Column domain:**S\_Id number(3),Ss\_Id number(2).

**4) Domain Integrity checks:**

S\_Id is the primary key in this table and not null and S\_Id is also the foreign key referencing from Student\_Applicant table.

Ss\_Id is the primary key in this table and not null and Ss\_Id is also the foreign key

Referencing from skillset table.

**D)EXTERNAL VIEW:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Table name** | **Company** | **University** | **Student\_Applicant** |
| Company | Select, insert, update, delete | Select | Select |
| University | Select | Select, insert, update, delete | Select, insert |
| Interview | Select, insert, update, delete | Select | Select |
| Student\_Applicant | Select | Select | Select, insert, update, delete |
| Job\_Requirement | Select, insert, update, delete | Select | Select |
| Skillset | Select, insert, update, delete | Select | Select |
| Has \_ skillset | Select | Select | Select, delete |
| Required\_Skillset | Select, insert, update, delete | Select | Select |
| Applied\_By\_Student | Select | Select | Select,delete |
| Visits | Select, insert, update. | Select, insert, update. | Select |

**E) INTERNAL VIEW:**

* **Built-In Functions:**

SELECT JOB\_REQUIREMENT.R\_ID,

COUNT(\*) AS "TOTAL STUDENTS"

FROM JOB\_REQUIREMENT,

APPLIED\_BY\_STUDENT,

STUDENT\_APPLICANT

WHERE JOB\_REQUIREMENT.R\_ID = APPLIED\_BY\_STUDENT.R\_ID

AND APPLIED\_BY\_STUDENT.S\_ID = STUDENT\_APPLICANT.S\_ID

GROUP BY JOB\_REQUIREMENT.R\_ID;

**Optimization:**

1. Π(Job\_Requirement.R\_Id) [σ(Job\_Requirement ×Applied\_By\_Student×Student\_Applicant)

(Job\_Requirement.R\_Id = Applied.By\_Student.R\_Id ∧Applied\_By\_Student.S\_Id = Student\_Applicant.S\_Id)

1. Π(Job\_Requirement.R\_Id)((Job\_Requirement ⋈ Applied\_By\_Student ⋈ Student\_Applicant)
2. Π(Job\_Requirement.R\_Id)( Applied\_By\_Student ⋈ Student\_Applicant ⋈ [Job\_Requirement.R\_Id 𝓖Count(\*) “Total Students”])

**File structure for the above query:**

Secondary b-tree ON R\_ID IN JOB\_REQUIREMENT table

Clustered B Tree on S\_ID in STUDENT\_APPLICANT table

Clustered B Tree on R\_ID in APPLIED\_BY\_STUDENT table

* **Range Query:**

SELECT INTERVIEW.I\_ID,

INTERVIEW.INTERVIEWDATE,

STUDENT\_APPLICANT.S\_ID,

STUDENT\_APPLICANT.SNAME,

STUDENT\_APPLICANT.SEMAIL

FROM STUDENT\_APPLICANT

INNER JOIN INTERVIEW

ON STUDENT\_APPLICANT.S\_ID = INTERVIEW.S\_ID

AND INTERVIEW.INTERVIEWDATE < '06-01-2018 10:30';

**Optimisation:**

1. Π(Interview.I\_Id, Interview.I\_Date,Student\_Applicant.S\_Id,Student\_Applicant.SName , Student\_Applicant.Semail)[ σ(Student\_Applicant × Interview)

(Interview.IDate<’06-01-2018 10:30’ ∧ Student\_Applicant.S\_Id = Interview.S\_Id)

1. Π(Interview.I\_Id, Interview.I\_Date,Student\_Applicant.S\_Id,Student\_Applicant.SName , Student\_Applicant.Semail)[ σ(Student\_Applicant ⋈ Interview)]

(Interview.IDate<’06-01-2018 10:30’) (Student\_Applicant.S\_Id =

Interview.S\_Id)

1. Π(Interview.I\_Id, Interview.I\_Date,Student\_Applicant.S\_Id,Student\_Applicant.SName , Student\_Applicant.Semail)(Student\_Applicant ⋈ σ Interview

(Student\_Applicant.S\_Id = Interview) (Interview.IDate < ’06-01-2018 10:30’)

**File structure for the above query:**

Secondary b-tree on INTERVIEW.INTERVIEWDATE in INTERVIEW table

* **Simple search and join query:**

SELECT

SKILLSET.SSNAME, JOB\_REQUIREMENT.RNAME

FROM SKILLSET,

JOB\_REQUIREMENT,

REQUIRED\_SKILLSET

WHERE JOB\_REQUIREMENT.R\_ID = REQUIRED\_SKILLSET.R\_ID

AND REQUIRED\_SKILLSET.SS\_ID = SKILLSET.SS\_ID

AND JOB\_REQUIREMENT.R\_ID = 45;

**Optimisation:**

1. Π(Skillset.SSname, Job\_Requirement.RName)( σ(skillset x job\_requriment x requriment\_skillset))

(job\_requirement.r\_id=required\_skillset.r\_id ∧ required\_skillset.ss\_id = skillset.ss\_id ∧ job\_requirement.r\_id = 45)

1. Π(Skillset.SSname, Job\_Requirement.RName)( σ(skillset ⋈ job\_requriment ⋈ requriment\_skillset))

(job\_requirement.r\_id = 45)

1. Π(Skillset.SSname, Job\_Requirement.RName)( σ(job\_requriment )⋈ requriment\_skillset ⋈ skillset)

(job\_requirement.r\_id = 45)

**File structure for the above query:**

Hashing on R\_ID in JOB\_REQUIREMENT.

Clustered B-Tree on SS\_ID in SKILLSET.

Clustered B-Tree on R\_ID in REQUIRED\_SKILLSET

**F) DATA DICTIONARY:**

1. **Company**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Null / Not Null** | **Data Type** |
| C\_Id | Not Null | Number(3) |
| Cname | Not Null | Char(10) |
| Clocation | Not Null | Char(10) |
| Cemail | Not Null | Varchar(10) |
| Cphone | Not Null | Number(10) |

**Primary Key:** C\_Id

**Privilages:**  Select on university.

Select, insert, update, delete on company.

1. **University**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Null/ Not Null** | **Data Type** |
| U\_Id | Not Null | Number(3) |
| Uniname | Not Null | Char(10) |
| Unilocation | Not Null | Char(15) |
| Unimail | Not Null | Varchar(10) |
| Uniphone | Not Null | Number(10) |

**Primary Key:** U\_Id

**Privilages:**  Select on company.

Select, insert, update, delete on university.

1. **Visit**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Null/Not Null** | **Data Type** |
| U\_Id | Not Null | Number(3) |
| C\_Id | Not Null | Number(3) |

**Primary Key:** U\_Id, C\_Id

**Foreign Key :** U\_Id References university,

C\_Id References company.

**Privilages:** Select, insert, update on university.

Select, insert, update on company.

1. **Job\_Requirement**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Null/ Not Null** | **Data Type** |
| R\_Id | Not Null | Number(2) |
| Rname | Not Null | Char(20) |
| Rlocation | Not Null | Char(10) |
| Intern\_Experience | Not Null | Varchar(10) |
| C\_Id | Not Null | Number(3) |

**Primary key:** R\_Id

**Foreign key:**C\_Id References company.

**Privilages:** Select, Insert , Update, Delete on Company

Select on university

1. **Student\_Applicant**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Null/ Not Null** | **Data Type** |
| S\_Id | Not Null | Number(3) |
| Sname | Not Null | Char(10) |
| Saddress | Not Null | Char(10) |
| Sbranch | Not Null | Char(10) |
| Semail | Not Null | VarChar(10) |
| Sphone | Not Null | Number(10) |

**Primary Key :** S\_Id

**Foreign Key:** There are no foreign keys

**Privilages:** Select on Company

Select on University

1. **Skillset**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Null/Not Null** | **Data Type** |
| Ss\_Id | Not Null | Number(2) |
| Ss\_name | Not Null | Char(10) |

**Primary Key:** Ss\_Id

**Foreign Key:**There are no foreign keys

**Privilages:** Select , Insert, Update, Delete on Company

Select on University

1. **Required\_Skillset**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Null/Not Null** | **Data Type** |
| R\_Id | Not Null | Number(2) |
| Ss\_Id | Not Null | Number(2) |

**Primary Keys:** R\_Id, Ss\_Id

**Foreign Keys:** R\_Id references Job\_Requirement

SS\_Id references Skillset

**Privilages:** Select, Insert, Update, Delete on Company

Select on University

**8)Appplied\_by\_Student**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Null/Not Null** | **Data Type** |
| R\_Id | Not Null | Number(2) |
| S\_Id | Not Null | Number(3) |

**Primary Key:** R\_Id, S\_Id

**Foreign Key:** R\_Id references Job\_Requirement

S\_Id references Student\_Applicant

**Privilages:** Select on Company

Select on University

1. **Has\_Skillset:**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Null/Not Null** | **Data Type** |
| S\_Id | Not Null | Number (3) |
| SS\_Id | Not Null | Number (2) |

**Primary key:** S\_Id , Ss\_Id

**Foreign key:** S\_Id references Student\_Applicant

Ss\_Id references SkillSet

**Privilages:** Select on Company

Select on University

1. **Interview:**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Null/Not Null** | **Data Type** |
| I\_Id | Not Null | Number (3) |
| Iname | Not Null | Char(10) |
| Ilocation | Not Null | Char(10) |
| Interviewdate | Not Null | Date |
| C\_Id | Not Null | Number(3) |
| S\_Id | Not Null | Number(3) |

**Primary keys:** I\_Id

**Foreign keys:** C\_Id references company

S\_Id references Student\_Applicant

**Privilages:** Select , Insert ,Update, Delete on Company

Select on University

* **VIEWS:**

**1) View created for interview date:**

CREATE VIEW interviewdate   
AS   
SELECT  interview.i\_id,   
          interview.interviewdate,   
          student\_applicant.s\_id,   
          student\_applicant.sname,   
          student\_applicant.semail   
  FROM   student\_applicant   
  INNER JOIN interview   
  ON student\_applicant.s\_id = interview.s\_id   
  AND interview.interviewdate < '06-01-2018 10:30';

**2) View created for skill required**

CREATE VIEW skillreq   
AS

 SELECT skillset.ssname,   
          job\_requirement.rname   
 FROM   skillset,   
          job\_requirement,   
          required\_skillset

 WHERE  job\_requirement.r\_id = required\_skillset.r\_id   
  AND required\_skillset.ss\_id = skillset.ss\_id   
  AND job\_requirement.r\_id = 45;

**3) View created for total students applied for jobs**

CREATE VIEW totalstudentapllied   
AS   
  SELECT job\_requirement.r\_id,   
          *Count*(\*) AS "TOTAL STUDENTS"   
  FROM   job\_requirement,   
          applied\_by\_student,   
          student\_applicant   
  WHERE  job\_requirement.r\_id = applied\_by\_student.r\_id   
  AND applied\_by\_student.s\_id = student\_applicant.s\_id   
  GROUP  BY job\_requirement.r\_id;

**4) View created for student applied company**

CREATE VIEW studentappliedcompany   
AS   
SELECT student\_applicant.sname,   
          company.cname,   
          university.uniname   
 FROM   student\_applicant,   
          company,   
          university,   
          job\_requirement,   
          applied\_by\_student,   
          visits   
 WHERE  university.u\_id = visits.u\_id   
 AND visits.c\_id = company.c\_id   
 AND company.c\_id = job\_requirement.c\_id   
 AND job\_requirement.r\_id = applied\_by\_student.r\_id   
 AND applied\_by\_student.s\_id = student\_applicant.s\_id   
 GROUP  BY student\_applicant.sname,   
             company.cname,   
             university.uniname;

* **Indexes:**

1. CREATE INDEX University\_Uid\_Idx

ON University(U\_Id)

Here it creates secondary B tree on U\_Id in university table.

1. CREATE INDEX Computer\_Cid\_Idx

ON Company(C\_Id)

Here it creates secondary B tree on C\_id in Company table.

1. CREATE INDEX Student\_Applicant\_Sid\_Idx

ON Student\_Applicant(S\_Id)

Here it creates secondary B tree on S\_Id in Student\_Applicant table.

1. CREATE INDEX Interview\_Iid\_Idx

ON Interview(I\_Id)

Here it creates secondary B tree on I\_Id in Interview table.