TABLE OF CONTENTS

S.No.	Contents		Page No.
1	Introduction		2
	1.1	Business Overview and Description	2
	1.2	Database Description	2
	1.3	Value addition to the business by designing the database	3
2	Entity Relationship Diagram		4
3	Logical Schema		5
4	Data Dictionary		6
5	SQL Table Creation Statements		14
6	Views and Procedures		18
	6.1	Materialized View 1	18
	6.2	Materialized View 2	20
	6.3	Materialized View 3	21
	6.4	Materialized View 4	23
7	SQL Statements for database triggers		25

1. INTRODUCTION

1.1 BUSINESS OVERVIEW AND DESCRIPTION

Job Bridge is a recruiting specialist company which acts as a platform between the applicants who have prior working experience in a role and are looking to change from one company to the other.

Job Bridge has a Resume Tracking System which is a database built to store the details of a prospective Job Seeker who is in search of a new job in the United States. Job Bridge acts as a third party where it accepts resumes from the prospective Job Seekers and stores them and provides them to the companies which are looking for job applicants. Recruiting can be a nightmare to companies of all stages and sizes. So, the recruiting companies play a very crucial role in the present world. As Job Bridge acts as a third party between the applicants and the companies which are looking for new job applicants it stands in a beneficial position as it has the right resources in its database about the applicants and provides the companies the right candidates depending on their role requirements. Job Bridge is associated with many top companies in the United States which would help the applicants secure a job in the most efficient way. The company gives a tough competition to its competitors as it is associated with many top companies.

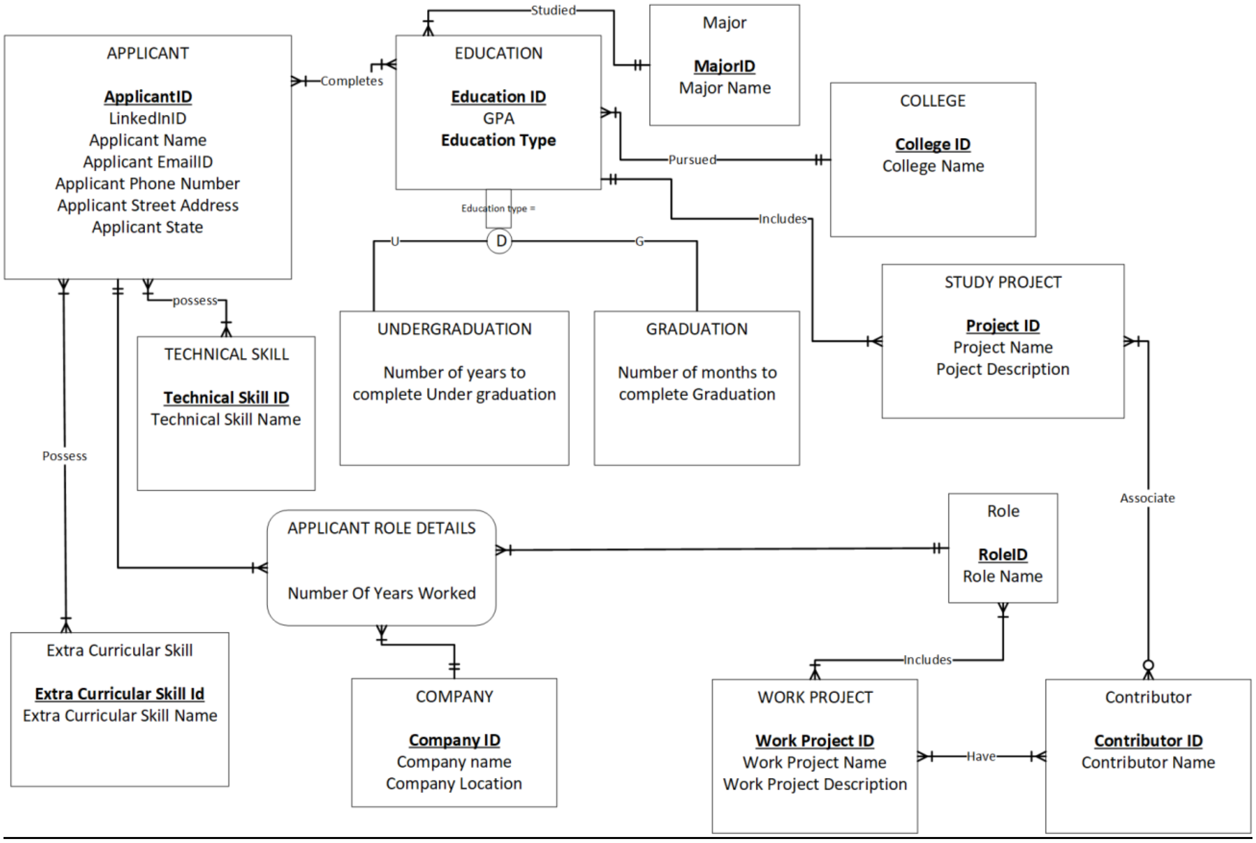
1.2 DATABASE DESCRIPTION

The database holds crucial details of the applicants which differentiate one applicant from the other. The database stores the Applicant Name, Applicant LinkedInID, Applicant Email Address, Applicant Street Address, and the state in which he lives. It also stores the Education Details of the Applicant which includes the Grade Point Average of the applicant in both the undergraduate and the graduation level depending upon the applicant. The company considers the applicants who have a major in Information Systems, Data Analytics, Computer Science, Business Analytics, Software Engineering, Computer Engineering, and Electronics. The database also stores the college name where the applicant has pursued his education. The database also stores the details of the Study projects that he has done during his education and the Work projects that he has done for a company in a role. This would help easy identification of the applicant work experience and the database also stores the number of years an applicant has worked for a company in a role. The company works with the applicants whose roles are restricted to Java Developer, Software Developer, Data Base administrator, Solutions Specialist, Cloud Engineer, Data Analyst, Electrical Engineer. The companies who are looking to recruit the applicants in these roles are only associated with the company. The database also stores the data of the contributor names who have worked with the applicant on the study projects or the Work Projects so that we can avoid the plagiarism of the projects. This plagiarism can be rectified if there is a new applicant in the database and he has the same project name and the project description just like the applicant who is present in our database and if the new applicant's name is not on the contributors' list his resume would be flagged until the issue is resolved.

1.3 BUSINESS VALUE

According to www.Statista.com the recent estimated the Staffing and recruiting industry market size has increased from 148.3 Billion USD to 153.2 USD. As the market size is huge the Return on Investment should be high as it is still an untapped market when it comes to specific role hiring. According to the Wall Street Journal, 3.4 million people in the US have chosen to give their boss their two-week notice in April 2018. Employees who leave their job are more likely to make as much as 30 percent more in a new role in a new company. And this is the main reason many people look for changing their jobs frequently. Job Bridge considers the applicants who have prior work experience of more than a year and who are looking for a new job unlike, the other recruiting companies who also take into the applicants who do not have any prior work experience. This factor would help Job Bridge play a key role in the recruiting market. In the next phase, Job Bridge would like to add some more roles that they would account for in their company and would also like to spread across Canada, India, and Europe in the next 5 years. Keeping in view the growth of Job Bridge over the past three years, we are sure that the value of Job Bridge would increase to 10 million dollars in the next two years. We also expect the increase in the applicants by 23% by the end of 2019.

2. ENTITY RELATIONSHIP DIAGRAM



4. DATA DICTIONARY

Applicant_T

Name	Data Type	Constraints	Key	Description	Example Value
Applicant ID	int	>0	PK	Unique identifier for an applicant	102
Applicant LinkedIn ID	nvarchar(max)			LinkedIn ID of an applicant	Rohan_Jain
Applicant Name	nvarchar(50)			Name of an applicant	Rohan Jain
Applicant Email ID	nvarchar			Email ID of an applicant	R.Jain@gmail.com
Applicant Phone Number	int			Phone number of an applicant	6692309150
Applicant Street Address	nvarchar			Street Address of an applicant	P.O. Box 255, 2853 Consectetuer St.
Applicant State	char(25)			State of an applicant	California

ApplicantEducation_T

Name	Data Type	Constraints	Key	Description	Example Value
Applicant ID	int	>0	PK, FK	Unique identifier for an applicant, composite identifier for an applicant in an education	102
Education ID	int	>0	PK, FK	Unique identifier for Education, composite identifier for an applicant in an education	300

TechnicalSkills_T

Name	Data Type	Constraints	Key	Description	Example Value
Technical Skill ID	int	>0	PK	Unique identifier for a technical skill	10
Technical Skill Name	varchar(50)			Name of the technical skill	Java

ApplicantTechnicalSkills_T

Name	Data Type	Constraints	Key	Description	Example Value
Applicant ID	int	>0	PK, FK	Unique identifier for an applicant, composite identifier for a technical skill in an applicant	102
Technical Skill ID	int	>0	PK, FK	Unique identifier for Technical Skills, composite identifier for a technical skill in an applicant	10

College_T

Name	Data Type	Constraints	Key	Description	Example Value
College ID	int	>0	PK	Unique identifier for a college	1000
College Name	nvarchar(max)			Name of the college	Santa Clara University

Company_T

Name	Data Type	Constraints	Key	Description	Example Value
Company ID	int	>0	PK	Unique identifier for a company	10
CompanyName	nvarchar(max)			Name of the company	Allient Energy
Company Location	nvarchar(50)			Location of the company	California

Contributor_T

Name	Data Type	Constraints	Key	Description	Example Value
Contributor ID	int	>0	PK	Unique identifier for a contributor	3002
Contributor Name	nvarchar(max)			Name of contributor	Srivatsav

Education_T

Name	Data Type	Constraints	Key	Description	Example Value
EducationID	int	>0	PK	Unique identifier for an Education	300
MajorID	int	>0	FK	Major associated with an Education, Unique identifier for a Major.	2000
College ID	int	>0	FK	College associated with an Education, Unique identifier for a College.	1000
GPA	nvarchar(2)			Grade Point Average of an Applicant	A-
Education Type	char(1)	('U','G')		Discriminator for Education Type, Under graduation(U) or Graduation(G)	U

Graduation_T

Name	Data Type	Constraints	Key	Description	Example Value
Graduation ID	int	>0	PK, FK	Unique identifier for Graduation Education	306
Graduation GPA	nvarchar(2)			Grade Point Average in Graduation	A
Graduation Type	Varchar(1)	('G')		Discriminator for Graduation Education type, Graduation(G)	G
Number of Months To Complete Graduation	int	>=10		Number of months to complete Graduation	112

Undergraduation_T

Name	Data Type	Constraints	Key	Description	Example Value
UEducation ID	int	>0	PK, FK	Unique identifier for an Under graduation Education	300
Number of Years To Complete Under graduation	int	>=3		Number of Years to complete Under graduation	3

Major_T

Name	Data Type	Constraints	Key	Description	Example Value
Major ID	int	>0	PK	Unique identifier for a major	2000
Major Name	varchar(25)	(‘Information Systems’, ‘Data Analytics’, ‘Computer Science’, ‘Business Analytics’, ‘Software Engineering’, ‘Computer Engineering’, ‘Electronics’)		Name of the major	Information Systems

Role_T

Name	Data Type	Constraints	Key	Description	Example Value
Role ID	int	>0	PK	Unique identifier for a Role	2000
Role Name	nvarchar(max)	(‘Java Developer’, ‘Software Developer’, ‘Data Base Administrator’, ‘Solution Specialist’, ‘Cloud Engineer’, ‘Data Analyst’, ‘Electrical Engineer’)		Name of the Role	Java Developer

StudyProject_T

Name	Data Type	Constraints	Key	Description	Example Value
Study Project ID	int	>0	PK	Unique identifier for a study Project	1003
Education ID	int	>0	FK	Unique identifier for education	300
Study Project Name	nvarchar(max)			Name of the study project	Respondre
Study Project Description	nvarchar(max)			Description of the study project	Designed a system to answer user queries

StudyProjectsContributors_T

Name	Data Type	Constraints	Key	Description	Example Value
Study Project ID	int	>0	PK, FK	Unique identifier for a Study Project, composite identifier for a contributor in a study project	1000
Contributor ID	int	>0	PK, FK	Unique identifier for a contributor, composite identifier for a contributor in a study project	3000

TechnicalSkill_T

Name	Data Type	Constraints	Key	Description	Example Value
Technical Skill ID	int	>0	PK	Unique identifier for a technical skill	10
Technical Skill Name	varchar(50)			Name of the technical skill	Java

WorkProject_T

Name	Data Type	Constraints	Key	Description	Example Value
Work Project ID	int	>0	PK	Unique identifier for a Work Project	9012
Work Project Name	nvarchar(max)			Name of the work project	Effective Key Generation For Media
Work Project Description	nvarchar(max)			Description of work project	Application developed to embed a video file in another video signal

WorkProjectsContributors_T

Name	Data Type	Constraints	Key	Description	Example Value
WorkProject ID	int	>0	PK, FK	Unique identifier for a work project, composite identifier for a contributor in a work project	9000
Contributor ID	int	>0	PK, FK	Unique identifier for a contributor, composite identifier for a contributor in a work project	3004

ApplicantRoleDetails_T

Name	Data Type	Constraints	Key	Description	Example Value
Applicant Role Details ID	int	>0	PK	Unique identifier for the applicant role details	70
Applicant ID	int	>0	FK	Unique identifier for an applicant, applicant associated with role, company, number of work experience	102
Role ID	int	>0	FK	Unique identifier for a role. Role associated with company, applicant and number of years of work experience	2000
Company ID	int	>0	FK	Unique identifier for a company. Company associated with role, applicant and number of years of work experience	10
Number of Years Of Work Experience	int	>0	FK	Number of work experience of an applicant associated with a company in a role	2

ExtraCurricularSkill_T

Extra Curricular Skill ID	int	>0	PK	Unique identifier of an Extra Curricular skill	60
Extra Curricular Skill Name	Nvarchar(max)			Name of the Extra Curricular Skill	Electric Guitar

ApplicantExtraCurricularSKills_T

Extra Curricular Skill ID	Int	>0	PK, FK	Unique identifier of an Extra Curricular skill, composite identifier of a skill in an applicant	60
Applicant ID	int	>0	PK, FK	Unique identifier of an Applicant, composite identifier of a skill in an applicant	102

5.

5. SQL TABLE CREATION STATEMENTS

(IN ALPHABETICAL ORDER)

Applicant_T

```
Create Table Applicant_T
(ApplicantID int not null Check(ApplicantID > 0),
ApplicantLinkedInID nvarchar(max) not null,
ApplicantName varchar(25) not null,
ApplicantEmailID nvarchar(max),
ApplicantPhoneNumber int,
ApplicantStreetAddress nvarchar(max),
ApplicantState char(25),
Constraint ApplicantID_PK PRIMARY KEY (ApplicantID))
```

ApplicantEducation_T

```
Create Table ApplicantEducation_T
(ApplicantID int not null Check(ApplicantID > 0),
EducationID int not null Check (EducationID > 0),
Constraint ApplicantEducation_PK Primary Key(ApplicantID, EducationID),
Constraint ApplicantEducation_FK1 Foreign Key(ApplicantID) References
Applicant_T(ApplicantID),
Constraint ApplicantEducation_FK2 Foreign Key(EducationID) References
Education_T(EducationID))
```

ApplicantExtraCurricularSkills_T

```
Create Table ApplicantExtraCurricularSkills_T
(ApplicantID int not null Check(ApplicantID > 0),
ExtraCurricularSkillID int not null Check(ExtraCurricularSkillID > 0)
Constraint ApplicantExtraCurricularSkills_PK Primary Key(ApplicantID,
ExtraCurricularSkillID) ,
Constraint ApplicantExtraCurricularSkills_FK1 Foreign Key(ApplicantID) References
Applicant_T(ApplicantID),
Constraint ApplicantExtraCurricularSkills_FK2 Foreign Key(ExtraCurricularSkillID)
References ExtraCurricularSkills_T(ExtraCurricularSkillID))
```

ApplicantRoleDetails_T

```
Create Table ApplicantRoleDetails_T
(ApplicantRoleDetailsID int not null Check(ApplicantRoleDetailsID > 0),
NumberOfYearsOfWorkExperience int not null Check (NumberOfYearsOfWorkExperience > 0),
ApplicantID int, RoleID int, CompanyID int,
Constraint ApplicantRoleDetailsID_PK Primary Key(ApplicantRoleDetailsID),
Constraint Applicant_FK Foreign Key(ApplicantID) References Applicant_T(ApplicantID),
Constraint Role_FK2 Foreign Key(RoleID) References Roles_T(RoleID),
Constraint Company_FK3 Foreign Key(CompanyID) References Company_T(CompanyID))
```

ApplicantTechnicalSkills_T

```
Create Table ApplicantTechnicalSkills_T
(ApplicantID int not null Check(ApplicantID > 0),
TechnicalSkillID int not null Check(TechnicalSkillID > 0),
Constraint ApplicantTechnicalSkills_PK Primary Key(ApplicantID, TechnicalSkillID) ,
Constraint ApplicantTechnicalSkills_FK1 Foreign Key(ApplicantID) References
Applicant_T(ApplicantID),
Constraint ApplicantTechnicalSkills_FK2 Foreign Key(TechnicalSkillID) References
TechnicalSkills_T(TechnicalSkillID))
```

College_T

```
Create table College_T CHECK(CollegeID > 0)
(CollegeID int not null,
CollegeName nvarchar(max) not null,
Constraint CollegeID_PK Primary Key (CollegeID))
```

Company_T

```
Create table Company_T
(CompanyID int not null CHECK (CompanyID > 0)
CompanyName nvarchar(max) not null,
CompanyLocation nvarchar(50),
Constraint CompanyID_PK Primary Key (CompanyID))
```

Contributor_T

```
Create Table Contributor_T
(ContributorID int not null Check (ContributorID > 0),
ContributorName nvarchar(max) not null,
Constraint ContributorID_PK PRIMARY KEY (ContributorID)Education_T


Create Table Education_T

(EducationID int not null CHECK (EducationID > 0),
MajorID int not null,
CollegeID int not null,
GPA nvarchar(2),
EducationType char(1) CHECK (EducationType IN ('U', 'G')) not null,
Constraint EducationID_PK Primary Key(EducationID))
```

ExtraCurricularSkills_T

```
Create Table ExtraCurricularSkills_T
(ExtraCurricularSkillID int not null Check (ExtraCurricularSkillID > 0),
ExtraCurricularSkill nvarchar(max) not null,
Constraint ExtraCurricularSkills_T_PK PRIMARY KEY (ExtraCurricularSkillID))
```

Graduation_T

```
Create Table Graduation_T
(GEducationID int not null Check (GEducationID >= 10),
NumberofMonthsToCompleteGraduation int not null Check (NumberofMonthsToCompleteGraduation
>= 10),
Constraint Graduation_PK Primary Key (GEducationID),
Constraint Graduation_FK Foreign key (GEducationID) References Education_T(EducationID))
```

Major_T

```
Create table Major_T
(MajorID int not null CHECK (MajorID > 0),
MajorName varchar(25) not null CHECK(MajorName in ('Information Systems', 'Data
Analytics', 'Computer Science', 'Business Analytics', 'Software Engineering','Computer
Engineering', 'Electronics'),
Constraint MajorID_PK Primary Key (MajorID))
```

Role_T

```
Create table Role_T
(RoleID int not null CHECK (RoleID > 0),
RoleName nvarchar(max) not null CHECK(RoleName in ('Java Developer', 'Software
Developer', 'Data Base Administrator', 'Solution Specialist', 'Cloud Engineer', 'Data
Analyst','Electrical Engineer'),
Constraint MajorID_PK Primary Key (MajorID))
```

StudyProject_T

```
Create  table StudyProject_T
(StudyProjectID int not null CHECK(StudyProjectID > 0),
EducationID int not null,
StudyProjectName nvarchar(max) not null,
StudyProjectDescription nvarchar(max) not null,
Constraint StudyProjectID_PK Primary Key(StudyProjectID),
Constraint EducationID_FK Foreign Key(EducationID) References Education_T(EducationID))
```

StudyProjectsContributors_T

```
Create Table StudyProjectsContributors_T
(StudyProjectID int not null Check(StudyProjectID > 0),
ContributorID int not null Check (ContributorID > 0),
Constraint StudyProjectsContributors_PK Primary Key(StudyProjectID, ContributorID),
Constraint StudyProjectsContributors_FK1 Foreign Key(StudyProjectID) References
StudyProject_T(StudyProjectID),
Constraint StudyProjectsContributors_FK2 Foreign Key(ContributorID) References
Contributor_T(ContributorID))
```

TechnicalSkills_T

```
Create Table TechnicalSkills_T
(TechnicalSkillID int not null CHECK(TechnicalSkillID > 0),
TechnicalSkillName varchar(50) not null,
Constraint TechnicalSkillID_PK Primary Key(TechnicalSKillID))
```

Undergraduation_T

```
Create Table Undergraduation_T
(UEducationID int not null Check (UEducationID > 0),
NumberofYearsToCompleteUndergraduation int not null Check
(NumberofYearsToCompleteUndergraduation >= 3),
Constraint Undergraduation_PK Primary Key (UEducationID),
Constraint Undergraduation_FK Foreign key (UEducationID) References
Education_T(EducationID))
```

WorkProject_T

```
Create Table WorkProject_T
(WorkProjectID int not null Check (WorkProjectID > 0),
WorkProjectName nvarchar(max) not null,
WorkProjectDescription nvarchar(max) not null,
Constraint WorkProjectID_PK PRIMARY KEY (WorkProjectID))
```

WorkProjectContributors_T

```
Create Table WorkProjectContributors_T
(WorkProjectID int not null Check(WorkProjectID > 0),
ContributorID int not null Check (ContributorID > 0),
Constraint WorkProjectsContributors_PK Primary Key(WorkProjectID, ContributorID),
Constraint WorkProjectsContributors_FK1 Foreign Key(WorkProjectID) References
WorkProject_T(WorkProjectID),
Constraint WorkProjectsContributors_FK2 Foreign Key(ContributorID) References
Contributor_T(ContributorID))
```

WorkProjectsRoles_T

```
Create Table WorkProjectsRoles_T
(WorkProjectID int not null Check(WorkProjectID > 0),
RoleID int not null Check (ContributorID > 0),
Constraint WorkProjectsRoles_PK Primary Key(WorkProjectID, RoleID),
Constraint WorkProjectsRoles_FK1 Foreign Key(WorkProjectID) References
WorkProject_T(WorkProjectID),
Constraint WorkProjectsRoles_FK2 Foreign Key(RoleID) References Role_T(ContributorID))
```

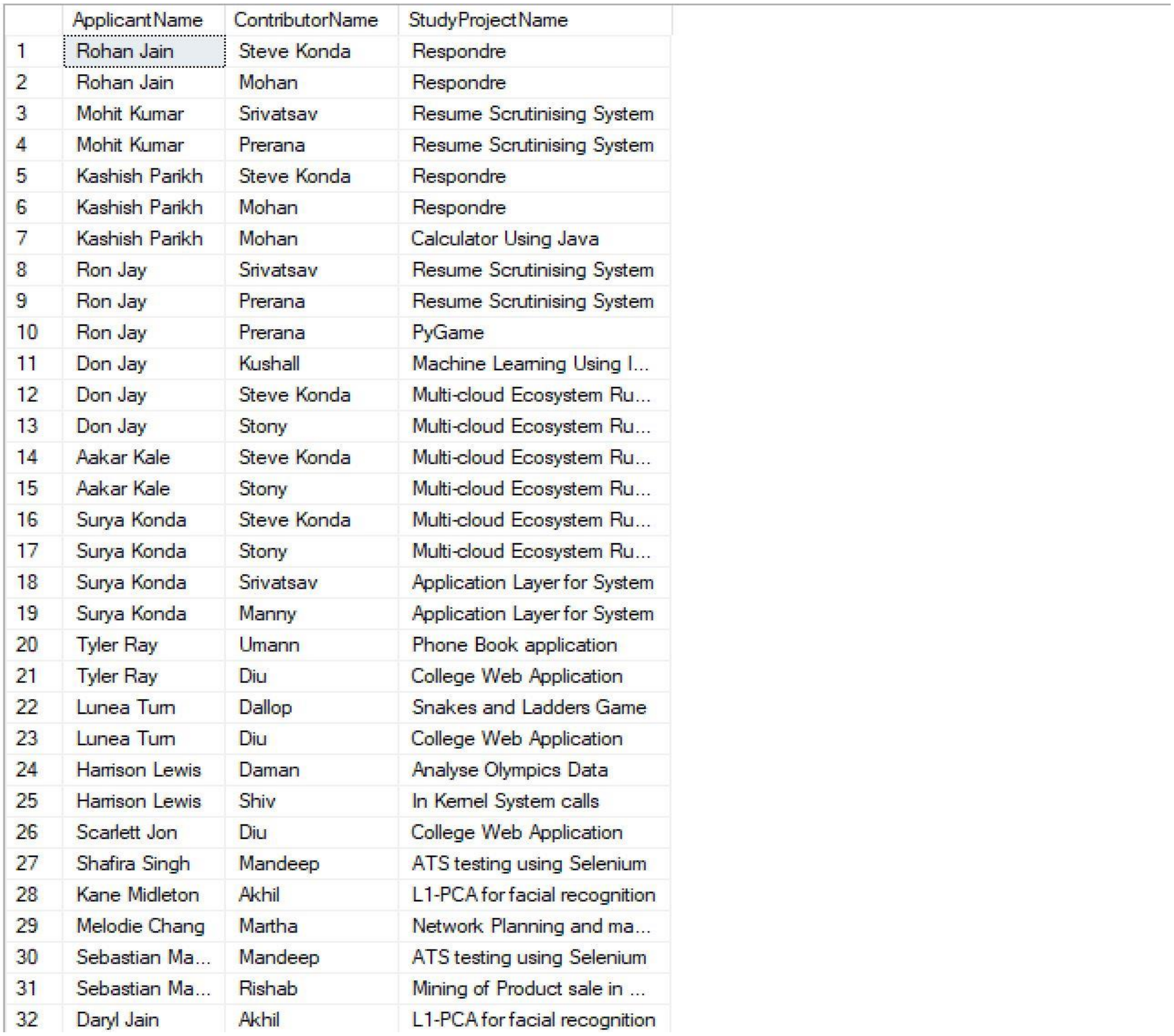
6.SQL STATEMENTS FOR VIEWS AND PROCEDURES

Materialized View 1: Applicant_Contributor_Project_View for Job Bridge

If an applicant is being shortlisted by a company as a prospective employee, then the company looks at the projects of the applicant and would want to know if he has really done the project, or it is a plagiarised study project. Many applicants usually plagiarise the projects and enter them in their resumes.

To avoid this Job Bridge keeps a track of the study projects that the applicant has done and the contributors for that project. If a new applicant enters into the database with the same project name, then Job Bridge compares the applicant's name with the contributors' list of names associated with that project. If the name of the new applicant is present in the contributors' list, then Job Bridge moves forward or else Job Bridge flags the applicant for further verification.

```
CREATE TABLE [dbo].[Applicant_Contributor_Project_view](
       [ApplicantName] [nvarchar](50) NULL,
       [ContributorName] [nvarchar](max) NOT NULL,
       [StudyProjectName] [nvarchar](max) NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO


create procedure refresh_Applicant_Contributor_Project_view as

delete from Applicant_Contributor_Project_view
insert into Applicant_Contributor_Project_view
SELECT Applicant_T.ApplicantName ,  Contributor_T.ContributorName,
StudyProject_T.StudyProjectName
FROM Applicant_T, StudyProject_T, Contributor_T, Education_T, ApplicantEducation_T,
StudyProjectsContributors_T
WHERE Applicant_T.ApplicantID = ApplicantEducation_T.ApplicantID and
ApplicantEducation_T.EducationID = Education_T.EducationID and
Education_T.EducationID = StudyProject_T.EducationID and StudyProject_T.StudyProjectID =
StudyProjectsContributors_T.StudyProjectID
and StudyProjectsContributors_T.ContributorID = Contributor_T.ContributorID

execute refresh_Applicant_Contributor_Project_view
SELECT * FROM Applicant_Contributor_Project_view
```



	ApplicantName	ContributorName	StudyProjectName
1	Rohan Jain	Steve Konda	Respondre
2	Rohan Jain	Mohan	Respondre
3	Mohit Kumar	Srivatsav	Resume Scrutinising System
4	Mohit Kumar	Prerana	Resume Scrutinising System
5	Kashish Parikh	Steve Konda	Respondre
6	Kashish Parikh	Mohan	Respondre
7	Kashish Parikh	Mohan	Calculator Using Java
8	Ron Jay	Srivatsav	Resume Scrutinising System
9	Ron Jay	Prerana	Resume Scrutinising System
10	Ron Jay	Prerana	PyGame
11	Don Jay	Kushall	Machine Learning Using I...
12	Don Jay	Steve Konda	Multi-cloud Ecosystem Ru...
13	Don Jay	Stony	Multi-cloud Ecosystem Ru...
14	Aakar Kale	Steve Konda	Multi-cloud Ecosystem Ru...
15	Aakar Kale	Stony	Multi-cloud Ecosystem Ru...
16	Surya Konda	Steve Konda	Multi-cloud Ecosystem Ru...
17	Surya Konda	Stony	Multi-cloud Ecosystem Ru...
18	Surya Konda	Srivatsav	Application Layer for System
19	Surya Konda	Manny	Application Layer for System
20	Tyler Ray	Umann	Phone Book application
21	Tyler Ray	Diu	College Web Application
22	Lunea Turn	Dallop	Snakes and Ladders Game
23	Lunea Turn	Diu	College Web Application
24	Harrison Lewis	Daman	Analyse Olympics Data
25	Harrison Lewis	Shiv	In Kernel System calls
26	Scarlett Jon	Diu	College Web Application
27	Shafira Singh	Mandeep	ATS testing using Selenium
28	Kane Midleton	Akhil	L1-PCA for facial recognition
29	Melodie Chang	Martha	Network Planning and ma...
30	Sebastian Ma...	Mandeep	ATS testing using Selenium
31	Sebastian Ma...	Rishab	Mining of Product sale in ...
32	Daryl Jain	Akhil	L1-PCA for facial recognition

Output Of View 1:

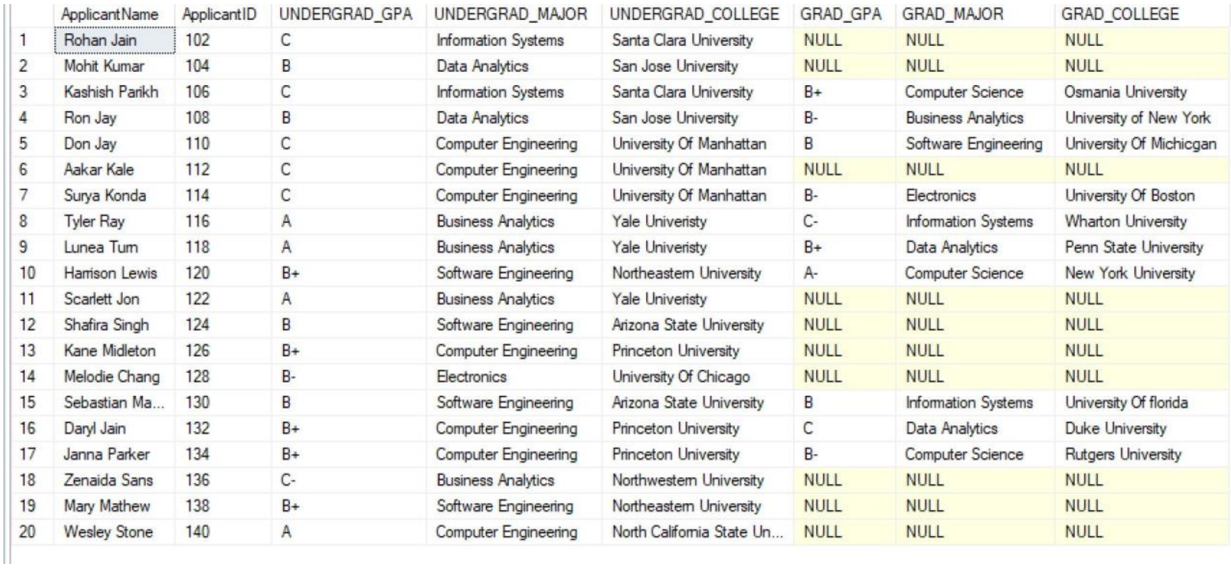
Materialized View 2: Applicant_Education_view for the hiring companies

If a company wants to take a look at an Applicants Education which would show his Undergraduae GPA, Undergraduate Major, Undergraduate College, Graduation GPA, Graduation Major, Graduation College to help them search for prospective candidates.

This view would be useful for them to get knowledge about the applicants' education.

```
CREATE TABLE [dbo].[applicant_Education_view](
       [ApplicantName] [nvarchar](50) NULL,
       [ApplicantID] [int] NOT NULL,
       [UNDERGRAD_GPA] [nvarchar](2) NULL,
       [UNDERGRAD_MAJOR] [varchar](25) NOT NULL,
       [UNDERGRAD_COLLEGE] [nvarchar](max) NOT NULL,
       [GRAD_GPA] [nvarchar](2) NULL,
       [GRAD_MAJOR] [varchar](25) NULL,
       [GRAD_COLLEGE] [nvarchar](max) NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

create or alter procedure refresh_Applicant_Education_view as
delete from applicant_Education_view
insert into applicant_Education_view
SELECT Applicant_T.ApplicantName,Applicant_T.ApplicantID, Education_T.GPA as
UNDERGRAD_GPA,  Major_T.MajorName as UNDERGRAD_MAJOR, College_T.CollegeName
as UNDERGRAD_COLLEGE, derivedtable.GPA as GRAD_GPA, derivedtable.MajorNAme as GRAD_MAJOR,
derivedtable.CollegeName AS GRAD_COLLEGE
FROM (Applicant_T left outer join (SELECT ApplicantEducation_T.ApplicantID,
Education_T.EducationID, Graduation_T.GEducationID, Education_T.GPA,  Major_T.MajorName,
College_T.CollegeName FROM Education_T, Major_T, College_T, ApplicantEducation_T,
graduation_T
WHERE ApplicantEducation_T.EducationID = Education_T.EducationID
and Education_T.CollegeID = College_T.CollegeID
and Education_T.MajorID = Major_T.MajorID
and Graduation_T.GEducationID = Education_T.EducationID ) as derivedtable on
Applicant_T.ApplicantID = derivedtable.ApplicantID ),
Education_T, Major_T, College_T, ApplicantEducation_T, Undergraduation_T
WHERE Applicant_T.ApplicantID = ApplicantEducation_T.ApplicantID and
ApplicantEducation_T.EducationID = Education_T.EducationID and Education_T.CollegeID =
College_T.CollegeID and Education_T.MajorID = Major_T.MajorID and
Undergraduation_T.UEducationID = Education_T.EducationID  ;

execute refresh_Applicant_Education_view

SELECT * FROM Applicant_Education_view
```

Output of View 2:



	ApplicantName	ApplicantID	UNDERGRAD_GPA	UNDERGRAD_MAJOR	UNDERGRAD_COLLEGE	GRAD_GPA	GRAD_MAJOR	GRAD_COLLEGE
1	Rohan Jain	102	C	Information Systems	Santa Clara University	NULL	NULL	NULL
2	Mohit Kumar	104	B	Data Analytics	San Jose University	NULL	NULL	NULL
3	Kashish Parikh	106	C	Information Systems	Santa Clara University	B+	Computer Science	Osmania University
4	Ron Jay	108	B	Data Analytics	San Jose University	B-	Business Analytics	University of New York
5	Don Jay	110	C	Computer Engineering	University Of Manhattan	B	Software Engineering	University Of Michicgan
6	Aakar Kale	112	C	Computer Engineering	University Of Manhattan	NULL	NULL	NULL
7	Surya Konda	114	C	Computer Engineering	University Of Manhattan	B-	Electronics	University Of Boston
8	Tyler Ray	116	A	Business Analytics	Yale Univeristy	C-	Information Systems	Wharton University
9	Lunea Turn	118	A	Business Analytics	Yale Univeristy	B+	Data Analytics	Penn State University
10	Harrison Lewis	120	B+	Software Engineering	Northeastern University	A-	Computer Science	New York University
11	Scarlett Jon	122	A	Business Analytics	Yale Univeristy	NULL	NULL	NULL
12	Shafira Singh	124	B	Software Engineering	Arizona State University	NULL	NULL	NULL
13	Kane Midleton	126	B+	Computer Engineering	Princeton University	NULL	NULL	NULL
14	Melodie Chang	128	B-	Electronics	University Of Chicago	NULL	NULL	NULL
15	Sebastian Ma...	130	B	Software Engineering	Arizona State University	B	Information Systems	University Of florida
16	Daryl Jain	132	B+	Computer Engineering	Princeton University	C	Data Analytics	Duke University
17	Janna Parker	134	B+	Computer Engineering	Princeton University	B-	Computer Science	Rutgers University
18	Zenaida Sans	136	C-	Business Analytics	Northwestern University	NULL	NULL	NULL
19	Mary Mathew	138	B+	Software Engineering	Northeastern University	NULL	NULL	NULL
20	Wesley Stone	140	A	Computer Engineering	North California State Un...	NULL	NULL	NULL

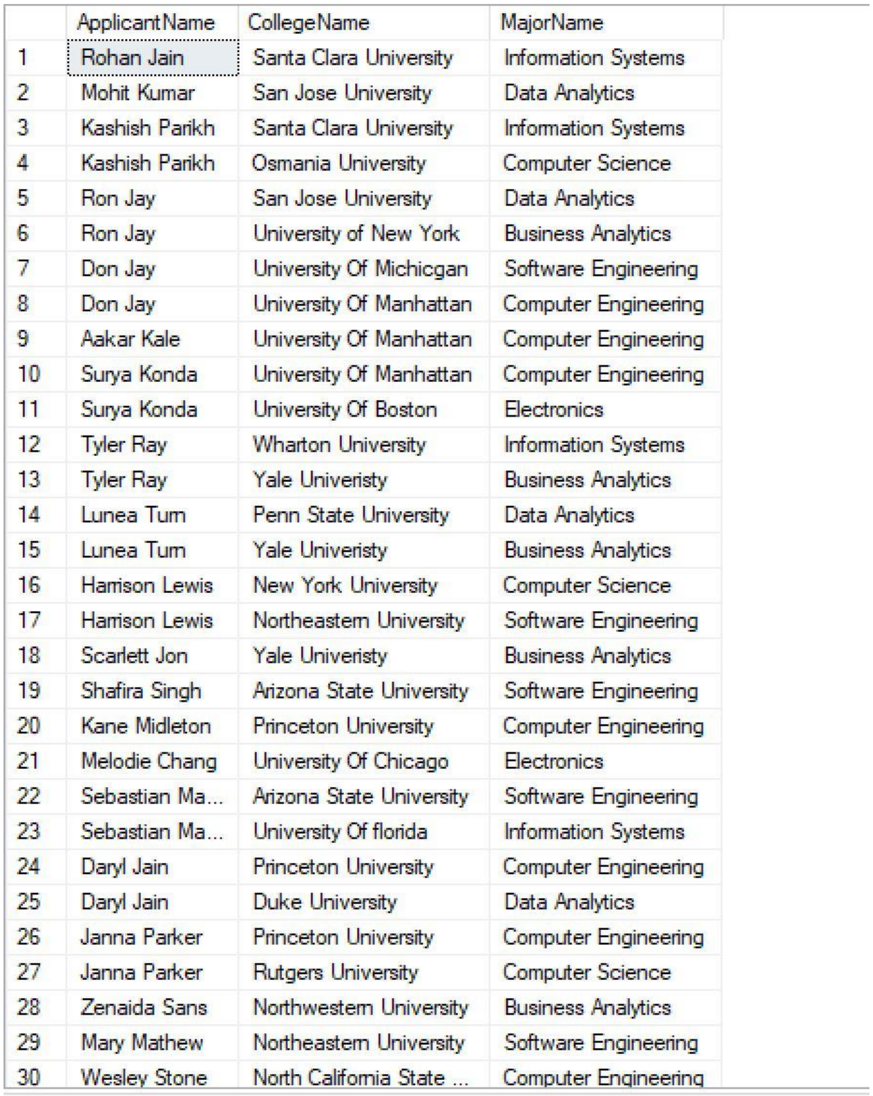
Materialized View 3: College_Alumni_View for Colleges

If a college wants to know about their alumni who are looking for a new job, to keep a track on them so that their other alumni can help them find a job quickly. This can be helpful to maintain a strong alumni network.

To help the colleges this view would be useful to show their alumni present in the Job Bridge data base.

```
CREATE TABLE [dbo].[College_Alumni_View](
       [ApplicantName] [nvarchar](50) NULL,
       [CollegeName] [nvarchar](max) NOT NULL,
       [MajorName] [varchar](25) NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

create procedure refresh_College_Alumni_view as
delete from College_Alumni_view
insert into College_Alumni_view
Select Applicant_T.ApplicantName, College_T.CollegeName, Major_T.MajorName
FROM ApplicantEducation_T, Education_T, Major_T, Applicant_T, College_T
WHERE Applicant_T.ApplicantID = ApplicantEducation_T.ApplicantID and
ApplicantEducation_T.EducationID = Education_T.EducationID
and Education_t.CollegeID = College_T.CollegeID and Education_T.MajorID = Major_T.MajorID

execute refresh_College_Alumni_view
select * FROM College_Alumni_view
```

Output Of View 3:



	ApplicantName	CollegeName	MajorName
1	Rohan Jain	Santa Clara University	Information Systems
2	Mohit Kumar	San Jose University	Data Analytics
3	Kashish Parikh	Santa Clara University	Information Systems
4	Kashish Parikh	Osmania University	Computer Science
5	Ron Jay	San Jose University	Data Analytics
6	Ron Jay	University of New York	Business Analytics
7	Don Jay	University Of Michicgan	Software Engineering
8	Don Jay	University Of Manhattan	Computer Engineering
9	Aakar Kale	University Of Manhattan	Computer Engineering
10	Surya Konda	University Of Manhattan	Computer Engineering
11	Surya Konda	University Of Boston	Electronics
12	Tyler Ray	Wharton University	Information Systems
13	Tyler Ray	Yale Univeristy	Business Analytics
14	Lunea Turn	Penn State University	Data Analytics
15	Lunea Turn	Yale Univeristy	Business Analytics
16	Harrison Lewis	New York University	Computer Science
17	Harrison Lewis	Northeastern University	Software Engineering
18	Scarlett Jon	Yale Univeristy	Business Analytics
19	Shafira Singh	Arizona State University	Software Engineering
20	Kane Midleton	Princeton University	Computer Engineering
21	Melodie Chang	University Of Chicago	Electronics
22	Sebastian Ma...	Arizona State University	Software Engineering
23	Sebastian Ma...	University Of florida	Information Systems
24	Daryl Jain	Princeton University	Computer Engineering
25	Daryl Jain	Duke University	Data Analytics
26	Janna Parker	Princeton University	Computer Engineering
27	Janna Parker	Rutgers University	Computer Science
28	Zenaida Sans	Northwestern University	Business Analytics
29	Mary Mathew	Northeastern University	Software Engineering
30	Wesley Stone	North California State ...	Computer Engineering

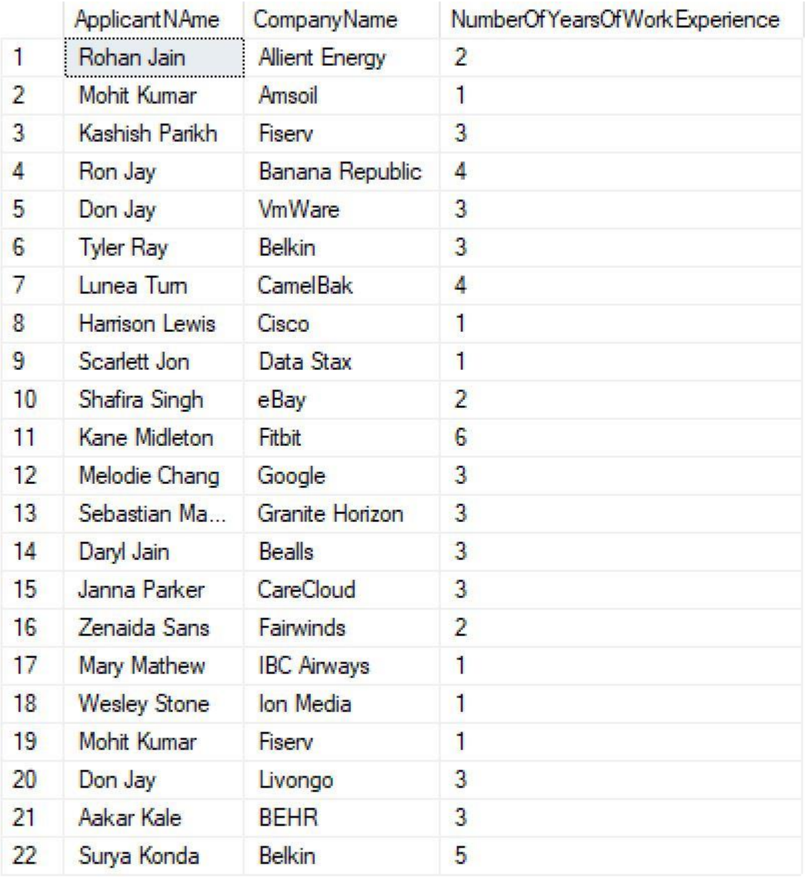
Materialized View 4: MaxWorkExperience_View for the hiring companies.

An Applicants previous work experience is very crucial for the new recruiters. The recruiters are usually interested in the applicants' number of years of experience in a role in a company.

To help the companies, this view shows them the company name and the maximum number of years of experience amongst all the work experience that the applicant has in his entire career.

```
CREATE TABLE [dbo].[MaxWorkExperince_view](
       [ApplicantNAme] [nvarchar](50) NULL,
       [CompanyName] [nvarchar](max) NOT NULL,
       [NumberOfYearsOfWorkExperience] [int] NOT NULL
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]
GO

create procedure refresh_MaxWorkExperince_view as

delete from MaxWorkExperince_view
insert into MaxWorkExperince_view
SELECT AppliCant_T.ApplicantNAme, Company_T.CompanyName, NumberOfYearsOfWorkExperience
FROM Applicant_T, ApplicantRoleDetails_T, Company_T,
( SELECT ApplicantRoleDetails_T.ApplicantID, max
(ApplicantRoleDetails_T.NumberOfYearsOfWorkExperience) as maxex FROM
ApplicantRoleDetails_T
  Group By ApplicantRoleDetails_T.ApplicantID) As maxexperience
  WHERE
Applicant_T.ApplicantID  = maxexperience.ApplicantID And Applicant_T.ApplicantID=
ApplicantRoleDetails_T.ApplicantID
 and Company_T.CompanyID = ApplicantRoleDetails_T.CompanyID
 and NumberOfYearsOfWorkExperience = maxex

execute refresh_MaxWorkExperince_view

SELECT * FROM MaxWorkExperince_view
```

Output Of View 4:



	ApplicantNAme	CompanyName	NumberOfYearsOfWorkExperience
1	Rohan Jain	Allient Energy	2
2	Mohit Kumar	Amsoil	1
3	Kashish Parikh	Fiserv	3
4	Ron Jay	Banana Republic	4
5	Don Jay	VmWare	3
6	Tyler Ray	Belkin	3
7	Lunea Turn	CamelBak	4
8	Harrison Lewis	Cisco	1
9	Scarlett Jon	Data Stax	1
10	Shafira Singh	eBay	2
11	Kane Midleton	Fitbit	6
12	Melodie Chang	Google	3
13	Sebastian Ma...	Granite Horizon	3
14	Daryl Jain	Bealls	3
15	Janna Parker	CareCloud	3
16	Zenaida Sans	Fairwinds	2
17	Mary Mathew	IBC Airways	1
18	Wesley Stone	Ion Media	1
19	Mohit Kumar	Fiserv	1
20	Don Jay	Livongo	3
21	Aakar Kale	BEHR	3
22	Surya Konda	Belkin	5

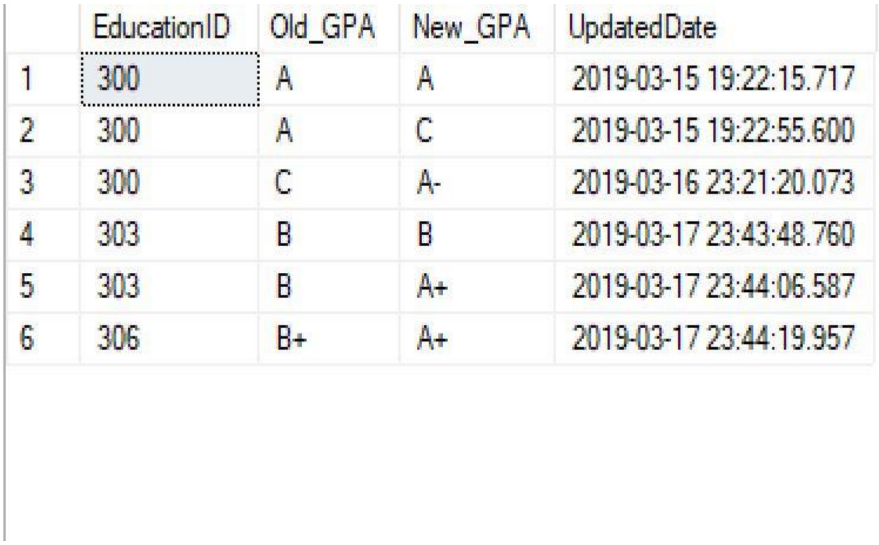
7. SQL STATEMENTS FOR DATABASE TRIGGERS

This trigger is created to store the updated GPA(Grade Point Average) of an Education so that no one can change the GPA for a malicious purpose. This trigger would also keep a record of the previous GPA, the new GPA and also the last updated date/time.

```
CREATE TABLE Education_GPA_Change_Log
(EducationID int,
Old_GPA nvarchar(2),New_GPA nvarchar(2),UpdatedDate datetime)
```

```
create trigger GPA_ChangeLog on Education_T for update
as
if
update(GPA)
begin insert into Education_GPA_Change_Log (EducationID, Old_GPA, New_GPA, UpdatedDate)
select inserted.EducationID,  deleted.GPA, inserted.GPA, GETDATE()
from inserted, deleted where inserted.EducationID = deleted.EducationID
end


Update Education_T SET GPA = 'A+' WHERE EducationID = 306
SELECT * FROM Education_GPA_Change_Log
```

Output of the trigger:



	EducationID	Old_GPA	New_GPA	UpdatedDate
1	300	A	A	2019-03-15 19:22:15.717
2	300	A	C	2019-03-15 19:22:55.600
3	300	C	A-	2019-03-16 23:21:20.073
4	303	B	B	2019-03-17 23:43:48.760
5	303	B	A+	2019-03-17 23:44:06.587
6	306	B+	A+	2019-03-17 23:44:19.957