

# Job Portal Database System

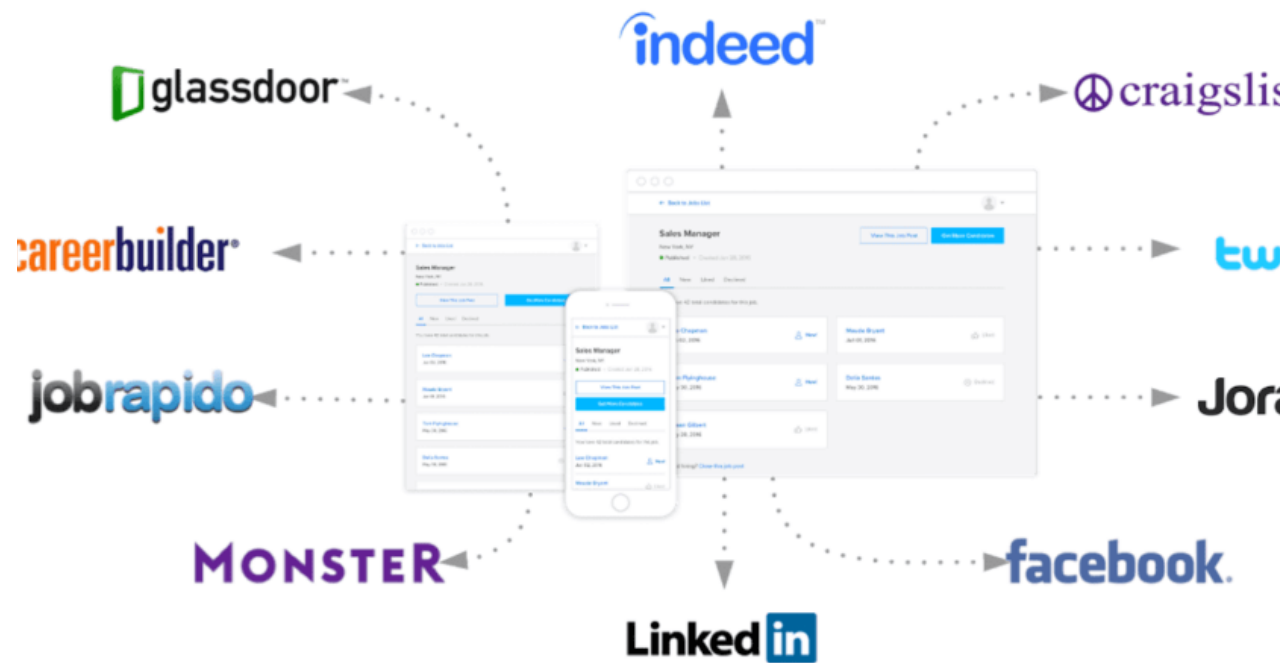
INFO6210 – Database Management and Database Design

Project Team-6

Aruna Divi | Alivia Guin | Arundathi Balangowda Patil |  
Harika Reddy Gurram



# Objective



- *About the topic*

Job portals are integral part of almost every hiring process and using them effectively will translate into qualified candidates

- *Goal*

The purpose of the database is to serve jobseekers to find available job vacancies and employers to identify eligible candidates for the desired position

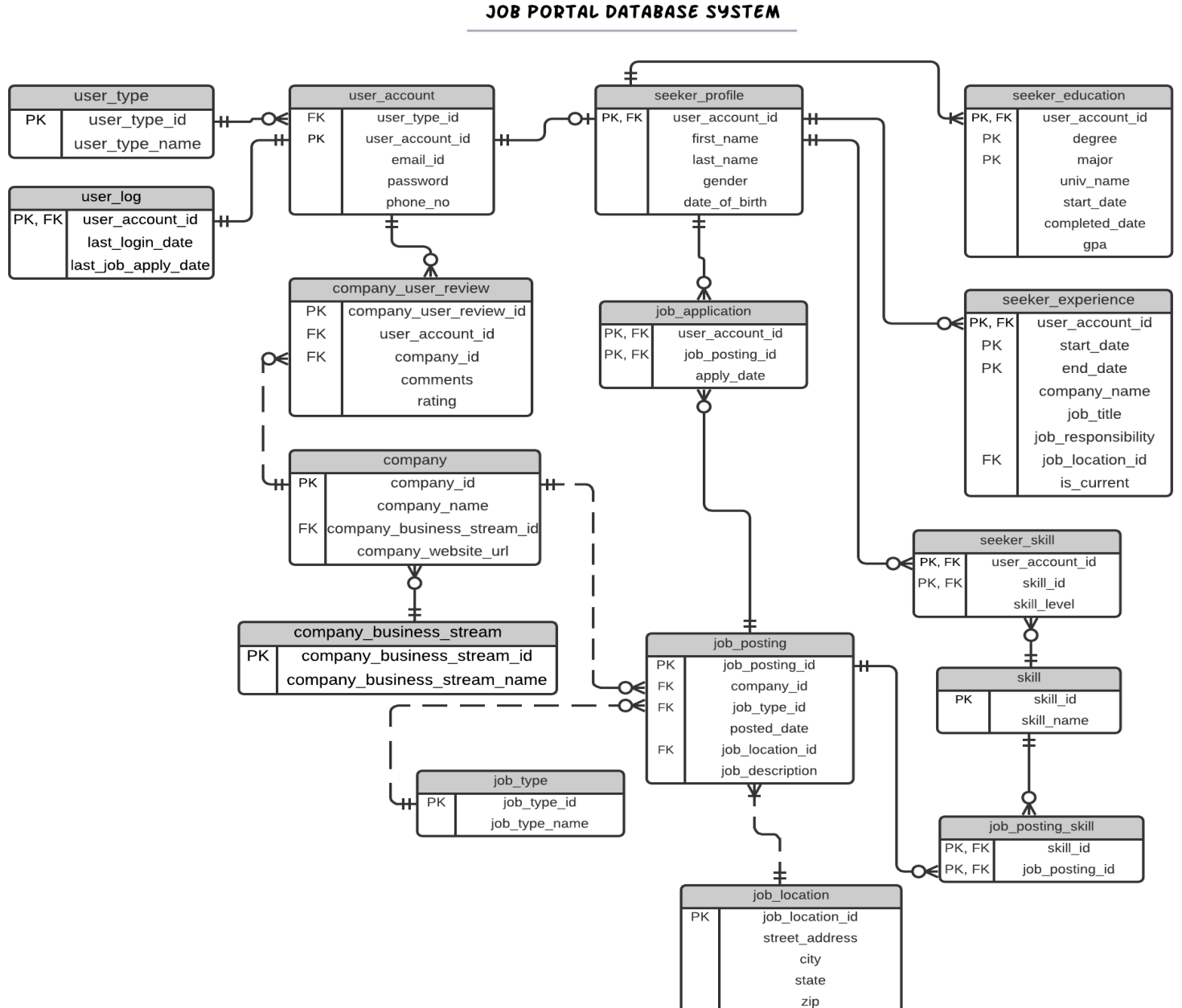


## *Business Problems Addressed*

- Allows jobseekers to find the postings
- Permits recruiters to post the job
- Feedback to the companies by job seekers



# Entity-Relationship Diagram



# Implementation

```
CREATE TABLE Person.UserType  
(  
    UserTypeID int PRIMARY KEY NOT NULL,  
    UserTypeName varchar(255) NOT NULL  
);
```

```
ALTER TABLE Person.UserAccount  
ADD CONSTRAINT unique_emailId UNIQUE (EmailID);  
GO
```

```
INSERT INTO Company.CompanyBusinessStream VALUES (1001, 'IT'),  
(1002, 'Construction'),  
(1003, 'Marketing'),  
(1004, 'Media'),  
(1005, 'Manufacturing'),  
(1006, 'Business'),  
(1007, 'Banking'),  
(1008, 'Automotive'),  
(1009, 'Sales'),  
(1010, 'Entertainment')
```

DDL

Computed columns

Functions

Triggers

# Encryption

```
/*Create Password protected Master key*/
CREATE MASTER KEY
    ENCRYPTION BY PASSWORD = 'TEAM_6_Password';

/*Create certificate to protect symetric key*/
CREATE CERTIFICATE TeamSixCertificate
    WITH SUBJECT = 'Team 6 Certificate',
    EXPIRY_DATE = '2026-10-31';

/*Create symmetric key to encrypt data*/
CREATE SYMMETRIC KEY TeamSixSymmetricKey
    WITH ALGORITHM = AES_128
    ENCRYPTION BY CERTIFICATE TeamSixCertificate;

/* Open symmetric key */
OPEN SYMMETRIC KEY TeamSixSymmetricKey
    DECRYPTION BY CERTIFICATE TeamSixCertificate;
```

Password
0x002FF40B9B5A594188FAA8B5839D5EFC0200000014411AC...
0x002FF40B9B5A594188FAA8B5839D5EFC0200000041A2827...
0x002FF40B9B5A594188FAA8B5839D5EFC02000000CF9E28...
0x002FF40B9B5A594188FAA8B5839D5EFC020000003EFA01F...
0x002FF40B9B5A594188FAA8B5839D5EFC020000000602851...
0x002FF40B9B5A594188FAA8B5839D5EFC02000000C18FAA...
0x002FF40B9B5A594188FAA8B5839D5EFC02000000FFF2CE...
0x002FF40B9B5A594188FAA8B5839D5EFC02000000E90AB34...
0x002FF40B9B5A594188FAA8B5839D5EFC02000000C90CD0...
0x002FF40B9B5A594188FAA8B5839D5EFC02000000723712B...

# Functions- Table level Constraint

```
CREATE FUNCTION [dbo].[udf_CheckPhoneNumberFormat]
(@strAlphaNumeric VARCHAR(14))
RETURNS smallint
AS
BEGIN
    DECLARE @intAlpha INT
    DECLARE @tempNum VARCHAR(14)
    DECLARE @result smallint
    SET @intAlpha = PATINDEX('%[^0-9]%', @strAlphaNumeric)
    SET @tempNum = @strAlphaNumeric
    SET @result = 0
    BEGIN
    WHILE @intAlpha > 0
        BEGIN
            SET @tempNum = STUFF(@tempNum, @intAlpha, 1, '' )
            SET @intAlpha = PATINDEX('%[^0-9]%', @tempNum )
        END
    SET @tempNum = '(' + STUFF(STUFF(@tempNum,7,0,'-'),4,0,') ' )
    IF @tempNum = @strAlphaNumeric
        BEGIN
            SET @result = 1
        END
    END
    RETURN @result
END
GO
```

PhoneNumber
(123) 123-9876
(123) 245-7685
(123) 678-9012
(123) 876-3846
(123) 287-1267
(123) 123-9876
(123) 369-9876
(123) 211-9876
(123) 221-9876
(123) 231-5454

# Computed Columns

## Jobs Applied

```
CREATE FUNCTION CountJobsApplied (@UserAccount int)
    RETURNS int
    AS
    BEGIN
        DECLARE @count int
        SELECT @count = COUNT(j.JobPostingID)
        FROM Job.JobApplication j
        WHERE j.UserAccountID = @UserAccount
        GROUP BY UserAccountID
        SET @count = ISNULL(@count,0)
        RETURN @count
    END
GO
```

```
ALTER TABLE Seeker.SeekerProfile ADD JobsAppliedCount AS
dbo.CountJobsApplied(UserAccountID)
```

## Current Job

```
ALTER TABLE Seeker.SeekerExperience
ADD IsCurrent AS
CASE
    WHEN (EndDate IS NULL) THEN 'TRUE'
    ELSE 'FALSE'
END
```

## Age

```
ALTER TABLE Seeker.SeekerProfile
ADD Age AS DATEDIFF(hour,DateOfBirth,GETDATE())/8766;
```



# Triggers

## Trigger – I

```
CREATE TRIGGER utrAddUserEntryInUserLogAfterUserCreation
ON Person.UserAccount
AFTER INSERT
AS
BEGIN
DECLARE @userAccountID INT;
SET @userAccountID = (SELECT UserAccountID FROM Inserted);
INSERT INTO Person.UserLog VALUES (@userAccountID,
GETDATE(), NULL)
END
GO
```

## Trigger – II

```
CREATE TRIGGER utrLASTJOBAPPLIEDDATE
ON Job.JobApplication
AFTER INSERT
AS
BEGIN
DECLARE @userAccountID INT;
SET @userAccountID = (SELECT UserAccountID FROM Inserted);
UPDATE Person.UserLog SET LastJobApplyDate = GETDATE()
WHERE UserAccountID = @userAccountID
END
GO
```

user_account	
FK	user_type_id
PK	user_account_id
	email_id
	password
	phone_no

user_log	
PK, FK	user_account_id
	last_login_date
	last_job_apply_date

Job_Application	
PK, FK	user_account_id
PK, FK	job_posting_id
	apply_date

# Views

## View – I

```
CREATE VIEW vwSDEApplication
AS
SELECT sp.FirstName, sp.LastName, ua.EmailID, c.CompanyID,
c.CompanyName, ua.PhoneNumber FROM Job.JobApplication ja
INNER JOIN Job.JobPosting jp ON ja.JobPostingID = jp.JobPostingID
INNER JOIN Company.Company c ON jp.CompanyID = c.CompanyID
INNER JOIN Seeker.SeekerProfile sp ON sp.UserAccountID =
ja.UserAccountID
INNER JOIN Person.UserAccount ua ON ua.UserAccountID =
ja.UserAccountID
WHERE jp.JobTypeID = 1 and jp.CompanyID = 123
GO
```

## View – II

```
CREATE VIEW vwTotalJobPostingsByCompanyView
AS
SELECT c.CompanyName, COUNT(jp.JobPostingID) AS
TotalJobPosting
FROM Company.Company c
INNER JOIN Job.JobPosting jp ON
c.CompanyID = jp.CompanyID
GROUP BY c.CompanyID, c.CompanyName
GO
```

Results		Messages				
	FirstName	LastName	EmailID	CompanyID	CompanyName	PhoneNumber
1	Kevin	Peterson	abc@gmail.com	123	Microsoft	(123) 123-9876
2	Chris	Campbell	pqr@gmail.com	123	Microsoft	(123) 245-7685
3	Katie	Ricker	mno@gmail.com	123	Microsoft	(123) 876-3846
4	Sandeep	Satyala	bnm@gmail.com	123	Microsoft	(123) 287-1267

Results		Messages	
	CompanyName	TotalJobPosting	
1	Microsoft	3	
2	Facebook	3	
3	Twitter	3	
4	Lane	1	

# Views Continued..

## View – III

```
CREATE VIEW userRating
```

```
AS
```

```
SELECT ua.UserAccountID, CompanyName, ur.Comments, ur.Rating
```

```
FROM Person.UserAccount ua
```

```
INNER JOIN Company.CompanyUserReview ur ON ur.UserAccountID = ua.UserAccountID
```

```
INNER JOIN Company.Company cc ON cc.CompanyID = ur.CompanyID
```

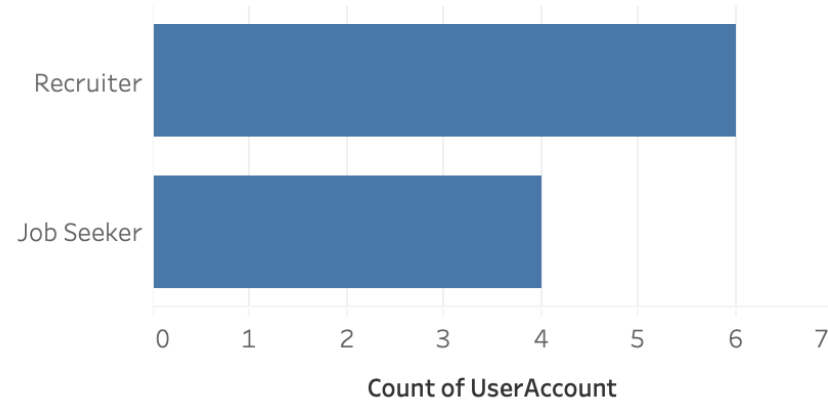
```
GO
```

Results Messages				
	UserAccountID	CompanyName	Comments	Rating
1	9	Tiktok		4.00
2	1	Facebook	Very Responsive and I higly reccommend one to ap...	5.00
3	2	Twitter	Pay scale is good	4.67
4	1	Lane	Recruiters are really helpful	4.90
5	3	Filmy	I do not reccommend this	1.00
6	4	Tiktok	good	4.00
7	5	Facebook		4.80
8	2	Tesla	Reccommended	4.00
9	1	Intel	good company	4.90
10	8	Ahold	Not bad	3.50

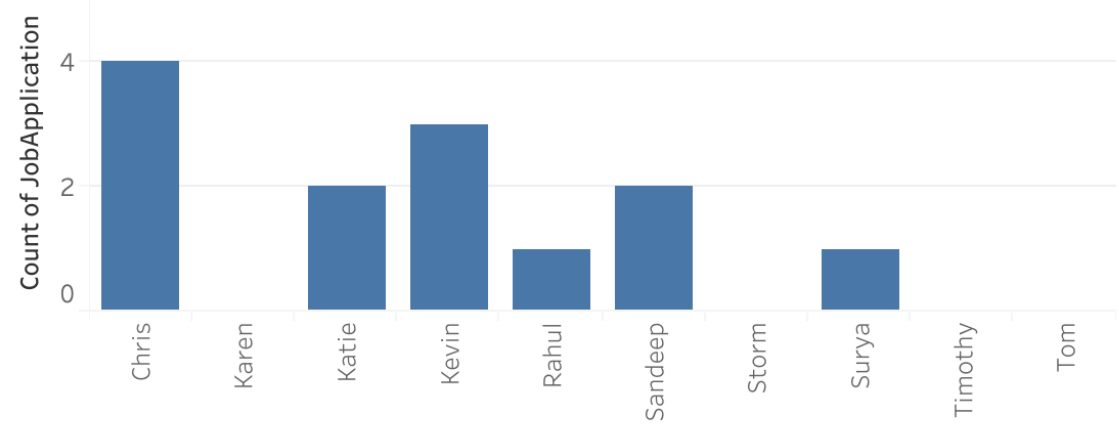


Data Analysis

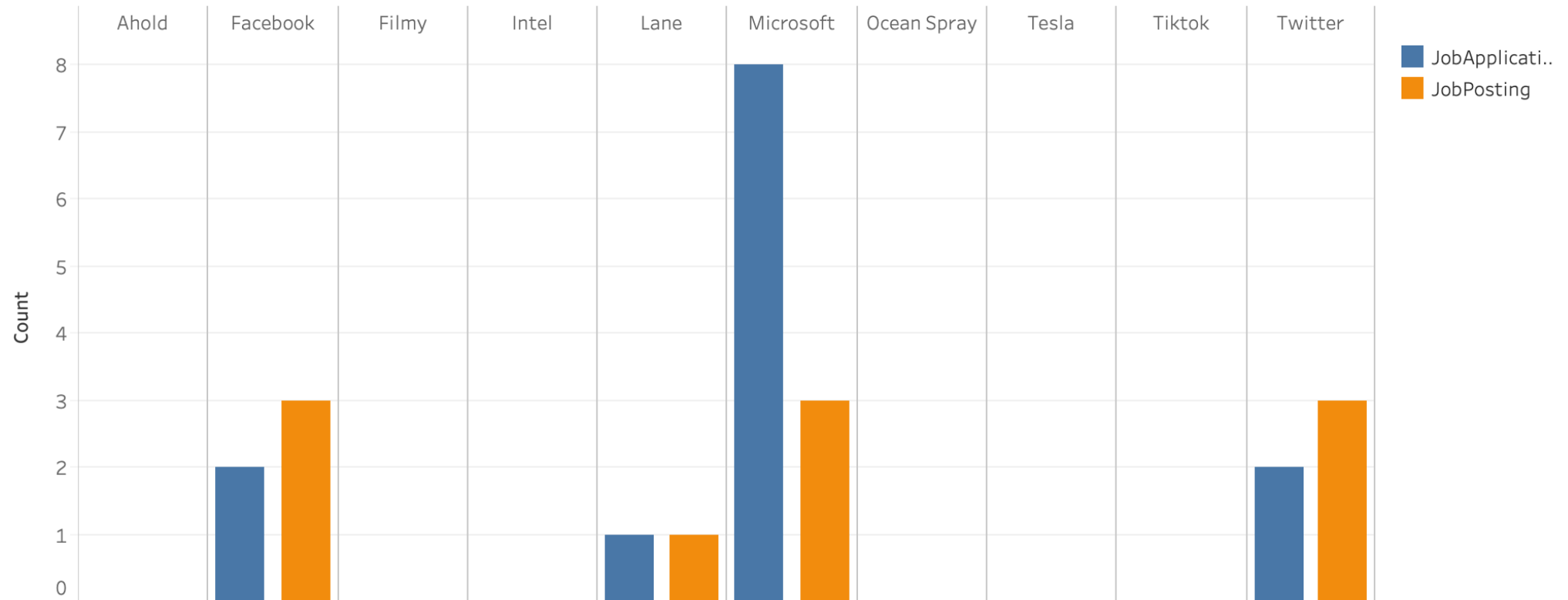
Number of user accounts by type



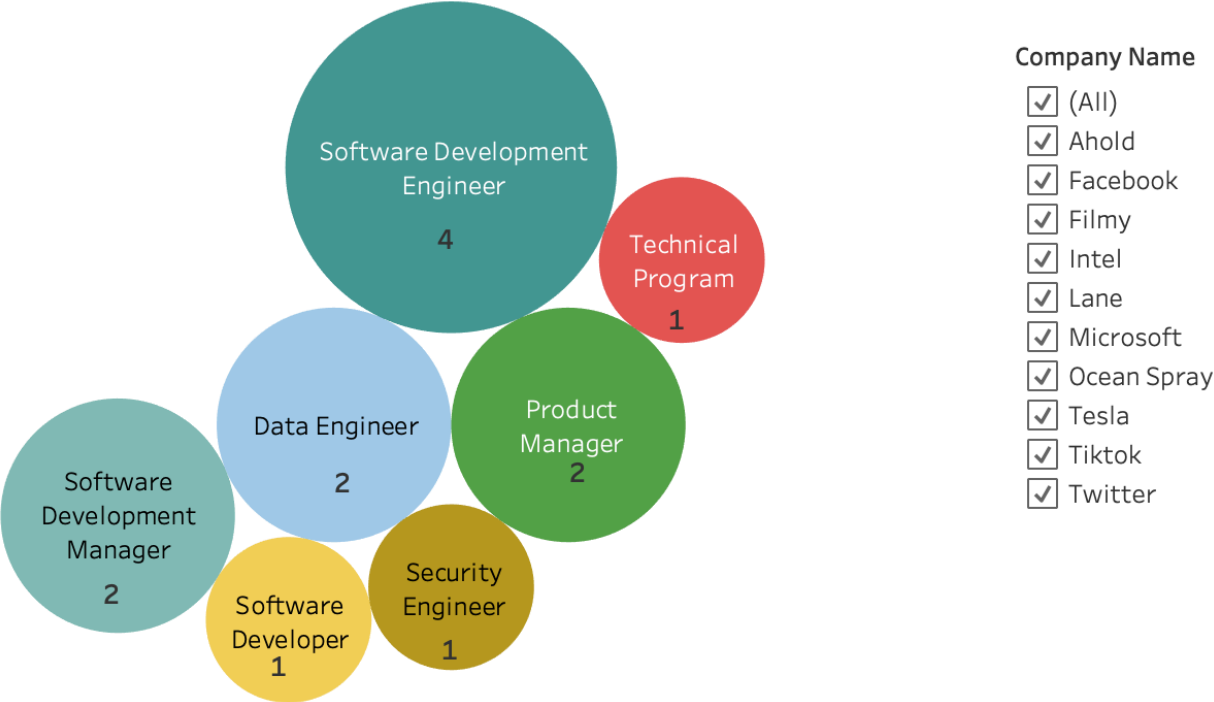
Number of job applications by seeker



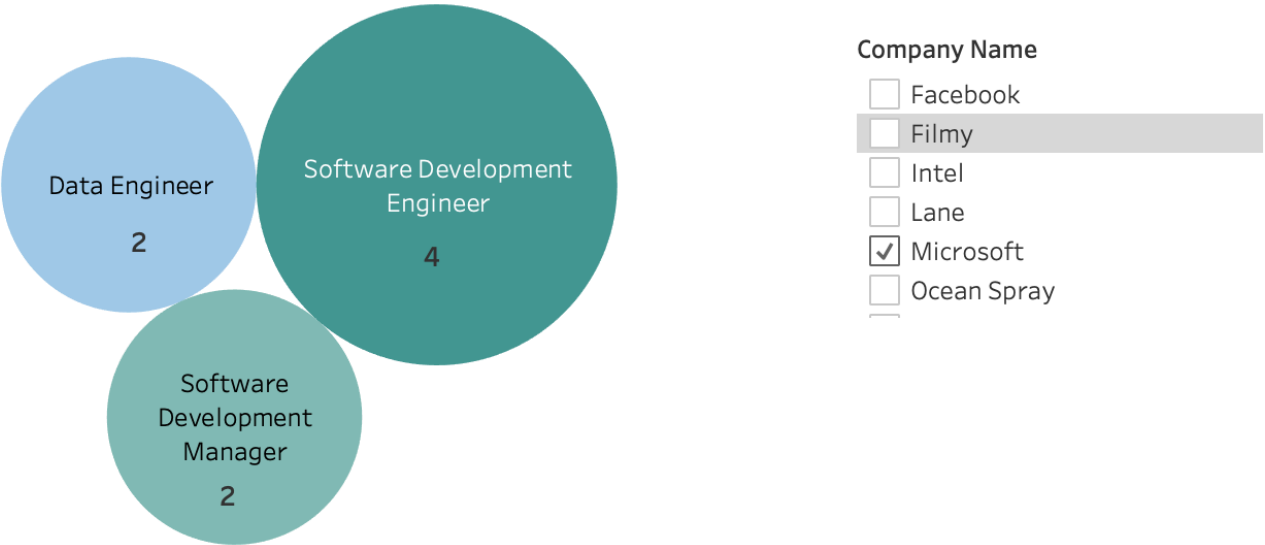
Number of job postings and applications by companies



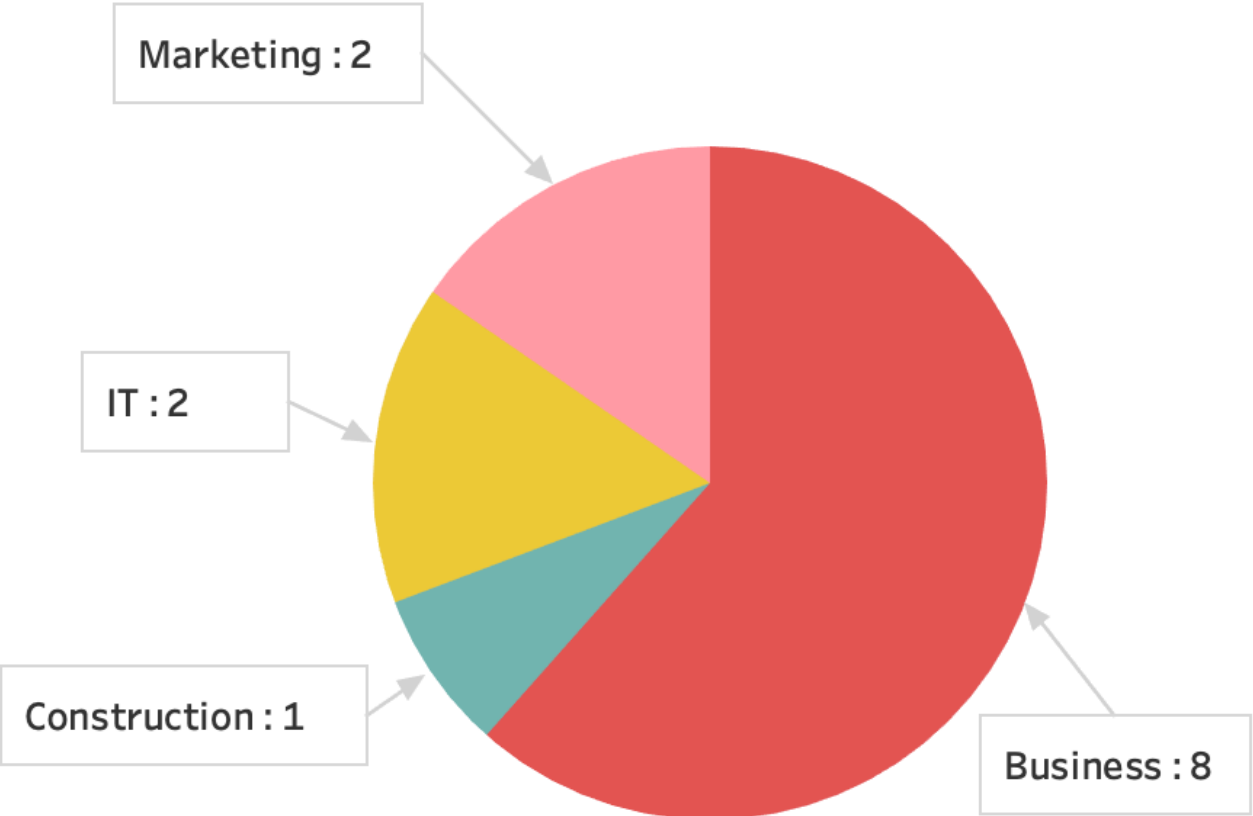
Job applications by type for each company



Job applications for Microsoft



# Job applications by business stream



## Company Business Stre..

- Automotive
- Banking
- Business
- Construction
- Entertainment
- IT
- Manufacturing
- Marketing
- Sales
- Media

## Company Name

- ☒ (All)
- ☒ Ahold
- ☒ Facebook
- ☒ Filmy
- ☒ Intel
- ☒ Lane
- ☒ Microsoft
- ☒ Ocean Spray
- ☒ Tesla
- ☒ Tiktok
- ☒ Twitter

Thank you!

