Northeastern University

Final Project Report

Module-6

[Identification Of Appropriate Skillset For Project]

[Kalyan Kumar Bhogi]

1. Introduction:

The idea for this project came from my work experience in India, where I worked as a Program Analyst at TCL-Electronics for three years. During my time at the company, I was awarded the Best Employee of the Year in both 2021 and 2022. As a result, I was assigned the role of Project Trainer for three projects. As a Project Trainer, my responsibilities included providing Knowledge Transfer sessions to new employees and evaluating their performances. However, I observed that many employees were not able to perform well due to a lack of appropriate skill sets. Despite this, we had to allocate them to projects due to high demand. Based on this experience, I decided to focus my final project on the "**Identification Of Appropriate Skillset For Project**"**.** By doing so, organizations can ensure that the right employees are assigned to the right projects, which can improve project performance and reduce costs associated with training and project delays.

➢ Project Overview:

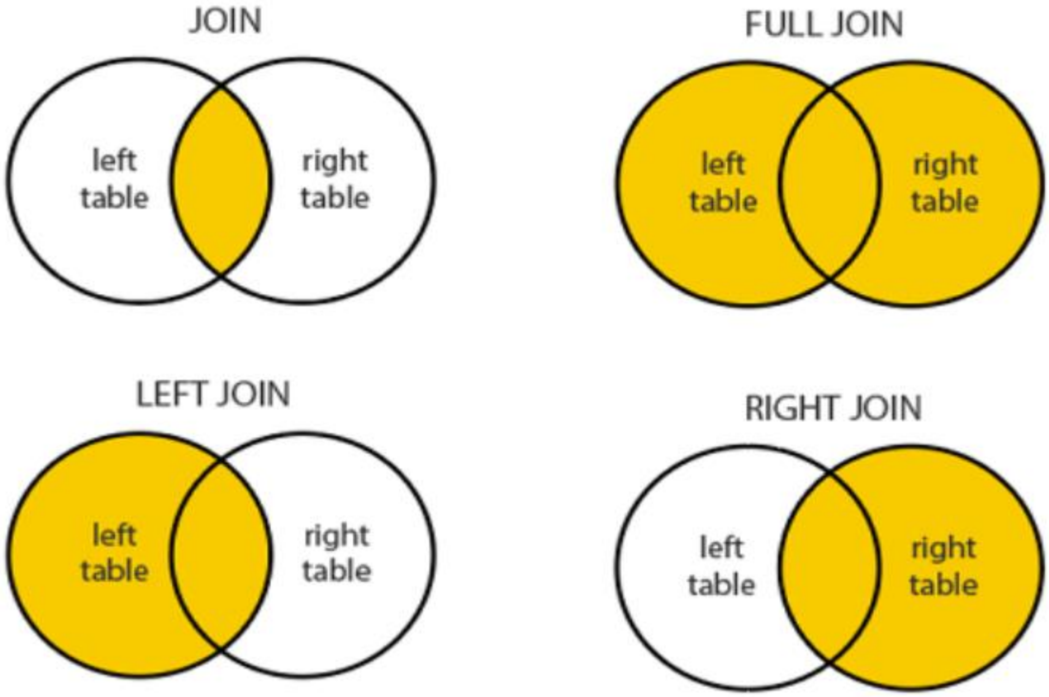
Generally allocating the right kind of skill set is very important for any project. An organization should recruit the right kind of skill set for the project if that is not happening then the project must need to bare many losses such as Training costs, reaching project closing time, etc. To avoid these types of problems Organizations must need to know the different types of skill set available for employees for the different types of upcoming projects in the organization. So here an organization wants to segregate the employees based on their skill set so that the organization can allocate them according to the project demands & requirements.

➢ Use Case:

This Project would be useful to the project management team, Scrum master, and Project Manager for effecting planning the project For Example, if a project comes with multi-platform technology, then having a database of employees with all the skill set available would be of great benefit. As it is easy to query with the appropriate requirement of the skill set required for the project. Then, we need to use the "JOINS" query to get appropriate data from Database. It would be of huge benefit to the organization. As it reduces the training cost for the training of employees and also decreases the time required in the planning phase in the software development life cycle(SDLC).As more time can be allocated to the design and analysis of the project. So, the project development phase can be given more preference. So, the project can be delivered efficiently.

2. Business Analysis:

Generally, In an organization, there would be different kinds of project with requires different skill sets. In order, to perfectly accommodate the employees of the organization it requires an SQL querying system for the scrum master and project manager. Moreover, having such an SQL query system which will help in populating employees with different combinations of skill sets. There are different kinds of queries available in SQL to help with use-case scenarios.



From the Image above, There are different kinds of joins available in SQL namely left join, right join, join, and Full Join or Union. Depending, On the scenario of the project requirements.

3. Personas:

- Recruitment Team
  - 20 Users
  - Recruitment of right skillset for projects

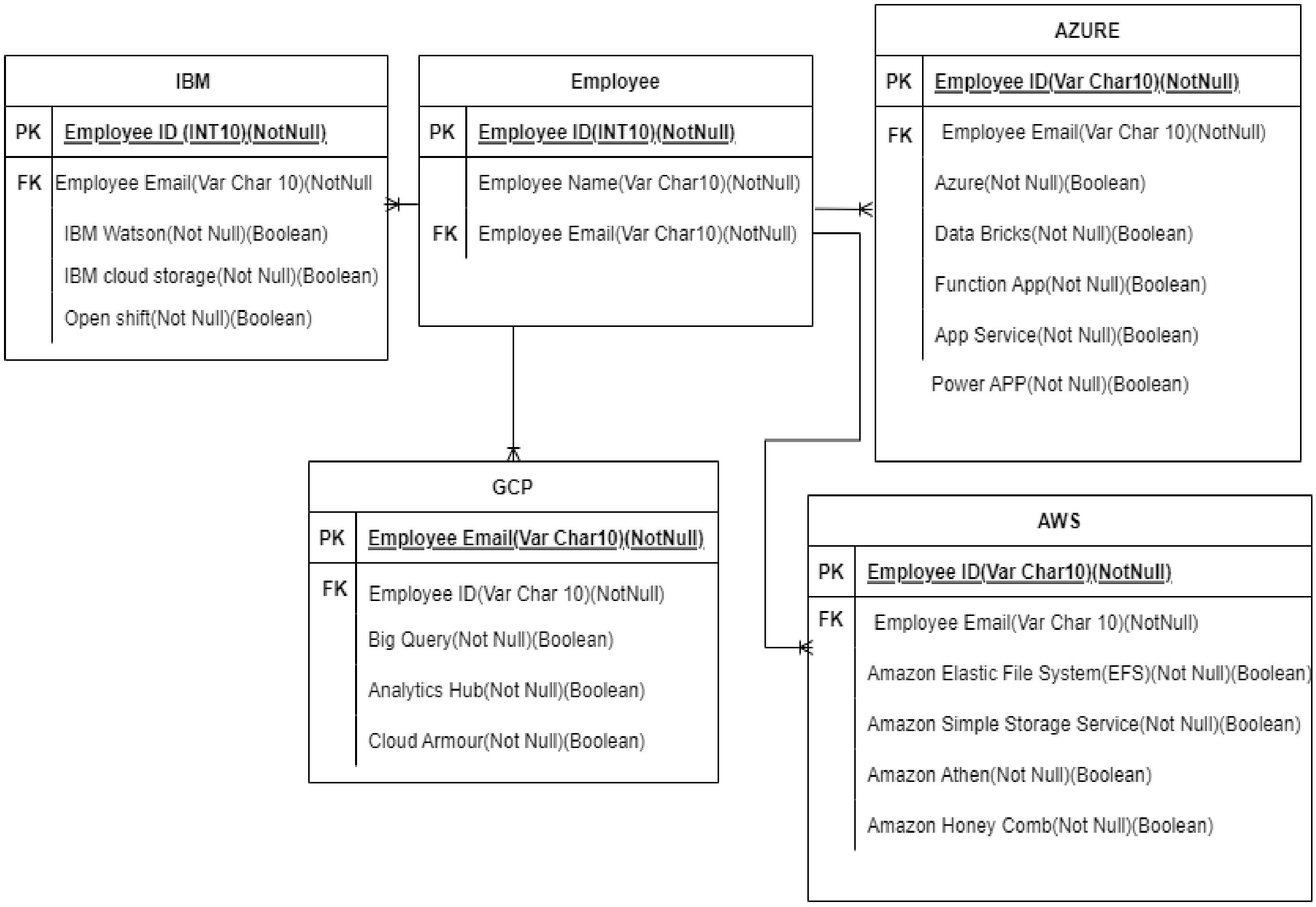
- Project Manager and Scrum Master

  - 10 Users
  - Allocation of correct skillset for correct Project
- Regional Managers
  - 10 Users
  - To access the recently recruited employee data in their company.

- Training Team
  - 5 Users
  - To get to know the details of the people and to check the requirement of the training for them.

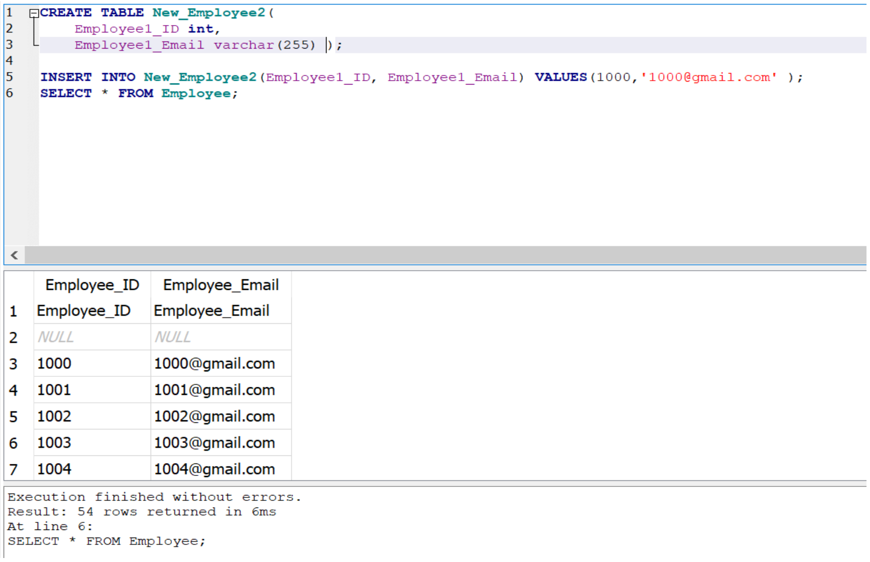
4. Table Design & Analysis:

- ER Diagram:
  - My project has 5 data entities(Tables). Those are Employee Table, IBM Table, AWS Table, GCP Table, AZURE Table.
  - Employee entity is in **one-to-many** relationship with all the entities.


IBM
PK
Employee ID (INT10)(NotNull)
FK
Employee Email(Var Char 10)(NotNull
IBM Watson(Not Null)(Boolean)
IBM cloud storage(Not Null)(Boolean)
Open shift(Not Null)(Boolean)
Employee
PK
Employee ID(INT10)(NotNull)
Employee Name(Var Char10)(NotNull)
FK
Employee Email(Var Char10)(NotNull)
AZURE
PK
Employee ID(Var Char10)(NotNull)
FK
Employee Email(Var Char 10)(NotNull)
Azure(Not Null)(Boolean)
Data Bricks(Not Null)(Boolean)
Function App(Not Null)(Boolean)
App Service(Not Null)(Boolean)
Power APP(Not Null)(Boolean)
GCP
PK
Employee Email(Var Char10)(NotNull)
FK
Employee ID(Var Char 10)(NotNull)
Big Query(Not Null)(Boolean)
Analytics Hub(Not Null)(Boolean)
Cloud Armour(Not Null)(Boolean)
AWS
PK
Employee ID(Var Char10)(NotNull)
FK
Employee Email(Var Char 10)(NotNull)
Amazon Elastic File System(EFS)(Not Null)(Boolean)
Amazon Simple Storage Service(Not Null)(Boolean)
Amazon Athen(Not Null)(Boolean)
Amazon Honey Comb(Not Null)(Boolean)

➢ Table Design Queries:

- Employee Table

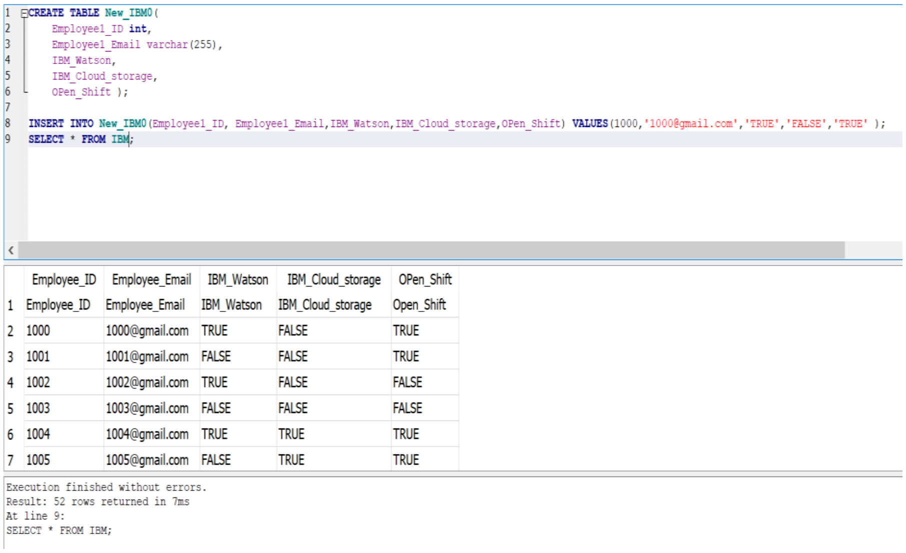


```
CREATE TABLE New_Employee2(
    Employee1_ID int,
    Employee1_Email varchar(255) );

INSERT INTO New_Employee2(Employee1_ID, Employee1_Email) VALUES(1000,'1000@gmail.com' );
SELECT * FROM Employee;
```

	Employee_ID	Employee_Email
1	Employee_ID	Employee_Email
2	NULL	NULL
3	1000	1000@gmail.com
4	1001	1001@gmail.com
5	1002	1002@gmail.com
6	1003	1003@gmail.com
7	1004	1004@gmail.com

```
Execution finished without errors.
Result: 54 rows returned in 6ms
At line 6:
SELECT * FROM Employee;
```

- IBM Table

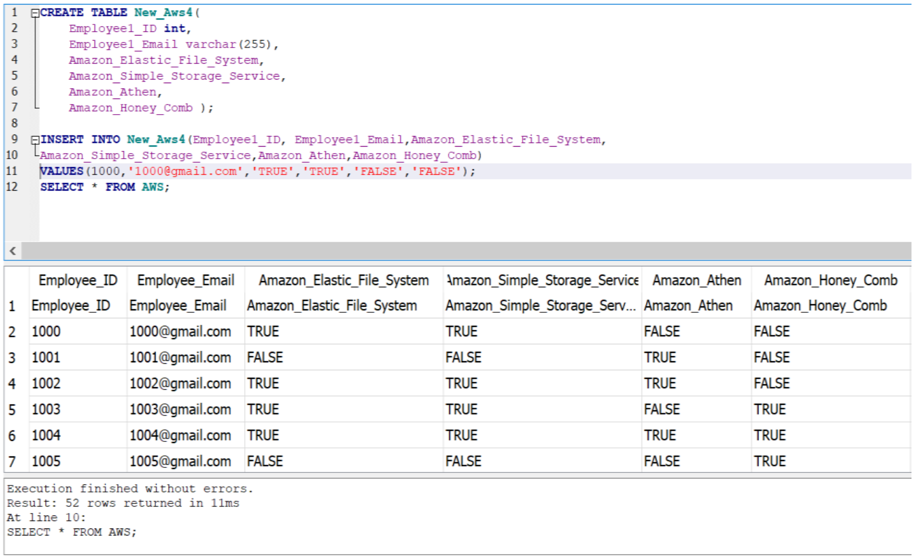


```
CREATE TABLE New_IBM0(
    Employee1_ID int,
    Employee1_Email varchar(255),
    IBM_Watson,
    IBM_Cloud_storage,
    OPen_Shift );

INSERT INTO New_IBM0(Employee1_ID, Employee1_Email,IBM_Watson,IBM_Cloud_storage,OPen_Shift) VALUES(1000,'1000@gmail.com','TRUE','FALSE','TRUE' );
SELECT * FROM IBM;
```

	Employee_ID	Employee_Email	IBM_Watson	IBM_Cloud_storage	OPen_Shift
1	Employee_ID	Employee_Email	IBM_Watson	IBM_Cloud_storage	Open_Shift
2	1000	1000@gmail.com	TRUE	FALSE	TRUE
3	1001	1001@gmail.com	FALSE	FALSE	TRUE
4	1002	1002@gmail.com	TRUE	FALSE	FALSE
5	1003	1003@gmail.com	FALSE	FALSE	FALSE
6	1004	1004@gmail.com	TRUE	TRUE	TRUE
7	1005	1005@gmail.com	FALSE	TRUE	TRUE

```
Execution finished without errors.
Result: 52 rows returned in 7ms
At line 9:
SELECT * FROM IBM;
```

- AWS Table

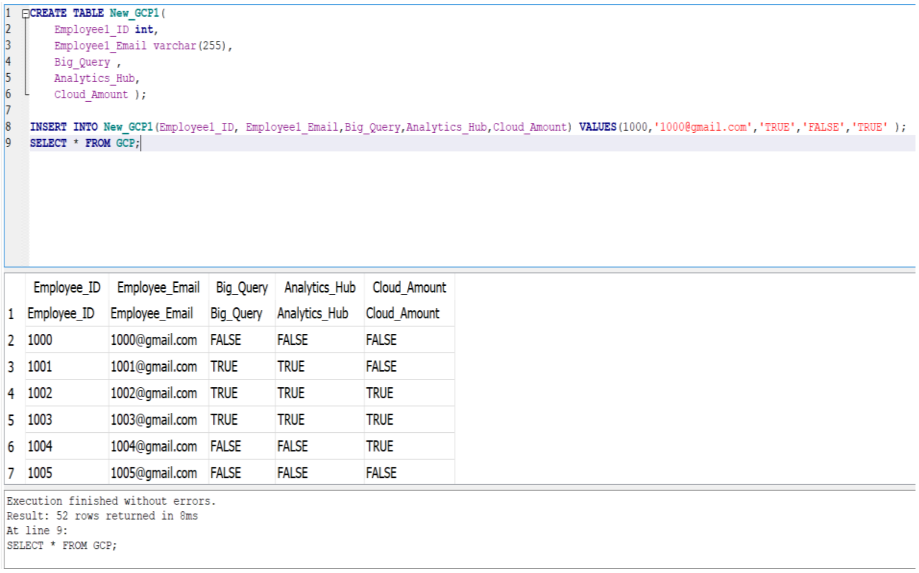


```sql
CREATE TABLE New_Aws4(
    Employee1_ID int,
    Employee1_Email varchar(255),
    Amazon_Elastic_File_System,
    Amazon_Simple_Storage_Service,
    Amazon_Athen,
    Amazon_Honey_Comb );

INSERT INTO New_Aws4(Employee1_ID, Employee1_Email,Amazon_Elastic_File_System,
Amazon_Simple_Storage_Service,Amazon_Athen,Amazon_Honey_Comb)
VALUES(1000,'1000@gmail.com','TRUE','TRUE','FALSE','FALSE');
SELECT * FROM AWS;
```

	Employee_ID	Employee_Email	Amazon_Elastic_File_System	Amazon_Simple_Storage_Service	Amazon_Athen	Amazon_Honey_Comb
1	Employee_ID	Employee_Email	Amazon_Elastic_File_System	Amazon_Simple_Storage_Serv...	Amazon_Athen	Amazon_Honey_Comb
2	1000	1000@gmail.com	TRUE	TRUE	FALSE	FALSE
3	1001	1001@gmail.com	FALSE	FALSE	TRUE	FALSE
4	1002	1002@gmail.com	TRUE	TRUE	TRUE	FALSE
5	1003	1003@gmail.com	TRUE	TRUE	FALSE	TRUE
6	1004	1004@gmail.com	TRUE	TRUE	TRUE	TRUE
7	1005	1005@gmail.com	FALSE	FALSE	FALSE	TRUE

```
Execution finished without errors.
Result: 52 rows returned in 11ms
At line 10:
SELECT * FROM AWS;
```

- GCP Table

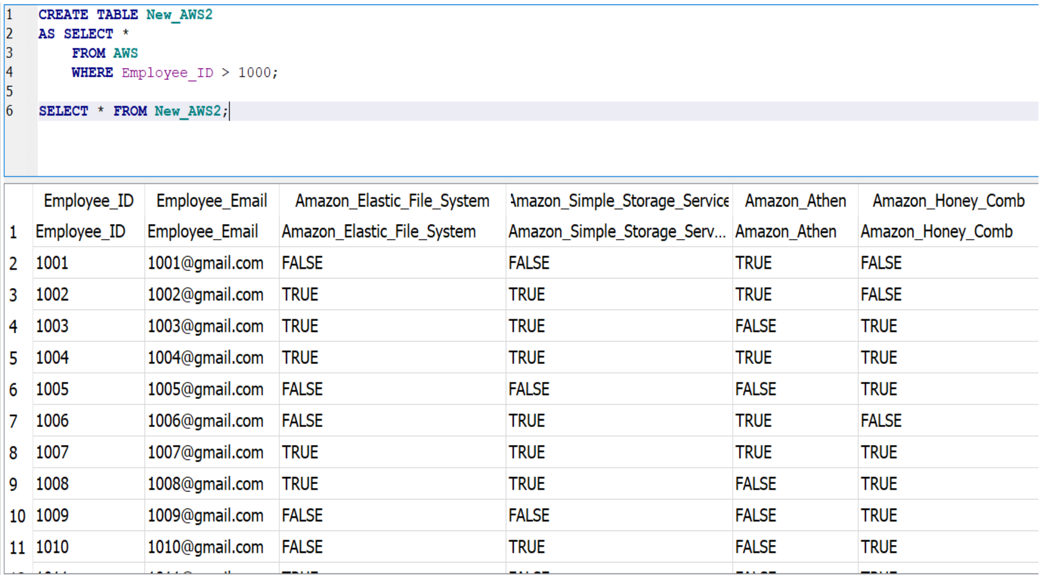


```sql
CREATE TABLE New_GCP1(
    Employee1_ID int,
    Employee1_Email varchar(255),
    Big_Query ,
    Analytics_Hub,
    Cloud_Amount );

INSERT INTO New_GCP1(Employee1_ID, Employee1_Email,Big_Query,Analytics_Hub,Cloud_Amount) VALUES(1000,'1000@gmail.com','TRUE','FALSE','TRUE' );
SELECT * FROM GCP;
```

	Employee_ID	Employee_Email	Big_Query	Analytics_Hub	Cloud_Amount
1	Employee_ID	Employee_Email	Big_Query	Analytics_Hub	Cloud_Amount
2	1000	1000@gmail.com	FALSE	FALSE	FALSE
3	1001	1001@gmail.com	TRUE	TRUE	FALSE
4	1002	1002@gmail.com	TRUE	TRUE	TRUE
5	1003	1003@gmail.com	TRUE	TRUE	TRUE
6	1004	1004@gmail.com	FALSE	FALSE	TRUE
7	1005	1005@gmail.com	FALSE	FALSE	FALSE

```
Execution finished without errors.
Result: 52 rows returned in 8ms
At line 9:
SELECT * FROM GCP;
```

- Azure Table



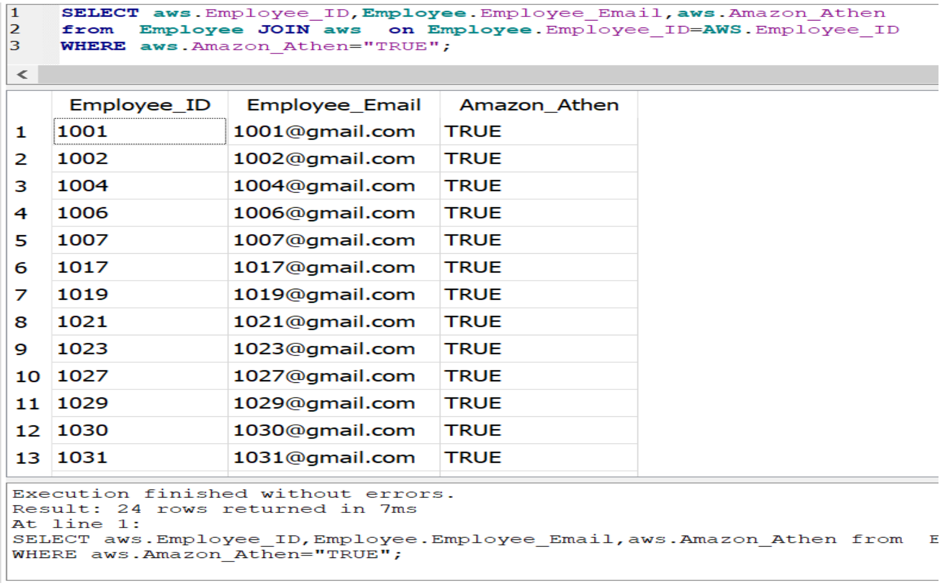
```
CREATE TABLE New_AWS2
AS SELECT *
    FROM AWS
    WHERE Employee_ID > 1000;

SELECT * FROM New_AWS2;
```

	Employee_ID	Employee_Email	Amazon_Elastic_File_System	mazon_Simple_Storage_Service	Amazon_Athen	Amazon_Honey_Comb
1	Employee_ID	Employee_Email	Amazon_Elastic_File_System	Amazon_Simple_Storage_Serv...	Amazon_Athen	Amazon_Honey_Comb
2	1001	1001@gmail.com	FALSE	FALSE	TRUE	FALSE
3	1002	1002@gmail.com	TRUE	TRUE	TRUE	FALSE
4	1003	1003@gmail.com	TRUE	TRUE	FALSE	TRUE
5	1004	1004@gmail.com	TRUE	TRUE	TRUE	TRUE
6	1005	1005@gmail.com	FALSE	FALSE	FALSE	TRUE
7	1006	1006@gmail.com	FALSE	TRUE	TRUE	FALSE
8	1007	1007@gmail.com	TRUE	TRUE	TRUE	TRUE
9	1008	1008@gmail.com	TRUE	TRUE	FALSE	TRUE
10	1009	1009@gmail.com	FALSE	FALSE	FALSE	TRUE
11	1010	1010@gmail.com	FALSE	TRUE	FALSE	TRUE

5. Database Implementation :

- **SQL1: Employees who all are having skill set of Amazon_Athen Skillset**
  - Here it’s just a simple query about the Amazon_Athen skill set by using JOIN.
  - Here we are comparing Employee ID in Employee Entity and AWS Employee ID in AWS Entity for Amazon_Athen
  - Whoever gets the True for Amazon Athen, They are familiar with Amazon_Athen skillset.

```
SELECT aws.Employee_ID,Employee.Employee_Email,aws.Amazon_Athen
from  Employee JOIN aws  on Employee.Employee_ID=AWS.Employee_ID
WHERE aws.Amazon_Athen="TRUE";
```



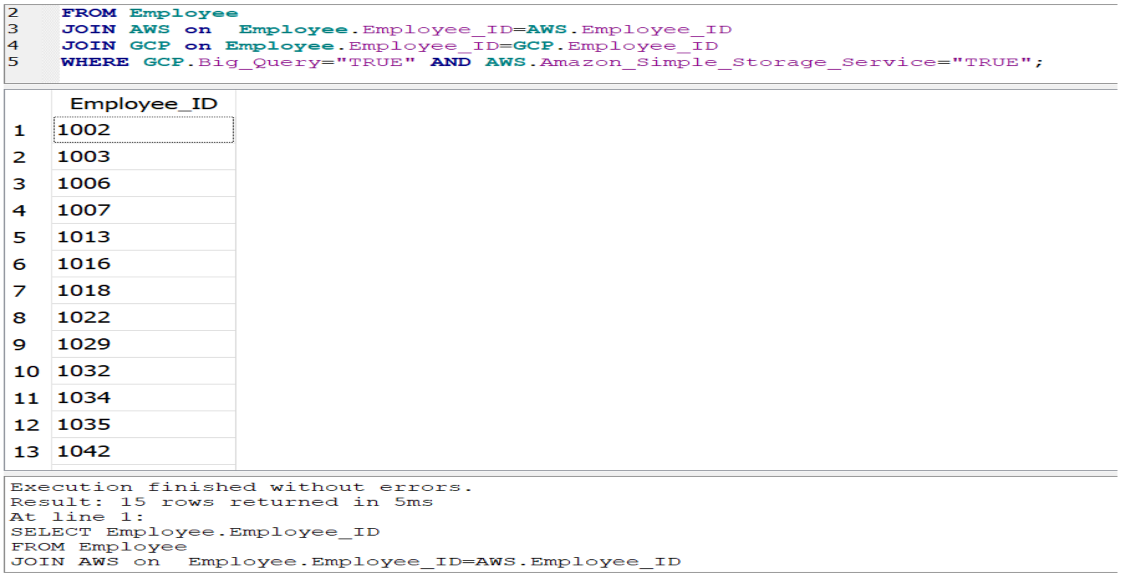
```
SELECT aws.Employee_ID,Employee.Employee_Email,aws.Amazon_Athen
from  Employee JOIN aws  on Employee.Employee_ID=AWS.Employee_ID
WHERE aws.Amazon_Athen="TRUE";
```

	Employee_ID	Employee_Email	Amazon_Athen
1	1001	1001@gmail.com	TRUE
2	1002	1002@gmail.com	TRUE
3	1004	1004@gmail.com	TRUE
4	1006	1006@gmail.com	TRUE
5	1007	1007@gmail.com	TRUE
6	1017	1017@gmail.com	TRUE
7	1019	1019@gmail.com	TRUE
8	1021	1021@gmail.com	TRUE
9	1023	1023@gmail.com	TRUE
10	1027	1027@gmail.com	TRUE
11	1029	1029@gmail.com	TRUE
12	1030	1030@gmail.com	TRUE
13	1031	1031@gmail.com	TRUE

```
Execution finished without errors.
Result: 24 rows returned in 7ms
At line 1:
SELECT aws.Employee_ID,Employee.Employee_Email,aws.Amazon_Athen from  E
WHERE aws.Amazon_Athen="TRUE";
```

- **SQL2: Employees who all are having common skill set of Amazon simple storage service & Big Query Skillset**
  - Here it’s one more query about the GCP_Big Query & AWS_Amazon_Simple _Storage_Service by using JOIN.
  - Here we are using JOIN for AWS & GCP by simply making them to compare with Employee table employee ID and AWS,GCP’s Employee ID’s
  - So whatever the ID’s we get those people are skilled in Both Amazon_Simple_Storage_Service & Big_Query as well.

```
SELECT Employee.Employee_ID
FROM Employee
JOIN AWS on  Employee.Employee_ID=AWS.Employee_ID
JOIN GCP on Employee.Employee_ID=GCP.Employee_ID
 WHERE GCP.Big_Query="TRUE" AND
AWS.Amazon_Simple_Storage_Service="TRUE";
```



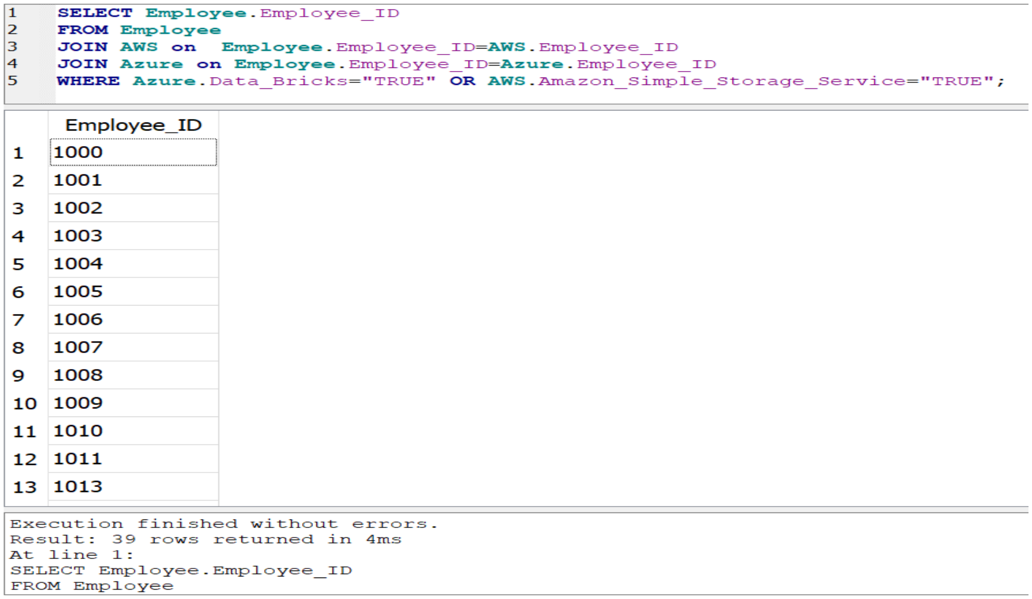
```
FROM Employee
JOIN AWS on  Employee.Employee_ID=AWS.Employee_ID
JOIN GCP on Employee.Employee_ID=GCP.Employee_ID
WHERE GCP.Big_Query="TRUE" AND AWS.Amazon_Simple_Storage_Service="TRUE";
```

```
   Employee_ID
1  1002
2  1003
3  1006
4  1007
5  1013
6  1016
7  1018
8  1022
9  1029
10 1032
11 1034
12 1035
13 1042
```

```
Execution finished without errors.
Result: 15 rows returned in 5ms
At line 1:
SELECT Employee.Employee_ID
FROM Employee
JOIN AWS on  Employee.Employee_ID=AWS.Employee_ID
```

- **SQL3: Employees who all having either Amazon simple storage service OR Big Query Skillset**
  - Here it’s one more query about the GCP_Big Query & AWS_Amazon_Simple _Storage_Service by using JOIN.
  - Here we are using JOIN for Azure & Aws by simply making them to compare with Employee table employee ID and AWS,Azure Employee ID’s
  - So whatever the ID’s we get those people are skilled in either Amazon_Simple_Storage_Service & Data Bricks

```
SELECT Employee.Employee_ID
FROM Employee
JOIN AWS on  Employee.Employee_ID=AWS.Employee_ID
JOIN Azure on Employee.Employee_ID=Azure.Employee_ID
WHERE Azure.Data_Bricks="TRUE" OR
AWS.Amazon_Simple_Storage_Service="TRUE";
```



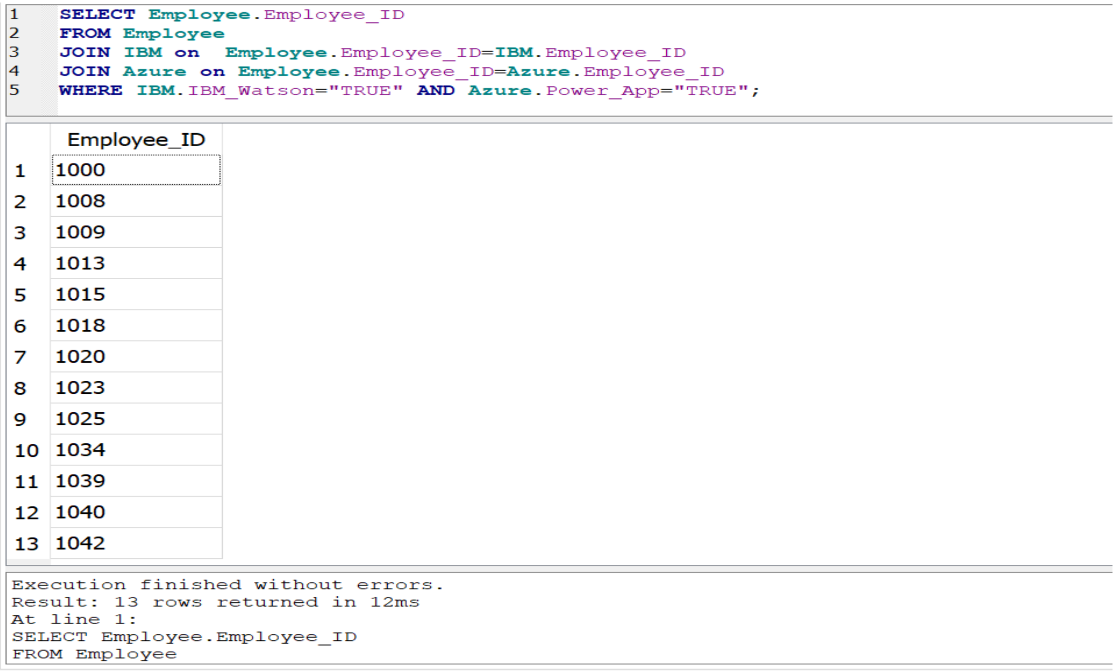
```
SELECT Employee.Employee_ID
FROM Employee
JOIN AWS on  Employee.Employee_ID=AWS.Employee_ID
JOIN Azure on Employee.Employee_ID=Azure.Employee_ID
WHERE Azure.Data_Bricks="TRUE" OR AWS.Amazon_Simple_Storage_Service="TRUE";
```

	Employee_ID
1	1000
2	1001
3	1002
4	1003
5	1004
6	1005
7	1006
8	1007
9	1008
10	1009
11	1010
12	1011
13	1013

```
Execution finished without errors.
Result: 39 rows returned in 4ms
At line 1:
SELECT Employee.Employee_ID
FROM Employee
```

- **SQL4: Employees who all are having common skill set of both Power App & IBM watson**
  - Here it’s one more query about the AZURE_Power_App & IBM_IBM_Watson by using JOIN.
  - Here we are using JOIN for Azure & IBM by simply making them to compare with Employee table employee ID and IBM ,Azure Employee ID’s
  - So whatever the ID’s we get those people are skilled in both Power _app& IBM_watson.

```
SELECT Employee.Employee_ID
FROM Employee
JOIN IBM on  Employee.Employee_ID=IBM.Employee_ID
JOIN Azure on Employee.Employee_ID=Azure.Employee_ID
WHERE IBM.IBM_Watson="TRUE" AND Azure.Power_App="TRUE";
```



```sql
SELECT Employee.Employee_ID
FROM Employee
JOIN IBM on  Employee.Employee_ID=IBM.Employee_ID
JOIN Azure on Employee.Employee_ID=Azure.Employee_ID
WHERE IBM.IBM_Watson="TRUE" AND Azure.Power_App="TRUE";
```

	Employee_ID
1	1000
2	1008
3	1009
4	1013
5	1015
6	1018
7	1020
8	1023
9	1025
10	1034
11	1039
12	1040
13	1042

```
Execution finished without errors.
Result: 13 rows returned in 12ms
At line 1:
SELECT Employee.Employee_ID
FROM Employee
```

6. Metrics and Analytics :

Metrics:

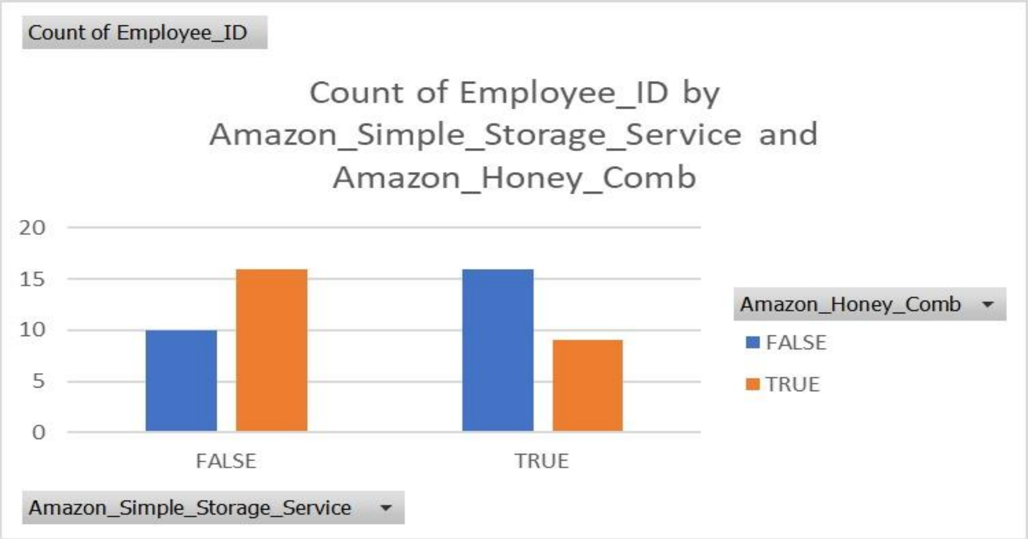
- **Recruitment Team**
  - **Diversity Metrics** – Recruitment from diverse background
  - **Time To Hire -** The time it takes to fill a position from the time
  - **Quality Hire-** How well new employees are performing in their roles
  - **Hiring Source-** How can Recruitment team on-board best candidates from, such as job boards, referrals, or social media.
- **Project Manager & Scrum Master**
  - **Allocation Of Right Skill** – Allocating skilled people into project
  - **Customer Satisfaction** – How satisfied customers are, after releasing the product.
  - **Issue Rate-** How many issues found in the product during testing or after releasing the project.
  - **Completion Rate Of Project-** Percentage of projects are completing with in due time and with great quality.
  - **Productivity Rate-** Helps to evaluate team's efficiency.
- **Regional Manager**

- **Time To Productivity-** The time it takes for new employees to become fully productive in their roles.
- **Employee Engagement-** how employees are to the company and its mission.
- **Training And Development Metrics-** This metrics helps to evaluate employees receiving training, the types of training being offered, and the impact of the training on employee performance.

➢ **Training Team**

- **Effectiveness Of Training-** how good the training is in terms of improving employee performance.
- **Participant Cost-** Cost of training per person.
- **Completion Rate Of Training-** Rate of employees who completed training.
- **Participation Rate-** the percentage of employees who participate in training programs.

Analysis:



➢ The analysis compares the familiarity of two Amazon services, Simple Storage Service and Honeycomb.

➢ The analysis indicates the level of familiarity of the individuals with each service, using the colors blue and orange.

➢ Blue indicates low familiarity, while orange indicates high

➢ The analysis shows that out of the total number of people, around 10 are not very familiar with Simple storage service, while approximately 18 are familiar with the service.

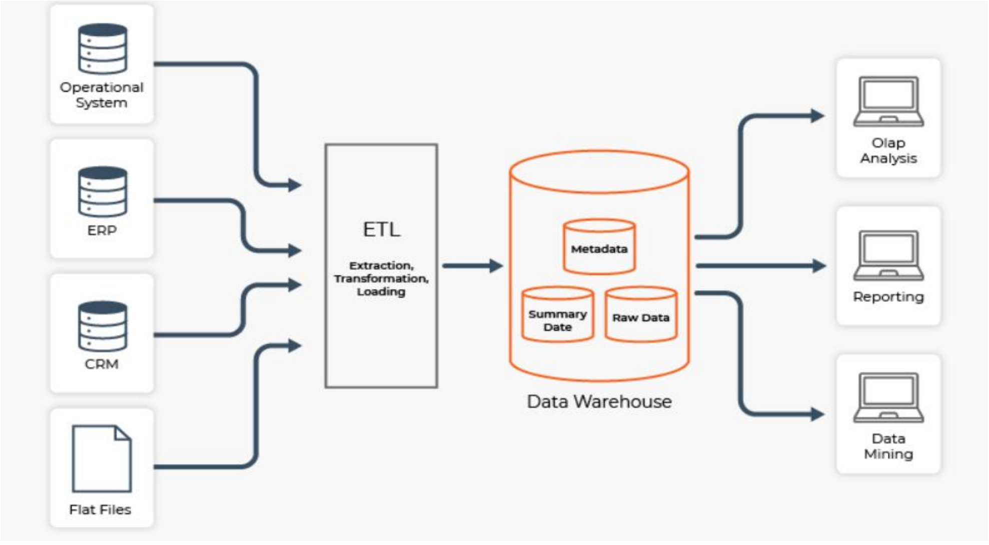
- On the other hand, for Honeycomb, around 8 people are not very familiar with it, while 17 have a high level of familiarity with the service.
- In summary, the analysis compares the level of familiarity of individuals with two Amazon services and shows that more people are familiar with Honeycomb than with Simple storage service. However, there are still some individuals who are not very familiar with either service.

7. Security & Privacy :

For Security and Privacy concerns, the following things I am going to consider

- Data can Stored in 3rd party storage centers.
- Access to limited number of users
- Every user will have a special training Privacy of data
- Monitoring the data base activity of my database
- Regular backups of the database can protect against data loss.
- Since my data includes personal information so to avoid unauthorized accesses i want to encrypt the data.

8. Architecture :



- A **data warehouse** is a collection of diverse data sources that are structured according to a common schema.

- **ETLArea:** :

As the data obtained from external sources may not be standardized, it's necessary to verify its accuracy before loading it into a data warehouse. To achieve this, it's suggested to utilize an ETL (Extract, Transform, Load) tool.

- **E(Extracted):** Data is extracted from outside data source(E).
- **T(Transform):** Finally, the converted data is loaded into a data warehouse (T).
- **L(Load):** After **Transform**, the converted data is loaded into data warehouse(L).

➢ Once the data has been cleaned, it is then stored in a data warehouse as a centralized repository.

➢ From the data warehouse we can access the data for further references.

9. Project WrapUp:

➢ **Future Enhancements:**

- More budget can be allocated to this project for maintenance and storage purposes.
- Additional data can be included in future.
- More user accesses can be given.
- Data can be stored in data centres for security and privacy purposes.

➢ **Lessons Learnt:**

- Got an idea about how an organization handles data.
- Secured an enough knowledge on Security and Privacy of data.
- Got a good understanding on manager’s view on any project.