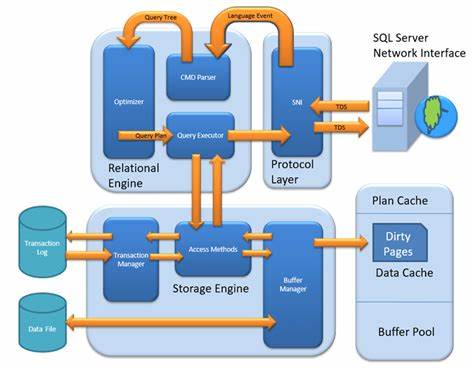
**Project Documentation: Alumni Career Choices Analysis**

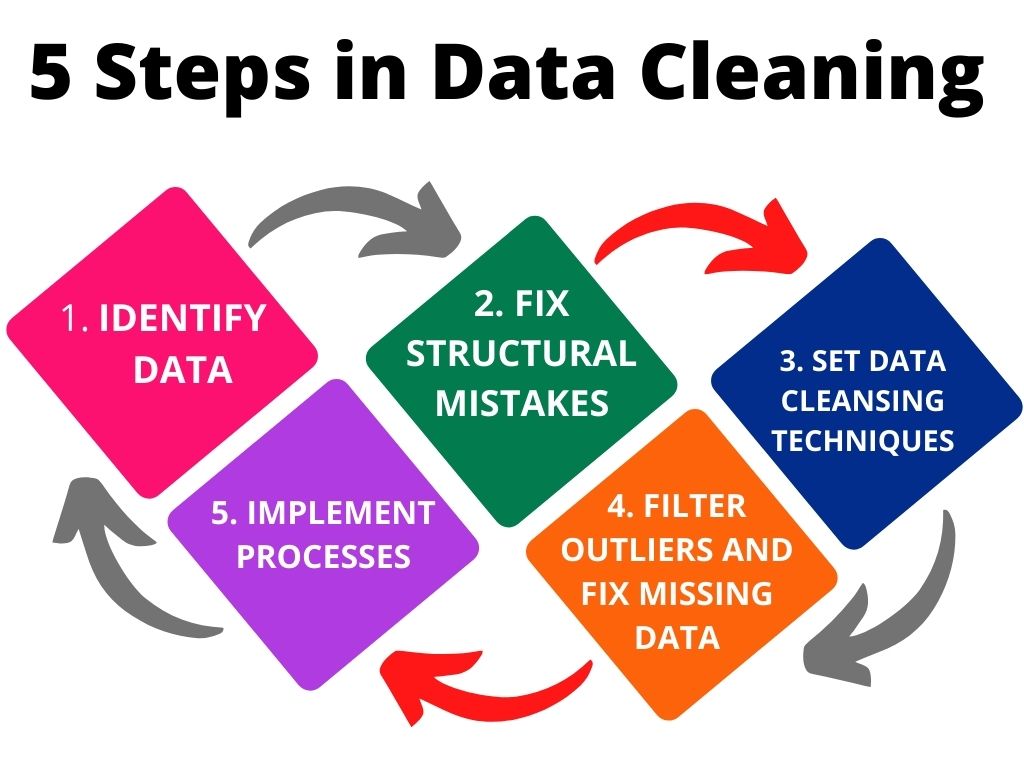
**Aim**

The aim of this project is to analyze the career choices of alumni from two universities with respect to their passing year and the courses they completed. This analysis helps in understanding the career growth of alumni, which plays a crucial role in the institute's ranking and networking activities.



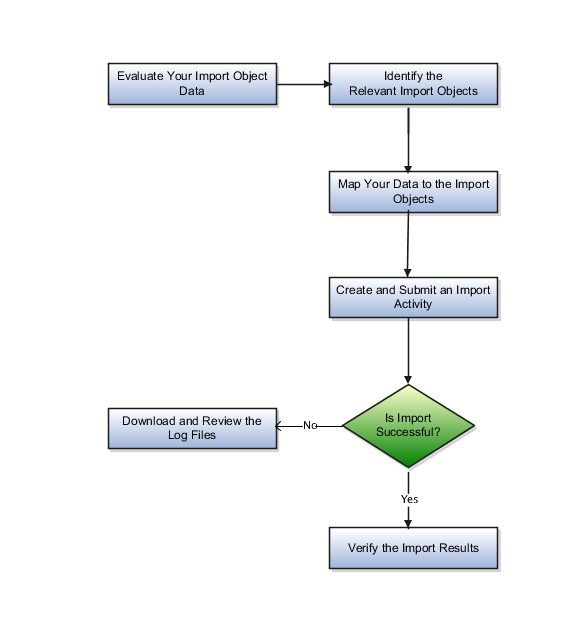
**Objectives**

1. Import and structure alumni data from six CSV files into a MySQL database.
2. Perform data cleaning to remove null values and ensure data consistency.
3. Create views for cleaned data for better management and analysis.
4. Use string functions to format name fields.
5. Analyze and visualize the location distribution of alumni using pivot charts in Excel.
6. Implement stored procedures to fetch and display alumni names.
7. Calculate and compare the percentage of career choices among alumni of the two universities.



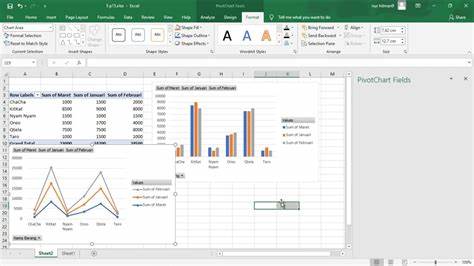
**Introduction**

This project involves analyzing the career trajectories of alumni from two universities, referred to as College A and College B. The data includes records of higher studies, self-employment, and job services. By processing and analyzing this data, we aim to derive insights into the career paths chosen by the alumni.



**Technology Stack Used**

* **Database**: MySQL
* **Programming Language**: Python (Jupyter Notebook)
* **Data Analysis**: MS Excel
* **Data Visualization**: MS Excel Pivot Charts



**Key SQL Queries and Their Functionality**

1. **Create Database Schema**

sql

CREATE DATABASE alumni;

1. **Import Data and View Structure**

USE alumni;

DESC college\_a\_hs;

DESC college\_a\_se;

DESC college\_a\_sj;

DESC college\_b\_hs;

DESC college\_b\_se;

DESC college\_b\_sj;

1. **Data Cleaning and Creating Views**

CREATE OR REPLACE VIEW College\_A\_HS\_V AS

SELECT RollNo, LastUpdate, Name, FatherName, MotherName, Batch, Degree, PresentStatus, HSDegree, EntranceExam, Institute, Location

FROM college\_a\_hs

WHERE RollNo IS NOT NULL AND LastUpdate IS NOT NULL AND Name IS NOT NULL AND FatherName IS NOT NULL AND MotherName IS NOT NULL AND Batch IS NOT NULL AND Degree IS NOT NULL AND PresentStatus IS NOT NULL AND HSDegree IS NOT NULL AND EntranceExam IS NOT NULL AND Institute IS NOT NULL AND Location IS NOT NULL;

CREATE OR REPLACE VIEW College\_A\_SE\_V AS

SELECT RollNo, LastUpdate, Name, FatherName, MotherName, Batch, Degree, PresentStatus, Organization, Location

FROM college\_a\_se

WHERE RollNo IS NOT NULL AND LastUpdate IS NOT NULL AND Name IS NOT NULL AND FatherName IS NOT NULL AND MotherName IS NOT NULL AND Batch IS NOT NULL AND Degree IS NOT NULL AND PresentStatus IS NOT NULL AND Organization IS NOT NULL AND Location IS NOT NULL;

CREATE OR REPLACE VIEW College\_A\_SJ\_V AS

SELECT RollNo, LastUpdate, Name, FatherName, MotherName, Batch, PresentStatus, Organization, Designation, Location

FROM college\_a\_sj

WHERE RollNo IS NOT NULL AND LastUpdate IS NOT NULL AND Name IS NOT NULL AND FatherName IS NOT NULL AND MotherName IS NOT NULL AND Batch IS NOT NULL AND Degree IS NOT NULL AND PresentStatus IS NOT NULL AND Organization IS NOT NULL AND Designation IS NOT NULL AND Location IS NOT NULL;

CREATE OR REPLACE VIEW College\_B\_HS\_V AS

SELECT RollNo, LastUpdate, Name, FatherName, MotherName, Branch, Batch, Degree, PresentStatus, HSDegree, EntranceExam, Institute, Location

FROM college\_b\_hs

WHERE RollNo IS NOT NULL AND LastUpdate IS NOT NULL AND Name IS NOT NULL AND FatherName IS NOT NULL AND MotherName IS NOT NULL AND Branch IS NOT NULL AND Batch IS NOT NULL AND Degree IS NOT NULL AND PresentStatus IS NOT NULL AND HSDegree IS NOT NULL AND EntranceExam IS NOT NULL AND Institute IS NOT NULL AND Location IS NOT NULL;

CREATE OR REPLACE VIEW College\_B\_SE\_V AS

SELECT RollNo, LastUpdate, Name, FatherName, MotherName, Branch, Batch, Degree, PresentStatus, Organization, Location

FROM college\_b\_se

WHERE RollNo IS NOT NULL AND LastUpdate IS NOT NULL AND Name IS NOT NULL AND FatherName IS NOT NULL AND MotherName IS NOT NULL AND Branch IS NOT NULL AND Batch IS NOT NULL AND Degree IS NOT NULL AND PresentStatus IS NOT NULL AND Organization IS NOT NULL AND Location IS NOT NULL;

CREATE OR REPLACE VIEW College\_B\_SJ\_V AS

SELECT RollNo, LastUpdate, Name, FatherName, MotherName, Branch, Batch, Degree, PresentStatus, Organization, Designation, Location

FROM college\_b\_sj

WHERE RollNo IS NOT NULL AND LastUpdate IS NOT NULL AND Name IS NOT NULL AND FatherName IS NOT NULL AND MotherName IS NOT NULL AND Branch IS NOT NULL AND Batch IS NOT NULL AND Degree IS NOT NULL AND PresentStatus IS NOT NULL AND Organization IS NOT NULL AND Designation IS NOT NULL AND Location IS NOT NULL;

1. **String Functions for Name Formatting**

SELECT LOWER(Name), LOWER(FatherName), LOWER(MotherName) FROM College\_a\_hs\_v;

SELECT LOWER(Name), LOWER(FatherName), LOWER(MotherName) FROM College\_a\_se\_v;

SELECT LOWER(Name), LOWER(FatherName), LOWER(MotherName) FROM College\_a\_sj\_v;

SELECT LOWER(Name), LOWER(FatherName), LOWER(MotherName) FROM College\_b\_hs\_v;

SELECT LOWER(Name), LOWER(FatherName), LOWER(MotherName) FROM College\_b\_se\_v;

SELECT LOWER(Name), LOWER(FatherName), LOWER(MotherName) FROM College\_b\_sj\_v;

1. **Stored Procedures for Fetching Alumni Names**

DROP PROCEDURE IF EXISTS get\_name\_collegeA;

DELIMITER $$

CREATE PROCEDURE get\_name\_collegeA (INOUT name1 TEXT(40000))

BEGIN

DECLARE na INT DEFAULT 0;

DECLARE namelist VARCHAR(16000) DEFAULT "";

DECLARE namedetail CURSOR FOR

SELECT Name FROM college\_a\_hs UNION SELECT Name FROM college\_a\_se UNION SELECT Name FROM college\_a\_sj;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET na = 1;

OPEN namedetail;

getame : LOOP

FETCH FROM namedetail INTO namelist;

IF na = 1 THEN

LEAVE getame;

END IF;

SET name1 = CONCAT(namelist, ";", name1);

END LOOP getame;

CLOSE namedetail;

END $$

DELIMITER ;

SET @Name = "";

CALL get\_name\_collegeA(@Name);

SELECT @Name AS Name;

DROP PROCEDURE IF EXISTS get\_name\_collegeB;

DELIMITER $$

CREATE PROCEDURE get\_name\_collegeB (INOUT name1 TEXT(40000))

BEGIN

DECLARE na INT DEFAULT 0;

DECLARE namelist VARCHAR(16000) DEFAULT "";

DECLARE namedetail CURSOR FOR

SELECT Name FROM college\_b\_hs UNION SELECT Name FROM college\_b\_se UNION SELECT Name FROM college\_b\_sj;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET na = 1;

OPEN namedetail;

getame : LOOP

FETCH FROM namedetail INTO namelist;

IF na = 1 THEN

LEAVE getame;

END IF;

SET name1 = CONCAT(namelist, ";", name1);

END LOOP getame;

CLOSE namedetail;

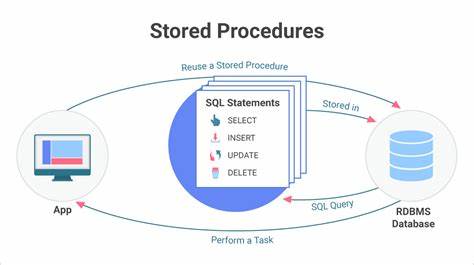
END $$

DELIMITER ;

SET @Name = "";

CALL get\_name\_collegeB(@Name);

SELECT @Name AS Name;



1. **Calculating Career Choice Percentages**

SELECT

Career\_choice,

ROUND((SUM(CASE WHEN College = 'A' THEN 1 ELSE 0 END) / COUNT(\*)) \* 100, 4) AS College\_A\_Percentage,

ROUND((SUM(CASE WHEN College = 'B' THEN 1 ELSE 0 END) / COUNT(\*)) \* 100, 4) AS College\_B\_Percentage

FROM (

SELECT 'Higher Studies' AS Career\_choice, 'A' AS College FROM college\_a\_hs WHERE PresentStatus = 'Higher Studies'

UNION ALL

SELECT 'Self Employed' AS Career\_choice, 'A' AS College FROM college\_a\_se WHERE PresentStatus = 'Self Employed'

UNION ALL

SELECT 'Service/Job' AS Career\_choice, 'A' AS College FROM college\_a\_sj WHERE PresentStatus = 'Service/Job'

UNION ALL

SELECT 'Higher Studies' AS Career\_choice, 'B' AS College FROM college\_b\_hs WHERE PresentStatus = 'Higher Studies'

UNION ALL

SELECT 'Self Employed' AS Career\_choice, 'B' AS College FROM college\_b\_se WHERE PresentStatus = 'Self Employed'

UNION ALL

SELECT 'Service/Job' AS Career\_choice, 'B' AS College FROM college\_b\_sj WHERE PresentStatus = 'Service/Job'

) AS combined

GROUP BY Career\_choice;

**Key Challenges Faced**

* Handling null values and ensuring data consistency.
* Structuring and managing a large volume of data from multiple sources.
* Creating efficient SQL queries to perform the required data transformations and analysis.
* Ensuring the accuracy of the percentage calculations for career choices.

**Future Improvements or Scope of Work**

* Automate the data import process to handle updates and new records.
* Implement more sophisticated data cleaning techniques to handle a broader range of data issues.
* Expand the analysis to include additional metrics such as salary ranges, job satisfaction, and industry trends.
* Develop a web-based dashboard for real-time visualization and interactive analysis of alumni data.

**References**

* [MySQL Documentation](https://dev.mysql.com/doc/)
* [Python Documentation](https://docs.python.org/3/)
* Pandas Documentation