**Part1:** Integrate the extracted data as one source of data. Take care of the format and semantic conflicts. Some information about the three data sources obtained from the data dictionaries. If you have a question, then you can interview me as the representative of the USRH via zoom. Explanations of all the actions that you have taken and their justifications must be included in a word document. (Name this document Project Report).

1. Convert pdf to csv format in order to manipulate the data from the tables and clean it.
2. Create new csv files from the data sources 1,2 and 3.

**DataSource 1-2 Colleges Table**

* This table is cleaned manually because it is a small subset of data. It is easy to manipulate all records and clearly see what should be changed.
* The column names are changed to lowercase to begin a lowercase column name convention throughout the future data warehouse and MySQL database.
* The names of the Colleges have missing data to some of the associated majors. It is implied that the names are categories for the majors and the missing data is in fact the majors associated with the name.

**Original**

| NAME | MAJORS |
| --- | --- |
| Cyber College | Computer Sc |
|  | Information Sc |
|  | Applied Sc |
| College of Business | Accounting |
|  | Business Admin |
|  | Economics |
| College of Education | Elementary Ed |
|  | Secondary Ed |
| College of Art and Science | Biology |
|  | Chemistry |
|  | English |

**Cleaned**

| NAme | majors |
| --- | --- |
| Cyber College | Computer Sc |
| Cyber College | Information Sc |
| Cyber College | Applied Sc |
| College of Business | Accounting |
| College of Business | Business Admin |
| College of Business | Economics |
| College of Education | Elementary Ed |
| College of Education | Secondary Ed |
| College of Art and Science | Biology |
| College of Art and Science | Chemistry |
| College of Art and Science | English |

**DataSource 3 Colleges Table**

* This table was also manually cleaned because of the size of the data table. The clear standardization that is consistent with datasource1-2 cleaned table, makes this table straight forward.
* The names column has attributes that are renamed to match the new name for each of the colleges.

**Original**

| NAME | MAJORS |
| --- | --- |
| Computer Communications | Computer Sc |
| (Old name for the Cyber College) | Information Sc |
|  | Applied Sc |
| College of Business and IS | Accounting |
| (Old name for the College of Business) | Business Admin |
|  | Economics |
| College of Education | Elementary Ed |
|  | Secondary Ed |
| College of Science | Biology |
| (Old name for the College of Art and Science) | Chemistry |

**Cleaned**

| name | majors |
| --- | --- |
| Cyber College | Computer Sc |
| Cyber College | Information Sc |
| Cyber College | Applied Sc |
| College of Business | Accounting |
| College of Business | Business Admin |
| College of Business | Economics |
| College of Education | Elementary Ed |
| College of Education | Secondary Ed |
| College of Art and Science | Biology |
| College of Art and Science | Chemistry |
| College of Art and Science | English |

Now that datasource1\_2 table and datasource3 are the same they can be merged and integrated into one Colleges table .

**Create New CSV Tables: Status and Degree for each source of data.**

**Degree Data Tables:**

* Utilized when analyzing the ‘Degree’ column from the different data sources of data.
  + ds3\_degree : ‘Classification’ of integers, and ‘Degrees’
  + ds1\_2\_degree: ‘Classification’ of abbreviations and ‘Degrees’

**Status Data Tables:**

* Utilized when analyzing the ‘Status’ column from the different data sources of data.
  + ds3\_status : ‘Classification’ of integers, and ‘Status’
  + ds1\_2\_status: ‘Classification’ of first initials and ‘Status’

**Clean and Integrate data from DataSource 1 and 2, and DataSource 3**

Set Up the Python coding environment and git, github repository.

For the data cleaning process we will use Python Jupyter Lab Notebook

Data Cleaning Process: Create Jupyter Notebook

* Utilize Python pandas - import pandas as pd

**DataSource 3 Cleaning**

**Degree** - *Ensure correct information, spelling, align with data dictionary.*

* + Because the datasources1\_2 has more total rows in the column ‘Degree’ I use the existing naming conventions from the larger table which are string abbreviations. Datasource3 will change to match datasources1\_2, converting the assigned integers that classify ‘Degree’ into their correlated strings that they represent.
  + From Datasource3 convert integer classifications to string classifications
    - ds3[‘Degrees’] = [1, 2, 3, 4, 5]
    - ds1\_2[‘Degrees’] = ['AS', 'BS', 'BA', 'MS', 'MA', 'MBA', 'EdD', 'PhD']
  + Convert Degree column for Students table from dataframe df3. Change integer values of degree to that of mapped string degree values like ds1\_2’s 'Degree' column.
  + Create a mapping based on datasources CSV files.
    - Create the mapping dictionary to map integer classification to string classifications.
    - Do the conversion for dataframe df3 ‘Degree’ based on classification mapping.

**Status**- *Ensure correct information, understand classification conversions align with data dictionaries.*

* + Because the datasources1\_2 has a ‘Status’ Column based on the initial abbreviations for the actual string status and datasource3 has status mapped to an integer representation, datasource3 ‘Status ’ column will be converted into the initial abbreviations that represent the actual string status defined in the datasources Status Table.
  + Convert the status of datasource3 from the numerical status distinction to the lettering format ION, since degree is also a string value.
  + Correct the df3 status to match df1\_2 status using the same mapping as the degree conversion mapping.
  + Base the df3 update on datasours1\_2\_status and datasource3\_status csv files.
  + Create a mapping dictionary for the "Status" column
    - Reverse the status mapping so it can be read properly from df3
  + Crate mapping to be used on the merged data and convert ‘Status’ abbreviations to full strings to be more useful in the future
  + After the merge the status column will be converted into the full string status description.

**DataSource1\_2 Cleaning**

* **MonthDay** - The ‘MonthDay’ column consists of each row containing a Month and The day of the month. The data dictionary defines the semester classifications based on the end dates, which in turn supply the start dates.
  + Many of the MonthDay row values had extra spaces, dashes or periods that needed to be stripped.
    - Some dates have (.), abbreviating months while others do not. We will strip those periods(.) from the rows entirely
    - Some dates have to many days valid to be in any month’s 30-31 day range.
      * Either can assume the date is the next month ahead by the days exceeding the days of the month constraint for that month
      * Or the date is meant to be a range within that month
      * Or drop the record and store it, until more information is obtained from the owner.
  + Most of the string values in the MonthDay for month had 3 characters while some had 4 such as June and July, these were converted into their 3 character date abbreviation.
  + **‘Y’ -** To account for the semester classification, the ‘y’ column need to be taken into account
    - Pandas Timestamp and datetime accounts for year and requires a 4-digit year input. So the ‘Y’ column prepends a ‘19’ to each row value.
  + **‘Semester’** - Convert MonthDay to Semester for datasource1\_2 - because datasource3 is in the semester format and does not specify the actual date of graduation like in datasources1\_2. I will classify MonthDay dates into semesters. Then merge df3 into df1\_2 after df1\_2 is converted.
    - A fulldate column is created as the column import to classify the semesters. This clarification builds the Semester row that aligns with the ‘Semester’ column from datasource3.
    - The ‘MonthDay’ column is then dropped, replaced by the ‘Semester’ column.

**Merge Semi-Cleaned DataSources**

* **Degree -** Convert degree abbreviation to full string degree description
* **Status -** convert to full string status descriptions
  + This step converts the status column from the initial status description mapping to the full string status description for all rows of the merged data
    - 5 rows in the status column, from datasource 3 were classified as a 5 which is out of the 3 tier range for status. For these values I manually classified them as a 3 (Non-of-the above).
* **Name** - check for duplicates, correct spelling, proper initial structure, ensure uppercase, lowercase conventions
  + Find duplicates in the ‘Name’
    - Determine why the name duplicate exists
      * Reson1: the person is the same but the graduation year is different and the degree is different. ---> if they are the same, check the status --->. If the status is the same delete the duplicate---> if they are different determine which one is erroneous
      * **Reject all duplicates and ask the owner about status inconsistencies use pandas shape attribute, 9 Name duplicates rejected and stored for questioning.**
* **ID** - ensure 3 digits, Ensure no duplicates in ID
  + Find duplicates in the 'ID' column
  + For duplicates found in the column ID, generate 3-digit numbers in order **to replace each duplicate after the first occurrence.**
  + For the generated 3-digit numbers, make sure they are unique by comparing the 3 digits to that of the entire original data frame.
* **Major** - ensure spelling is correct, no missing data, no missing parts of words, aligns to data dictionaries
  + Clean column for missing data, extra characters and spaces.
  + Clean this column based on the Colleges Table created by the data dictionary given. Name of colleges and their majors.
  + Create a list of valid majors from the colleges table.
  + Check each row to determine invalid, misspelled majors and correct them by using string matching. Once this is complete replace the old major's column for each row with the correct valid major.
* **Address** - ensure accurate city names, spelling
  + Fix spelling, extra characters. symbols and spaces.
  + Manually create a valid address city name list for comparison.
  + Clean this column with the same string matching method used for the major column.
  + Replace the entire column for each row with the correct city name for address.
* **GPA** - ensure GPA’s are in range, should also be float values for gpa decimals which are important
  + All GPA row values were valid
  + GPARank - Column created to determine if a student has a high, medium or low gpa which is based on the scale provided in extra information.

**Integrated Data Output Snippet:**

|  | **ID** | **Name** | **Address** | **Major** | **Degree** | **GPA** | **GPARank** | **Semester** | **Y** | **Status** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **0** | 999 | CK. Brown | Denver | Applied Sc | Associate degree | 4.0 | High | Sum1 | 1989 | International Student |
| **1** | 961 | QT. Brady | Pine Bluff | Computer Sc | Associate degree | 3.8 | High | Sum2 | 1987 | International Student |
| **2** | 962 | UL. Jones | Little Rock | Computer Sc | Associate degree | 3.1 | High | Sum2 | 1984 | Out of state |
| **3** | 970 | Bf. Brady | Little Rock | Applied Sc | Associate degree | 3.2 | High | Sum1 | 1992 | Out of state |
| **4** | 975 | CM. Cook | Little Rock | Biology | Bachelor of Science | 2.2 | Medium | F | 1987 | Out of state |
| **5** | 412 | DP. Morty | Atlanta | Biology | Bachelor of Art | 3.9 | High | S | 1984 | Out of state |

**Data Warehouse Design**

**Part 2:** Design a data warehouse using one of the schemas presented in class. Justify your

choice of schema. For your schema, identify and justify dimensions and their

attributes, fact table and its attributes, concept and schema hierarchy (ies) for each

dimension. All of your dimensions, their table’s definitions, and justifications

along with all of your diagrams (Only the computer-generated diagrams are

acceptable) must be included in the document Project Report under proper

headings of your choice.

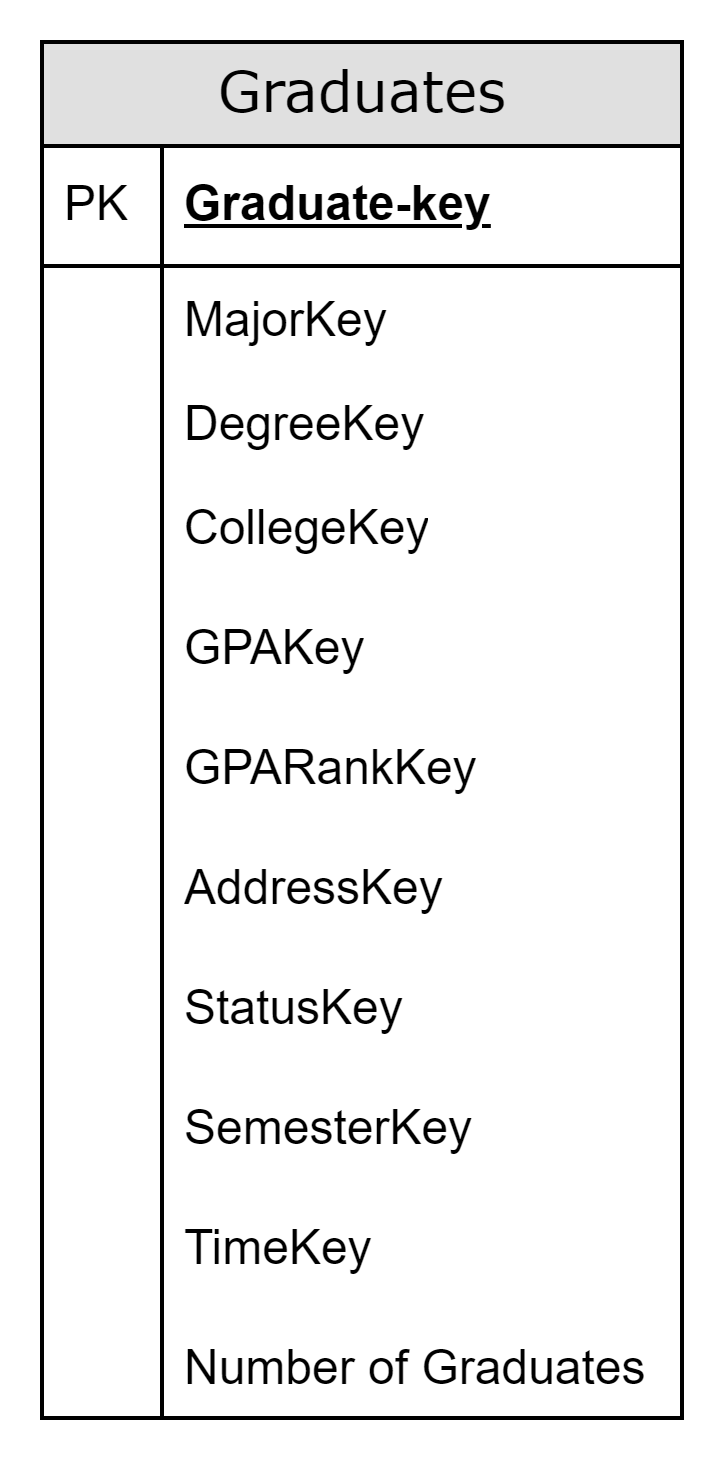
**Star Schema**

This data warehouse will follow the star schema , because the system will follow simplistic querying, which follows a central fact table connected directly to the dimension tables.

The dimensions chosen for this fact table will focus on ensuring plain english based queries can be executed for achieving the number of graduates depending on specific parameters. Each dimension will have its own concept hierarchy, and the fact table will use foreign keys to link the dimension table. Data is structured efficiently making it easier for users to work with the data. Foreign keys in the Graduates dimensions are separated to avoid redundancy and complexity.

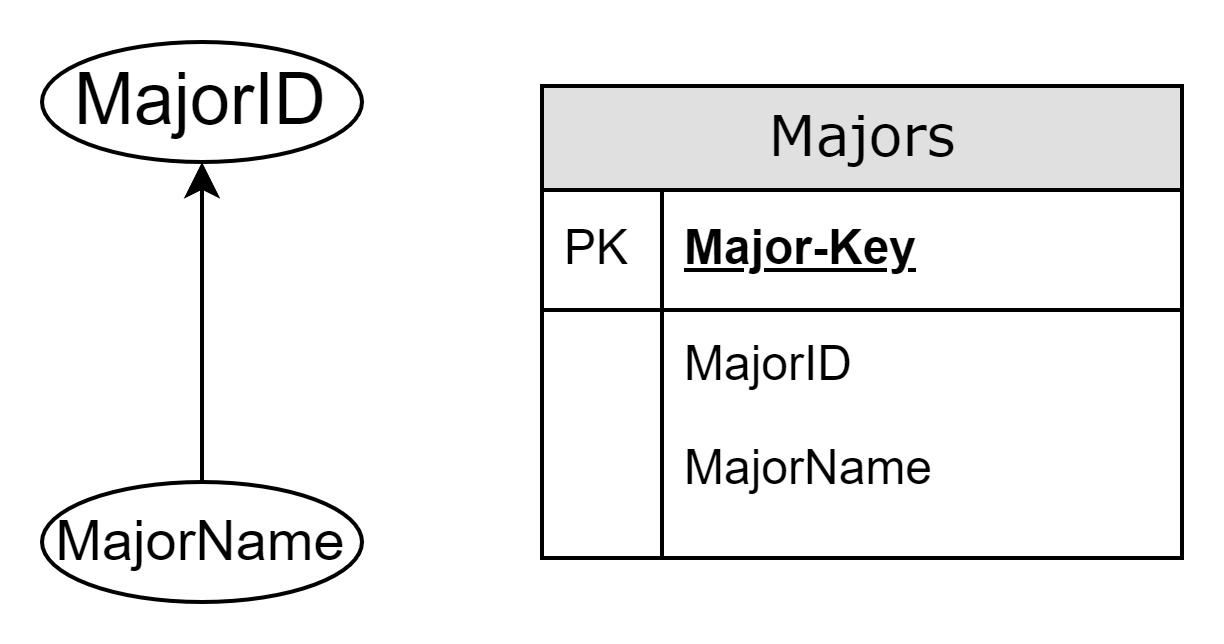
**Number of Graduates Fact Table:**

* GraduateKey
* MajorKey
* DegreeKey
* CollegeKey
* GPAKey
* GPARankKey
* AddressKey
* StatusKey
* SemesterKey
* TimeKey
* Number of Graduates

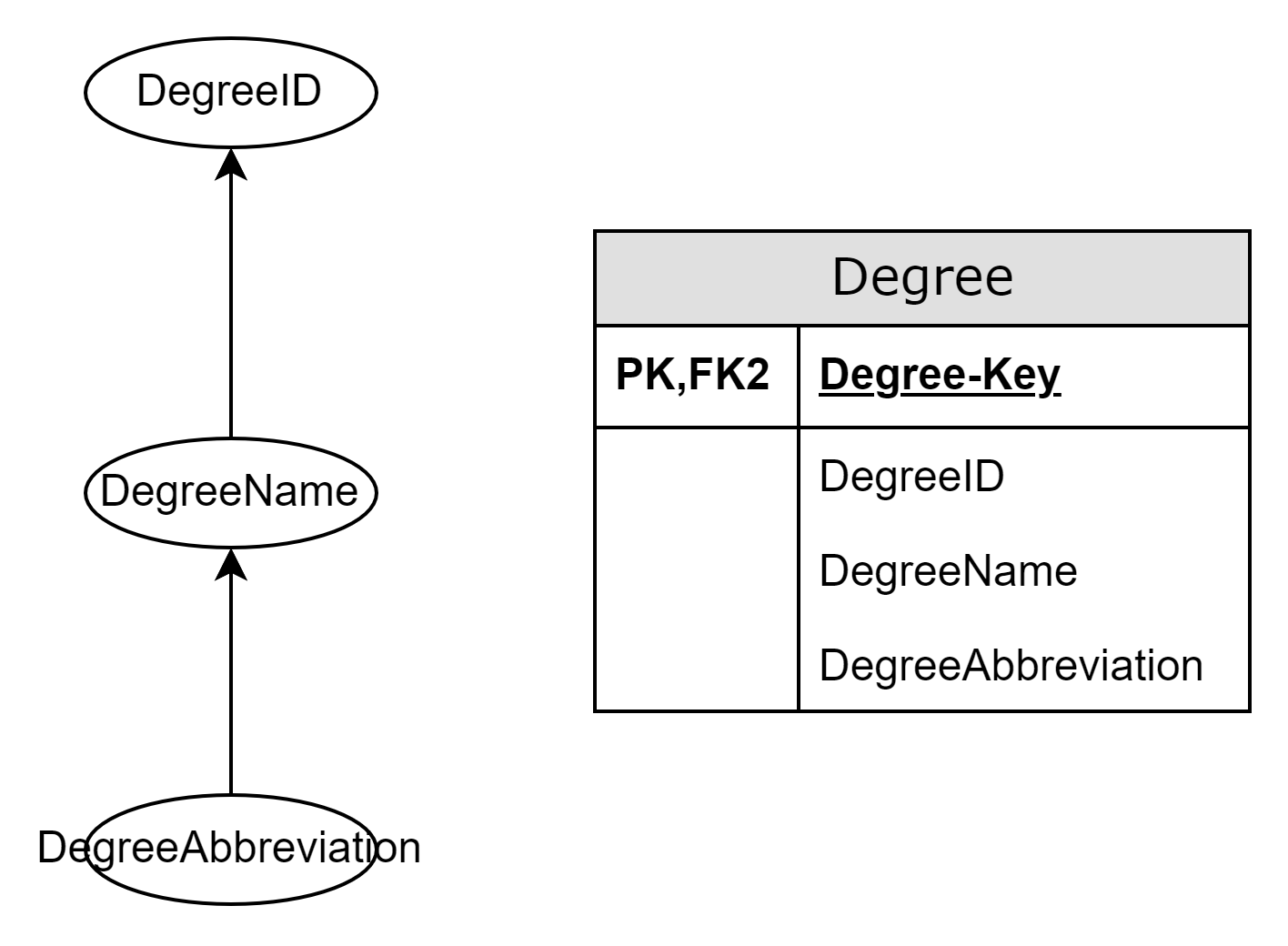


**Dimensions**:

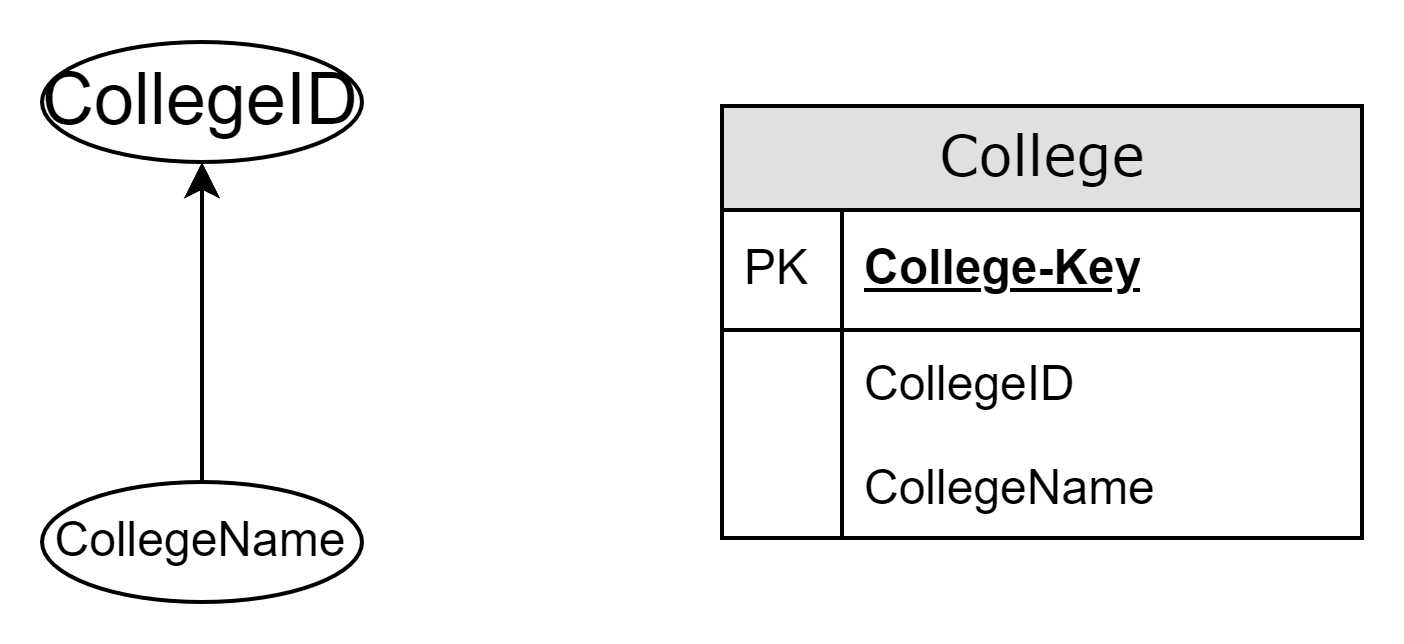
* **Graduates**
  + **Attributes:**
    - **GraduateKey (Primary Key, unique identifier for each graduate)**
      * **GraduateID**
      * **GraduateName**
  + **Concept Hierarchy**
* **MajorKey(Foreign Key, references Major Dimension)**
  + **Attributes**
    - **MajorID**
    - **MajorName**
  + **Concept Hierarchy and Table**

****

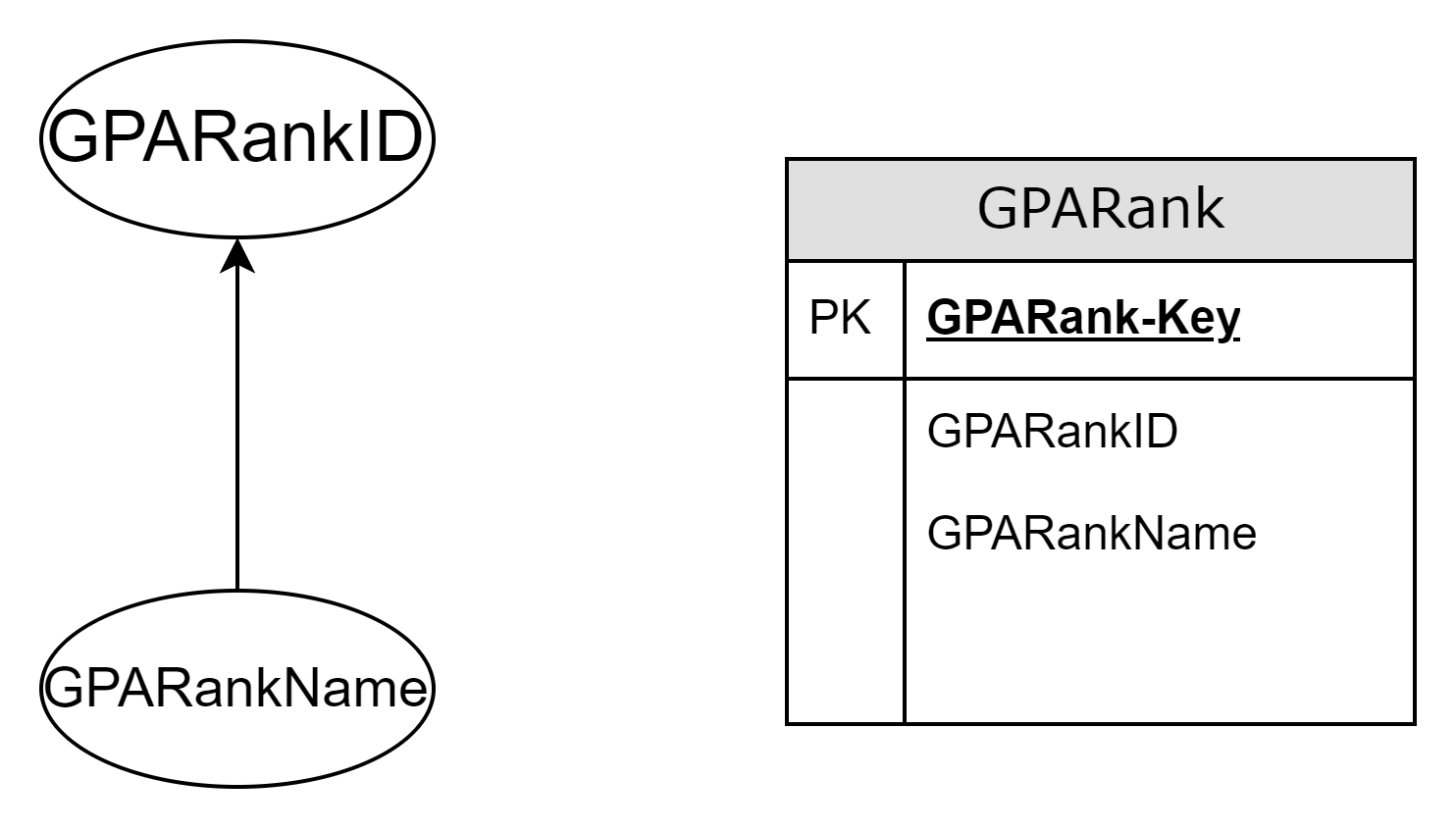
* **DegreeKey(Foreign Key, references Degree Dimension)**
  + **Attributes**
    - **DegreeID**
    - **DegreeName**
    - **DegreeAbrreviaton**
  + **Concept Hierarchy and Table**

****

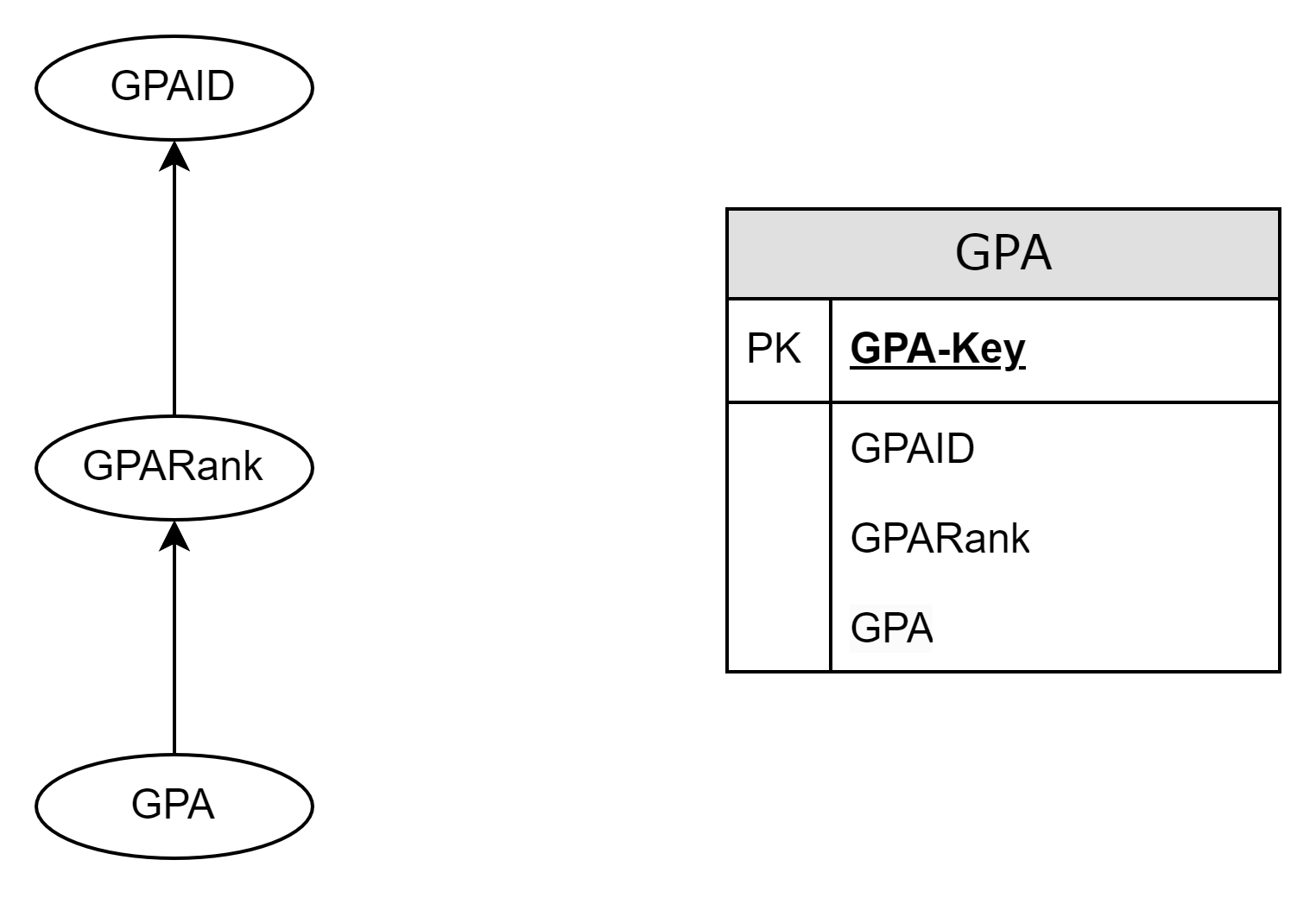
* **CollegeKey(Foreign Key, references College Dimension)**
  + **Attributes**
    - **CollegeID**
    - **CollegeName**
  + **Concept Hierarchy**

****

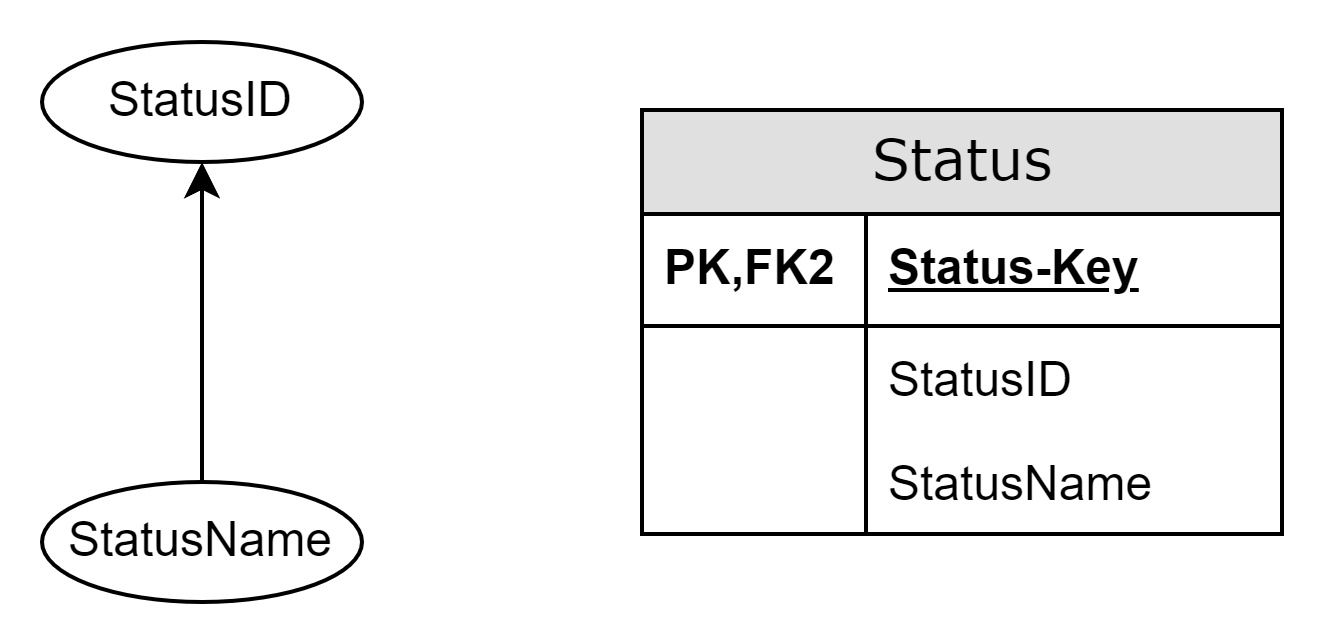
* **GPARankKey(Foreign Key, references GPA Rank Dimension)**
  + **Attributes**
    - **GPARank**

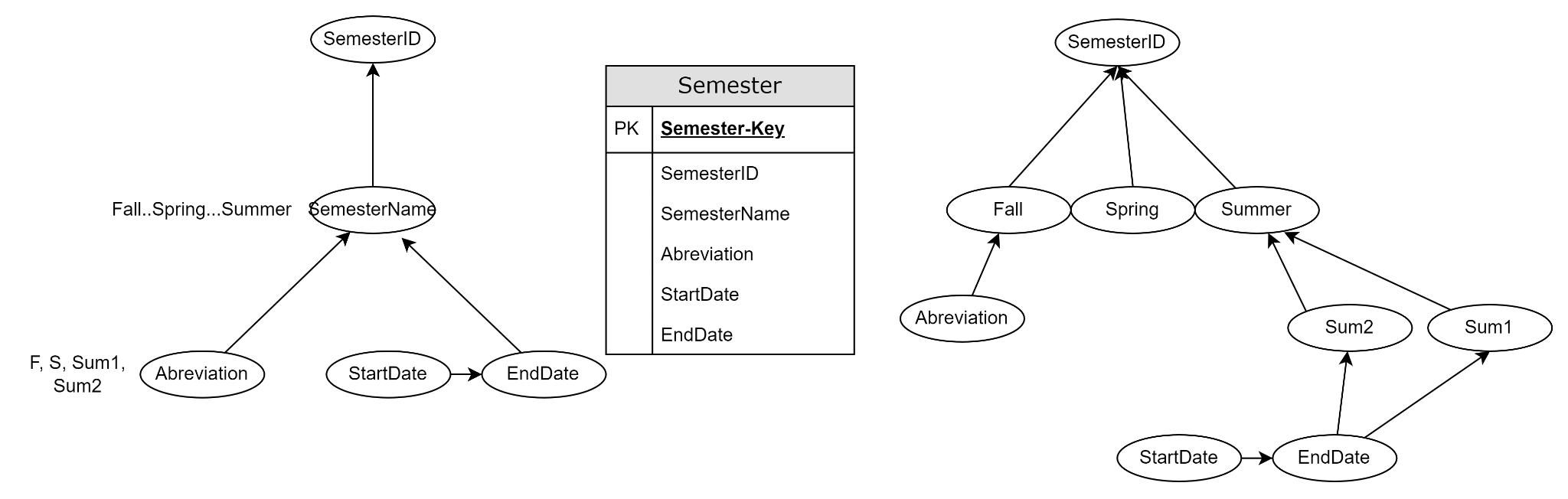
****

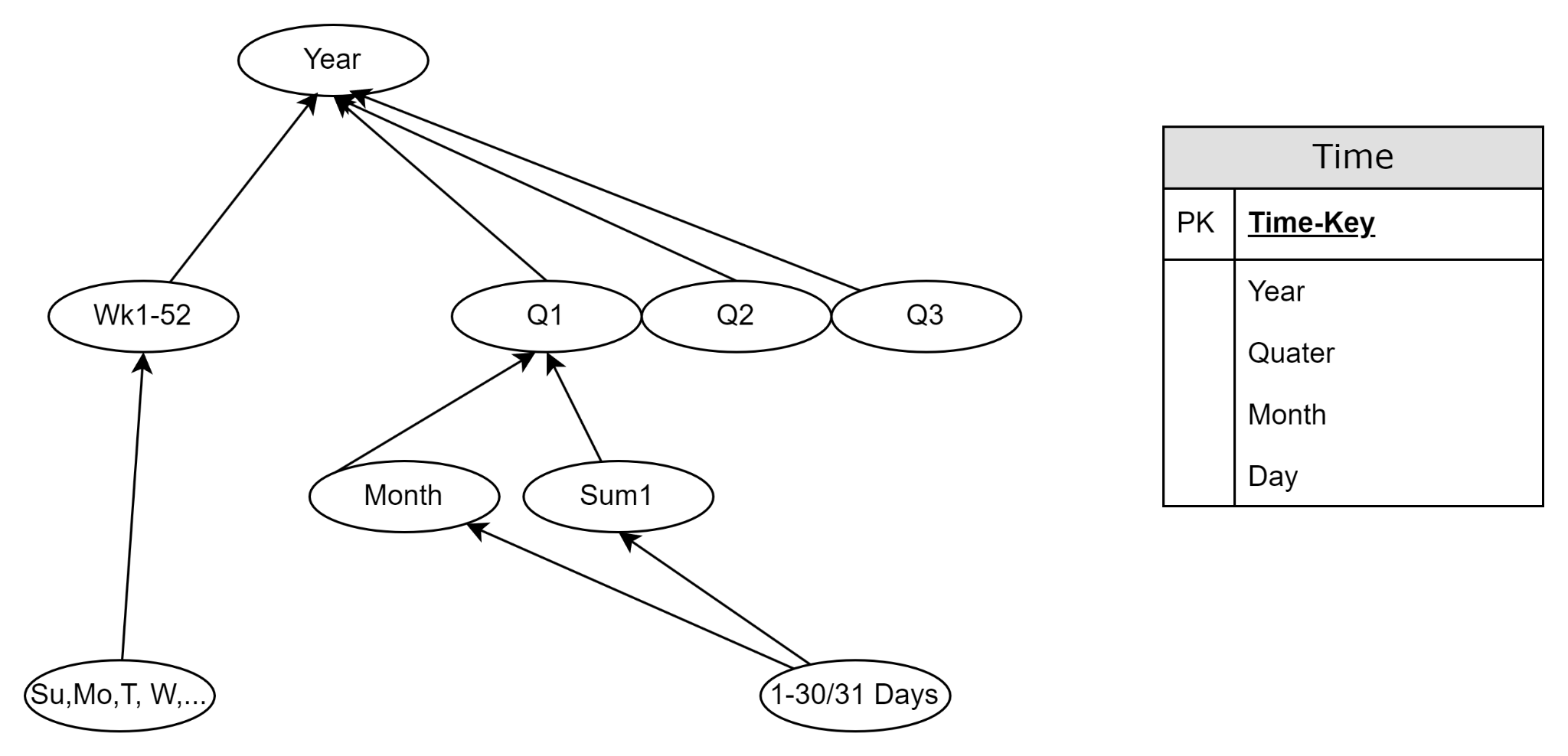
* **GPAKey (Foreign Key, references GPA Dimension)**
  + **Attributes**
    - **GPAID**
    - **GPA**

****

* **StatusKey(Foreign Key, references Status Dimension)**
  + **Attributes**
    - **StatusID**
    - **StatusName**

****

* **SemesterKey(Foreign Key, references Semester Dimension)**
  + **Attributes**
    - **SemesterID**
    - **SemesterName**
    - **Abbreviation**
    - **StartDate**
    - **EndDate**
  + **Concept Hierarchy**
    - **Highest level is categorized into Fall, Spring, Summer**
    - **Summer has another detailed lower level Summer1 and Summer2**
* **TimeKey(Foreign Key, references Year Dimension)**
  + **Attributes**
    - **Year**
    - **Quarter**
    - **Month**
    - **Day**
  + **Concept Hierarchy**



* **GraduateCount (The measure, which is the count of graduates)**
  + An aggregated measure that summarizes data across all graduates allowing the view of data in terms of the overall number of graduates.

