Smith Rankings Database Documentation

Release Date: Dec 12, 2023	Version: 1.0	Issued for Approval

Project Team:

Project Leader: Helena Getachew	Database Architect: Sharjeel Nawaz	Lead Developer: Anusha Aligeti	Quality Assurance: Ei Aung

Ownership

Developed by UMD Robert H. Smith School of Business for internal use. Refer to data usage policy document for guidelines.

Compliance & Standards

This database documentation complies with:

- SQL Server 2017 documentation standards
- UMD data classification and storage standards
- ISO 9001:2015 software development processes
- Object Management Group (OMG) Standard

Revision History

Version 1.0 - Initial release (Issued for Approval)

Distribution List

Adam Lee (DBMS Professor for BUDT 702 at R. H. Smith, University of Maryland)

Jia-Pei (Teaching Assistant for BUDT 702)

Dean, Robert H. Smith School of Business, University of Maryland

Management Team, Robert H. Smith School of Business, University of Maryland

Other stakeholders, with prior approval from Professor Adam Lee

Exclusion and Augmentation Notice

The team has tried to use actual data as much as possible. However, UIDs of students and teachers have been replaced with dummy augmented values. Before formal roll out, UIDs must be updated as per actual records.

Confidentiality Notice

This document contains confidential information related to data architecture and IT infrastructure. Do not share any details without authorization.

For any information, or inquiries, please contact Professor Adam Lee.

Overview

The Smith Rankings database is implemented in SQL Server and tracks rankings and related data on graduate school programs over multiple years. It enables analysis of program KPI trends and correlation with rankings.

Before using this document and SQL code, at any given date beyond December 10, 2023, please review following:

1. Rankings Update

- New rankings data is released yearly from publishers like U.S. News
- Update the Year entity with the current year
- Insert the new rankings into Rate table for each program and source
- Retain historical rankings to allow trend analysis

2. New Session Student Batch Ingestion

Before start of Fall session every year:

- Capture new student intake via batch insert into Student entity
- Enroll new students in their programs via Enroll table inserts

3. Program Metadata Updates

- Any changes in program details like name, number of credits etc. should be captured by updating Program entity records

4. Faculty Changes

- As faculty members join/leave, insert/update/delete records in the Faculty table accordingly
- Update links to students taught and programs employed in respectively

5. Database Reorganization

- To optimize space utilization and access efficiency, reorganize tables periodically
- Can rebuild indexes after large batch inserts of rankings or student intake

6. Backups

- Schedule periodic database backups for disaster recovery
- Balance frequency vs overhead to balance RPO and RTO

7. Documentation

- All schema changes must be updated in documentation shared with stakeholders
- Maintain data dictionary mapping all attributes to business meaning

8. Testing

- Validate any schema changes, new code deployment in staging environment before production rollout.

1.1 Select your SQL server database

To select an appropriate database, make sure you have access to the SQL server database. For testing and docker operations, make sure you create and use a separate database.

As default, we are using our project database 'BUDT702_Project_0503_04.

```
-- Selecting the database:
USE BUDT702_Project_0503_04
```

The USE operator will set the database for next SQL instructions.

1.2. Drop tables to maintain data consistency

Since our code has CREATE TABLE instructions, it is always good to drop these tables to avoid any conflicts and to ensure data integrity. DROP TABLE IF EXISTS, only drops the table if it is already in the database. This way, if we run the code for the first time, there will be no impact on the code.

```
-- SQL DROP TABLE:
DROP TABLE IF EXISTS [Smith_Ranking.Rate]
DROP TABLE IF EXISTS [Smith_Ranking.Enroll]
DROP TABLE IF EXISTS [Smith_Ranking.Employ]
DROP TABLE IF EXISTS [Smith_Ranking.Teach]
DROP TABLE IF EXISTS [Smith_Ranking.Program]
DROP TABLE IF EXISTS [Smith_Ranking.Student]
DROP TABLE IF EXISTS [Smith_Ranking.Faculty]
DROP TABLE IF EXISTS [Smith_Ranking.Year]
DROP TABLE IF EXISTS [Smith_Ranking.Source]
```

1.3. Create new tables

CREATE TABLE instruction is used to create new tables in the database. Tables are initially created without any records. It is important to make sure that the sequence of table creation is consistent with our constraints, primary and foreign keys.

For our database, we have created following entity tables:

- [Smith_Ranking.Program]
- [Smith_Ranking.Student]
- [Smith_Ranking.Source]
- [Smith_Ranking.Year]
- [Smith_Ranking.Faculty]

```sql
-- Creating the Program table
CREATE TABLE [Smith_Ranking.Program]  (
     prId CHAR (4) NOT NULL,
     prName VARCHAR (30) NOT NULL,
     prDegType VARCHAR (3),
     prType VARCHAR (16),
     prEmpRate INTEGER,
     prAvgStartSalary INTEGER,
     prPercentJobGrowth INTEGER,
     prNumofCredits INTEGER,
     prAvgGPA DECIMAL (3,2),
     prNumofFaculty INTEGER,
     CONSTRAINT pk_Program_prId PRIMARY KEY (prId)
     );
```

```sql
-- Creating the Student table
CREATE TABLE [Smith_Ranking.Student](
     stuId VARCHAR(9) NOT NULL,
     stuFirstName VARCHAR(20) NOT NULL,
     stuMidInitial VARCHAR(1),
     stuLastName VARCHAR(20) NOT NULL,
     stuGPA DECIMAL(3,2),
     stuGMAT INTEGER,
     stuGRE INTEGER,
     stuEnrolLYear DATE NOT NULL,
     stuSemester VARCHAR (6),
     stuGradYear DATE,
     stuGradSemester VARCHAR (6),
     CONSTRAINT pk_Student_stuId PRIMARY KEY (stuId)
     );
```

```sql
-- Creating the Source table
CREATE TABLE [Smith_Ranking.Source] (
     souId CHAR (3) NOT NULL,
     souName VARCHAR (20) NOT NULL,
     souRankType CHAR (15),
     CONSTRAINT pk_Source_souId PRIMARY KEY (souId)
     );
```

```sql
-- Creating the Year table
CREATE TABLE [Smith_Ranking.Year]  (
    rankYear DATE NOT NULL,
    CONSTRAINT pk_Year_rankYear PRIMARY KEY (rankYear)
    );
```

```
-- Creating the Faculty table
CREATE TABLE [Smith_Ranking.Faculty] (
    facId VARCHAR (9) NOT NULL,
    facFirstName VARCHAR (20) NOT NULL,
    facMidInitial VARCHAR (3) ,
    facLastName VARCHAR (20) NOT NULL,
    facCourse VARCHAR (40),
    facDesgination VARCHAR (20),
    facEmailAddress VARCHAR (40),
    CONSTRAINT pk_Faculty_facId PRIMARY KEY (facId)
    );
```

For our database, we have created following relationship tables:

- [Smith_Ranking.Employ]
- [Smith_Ranking.Enrol]
- [Smith_Ranking.Rate]
- [Smith_Ranking.Teach]

```
-- Creating the Employ table
CREATE TABLE [Smith_Ranking.Employ] (
    facId VARCHAR (9) NOT NULL,
    prId CHAR (4) NOT NULL,
    CONSTRAINT pk_Employ_facId_prId PRIMARY KEY (facId, prId),
    CONSTRAINT fk_Employ_facId FOREIGN KEY (facId)
        REFERENCES [Smith_Ranking.Faculty] (facId)
        ON DELETE NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT fk_Employ_prId FOREIGN KEY (prId)
        REFERENCES [Smith_Ranking.Program] (prId)
        ON DELETE NO ACTION ON UPDATE NO ACTION
        ) ;
-- Creating the Enroll table
CREATE TABLE [Smith_Ranking.Enroll]  (
    stuId VARCHAR(9) NOT NULL,
    prId CHAR (4) NOT NULL,
    enrolDate DATE,
    CONSTRAINT pk_Enroll_stuId_prId PRIMARY KEY (stuId, prId),
    CONSTRAINT fk_Enroll_stuId FOREIGN KEY (stuId)
        REFERENCES [Smith_Ranking.Student] (stuId)
        ON DELETE NO ACTION ON UPDATE CASCADE,
    CONSTRAINT fk_Enroll_prId FOREIGN KEY (prId)
        REFERENCES [Smith_Ranking.Program] (prId)
        ON DELETE NO ACTION ON UPDATE CASCADE
    );
```

```
--Creating the Rate table
CREATE  TABLE   [Smith_Ranking.Rate]    (
    prId    CHAR (4)    NOT NULL,
    souId CHAR (3) NOT NULL,
    rankYear    DATE    NOT NULL,
    ranking INTEGER,
    CONSTRAINT  pk_Rate_prId_souId_rankYear PRIMARY KEY (prId, souId, rankYear),
    CONSTRAINT  fk_Rate_prId    FOREIGN KEY (prId)
        REFERENCES  [Smith_Ranking.Program] (prId)
        ON  DELETE NO ACTION ON UPDATE  NO ACTION,
    CONSTRAINT  fk_Rate_souId   FOREIGN KEY (souId)
        REFERENCES  [Smith_Ranking.Source]  (souId)
        ON  DELETE NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT fk_Rate_rankYear FOREIGN KEY (rankYear)
        REFERENCES [Smith_Ranking.Year] (rankYear)
        ON  DELETE NO ACTION ON UPDATE NO ACTION
        );
```

```
--Creating the Teach table
CREATE TABLE [Smith_Ranking.Teach] (
    facId VARCHAR (9) NOT NULL,
    stuId VARCHAR(9) NOT NULL,
    CONSTRAINT pk_Teach_facId_stuId PRIMARY KEY (facId, stuId),
    CONSTRAINT fk_Teach_facId FOREIGN KEY (facId)
        REFERENCES [Smith_Ranking.Faculty] (facId)
        ON DELETE NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT fk_Teach_stuId FOREIGN KEY (stuId)
        REFERENCES [Smith_Ranking.Student] (stuId)
        ON DELETE NO ACTION ON UPDATE NO ACTION
        );
```

1.4. Queries for analytics

To analyze the data available in the database, we write various queries and display tables with requested information. SELECT chooses the columns to be displayed, and FROM chooses TABLE of interest. We can filter, arrange, group and join the outputs using various keywords like WHERE, ORDER BY, GROUP BY, etc.

```
-- What are the rankings of Smith school programs by U.S. News and QS World in 2023?
SELECT r.ranking AS 'Ranking', p.prName AS 'Program', s.souName AS 'Source', YEAR(r.rankYear) AS 'Year'
FROM [Smith_Ranking.Program] p JOIN [Smith_Ranking.Rate] r
ON p.prId = r.prId JOIN [Smith_Ranking.Source] s
ON s.souId = r.souId
WHERE r.ranking IS NOT NULL
    AND r.rankYear = '2023'
ORDER BY r.rankYear ASC, r.ranking ASC, p.prName ;
```

	Ranking	Program	Source	Year
1	16	Information Systems	U.S. News	2023
2	28	Supply Chain Mgmt	QS World	2023
3	38	Business Analytics	QS World	2023
4	39	Marketing Analytics	QS World	2023
5	44	Finance	QS World	2023

```
-- What were the number of faculty for programs ranked by QS World in 2023 and their respective rankings?
SELECT p.prName AS 'Program', p.prNumOfFaculty AS 'Number of Faculty', r.ranking AS 'Ranking'
FROM [Smith_Ranking.Program] p, [Smith_Ranking.Rate] r
WHERE r.souId = '2'
    AND p.prId = r.prId
    AND r.rankYear = '2023'
ORDER BY r.ranking, p.prNumofFaculty ;
```

	Program	Number of Faculty	Ranking
1	Supply Chain Mgmt	10	28
2	Business Analytics	12	38
3	Marketing Analytics	8	39
4	Finance	13	44

```sql
-- What is the trend of Smith school program rankings issued by QS World between 2021-2023?
SELECT p.prName AS 'Program', YEAR(r.rankYear) AS 'Year', r.ranking AS 'Ranking'
FROM [Smith_Ranking.Program] p JOIN [Smith_Ranking.Rate] r
ON p.prId = r.prId
WHERE r.souId = '2'
ORDER BY p.prName, r.rankYear ASC, r.ranking ASC ;
```

	Program	Year	Ranking
1	Business Analytics	2021	25
2	Business Analytics	2022	31
3	Business Analytics	2023	38
4	Finance	2021	41
5	Finance	2022	44
6	Finance	2023	44
7	Marketing Analytics	2021	34
8	Marketing Analytics	2022	37
9	Marketing Analytics	2023	39
10	Supply Chain Mgmt	2021	24
11	Supply Chain Mgmt	2022	26
12	Supply Chain Mgmt	2023	28

```sql
-- What are the average starting salaries and their average rankings issued between 2021-2023 of Smith school programs?
SELECT p.prName AS 'Program', AVG(p.prAvgStartSalary) AS 'Average Starting Salary', AVG(r.ranking) AS 'Average Ranking', s.souName AS 'Source'
FROM [Smith_Ranking.Program] p JOIN [Smith_Ranking.Rate] r
ON p.prId = r.prId JOIN [Smith_Ranking.Source] s
ON r.souId = s.souId
GROUP BY p.prName, s.souName
ORDER BY AVG(r.ranking) ASC, AVG(p.prAvgStartSalary) DESC ;
```

	Program	Average Starting Salary	Average Ranking	Source
1	Information Systems	85666	10	U.S. News
2	Supply Chain Mgmt	59789	18	U.S. News
3	Accounting	63229	24	U.S. News
4	Supply Chain Mgmt	59789	26	QS World
5	Business Analytics	71389	31	QS World
6	Marketing Analytics	87953	36	QS World
7	Finance	74285	43	QS World

```sql
-- What are the employment rates six months post-graduation for Smith programs ranked by by QS World in 2021?
SELECT p.prName AS 'Program Name', p.prEmpRate AS 'Employment Rate', r.ranking AS 'Rank', YEAR(r.rankYear) AS 'Year'
FROM [Smith_Ranking.Program] p JOIN [Smith_Ranking.Rate] r
ON p.prId = r.prId
WHERE r.souId = '2'
    AND r.rankYear = '2021'
ORDER BY r.ranking, p.prEmpRate DESC ;
```

	Program Name	Employment Rate	Rank	Year
1	Supply Chain Mgmt	90	24	2021
2	Business Analytics	96	25	2021
3	Marketing Analytics	82	34	2021
4	Finance	74	41	2021

```sql
-- What are the average GPAs and rankings of Smith programs by QS World in 2023?
SELECT (p.prName) AS 'Program Name', p.prAvgGPA AS 'Average GPA', r.ranking AS 'Rank', YEAR(r.rankYear) AS 'Year'
FROM [Smith_Ranking.Program] p JOIN [Smith_Ranking.Rate] r
ON p.prId = r.prId
WHERE p.prAvgGPA IS NOT NULL
    AND r.rankYear = '2023'
    AND r.souId = '2'
ORDER BY p.prAvgGPA, r.ranking ;
```

	Program Name	Average GPA	Rank	Year
1	Supply Chain Mgmt	3.30	28	2023
2	Business Analytics	3.30	38	2023
3	Finance	3.40	44	2023

References

Average Starting Salaries, 2020-2023- https://careers.rhsmith.umd.edu/hiresmith/

Accounting Program Data - https://www.rhsmith.umd.edu/programs/business-masters/academics/accounting

Finance Program Data - https://www.rhsmith.umd.edu/programs/business-masters/academics/finance

Business Analytics Program Data - https://www.rhsmith.umd.edu/programs/business-masters/academics/business-analytics-person

Information Systems Program Data- https://www.rhsmith.umd.edu/programs/business-masters/academics/information-systems

Management Studies Program Data - https://www.rhsmith.umd.edu/programs/business-masters/academics/management-studies-person

Marketing Analytics Program Data - https://www.rhsmith.umd.edu/programs/business-masters/academics/marketing-analytics

Quantitative Finance Program Data - https://www.rhsmith.umd.edu/programs/business-masters/academics/quantitative-finance

Supply Chain Management Data - https://www.rhsmith.umd.edu/programs/business-masters/academics/supply-chain-management

Faculty Data - https://www.rhsmith.umd.edu/directory

MSIS 2023 First-Year Student Data - https://umd.instructure.com/courses/1348596/users

QS World Rankings -https://www.topuniversities.com/world-university-rankings/2023

U.S. News Rankings- https://www.usnews.com/best-graduate-schools/top-business-schools/university-of-maryland-college-park-01104