

USE BUDT702_Project_0501_05

-- SQL drop tables:

```
DROP TABLE IF EXISTS [Project.Uses];
DROP TABLE IF EXISTS [Project.Teaches];
DROP TABLE IF EXISTS [Project.Ranks];
DROP TABLE IF EXISTS [Project.RankingCriteria];
DROP TABLE IF EXISTS [Project.GraduateProgram];
DROP TABLE IF EXISTS [Project.RankSource];
DROP TABLE IF EXISTS [Project.Faculty];
```

--SQL create tables:

```
CREATE TABLE [Project.GraduateProgram] (
gradPrgID CHAR (20) NOT NULL,
gradPrgName VARCHAR(30),
gradPrgCategory VARCHAR(5),
gradPrgDescription VARCHAR(200),
gradPrgDuration INT,
gradPrgFee CHAR(10),
gradPrgCredits INT,
stemDesignation VARCHAR(10),
CONSTRAINT pk_GraduateProgram_gradPrgID PRIMARY KEY (gradPrgID),
);
```

```
CREATE TABLE [Project.RankSource] (
rankSourceID CHAR (20) NOT NULL,
rankSourceName VARCHAR(150),
rankSourceDescription VARCHAR(200),
rankSourceURL VARCHAR(100),
CONSTRAINT pk_RankSource_rankSourceID PRIMARY KEY (rankSourceID),
);
```

```
CREATE TABLE [Project.Faculty] (
facultyID CHAR (20) NOT NULL,
facultyFirstName VARCHAR(20),
facultyMiddleName VARCHAR(10),
facultyLastName VARCHAR(20),
facultyQualification CHAR(30),
facultyJoiningDate DATE,
facultyDesignation VARCHAR(100),
researchPapersPublished INT,
CONSTRAINT pk_Faculty_facultyID PRIMARY KEY (facultyID),
);
```

```
CREATE TABLE [Project.RankingCriteria] (
criteriaID CHAR (20) NOT NULL,
criteriaName VARCHAR(30),
criteriaDescription VARCHAR(200),
CONSTRAINT pk_RankingCriteria_criteriaID PRIMARY KEY (criteriaID)
);
```

```
CREATE TABLE [Project.Ranks] (
gradPrgID CHAR (20) NOT NULL,
rankSourceID CHAR(20),
rankYear INT,
```

```

overallRank INT,
CONSTRAINT pk_Ranks_gradPrgID_rankSourceID_rankYear PRIMARY
KEY(gradPrgID,rankSourceID,rankYear),
CONSTRAINT fk_Ranks_rankSourceID FOREIGN KEY (rankSourceID)
REFERENCES [Project.RankSource] (rankSourceID)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT fk_Ranks_gradPrgID FOREIGN KEY (gradPrgID)
REFERENCES [Project.GraduateProgram] (gradPrgID)
ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE [Project.Teaches] (
gradPrgID CHAR (20) NOT NULL,
facultyID CHAR(20) NOT NULL,
numberOfCourses INT,
CONSTRAINT pk_Teaches_gradPrgID_facultyID PRIMARY KEY(gradPrgID,facultyID),
CONSTRAINT fk_Teaches_gradPrgID FOREIGN KEY (gradPrgID)
REFERENCES [Project.GraduateProgram] (gradPrgID)
ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT fk_Teaches_facultyID FOREIGN KEY (facultyID)
REFERENCES [Project.Faculty] (facultyID)
ON DELETE NO ACTION ON UPDATE NO ACTION,
);

```

```

CREATE TABLE [Project.Uses] (
rankSourceID CHAR (20) NOT NULL,
criteriaID CHAR(20) NOT NULL,
criteriaWeightage DECIMAL(5,2),
CONSTRAINT pk_Uses_rankSourceID_criteria_ID primary key(rankSourceID,criteriaID),
CONSTRAINT fk_Uses_rankSourceID FOREIGN KEY (rankSourceID)
REFERENCES [Project.RankSource] (rankSourceID)
ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT fk_Uses_criteriaID FOREIGN KEY (criteriaID)
REFERENCES [Project.RankingCriteria] (criteriaID)
ON DELETE NO ACTION ON UPDATE NO ACTION,
);

```

```

-- INSERT Statements
--Insert for GraduateProgram table
INSERT INTO [Project.GraduateProgram] VALUES
('GP001','Information Systems','MS','Covers a range of courses related to
information systems, technology management, and business',18,70000,30,'Yes')

```

```

INSERT INTO [Project.GraduateProgram] VALUES
('GP002','Business Analytics','MS','Covers fundamental principles of data analysis,
including statistical methods, data visualization, and interpretation of
results',18,65000,30,'Yes')

```

```

INSERT INTO [Project.GraduateProgram] VALUES
('GP003','Marketing Analytics','MS','Covers basic principles of marketing
analytics, including data collection, analysis, and interpretation for marketing
purposes',18,60000,30,'Yes')

```

```

INSERT INTO [Project.GraduateProgram] VALUES

```

```
('GP004','Supply chain management','MS','Introduces the basic principles and concepts of supply chain management, including the flow of goods, information, and finances across the supply chain.',24,62500,30,'Yes')
```

```
INSERT INTO [Project.GraduateProgram] VALUES  
('GP005','Business Administration','MBA','Prepare students for leadership roles in various industries by imparting a broad understanding of business concepts, strategic thinking, and management skills',24,90000,54,'No')
```

--Insert for RankSource Table

```
INSERT INTO [Project.RankSource] VALUES  
('RS001','U.S. News & World Report','Provide rankings for a variety of fields, including business, law, engineering, education, and more','https://www.usnews.com/best-graduate-schools')
```

```
INSERT INTO [Project.RankSource] VALUES  
('RS002','QS World University Rankings','Rankings cover a variety of subjects, including business, engineering, social sciences, and more','https://www.topuniversities.com/qs-world-university-rankings')
```

```
INSERT INTO [Project.RankSource] VALUES  
('RS003','U.S. News & World Report','Reviews of graduate programs, often highlighting specific aspects such as academic quality, financial aid, and campus life','https://www.princetonreview.com/business-school')
```

```
INSERT INTO [Project.RankSource] VALUES  
('RS004','Times Higher Education (THE) World University Rankings','Rankings consider factors such as teaching, research, international outlook, and industry income','https://www.timeshighereducation.com/world-university-rankings')
```

```
INSERT INTO [Project.RankSource] VALUES  
('RS005','QS Business Masters Rankings','Provides rankings specifically for business master's programs','https://www.topuniversities.com/university-rankings/business-masters-rankings')
```

--INSERT For Faculty table

```
INSERT INTO [Project.Faculty] VALUES  
('FAC001','John','A','Bono','EdD','10-03-2001','Associate Academic Director, MS Information Systems',14)
```

```
INSERT INTO [Project.Faculty] VALUES  
('FAC002','Sujin','B','Kim','PhD','07-19-2012','Associate Clinical Professor',12)
```

```
INSERT INTO [Project.Faculty] VALUES  
('FAC003','Woei-jyh','A','Lee','PhD','08-11-2012','Associate Clinical Professor',18)
```

```
INSERT INTO [Project.Faculty] VALUES  
('FAC004','Siva','I','Viswanathan','PhD','01-05-2001','Dean's Professor of Information Systems',16)
```

```
INSERT INTO [Project.Faculty] VALUES  
('FAC005','Tejwansh','S','Anand','PhD','08-20-2021','Academic Director, MS in Information Systems',18)
```

```

--Insert for RankingCriteria table

INSERT INTO [Project.RankingCriteria] VALUES
('CR001','Academic quality','Quality of course content')

INSERT INTO [Project.RankingCriteria] VALUES
('CR002','Industry income','Range of income of graduating students')

INSERT INTO [Project.RankingCriteria] VALUES
('CR003','Research','Based on the quantity and quality of research papers
published')

--Insert for Rank Table

INSERT INTO [Project.Ranks] VALUES
('GP001','RS001',2022,8)

INSERT INTO [Project.Ranks] VALUES
('GP002','RS002',2023,10)

INSERT INTO [Project.Ranks] VALUES
('GP003','RS003',2021,15)

--Insert for Teaches Table

INSERT INTO [Project.Teaches] VALUES
('GP001','FAC001',1)

INSERT INTO [Project.Teaches] VALUES
('GP002','FAC002',1)

INSERT INTO [Project.Teaches] VALUES
('GP001','FAC005',3)

--Insert for Uses Table

INSERT INTO [Project.Uses] VALUES
('RS001','CR001',0.4)

INSERT INTO [Project.Uses] VALUES
('RS002','CR002',0.3)

INSERT INTO [Project.Uses] VALUES
('RS003','CR003',0.55)

-- SELECT statements

-- What is the latest rank of all the programs offered at Robert H Smith School of
Business
GO
DROP VIEW IF EXISTS Latest_Rank_View
GO
CREATE VIEW Latest_Rank_View AS
SELECT gp.gradPrgName AS 'Graduate Program',
rs.rankSourceName AS 'Rank Source',
r.rankYear AS 'Year',
r.overallRank AS 'Rank'
FROM [Project.GraduateProgram] gp

```

```

JOIN [Project.Ranks] r ON gp.gradPrgID = r.gradPrgID
JOIN [Project.RankSource] rs ON r.rankSourceID = rs.rankSourceID
INNER JOIN (
    SELECT gradPrgID, rankSourceID, MAX(rankYear) as maxYear
    FROM [Project.Ranks]
    GROUP BY gradPrgID, rankSourceID
) as latestRanks ON r.gradPrgID = latestRanks.gradPrgID
AND r.rankSourceID = latestRanks.rankSourceID
AND r.rankYear = latestRanks.maxYear
GO
SELECT * FROM Latest_Rank_View
ORDER BY 'Graduate Program', 'Rank Source'

```

--Which program has the highest ranking across all sources in the year 2023?

```

GO
DROP VIEW IF EXISTS Highest_Rank_View
GO
CREATE VIEW Highest_Rank_View AS
SELECT gp.gradPrgName AS 'Graduate Program',
r.rankYear AS 'Year',
r.overallRank AS 'Best Rank'
FROM [Project.GraduateProgram] gp
JOIN [Project.Ranks] r ON gp.gradPrgID = r.gradPrgID
WHERE r.rankYear = 2023 AND r.overallRank = (
    SELECT MIN(overallRank)
    FROM [Project.Ranks]
    WHERE rankYear = 2023
)
GO
SELECT * FROM Highest_Rank_View
ORDER BY 'Graduate Program'

```

--What is the historical ranking of Business Administration program over the years?

```

GO
DROP VIEW IF EXISTS BusinessAdmin_View
GO
CREATE VIEW BusinessAdmin_View AS
SELECT gradPrgName AS 'Graduate Program',
rs.rankSourceName AS 'Rank Source',
r.rankYear AS 'Year',
r.overallRank AS 'Rank'
FROM [Project.GraduateProgram] gp
JOIN [Project.Ranks] r ON gp.gradPrgID = r.gradPrgID
JOIN [Project.RankSource] rs ON r.rankSourceID = rs.rankSourceID
WHERE gp.gradPrgName = 'Business Administration'
GO
SELECT * FROM BusinessAdmin_View
ORDER BY 'Year', 'Rank Source'

```

--Who are the faculty members teaching in the programs ranked below 25 in the year 2022 and how many courses do they teach?

```

GO
DROP VIEW IF EXISTS Top_Rank_Faculty_View
GO
CREATE VIEW Top_Rank_Faculty_View AS
SELECT gp.gradPrgName AS 'Graduate Program',
MinRank.ProgramRank AS 'Rank',
CONCAT(f.facultyFirstName, ' ', f.facultyMiddleName, ' ', f.facultyLastName) AS
'Faculty Name',

```

```

t.numberofCourses AS 'Number of Courses'
FROM [Project.Faculty] f
JOIN [Project.Teaches] t ON f.facultyID = t.facultyID
JOIN [Project.GraduateProgram] gp ON t.gradPrgID = gp.gradPrgID
JOIN (
    SELECT gradPrgID, MIN(overallRank) as ProgramRank
    FROM [Project.Ranks]
    WHERE rankYear = 2022
    GROUP BY gradPrgID
) MinRank ON gp.gradPrgID = MinRank.gradPrgID
WHERE MinRank.ProgramRank <= 25
GO
SELECT * FROM Top_Rank_Faculty_View
ORDER BY 'Graduate Program','Faculty Name'

```

--What are the specific criteria used by each ranking source to evaluate the program?

```

GO
DROP VIEW IF EXISTS Ranking_Criteria_View
GO
CREATE VIEW Ranking_Criteria_View AS
SELECT rs.rankSourceName AS 'Rank Source',
rc.criteriaName AS 'Criteria Name',
rc.criteriaDescription AS 'Criteria Description'
FROM [Project.RankSource] rs
JOIN [Project.Uses] u ON rs.rankSourceID = u.rankSourceID
JOIN [Project.RankingCriteria] rc ON u.criteriaID = rc.criteriaID
GO
SELECT * FROM Ranking_Criteria_View
ORDER BY 'Rank Source','Criteria Name'

```

--Which year did the programs achieve their best rank and what was the rank?

```

GO
DROP VIEW IF EXISTS GraduateProgram_Best_Rank_View
GO
CREATE VIEW GraduateProgram_Best_Rank_View AS
SELECT gp.gradPrgName AS 'Graduate Program',
r.rankYear AS 'Year',
pr.BestRank AS 'Best Rank'
FROM [Project.GraduateProgram] gp
JOIN (
    SELECT gradPrgID, MIN(overallRank) as BestRank
    FROM [Project.Ranks]
    GROUP BY gradPrgID
) pr ON gp.gradPrgID = pr.gradPrgID
JOIN [Project.Ranks] r ON gp.gradPrgID = r.gradPrgID AND pr.BestRank =
r.overallRank
GO
SELECT * FROM GraduateProgram_Best_Rank_View
ORDER BY 'Graduate Program'

```

--When was the most significant change in rankings observed for any graduate program?

```

GO
DROP VIEW IF EXISTS GraduateProgram_Rank_Change_View
GO
CREATE VIEW GraduateProgram_Rank_Change_View AS
SELECT TOP 1 gp.gradPrgName AS 'Graduate Program',
r1.rankYear AS 'Current Year',

```

```

r2.rankYear AS 'Previous Year',
r1.overallRank AS 'Current Rank',
r2.overallRank AS 'Previous Rank',
ABS(r1.overallRank - r2.overallRank) AS 'Rank Change'
FROM [Project.Ranks] r1
JOIN [Project.Ranks] r2 ON r1.gradPrgID = r2.gradPrgID AND r1.rankYear =
r2.rankYear + 1
JOIN [Project.GraduateProgram] gp ON r1.gradPrgID = gp.gradPrgID
GO
SELECT * FROM GraduateProgram_Rank_Change_View
ORDER BY 'Rank Change' DESC;

```

--What are the ranking criterias used by the source when Information Systems Program was ranked the worst?

```

GO
DROP VIEW IF EXISTS GraduateProgram_Rank_Criteria_View
GO
CREATE VIEW GraduateProgram_Rank_Criteria_View AS
SELECT rs.rankSourceName AS 'Rank Source',
rc.criteriaName AS 'Criteria Name',
rc.criteriaDescription AS 'Criteria Description'
FROM [Project.Ranks] r
JOIN [Project.GraduateProgram] gp ON r.gradPrgID = gp.gradPrgID
JOIN [Project.RankSource] rs ON r.rankSourceID = rs.rankSourceID
JOIN [Project.Uses] u ON rs.rankSourceID = u.rankSourceID
JOIN [Project.RankingCriteria] rc ON u.criteriaID = rc.criteriaID
WHERE gp.gradPrgName = 'Information Systems'
AND r.overallRank = (
    SELECT MAX(overallRank)
    FROM [Project.Ranks] rk
    JOIN [Project.GraduateProgram] g ON rk.gradPrgID = g.gradPrgID
    WHERE g.gradPrgName = 'Information Systems'
)
GO
SELECT * FROM GraduateProgram_Rank_Criteria_View
ORDER BY 'Criteria Name'

```