**Project proposal and data**

**Q1) State mission statement(s) and mission objectives for the client.**

**Mission statements for Client: -**

1. To provide optimum solutions to the student queries based on student preferences and priorities.
2. Aid students to understand which course was ranked at what position over the years.

**Mission Objectives for Client: -**

1. Be as efficient as possible in providing optimal solutions to as many students as possible.
2. Help students analyze not only the historical rankings specified by different sources but also other important factors that the students can look into while deciding on a course.
3. Based on alumni and research database — >provide students with appropriate course to choose based on his choice.
4. Solving the queries of students if they want to understand how the rankings of different department change overtime.
5. Generate maximum revenue by increasing the traffic of the website developed.

**Q2) Finalize ER schema and diagram.**

**ER Schema:**

**Entities, Attributes and Primary Keys:**

Course (**courseId**, courseName, courseType, courseDuration)

Subject (**subId**, subName)

Professor (**profId**, profName, profResearchArea, profHighestDegree)

Source (**sourceId**, sourceName)

Alumni (**alumMailId**, alumName, alumCompany, alumSalary, alumGradYear, alumPlacedDate)

Publication (**publicationId,** publicationName, publicationDate)

**Relationships, Attributes, Degrees, Participating Entities and Constraints:**

Attended: binary relationship

1 Alumni to 1 Course

1 Course to 1 or more Alumni

    Ranking (rYear, rRank, rBatchSize, rGradRate): binary relationship

1 Course to 1 or more Source

1 Source to 1 or more Course

    Teaches (tYear): ternary relationship

1 Professor and 1 Course to 0 or more Subjects

1 Professor and 1 Subject to 1 or more Courses

1 Subject and 1 Course to 1 Professor

     Research: ternary relationship

1 Publication and 1 Professor to 1 or more Alumni

1 Publication and 1 Alumni to 1 or more Professor

1 Alumni and 1 Professor to 0 or more Publication

Relations:

Course (**courseId**, courseName, courseType, courseDuration)

Subject (**subId**, subName)

Professor (**profId**, profName, profResearchArea, profHighestDegree)

Source (**sourceId**, sourceName)

Alumni (**alumId**, alumName, alumCompany, alumSalary, alumGradYear, alumPlacedDate)

Publication(**publicationId,** publicationName, publicationDate)

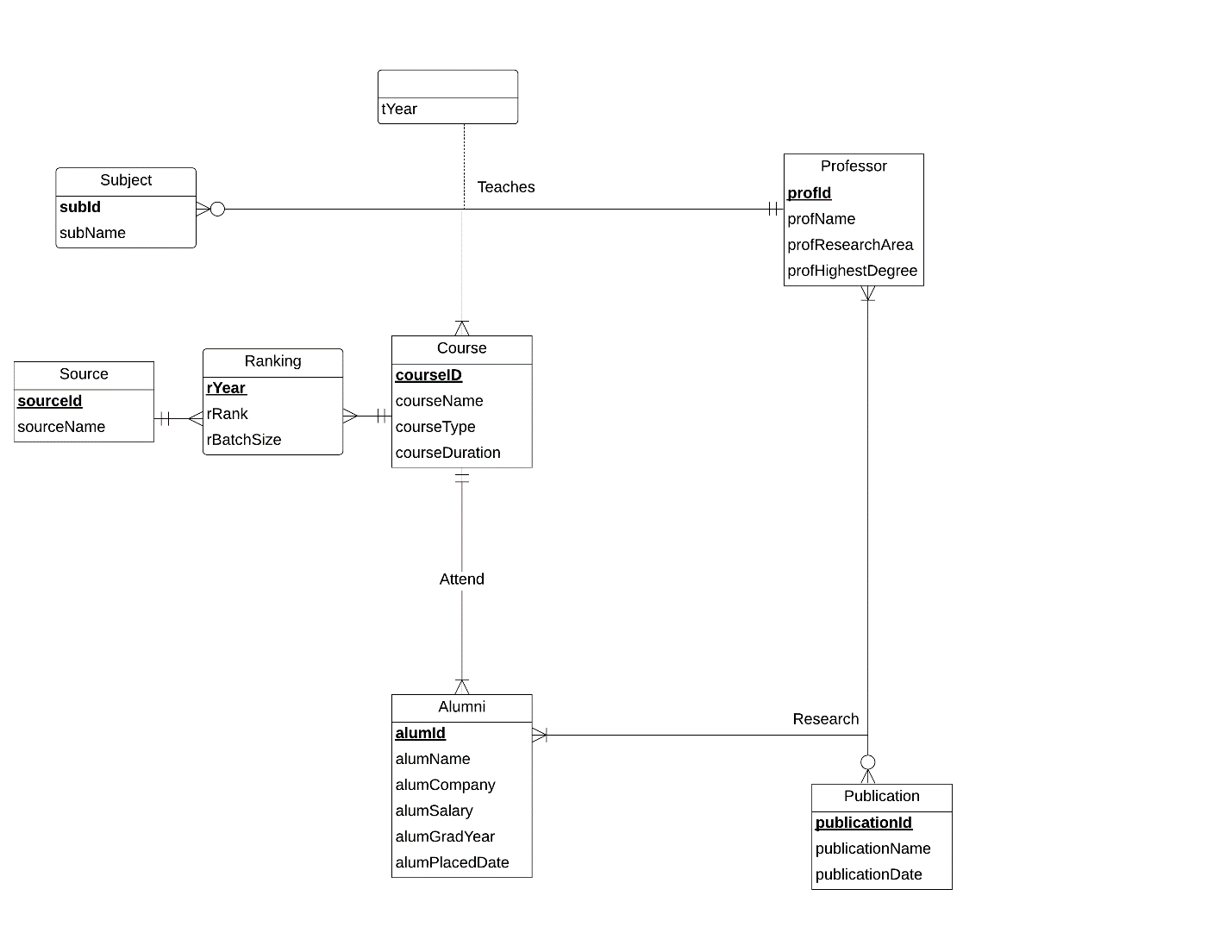
Attended(***courseId, alumId****,* attyear**)**

Ranking(***courseId***, ***sourceId***, **rYear**, rRank, rBatchSize, rGradRate)

Teaches(***profId***, ***courseId***, ***subId***, tYear)

Research(***profId*, *publicationId*, *alumId***)

**ER diagram**



**Q3) Convert ER model into relational schema and identify primary and foreign keys.**

**Relationships, Attributes, Degrees, Participating Entities and Constraints:**

Attended (attYear): binary relationship

1 Alumni to 1 Course

1 Course to 1 or more Alumni

      Ranking (rYear, rRank, rBatchSize, rGradRate): binary relationship

1 Course to 1 or more Source

1 Source to 1 or more Course

     Teaches (tYear): ternary relationship

1 Professor and 1 Course to 0 or more Subjects

1 Professor and 1 Subject to 1 or more Courses

1 Subject and 1 Course to 1 Professor

     Research (researchDate, -month, -day, -year): ternary relationship

1 Publication and 1 Professor to 1 or more Alumni

1 Publication and 1 Alumni to 1 or more Professor

1 Alumni and 1 Professor to 0 or more Publication

**Relations:**

Course (**courseId**, courseName, courseType, courseDuration)

Subject (**subId**, subName)

Professor (**profId**, profName, profResearchArea, profHighestDegree)

Source (**sourceId**, sourceName)

Alumni (**alumId**, alumName, alumCompany, alumSalary, alumGradYear, alumPlacedDate)

Publication (**publicationId,** publicationName, publicationDate)

Attended (***courseId, alumId*,** attyear**)**

Ranking (***courseId***, ***sourceId***, **rYear**, rRank, rBatchSize, rGradRate)

Teaches (***profId***, ***courseId***, ***subId***, tYear)

Research (***profId*, *publicationId*, *alumId***)

**Q4) Determine functional dependencies and perform normalization to 3NF.**

**Functional Dependency:**

courseId—> courseName, courseType, courseDuration

subId —> subName

profId —> profName, profResearchArea, profHighestDegree

sourceId —> sourceName

alumId —> alumName, alumCompany, alumSalary, alumGradYear, alumPlacedDate

publicationId —> publicationName, publicationDate

courseId, alumId—> attyear

courseId, sourceId, rYear —> rRank, rBatchSize, rGradRate

profId, courseId, subId —> tYear

profid, publicationId, alumId —>

**Normalization:**

Course (**courseId**, courseName, courseType, courseDuration) -3NF

Subject (**subId**, subName) -3NF

Professor (**profId**, profName, profResearchArea, profHighestDegree) -3NF

Source (**sourceId**, sourceName) -3NF

Alumni (**alumId**, alumName alumCompany, alumSalary, alumGradYear, alumPlacedDate) -3NF

Publication (**publicationId,** publicationName, publicationDate) -3NF

Attended (***courseId, alumId* )**-3NF

Ranking (***courseId***, ***sourceId***, **rYear**, rRank, rBatchSize, rGradRate) -3NF

Teaches (***profId***, ***courseId***, ***subId***, tYear) -3NF

Research (***profId*, *publicationId*, *alumId***)-3NF

For us all relations were already in 3NF form and thus there were no changes needed in terms of normalization.

**Q5) Generate business rules and determine referential integrity actions.**

**Business Rules:**

1. When a course is discontinued or changed then the corresponding alumni records are not deleted or updated from the database.

2. When alumni recognizing details are deleted from the database then corresponding attendance details are also deleted.

3. When alumni recognizing details are updated in the database then corresponding attendance details are also updated in the database.

4. When identification details of the course are changed or removed then the corresponding ranking records are updated or deleted as well.

5. When the source’s website is shut down then the corresponding ranking records are removed from the database.

6.  The ranking records in the database cannot be updated if there are corresponding source record in the database.

7.  If a professor quits and their details are removed from the database then the corresponding teaching records are removed from the database.

8.  If a professor details are changed then the corresponding teaching records are updated as well.

9. If the course details are deleted or updated from the database then it corresponding teaching details are changed accordingly.

10.  If the subject is discontinued or there is a change in its subject identification details then the teaching records would reflect the same change.

11.  If a professor resigns from the university then the corresponding research records are not deleted.

12.  When the professor records are changed then the corresponding research records are also updated in the database.

13.  When the publication details are removed or changed then the corresponding research records are updated or deleted.

14.  If the alumni details are removed from the database then the corresponding research records are not removed.

15.  If the alumni details are updated in the database then the corresponding research details are updated accordingly.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Relation | Foreign Key | Base Relation | Primary Key | Business Rule | Constraint: ON DELETE | Business Rule | Constraint: ON UPDATE |
| ATTENDED | courseId | Course | courseId | R5 | NO ACTION | R5 | NO ACTION |
| ATTENDED | alumId | Alumni | alumId | R6 | CASCADE | R7 | CASCADE |
| RANKING | courseId | Course | courseId | R8 | CASCADE | R8 | CASCADE |
| RANKING | sourceId | Source | sourceId | R9 | CASCADE | R10 | NO ACTION |
| TEACHES | profId | Professor | profId | R14 | CASCADE | R15 | CASCADE |
| TEACHES | courseId | Course | course | R16 | CASCADE | R16 | CASCADE |
| TEACHES | subId | Subject | subId | R17 | CASCADE | R17 | CASCADE |
| RESEARCH | profId | Professor | profId | R18 | NO ACTION | R19 | CASCADE |
| RESEARCH | pubIicationId | Publication | publicationId | R20 | CASCADE | R20 | CASCADE |
| RESEARCH | alumId | Alumni | alumId | R21 | NO ACTION | R22 | CASCADE |

**Q6) Describe sample data for every relation.**

Course (‘100054’,’Information System’,’Masters’,’16 months’)

Subject (‘100345’,’Data Mining’)

Professor (‘987654’,’Adam Lee’,’Data analysis’,’Phd’)

Source (‘111111’,’US News’)

Alumni (‘222222’,’Michael Jordan’,’EY’,’70000USD’,’2019’, ‘21st September 2019’)

Publication (‘345678’,’Bitcoin analysis’,’23rd october 2019’)

Attended (‘100054’,’222222’)

Ranking (‘100054’,’111111’,’2017’,’9’,’122’,85%’)

Research (‘987654’,’345678’,’222222’)

**DDL**

**SQL Create and drop -**

DROP TABLE [RESEARCH];

DROP TABLE [TEACHES];

DROP TABLE [RANKING];

DROP TABLE [ATTENDED];

DROP TABLE [PUBLICATION];

DROP TABLE [ALUMNI];

DROP TABLE [SOURCE];

DROP TABLE [PROFESSOR];

DROP TABLE [SUBJECT];

DROP TABLE [COURSE];

CREATE TABLE [COURSE](

courseId CHAR(6) NOT NULL,

     courseName VARCHAR(50),

courseType VARCHAR(15),

courseDuration TINYINT,

PRIMARY KEY (courseId)

);

CREATE TABLE [SUBJECT](

subId CHAR(6) NOT NULL,

subName VARCHAR (50),

PRIMARY KEY (subId)

);

CREATE TABLE [PROFESSOR](

profId CHAR(6) NOT NULL,

 profName VARCHAR(80),

profResearchArea VARCHAR(80),

profHighestDegree VARCHAR(80),

PRIMARY KEY(profId)

);

CREATE TABLE [SOURCE](

sourceId CHAR(6) NOT NULL,

sourceName VARCHAR(20),

PRIMARY KEY(sourceId)

);

CREATE TABLE ALUMNI(

alumId CHAR(6) NOT NULL,

alumName VARCHAR(20),

alumCompany VARCHAR(15),

alumSalary INTEGER,

alumGradYear DATE,

alumPlacedDate DATE,

PRIMARY KEY (alumId)

);

CREATE TABLE PUBLICATION(

publicationId CHAR(6) NOT NULL,

publicationName VARCHAR(80),

publicationDate DATE,

PRIMARY KEY (publicationId)

);

-------------------

CREATE TABLE ATTENDED(

coursed CHAR(6) NOT NULL,

alumId CHAR(6) NOT NULL,

CONSTRAINT pk\_Attended\_courseId\_alumId PRIMARY KEY(courseId, alumId),

CONSTRAINT fk\_Attended\_courseId FOREIGN KEY (courseId)

REFERENCES [COURSE] (courseId)

ON DELETE NO ACTION ON UPDATE NO ACTION ,

CONSTRAINT fk\_Attended\_alumId FOREIGN KEY (alumId)

REFERENCES [ALUMNI] (alumId)

ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE RANKING(

courseId CHAR(6) NOT NULL,

sourceId CHAR(6) NOT NULL,

rYear INTEGER NOT NULL,

rRank INTEGER,

rBatchSize INTEGER,

rGradRate VARCHAR(5),

CONSTRAINT pk\_Ranking\_courseId\_sourceId\_ryear PRIMARY KEY(courseId, sourceId, rYear),

CONSTRAINT fk\_Ranking\_courseId FOREIGN KEY (courseId)

REFERENCES [COURSE] (courseId)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT fk\_Ranking\_sourceId FOREIGN KEY (sourceId)

REFERENCES [SOURCE] (sourceId)

ON DELETE CASCADE ON UPDATE NO ACTION

);

CREATE TABLE TEACHES(

profId CHAR(6) NOT NULL,

courseId CHAR(6) NOT NULL,

subId CHAR(6) NOT NULL,

tYear INTEGER,

CONSTRAINT pk\_Teaches\_profId\_courseId\_subId PRIMARY KEY(profId, courseId, subId),

CONSTRAINT fk\_Teaches\_profId FOREIGN KEY (profId)

REFERENCES [PROFESSOR] (profId)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT fk\_Teaches\_courseId FOREIGN KEY (courseId)

REFERENCES [COURSE] (courseId)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT fk\_Teaches\_subId FOREIGN KEY (subId)

REFERENCES [SUBJECT] (subId)

ON DELETE CASCADE ON UPDATE CASCADE

);

CREATE TABLE RESEARCH(

profId CHAR(6) NOT NULL,

publicationId CHAR(6) NOT NULL,

alumId CHAR(6) NOT NULL,

CONSTRAINT pk\_Research\_profId\_publicationId\_alumId PRIMARY KEY(profId, publicationId, alumId),

CONSTRAINT fk\_Research\_profId FOREIGN KEY (profId)

REFERENCES [PROFESSOR] (profId)

ON DELETE NO ACTION ON UPDATE CASCADE,

CONSTRAINT fk\_Research\_publicationId FOREIGN KEY (publicationId)

REFERENCES [PUBLICATION] (publicationId)

ON DELETE CASCADE ON UPDATE CASCADE,

CONSTRAINT fk\_Research\_alumId FOREIGN KEY (alumId)

REFERENCES [ALUMNI] (alumId)

ON DELETE NO ACTION ON UPDATE CASCADE

);

**SQL Insert statements**:

INSERT INTO [COURSE] VALUES

('111111','Information Systems','Masters','16'),

('111112','Business Analytics','Masters','16'),

('111115','Finance','Masters','16'),

('111117','Marketing Analytics','Masters','16'),

('111118','Finance','PHD','80'),

('111124','Finance','MBA','24'),

('111127','Consulting','MBA','24'),

('111128','MBA/MRED DEGREE (MASTER OF REAL ESTATE DEVELOPMENT','Dual Degree','36'),

('111131','Information Systems','Undergraduate','24'),

('111133','Supply Chain Management','Undergraduate','48'),

('111135','Finance','Undergraduate','48');

INSERT INTO [SUBJECT] VALUES

('211111','Database Management System'),

('211113','Managing Digital Business Markets'),

('311111','Operation systems'),

('311112','Business Communication'),

('611111','Financial Management'),

('611116','International Investment '),

('811111','Market-Based Management'),

('811116','Business Communication'),

('456789','Securities analysis'),

('456780','Hedge fund management'),

('911112','Capital Markets'),

('911116','Portfolio Management'),

('141111','Financial Strategy for Corporations '),

('141112','Global Strategy '),

('151111','Fundamentals of Development Finance'),

('151116','Construction Management'),

('181111','Database systems'),

('181112','Systems analysis and design'),

('201111','Carrier Management'),

('201113','Sales Management'),

('231111','Computational Finance'),

('231112','Data modeling in business');

Insert into [PROFESSOR] values

('251110','Adam Lee','Data Science','PHD in Information Systems'),

('251111','Siva Viswanathan','Marketing','PHD in Information Systems'),

('251112','Arsenio Dalton','Business Analysis','PHD in Business Intelligence'),

('251113','Arsenio Dalton','Computer Science','PHD in CSE'),

('251114','Zachery Bailey','Data Science','PHD in Data Algorithms'),

('251115','Jesse Horn','Algorithms','PHD in CSE'),

('251116','Gannon Clarke','Business Analysis','MS in Business Administration'),

('251117','Derek Frye','Finance Management','MS in Finance'),

('251118','Harper Dunn','Marketing Analysis','MBA in Corporate Finance'),

('251119','Walter Cain','Real time Marketing Analysis','PHD in Finance'),

('251120','Perry Knox','Data Manipulation','MS in Information Systems'),

('251121','Brent Madden','Data Science','MS in Business Analytics'),

('251122','Grady Byrd','Human Computer Interaction','MS in Artificial Intelligence'),

('251123','Basil Mullen','Big Data','MS in Computer Science'),

('251124','Galvin Martinez','Data Science','MS in Information Systems'),

('251125','Ignatius Finley','Data Science','PHD in Business Analytics'),

('251126','Zephania Morrison','Computer Science','MS in Computer Science'),

('251127','Cain Whitney','Marketing Data Analysis','PHD in Information Systems'),

('251128','Adrian Monroe','Data Science','PHD in Business Analytics'),

('251129','Malik Mueller','Data Analysis','PHD in Information Systems'),

('251130','Charles Garrison','Business Data Analysis','MS in Business Administration'),

('251140','Hayes Sloan','Computer Science','PHD in Computer Science')

INSERT INTO [ALUMNI] VALUES

('116911', 'Elvina Locks', 'EY', 120000, '05/15/2016', '06/14/2016'),

('116912', 'Nickey Pheasey', 'Acura', 98000, '05/15/2016', '07/17/2016'),

('116913', 'Ki Winn', 'Deloitte', 100000, '05/15/2017', '04/19/2017'),

('116914', 'Leo Swaton', 'PwC', 85000, '05/15/2018', '07/13/2018'),

('116915', 'Dael Crabbe', 'Morgan-Stanley', 221000, '05/15/2018', '06/03/2018'),

('116916', 'Dillie Mantripp', 'LinkedIn', 190000, '05/15/2018', '09/06/2018'),

('116917', 'Martin Kastel', 'Facebook', 156000, '05/15/2017', '08/19/2017'),

('116918', 'Sophie Uccelli', 'Deloitte', 180000, '05/15/2017', '06/14/2017'),

('116919', 'Mikel Donaldson', 'Google', 105000, '05/15/2016','07/17/2016'),

('116920', 'Garv Saxena', 'Apple', 86000, '05/15/2018', '08/26/2018'),

('116921', 'Steve Rogers', 'PwC', 95000, '05/15/2016', '10/13/2016');

INSERT INTO [PUBLICATION] VALUES

('451111','Bitcoin analysis','10/23/2019'),

('451112','Public finance in theory and practice','11/21/2019'),

('451113','Performance Measurement for green supply chain management','12/13/2019'),

('451114','Operation systems ','03/15/2019'),

('451115','Claim severity analysis','12/20/2019'),

('451116','AI in Soccer','04/13/2019'),

('451117','3D printing for cheap toilets','12/16/2019'),

('451118','Greenhouse effects on logistics','09/03/2019'),

('451119','Positive Accounting Theory','11/22/2019');

INSERT INTO [SOURCE] VALUES

('351111','US News & Ranking'),

('351112','Economist'),

('351113','Rankings'),

('351114','Forbes');

INSERT INTO [TEACHES] VALUES

('251110','111111','211111',2018),

('251111','111111','211113',2017),

('251112','111112','311111',2018),

('251113','111112','311112',2017),

('251114','111115','611111',2015),

('251115','111115','611116',2017),

('251116','111117','811111',2018),

('251117','111117','811116',2016),

('251118','111118','456789',2015),

('251119','111118','456780',2016),

('251120','111124','911112',2014),

('251121','111124','911116',2015),

('251122','111127','141111',2017),

('251123','111127','141112',2015),

('251124','111128','151111',2011),

('251125','111128','151116',2013),

('251126','111131','181111',2015),

('251127','111131','181112',2017),

('251128','111133','201111',2014),

('251129','111133','201113',2012),

('251130','111135','231111',2014),

('251140','111135','231112',2013);

INSERT INTO [RESEARCH] VALUES

('251110','451111','116911'),

('251111','451112','116912'),

('251112','451113','116913'),

('251113','451114','116914'),

('251114','451115','116915'),

('251115','451116','116916'),

('251116','451117','116917'),

('251117','451118','116918'),

('251118','451119','116919');

INSERT INTO [ATTENDED] VALUES

('111111','116911'),

('111112','116912'),

('111115','116913'),

('111117','116914'),

('111118','116915'),

('111124','116916'),

('111127','116917'),

('111128','116918'),

('111131','116919'),

('111133','116920'),

('111135','116921')

INSERT INTO [RANKING] VALUES

('111111','351111' , 2018, 8, 100, '95%'),

('111111','351112' , 2018, 7, 100, '95%'),

('111111','351113' , 2018, 7, 100, '95%'),

('111111','351114' , 2018, 8, 100, '95%'),

('111111','351111' , 2017, 5, 100, '94%'),

('111111','351112' , 2017, 6, 100, '94%'),

('111111','351113' , 2017, 5, 100, '94%'),

('111111','351114' , 2017, 6, 100, '94%'),

('111111','351111' , 2016, 7, 100, '93%'),

('111111','351112' , 2016, 6, 100, '93%'),

('111111','351113' , 2016, 6, 100, '93%'),

('111111','351114' , 2016, 7, 100, '93%'),

('111111','351111' , 2015, 7, 100, '94%'),

('111111','351112' , 2015, 6, 100, '94%'),

('111111','351113' , 2015, 7, 100, '94%'),

('111111','351114' , 2015, 6, 100, '94%'),

('111112','351111' , 2018, 12, 120, '94%'),

('111112','351112' , 2018, 11, 120, '94%'),

('111112','351113' , 2018, 9, 120, '94%'),

('111112','351114' , 2018, 10, 120, '94%'),

('111112','351111' , 2017, 11, 120, '93%'),

('111112','351112' , 2017, 11, 120, '93%'),

('111112','351113' , 2017, 7, 120, '93%'),

('111112','351114' , 2017, 16, 120, '93%'),

('111112','351111' , 2016, 17, 120, '92%'),

('111112','351112' , 2016, 14, 120, '92%'),

('111112','351113' , 2016, 13, 120, '92%'),

('111112','351114' , 2016, 15, 120, '92%'),

('111112','351111' , 2015, 11, 120, '93%'),

('111112','351112' , 2015, 12, 120, '93%'),

('111112','351113' , 2015, 11, 120, '93%'),

('111112','351114' , 2015, 10, 120, '93%'),

('111115','351111' , 2018, 6, 110, '94%'),

('111115','351112' , 2018, 5, 110, '94%'),

('111115','351113' , 2018, 7, 110, '94%'),

('111115','351114' , 2018, 5, 110, '94%'),

('111115','351111' , 2017, 7, 110, '92%'),

('111115','351112' , 2017, 4, 110, '92%'),

('111115','351113' , 2017, 4, 110, '92%'),

('111115','351114' , 2017, 5, 110, '92%'),

('111115','351111' , 2016, 7, 110, '93%'),

('111115','351112' , 2016, 8, 110, '93%'),

('111115','351113' , 2016, 7, 110, '93%'),

('111115','351114' , 2016, 5, 110, '93%'),

('111115','351111' , 2015, 8, 110, '92%'),

('111115','351112' , 2015, 7, 110, '92%'),

('111115','351113' , 2015, 7, 110, '92%'),

('111115','351114' , 2015, 8, 110, '92%'),

('111117','351111' , 2018, 13, 105, '93%'),

('111117','351112' , 2018, 11, 105, '93%'),

('111117','351113' , 2018, 19, 105, '93%'),

('111117','351114' , 2018, 16, 105, '93%'),

('111117','351111' , 2017, 19, 105, '94%'),

('111117','351112' , 2017, 23, 105, '94%'),

('111117','351113' , 2017, 21, 105, '94%'),

('111117','351114' , 2017, 22, 105, '94%'),

('111117','351111' , 2016, 31, 105, '96%'),

('111117','351112' , 2016, 27, 105, '96%'),

('111117','351113' , 2016, 28, 105, '96%'),

('111117','351114' , 2016, 24, 105, '96%'),

('111117','351111' , 2015, 30, 105, '94%'),

('111117','351112' , 2015, 28, 105, '94%'),

('111117','351113' , 2015, 24, 105, '94%'),

('111117','351114' , 2015, 33, 105, '94%'),

('111118','351111' , 2018, 17, 90, '95%'),

('111118','351112' , 2018, 14, 90, '95%'),

('111118','351113' , 2018, 19, 90, '95%'),

('111118','351114' , 2018, 18, 90, '95%'),

('111118','351111' , 2017, 14, 90, '94%'),

('111118','351112' , 2017, 12, 90, '94%'),

('111118','351113' , 2017, 16, 90, '94%'),

('111118','351114' , 2017, 11, 90, '94%'),

('111118','351111' , 2016, 18, 90, '94%'),

('111118','351112' , 2016, 18, 90, '94%'),

('111118','351113' , 2016, 19, 90, '94%'),

('111118','351114' , 2016, 13, 90, '94%'),

('111118','351111' , 2015, 24, 90, '93%'),

('111118','351112' , 2015, 21, 90, '93%'),

('111118','351113' , 2015, 26, 90, '93%'),

('111118','351114' , 2015, 22, 90, '93%'),

('111124','351111' , 2018, 10, 80, '90%'),

('111124','351112' , 2018, 11, 80, '90%'),

('111124','351113' , 2018, 9, 80, '90%'),

('111124','351114' , 2018, 11, 80, '90%'),

('111124','351111' , 2017, 8, 80, '92%'),

('111124','351112' , 2017, 8, 80, '92%'),

('111124','351113' , 2017, 9, 80, '92%'),

('111124','351114' , 2017, 10,80, '92%'),

('111124','351111' , 2016, 13, 80, '91%'),

('111124','351112' , 2016, 14, 80, '91%'),

('111124','351113' , 2016, 12, 80, '91%'),

('111124','351114' , 2016, 17, 80, '91%'),

('111124','351111' , 2015, 16, 80, '94%'),

('111124','351112' , 2015, 16, 80, '94%'),

('111124','351113' , 2015, 13, 80, '94%'),

('111124','351114' , 2015, 12, 80, '94%'),

('111127','351111' , 2018, 9, 100, '91%'),

('111127','351112' , 2018, 9, 100, '91%'),

('111127','351113' , 2018, 8, 100, '91%'),

('111127','351114' , 2018, 8, 100, '91%'),

('111127','351111' , 2017, 13, 100, '93%'),

('111127','351112' , 2017, 12, 100, '93%'),

('111127','351113' , 2017, 14, 100, '93%'),

('111127','351114' , 2017, 11, 100, '93%'),

('111127','351111' , 2016, 16, 100, '93%'),

('111127','351112' , 2016, 13, 100, '93%'),

('111127','351113' , 2016, 14, 100, '93%'),

('111127','351114' , 2016, 15, 100, '93%'),

('111127','351111' , 2015, 17, 100, '95%'),

('111127','351112' , 2015, 19, 100, '95%'),

('111127','351113' , 2015, 13, 100, '95%'),

('111127','351114' , 2015, 15, 100, '95%'),

('111128','351111' , 2018, 9, 90, '94%'),

('111128','351112' , 2018, 11, 90, '94%'),

('111128','351113' , 2018, 13, 90, '94%'),

('111128','351114' , 2018, 10, 90, '94%'),

('111128','351111' , 2017, 13, 90, '95%'),

('111128','351112' , 2017, 14, 90, '95%'),

('111128','351113' , 2017, 12, 90, '95%'),

('111128','351114' , 2017, 15, 90, '95%'),

('111128','351111' , 2016, 14, 90, '95%'),

('111128','351112' , 2016, 12, 90, '95%'),

('111128','351113' , 2016, 15, 90, '95%'),

('111128','351114' , 2016, 16, 90, '95%'),

('111128','351111' , 2015, 17, 90, '93%'),

('111128','351112' , 2015, 13, 90, '93%'),

('111128','351113' , 2015, 14, 90, '93%'),

('111128','351114' , 2015, 13, 90, '93%'),

('111131','351111' , 2018, 18, 110, '93%'),

('111131','351112' , 2018, 19, 110, '93%'),

('111131','351113' , 2018, 15, 110, '93%'),

('111131','351114' , 2018, 14, 110, '93%'),

('111131','351111' , 2017, 11, 110, '93%'),

('111131','351112' , 2017, 12, 110, '93%'),

('111131','351113' , 2017, 15, 110, '93%'),

('111131','351114' , 2017, 14, 110, '93%'),

('111131','351111' , 2016, 14, 110, '92%'),

('111131','351112' , 2016, 16, 110, '92%'),

('111131','351113' , 2016, 17, 110, '92%'),

('111131','351114' , 2016, 13, 110, '92%'),

('111131','351111' , 2015, 13, 110, '93%'),

('111131','351112' , 2015, 15, 110, '93%'),

('111131','351113' , 2015, 12, 110, '93%'),

('111131','351114' , 2015, 16, 110, '93%'),

('111133','351111' , 2018, 10, 105, '92%'),

('111133','351112' , 2018, 19, 105, '92%'),

('111133','351113' , 2018, 17, 105, '92%'),

('111133','351114' , 2018, 15, 105, '92%'),

('111133','351111' , 2017, 13, 105, '92%'),

('111133','351112' , 2017, 14, 105, '92%'),

('111133','351113' , 2017, 12, 105, '92%'),

('111133','351114' , 2017, 15, 105, '92%'),

('111133','351111' , 2016, 12, 105, '90%'),

('111133','351112' , 2016, 14, 105, '90%'),

('111133','351113' , 2016, 15, 105, '90%'),

('111133','351114' , 2016, 16, 105, '90%'),

('111133','351111' , 2015, 17, 105, '92%'),

('111133','351112' , 2015, 15, 105, '92%'),

('111133','351113' , 2015, 16, 105, '92%'),

('111133','351114' , 2015, 15, 105, '92%'),

('111135','351111' , 2018, 7, 100, '94%'),

('111135','351112' , 2018, 14, 100, '94%'),

('111135','351113' , 2018, 13, 100, '94%'),

('111135','351114' , 2018, 12, 100, '94%'),

('111135','351111' , 2017, 11, 100, '93%'),

('111135','351112' , 2017, 9, 100, '93%'),

('111135','351113' , 2017, 8, 100, '93%'),

('111135','351114' , 2017, 12, 100, '93%'),

('111135','351111' , 2016, 15, 100, '94%'),

('111135','351112', 2016, 13, 100, '94%'),

('111135','351113', 2016, 18, 100, '94%'),

('111135','351114', 2016, 19, 100, '94%'),

('111135','351111', 2015, 16, 100, '93%'),

('111135','351112', 2015, 19, 100, '93%'),

('111135','351113', 2015, 18, 100, '93%'),

('111135','351114', 2015, 17, 100, '93%')

**DML**

**SELECT QUERIES**:

Q1:

---- How has the ranking of masters in Information Systems fluctuated over last four years amongst all the sources?

SELECT Source.sourceName,rYear,rRank FROM [RANKING] INNER JOIN SOURCE on RANKING.sourceId=SOURCE.sourceId

WHERE courseId=(SELECT courseId FROM [COURSE] WHERE courseName='Information Systems' AND courseType='Masters')

Q2:

---- What are the names and details of the professors of the course which is ranked best for course type undergraduate amongst all the sources?

SELECT courseName,courseType,TEACHES.profId,profName,profResearchArea,profHighestDegree

FROM TEACHES INNER JOIN PROFESSOR on PROFESSOR.profId=TEACHES.profId INNER JOIN COURSE on COURSE.courseId=TEACHES.courseId

WHERE Course.courseId in (SELECT distinct(courseId) FROM RANKING  WHERE rRank=(SELECT Min(rRank) FROM RANKING INNER JOIN Course on Course.courseId=RANKING.courseId WHERE Course.courseType=’Undergraduate’))

and courseType=’Undergraduate’

Q3:

---- Who are the alumni which have highest salary among each course types and display their company and salary details?

SELECT alumName,alumCompany,alumSalary,alumGradYear,courseName,Course.courseType FROM ALUMNI INNER JOIN  ATTENDED on ALUMNI.alumId=ATTENDED.alumId INNER JOIN COURSE on COURSE.courseId=ATTENDED.courseId

 INNER JOIN (SELECT courseType,max(alumSalary) as salary FROM ALUMNI INNER JOIN  ATTENDED on ALUMNI.alumId=ATTENDED.alumId INNER JOIN COURSE on COURSE.courseId=ATTENDED.courseId

GROUP BY courseType) as D on D.courseType=COURSE.courseType and D.salary=ALUMNI.alumSalary

Q4:

---- What is the best ranking of each course over the years irrespective of the source, and display all the details of each of those courses?

SELECT ranking.courseId,course.courseName,course.courseType,rRank,sourceName FROM RANKING INNER JOIN COURSE on course.courseId=RANKING.courseId

INNER JOIN SOURCE on SOURCE.sourceId=RANKING.sourceId INNER JOIN (SELECT courseId,min(rRank) as BestRank FROM RANKING

GROUP BY courseId) as D on D.courseId=COURSE.courseId and D.BestRank=RANKING.rRank