1. Introduction and motivation (including the requirements of the organization)

Our organization is focused on providing educational information for Mainland China undergraduate students who wish to study abroad for graduate learning. Therefore, we designed EduSpark which is a graduate programs application database. The database should store information about universities, their programs, standardized tests, employment outcomes, and applicant details. Based on the information, students can get the information they want and also get the data analysis provided by us. We want to offer application support and suggestions to students. Therefore, the database should be able to manage application details, including the application status and necessary documents. We believe that with our help, there’s no obstacles in graduate program application!

1. Design and implementation
2. entities, attributes, and relationships together with any constraints and properties

Entities:

* University
* Uni\_maj (University Majors)
* Program
* Standardized\_Test
* Employment
* Undergra\_univers (Undergraduate Universities)
* Appliers
* Applications

Attributes:

* University: UID, University\_name, Abbreviation, Branch\_unniversity, Region, Location, QSranking
* Uni\_maj: Rank, UID, MID, Major\_name, Major\_type, Teaching\_language, Major\_scale
* Program: Program\_ID, University\_ID, Program\_Name, Early\_DDL, Final\_DDL, Program\_Link, GPA, Recommendation\_Min\_Amount, Recommendation\_Requirement, Tuition\_Fee
* Standardized\_Test: Program\_ID, GRE\_Required, GRE\_Cut\_Off\_Point, TOEFL\_Accepted, TOEFL\_Requirement, IELTS\_Accepted, IELTS\_Requirement
* Employment: Program\_ID, Avg\_Salary\_Per\_Year, Employment\_Rate, Employer\_Satisfaction, Company, Industry
* Undergra\_univers: univer\_id, univer\_name, univer\_city
* Appliers: applier\_id, first\_name, last\_name, email\_address, phone\_number, gender, date\_of\_birth, undergraduate\_program, univer\_id, GPA, toefl\_score, ielts\_score, gre\_score, gmat\_score
* Applications: program\_id, applier\_id, status, date, recom\_id, ps\_id

Relationships:

* University (1) -- (N) Uni\_maj
* University (1) -- (N) Program
* Program (1) -- (1) Standardized\_Test
* Program (1) -- (1) Employment
* Undergra\_univers (1) -- (N) Appliers
* Program (N) -- (N) Appliers (through Applications)

Constraints:

* Primary keys and foreign keys are clearly specified in the schema.
* On delete cascade for foreign keys to maintain referential integrity.

1. ER 图（待周一询问）
2. relational schemas

university(UID, University\_name, Abbreviation, Branch\_unniversity, Region, Location, QSranking)

Primary key: UID

FD: The primary key is UID, and all other attributes are dependent on it.

No MVD.

uni\_maj(Rank, UID, MID, Major\_name, Major\_type, Teaching\_language, Major\_scale)

Primary key: UID and MID

Foreign key: UID, which references the primary key “UID ” in table “university”

FD: The primary key is UID and MID, and all other attributes are dependent on it.

No MVD.

program(Program\_ID, University\_ID, Program\_Name, Early\_DDL, Final\_DDL, Program\_Link, GPA, Recommendation\_Min\_Amount, Recommendation\_Requirement, Tuition\_Fee)

PK: Program\_ID

FK: university\_UID

FD: The primary key is Program\_ID, and all other attributes are dependent on it.

No MVD.

standardized\_test(Program\_ID, GRE\_Required, GRE\_Cut\_Off\_Point, TOEFL\_Accepted, TOEFL\_Requirement, IELTS\_Accepted, IELTS\_Requirement)

PK: Program\_ID

FK: Program\_ID

FD: The primary key is Program\_ID, and all other attributes are dependent on it.

No MVD.

employment(Program\_ID, Avg\_Salary\_Per\_Year, Employment\_Rate, Employer\_Satisfaction, Company, Industry)

PK: Program\_ID

FK: Program\_ID

FD: The primary key is Program\_ID, and all other attributes are dependent on it.

No MVD.

appliers(applier\_id, first\_name, last\_name, email\_address, phone\_number, gender, date\_of\_birth, undergraduate\_program, univer\_id, GPA, toefl\_score, ielts\_score, gre\_score, gmat\_score)

PK: applier\_id

FK: univer\_id

FD: The primary key is applier\_id, and all other attributes are dependent on it.

No MVD.

Undergrad\_univers(univer\_id, univer\_name, univer\_city)

PK: univer\_id

FK: univer\_id

FD: The primary key is univer\_id, and all other attributes are dependent on it.

No MVD.

Applications(program\_id, applier\_id, status, date, recom\_id, ps\_id)

PK: applier\_id and program\_id

FK: program\_id

FD: The primary key is applier\_id and program\_id, and all other attributes are dependent on it.

No MVD.

Normal form: according to the above functional dependencies, the design is in 3NF, as there are no partial or transitive dependencies.

1. Data (See the appendices)
2. Sample queries (Only cover some samples. Others see the appendices.)

贴一两个 SQL 过来

1. Data mining: decision tree

贴决策树

1. Index and hashing

Indexes:

* University(UID): As the primary key, indexing the UID will speed up queries that filter or join based on the university ID. Additionally, it will be helpful for optimizing queries that involve grouping or aggregations based on university information.
* Program(Program\_ID): As the primary key, indexing the Program\_ID will improve performance for queries that analyze data across multiple programs, such as comparing application rates or admission requirements.
* Appliers(applier\_id): As the primary key, indexing the applier\_id will speed up queries that filter or join based on the applicant ID. This index will also be beneficial for queries analyzing applicant demographics, educational backgrounds, or standardized test scores.
* Applications(program\_id, applier\_id): As primary key attributes, indexing both program\_id and applier\_id will enhance the performance of queries that involve filtering or joining based on program\_id and/or applier\_id.

Hashing:

In this database, there are no suitable data fields that would benefit from hashing.

1. Conclusion and self-evaluation

In conclusion, the EduSpark has been carefully designed to efficiently store and manage information related to universities, programs, applicants, and their applications. The schema ensures data integrity and consistency by using primary and foreign keys. The database is normalized to minimize data redundancy and improve data consistency. Analytical SQL queries can be executed efficiently by leveraging indexes on key attributes. By implementing this well-designed database, EduSpark can provide valuable insights and make informed decisions to better serve prospective students in their pursuit of higher education.

Self-evaluation: all group members contribute greatly to this project. Hence, the contribution is equal for everyone.

1. References

Data from: <https://opencs.app>

其他参考可以继续补充

1. Appendices 文件名待补充
2. Sample data: see excel files. We use python to insert those .xlsx files to the database.
3. Sample queries:
4. PPT