Data Warehousing Assignment 1: Data Warehouse Design

Virginia Li

Gabriel Guo

Mathew Gao

Ahmad Faig Igro

Danica Dong

Word Count: 3427

1. Executive Summary

The era of data has urged the need for businesses to take up the data-driven approach for decision making and business planning. Yu (2020) revealed the importance of a data warehouse to modern higher education providers. There was a sharp increase in total sales generated from data warehouse services from 3.5 billion in 2003 to 15.5 billion in 2015 (Yu, 2020). As one of the most historical and prestigious higher education providers, Melbourne Business School (MBS) has been offering the intensive one-year full time Master of Business Analytics course over the past 7 years. The competition with other education providers is getting fierce especially after the pandemic. Students are not only honourable customers, but also products. Data speaks louder than language. Evidence shows that the data warehouse not only plays a role in supporting the operation of the business, but also driving revenue (Bhansali, 2007).

In this document, we have developed a graduated student employability management system. We firstly design a database system to record every year's MBS graduate students' employability and career training data. Two data warehouse schemas are developed based on this database. Through the following tests on different queries with simulated data, we have proved that the system is effective and efficient in answering the business analytics questions.

Our designed data warehouses provide MBS with quantified measures to assess employability of MBS graduates, evaluate performance of various career support methods and provide data support to marketing and branding. Analysts are able to review past performance and conduct variance analysis across years. Based on historical data, trend analysis can be utilized to predict future movement and forecasting long term performance of business. The variance analysis assists MBS to better understand business performance, find out shortcomings and take actions in time. Therefore, MBS can mitigate risk of data mismanagement and reduce costs associated with data processing and analysis. The trend analysis prepares MBS better in expanding the network of corporate members as well as the scale of career support to meet students' demand. Besides data analytics, the designed data warehouse also empowers data visualization. Marketing department is able to utilize the data to demonstrate MBS' strong ability to promote students' employability and thus attract more students to generate revenue.

2. Background on Data Warehousing

Data warehousing is the development and utilization of a data warehouse. The data warehouse is a highly intelligent storage of organized data resources which allows for not only recording and maintenance of data, but also analysis and integration of reports (Amazon Web Services, 2022). To put it simply, a data warehouse can be understood as an intelligent storage of data. The data are organized in files and can be easily extracted and updated if needed.

Online Analytical Processing (OLAP)) databases are techniques specifically designed to support business intelligence use for storing, managing and querying data. A data warehouse is an OLAP database, developed from the relational database (which is the OLTP system). While the data warehouse is an approach that enables users to analyze multiple data dimensions at high speed with large volumes of data, OLTP is the real time execution of database transactions by a larger number of users usually via the internet (Sinha, 2021). OLAP systems are mostly optimized for reading (Stitch, 2022). Extract, Transform, and load (ETL) systems involves processes like cleaning, transforming, combining, and constructing data, and then will be used in a data warehouse.

With the help of a data warehouse, businesses are able to get more insights from historical data, ensure high conformance and quality of data, improve the efficiency of data generation and analytics, and thus drive revenue (Maguire, 2020). Schemas are developed from the database to generate insights from data. By looking at the schema, users can make specified requests of data needed and extract information efficiently (Lane, 2002). Therefore, processing time of data will be shortened dramatically (Maguire, 2020). Additionally, the data warehouse keeps historical data well maintained and empowers users to solve business problems. For instance, the adoption of a data warehouse enables managers and analysts in MBS to efficiently extract key performance indicators to understand employability of students, evaluate staff's performance and formulate a talent database for wider research and marketing purposes. Another value a warehouse can bring is scalability. The data warehouse accelerates the process of data integration and analysis. This provides greater flexibility for businesses to expand and scale up.

3. Business Case

Master of Business Analytics (MBusA) Program records every student's personal information, final grade and current Visa states when they graduate. For career support, MBusA provides a Mentoring Program and a Personal Effectiveness Program (PEP). The PEP is compulsory, and every student is assigned to a career coach throughout the program for training and consulting. If students choose to apply for a Mentoring Program, they will receive one-on-one support from MBusA alumni. Students are required to provide their past degree and work experience including length of work and position information if they have when they are applying for the MBusA Program. The program provides at least one internship opportunity for every student. Teams of students work on a real analytics project for a MBusA Corporate Member. The project is under one of business analytics topics such as Forecasting demand and Service delivery. A supervisor from that company is assigned to each project and supervises the interns doing the project. At the end of the project, the supervisor must provide an evaluation score for all interns, and all interns must report their individual working hours. Once a student graduates, MBusA requires the student to provide the information of offers they have received from companies in different industries. One offer includes offer date, salary and position information. Some offers can be the return offers from previous companies the students work in as interns.

4. Design of a Database

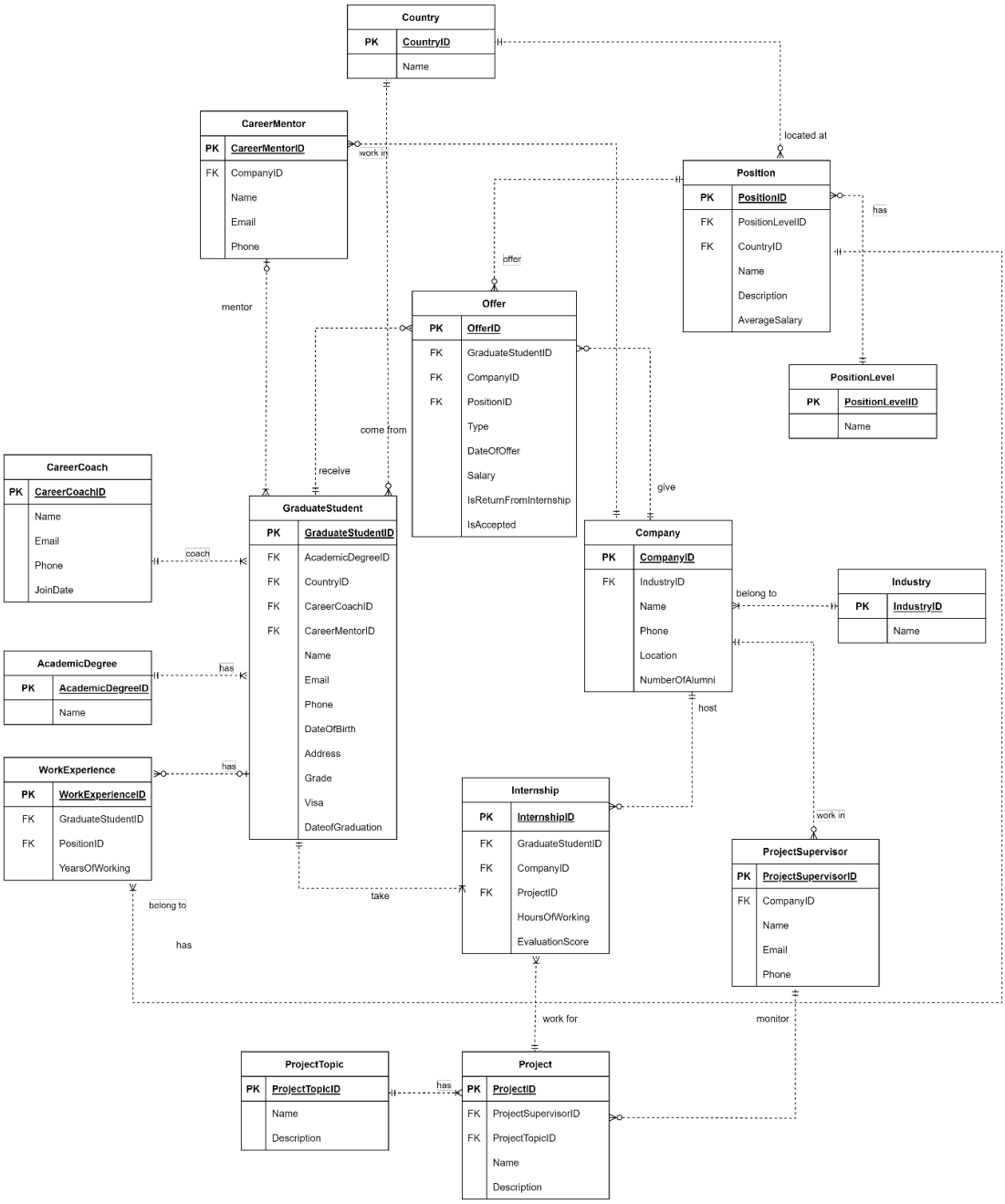

Fig 4.1 ER diagram of the database

There are 15 entities in our database schema (Fig 4.1). The relationship between any two entities is either one to many or many to one. To further clarify some relationships in the ER diagram used for our business case, some other important considerations and assumptions are listed below:

- Since the interest of our business case is to track and evaluate the employment performance of students after graduation, the database should be viewed as a "snapshot" record for already graduated students.
- Only the highest prior education for each student is recorded.
- Every student needs a degree to apply to MBS.
- We expect that all students choose the internship pathway of A-lab in module 4 and thus each graduated student had at least one internship.
- Each position can be categorized into one of the job levels such as entry, intermediate, experienced, advanced and expert, etc.
- MBS has access to the data on the average salary for a position and the alumni counts in a company.

The records in our schema are unique and atomic, our database is in the first normal form. In addition, as we do not have composite primary keys in any of our tables, there is no partial dependency, and our database is in second normal form. Finally, all non-key attributes in our table depend on only the primary key, there is no transitive dependency, and our database is, therefore, the third normal form schema. All function dependencies in our schema are listed below as:

Table name	Functional dependency
AcademicDegree	$\boldsymbol{AcademicDegreeID} \rightarrow Name$
WorkExperience	$\boldsymbol{WorkExperienceExpID} \rightarrow GraduateStudentID, PositionID, YearsOfWorking$
GraduateStudent	$\boldsymbol{GraduateStudentID} \rightarrow AcademicDgreeID, CareerCoachID, CareerMentorID, CountryID, Name, Email, Phone, DateOfBirth, , Address, Grade, Visa, Dateof Graduation$
CareerCoach	$\boldsymbol{CareerCoachID} \rightarrow Name, Email, Phone, JoinDate$
CareerMentor	$\boldsymbol{CareerMentorID} \rightarrow CompanyID, Name, Email, Phone$
Country	$\boldsymbol{CountryID} \rightarrow Name$
Offer	$\boldsymbol{OfferID} \rightarrow GraduateStudentID, CompanyID, PositionID, Type, DateofOffer, Salary, IsReturnFromInternship, IsAccepted$
Position	$\boldsymbol{PositionID} \rightarrow PositionLevelID, CountryID, Name, Description, AverageSalary$
PositionLevel	$\boldsymbol{PositionLevelID} \rightarrow Name$
Internship	$\boldsymbol{InternshipID} \rightarrow GraduateStudentID, CompanyID, ProjectID, HoursOfWorking, EvaluationScore$
Project	$\boldsymbol{ProjectID} \rightarrow ProjectSupervisorID, ProjectTopicID, Name, Description$
ProjectTopic	$\boldsymbol{ProjectTopicID} \rightarrow Name, Description$
ProjectSupervisor	$\boldsymbol{ProjectSupervisorID} \rightarrow CompanyID, Name, Email, Phone$
Company	$\boldsymbol{CompanyID} \rightarrow IndustryID, Name, Location, Phone, NumberOfAlumni$
Industry	$\boldsymbol{IndustryID} \rightarrow Name$

Table 4.1 Functional dependency table of the ER diagram

5. Design of a Data Warehousing

For the design of data warehouse, we followed the four steps of the procedure by Moody and Kortink (Moody and Kortink, 2003):

1. Classify entities into three types:
   1.1 Transaction entity
   Offer is one transaction entity. It records the details about the offers a graduate student has received including the salary offered and the duration of time in weeks between application and receiving the offer.

   Internship is another transaction entity in the ER diagram. It records the graduate students' performance in their internship, which is measured by the evaluation score given by the monitor, and the total hours he/she spent in the project.

   1.2 Component entity
   Graduate Student, Position and Company entities are the three component entities which relate to transaction entity Offer by a one-to-many relationship. These three entities answer the questions: who the recipient is, what position is offered, and what company sends the offer respectively.

   Like the Offer, Internship has three component entities too. They are Graduate Student, Project and Company entities have a one-to-many relationship with Internship. These three entities answer the questions: who takes the internship, what project the internship is under, and what company provides the internship respectively.

   1.3 Classification entity
   Career Mentor, Career Coach, Previous Degree and Country are four classification entities since they have one-to-many relationships with component entity Graduate Student. Based on the same reason: Country and Level are the classification entities of Position; Monitor and Project Topic are two classification entities relating to the component entity Project; Industry is the only classification entity of component entity Company.

2. Design high-level star schema:
   2.1 Identify star schemas
   Offer is one transaction entity. Salary and Waiting Time in Weeks are two important measurements of how effective the MBS's training is on a student's job seeking process. Thus, we make it our first star schema's fact table.

   Internship is another transaction entity. Evaluation score and the total number of hours a student worked on a project are facts that are crucial when analysing the relationship between students' performance in the internship and the speed and quality of the offers they get. Hence, we make it the fact table of our second star schema.

   We also add an attribute "Total_Work_Duration" into the Graduate Student component which is summarized by adding the duration across all previous work experience records a graduate student has.

2.2 Define level of summarisation/granularity

We chose unsummarised (transaction level granularity) for both fact tables. This means that each Offer fact table row corresponds to a single offer, and each Internship fact table row corresponds to a single internship experience. By doing so, we will not lose any information of the facts from the original database. We have much more flexibility for doing analysis such as averaging the salary of all offers for one student or getting the highest score of one student's internship. However, the trade-off/ drawback is that more storage space is needed for storing the data, and the queries for business analysis become slower (Moody and Kortink, 2003). Since our data size is not too big, the drawback is not too impactful. Thus, we go with the transaction level granularity.

2.3 Identify relevant dimensions

All the three component entities of Offer are crucial for potential business analysis. The relevant dimensions for Offer fact table are hence Graduate Student Dimension, Position Dimension, and Company Dimension.

Similarly, all the three component entities relating to Internship are relevant for our purpose of analysis. The relevant dimensions for Offer fact table are hence Graduate Student Dimension, Project Dimension, and Company Dimension.

Time Dimension will also be added to identify different cohorts of MBUSA students. This is to support different types of historical analyses such as comparing the general internship performance of graduate students from different cohorts or analyzing the trend of salary provided by a specific industry across the years.

3. Design detailed fact table

3.1 Define key (for each fact table)

Since we do not have any degenerate dimensions in both fact tables, the keys of all dimension tables consist of the composite key of the fact table.

For Offer fact table, the composite key is consisting of StudentID, ComapnyID, and PositionID. For Offer fact table, the composite key is consisting of StudentID, ComapnyID, and ProjectID.

3.2 Define facts (for each fact table)

The facts for Offer fact table are time duration between graduation and students receiving the offer, which is recorded in number of weeks, and the salary provided in the offer. Time duration is a non-additive fact which cannot be meaningfully added across any dimensions. However, it can be sensibly averaged across Student, Company, Position and Time Dimension. Salary is also a non-additive, but it can be sensibly averaged across all the four dimensions.

The facts for Internship fact table are the evaluation score given by each internship's monitor, and count of working hours. Score is a non-additive fact. However, it can be sensibly averaged across Student, Company, and Time Dimension. Working hours is a semi-additive fact which can only be added across the Graduate Student Dimension. The sum of working hours indicates the total experience the student gained through out all internships.

We chose not to convert the non-additive facts to additive facts. Despite the drawbacks that there might be errors when doing the queries, it is hard to do the transformation, and we do not want to lose the information of salary, time duration, working hours or score in each student's offer and internship (e.g., maximum salary a student gets from multiple offers).

3.3 Offer & Internship Fact Tables

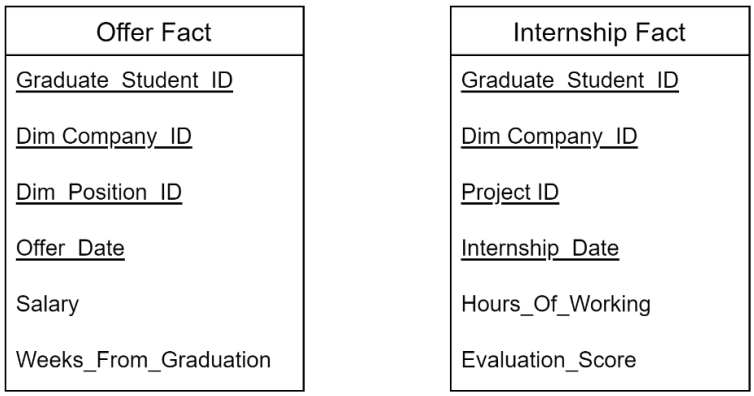

Fig 5.1 Offer and Internship Fact Tables

4. Design detailed dimensional table

4.1 Define dimensional key (slowly changing dimensions will be addressed later)

For Offer fact table, StudentID, ComapnyID, and PositionID are the dimensional keys for Graduate Student Dimension, Company Dimension, and Position Dimension respectively. TimeKey will be the dimensional key for Time Dimension.

For Internship fact table, StudentID, ComapnyID, and ProjectID are the dimensional keys for Graduate Student Dimension, Company Dimension, and Project Dimension respectively. TimeKey will be the dimensional key for Time Dimension.

4.2 Collapse hierarchies

We firstly collapse the hierarchies into the component entities Graduate Student, Company, and Position for star schema having Offer fact table:

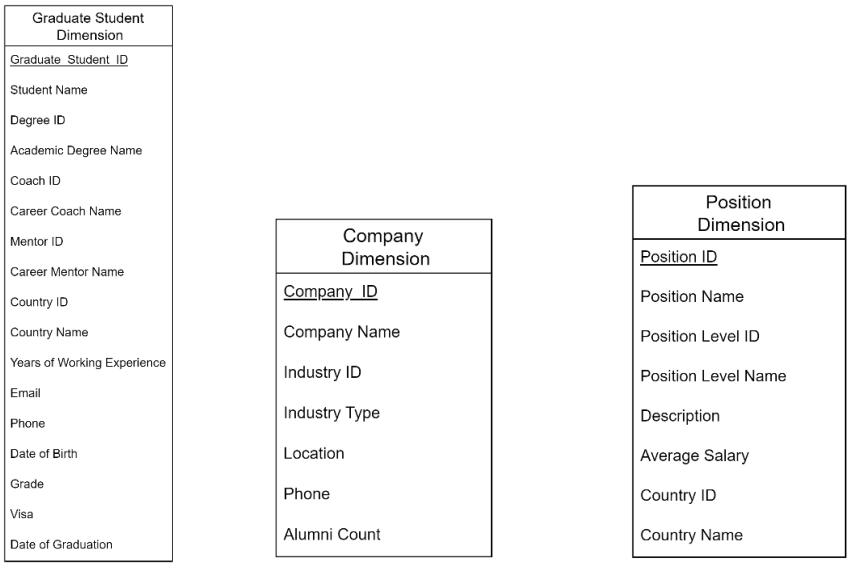

Fig 5.2 Offer Fact Table's Dimension Tables After Collapsing hierarchies

We then collapse the hierarchies into the component entities Graduate Student, Company, and Project for star schema having Internship fact table:

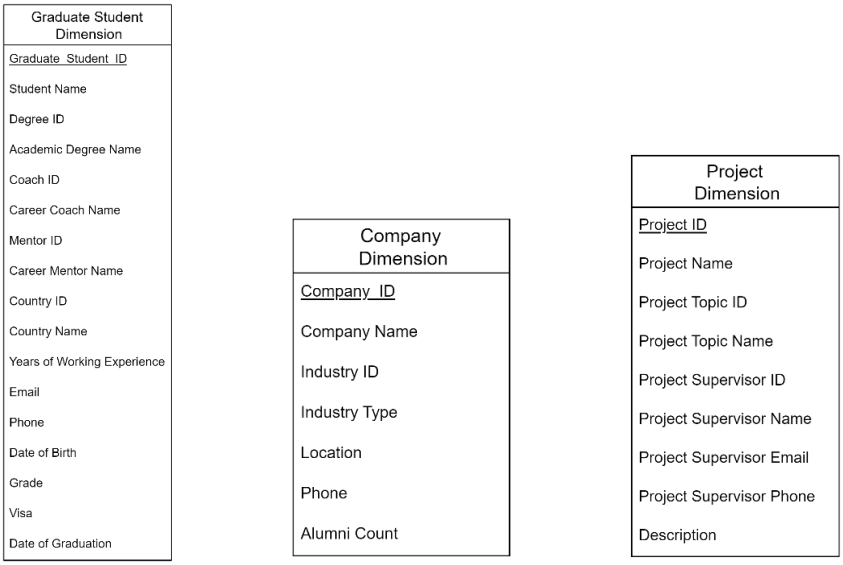

Fig 5.3 Internship Fact Table's Dimension Tables After Collapsing hierarchies

All the resulting dimension tables are in second normal form (2NF).

4.3 Replace codes and abbreviations

Finally, we remove the unnecessary IDs in each dimension tables. Meanwhile, we only keep those attributes which are important for our analysis purpose. Attributes such as Phone and Email are removed from the dimension tables. By doing so, we save even more storage spaces. The resulted dimension tables are shown below:

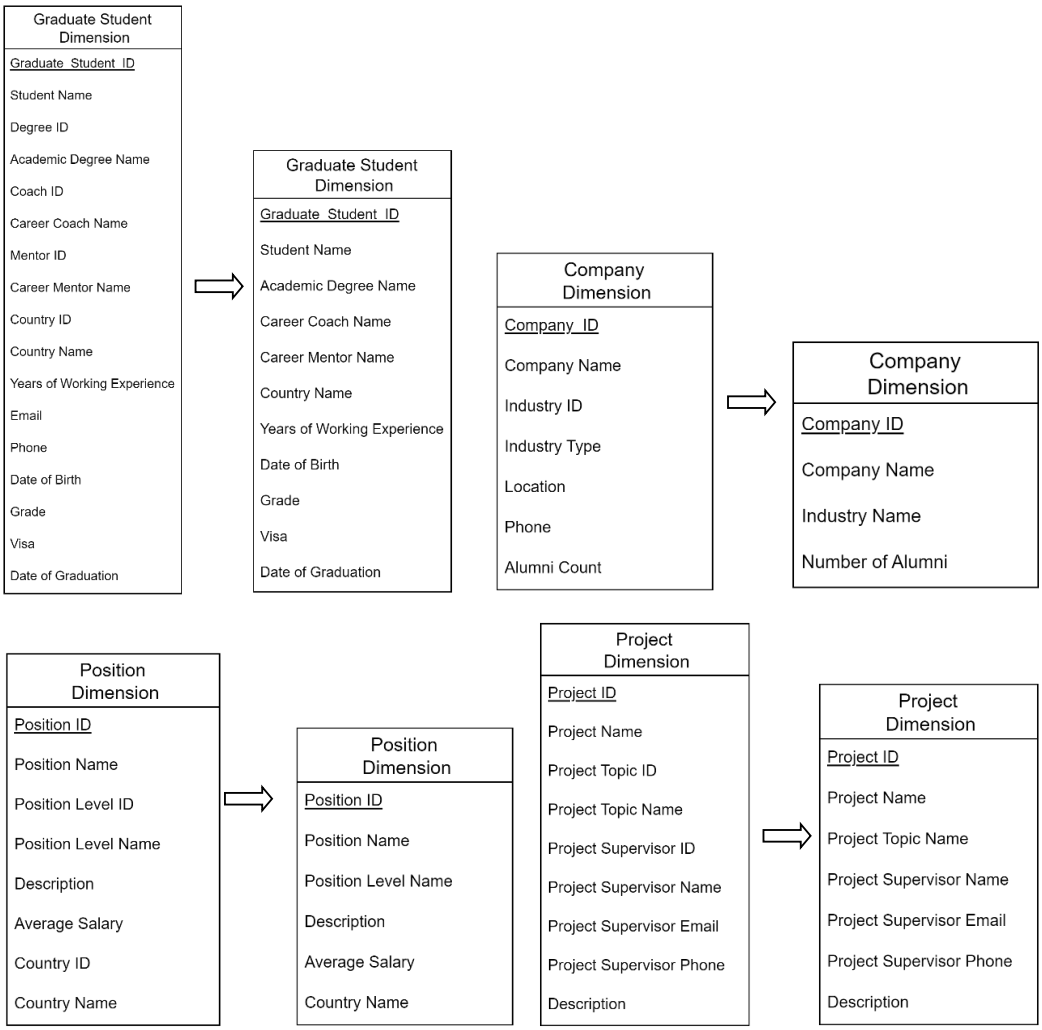

Fig 5.4 Transformation of Dimension Tables

After the four steps of the procedure by Moody and Kortink, we also consider the choices of slowly changing dimensions and mini-dimension problems.

Slowly changing dimensions:

1. We have noticed that Position Dimension's Average Salary and Company Dimension's Alumni Count need to be tracked over time.
2. We decide to create a new row for each update of the attribute mentioned above in the corresponding dimension by adding a version number to the original key of the dimension. The benefit for doing so is obviously that we will not lose any historical data. However, we need more space for storing the new generated data.
3. Instead of using the combination of original dimensional key and the version number as a composite primary key for the slowly changing dimension table, we choose to create an anonymous integer primary key as the surrogate key for every dimension. For example, for the company dimension, we create Company_Dim_ID as the only primary key for this dimension. The utilization of surrogate keys optimizes queries' performance. It allows users to join tables

across multiple dimensions and ensures the correctness in joining the fact table and dimension (Stiglich, 1997). On the other hand, surrogate keys improve the efficiency in construction and maintenance of indexes by reducing elements in the fact table (Stiglich, 1997).

Mini-dimension problems:

In our case, no dimension table is getting woo many rows since the recorded company and students are limited.

Therefore, our final star schemas are:

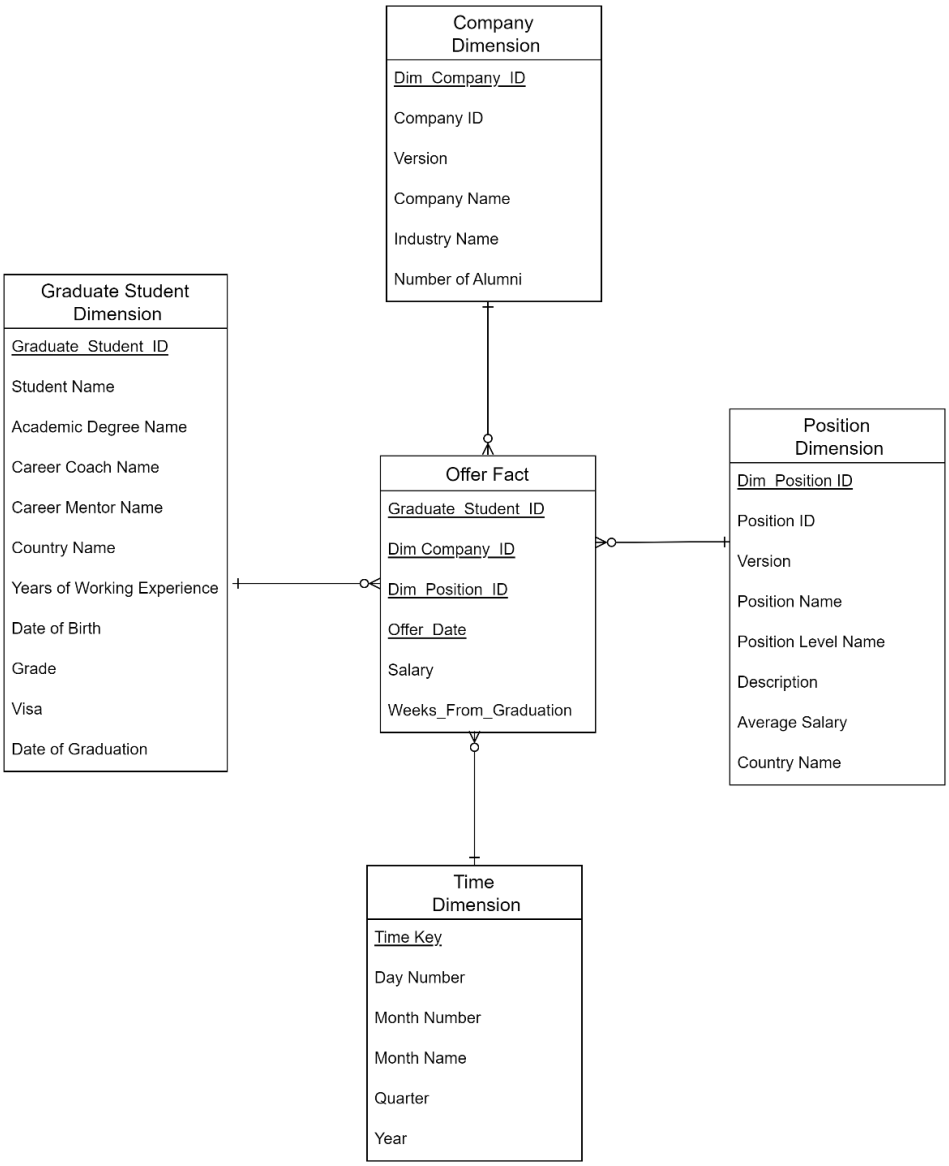

Fig 5.5 Final Star Schema for Offer Facts

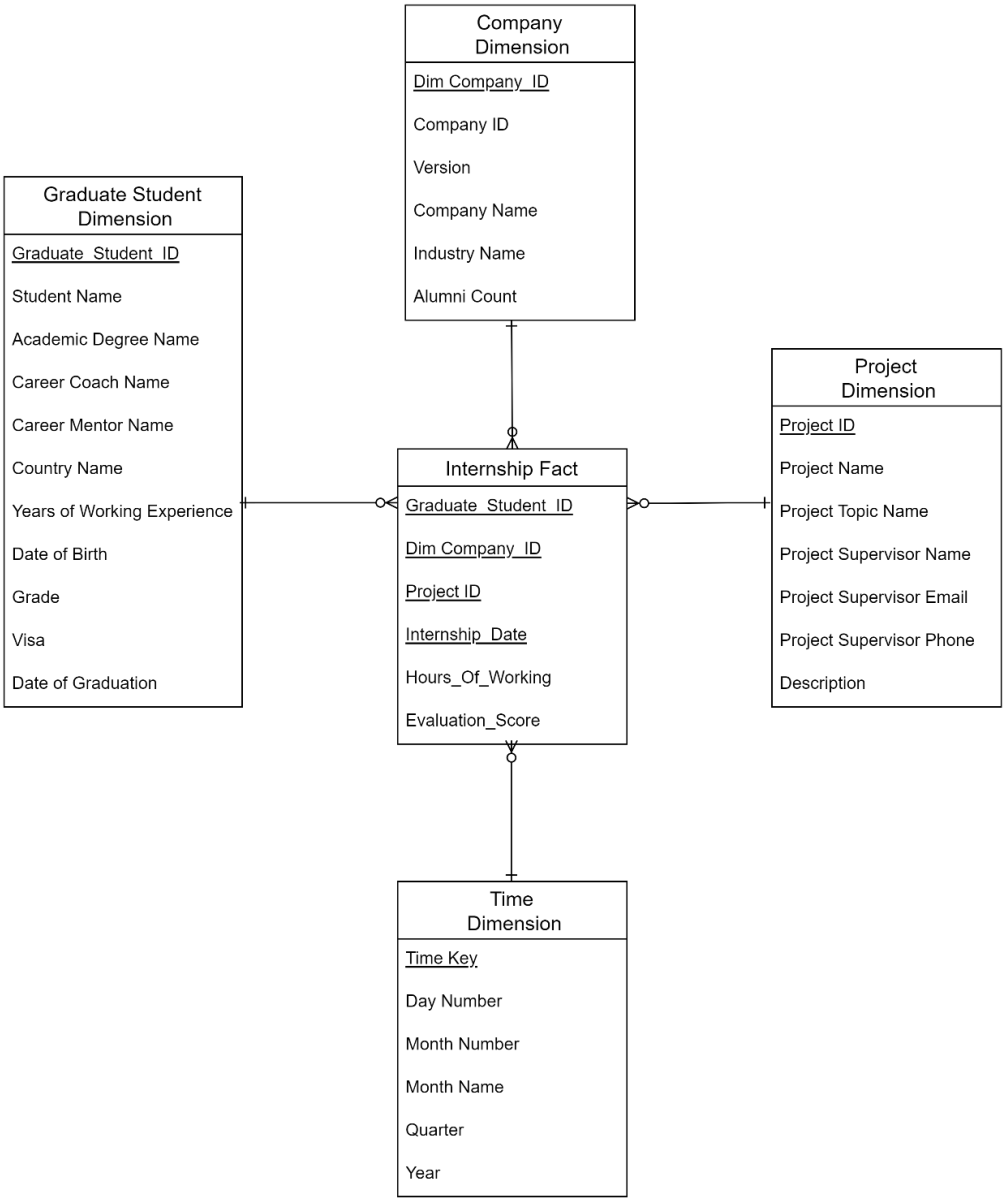

Fig 5.6 Final Star Schema for Internship Facts

6. Data Warehouse Use Cases

Case 1

The higher management of MBS is investigating performance of the business analytics program in terms of students' employability. The specific focus is to analyze the quality and quantity of offers received by MBusA students during the pandemic. The adoption of big data in healthcare and the emergence of pandemic has boosted the employment opportunities in the health industry. In response to the industry need, the management wants to explore if there is a growing number of students who have received offers from the health industry and also the growth in salary level. This can be done by using SQL queries to extract time taken for the graduates to get the first offer, average salary offered and the number of graduates who were offered salary above healthcare industry standard. The queries are performed as below.

```
SELECT student_name "Student Name", visa "Visa Status",
years_of_working_experience "Years of Working Experience",
grade"Student Final Grade", graduation_year "Graduation Year",
count(salary) "Offers Received", avg(salary)"Average Offers Salary",
avg(weeks_from_graduation) "Average Weeks From Graduation Date to Offer Date",
sum(is_salary_higher_than_average::INT) "Offers with Salary Above Industry
Standard",
sum(is_healthcare_industry::INT) "Offers in Healthcare Industry"
FROM
(
    SELECT gsd.student_name, gsd.visa,
    gsd.years_of_working_experience,
    gsd.grade, extract(year FROM gsd.date_of_graduation) "graduation_year",
    o_fact.salary, o_fact.weeks_from_graduation,
    cd.company_name, cd.industry_name,
    pd.position_name, pd.average_salary,
    td.year,
    CASE
        WHEN o_fact.salary > pd.average_salary THEN TRUE
        ELSE FALSE
    END AS "is_salary_higher_than_average",
    CASE
        WHEN cd.industry_name = 'Healthcare' THEN TRUE
        ELSE FALSE
    END AS "is_healthcare_industry"
    FROM offer_fact o_fact
    INNER JOIN graduate_student_dimension gsd ON gsd.graduate_student_id =
o_fact.graduate_student_id
    INNER JOIN position_dimension pd ON pd.dim_position_id =
o_fact.dim_position_id
    INNER JOIN company_dimension cd ON cd.dim_company_id =
o_fact.dim_company_id
    INNER JOIN time_dimension td ON td.time_key = o_fact.offer_date
    WHERE td.year BETWEEN 2020 AND 2022
) subquery
```

```
GROUP BY student_name, visa, years_of_working_experience, grade,
graduation_year
ORDER BY "Graduation Year", "Student Name";
```

The query aggregated important parameters from multiple data sources (tables) for the analysis of MBS student employability, and it is filtered specifically only for the covid period. The query helps indicate whether the offer comes from the health industry, which is the team assumption of a high performing industry during the period. With timely updates, they would help us to understand performance of different customer segments. International students are the largest customer segment of most business schools in Melbourne according to the Age (2019), taking up 67% of the total. By comparing the figures with other competitors, MBS is able to understand the course efficiency and reform strategies.

Case 2

The school is reviewing the efficiency of career coaches over the past 7 years of operation of the Master of Business Analytics Program. The coaches' performance is assessed by looking at the career outcome of the graduates they are responsible for. In universities located in Melbourne, overseas students account for more than 66% of total enrolments in the business schools. As international students are the largest customer segment for the MBusA course and it is more difficult for them to secure employment due to the non permanent visa, the success rate of international students getting offes shall be one of the key criteria. Other assessment criteria consist of the average time taken for the responsible students to get the first offer and the student's average salary. The following queries can resolve this business problem.

```
SELECT career_coach_name "Career Coach Name",
count(student_name) "Students Coached",
avg(years_of_working_experience) "Average Students Years of Working
Experience",
avg(grade) "Average Students Grade",
sum(CASE
        WHEN visa = 'Student Visa' THEN 1
        ELSE 0
        END
    ) AS "Students with Student Visa",
avg(offers_received) "Average Offers Received per Student",
avg(highest_salary)"Average Highest Offers Salary",
avg(min_weeks_from_graduation) "Average Earliest Weeks from Graduation Date to
Offer Date",
avg(offers_with_salary_above_industry_standard) "Average Offers with Salary
Above Industry Standard"
FROM
(
    SELECT gsd.career_coach_name, gsd.student_name,
    gsd.years_of_working_experience, gsd.grade, gsd.visa,
    extract(year FROM gsd.date_of_graduation) "graduation_year",
    count(o_fact.salary) offers_received, max(o_fact.salary) highest_salary,
```

```
    min(o_fact.weeks_from_graduation) min_weeks_from_graduation,
    sum(CASE
            WHEN o_fact.salary > pd.average_salary THEN 1
            ELSE 0
            END
        ) AS "offers_with_salary_above_industry_standard"
    FROM offer_fact o_fact
    INNER JOIN graduate_student_dimension gsd ON gsd.graduate_student_id =
o_fact.graduate_student_id
    INNER JOIN position_dimension pd ON pd.dim_position_id =
o_fact.dim_position_id
    INNER JOIN company_dimension cd ON cd.dim_company_id =
o_fact.dim_company_id
    INNER JOIN time_dimension td ON td.time_key = o_fact.offer_date
    WHERE td.year BETWEEN 2017 AND 2022
    GROUP BY gsd.career_coach_name, gsd.student_name, gsd.visa,
    gsd.years_of_working_experience, gsd.grade, graduation_year
) subquery
GROUP BY career_coach_name
HAVING count(student_name) >= 5
ORDER BY "Average Offers Received per Student" DESC, "Career Coach Name";
```

These queries generate a dashboard to show us the key performance indicators of career coaches. It presents the average offers received per student, average highest salary offered, average weeks of waiting from graduation to offer date and average number of offers with salary above industry standard for each career coach. This is very useful for management to conduct performance review and yearly appraisal with the coaches. By referring to the dashboard, managers can easily compare performance of different coaches, explore discrepancies or mismatch between coaches and students and allocate human resources in a more organized manner. The query also filters career coaches with less than 5 students coached to prevent small sample problems.

Case 3

MBS cooperates with an extended network of corporate partners. These partners select projects for the interns from MBS. MBS is also looking to recruit more partner companies as well as enrich projects for the internship program. The coordinator wants to have the relevant data to support his presentation by showing the variety of projects covered and the performances of the students. The pandemic has affected many different industries. Top three industries that fluctuated most during the pandemic are healthcare, tourism and the technology industry. The coordinator considers presenting the evaluation score of the students, the project they have completed, alumni number in the partner companies, the total hours students spent on the project and variation across industries. The queries below extract the data needed for users.

```sql
SELECT '2017-2019' "Period",
pd.project_topic_name "Project Topic",
count(distinct pd.project_topic_name) "Number of Projects",
count(i_fact.evaluation_score) "Number of MBS Students",
sum(i_fact.hours_of_working)"Total MBS Students Work Hours",
avg(i_fact.evaluation_score)"Average MBS Students Evaluation Score"
FROM internship_fact i_fact
INNER JOIN graduate_student_dimension gsd ON gsd.graduate_student_id =
i_fact.graduate_student_id
INNER JOIN project_dimension pd ON pd.project_id = i_fact.project_id
INNER JOIN company_dimension cd ON cd.dim_company_id = i_fact.dim_company_id
INNER JOIN time_dimension td ON td.time_key = i_fact.internship_date
WHERE td.year BETWEEN 2017 AND 2019
AND cd.industry_name in ('Healthcare', 'Tourism', 'Technology')
GROUP BY project_topic_name
UNION ALL
SELECT '2020-2022' "Period",
pd.project_topic_name "Project Topic",
count(distinct pd.project_topic_name) "Number of Projects",
count(i_fact.evaluation_score) "Number of MBS Students",
sum(i_fact.hours_of_working)"Total MBS Students Work Hours",
avg(i_fact.evaluation_score)"Average MBS Students Evaluation Score"
FROM internship_fact i_fact
INNER JOIN graduate_student_dimension gsd ON gsd.graduate_student_id =
i_fact.graduate_student_id
INNER JOIN project_dimension pd ON pd.project_id = i_fact.project_id
INNER JOIN company_dimension cd ON cd.dim_company_id = i_fact.dim_company_id
INNER JOIN time_dimension td ON td.time_key = i_fact.internship_date
WHERE td.year BETWEEN 2020 AND 2022
AND cd.industry_name in ('Healthcare', 'Tourism', 'Technology')
GROUP BY project_topic_name
ORDER BY "Number of Projects" DESC, "Project Topic";
```

The queries have aggregated not only information on the performance of students in the projects, but also comparison between projects held in companies from different industries. The table created by the queries assists MBS with business planning and budgeting every year. It reveals the topics covered by projects, companies and industries available for students, as well as interns' performances. It helps the coordinator to plan each year to promote richness in the offering of internships. Through reviewing performance of the students in the internship, it also helps coordinators and managers to improve teaching quality and adjust the course offerings accordingly. The query also divides the analysis to pre-pandemic internship opportunities (2017 to 2019) and post-pandemic internship opportunities (2020-2022) to illustrate the dynamic changes between the 2 periods.

Reference

Amazon Web Services. (2022). *Data Warehouse Concepts*. 2nd ed. [online] Available at: https://aws.amazon.com/data-warehouse/ [Accessed 30 Jun. 2022].

Bhansali, N. (2007). *Strategic Alignment in Data Warehouses Two Case Studies*. [online] *https://researchrepository.rmit.edu.au/discovery/delivery/61RMIT_INST:ResearchRepository/12247552650001341#13248417290001341*. Available at: https://researchrepository.rmit.edu.au/discovery/delivery/61RMIT_INST:ResearchRepository/12247552650001341#13248417290001341 [Accessed Jun. 2022].

Carey, A. (2003). *Design Tip #43: Dealing With Nulls In The Dimensional Model*. 2nd ed. [online] Kimball Group. Available at: https://www.kimballgroup.com/2003/02/design-tip-43-dealing-with-nulls-in-the-dimensional-model/ [Accessed 28 Jun. 2022].

Carey, A. (2019). Melbourne's uni sector has highest proportion of international students. The Age. [online] 23 Aug. Available at: https://www.theage.com.au/national/victoria/melbourne-s-uni-sector-has-highest-proportion-of-international-students-20190823-p52k6w.html [Accessed 24 Jun. 2022].

Lane, P. (2002). *Oracle9i Data Warehousing Guide*. 2nd ed. [online] Available at: https://docs.oracle.com/cd/B10501_01/server.920/a96520/title.htm [Accessed 30 Jun. 2022].

Maguire, J. (2020). *Top 10 Benefits of a Data Warehouse*. 2nd ed. [online] Datamation. Available at: https://www.datamation.com/big-data/top-10-benefits-of-a-data-warehouse/ [Accessed 27 Jun. 2022].

Moody, D. and Kortink, M. (2003). *From ER Models to Star Schemas: Mapping From Operational Databases to Data Warehouse Designs*. *ACIS 2003 Proceedings*. 65. https://aisel.aisnet.org/acis2003/65.

Sinha, T. (2021). OLAP vs. OLTP: What's the Difference? *IBM*. Available at: https://www.ibm.com/cloud/blog/olap-vs-oltp [Accessed 30 Jun. 2022].

Stiglich, P. (1997). *Performance Benefits of Surrogate Keys in Dimensional Models*. 2nd ed. [online] DataManagementU. Available at: https://www.ewsolutions.com/performance-benefits-surrogate-keys-dimensional-models/ [Accessed 29 Jun. 2022].

Stitch. (2022). *OLTP and OLAP: a practical comparison*. 2nd ed. [online] Available at: https://www.stitchdata.com/resources/oltp-vs-olap [Accessed 25 Jun. 2022].

Yu, X. (2021). The Application of Data Warehouse in Teaching Management in Colleges and Universities. *Journal of Physics: Conference Series*. [online] Available at: https://iopscience.iop.org/article/10.1088/1742-6596/1738/1/012090/pdf [Accessed 30 Jun. 2022].

Appendix

Data Dictionary

For ER diagram

Offer table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
OfferID	Unique identifier for the Offer table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
GraduateStudentID	Foreign Key to Graduate Student Table	Integer	Required	12345678	GraduateStudentID attribute in Graduate Student database table	Foreign	
CompanyID	Foreign Key to Company Table	Integer	Required	1	Company ID attribute in Company database table	Foreign	
PositionID	Foreign Key to Position Table	Integer	Required	1	Position ID attribute in Position database table	Foreign	
Type	Type of offer that the student receive	Varchar	Required	Full-time	Submission of student offer survey form		
DateOfOffer	Date of offer	Integer	Required	3	Submission of student offer survey form		
Salary	Salary of the offer	Integer	Required	80000	Submission of student offer survey form		
IsReturnFromInternship	Is the offer is return offer from internship	Boolean	Required	TRUE	Submission of student offer survey form		
IsAccepted	Is the offer accepted by the student	Boolean	Required	TRUE	Submission of student offer survey form		

GraduateStudent table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
GraduateStudentID	Student Number ID	Integer	Required	12345678	Student Registration data	Primary	Unique
AcademicDegreeID	Foreign Key to Academic Degree Table	Integer	Required	1	Degree ID attribute in PreviousDegree database table	Foreign	
CountryID	Foreign Key to Country Table	Integer	Required	1	Country ID attribute in Country table	Foreign	
CareerCoachID	Foreign Key to Career Coach Table	Varchar	Required	1	Coach ID attribute in CareerCoach Student database table	Foreign	
CareerMentorID	Foreign Key to Career Mentor Table	Integer	Optional	1	Mentor ID attribute in CareerMentor database table	Foreign	
Name	Name of the student	Varchar	Required	David Heston	Student Registration data		
Email	Email address of the student	Varchar	Required	david3@student.unimelb.edu.au	Student Registration data		Unique
Phone	Phone number of the student	Integer	Required	401234567	Student Registration data		Unique
DateOfBirth	Date of birth of the student	Date	Required	1/01/2023	Student Registration data		
Address	Home address of the student	Varchar	Required		Student Registration data		
Grade	Final grade of the Student	Numeric	Required	85.8	Student Registration data		
Visa	Whether the student's visa has expired	Varchar	Required	Student Visa	Student Registration data		
DateOfGraduation	The exact date of a student's graduation	Date	Required	1/01/2023	Student Registration data		

Company table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
CompanyID	Unique identifier for the Company table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
IndustryID	Foreign Key to Industry Table	Integer	Required	1	Industry ID attribute in Industry database table	Foreign	
Name	The name of the company	Varchar	Required	David Heston	Manual Data Entry		Unique
Location	The location of the company	Varchar	Required	Melbourne	Manual Data Entry		
Phone	Phone number of the company	Varchar	Required	401234567	Manual Data Entry		Unique
NumberOfAlumni	The number of alumni in the company	Integer	Required	3	Student Survey data		

Internship table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
InternshipID	Unique identifier for the Internship table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
GraduateStudentID	Foreign Key to GraduateStudent Table	Integer	Required	12345678	Student ID attribute in GraduateStudent database table	Foreign	
CompanyID	Foreign Key to Company Table	Integer	Required	1	Company ID attribute in Company database table	Foreign	
ProjectID	Foreign Key to Project Table	Integer	Required	1	Project ID attribute in Project database table	Foreign	
HoursOfWorking	Hours of internship working	Integer	Required	200	Student Internship Report		
EvaluationScore	The final mark given to the student by the supervior of internship	Integer	Required	85	Student Internship Report		

CareerCoach table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
CareerCoachID	Career coach Number ID	Varchar	Required	12345678	Career Coach data	Primary	Unique
Name	Name of the career coach	Varchar	Required	David Heston	Career Coach data		
Email	Email of the career coach	Varchar	Required	david3@student.unimelb.edu.au	Career Coach data		Unique
Phone	Phone number of the career coach	Varchar	Required	401234567	Career Coach data		Unique
JoinDate	The date the coach joins MBS	Date	Required	1/01/2017	Career Coach data		

Position table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
PositionID	Unique identifier for the Position table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
PositionLevelID	Foreign Key to PositionLevel Table	Integer	Required	1	Position Level ID attribute in Position Level database table	Foreign	
CountryID	Foreign Key to Country Table	Integer	Required	1	Country ID attribute in Country database table	Foreign	
Name	The title of the position	Varchar	Required	Data Scientist	Manual Data Entry		
Description	The job description about the position	Varchar	Required	a Scientist for Fina	Manual Data Entry		
AverageSalary	Average salary of the position	Numeric	Required	83666.67	Manual Data Entry		

AcademicDegree table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
AcademicDegreeID	Unique identifier for the AcademicDegree table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
Name	Degree Name	Varchar	Required	of Compute	Manual Data Entry		Unique

Industry table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
IndustryID	Unique identifier for the Industry table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
Name	Name of company's industry	Varchar	Required	real-estate	Manual Data Entry		Unique

ProjectSupervisor table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
ProjectSupervisorID	Unique identifier for the ProjectSupervisor table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
CompanyID	Foreign Key to Company Table	Integer	Required	1	Company ID attribute in Company database table	Foreign	
Name	Name of project supervisor	Varchar	Required	David Heston	Student Internship Report		
Email	Email of project supervisor	Varchar	Required	david3@student.unimelb.edu.au	Student Internship Report		Unique
Phone	Phone number of project monitor	Integer	Required	401234567	Student Internship Report		Unique

WorkExperience table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
WorkExperienceID	Unique identifier for the WorkExperience table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
GraduateStudentID	Foreign Key to GraduateStudent Table	varchar	Required	1234567	Student ID attribute in Graduate database table	Foreign	
PositionID	Foreign Key to Position Table	Integer	Required	1	Position ID attribute in Position database table	Foreign	
YearsOfWorking	Years of working	Integer	Required	1	Student Resume		

Country table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
CountryID	Unique identifier for the Country table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
CountryName	Name of the country	varchar	Required	Australia	Manual Data Entry		Unique

PositionLevel table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
PositionLevelID	Unique identifier for the PositionLevel table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
Name	Name of level of position	varchar	Required	Senior Level	Manual Data Entry		Unique

CareerMentor table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
CareerMentorID	Unique identifier for the CareerMentor table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
CompanyID	Foreign Key to Company Table	Integer	Required	1	Company ID attribute in Company database table	Foreign	
Name	Name of career mentor	Varchar	Required	David Heston	Name attribute in CareerMentor database table		
Email	Email of career mentor	Varchar	Required	david3@student.unimelb.edu.au	Email attribute in CareerMentor database table		Unique
Phone	Phone number of career mentor	Varchar	Required	401234567	Phone attribute in CareerMentor database table		Unique

Project table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
ProjectID	Unique identifier for the Project table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
ProjectSupervisorID	Foreign Key to ProjectSupervisor Table	Integer	Required	1	ProjectSupervisorID ID attribute in Monitor database table	Foreign	
ProjectTopicID	Foreign Key to ProjectTopic Table	Integer	Required	1	Topic ID attribute in ProjectTopic database table	Foreign	
Name	Name of project that the student works for in internship	Varchar	Required	Charity City	ProjectName attribute in Project database table		Unique
Description	Description of project	Text	Required	Charity City			

ProjectTopic table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
ProjectTopicID	Unique identifier for the ProjectTopic table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
Name	Name of the project topic	Varchar	Required	Charity City	TopicName attribute in ProjrctTopic database table		Unique
Description	Description of the project topic	Text	Required	Charity City			

For Star Schema1

Offer Fact table

Attribute	Description	Data Type	Required/ Optional	Example	Data Source	Key	Unique
GraduateStudentID	Foreign Key to Graduate Student Dimension Table	Varchar	Required	12345678	Student ID attribute in Graduate Student database table	Composite Primary	
DimCompanyID	Foreign Key to Company Dimension Table	Integer	Required	1	Company ID attribute in Company database table	Composite Primary	
DimPositionID	Foreign Key to Position Dimension Table	Integer	Required	1	Position ID attribute in Position database table	Composite Primary	
OfferDate	Foreign Key to Time Dimension Table	Integer	Required	1/01/2021	Generated from offer date in offer table	Composite Primary	
Salary	Salary offer	Integer	Required	80000	Salary attribute in Offer database table		
WeeksFromGraduatio	Time between graduation and offer	Integer	Required	3	TimeDurationInWeek attribute in Offer database table		

Graduate Student Dimension table

Attribute	Description	Data Type	Required/ Optional	Example	Data Source	Key	Unique
GraduateStudentID	Unique identifier for a graduate student	varchar	Required	123345678	Student ID attribute in Graduate Student database table	Primary	Unique
StudentName	Name of the student	varchar	Required	David Heston	Name attribute in GraduateStudent database table		
AcademicDegreeName	Name of student's degree	varchar	Required	Master	Type attribute in PreviousDegree database table		
CareerCoachName	Name of student's coach	varchar	Required	David Heston	Name attribute in CareerCoach database table		
CareerMentorName	Name of student's mentor	varchar	Optional	David Heston	Name attribute in Careermentor database table		
CountryName	Name of student's nationality	varchar	Required	Australia	CountryName attribute in Country database table		
YearOfWorkingExperien	Total time of working experiences	Integer	Required	3	Exp attribute in WorkExperience database table is aggregated for each student's latest experience		
Grade	Average marks of the Student	decimal	Required	85	Grade attribute in GraduateStudent database table		
DateOfBirth	Date of birth of the student	date	Required	1/01/2023	DateOfBirth attribute in GraduateStudent database table		
Visa	Whether the student's visa has expired	varchar	Required	PR	VisaState attribute in GraduateStudent database table		
DateOfGraduation	The exact date of a student's graduation	date	Required	1/01/2023	GraduationDate attribute in GraduateStudent database table		

Company Dimension table

Attribute	Description	Data Type	Required/ Optional	Example	Data Source	Key	Unique
DimCompanyID	Unique identifier for the CompanyDimension table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
CompanyID	Identifier for a company	Integer	Required	1	Company ID attribute in Company database table		
Version	A label for yearly update	Integer	Required	1	A number generated for data warehousing		
CompanyName	The name of the company	Varchar	Required	David Heston	Name attribute in Company database table		
IndustryName	Name of company's industry	Varchar	Required	real-estate	Name attribute in Industry database table		
NumberOfAlumni	The number of alumni in the company	Integer	Required	3	AlumniCount attribute in Company database table		

Position Dimension table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
DimPositionID	Unique identifier for the PositionDimension table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
PositionID	Identifier for student's position in company	Integer	Required	1	Position ID attribute in Position database table		
Version	A label for yearly update	Integer	Required	1	A number generated for data warehousing		
PositionName	The title of the student's position in the company	varchar	Required	Data Science	Name attribute in Position database table		
PositionLevelNam	Name of level of position	varchar	Required	Senior Level	Name attribute in PositionLevel database table		
Description	The job description about student's position	varchar	Required	ta Science for Finar	Description attribute in Position database table		
AverageSalary	Average salary of the position	numeric	Required	836666.67	AverageSalary attribute in Position database table		
CountryName	Name of country	varchar	Required	Australia	Name attribute in Country database table		

Time Dimension table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
TimeKey	Unique identifier for the TimeDimension table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
day_number	Day Value	Integer	Required	1	System Generated		
month_number	Month Value	Integer	Required	1	System Generated		
month_name	Month Name	Varchar	Required	June	System Generated		
quarter	Quarter Value	Integer	Required	3	System Generated		
year	Year Value	Integer	Required	2019	System Generated		

For Star Schema2

Internship Fact table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
GraduateStudentID	Foreign Key to Graduate Student Dimension Table	Varchar	Required	12345678	Student ID attribute in Graduate database table	Composite Primary	
DimCompanyID	Foreign Key to Company Dimension Table	Integer	Required	1	Company ID attribute in Company database table	Composite Primary	
ProjectID	Foreign Key to Project Dimension Table	Integer	Required	1	Project ID attribute in Project database table	Composite Primary	
InternshipDate	Foreign Key to Time Dimension Table	Integer	Required	1	Generated from internship date in internship table	Composite Primary	
HoursOfWorking	Hours of Internship	Integer	Required	14	Duration attribute in Internship database table		
EvaluationScore	The final mark of the student when internship/A-Lab end	Integer	Required	85	EvaluationScore attribute in Internship database table		

Graduate Student Dimension table

Attribute	Description	Data Type	Required/ Optional	Example	Data Source	Key	Unique
GraduateStudentID	Unique identifier for a graduate student	varchar	Required	123345678	Student ID attribute in Graduate Student database table	Primary	Unique
StudentName	Name of the student	varchar	Required	David Heston	Name attribute in GraduateStudent database table		
AcademicDegreeNan	Name of student's degree	varchar	Required	Master	Type attribute in PreviousDegree database table		
CareerCoachName	Name of student's coach	varchar	Required	David Heston	Name attribute in CareerCoach database table		
CareerMentorName	Name of student's mentor	varchar	Optional	David Heston	Name attribute in Careermentor database table		
CountryName	Name of student's nationality	varchar	Required	Australia	CountryName attribute in Country database table		
YearOfWorkingExper	Total time of working experiences	Integer	Required	3	Exp attribute in WorkExperience database table is aggregated for each student's latest experience		
Grade	Average marks of the Student	decimal	Required	85	Grade attribute in GraduateStudent database table		
DateOfBirth	Date of birth of the student	date	Required	1/01/2023	DateOfBirth attribute in GraduateStudent database table		
Visa	Whether the student's visa has expired	varchar	Required	PR	VisaState attribute in GraduateStudent database table		
DateOfGraduation	The exact date of a student's graduation	date	Required	1/01/2023	GraduationDate attribute in GraduateStudent database table		

Company Dimension table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
DimCompanyID	Unique identifier for the CompanyDimension table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
CompanyID	Identifier for a company	Integer	Required	1	Company ID attribute in Company database table		
Version	A label for yearly update	Integer	Required	1	A number generated for data warehousing		
CompanyName	The name of the company	Varchar	Required	David Heston	Name attribute in Company database table		
IndustryName	Name of company's industry	Varchar	Required	Technology	Name attribute in Industry database table		
NumberOfAlum	The number of alumni in the company	Integer	Required	3	AlumniCount attribute in Company database table		

Project Dimension table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
ProjectID	Unique identifier for project that the student works for in internship	Integer	Required	1	Project ID attribute in Project database table	Primary	
ProjectName	Name of project that the student works for in internship	Varchar	Required	Charity City	ProjectName attribute in Project database table		Unique
ProjectTopicName	Name of topic ofproject that the student works for in internship	Varchar	Required	Charity City	TopicName attribute in ProjrctTopic database table		
ProjectSupervisorName	Name of student's supervisor	Varchar	Required	David Heston	Name attribute in ProjectSupervisor database table		
ProjectSupervisorEmail	Email of student's supervisor	Varchar	Required	david3@student.unimelb.edu.au'	Email attribute in ProjectSupervisor database table		
ProjectSupervisorPhone	Phone of student's supervisor	Varchar	Required	401234567	Phone attribute in ProjectSupervisor database table		
Description	Project description	Text	Required	Finance Project	Description attribute in Project database table		

Time Dimension table

Attribute	Description	Data Type	Required/Optional	Example	Data Source	Key	Unique
TimeKey	Unique identifier for the TimeDimension table	Integer	Required	1	Automatically generated using BIGSERIAL (Surrogate Key)	Primary	Unique
day_number	Day Value	Integer	Required	1	System Generated		
month_number	Month Value	Integer	Required	1	System Generated		
month_name	Month Name	Varchar	Required	June	System Generated		
quarter	Quarter Value	Integer	Required	3	System Generated		
year	Year Value	Integer	Required	2019	System Generated		

Create SQL

```
--ERD

CREATE DATABASE assignment;

\c assignment;

CREATE TABLE country (
    country_id BIGSERIAL PRIMARY KEY,
    "name" VARCHAR(255) UNIQUE NOT NULL
);

CREATE TABLE academic_degree (
    academic_degree_id BIGSERIAL PRIMARY KEY,
    "name" VARCHAR(255) UNIQUE NOT NULL
);

CREATE TABLE career_coach (
    career_coach_id VARCHAR(255) PRIMARY KEY,
    "name" VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    phone VARCHAR(255) UNIQUE NOT NULL,
    join_date DATE NOT NULL
);

CREATE TABLE industry (
    industry_id BIGSERIAL PRIMARY KEY,
    "name" VARCHAR(255) UNIQUE NOT NULL
);

CREATE TABLE company (
    company_id BIGSERIAL PRIMARY KEY,
    industry_id BIGINT NOT NULL,
    "name" VARCHAR(255) UNIQUE NOT NULL,
    phone VARCHAR(255) UNIQUE NOT NULL,
    "location" VARCHAR(255) NOT NULL,
    number_of_alumni SMALLINT NOT NULL,
    FOREIGN KEY (industry_id) REFERENCES industry (industry_id)
    ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE TABLE career_mentor (
    career_mentor_id BIGSERIAL PRIMARY KEY,
    company_id BIGINT NOT NULL,
    "name" VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    phone VARCHAR(255) UNIQUE NOT NULL,
```

```
    FOREIGN KEY (company_id) REFERENCES company (company_id)
    ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE TABLE project_topic (
    project_topic_id BIGSERIAL PRIMARY KEY,
    "name" VARCHAR(255) UNIQUE NOT NULL,
    "description" TEXT NOT NULL
);

CREATE TABLE project_supervisor (
    project_supervisor_id BIGSERIAL PRIMARY KEY,
    company_id BIGINT NOT NULL,
    "name" VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    phone VARCHAR(255) UNIQUE NOT NULL,
    FOREIGN KEY (company_id) REFERENCES company (company_id)
    ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE TABLE project (
    project_id BIGSERIAL PRIMARY KEY,
    project_supervisor_id BIGINT NOT NULL,
    project_topic_id BIGINT NOT NULL,
    "name" VARCHAR(255) UNIQUE NOT NULL,
    "description" TEXT NOT NULL,
    FOREIGN KEY (project_supervisor_id) REFERENCES project_supervisor
(project_supervisor_id)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (project_topic_id) REFERENCES project_topic (project_topic_id)
    ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE INDEX ON project (project_topic_id);

CREATE TABLE position_level (
    position_level_id BIGSERIAL PRIMARY KEY,
    "name" VARCHAR(255) UNIQUE NOT NULL
);

CREATE TABLE position (
    position_id BIGSERIAL PRIMARY KEY,
    position_level_id BIGINT NOT NULL,
    country_id BIGINT NOT NULL,
    "name" VARCHAR(255) NOT NULL,
    "description" TEXT NOT NULL,
    average_salary SMALLINT NOT NULL,
```

```
    FOREIGN KEY (position_level_id) REFERENCES position_level
(position_level_id)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (country_id) REFERENCES country (country_id)
    ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE TABLE graduate_student (
    graduate_student_id VARCHAR(255) PRIMARY KEY,
    academic_degree_id BIGINT NOT NULL,
    country_id BIGINT NOT NULL,
    career_coach_id VARCHAR(255) NOT NULL,
    career_mentor_id BIGINT NULL,
    "name" VARCHAR(255) NOT NULL,
    email VARCHAR(255) UNIQUE NOT NULL,
    phone VARCHAR(255) UNIQUE NOT NULL,
    date_of_birth DATE NOT NULL,
    "address" VARCHAR(255) NOT NULL,
    grade NUMERIC(4,2) NOT NULL,
    visa VARCHAR(255) NOT NULL,
    date_of_graduation DATE NOT NULL,
    FOREIGN KEY (academic_degree_id) REFERENCES academic_degree
(academic_degree_id)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (country_id) REFERENCES country (country_id)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (career_coach_id) REFERENCES career_coach (career_coach_id)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (career_mentor_id) REFERENCES career_mentor (career_mentor_id)
    ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE INDEX ON graduate_student ("name");
CREATE UNIQUE INDEX ON graduate_student (email);

CREATE TABLE work_experience (
    work_experience_id BIGSERIAL PRIMARY KEY,
    graduate_student_id VARCHAR(255) NOT NULL,
    position_id BIGINT NOT NULL,
    years_of_working SMALLINT NOT NULL,
    FOREIGN KEY (graduate_student_id) REFERENCES graduate_student
(graduate_student_id)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (position_id) REFERENCES position (position_id)
    ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE TABLE offer (
```

```
    offer_id BIGSERIAL PRIMARY KEY,
    graduate_student_id VARCHAR(255) NOT NULL,
    company_id BIGINT NOT NULL,
    position_id BIGINT NOT NULL,
    "type" VARCHAR(255) NOT NULL,
    date_of_offer DATE NOT NULL,
    salary SMALLINT NOT NULL,
    is_return_from_internship BOOLEAN NOT NULL,
    is_accepted BOOLEAN NOT NULL,
    FOREIGN KEY (graduate_student_id) REFERENCES graduate_student
(graduate_student_id)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (company_id) REFERENCES company (company_id)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (position_id) REFERENCES position (position_id)
    ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE INDEX ON offer (company_id);
CREATE INDEX ON offer (position_id);

CREATE TABLE internship (
    internship_id BIGSERIAL PRIMARY KEY,
    graduate_student_id VARCHAR(255) NOT NULL,
    company_id BIGINT NOT NULL,
    project_id BIGINT NOT NULL,
    hours_of_working SMALLINT NOT NULL,
    evaluation_score SMALLINT NOT NULL,
    FOREIGN KEY (graduate_student_id) REFERENCES graduate_student
(graduate_student_id)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (company_id) REFERENCES company (company_id)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (project_id) REFERENCES project (project_id)
    ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE INDEX ON internship (company_id);

--DIMENSION TABLE
CREATE TABLE company_dimension(
    dim_company_id BIGSERIAL PRIMARY KEY,
    company_id BIGINT NOT NULL,
    "version" INTEGER NOT NULL,
    company_name VARCHAR(255) NOT NULL,
    industry_name VARCHAR(255) NOT NULL,
    number_of_alumni SMALLINT NOT NULL,
    UNIQUE (company_id, "version")
```

```
);

CREATE TABLE graduate_student_dimension(
    graduate_student_id VARCHAR(255) PRIMARY KEY,
    student_name VARCHAR(255) NOT NULL,
    academic_degree_name VARCHAR(255) NOT NULL,
    career_coach_name VARCHAR(255) NOT NULL,
    career_mentor_name VARCHAR(255) NULL,
    country_name VARCHAR(255) NOT NULL,
    years_of_working_experience SMALLINT NOT NULL,
    date_of_birth DATE NOT NULL,
    grade DECIMAL(4,2) NOT NULL,
    visa VARCHAR(255) NOT NULL,
    date_of_graduation DATE NOT NULL
);

CREATE TABLE position_dimension(
    dim_position_id BIGSERIAL PRIMARY KEY,
    position_id BIGINT NOT NULL,
    "version" INTEGER NOT NULL,
    position_name VARCHAR(255) NOT NULL,
    position_level_name VARCHAR(255) NOT NULL,
    "description" TEXT NOT NULL,
    average_salary SMALLINT NOT NULL,
    country_name VARCHAR(255) NOT NULL,
    UNIQUE (position_id, "version")
);

CREATE TABLE project_dimension(
    project_id BIGINT PRIMARY KEY,
    project_name VARCHAR(255) UNIQUE NOT NULL,
    project_topic_name VARCHAR(255) NOT NULL,
    project_supervisor_name VARCHAR(255) NOT NULL,
    project_supervisor_email VARCHAR(255) NOT NULL,
    project_supervisor_phone VARCHAR(255) NOT NULL,
    "description" TEXT NOT NULL
);

CREATE TABLE time_dimension(
    time_key BIGSERIAL PRIMARY KEY,
    day_number SMALLINT NOT NULL,
    month_number SMALLINT NOT NULL,
    month_name VARCHAR(255) NOT NULL,
    quarter SMALLINT NOT NULL,
    year SMALLINT NOT NULL,
    UNIQUE (day_number, month_number, month_name, quarter, year)
);
```

```sql
--FACT TABLE 1 (Offer)

CREATE TABLE offer_fact(
    graduate_student_id VARCHAR(255) NOT NULL,
    dim_company_id BIGINT NOT NULL,
    dim_position_id BIGINT NOT NULL,
    offer_date BIGINT NOT NULL,
    salary SMALLINT NOT NULL,
    weeks_from_graduation SMALLINT NOT NULL,
    PRIMARY KEY(graduate_student_id, dim_company_id, dim_position_id,
offer_date),
    FOREIGN KEY (graduate_student_id) REFERENCES graduate_student_dimension
(graduate_student_id)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (dim_company_id) REFERENCES company_dimension (dim_company_id)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (dim_position_id) REFERENCES position_dimension
(dim_position_id)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (offer_date) REFERENCES time_dimension (time_key)
    ON UPDATE CASCADE ON DELETE NO ACTION
);

CREATE INDEX ON offer_fact (graduate_student_id);
CREATE INDEX ON offer_fact (dim_company_id);
CREATE INDEX ON offer_fact (dim_position_id);
CREATE INDEX ON offer_fact (offer_date);

--FACT TABLE 2 (Internship)

CREATE TABLE internship_fact(
    graduate_student_id VARCHAR(255) NOT NULL,
    dim_company_id BIGINT NOT NULL,
    project_id BIGINT NOT NULL,
    internship_date BIGINT NOT NULL,
    hours_of_working SMALLINT NOT NULL,
    evaluation_score SMALLINT NOT NULL,
    PRIMARY KEY(graduate_student_id, dim_company_id, project_id,
internship_date),
    FOREIGN KEY (graduate_student_id) REFERENCES graduate_student_dimension
(graduate_student_id)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (dim_company_id) REFERENCES company_dimension (dim_company_id)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (project_id) REFERENCES project_dimension (project_id)
    ON UPDATE CASCADE ON DELETE NO ACTION,
    FOREIGN KEY (internship_date) REFERENCES time_dimension (time_key)
    ON UPDATE CASCADE ON DELETE NO ACTION
```

```
);

CREATE INDEX ON internship_fact (graduate_student_id);
CREATE INDEX ON internship_fact (dim_company_id);
CREATE INDEX ON internship_fact (project_id);
CREATE INDEX ON internship_fact (internship_date);

--USE CASE 1
--Analyze the quality and quantity of offers received by MBS student during
Covid period
--Check whether they received offers from the health industry

SELECT student_name "Student Name", visa "Visa Status",
years_of_working_experience "Years of Working Experience",
grade"Student Final Grade", graduation_year "Graduation Year",
count(salary) "Offers Received", avg(salary)"Average Offers Salary",
avg(weeks_from_graduation) "Average Weeks From Graduation Date to Offer Date",
sum(is_salary_higher_than_average::INT) "Offers with Salary Above Industry
Standard",
sum(is_healthcare_industry::INT) "Offers in Healthcare Industry"
FROM
(
    SELECT gsd.student_name, gsd.visa,
    gsd.years_of_working_experience,
    gsd.grade, extract(year FROM gsd.date_of_graduation) "graduation_year",
    o_fact.salary, o_fact.weeks_from_graduation,
    cd.company_name, cd.industry_name,
    pd.position_name, pd.average_salary,
    td.year,
    CASE
        WHEN o_fact.salary > pd.average_salary THEN TRUE
        ELSE FALSE
    END AS "is_salary_higher_than_average",
    CASE
        WHEN cd.industry_name = 'Healthcare' THEN TRUE
        ELSE FALSE
    END AS "is_healthcare_industry"
    FROM offer_fact o_fact
    INNER JOIN graduate_student_dimension gsd ON gsd.graduate_student_id =
o_fact.graduate_student_id
    INNER JOIN position_dimension pd ON pd.dim_position_id =
o_fact.dim_position_id
    INNER JOIN company_dimension cd ON cd.dim_company_id =
o_fact.dim_company_id
    INNER JOIN time_dimension td ON td.time_key = o_fact.offer_date
    WHERE td.year BETWEEN 2020 AND 2022
) subquery
```

```sql
GROUP BY student_name, visa, years_of_working_experience, grade,
graduation_year
ORDER BY "Graduation Year", "Student Name";

--USE CASE 2
--Analyze Career Coach Performance
--Check how many student with student visa got offers

SELECT career_coach_name "Career Coach Name",
count(student_name) "Students Coached",
avg(years_of_working_experience) "Average Students Years of Working
Experience",
avg(grade) "Average Students Grade",
sum(CASE
        WHEN visa = 'Student Visa' THEN 1
        ELSE 0
        END
    ) AS "Students with Student Visa",
avg(offers_received) "Average Offers Received per Student",
avg(highest_salary)"Average Highest Offers Salary",
avg(min_weeks_from_graduation) "Average Earliest Weeks from Graduation Date to
Offer Date",
avg(offers_with_salary_above_industry_standard) "Average Offers with Salary
Above Industry Standard"
FROM
(
    SELECT gsd.career_coach_name, gsd.student_name,
    gsd.years_of_working_experience, gsd.grade, gsd.visa,
    extract(year FROM gsd.date_of_graduation) "graduation_year",
    count(o_fact.salary) offers_received, max(o_fact.salary) highest_salary,
    min(o_fact.weeks_from_graduation) min_weeks_from_graduation,
    sum(CASE
            WHEN o_fact.salary > pd.average_salary THEN 1
            ELSE 0
            END
        ) AS "offers_with_salary_above_industry_standard"
    FROM offer_fact o_fact
    INNER JOIN graduate_student_dimension gsd ON gsd.graduate_student_id =
o_fact.graduate_student_id
    INNER JOIN position_dimension pd ON pd.dim_position_id =
o_fact.dim_position_id
    INNER JOIN company_dimension cd ON cd.dim_company_id =
o_fact.dim_company_id
    INNER JOIN time_dimension td ON td.time_key = o_fact.offer_date
    WHERE td.year BETWEEN 2017 AND 2022
    GROUP BY gsd.career_coach_name, gsd.student_name, gsd.visa,
    gsd.years_of_working_experience, gsd.grade, graduation_year
) subquery
```

```
GROUP BY career_coach_name
HAVING count(student_name) >= 5
ORDER BY "Average Offers Received per Student" DESC, "Career Coach Name";

--USE CASE 3
--Analyze internship grouped by topic name
--Compare pre-Covid and post-Covid period in 3 industry: Healthcare, Tourism,
and Technology

SELECT '2017-2019' "Period",
pd.project_topic_name "Project Topic",
count(distinct pd.project_topic_name) "Number of Projects",
count(i_fact.evaluation_score) "Number of MBS Students",
sum(i_fact.hours_of_working)"Total MBS Students Work Hours",
avg(i_fact.evaluation_score)"Average MBS Students Evaluation Score"
FROM internship_fact i_fact
INNER JOIN graduate_student_dimension gsd ON gsd.graduate_student_id =
i_fact.graduate_student_id
INNER JOIN project_dimension pd ON pd.project_id = i_fact.project_id
INNER JOIN company_dimension cd ON cd.dim_company_id = i_fact.dim_company_id
INNER JOIN time_dimension td ON td.time_key = i_fact.internship_date
WHERE td.year BETWEEN 2017 AND 2019
AND cd.industry_name in ('Healthcare', 'Tourism', 'Technology')
GROUP BY project_topic_name
UNION ALL
SELECT '2020-2022' "Period",
pd.project_topic_name "Project Topic",
count(distinct pd.project_topic_name) "Number of Projects",
count(i_fact.evaluation_score) "Number of MBS Students",
sum(i_fact.hours_of_working)"Total MBS Students Work Hours",
avg(i_fact.evaluation_score)"Average MBS Students Evaluation Score"
FROM internship_fact i_fact
INNER JOIN graduate_student_dimension gsd ON gsd.graduate_student_id =
i_fact.graduate_student_id
INNER JOIN project_dimension pd ON pd.project_id = i_fact.project_id
INNER JOIN company_dimension cd ON cd.dim_company_id = i_fact.dim_company_id
INNER JOIN time_dimension td ON td.time_key = i_fact.internship_date
WHERE td.year BETWEEN 2020 AND 2022
AND cd.industry_name in ('Healthcare', 'Tourism', 'Technology')
GROUP BY project_topic_name
ORDER BY "Number of Projects" DESC, "Project Topic";
```

Indexing:

- GraduateStudent Table
    - Name: A lot of read operations from this table will be filtered by the graduate student's name.

    - Email (Unique Index): A lot of read operations from this table will be filtered by the graduate student's email.
- Project Table
    - ProjectTopicId: Project is usually grouped by its project topic so creating an index on attribute ProjectTopicId will increase read query efficiency.
- Offer Table
    - CompanyId: CompanyId will keep generating offers so it would make sense to create index on CompanyId attribute as a Company could generate a lot of offers and potentially a lot of queries will be executed by filtering or grouping using this attribute.
    - PositionId: It is common to analyse offer data by grouping it using the position of the offer hence it would increase performance of read query with grouping.
- Internship Table
    - CompanyId: CompanyId will keep offering internships so it would make sense to create index on CompanyId attribute as a Company could generate a lot of internships and potentially a lot of queries will be executed by filtering or grouping using this attribute.
- Offer Fact Table: we created indexes for each foreign key to the dimension tables as this table will be queried a lot by filtering or grouping using all of those foreign keys
- Internship Fact Table: we created indexes for each foreign key to the dimension tables as this table will be queried a lot by filtering or grouping using all of those foreign keys

Work Breakdown

The business case is based on Virginia's idea. Every group member has contributed ideas to the detailed Database design and Data warehouse design.

Each member takes charge of some sections of the assignment. All the sections (10 sections in report), however, are carefully discussed and agreed by everyone through the group meetings. Here is the work breakdown table:

SYNDICATE MEMBER	SECTIONS IN CHARGE
VIRGINIA LI	Section 2, 3, 4, 7
GABRIEL GUO	Section 4, 5, 6, 10
KUAN GAO	Section 2, 3, 5
AHMAD FAIG IGRO	Section 5, 6, 7, 9
DANICA DONG	Section 1, 2, 3, 8