CSCI 5408  
DATA MANAGEMENT AND  
WAREHOUSING  
ASSIGNMENT - 1

**Banner ID:** B00981016

**GitLab Link:** [GitLab\_Assignment\_1](https://git.cs.dal.ca/guntipalli/csci5408_w24_b00981016_ashish-kumar_guntipalli/-/tree/main/A1)

Table of Contents

[Problem – 1: 3](#_Toc160004597)

[Entities, Attributes and Justification 3](#_Toc160004598)

[Initial Entity-Relationship diagram 5](#_Toc160004599)

[Identification Of Design Issues and Modifications 5](#_Toc160004600)

[Final Entity-Relationship diagram 6](#_Toc160004601)

[Enhanced Entity-Relationship diagram 6](#_Toc160004602)

[Problem – 2: 8](#_Toc160004603)

[Design principles 8](#_Toc160004604)

[Evidence of Testing and Testcases 8](#_Toc160004605)

[Task – A 8](#_Toc160004606)

[Task – B1 9](#_Toc160004607)

[Task – B2 9](#_Toc160004608)

[Task – C 10](#_Toc160004609)

[References 12](#_Toc160004610)

# Problem – 1:

## Entities, Attributes and Justification

|  |  |  |
| --- | --- | --- |
| **Entity** | **Attributes** | **Justification** |
| city\_university | uni\_id,name, contact\_info, country | Represents the central institution itself which has educational programs and related services. |
| building | address, name, count\_of\_rooms  floors | Captures the physical infrastructure of the university like classrooms, cafeteria, auditorium etc. |
| department | dept\_id, dept\_name, description, head\_staff | Represents academic units within the university focusing on specific disciplines and offering related programs. |
| events | event\_id, event\_name, description, event\_location | Represents various activities organized by the university or student groups, fostering engagement and community building. |
| application | app\_id, status, applicant\_country, applicant\_name, applicant\_email | Represents the process of submitting a request for admission to a program within the university. |
| library | lib\_address, resources, opening\_time, closing\_time, services\_offered, | Represents the central repository for books, journals, and other information resources supporting teaching and research. |
| career\_service\_  center | location, services\_offered, working\_hours | Provides guidance and resources to students for career exploration, job search, and professional development. |
| job\_board | job\_id, job\_title, description, requirements | Facilitates connecting employers with potential student candidates seeking job opportunities. |
| research | research\_id, title, description, domain | Represents ongoing scholarly activities conducted by faculty and staff, contributing to knowledge creation. |
| laboratory | lab\_id, location, availability, equipment\_availability | Specialized facilities equipped for conducting experiments and research activities. |
| alerts | Alert\_id, title, description, urgency, status | Provides a communication channel for sending important information and notifications to various stakeholders. |
| staff | staff\_id, department, name, age | Represents university employees supporting various administrative, operational, and technical functions. |
| degree | degree\_id, name, level | Represents academic qualifications awarded upon successful completion of a program, signifying mastery in a specific field. |
| program | program\_id, program\_name, program\_­duration, specialization | Represents a planned course of study leading to a specific degree, consisting of various courses and learning experiences. |
| course | course\_id, course\_name, domain, credit\_hours | Represents individual units within a program focusing on specific topics and delivering essential knowledge and skills. |
| tuition | tuition\_id, duration, fee\_amount | Represents the financial fees charged by the university for attending a program. |
| alumni | alum\_id, alumni\_name, alumni\_contact\_info, alumni\_grad\_year, alumni\_program | Represents individuals who have graduated from the university, maintaining an ongoing connection with the institution. |
| scholarship | scholarship\_id, name, max\_amount\_sanctionable, eligibility\_criteria | Represents financial aid awarded to students based on merit, need, or specific criteria, supporting their academic pursuits. |
| student | student\_id, name, age | Represents individuals currently enrolled in the university's programs, pursuing educational qualifications and development. |
| housing | unit\_number, rent, address, is\_shared | Represents accommodation option chosen by the students |

Table 1: Entities and attributes used for conceptual model and their justification

## Initial Entity-Relationship diagram

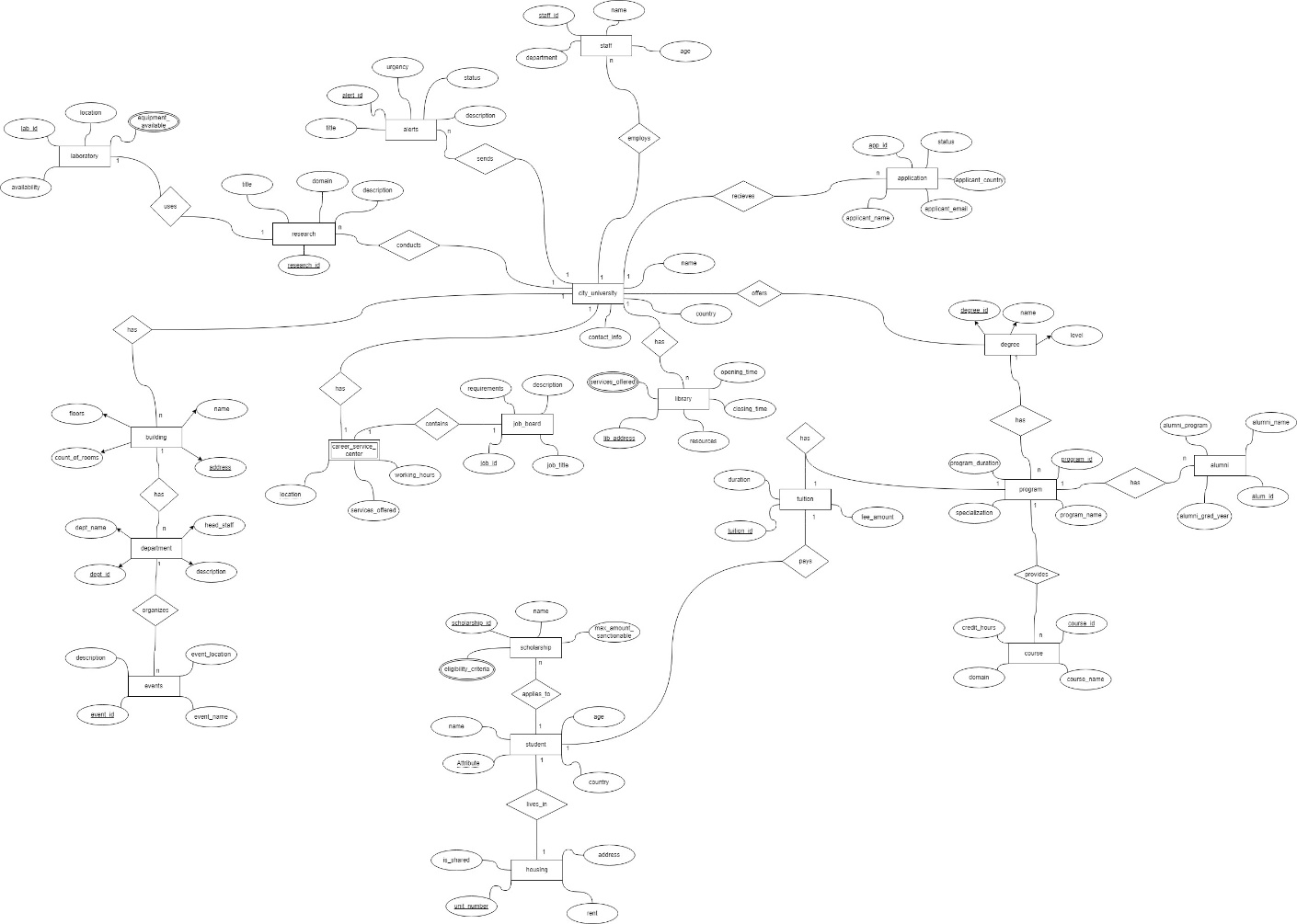


Figure 1: Initial ER diagram

Link for the full image: [ERD\_Initial](https://dalu-my.sharepoint.com/:i:/r/personal/as589490_dal_ca/Documents/CSCI%205408%20Assignment%20-%201/ERD_Initial.jpg?csf=1&web=1&e=9S8s3X)

## Identification Of Design Issues and Modifications

**Issue – 1**: A chasm trap has been identified within the Degree, Program, and Course entities. In the existing ERD, while Degree is associated with Program, and Program offers Course, there lacks a clear definition of the relationship where Course also pertains to Degree, thereby creating ambiguity. To fix this issue, we can introduce a new relationship between Course and Degree entities, clarifying the connection between them. This relationship can be established as "course belongs to degree," representing a many-to-one (N:1) relationship, where multiple courses are affiliated with a single degree.

**Issue – 2**: The alumni entity in the ER diagram is an ever-changing entity which relies on historical data of students who graduate from the university. So, this entity has the issue of time-variant data. This can be resolved by maintaining history of alumni in a separate entity.

## Final Entity-Relationship diagram

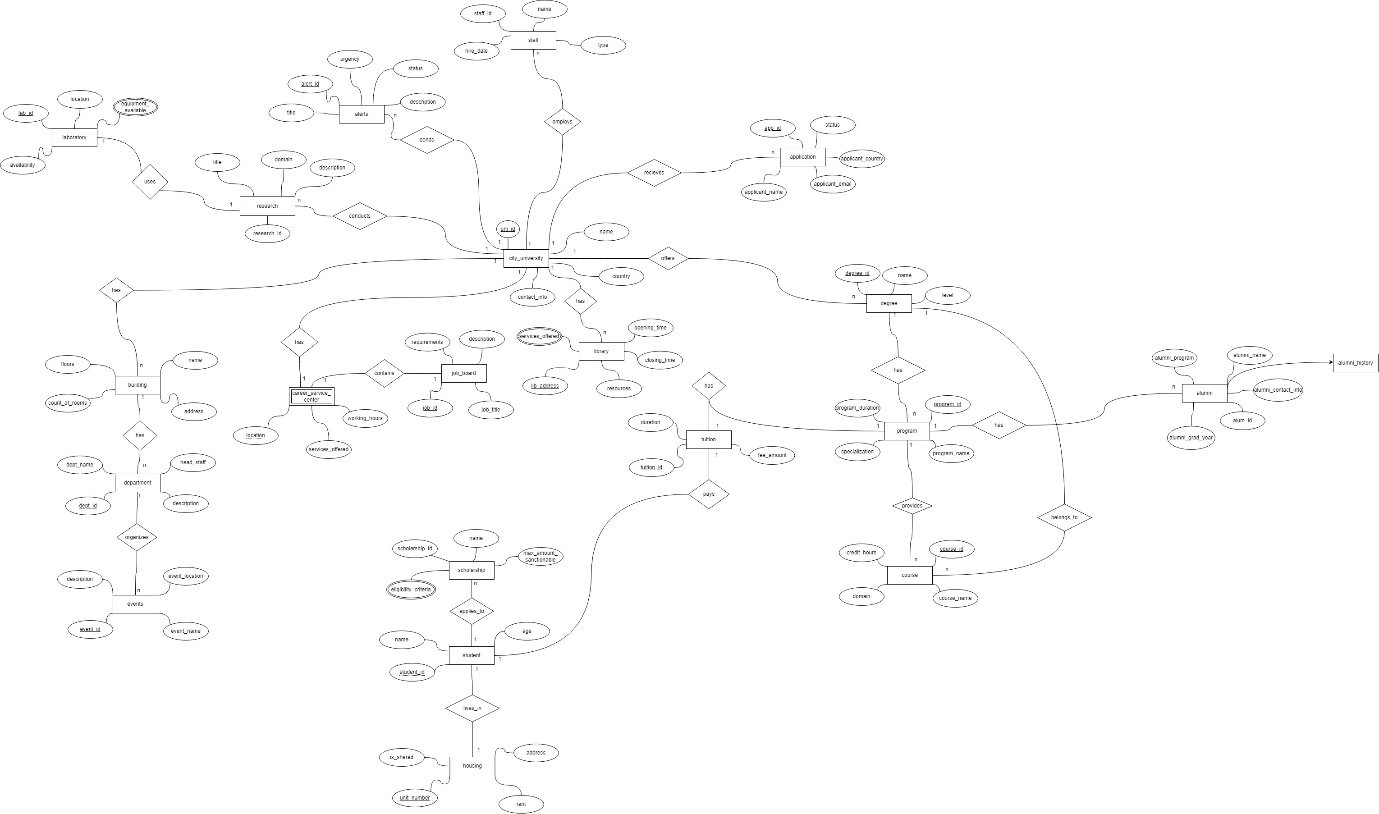


Figure 2: ER diagram after fixing design issues

Link to full image: [ERD\_Final](https://dalu-my.sharepoint.com/:i:/r/personal/as589490_dal_ca/Documents/CSCI%205408%20Assignment%20-%201/ERD_Final.jpg?csf=1&web=1&e=i51tjf)

## Enhanced Entity-Relationship diagram

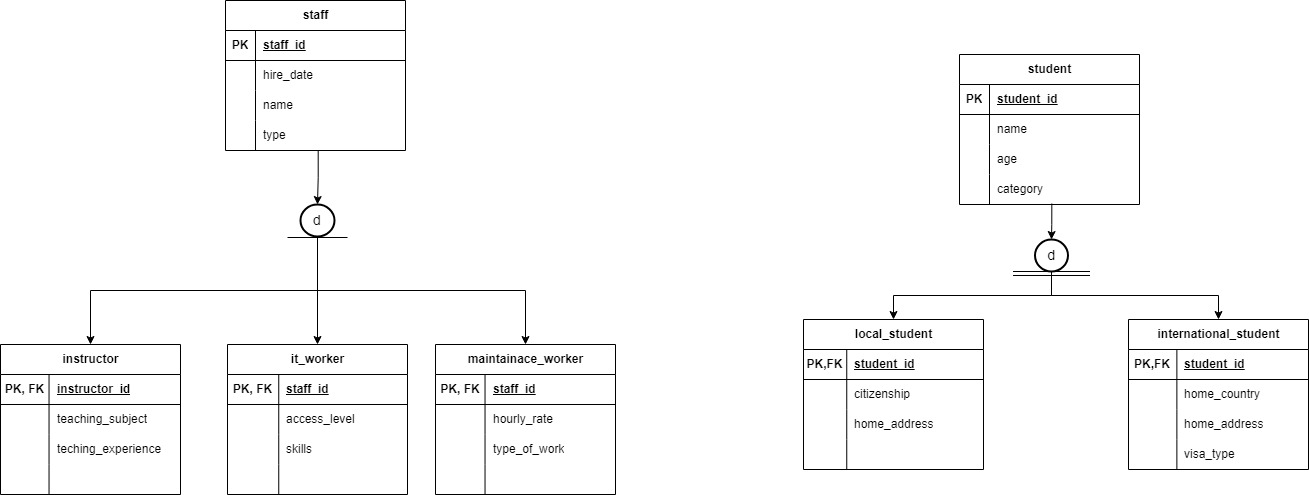


Figure 3: Extended/Enhanced Entity-Relationship Diagram

Link for the full image: [Final\_EERD](https://dalu-my.sharepoint.com/:i:/r/personal/as589490_dal_ca/Documents/CSCI%205408%20Assignment%20-%201/Final_EERD.jpg?csf=1&web=1&e=kIEXr5)

In the final ER diagram, it is possible to extend entities such as staff and student a further. This extension of few entities by adding subtypes leads to creation of specialization hierarchy. In this case of university, student can be sub divided into “local\_student” and “international\_student”, since a student can either be a local or international, disjoint and complete participation constraints apply here. Staff entity can be further divided into subtypes “instructor”, “it\_worker” and “maintainance\_worker”, since this not an exhaustive list and one type of staff cannot be in other staff role, disjoint and partial participation constraints apply here.

# Problem – 2:

## Design principles

The code that is written to implement a light weight database management system (DBMS) is built upon some standard coding principles which helps to write clean, organized and easy-to-change code. The core principles the were followed to solve this problem are Object oriented principles and few of the SOLID principles.

|  |  |
| --- | --- |
| **Principle** | **Description** |
| Single Responsibility Principle | A class (a chunk of code) should do one thing and one thing only. |
| Open-Closed Principle | Code should be easily extended (add new features) without changing what's already there. |
| Liskov Substitution Principle | If you're using a parent class, you should be able to swap in a child class without breaking anything. |
| Interface Segregation Principle | Break down big, bulky interfaces (code contracts) into smaller, more specific ones. |
| Dependency Inversion Principle | Code should depend on ideas (interfaces) and not concrete details. |

Table 2: SOLID design principles

## Evidence of Testing and Testcases

### Task – A

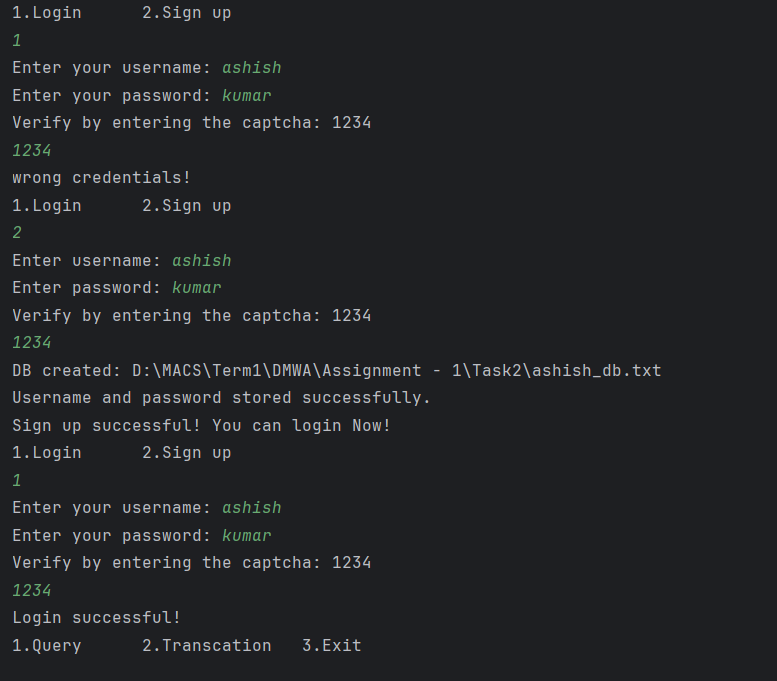


Figure 4: Testing Authentication functionality

### Task – B1

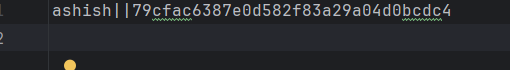


Figure 5: Storing of user details in text file using custom delimiter

### Task – B2

#### CREATE command

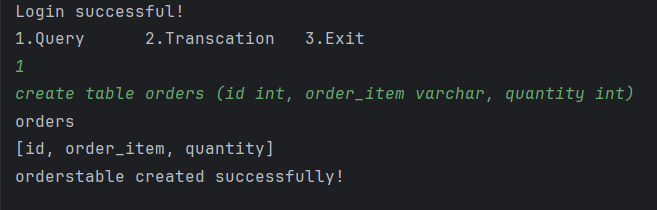


Figure 6: User inputs Create command with column names

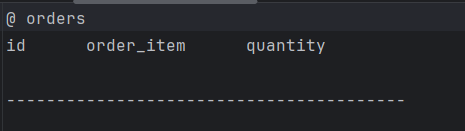


Figure 7: Table gets created in text file with custom delimiters @ and ------

#### INSERT command

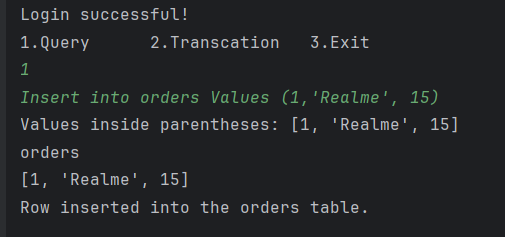


Figure 8: User inputs insert command with desired values

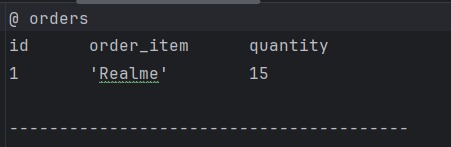


Figure 9: Row gets inserted into table with given values in the insert command

#### SELECT command

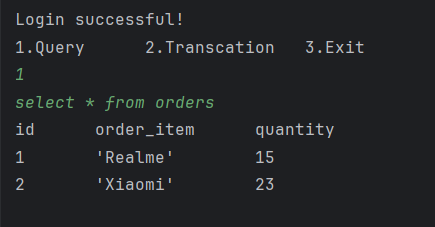


Figure 10: User inputs Select command and the rows from the table are displayed

### Task – C

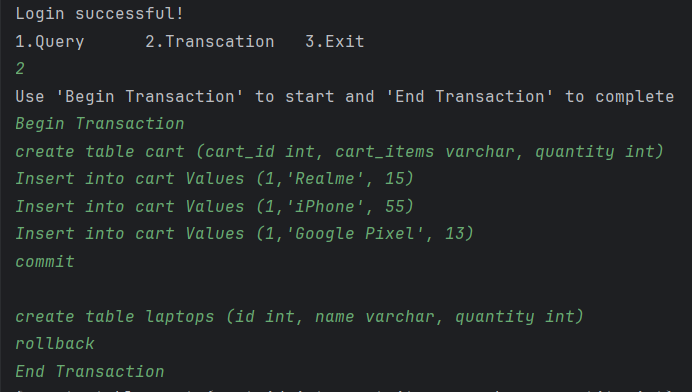


Figure 11: Transaction input with insert, create, commit and rollback commands

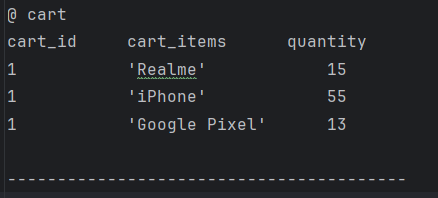


Figure 12: Output of the given transaction statements (rollback was performed and new table was not created)

# References

[1] Educative, "What are the SOLID principles in Java?", educative.io. https://www.educative.io/answers/what-are-the-solid-principles-in-java (Accessed: Feb 26, 2024)