Data Management and Warehousing Analytics (CSCI5408), Winter 2023 – Assignment 1

Part 1:

Step 1: Completed checking the dal.ca website to find the required unique entities.

Step 2: The entities I chose are Campus, Accommodation, Employee/Staff, Research and Innovation, Food Outlets, Food outlet menus, Clubs, club events, Gym, Library, Department/Faculty, Student, Course

Step 3: The Chen model ERD is attached along with the report it was done using draw.io. The assumptions I have taken are mentioned below:

- There are multiple campuses.
- There are multiple gyms.
- There are multiple libraries.
- There are multiple accommodations.
- There are multiple clubs.
- Clubs can only hold one event per time.
- Food outlets need to have at least 1 menu.
- Employee must belong to a faculty.
- Faculty must belong to a campus.

Step 4: A few of the relationships where M:N in reality but I converted them to 1:N relations to make a meaningful logical model

Step 5:

- Campus

CampusID(PK)	Name	Address

The campus table defines all the Dalhousie campuses (Agri Truro and Halifax) with their name and addresses.

- Accommodation

BuildingNo (PK)	Name	Address	CampusID(FK)

This table includes information about on campus accommodations it has the BuildingNo as the primary key as all the buildings are assigned a unique number apart from this it displays the name of the accommodation and the address of it. The campus ID links it to the campus where it is present.

- Employee/Staff

EmployeeID(PK)	Name	Email	Phone	Address	FacultyID(FK)

This was normalized to

EmployeeID(PK)	First Name	Last Name	Email	Primary Phone	Secondary Phone	Address	FacultyID(FK)

The Employee/Staff table has all the persona information of all the employees working in the campus for the university

- Research and Innovation

ResearchID(PK)	Name	Description	Research Head	CampusID(FK)

The research is funded by the campus where it is being done, so to know which campus has funded the research we have campusID as the foreign key , the research head here is the go to person if you want to know anything about the research as he is the one leading the research team.

- Food Outlets

RestaurantID(PK)	Name	Address	Timings	CampusID(FK)

- Food Outlet menu

Food Item	Price	RestaurantID(PK,FK)

These both together define the food places on campus (eg tim hortons and subway). The food outlet must have a menu is the assumption that I have taken

- Clubs

ClubID(PK)	Name	Type	Club Head	CampusID(FK)

- Club Event

Name	Description	Date	ClubID(PK,FK)

Club and Club events are the tables that give you information on all the clubs present on campus, the assumption done here is that a club can only have one event at a time

- Gym

BuildingNo(PK)	Name	Address	Timings	CampusID(FK)

Halifax campus alone has 2-3 gyms and this table shows you the information about them

- Library

BuildingNo(PK)	Name	Address	Timings	CampusID(FK)

Dalhousie has many places to study in peace and the library information can be found here

- Department/Faculty

FacultyID(PK)	Name	Dean	Office/Address	Email	Phone	CampusID(FK)

This was normalized to

FacultyID(PK)	Name	Dean	Office/Address	Email	Primary Phone	Secondary Phone	CampusID(FK)

Hold the information of all the faculties present on campus

- Student

StudentID(PK)	First Name	Middle Name	Last Name	Email	Phone	Address	FacultyID(FK)

Holds personal information of all the students studying at the university

- Course

CourseID(PK)	Name	Description	FacultyID(FK)

This table gives you information about the courses Dalhousie has on offer by multiple faculties

Step 6: The normalization of the required tables was done above.

Step 7: The report has a attached DDLStatements.sql file that contains the DDL statements

Part 2:

The whole project was done in java and all the required functionalities where separated into their specific classes, if the functionality has different use cases different functions where created per functionality.

Camel casing naming convention was followed for functions and variables, pascal casing was used for classes.

Comments where provided wherever required.

Recursion has been avoided wherever possible.

Proper indentation has been done throughout the project.

Project Details:

- Language Used: Java
- Hashing method used: md5
- Functionalities implemented: show, use, desc, drop, delete, insert, delete, create, select, update, truncate
- Personalized delimiters used: :_P_: for tables, :: for differentiating datatypes and values in create
- Hide password: people can't view the password when you are typing
- Force the user to get into a schema after login by default so that folder structure is maintained
- **<u>Limitations</u>**:
  1. Although the metadata of the file is saved has the proper datatypes all the inputs are taken as string itself instead of the actual datatype.
  2. While some of the error handling is done you might face program exit if any other syntax is followed apart from the ones given below
  3. All the functionalities do not have all the variation that an actual DBMS has.
  4. Only one query is processed per time.

Functionalities Overview and Syntaxes:

None of the commands need to end with a semi colon, they might not work properly if you add semi colon after the query

(PS: Do not add any inverted commas to any query, and please avoid any other whitespaces except the ones shown in the syntax)

- Create Functionality: Use to create a new table and its metadata
  1. Table Creation
     Syntax: create table table_name column_1::datatype_1,column_2::datatype_2…
     Example: create table table phone model::string,price::double
     (PS: Please make sure there is no gap between the commas "," when you are entering the value datatype pairs)
  2. Schema Creation
     Syntax: create schema schema_name
     Example: create schema university
- Delete Functionality: use to delete a single row in an existing table

  Syntax: delete from table_name where column_name=value

  Example: delete from person where name=pratik
- Drop Functionality: delete whole table or schema
  1. Table Dropping
     Syntax: drop table table_name
     Example: drop table user
  2. Schema Dropping
     Syntax: drop schema schema_name
     Example: drop schema university
- Description Functionality: show metadata of a table
  Syntax: desc table_name
  Example: desc person
- Truncate Functionality: delete everything in the table except column headers
  Syntax: truncate table table_name
  Example: truncate table person

- <u>Insert Functionality</u>: insert values into an already existing table
  Syntax: insert into table_name values value1,value2,value3
  Example: insert into person values Pratik,3245234,pra@da.ca
  (PS: please enter all the columns of the table so that the table format is proper **Example**: insert into person values Pratik,,)
- <u>Select Functionality</u>: display contents of a existing table according to various conditions
  1. Output all rows

     Syntax: select * from table_name

     Example: select * from person
  2. Output all rows where condition matches
     Syntax: select * from table_name where column_name=value
     Example: select * from person where name=Pratik
  3. Output selected columns
     Syntax: select column1,column2,.. from table_name
     Example1: select name,email from person
     Example2: select name from person
  4. Output selected columns where conditions match
     Syntax: select column1,column2,.. from table_name where column_name=value
     Example: select name,email from person where phone=75653245
- <u>Show Functionality</u>: display various parameters of the DBMS
  1. Users
     Syntax: show users
  2. Schemas
     Syntax: show schemas
  3. Tables
     Syntax: show tables
- <u>Update Functionality</u>: change values of cells in an existing database according to a condition
  Syntax: update table_name set column1=value,column2=value,.. where column_name=value
  Example1: update person set name=patil where name=Pratik

Example2: update person set name=Pratik,phone=345656 where email=pra@da.ca

References:

- GeeksForGeeks, “List Interface in Java with Examples” GeeksForGeeks.com, [Online].
  Available: https://www.geeksforgeeks.org/list-interface-java-examples/ [Accessed: 22-Feb-2023]
- JavaTPoint, “Java MD5 hashing example” javatpoint.com, [Online].
  Available: https://www.javatpoint.com/java-md5-hashing-example [Accessed: 22-Feb-2023]