1. **Project Title**

Family Tree Database

1. **Team**

* Paramjot Kaur (MySQL Expert)
* Alisher Matianiu (MS SQL Server Expert)

1. **Weekly Meeting Hours**

We will meet and work on the project every Wednesday from 15:00 to 18:00.

1. **Project Description**

In this project we will be designing Family Tree database which will store the full family information like a personal information of each family member, relationships with other members, history of education, work, residence and special events, marital information, children information if any.

Based on our requirement gathering and brainstorming, we formed entities and relationships listed below. Please Note: this is a draft version and information can be updated in later steps.

* **Person (strong entity)**: This entity will contain the basic personal information of a family member and will have the following attributes:

SSN

Name (FName, MName, LName)

Date of Birth

Gender

Picture

Date of demise, if applicable

* **Institution (strong entity)**: This entity will be responsible for storing education details of a person using the attributes listed below:

Institution No

Institution Name

**Institution Address No**

Contact No

* **Education (weak):**

**Person SSN**

**Institution No**

Degree

GPA

Start Date

Completion Date

* **Company (strong entity):** This entity will contain the information about the work history of a person using the following attributes:

Company No

Company Name

**Company Address No**

Contact No

* **Job (weak entity):**

**Person SSN**

**Company No**

Position

Start Date

End Date, if applicable (e.g. getting promoted, moving to another company)

* **Address (strong)**: This entity will contain the list of residential, institution and company addresses. Attributes are as follows:

Address No (surrogate key)

Address Line (e.g. 25-4660 Berkeley st.)

City

Province/State

Country

Postal Code

* **Residency (weak):** This entity will hold the current and past residential addresses of a person.

**Person SSN**

**Residential Address No**

Date of Entry

* **Marries (recursive relationship of Person)**: This entity will hold the information of the person’s marriages and divorces. Its attributes are as follows:

**Person 1 SSN**

**Person 2 SSN**

Marriage Date

* **Divorces (recursive relationship of Person):**

**Person 1 SSN**

**Person 2 SSN**

Divorce Date

* **Event (strong):** This entity will hold information about the person’s special events using the following attributes:

**Person SSN**

Event date

Event description

* **Is mother of (recursive relationship of Person):**

**Person 1 SSN** (mother)

**Person 2 SSN** (child)

* **Is father of (recursive relationship of Person):**

**Person 1 SSN** (father)

**Person 2 SSN** (child)

**Note:** Below relationships can be derived using above entities

- sister, brother, uncle, aunt, cousins

**Example:** finding the brother of a person -> find father -> children with Gender Male are his brothers.

**Example:** Uncle relationship can be driven from -> the person's father brother

In the similar way we can find out other relationships as well.

1. **Assumptions about Cardinality and Participations**

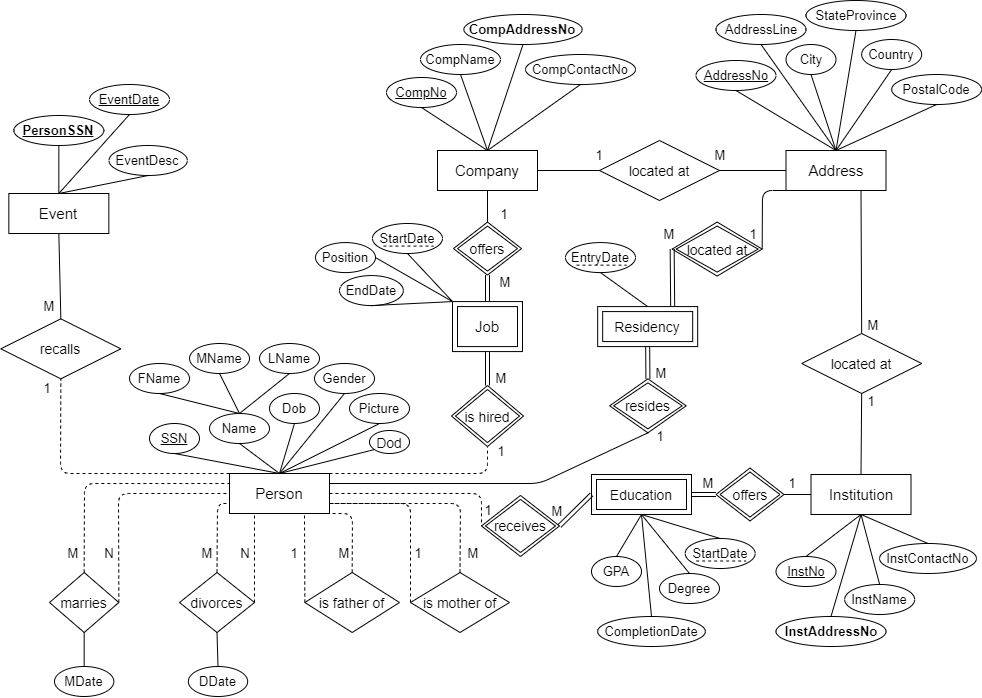
Based on our discussion and requirement gathering and planning, below are assumptions that we have made:

* (Person, Institution) A person can have zero or more qualifications.
* (Person, Institution) A person can study only at one institution at a time.
* (Person, Institution) A person can transfer to another institution.
* (Person, Company) A person can have zero or more employers at the same time, i.e. person might be doing many jobs at a time or may be unemployed.
* (Person, Company) A person cannot have multiple positions in a company at the same time.
* A person can be studying and working at the same time.
* (Person, Marriage/Divorce) A person can have history of marriages (i.e. previous marriages) but a person can be married to only one person at a time.
* (Person, Motherhood, Fatherhood) A person can have children from different marriages.
* (Person) SSN is never changed.
* (Person, Marriage, Divorce) A person cannot marry the same person again.
* (Person, Residency) A person can move back to the previous address. (e.g. children studying in a different city move back after graduation)
* (Person, Residency) The date of entry for person’s very 1st residential address is his/her birth date.

**Relationships**

* Person can be a parent
* Person studies at institution
* Person works for a company
* Person can marry and/or divorce
* Person can have events
* Person, company and institution have residential addresses

1. **EER Modeling Diagram**



1. **ER-Model Mapping to Database Relational Schema**

Person (SSN, FName, MName, LName, Dob, Gender, Picture, Dod)

Address (AddressNo, AddressLine, City, StateProvince, Country, PostalCode)

Institution (InstNo, **InstAddressNo**, InstName, InstContactNo)

Company (CompNo, CompName, **CompAddressNo**, CompContactNo)

Event (**PersonSSN**, EventDate, EventDesc)

Residency (**PersonSSN**, **AddressNo**, EntryDate)

Education (**PersonSSN**, **InstNo**, StartDate, CompletionDate, Degree, GPA)

Job (**PersonSSN**, **CompNo**, StartDate, EndDate, Position)

Person\_Marries\_Person (**P1\_SSN**, **P2\_SSN**, MDate)

Person\_Divorces\_Person (**P1\_SSN**, **P2\_SSN**, DDate)

Person\_Father\_Person (**P1\_SSN**, **P2\_SSN**)

Person\_Mother\_Person (**P1\_SSN**, **P2\_SSN**)

1. **Normalization**

1NF: all entities are in 1st Normal Form since every attribute has only 1 value.

2NF: all attributes are uniquely identified by their whole keys.

3NF: because of transitive functional dependency (AddressNo > PostalCode > {StateProvince, City, Country}) the Address entity should be decomposed as follows:

Address (AddressNo, AddressLine, **PostalCode**)

Postal\_Info (PostalCode, City, StateProvince, Country)

BCNF: Already in Boyce-Codd Normal Form, since there is no transitivity in Address and Postal\_Info entities.

1. **Determining Data Types (Domain) and Constraints**

INT data type was used for surrogate keys such as Address\_No, Inst\_No and Comp\_No.

CHAR(distinct) data type was used for fields with distinct number of characters, i.e. CHAR(6) for Postal\_Code, making the postal code always 6 characters long.

DATE type was used to keep dates, for example Start\_date and End\_date fields in Job table.

DECIMAL(3, 2) was used for storing GPA in Education table.

VARCHAR(maximum number of characters) was used for many fields in our database, since they have variable length. A typical example can be FName and LName, both using VARCHAR(20).

1. **Creating Database and Tables – SQL DDL**

**Create.txt** - script for creating tables.

**Drop.txt** - script for dropping table.

1. **Inserting Values in Tables**

**Insert.txt** - script for inserting values in tables.

1. **SQL Queries and Views (13)**

**Queries.txt** – file containing queries and views.