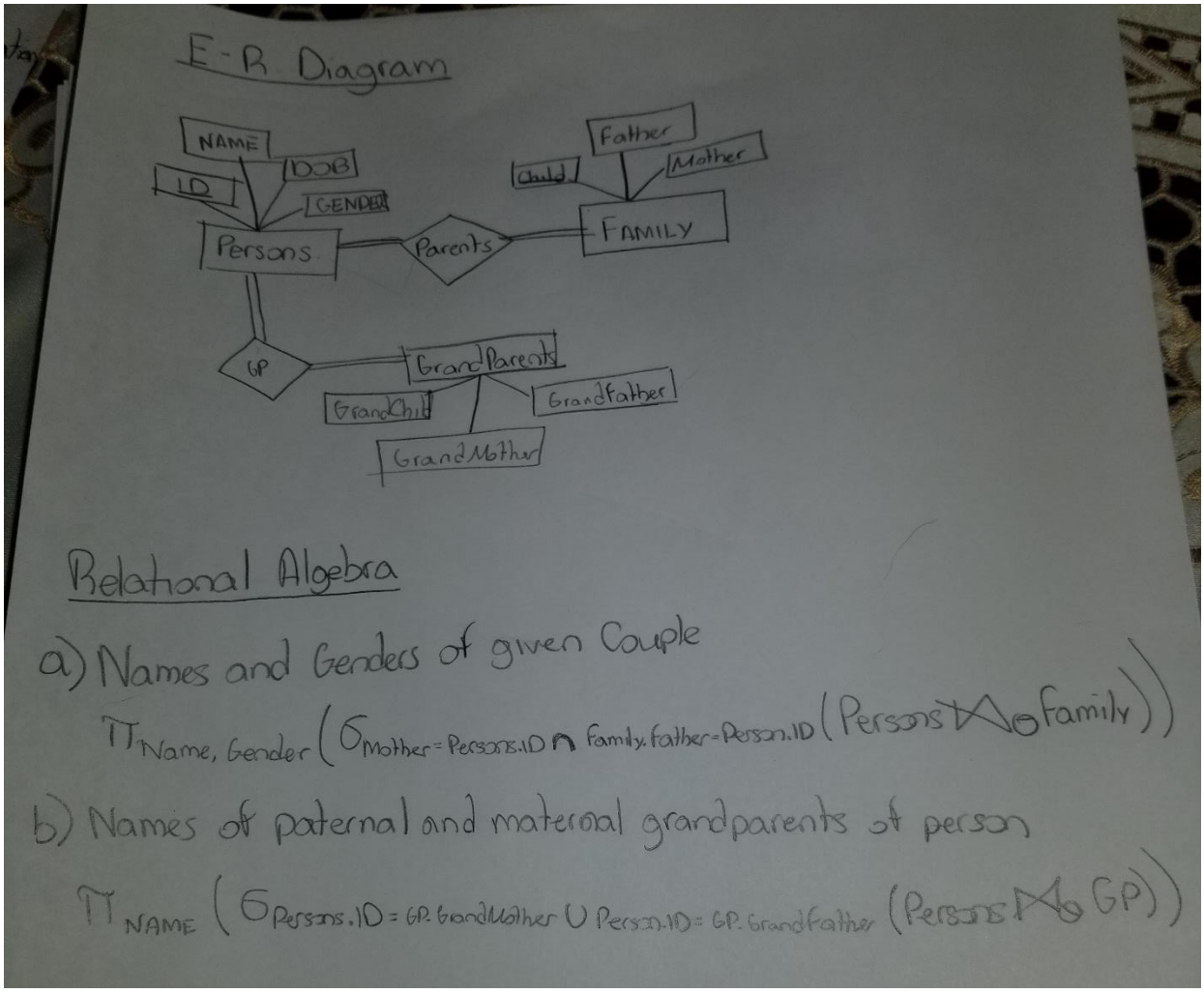
Jason Mejia 6/21/20

CSC 336 Assignment 1

Family Relations

1) We are given the task for creating a E-R diagram and relational algebra expression for the relations of PERSONS and FAMILY. PERSONS will contain the attributes of ID, name, date of birth and the gender. FAMILY will the person from PERSONS and the father and mother of that specific person.
2) Here is the solution for the E-R diagram and the Relational algebra expression for the relations of PERSONS and FAMILY.
   For the code itself I have used a sql server compiler for the solution. Tables for PERSONS and FAMILY. To get the names for the children for specific couples, I have used a function for that purpose. It uses information from the tables PERSON and FAMILY. Same is said for the next function of finding the grandparents of a given person.



3) *--This is creating the table for Persons that contains the ID, Name, DOB, and Gender for each person*

```
CREATE TABLE Persons (
   ID INT PRIMARY KEY,
   Name VARCHAR(20) NOT NULL,
   DOB VARCHAR(20) DEFAULT 'Unknown',
   Gender VARCHAR(1) CHECK(Gender = 'M' OR Gender = 'F')
);

-- Inserting each person part of the family
INSERT INTO Persons VALUES(1, ‘Jason’, '11/30/1996', 'M');
INSERT INTO Persons VALUES(2, ‘Nancy’, ‘12/01/1989’, ‘F’);
INSERT INTO Persons VALUES(3, ‘Alfredo’, ‘03/03/1987’, ‘M’);
INSERT INTO Persons VALUES(4, ‘Raquel’, ‘04/27/1961’, ‘F’);
INSERT INTO Persons VALUES(5, ‘Giselo’, ‘05/07/1960’, ‘M’);
INSERT INTO Persons VALUES(6, ‘Evelyn’, ‘08/16/2012’, ‘F’);
INSERT INTO Persons VALUES(7, ‘Julian’, ‘11/29/2014’, ‘M’);
INSERT INTO Persons VALUES(8, ‘Chris’, ‘12/02/1989’, ‘M’);
INSERT INTO Persons VALUES(9, ‘Irene’, ‘02/16/1908’, ‘F’);
INSERT INTO Persons VALUES(10, ‘Hector’, ‘6/28/1908’, ‘M’);
INSERT INTO Persons VALUES(11, ‘Bertha’, ‘6/28/1943’, ‘F’);
INSERT INTO Persons VALUES(12, ‘Michael’, ‘6/28/1943’, ‘M’);
INSERT INTO Persons VALUES(13, ‘Lisa’, ‘8/15/1996’, ‘F’);
INSERT INTO Persons VALUES(14, ‘Oliver’, ‘3/21/2021’, ‘M’);

--Displaying all the content of the table Persons
SELECT * FROM Persons;

--Creating the table for the child and each of their parents
CREATE TABLE Family (
   Child INT FOREIGN KEY REFERENCES Persons(ID),
   Father INT FOREIGN KEY REFERENCES Persons(ID),
   Mother INT FOREIGN KEY REFERENCES Persons(ID),
```

```
    PRIMARY KEY(Child)
);

--Inserting every child to their parents from the table of Persons
INSERT INTO Family VALUES(1, 5, 4);
INSERT INTO Family VALUES(2, 5, 4);
INSERT INTO Family VALUES(3, 5, 4);
INSERT INTO Family VALUES(6, 8, 2);
INSERT INTO Family VALUES(7, 8, 2);
INSERT INTO Family VALUES(5, 10, 9);
INSERT INTO Family VALUES(4, 12, 11);
INSERT INTO Family VALUES(14, 1, 13);

--Displaying all the contents of table Family
SELECT * FROM Family;

/* The function ChildrenOf will take two parameters of INT
These are the Mother and Father, the output will give all the children of that specific couple */
GO
CREATE FUNCTION ChildrenOf(@Mother INT, @Father INT)
RETURNS @par TABLE (
    Child_ID INT,
    Children VARCHAR(20),
    Gender VARCHAR(20)
)
AS
BEGIN
    INSERT INTO @par(Child_ID, Children, Gender)
    SELECT Family.Child, Persons.Name, Persons.Gender FROM Family
    JOIN Persons
    ON Persons.ID = Family.Child
    WHERE Mother = (SELECT ID FROM Persons WHERE ID = @Mother)
```

```
        AND Father = (SELECT ID FROM Persons WHERE ID = @Father);
RETURN
END
GO

--Call the function ChildrenOf with 4 and 5 which are Raquel and Giselo
SELECT * FROM dbo.ChildrenOf(4, 5);
--Call the function ChildrenOf with 2 and 8 which are Nancy and Chris
SELECT * FROM dbo.ChildrenOf(2, 8);
--Call the function ChildrenOf with 4 and 5 which are Irene and Hector
SELECT * FROM dbo.ChildrenOf(9, 10);
--Call the function ChildrenOf with 4 and 5 which are Bertha and Michael
SELECT * FROM dbo.ChildrenOf(11, 12);
--Call the function ChildrenOf with 1 and 13 which are Lisa and Jason
SELECT * FROM dbo.ChildrenOf(13, 1);

--Creating a table for a person and each of their respective grandparents
CREATE TABLE GP (
   GrandChild INT FOREIGN KEY REFERENCES Persons(ID),
   GrandMother INT FOREIGN KEY REFERENCES Persons(ID),
   GrandFather INT FOREIGN KEY REFERENCES Persons(ID),d
);

--Inserting all the values for the table GP, the Grandchild and Grandparents
INSERT INTO GP VALUES(6, 4, 5);
INSERT INTO GP VALUES(7, 4, 5);
INSERT INTO GP VALUES(1, 9, 10);
INSERT INTO GP VALUES(1, 11, 12);
INSERT INTO GP VALUES(2, 9, 10);
INSERT INTO GP VALUES(2, 11, 12);
INSERT INTO GP VALUES(3, 9, 10);
INSERT INTO GP VALUES(3, 11, 12);
```

```
--Displaying all the contents of the table GP
SELECT * FROM GP;

/* This function only takes one parameter, it is the granchild with data type INT
The outcome of the function will give the GrandParents of that specific GrandChild*/
GO
CREATE FUNCTION GrandParentsOf (@GC INT)
RETURNS @gpar TABLE (
   GP_ID INT,
   GrandParents VARCHAR(20)
)
AS
BEGIN
   INSERT INTO @gpar(GP_ID, GrandParents)
   SELECT Persons.ID, Persons.Name FROM GP
   JOIN Persons
   ON Persons.ID = GP.GrandMother OR Persons.ID = GP.GrandFather
   WHERE GP.GrandChild = (SELECT ID FROM Persons WHERE ID = @GC)
RETURN
END
GO

--Call the function GrandParentsOf with input 6 which is Evelyn
SELECT * FROM dbo.GrandParentsOf(6);
--Call the function GrandParentsOf with input 7 which is Julian
SELECT * FROM dbo.GrandParentsOf(7);
--Call the function GrandParentsOf with input 1 which is Jason
SELECT * FROM dbo.GrandParentsOf(1);
--Call the function GrandParentsOf with input 2 which is Nancy
SELECT * FROM dbo.GrandParentsOf(2);
--Call the function GrandParentsOf with input 3 which is Alfredo
SELECT * FROM dbo.GrandParentsOf(3);
```

4)

	ID	Name	DOB	Gender
1	1	Jason	11/30/1996	M
2	2	Nancy	12/01/1989	F
3	3	Alfredo	03/03/1987	M
4	4	Raquel	04/27/1961	F
5	5	Giselo	05/07/1960	M
6	6	Evelyn	08/16/2012	F
7	7	Julian	11/29/2014	M
8	8	Chris	12/02/1989	M
9	9	Irene	02/16/1908	F
10	10	Hector	6/28/1908	M
11	11	Bertha	6/28/1943	F
12	12	Michael	6/28/1943	M
13	13	Lisa	8/15/1996	F
14	14	Oliver	3/21/2021	M

```
31 --Dispalying all the content of the table Persons
32 SELECT * FROM Persons;
```

	Child	Father	Mother
1	1	5	4
2	2	5	4
3	3	5	4
4	4	12	11
5	5	10	9
6	6	8	2
7	7	8	2
8	14	1	13

```
52 --Displaying all the contents of table Family
53 SELECT * FROM Family;
54
```

	Child_ID	Children	Gender
1	1	Jason	M
2	2	Nancy	F
3	3	Alfredo	M

```
76 --Call the function ChildrenOf with 4 and 5 which are Raquel and Giselo
77 SELECT * FROM dbo.ChildrenOf(4, 5);
```

	Child_ID	Children	Gender
1	6	Evelyn	F
2	7	Julian	M

```
78 --Call the function ChildrenOf with 2 and 8 which are Nancy and Chris
79 SELECT * FROM dbo.ChildrenOf(2, 8);
```

	Child_ID	Children	Gender
1	5	Giselo	M

```
80 --Call the function ChildrenOf with 4 and 5 which are Irene and Hector
81 SELECT * FROM dbo.ChildrenOf(9, 10);
```

	Child_ID	Children	Gender
1	4	Raquel	F

```
82 --Call the function ChildrenOf with 4 and 5 which are Bertha and Michael
83 SELECT * FROM dbo.ChildrenOf(11, 12);
```

	Child_ID	Children	Gender
1	14	Oliver	M

```sql
--Call the function ChildrenOf with 1 and 13 which are Lisa and Jason
SELECT * FROM dbo.ChildrenOf(13, 1);
```

	GrandChild	GrandMother	GrandFather
1	6	4	5
2	7	4	5
3	1	9	10
4	1	11	12
5	2	9	10
6	2	11	12
7	3	9	10
8	3	11	12

```sql
--Displaying all the contents of the table GP
SELECT * FROM GP;
```

	GP_ID	GrandParents
1	4	Raquel
2	5	Giselo

```sql
--Call the function GrandParentsOf with input 6 which is Evelyn
SELECT * FROM dbo.GrandParentsOf(6);
```

	GP_ID	GrandParents
1	4	Raquel
2	5	Giselo

```sql
--Call the function GrandParentsOf with input 7 which is Julian
SELECT * FROM dbo.GrandParentsOf(7);
```

	GP_ID	GrandParents
1	9	Irene
2	10	Hector
3	11	Bertha
4	12	Michael

```sql
--Call the function GrandParentsOf with input 1 which is Jason
SELECT * FROM dbo.GrandParentsOf(1);
```

	GP_ID	GrandParents
1	9	Irene
2	10	Hector
3	11	Bertha
4	12	Michael

```sql
--Call the function GrandParentsOf with input 2 which is Nancy
SELECT * FROM dbo.GrandParentsOf(2);
```

	GP_ID	GrandParents
1	9	Irene
2	10	Hector
3	11	Bertha
4	12	Michael

```sql
--Call the function GrandParentsOf with input 3 which is Alfredo
SELECT * FROM dbo.GrandParentsOf(3);
```