```sql
USE master
IF DB_ID('MyDB_EricWilliamson') IS NOT NULL
BEGIN
	ALTER DATABASE MyDB_EricWilliamson SET OFFLINE WITH ROLLBACK IMMEDIATE;
	ALTER DATABASE MyDB_EricWilliamson SET ONLINE;
	DROP DATABASE MyDB_EricWilliamson;
END
/*
1.	Create a new database called MyDB_[First and Last Name].
Example: MyDB_EricWilliamson
Add the following script after the create database command:
GO
USE [Your database name]
(The GO statement tells SQL to run any code that follows it as a separate statement. This will allow me
to run your entire homework script in one pass.)

*/
CREATE DATABASE MyDB_EricWilliamson
GO
USE MyDB_EricWilliamson

/*
2.	Copy all records and columns from the Chinook.dbo.Customer table into a new table in your
database called “Users”.
Include the USE keyword to select your new database.
Hint: Let SQL build the table for you using your SELECT statement.
*/
SELECT *
INTO Users
FROM Chinook.dbo.Customer

/*
3.	Delete all records from the Users table that have an odd CustomerId number .
Hint: There is a math operator to help you with this.
*/
DELETE Users
WHERE CustomerId%2 = 1

/*
4.	Update the Company column in the User table.
If the User Email contains “gmail” then write “Google” as the company.
If the User Email contains “yahoo” then write “Yahoo!” as the company.
Otherwise leave the current value for Company as is.
Hint: You can achieve this without using a WHERE clause.
*/
UPDATE Users
SET Company = CASE
	WHEN Email LIKE '%gmail%' THEN 'Google'
	WHEN Email LIKE '%Yahoo%' THEN 'Yahoo!'
	ELSE Company
	END

/*
5.	Rename the Users CustomerId column to “UserId”.
*/
EXEC sp_rename 'Users.CustomerId', 'UserId', 'COLUMN'
```

```sql
/*
6.	Add a Primary Key to the Users table.
Set the UserId column as the primary key.
Name the primary key “pk_Users”
*/
ALTER TABLE Users
ADD CONSTRAINT pk_Users PRIMARY KEY (UserId)
/*
7.	Create an Address table in the database
Add the following columns to the table: AddresId int, AddressType varchar(10), AddressLine1
varchar(50), City varchar(30), State varchar(2), UserId int, CreateDate datetime.
Make the AddressId column the primary key for the table and also make it an identity column with a
starting point of 1 and an increment of 1.
The CreateDate column should default to the date and time a record is added to the table.
*/
CREATE TABLE Address (
	AddressId int IDENTITY(1,1) PRIMARY KEY
	,AddressType varchar(10)
	,AddressLine1 varchar(50)
	,City varchar(30)
	,State varchar(2)
	,UserID int
	,CreateDate datetime DEFAULT GETDATE()
	)
/*
8.	Add a unique constraint to the Address table.
The constraint should not allow a duplicate combination of the UserId and AddressType columns.
Name the constraint “uc_AddressType”.
*/
ALTER TABLE Address
ADD CONSTRAINT uc_AddressType UNIQUE(UserId,AddressType)
/*
9.	Add a foreign key to the Address table.
The foreign key is UserId.
The primary key is the UserId column in the Users table.
Name the constraint “fk_UserAddress”.
*/
ALTER TABLE Address
ADD CONSTRAINT fk_UserAddress FOREIGN KEY (UserId) REFERENCES Users(UserId)
/*
10.	Insert 3 records into the Address table.
Insert the following data into the correct columns on the Address table.
Hint: The AddressId and CreateDate columns should automatically populate with a value.
*/
INSERT INTO Address (AddressType,AddressLine1,City,State,UserID)
VALUES('home','111 Elm St.','Los Angeles','CA',2)
,('home','222 Palm Ave.','San Diego','CA',4)
,('work','333 Oak Ln.','La Jolla','CA',4)

/*
11.	Select all records from the Address and Users tables.
Two separate queries. One for Address and one for Users
No tricks here.  I want to see the content of your tables so I can grade the homework faster.
*/
SELECT *
FROM Address

SELECT *
FROM Users
```