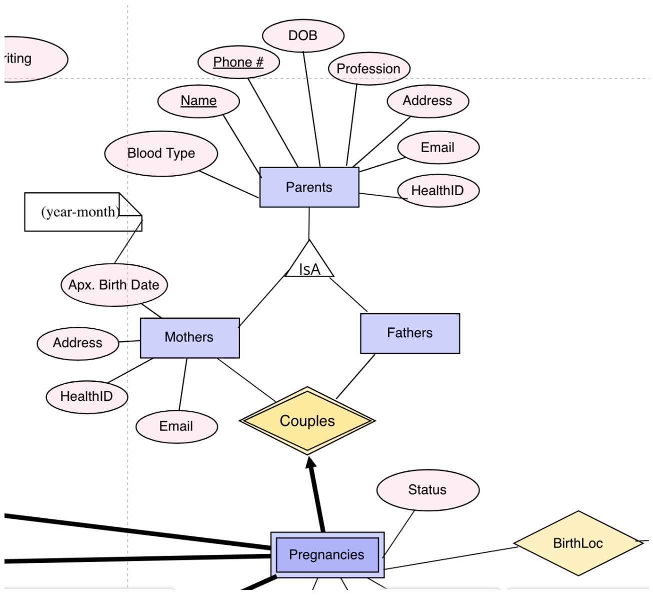
Assumptions:

1. Each couple (or mother alone) will initially register for the program they will register for at least one info session and if they Attend it (attribute of the registered_sess relationship), then their profile will be assessed by the medical staff and if they are Approved (attribute for the couple entity set), then they will be assigned a midwife by the system. This could not be added to the ER model and will be addressed by the programmer.
2. A couple only has to attend one info session, even for multiple pregnancies.
3. In the assignment handout, it is mentioned that some attributes will have to be assigned as we go ahead in the process. For example, the birth location has to be chosen later on in the process, or the couple have to attend an info session to then potentially be assigned a midwife. Due to the nature of ER models, I had to either use participation constraints which would force us to provide the value when a couple/pregnancy is created which wouldn't be possible, so instead I have decided to address these in the coding part (addressed in the restrictions part).
4. A test (either sampled by the midwife or taken somewhere else) first has to be prescribed. In the model the sampling tests are prescribed by the midwife, then they may be taken by the midwife at the same time of the prescription date or at a later date, and they are then sent to a lab to be checked. The referral tests will be prescribed by the midwife but taken somewhere else by a technician.
5. Tests are related to appointments since they are prescribed in the appointments by the midwife. The Tests entity set has a Type and For_whom attribute, which in order will refer to the type of test (blood, ultrasound, …) and the person who the test is prescribed for (mother, babies,...).
6. Each couple have to have at least one pregnancy which will be created and set to default values when they register for the program.
7. Please note that I could have used an IsA hierarchy for the parents (mothers and fathers). But since having a father is optional and some of the values for the fathers are optional (e.g. Health ID, ...) that would cause some extra complications in the model, I decided not to use an IsA hierarchy here and simply create two different entities for each parent. This would be the IsA alternative:



8. A midwife can't be hosting two different info sessions at the same time. And all info sessions have one and only one presenter.
9. The number of the pregnancy will be stored in the pregnancy entity set itself (as nth_time) and not for the couple.
10. Artificial Keys:
    a. Couple ID (Couples): used to identify couples. Needed since the couples can be a pair of people of the mother alone and have to be unique.
    b. TmpID (Babies): used to identify each baby of each pregnancy. Given that the data related to the babies will be provided gradually and test by test an artificial key will be needed to keep track of each baby of the pregnancy.
    c. ID (Appointments): used to identify appointments between each couple and their midwife.
11. Clarification on some attributes:
    a. Approved (Couples): the medical staff have to set this to true to approve a couple for the program (after they have attended the info session)
    b. Status (pregnancies): used to keep track of what state the pregnancy is at. E.g. approved, waiting for primary midwife, waiting for backup midwife, …
    c. Type (Tests): refers to the type of test that is being done. E.g. blood, ultra sound, …
    d. For_who (Tests): the person that the test is prescribed for (mother, baby, father, maybe other family members for genetics testing, …)
    e. Birthing Capacity (Birthing Centers): the number of people that can give birth at a birthing center at a time.

Restrictions:

1. Storing the data for Fathers should be optional. If the clients decide to provide the data for the father, they have to minimally provide their name, date of birth, phone number, and profession, the other attributes as shown on the ER model are optional.
2. All the attributes for the mother have to be provided initially, except for the blood type which should be provided/updated later on by the medical people.
3. There is not participation constraint on backup_midwife, prim_midewife, birth_loc while if a couple is approved for the program they should definitely be assigned these. This has to be taken care of in the program such that at the right time for the pregnancy the couple get assigned the values.
4. If the couple Attend at least one info session (attribute of the registered_sess relationship), then their profile will be assessed by the medical staff and if they are Approved (attribute for the couple entity set), then they should be assigned a primary midwife and so on.
5. Info Sessions have a capacity that will need to be checked and changed with every registration/cancelation

Relational Translation:

Entity Sets:

Mothers (HealthID, Name, Address, Phone#, Email, DOB, Blood Type, Profession, App. Due date)

Fathers (Name, Phone#, Address, HealthID, Email, DOB, Blood Type, Profession)

Couples (CoupleID, Approved, HealthID) {HealthID foreign key referencing relation Mothers}

Pregnancies (nth time, Couple ID, status, BName, BAddress, PrimID, BackupID) {Couple ID foreign key referencing relation Couples} {BName, BAddress foreign key referencing relation Facilities} {PrimID, BackupID refrancing relation Midwives}

Babies (TmpID, Name, DOB, BloodType, Gender, nth time, Couple ID) {nth time, Couple ID foreign key referencing relation Pregnancies}

Due Dates (App. Date, nth time, Couple ID, info,) {nth time, Couple ID foreign key referencing relation Pregnancies}

Facilities (Name, Address, Email, Website, Phone#)

Community Clinic (Name, Address) {Name, Address foreign key referencing relation Facilities}

Birthing Center (Name, Address, Capacity) { Name, Address foreign key referencing relation Facilities}

Midwives (ID, Name, Email, Phone#, FAddress, FName) {FAddress, FName foreign key referencing relation Facilities}

Technicians (ID, Phone#, Name)

Tests (Prescription Date, ID, Type, Results, For whom) {ID foreign key referencing relation Appointments}

Sampling Tests (Prescription Date, ID, Sampling Date, Labwork Date) { Prescription Date, ID foreign key referencing relation Tests}

Referral Tests (Prescription Date, ID, Testing Date) { Prescription Date, ID foreign key referencing relation Tests}

Notes (Time Stamp, ID, Text) {ID foreign key referencing relation Appointments}

Appointments (ID, Date&Time, PrimID, BackupID){PrimID, BackupID refrancing relation Midwives}

Info Sessions (Time, ID, Lang, Capacity){ID referencing relation Midwives}

Relationships:

father_of (Name, Phone#, coupleID) {Couple ID foreign key referencing relation Couples}{Name, Phone# foreign key referencing relation Fathers}

registred_info (Time, ID, Couple ID, Attended){Time, ID referencing relation Info Sessions} {Couple ID referencing relation Couples}

technician_of (Prescription Date, TID, ID) {Prescription Date, TID referencing relation Tests} {ID referencing relation Technicians}

Indicate any ER model aspects that your relational model does not capture in this section:
I believe my relational model addressed the general specification needed for the database and matches my ER model. Some stuff that is not represented in the relational model are for example some relationships that we have in the ER model but can be avoided in the relational model (given that I have used Alternative II of creating relational model introduce in the lectures). Furthermore, we have some participation constraints in the ER model that can't be represented in the relational model (since we haven't added NOT NULL)

Are there opportunities to combine relations without introducing redundancy?
I'm not sure it's possible to combine the entities, for example, we could possibly combine the pregnancies and couples with each other, but I'm not sure if that would make the model simpler or reduce redundancy.