Project PART 1

Scenario:

Your team is hired by Dr. Hope Smilow. She is opening her own dental practice “Smilow Dentistry” and needs a database to hold all required data. Her office needs to maintain a list of accepted insurance policy types, a list of standard dental procedures and standard per unit charges for those procedures for each insurance plan. The office will employ various medical professionals such as other dentists, hygienists, dental assistants, receptionists, etc., and will need to maintain standard employee info and licensure details where applicable for billing and medical documentation needs. They need to keep careful track of patients, their complete demographics, emergency contacts, insurance info, medical history, and appointments.

In addition to storing basic demographics info, the system needs to keep track of if and when a patient signed HIPAA form, date of last x-ray, medications, and allergy information and allow to check that the information is still current.

Any time a patient has an appointment, a billing record needs to be generated with the date, doctor / supervising doctor/ medical professional, and procedures performed. They will also keep track of patient payments. Each payment consists of an amount, a date, and a type of payment (cash, check, credit card), and invoices it is applied to. The system should allow to store patients’ payment information and use it when instructed.

Each appointment can involve multiple treatments/procedures. We can assume that the entire invoice is billed under one main supervising dentist’s name. There will be one invoice (bill) per appointment. Not all patients will have insurance coverage, and some may pay out of pocket. For patients who have insurance coverage, insurance payments are applied first, and the patient is billed for the remainder when applicable. We will assume that each patient can have only one insurance plan. We are not going to worry about secondary coverage at this time. Cancelled appointments are not billed. Specific procedures can be performed by certain licensed medical professionals.

1. Conduct additional research and information gathering to identify initial list of entities and attributes for your DB. **Take this phase very seriously.** You will need to research how a small business of this type operates, collects, stores, and uses the information. Get full understanding of how registration, scheduling, and billing processes work. Understand what information needs to be stored for a patient, employee, licensure, appointment, billing, insurance, payments,

medical history, etc. To correctly complete the logical design of your DB, you need to have full clarity of business processes and data requirements. You can use online resources, interview friends and family members, etc. Refer to the project discussion lecture as well. Document all the work completed, and resources used.

2. Provide at least two additional features to supplement the minimum requirements. Each feature must be represented by at least one additional entity with multiple attributes. List entities, attributes, and relationships you plan to use to accommodate those additional features. Give a brief rationale for why these extra features would be interesting/useful to the stakeholders. The proposed extra features should involve the use of at least two additional entities. These entities will need to be used continuously in your design and should not disappear after this portion of the project.
3. Create a list of any other applicable additional requirements and assumptions that you are going to make for your design.
4. Based on the requirements given in the project overview, list the entities to be modeled in this database. For each entity, provide a list of associated attributes. Make sure that your design allows for proper handling of scheduling and billing.
5. Based on the requirements given in the project overview, what are the various relationships between entities? (For example, "A patient schedules an appointment").
6. Give at least four examples of some informal queries/reports that might be useful for this database to support. Informal means that you can state them in a plain English sentence format

without use of any type of prototyping or coding. (For example, "A patient wants to see a list of charges for last year").

7. Provide a full (E)ER diagram for your database. Use Chen's notation as learned in class. Do not use any other types of notations. Make sure you include ALL the entities, attributes, and relationships to fully cover all the minimum requirements as well as the additional features. Remember that (E)ER model cannot have any standalone entities. Ensure that you use a proper notation and include a legend. Each entity must have a proper key. All relationships must have cardinality and participation shown. Be mindful about using weak entities. Check for presence and correct identification of all attributes. Use draw.io for your diagram or another drawing tool if preferred. Hand drawn diagrams will not be accepted.
8. Construct a small sample MS Access DB or MS Excel Spreadsheets with sample data to serve as a first prototype for your DB design. You should have approx. 5-7 records per entity to start with and will need to add more data when the actual DB is going to be constructed. Remember that we are not implemented the DB yet and the only purpose of this task is to show sample data for your main entities.
9. CROSS-CHECK 1: Suppose we want to add a new appointment record to the database. How would we do that given the entities and relationships you have outlined above for a new patient vs for an existing patient? What other information do you need to correctly generate a new appointment? If a patient has multiple allergies and medications, how would your DB store them?

10. CROSS-CHECK 2: Confirm that your model supports ALL listed minimum requirements to correctly process billing and payment handling. If it does not, make changes that allow your design to fully meet all listed requirements.

11. CROSS-CHECK 3: Remember to properly utilize EERD approach and necessary generalizations. Each data item should only be stored in one place in the DB. If entities have common elements, they likely need to be generalized. Remember to use type attributes (flags) to keep track of specialization participation. Do NOT over produce specializations. A specialization should either hold data or participate in a relationship. Create necessary category entities (lists) to standardize your data. Do not mix up application features (such as a portal) and actual entities that hold data.

12. List participating team members and their contributions towards PART 1. Describe how your team has functioned so far and plans to approach this project going forward. List each team member contributions.

13. Submit a professionally written and well formatted report showing **ALL** your work. Do not submit separate files or links. All work must be fully included in one report. When inserting images, make sure that they are in high resolution and easily readable. Failure to do this will result in loss of points. Resubmissions or additions to the submitted report will not be accepted.

14. Save all your work as you will need to use it for the next phase of the project.