Project Part 1

Regina Powers, Ben Borszcz, Lohith Maralla, Bryan Vales

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. They need to keep careful track of patients, their complete demographics, 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 them to check that the information is still current.

Any time a patient has an appointment, a billing record needs to be generated with 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 users 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. Canceled appointments are not billed. Specific procedures can be performed by certain licensed medical professionals.

1. Conduct additional research and information gathering to identify the 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 a 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 complete the correct 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

Resources:

<https://www.colgate.com/en-us/oral-health/dental-visits/whats-included-in-dental-records-and-how-can-you-access-yours#>

<https://www.oralhealthgroup.com/features/dental-record-keeping-101/>

Research:

The first step in creating a database for a dental office is to research and understand how a small business of this type operates and collects, stores, and uses information. Through research, I have determined that a small dental office needs to store information about patients, employees, licensure, appointments, billing, insurance, payments, and medical history.

Patient information includes basic personal information such as name, address, phone number, and employment information. There is also the need to store information about the patient’s health history, including any health issues that could potentially impact their dental care.

Employee information includes any necessary paperwork such as licensure, background checks, and payroll information.

Appointments need to be stored in the database, including the patient’s name, date/time of the appointment, and the services rendered.

Billing information needs to be stored in the database, including the patient’s name, date of service, insurance company, and payment method.

Insurance information needs to be stored in the database, including the patient’s name, insurance company, policy number, and contact information.

Payment information includes the patient’s name, date of payment, amount paid, and payment method.

Medical history needs to be stored in the database for each patient, including past illnesses and treatments.

Overall, it is important to have all of this information stored in a secure database in order for a dental office to run smoothly and efficiently. The database should be tailored to the needs of the business and should be updated regularly to keep track of any changes in the patient’s health or the employee’s licensure.

1. 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.

Our first additional requirement is the ability for patients to review their appointments. We believe this would be useful for stakeholders as they would be able to use this information to see how the dental practice is being received by the people who use it. Additionally, this could be useful for the dental practice as they could use this information to see if there are patterns of certain employees performing better or worse.

Our second additional requirement would be keeping track of the equipment in quality and quantity. This would be useful for the stakeholders since they would be able to see how well equipped the dental practice is. In addition, this is useful for the dental practice since this information would let them know if they need more equipment or if they need to discard the equipment if the quality is not up to par.

1. Create a list of any other applicable additional requirements and assumptions that you are going to make for your design.

A person could be both an employee and a patient as even as an employee, they could get treated at this dental clinic.

A person does not need to be covered by insurance, but for an insurance to exist it must cover at least one person.

Each appointment generated one, and only one billing. However, if there is a cancellation there may be no billing, hence a partial relation. Each billing must be for specifically one appointment in order for it to exist, hence a total relation.

A payment is created by either one Insurance or one patient, it is not possible for multiple insurances to be used, or for multiple patients to create a payment for the same thing.

Procedures can be performed by multiple practitioners as hygienists, assistants, and dentists could all be assigned to a procedure, or just some of them.

The medical history at this specific dental practice can be seen by looking at past procedures of a patient.

Medical history is under patient, as only medical history of patients needs to be kept on file. If an employee is also a patient, then their information will be stored as well.

1. 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.
2. Person

* SSN (Primary Key, INT)
* Phone\_Number (VARCHAR)
* Email (VARCHAR)
* Date\_of\_Birth (DATE)
* Gender (ENUM ('Male', 'Female', 'Other'))
* Name (Compound)
  + - First (VARCHAR)
    - Middle (VARCHAR)
    - Last (VARCHAR)

1. Patient:

* Patient\_ID (Primary Key, INT)
* Signed\_HIPPA (BOOLEAN)
* Last\_XRay (DATE)
* Last\_Information\_Update (DATE)

1. Employee:

* Employee\_ID (Primary Key, INT)
* Date\_Hired (DATE)
* Salary (VARCHAR)

1. Practitioner:

* Practioner\_ID (Primary Key, INT)
* Type (VARCHAR)

1. Receptionist:

* Receptionist\_ID (Primary Key, INT)

1. Insurance:

* Insurance\_ID (Primary Key, INT)
* Insurance\_Plan (VARCHAR)
* Policy\_Number (VARCHAR)
* Comments (VARCHAR)
* Company (VARCHAR)
* Payment\_Schedule (VARCHAR)

1. Appointment:

* Appointment\_ID (Primary Key, INT)
* Date (DATE)
* IsCanceled (BOOLEAN)
* Comments (VARCHAR)

1. Billing:

* Billing\_ID (Primary Key, INT)
* Date (DATE)
* Amount (DECIMAL)

1. Payment:

* Payment\_ID (Primary Key, INT)
* Amount (DECIMAL)
* Date (DATE)

1. Cash
   * Cash\_ID (Primary Key, INT)
2. Check
   * Check\_ID (Primary Key, INT)
   * Memo (VARCHAR)
   * Routing\_Number (INT)
   * Account\_Number (INT)
   * Check\_Number (INT)
3. CreditCard
   * Credit\_Card\_ID (Primary Key, INT)
   * Expiration\_Date (DATE)
   * Number (INT)
   * CVV (INT)
   * Network (VARCHAR)
4. Procedure:

* Procedure\_ID (Primary Key, INT)
* Procedure\_Name (VARCHAR)
* Standard\_Pert\_Unit\_Charges (DECIMAL)
* Comments (VARCHAR)

1. Address:

* Address\_ID (Primary Key, INT)
* Apt\_Number (Optional, VARCHAR)
* Street\_Address (VARCHAR)
* City (VARCHAR)
* State (VARCHAR)
* Zip\_Code (VARCHAR)
* Type (VARCHAR)

1. Licensure:

* Licensure\_ID (Primary Key, INT)
* Date\_Received (DATE)
* Licensure\_Provider (VARCHAR)
* Expiration\_Date (DATE)

1. Review:

* Review\_ID (Primary Key, INT)
* Date (DATE)
* Rating (ENUM)
* Comments (VARCHAR)

1. Medical History
   * Medical\_History\_ID (Primary Key, INT)
2. Medication
   * Medication\_ID (Primary Key, INT)
   * Medication\_Name (VARCHAR)
   * Duration\_Used (VARCHAR)
   * Frequency\_Used (VARCHAR)
3. Allergy
   * Allergy\_ID (Primary Key, INT)
   * Allergy\_Name (VARCHAR)
4. Medical Condition
   * Medical\_Condition\_ID (Primary Key, INT)
   * Date\_Diagnosed (DATE)
   * Condition\_Name (VARCHAR)
   * Description (VARCHAR)
5. Equipment:

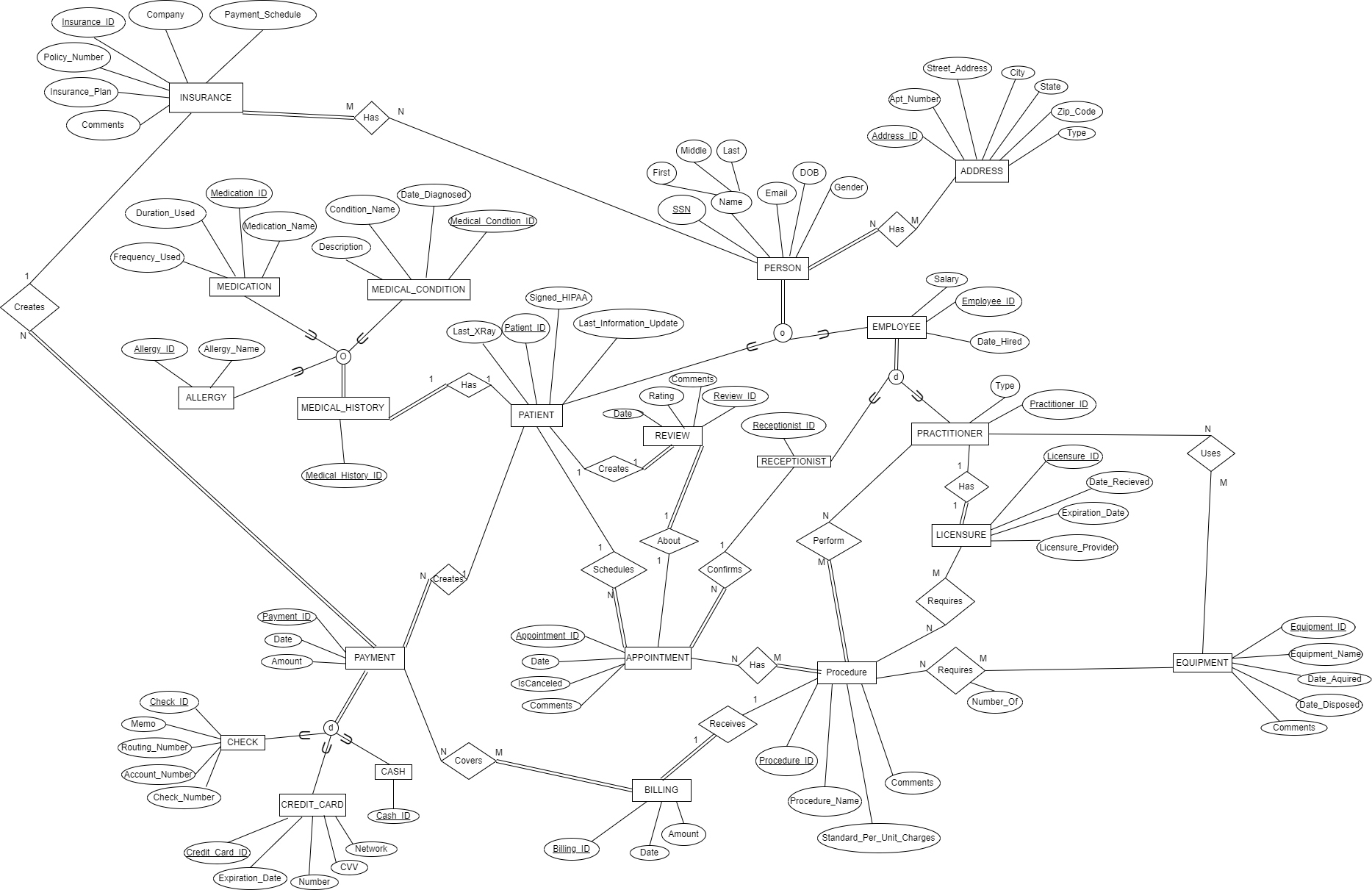
* Equipment\_ID (Primary Key, INT)
* Equipment\_Name (ENUM)
* Date\_Acquired (DATE)
* Date\_Disposed (DATE)
* Comments (VARCHAR)

1. Based on the requirements given in the project overview, what are the various relationships between entities? (For example, “A patient schedules an appointment”).

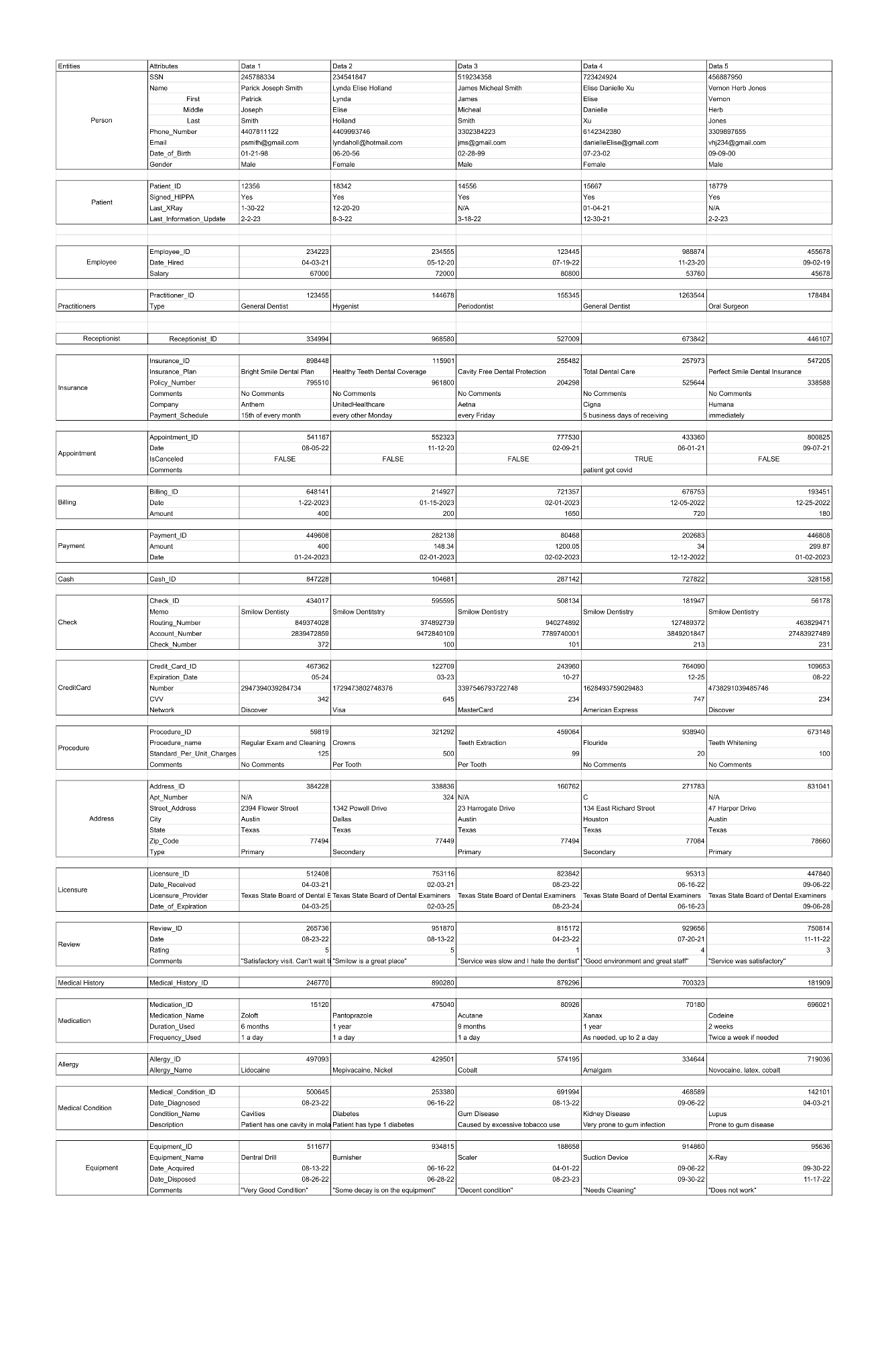
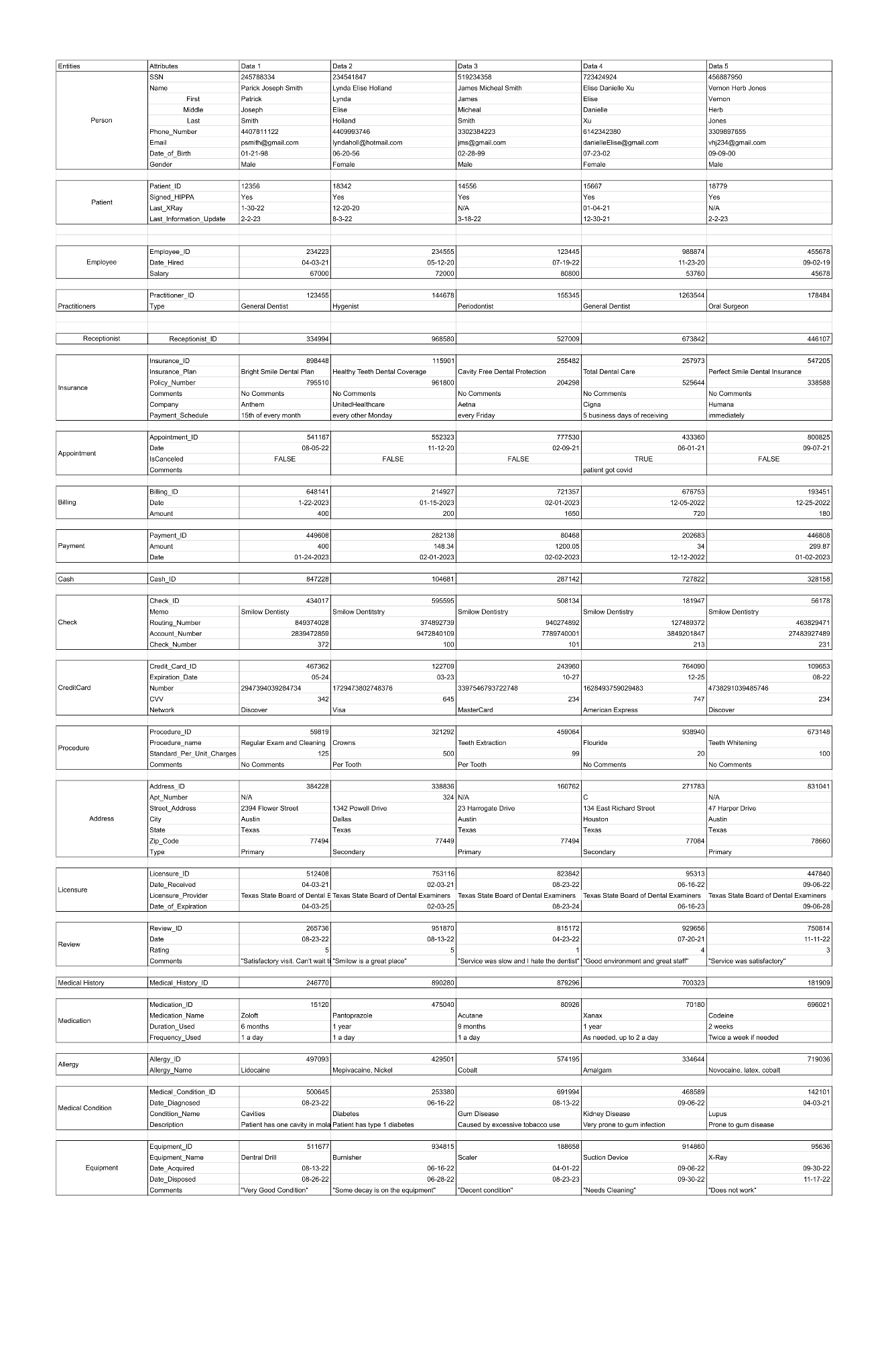
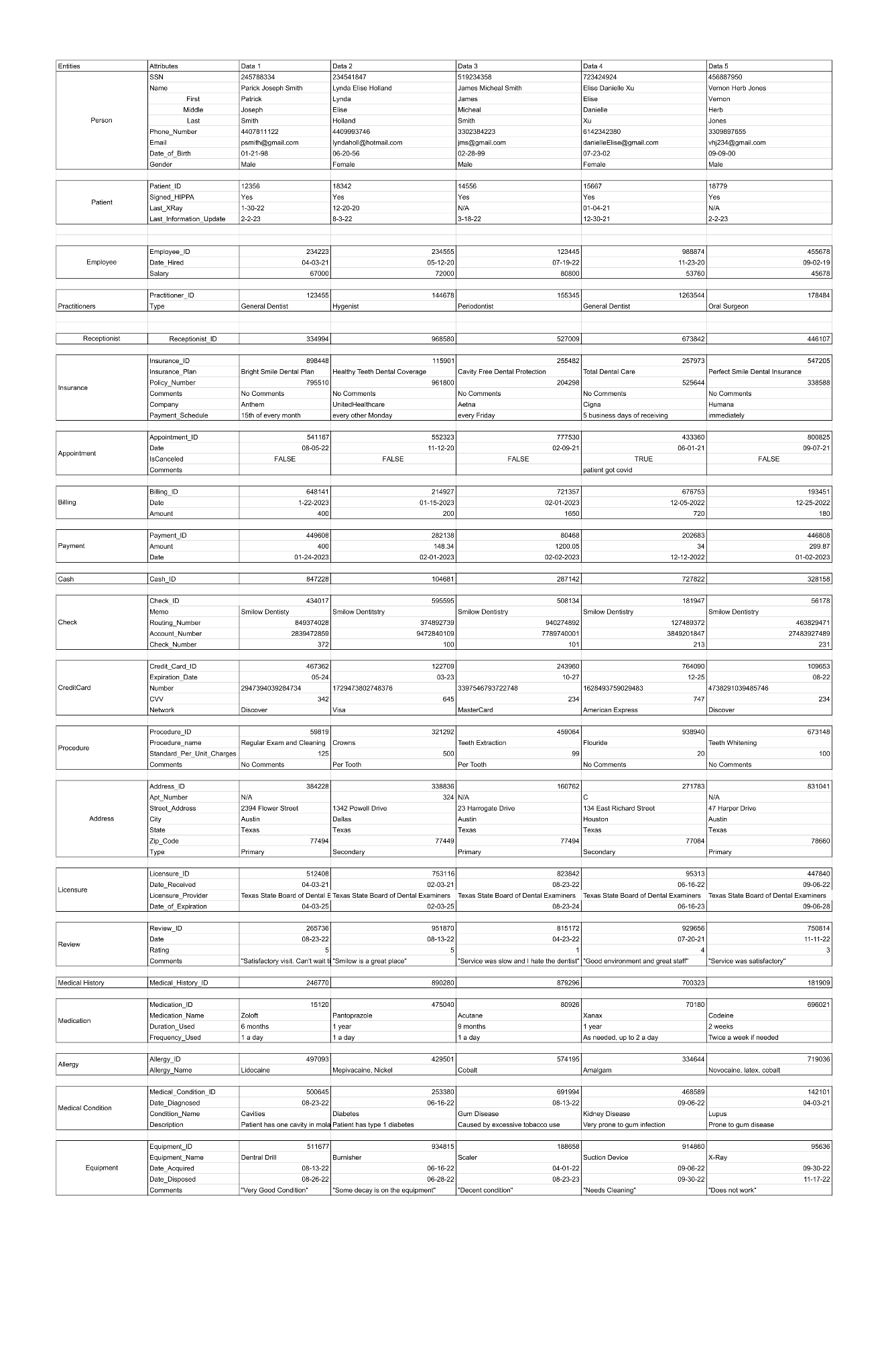
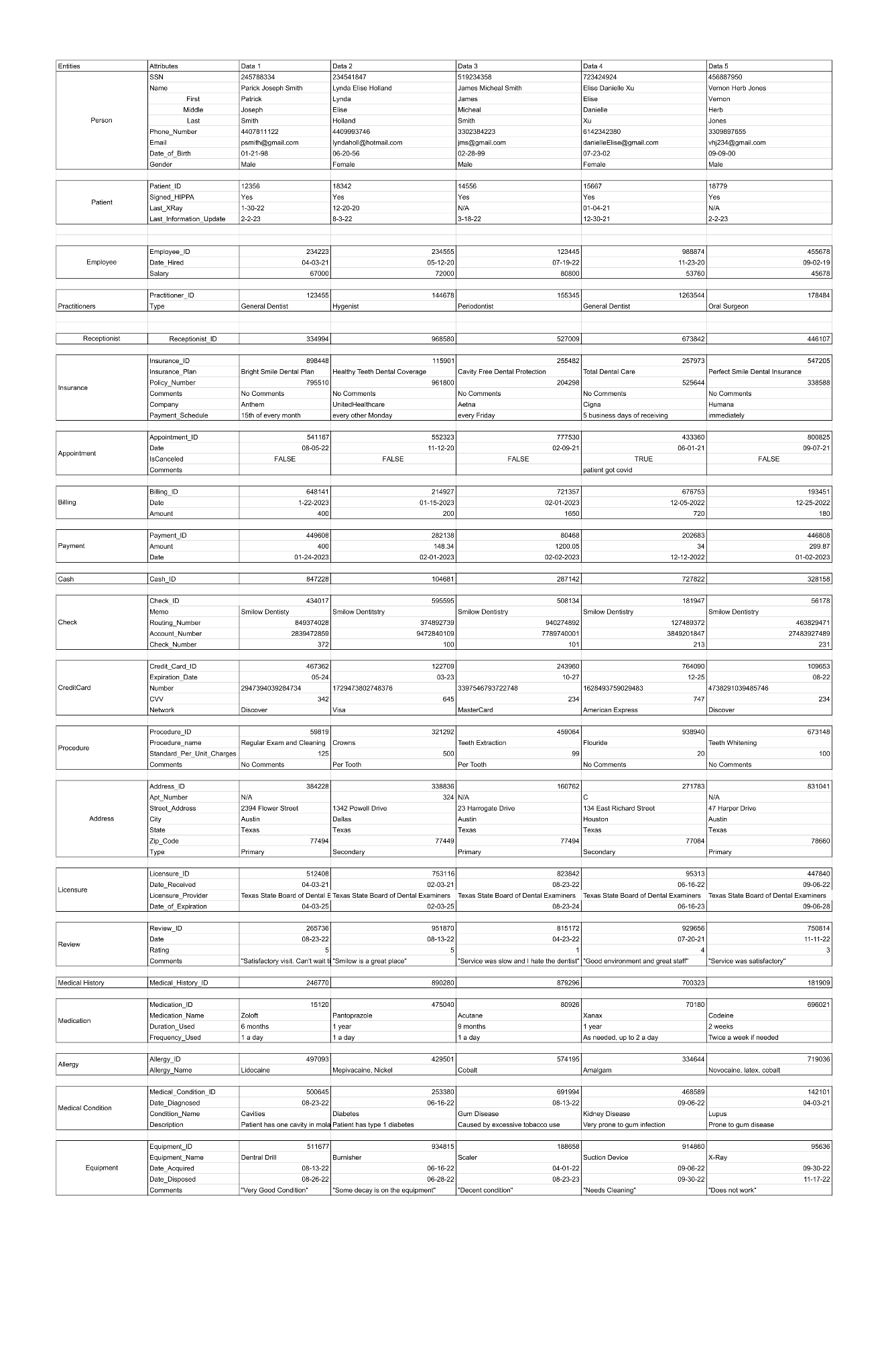
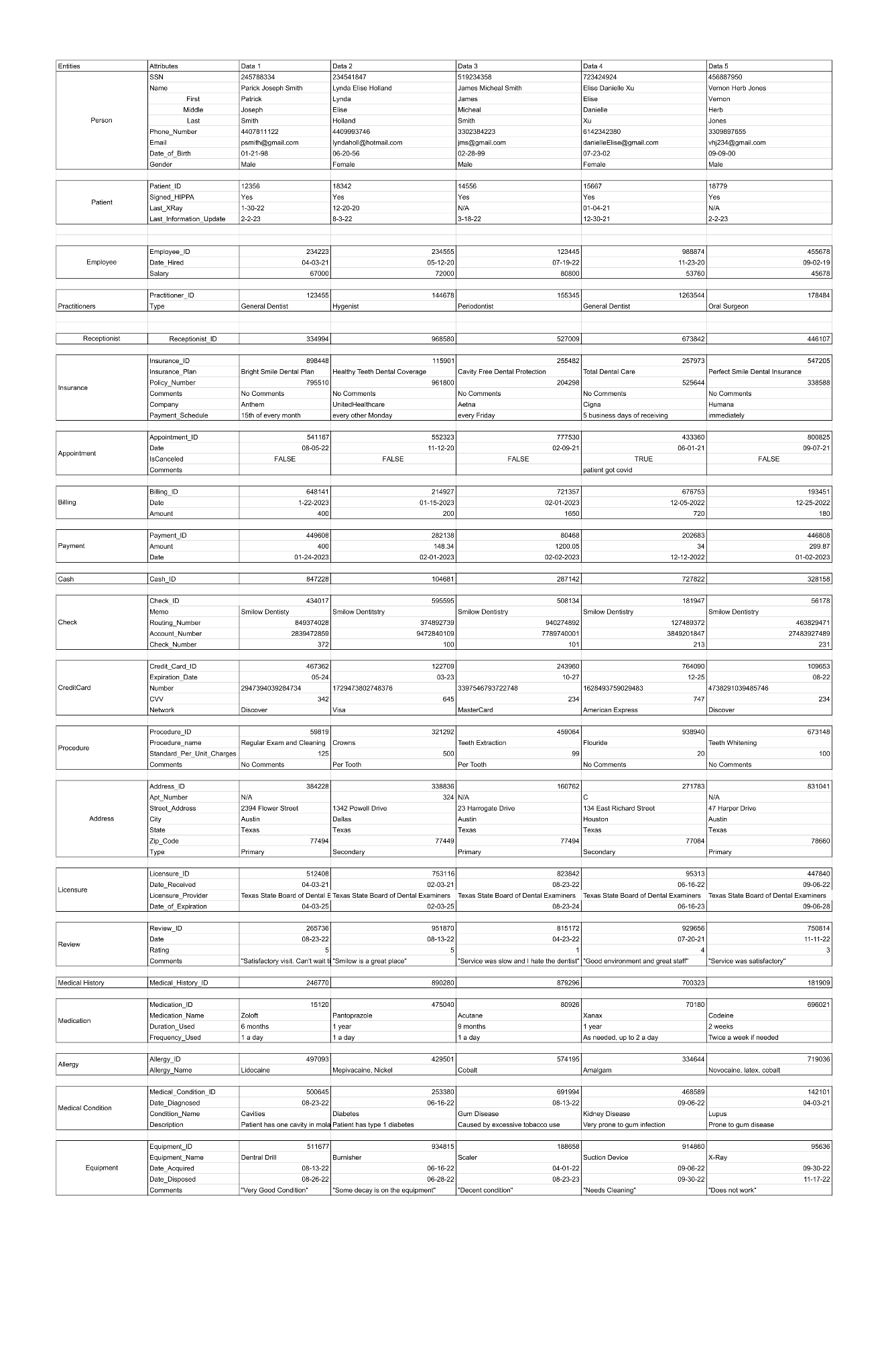
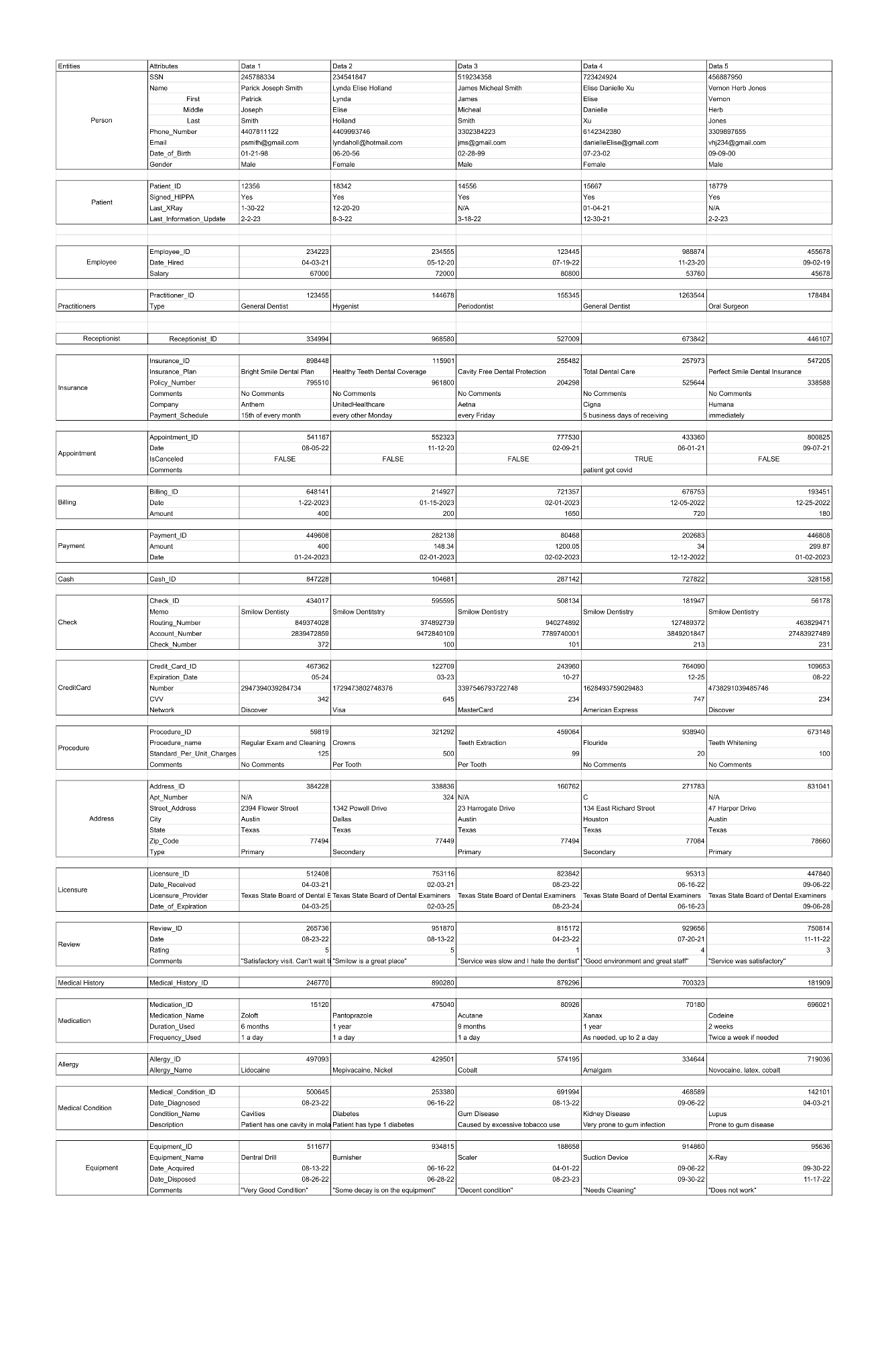
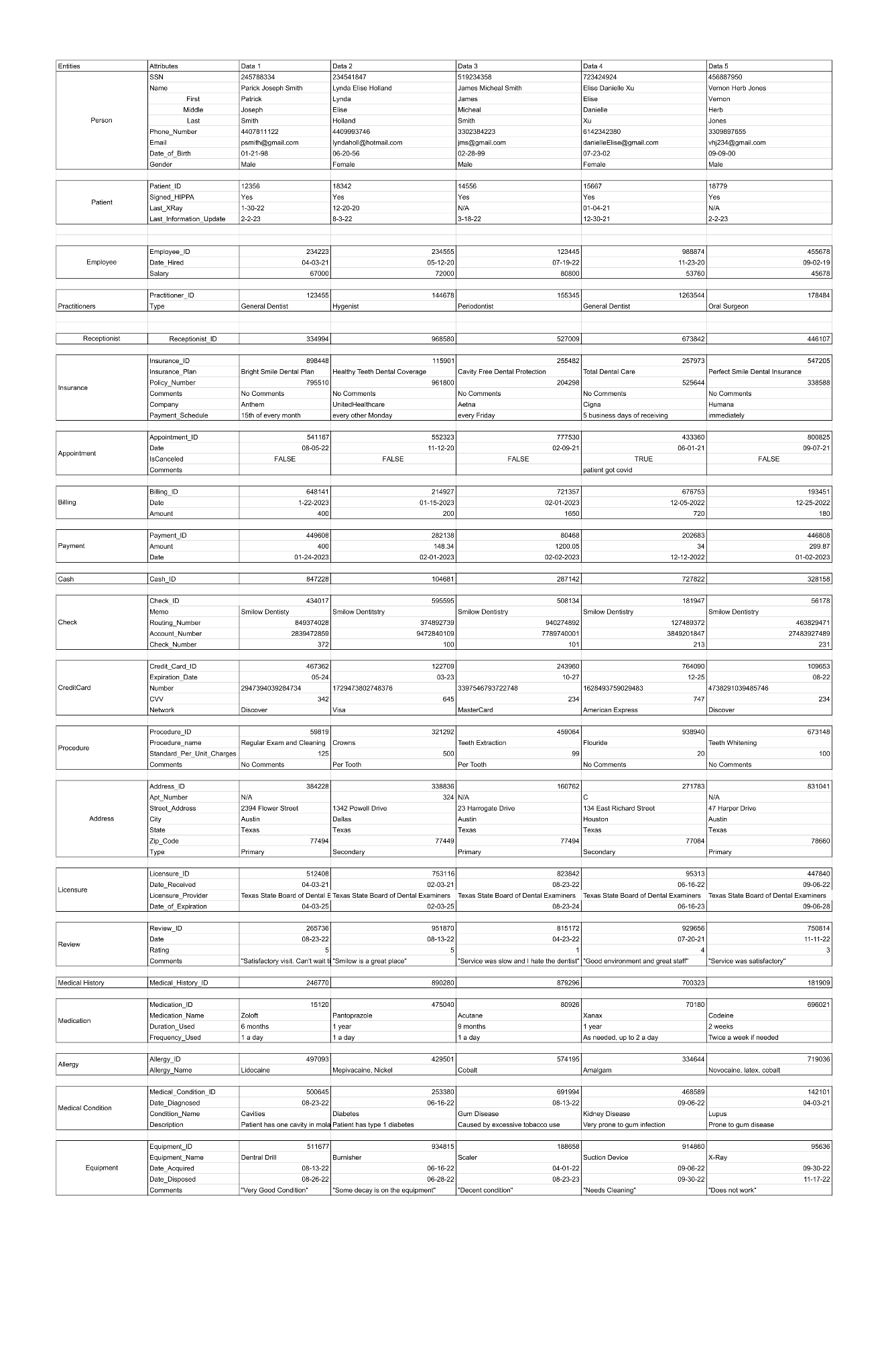
* A person has insurance
* A patient schedules a appointments
* A receptionist confirms appointments
* Each procedure receives billing
* An Insurance creates payment
* A patient creates payment
* A payment covers billing
* Appointments have procedures
* Practitioners perform procedures
* A person has an address
* A practitioner has licensure
* A procedure requires licensure
* A patient creates a review
* A review is about an appointment
* A patient has medical history
* A practitioner uses equipment
* A procedure requires a number of equipment

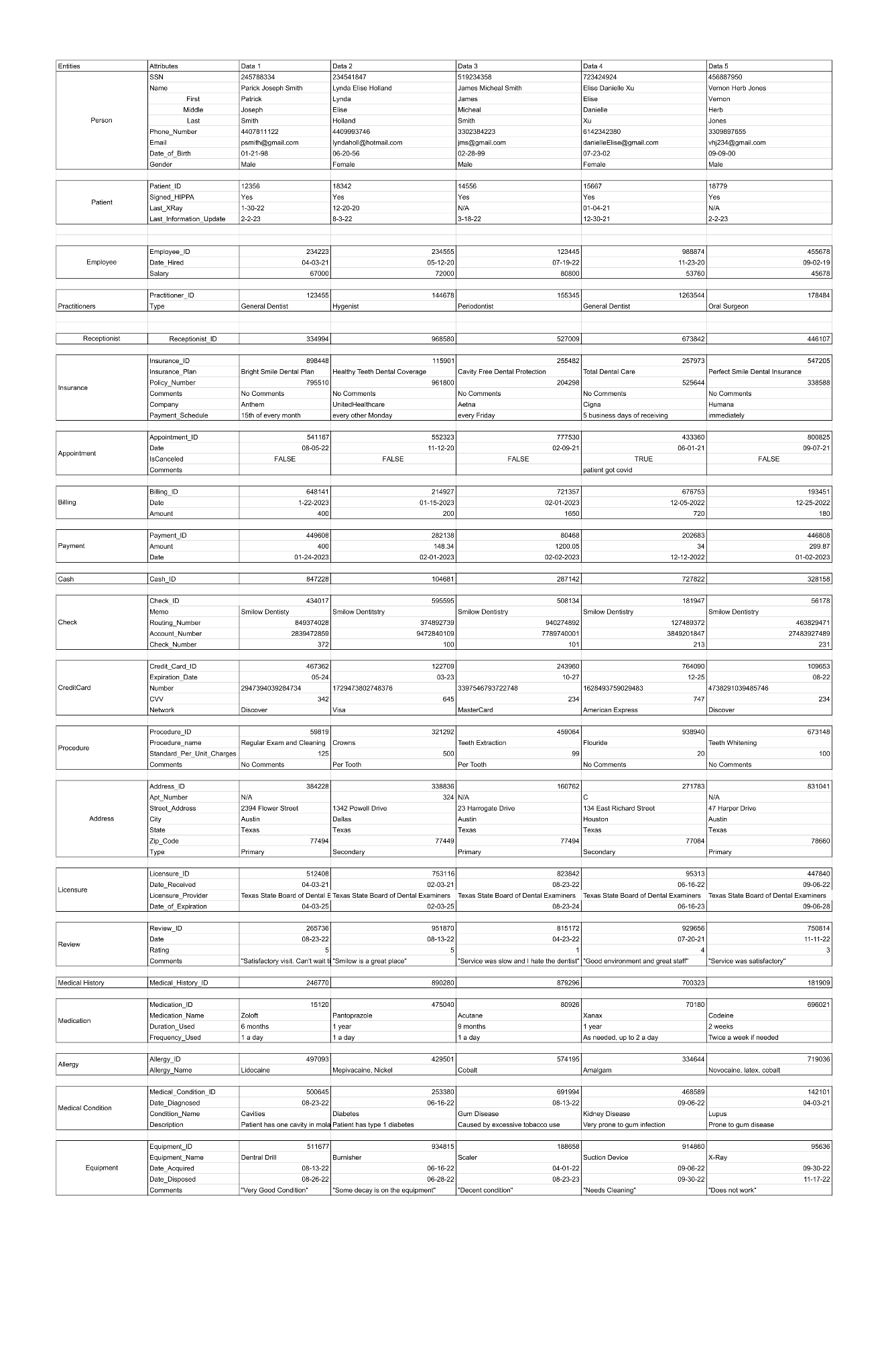
1. Give at least four examples of some informal queries/reports that it might be useful for this database to support. Informal means that you can state them in a plain English sentence format without use of any type of prototyping or coding. (For example, “A patient wants to see a list of charges for last year”).
2. A patient wants to see a list of all past procedures.
3. An employee wants to see how much equipment is in stock.
4. An employee wants to see how many procedures they are assigned to for the next week.
5. A patient wants to see how many procedures they will get in a specific appointment.

1. 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



1. Construct a small sample MS Access DB or MS Excel Spreadsheets with sample data to serve as a first prototype for your DB design. It should match your (E)ERD. 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





1. 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?

To add a new appointment record to the database, we first need to determine if the patient is new or existing. For an existing patient, a new Appointment entity would be created along with its associated entities such as Procedures, and Receptionist confirmation. However, if it's a new patient, we would create new Patient, Medical History, Insurance, and Appointment entities, as well as the relationships between them.

Regarding the patient's medical history, such as allergies and medications, these are stored as part of the Medical History entity, which acts as a generalization for specific medical information. Each Patient has multiple Medical History entities, thus allowing for the storage of multiple allergies and medications.

1. 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 support all listed requirements.

Confirmed

1. 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 overproduce 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 between application features (such as a portal) and actual entities that hold data.

Checked

1. 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.

Regina Powers: Worked on Entity List, ERD, relations, assumptions, and queries

Lohith Maralla: Worked on the Excel Mock DB and Entity relationships

Bryan Vales: Drafted list of relationships between entities, Worked on Excel Mock DB

Ben Borszcz: Worked on Entity List, ERD, and Excel Mock DB

Future Plans:

Our strategies for completing this database project have proven to be successful. Moving forward, we will continue to collaborate and communicate effectively through virtual meetings on Zoom and seek guidance from the professor during their office hours. Our goal is to ensure the project stays on track and meets the requirements in a timely and efficient manner. Additionally, we will continuously review and update our approach as needed to optimize our process and deliver the best possible outcome.

1. Submit a professionally written and well formatted report showing ALL your work.
2. Save all your work as you will need to use it for next phase of the project