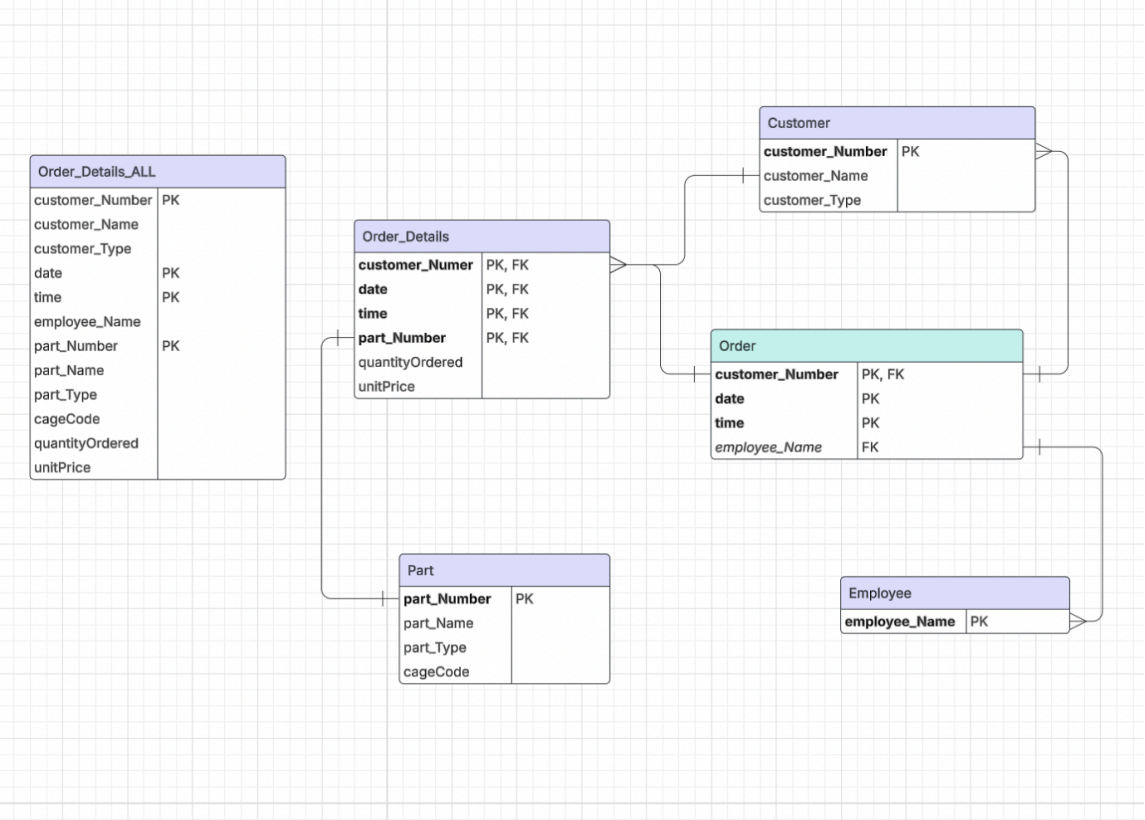
Question 1



a)

- Some assumptions that we need to make are that any employee is allowed to help out a customer and is not restricted to one. Another can be that to determine a single order, it relies on a customer number, date, and time.

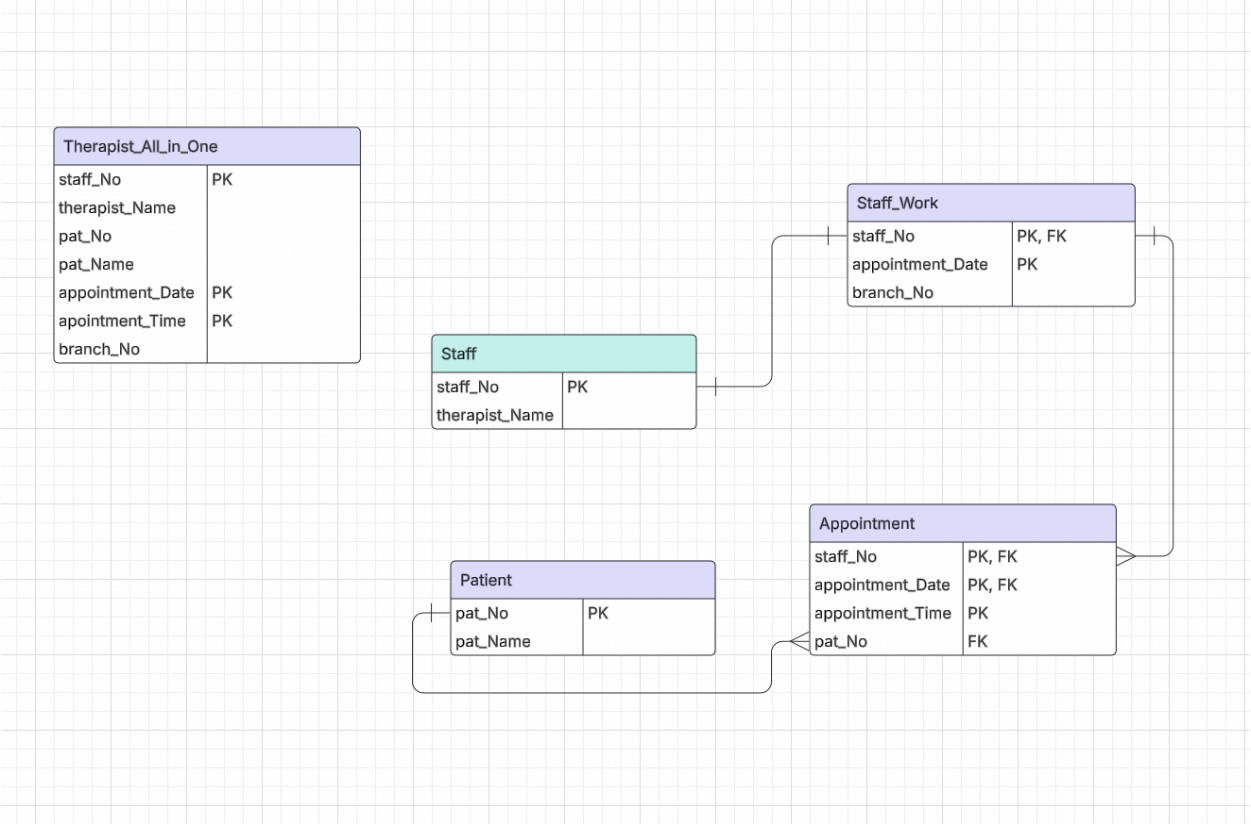
b)

- From the all in one table, we want to split the data to where we have tables that only contain information that is needed and only needed. We can separate the customer number, name, and type to be its own table. We just continue this process throughout the whole file to separate the data. The next table we could look at would be the parts table and see that only part number, name, type, and cage code really matter for this table.

c)

- Primary Keys for Order_Details_All : Customer_Number, Date, Time, Part Number
- 3NF Primary Keys:
    - Customer (PK): customer_Number | Part (PK): part_Number |
    - Employee (PK): employee_Name |
    - Order (PK): customer_Number, date, time (FK): emplyee_Name → Employee customer_Number → Customer |
    - Order_Details (PK): customer_Number, date, time, part_Number (FK):customer_Number, date, time → Order part_Number → Part

Question 2



a)

- Assumptions to make are that a single appointment is to be made up of a staff_No, an appointment date, and an appointment time. We would also have to assume that a staff can only work at one branch per specific day.

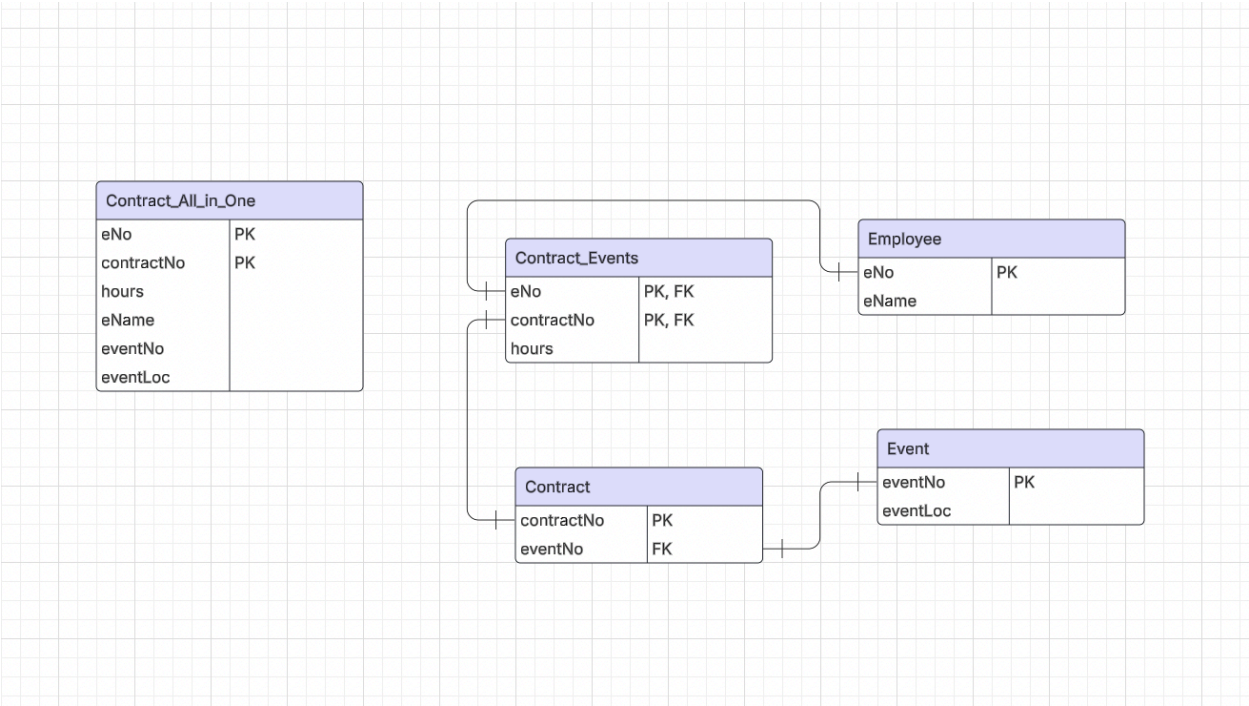
b)

- Looking at the all in one table, we can see that we can break out our dataset into multiple tables. These tables could include a staff table, a patient table, and a table for an appointment. We are trying to make sure that all attributes within a table must all relate to the primary keys involved within the table. We can see that within the staff table, we have the primary key of a staff number and an attribute of the therapist's name. This will also be done towards the patient table where the patient name will only relate to a patient number.

c)

- All in One Table (PK): staff_No, appointment_Date, appointment_Time
- 2NF (PK): Staff (PK): staff_No | Staff_Work (PK): staff_No, appointment_Date | Appointment (PK): staff_No, apointment_Date, appointment_Time
- 3NF (PK)
    - Staff (PK): staff_No
    - Patient (PK): pat_No
    - Staff_Work (PK): staff_No, appointment_Date (FK): staff_No → Staff
    - Appointment (PK): staff_No, appointment_Date, appointment_Time (FK): staff_No, appointment_Date → Staff_Work pat_No → Patient

Question 3



a)

- An assumption we need to make is that for each contract, it can only relate to only 1 event. With each event, each event should take place at only 1 location.

b)

- When trying to normalize the data, we see that we can split a lot of our data up into different tables relating to each other. We see that we can do this with an Employee table, Event table, and a Contract table. Describing these tables, are attributes such as eName which is an attribute to the primary key eNo. However, we see that we need a bridging table to connect information from the employee table and the events table.

c)

- All in One Table (PK): eNo, contactNo
- 2NF (PK): Employee (PK): eNo | Contract (PK): contractNo | Contract_Events (PK): eNo, contractNo
- 3NF (PK):
    - Employee (PK): eNo
    - Event (PK): eventNo
    - Contract (PK): contractNo (FK): eventNo → Event
    - Contract_Events (PK): eNo, contractNo
      (FK): eNo → Employee contractNo → Contract