Assignment – Database Design and Development and Data Management

# Problem 1

Diagram

Description automatically generated

Figure 1: Enhanced Entity Relationship Diagram for the scenario described in Problem 1

# Problem 2

Graphical user interface, diagram

Description automatically generated

Figure 2: Relational Schema representing the database displayed in Problem 2

Firstly, I identified all the entities, their respective attributes, including those which are to be used as primary keys. In the original diagram, the tables are identified by rectangles, and their attributes are represented by circles. Next, I identified the relationships between the entities, which are represented by diamonds. This is shown in the upper diagram in *Figure 2* by the lines linking each foreign key, shown with asterisks (\*) to the appropriate primary key (represented by underlines) in its respective table. I then inserted the primary key from one entity into the entities that uphold relationships with it in the form of foreign keys. The cardinalities in each relationship in this scenario show ‘One-to-Many’ relationships, so the primary key from the ‘One’ side goes into the entitiy on the ‘Many’ side. Lastly, I aggregated the attributes of each entity, and their foreign keys and formatted them into a relational schema, displayed in the lower diagram in *Figure 2*. The upper diagram in *Figure 2* is present to show the process of arriving at the lower diagram in *Figure 2*, which is the final solution for Problem 2.

# Problem 3

1. Answer
2. The table I have identified as the best candidate for horizontal partitioning is the ‘Medical Record’ table. Horizontal partitioning involves splitting the table into chunks that would appear to be vertically stacked, so an implementation of this could look like separate tables for each year of the examination, using the ‘date\_examination’ attribute. This would split what could be millions of records into a few tens of thousands per table. This provides the fastest access due to the reduced number of records returned per query, and the highest security because if a table is compromised, only the records within that single year will be affected. However, the regulator will be able to return all medical records if needed with a union.

SELECT \* WHERE DEEZ\_NUTS == “GOTTEM”

1. The table I have identified as the best candidate for vertical partitioning is the ‘Doctor’ table. Vertical partitioning involves splitting the table into separate tables in which each includes a share of the columns in the original table. An implementation of this would look like a table that holds information of the doctors themselves with the columns ‘d\_id’, ‘d\_name’, and ‘qualification’, and a table that holds information of their roles with the columns ‘h\_id\*’, and ‘salary’. This would split the original table including both static and dynamic data into their respective tables. This provides the fastest access because the regulator would likely be accessing the static data like the doctors’ names more frequently than the dynamic data like their salaries. However, the regulator will be able to return all columns to create the original table by creating a view in which all columns are displayed together.

SELECT \* WHERE DEEZ\_NUTS == “GOTTEM”

# Problem 4

Table

Description automatically generated

Figure 3: Determination of the candidate keys for relation R described in Problem 4

1. The superkeys I have identified are ACD, BCD, CDE, ABCD, ACDE, and BCDE. However, the candidate keys I have identified are **ACD**, **BCD**, and **CDE**.
2. Answer