**Define and describe the Relational Model. (2 pts)**

The relational model is made up of two key parts – relations (tables) and the tuples (rows) that compose them. Relational databases were conceived in the early 70s and formed the basis of transactional database processing, including what we commonly refer to as “SQL” databases today. Relations (tables) are often connected to one another through primary and foreign keys, which make up unique identifiers. These interact in such a way that information (and the tables that make them up) can be One remarkable attribute about the relational model is its persistence – through several different iterations of competitors that seek to usurp its dominance in business, relational databases have continued to be the most widely used database structure for upwards of 40 years.

**Define and describe Document Databases. (2 pts)**

Document databases are a relatively new type of database structure, great in use cases where different documents (records) have few relations to one another. It’s also good when certain types of fields (like diagnosis codes or educational certificates) may have an undefined length. While a relatively new database system, it is robust and has gained a good deal of popularity in business throughout the 2010s.

**Define and describe Graph Databases. (2 pts)**

Graph databases are well-designed databases that focus on many-to-many relationships and the interconnectedness of their elements. The structure of graph databases come in three parts: the subject, the object, and the predicate. The predicate is the “action” that connects the subject and object. For example, the predicate *STUDIED\_AT* may connect an individual (subject) with a university (object). The predicate may also be called an “arc,” and there can be any number of arcs both emanating from and coming into any given entity (tail and head, respectively).

**Discuss the differences between three difference Query Langues covered in this chapter. (2 pts)**

SQL is a declarative coding language – that is to say that it tells the computer the result that it would like to receive, but not necessarily the order in which to perform the operations. This means that SQL (and languages like it) leave it up to the built-in optimizer to determine the order in which the operations are performed.

IMS is the in-house database system from IBM. It is based on the hierarchical model, which is a tree of records-within-records. Most often, the IMS model is represented in JSON files (represented as dictionaries). While one-to-many relationships are well-maintained in IMS, many-to-many relationships are often difficult to represent, unlike graph databases.

CODASYL is a standardized network model that has been standardized by the *Conference On DAta SYstems Languages.* It provides a network model where every item has exactly one parent, where the only way to access a given record was to use its *access path* (i.e. use all of the previous pointers to get there from its genesis).

**Discuss the differences between Linked Lists and Arrays from the Algorithms reading (Bhargava) (2 pts)**

Linked Lists essentially contain pointers to the next item and a pointer from the following item. It is useful especially in exceptionally long lists where there may not be enough contiguous disk space to store an entire array. By contrast (and implied in my prior definition of Linked Lists), an array is a data structure that exists entirely within the same continuous disk space and whose elements can be accessed by indexing. The advantage of an array over a list is that finding an element is easy, whereas in Linked Lists, the insertion or deletion of an element is not computationally complex (it has an O(1) complexity, where as in an array it would be O(n)).