Week 1.

A conceptual model has no **direct**, **visible**, or **usable** counterpart in a database. **Given that fact**, discuss whether it is ever worth having one and thus spending effort on producing one.

Planning is an important procedure in implementing a successful database application. It is also identified as the first step of the universe of discourse. There are many benefits from conceptual model. And It is worth having conceptual model and thus spending effort on producing it. Without conceptual models, it is very difficult to communicate database designs to users, those people who has lack technical skills. Some of key factors bring a lot of conveniences in database design such as visual representation, requirement engineering, high collaborations with other data model and so on.

Other than logic model, conceptual model provides a clear visual representation on entity relationship by using some unique symbols and graphs. Regardless, it has no direct, visible, or useable counterpart in a database. For instance, it is quite straightforward to see the whole entity relationships in ER diagram. However, it takes more effort on understanding the big picture through logic model (i.e. SQL). Due to a clear conceptual model, it saves people a lot of time on explanation database schema.

In order to figure out the requirement of a database, it is always helpful to have such conceptual models. For example, in database system design process, data flow diagram shows flow of data between processes, data stores and external processes. And all entity relationships are given by ER diagrams. Those two diagrams basically describe one goal in database design, which is the deep understanding on requirement of database. Lack of requirement understanding may mislead in user’s data requirement.

Furthermore, most conceptual model can be completely integrated with others. ER diagram can be neatly mapped to a relational schema. However, there is no industry standard notation yet. In consequence, there is no professional rule for conceptual model. Nevertheless, it is still worth building a conceptual model before implementing it.