

Unit 8

Hands-on with Database Design (Reflection)

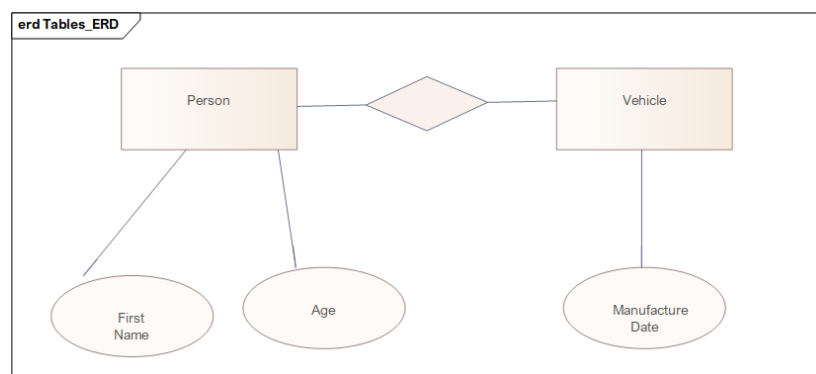
The focus in the week was on an introduction to developing entity-relationship diagrams (ERD) that help drive the structure of a (relational) database. As part of the consideration of developing database structures, there was an engagement in a collaborative discussion to consider alternates to traditional relational SQL structures. The main alternative considered is NoSQL databases.

Graph databases are interesting NoSQL databases, and I have had the fortune to work with them for several years. The most significant difference between graph databases and relational databases is the idea that relationships are first-class citizens. This concept is essential in graph databases. In relational databases, we focus solely on the data sets (the schema) and then consider *how* to traverse each group using relevant joins. However, with graph databases, we think in terms of *relations*. Graph databases, therefore, allow us to model our data using Subject-Verb-Object sentence structures. Not an easy task with relational databases because relational databases do not have a matching paradigm for a Verb.

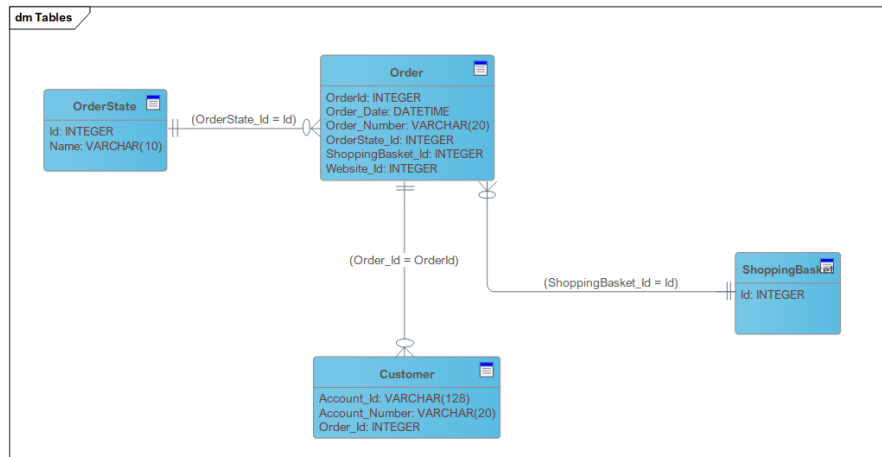
Two database concepts, ACID and BASE, refer to models for how relational (ACID) and NoSQL (BASE) databases provide consistency and reliability.

Modelling ERD

ERD typically follow Chen's notation as in



However, I have hardly ever encountered this notation in my working career. Instead, diagrams similar to below are plentiful and make use of the familiar “crow’s feet” connectors to represent the cardinality of relations between two entities:



I prefer using crow’s feet notation because the attributes of each entity table are grouped into a single visual block. Whereas in Chen’s notation, each attribute becomes a single bubble. Such explosion of attribute bubbles will, in my opinion, quickly create database models that are unwieldy and hard to read! I think back to unit 7, where students were challenged to practice their normalization skills and observed that all submitted diagrams did not utilize Chen’s notation. I believe this is due in part to two reasons: path-of-least resistance and prior exposure to crow’s feet notation.

Moving from a UML class diagram to an ERD is relatively straightforward, except in a specific scenario where a class **composes** another class. Modelling a database for the UML model in unit7 challenged me to consider why I often use many-to-many tables. The standard approach uses the parent table’s primary key in the child table as a foreign key; however, this is not always the case as experienced in this unit. *Please see the artefact for Unit 8’s DB design.*

This unit also gave a task to practice database normalization. Database normalization is a valuable part of designing databases and ensures data duplication is minimized. However, from researching NoSQL databases as part of the collaborative discussion, NoSQL databases are growing in popularity mainly because businesses need to handle large volumes of data far quicker than relational databases allow.