Unit 8

Hands-on with Database Design

(Practical)

In this unit, I developed an Entity-Relationship diagram for the system modelled in unit 7.

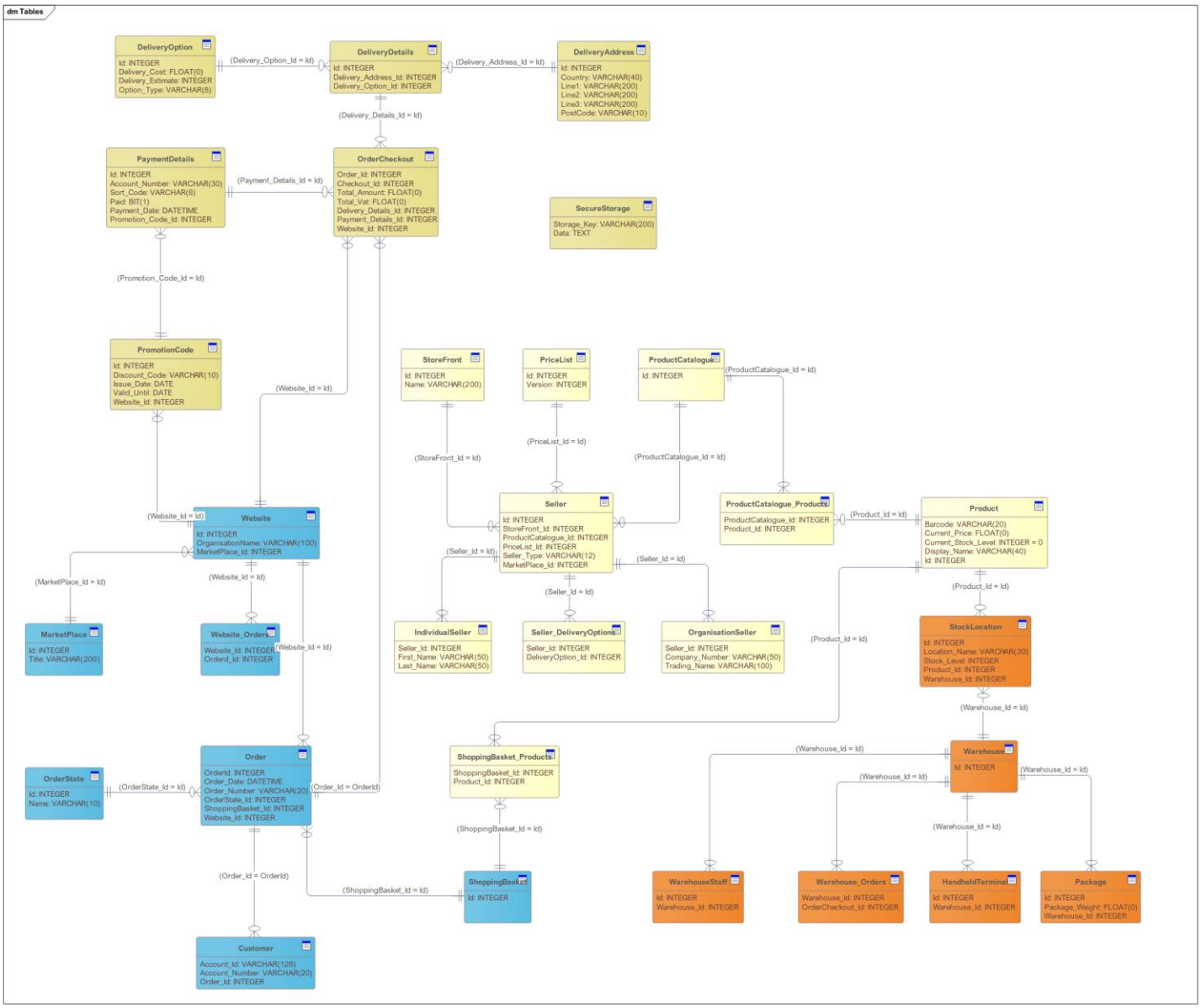
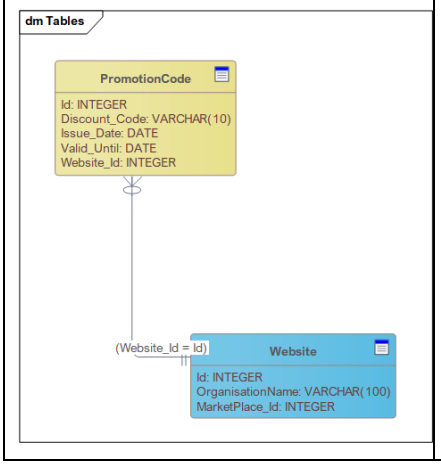
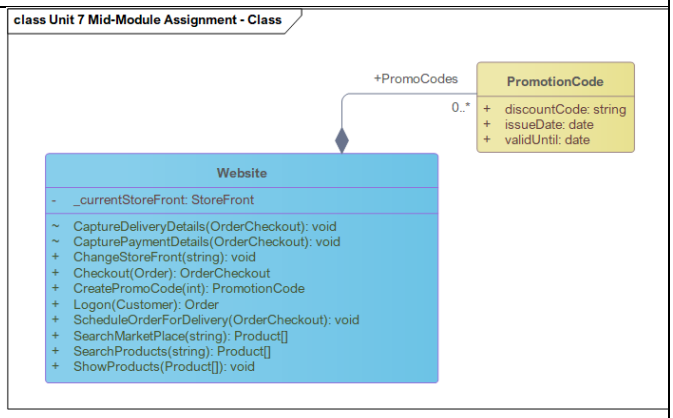


Figure 1 Database model for Unit 7

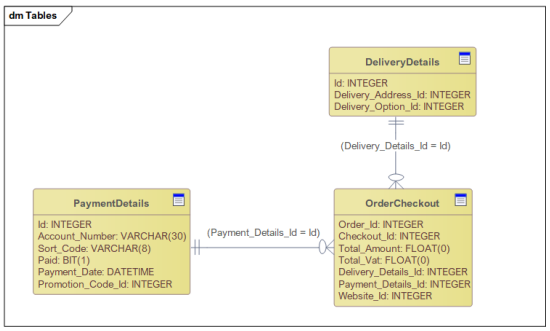
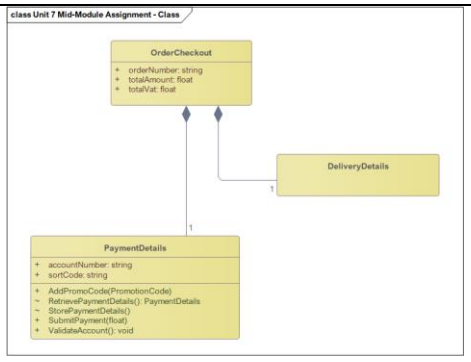
What I found interesting in the database modelling exercise was the need to address the UML **composition** relationships. In the DB model, two modelling solutions are used:

- One-to-zero or more

Database	UML
	

In the DB model, the *primary key* of the ONE (Website) side is used as a *foreign key* in the zero-or-more side (PromotionCode). The foreign key ensures that when a Website record is removed, any linked PromotionCode records will be cascase-deleted as well.

- One-to-one

Database	UML
	

In this scenario, the DB structure is slightly different. The UML model dictates that OrderCheckout has only ONE instance of PaymentDetails and DeliveryDetails. If I used the same approach as “One-to-zero_or_more” above, such a foreign key relationship would *not* correctly adhere to the requirements of the UML model. This is because having the ONE’s primary key column in a child table, implies *zero or more.*

Which is not the case for the One-to-One scenario. So, the modelling solution is to pull the primary key of the referenced/child table into the referencing/parent table and establish a foreign key relationship between the two tables.

For reference, the original UML model from unit 7 is presented below:

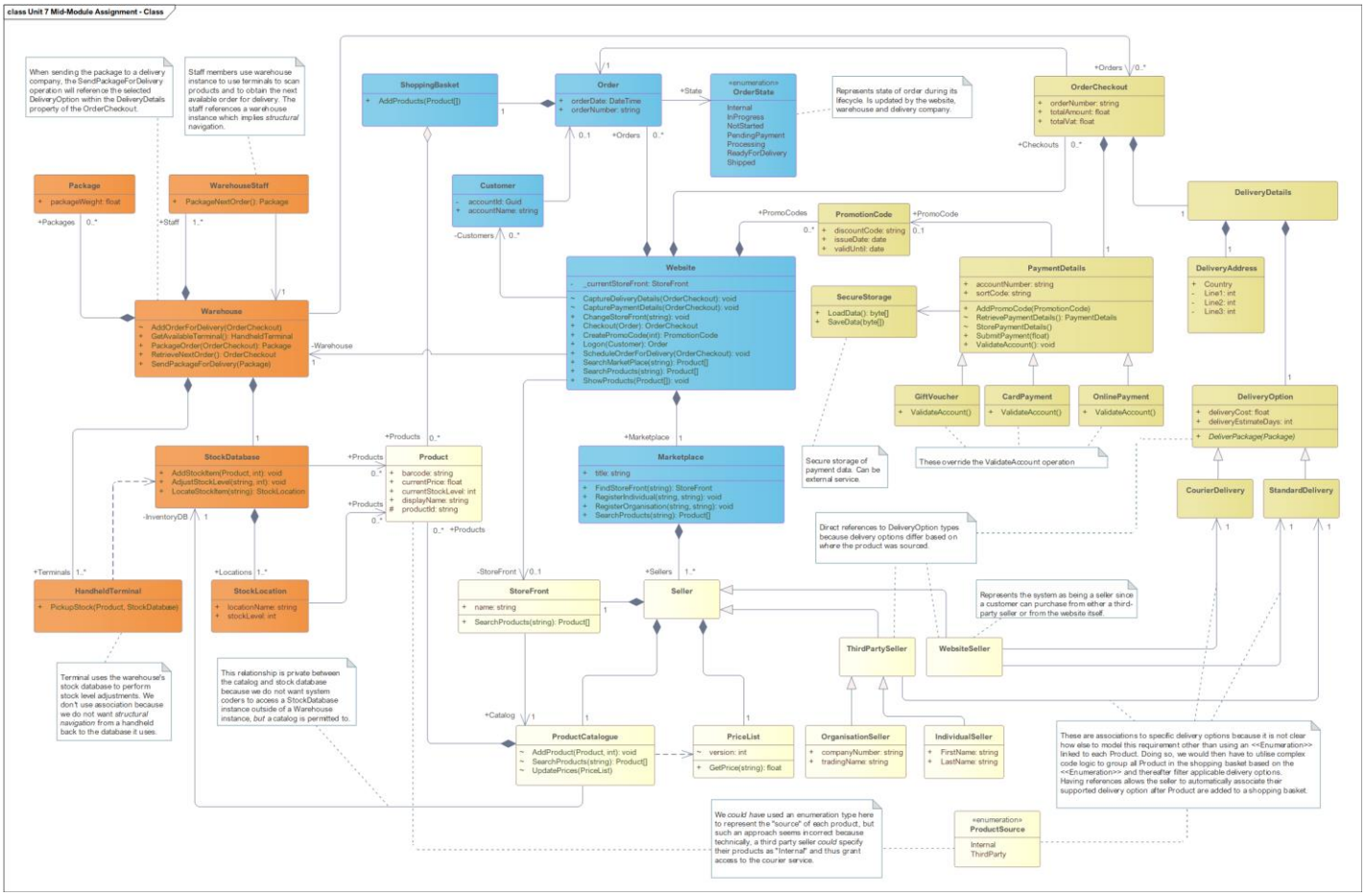


Figure 2 UML class model for unit 7