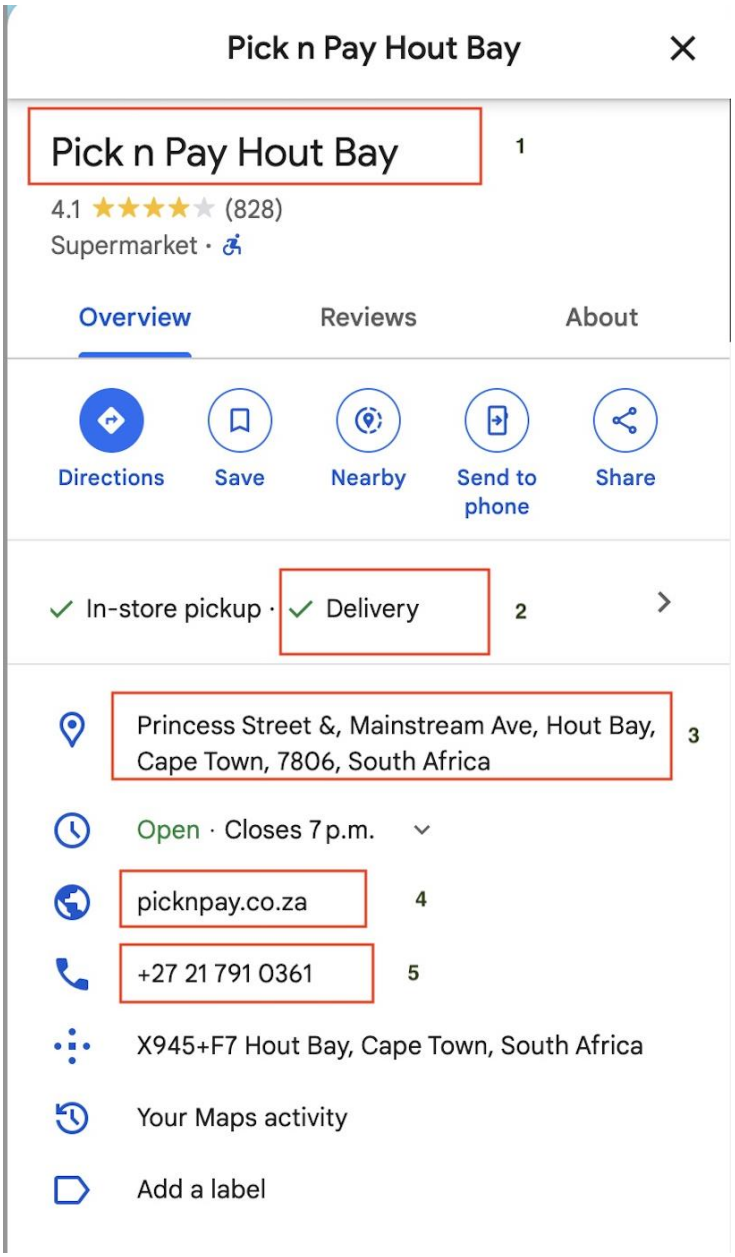
Stores

Column Name	Data Type
StoreId (PK)	Int
StoreName	String
Brand	String
Address	String
Country	String
Latitude	Float
Longitude	Float
PhoneNumber	String
WebsiteAddress	String
DeliveryAvailability	Boolean
OpenSince	Int

Source 1: Google Maps



1. Stores.StoreName
2. Stores.DeliveryAvailability
3. Stores.Address
4. Stores.WebsiteAddress
5. Stores.PhoneNumber

Items & ItemCategory

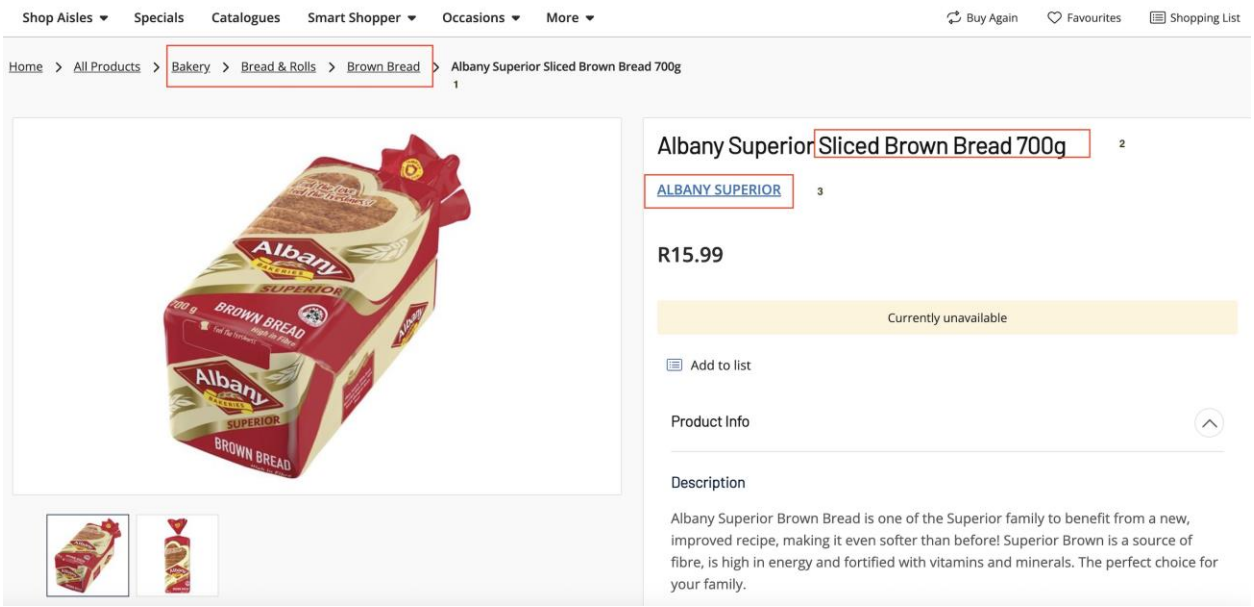
1. Items

Column Name	Data Type
ItemId (PK)	Int
ItemName	String
Brand	String
CategoryId (FK)	Int
Rating	Float

2. ItemCategory

Column Name	Data Type
Id (PK)	Int
Category	String

Source 1: Pick n Pay product page



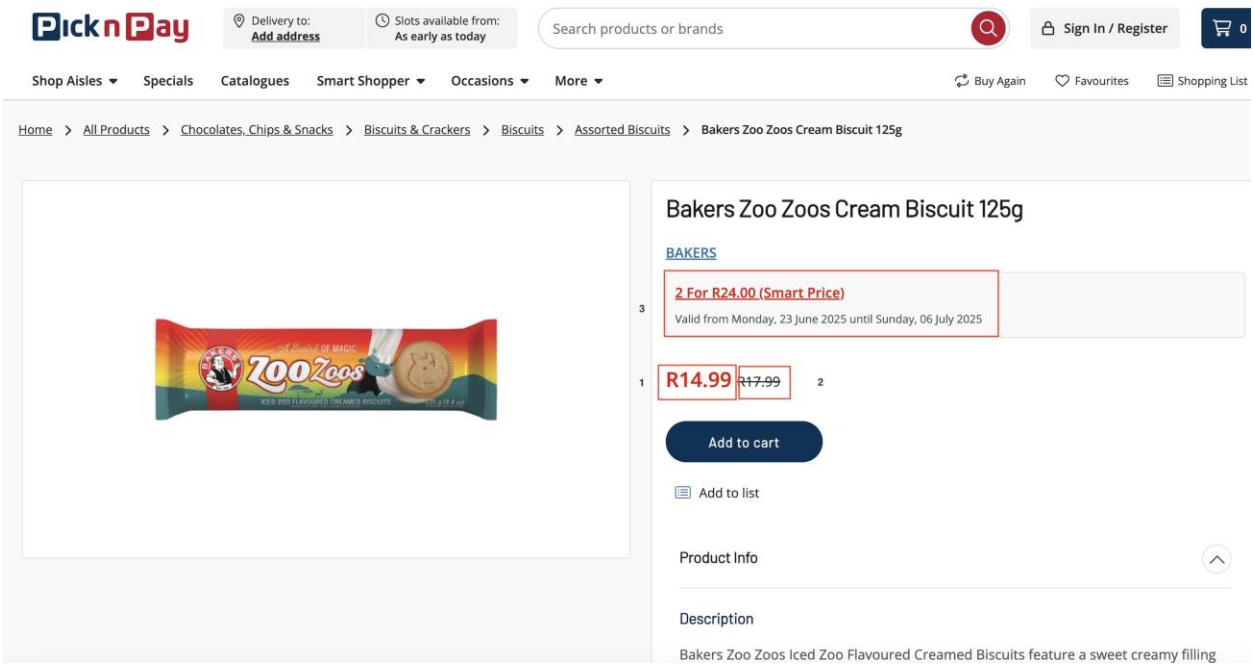
1. ItemCategory.Category
2. Items.ItemName

3. Items.Brand

Prices

Column Name	Data Type
PriceId (PK)	Int
Date	Date
UnitPrice	Float
Currency	String
DiscountDescription	String
IsPromotion	Boolean
ItemId (FK)	Int

Source: Pick n Pay product page



1. Prices.UnitPrice, Prices.Currency
2. Prices.IsPromotion : if there is a strikethrough price, it means there is a price drop.
3. Prices.DiscountDescription : The current database design does not distinguish between standard price drops and promotional offers (such as "buy more for less" deals). To address this, future iterations should introduce a dedicated Promotion table to specifically capture detailed information about promotions. The Prices table would then be reserved for recording the

current and original prices, while all promotion-related details would be managed separately, ensuring more accurate and structured data representation.