A5: Relational Schema, validation and schema refinement

This artifact contains the Relational Schema obtained by mapping from the Conceptual Data Model. The Relational Schema includes the relation schema, attributes, domains, primary keys, foreign keys and other integrity rules: UNIQUE, DEFAULT, NOT NULL, CHECK.

1. Relational Schema

Relation schemas are specified in the compact notation:

Relation reference	**Relation Compact Notation**
R01	user(**id**, name NN, username UK NN, email UK NN, NIF UK, password NN)
R02	address(**id**, name NN, street NN, postalCode NN, idCity → city NN, idUser → user NN)
R03	faq(**id**, question UK NN, answer NN)
R04	purchase(**id**, date NN DF Today, total NN CK total > 0, idUser → user NN, idAddress → address NN, status NN CK status IN Purchase_Status DF 'Processing')
R05	delivery_type(**id**, name UK NN, cost NN CK cost >=0)
R06	product_cart(**idProduct** → product, **idUser** → user, quantity NN CK quantity > 0)
R07	product(**id**, name NN, price NN CK price > 0, quantityAvailable NN CK quantityAvailable >= 0, score NN CK score >= 0 AND score <= 5, idCategory → category NN)
R08	product_purchase(**idProduct** → product, **idPurchase** → purchase, quantity NN CK quantity > 0, price NN CK price > 0)
R09	photo(**id**, path NN, idProduct → product NN)

R10	category(**id**, name UK NN, isNavBarCategory NN DF FALSE)
R11	property(**id**, name UK NN)
R12	category_property(**id**, idCategory → category NN, idProperty → property NN, isRequiredProperty NN DF FALSE)
R13	review(**idUser** → user, **idProduct** → product, score NN CK score >= 0 AND score <= 5, title NN, content NN)
R14	wishlist(**idUser** → user, **idProduct** → product)
R15	admin(**id** → user)
R16	archived_product(**id** → product)
R17	values_list(**id**, idCategoryProperty → CategoryProperty NN, idProduct → Product NN)
R18	value(**id**, name, idValueList → ValuesList NN)
R19	country(**id**, name NN UK)
R20	city(**id**, name NN UK,idCounty → country NN)

Note: UK means UNIQUE KEY, NN means NOT NULL, DF means DEFAULT and CK means CHECK.

2. Domains

The specification of additional domains can also be made in a compact form, using the notation:

Domain Name	**Domain Specification**
Purchase_Status	ENUM (‘Processing’, ‘Shipped’, ‘Delivered’)
Today	DATE DEFAULT CURRENT_DATE

3. Functional Dependencies and schema validation

To validate the Relational Schema obtained from the Conceptual Model, all functional dependencies are identified and the normalization of all relation schemas is accomplished.

Table R01 (user)	
Keys: { id, username, email, NIF }	
Functional Dependencies	
FD0101	{ id } :- { name, username, email, NIF, password }
FD0102	{ username } :- { id, name, email, NIF, password }
FD0103	{ email } :- { id, username, name, NIF, password }
Normal Form	BCNF

Table R02 (address)	
Keys: { id }	
Functional Dependencies	
FD0201	{ id } :- { name, street, postalCode, idCity, idUser }
Normal Form	BCNF

Table R03 (faq)	
Keys: { id, question }	
Functional Dependencies	
FD0301	{ id } :- { question, answer }
FD0302	{ question } :- { id, answer }
Normal Form	BCNF

Table R04 (purchase)	
Keys: { id }	
Functional Dependencies	
FD0401	{ id } :- { date, total, idUser, idAddress, status }
Normal Form	BCNF

Table R05 (delivery_type)	
Keys: { id, name }	
Functional Dependencies	
FD0501	{ id } :- { name, cost }
FD0502	{ name } :- { id, cost }
Normal Form	BCNF

Table R06 (product_cart)	
Keys: { idProduct, idUser }	
Functional Dependencies	
FD0601	{ idProduct, idUser } :- { quantity }
Normal Form	BCNF

Table R07 (product)	
Keys: { id }	
Functional Dependencies	
FD0701	{ id } :- { name, price, quantityAvailable, score, idCategory }
Normal Form	BCNF

Table R08 (product_purchase)	
Keys: { idProduct, idPurchase }	
Functional Dependencies	
FD0801	{ idProduct, idPurchase } :- { quantity, price }
Normal Form	BCNF

Table R09 (photo)	
Keys: { id }	
Functional Dependencies	
FD0901	{ id } :- { path, idProduct }
Normal Form	BCNF

Table R10 (category)	
Keys: { id, name }	
Functional Dependencies	
FD1001	{ id } :- { name, isNavBarCategory }
FD1002	{ name } :- { id, isNavBarCategory }
Normal Form	BCNF

Table R11 (property)	
Keys: { id, name }	
Functional Dependencies	
FD1101	{ id } :- { name }
FD1102	{ name } :- { id }
Normal Form	BCNF

Table R12 (category_property)	
Keys: { id }	
Functional Dependencies	
FD1201	{ id } :- { idCategory, idProperty, isRequiredProperty }
Normal Form	BCNF

Table R13 (review)	
Keys: { idUser, idProduct }	
Functional Dependencies	
FD1301	{ idUser,idProduct} :- { score, title, content }
Normal Form	BCNF

Table R14 (wishlist)	
Keys: { idUser, idProduct }	
Normal Form	BCNF

Table R15 (admin)	
Keys: { id }	
Normal Form	BCNF

Table R16 (archived_product)	
Keys: { id }	
Normal Form	BCNF

Table R17 (values_list)	
Keys: { id }	
Functional Dependencies	
FD1701	{ id } :- { idCategory, idProduct }
Normal Form	BCNF

Table R18 (value)	
Keys: { id }	
Functional Dependencies	
FD1801	{ id } :- { name, idValueList }
Normal Form	BCNF

Table R19 (country)	
Keys: { id, name }	
Functional Dependencies	
FD1901	{ id } :- { name }
FD1902	{ name } :- { id }
Normal Form	BCNF

Table R20 (city)	
Keys: { id, name }	
Functional Dependencies	
FD2001	{ id } :- { name, idCountry }
FD2002	{ name } :- { id, idCountry }
Normal Form	BCNF

As all relations schemas are in the Boyce–Codd Normal Form (BCNF), the relational schema is also in the BCNF and therefore there is no need to be refined using normalisation.

4. SQL Code

Database SQL script link here

(https://github.com/literallysofia/lbaw1761/blob/master/artefacts/a5/create.sql).

GROUP1761, 17/03/2018

- Bárbara Sofia Lopez de Carvalho Ferreira da Silva, up201505628@fe.up.pt (mailto:up201505628@fe.up.pt)
- Carlos Miguel da Silva de Freitas, up201504749@fe.up.pt (mailto:up201504749@fe.up.pt)
- Julieta Pintado Jorge Frade, up201506530@fe.up.pt (mailto:up201506530@fe.up.pt)
- Luís Noites Martins, up201503344@fe.up.pt (mailto:up201503344@fe.up.pt)