A5: Relational Schema, validation and schema refinement

Our project features an information system capable of supporting an online store, which would allow users to buy products from a wide range of categories. In this artefact we present the physical schema of the database, the identification and characterisation of the indexes, the support of data integrity rules with triggers and the definition of the database user-defined functions. We also include the database's workload as well as the complete database creation script.

1 Relational Schema

R01	users(id, username **UK NN**, firstName **NN**, lastName **NN**, email **UK NN**, password **NN**, imageURL **NN**, dateCreated **NN DF** Today, dateModified **NN DF** Today, active **NN**)
R02	client(id → users, cellphoneNumber)
R03	brandManager(id → users)
R04	chatSupport(id → users)
R05	admin(id → brandManager)
R06	product(id, name **UK NN**, quantityInStock **NN DF** 0, dateCreated **NN DF** Today, modelNumber **NN**, weight **NN**, price **NN**, imageURL **UK NN**, bigDescription **NN**, shortDescription **NN**, id_brand → brand **NN**, id_category → productCategory **NN**)
R07	cartProduct(id_product → product, id_client → client, quantity **NN CK** quantity > 0)
R08	productReview(id_product → product **NN**, id_purchase → purchase **NN** , textReview **NN**, rating **NN CK** rating ≤ 5 & rating ≥ 0, reviewDate **NN DF** Today)
R09	productCategory(id , categoryName **NN UK**)
R10	productWishlist(id_product → product, id_client → client)
R11	address(id , addressName **NN**, zipCode **NN**, id_city → city **NN**)
R12	country(id , country **NN UK**)
R13	city(id , city **NN UK**, id_country → country **NN**)
R14	clientAddress(id_client → client , id_address → address)
R15	purchase(id , purchaseDate **NN DF** TODAY, purchaseState **NN**, cost **NN CK** cost >= 0, paymentType **NN**, cardNumber **NN**, cardName **NN**, cardExpirationDate **NN CK** cardExpirationDate > purchaseDate, nif **NN**, id_address → address **NN**, id_client → client **NN**)
R16	purchaseProduct(id_purchase → purchase, id_product → product **NN**, cost **NN CK** cost > 0, quantity **NN CK** quantity > 0)
R17	brand(id, name **NN UK**, storeContact)
R18	brandBrandManager(id_brand → brand **NN**, id_brandManager → brandManager **NN**)
R19	ban(id_user → users, id_admin → admin **NN**, banDate **NN DF** Today)
R20	message(id, message **NN**, dateSent **NN DF** Today, sender **NN CK** sender **IN** Senders, id_chatSupport → chatSupport **NN**, id_client → client **NN**)

Table 1: Relational Schema

2 Domain

Today	DATE DEFAULT CURRENT DATE
Senders	ENUM('Client', 'ChatSupport')

Table 2: Domains

3 Functional Dependencies and Schema Validation

Table R01 (user)	
Keys: {id, username, email}	
Functional Dependencies	
FD0101	{id} → {username, email, firstName, lastName, password, imageURL, dateCreated, dateModified, active}
FD0102	{username} → {id, email, firstName, lastName, password, imageURL, dateCreated, dateModified, active}
FD0103	{email} → {id, username, firstName, lastName, password, imageURL, dateCreated, dateModified, active}
Normal Form	BCNF

Table 3: R01 Dependencies

Table R02 (client)	
Keys: {id}	
Functional Dependencies	
FD0201	{id} → {cellphoneNumber}
Normal Form	BCNF

Table 4: R02 Dependencies

Table R03 (brandManager)	
Keys: {id}	
Functional Dependencies	
none	
Normal Form	BCNF

Table 5: R03 Dependencies

Table R04 (chatSupport)	
Keys: {id}	
Functional Dependencies	
none	
Normal Form	BCNF

Table 6: R04 Dependencies

Table R05 (admin)	
Keys: {id}	
Functional Dependencies	
none	
Normal Form	BCNF

Table 7: R05 Dependencies

Table R06 (product)	
Keys: {id, name, imageURL}	
Functional Dependencies	
FD0601	{id} → {name, imageURL, quantityInStock, dateCreated, modelNumber, weight, price, bigDescription, shortDescription, id_brand, id_productCategory }
FD0602	{name} → {id, imageURL, quantityInStock, dateCreated, modelNumber, weight, price, bigDescription, shortDescription, id_brand, id_productCategory }
FD0603	{imageURL} → {id, name, quantityInStock, dateCreated, modelNumber, weight, price, bigDescription, shortDescription, id_brand, id_productCategory }
Normal Form	BCNF

Table 8: R06 Dependencies

Table R07 (cartProduct)	
Keys: {id_client, id_product}	
Functional Dependencies	
FD0801	{id_client, id_product} → {quantity}
Normal Form	BCNF

Table 9: R07 Dependencies

Table R08 (productReview)	
Keys: {id_product, id_purchase}	
Functional Dependencies	
FD0901	{id_product, id_purchase} → {textReview, rating, reviewDate}
Normal Form	BCNF

Table 10: R08 Dependencies

Table R09 (productCategory)	
Keys: {id, categoryName}	
Functional Dependencies	
FD1001	{id} → {categoryName}
FD1002	{categoryName} → {id}
Normal Form	BCNF

Table 11: R09 Dependencies

Table R10 (productWishlist)	
Keys: {id_product, id_client }	
Functional Dependencies	
none	
Normal Form	BCNF

Table 12: R10 Dependencies

Table R11 (address)	
Keys: {id}	
Functional Dependencies	
FD1301	{id} → {addressName, zipCode, *id_city*}
Normal Form	BCNF

Table 13: R11 Dependencies

Table R12 (country)	
Keys: {id}	
Functional Dependencies	
FD1301	{id} → {country}
FD1302	{country} → {id}
Normal Form	BCNF

Table 14: R12 Dependencies

Table R13 (city)	
Keys: {id}	
Functional Dependencies	
FD1301	{id} → {city, id_city}
FD1302	{city} → {id, id_country}
Normal Form	BCNF

Table 15: R13 Dependencies

Table R14 (clientAddress)	
Keys: {id_client, id_address }	
Functional Dependencies	
none	
Normal Form	BCNF

Table 16: R14 Dependencies

Table R15 (purchase)	
Keys: {id}	
Functional Dependencies	
FD1501	{id} → {purchaseDate, purchaseState, cost, paymentType, cardNumber, cardName, cardExpirationDate, nif, id_address, id_client}
Normal Form	BCNF

Table 17: R15 Dependencies

Table R16 (purchaseProduct)	
Keys: {id_purchase, id_product}	
Functional Dependencies	
FD1601	{id_purchase, id_product} → {cost, quantity}
Normal Form	BCNF

Table 18: R16 Dependencies

Table R17 (brand)	
Keys: {id, name}	
Functional Dependencies	
FD1701	{id} → {name, storeContact}
FD1702	{name} → {id, storeContact}
Normal Form	BCNF

Table 19: R17 Dependencies

Table R18 (brandBrandManager)	
Keys: {id_brand, id_brandManager }	
Functional Dependencies	
none	
Normal Form	BCNF

Table 20: R18 Dependencies

Table R19 (ban)	
Keys: {id_client}	
Functional Dependencies	
FD1901	{id_client} → {id_admin, banDate}
Normal Form	BCNF

Table 21: R19 Dependencies

Table R20 (message)	
Keys: {id}	
Functional Dependencies	
FD2001	{id} → {message, dateSent, sender, id_chatSupport, id_client}
Normal Form	BCNF

Table 22: R20 Dependencies

Because all relations 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 it using normalisation.

4 SQL Code

```
DROP Table IF EXISTS brandBrandManager CASCADE;
DROP TABLE IF EXISTS productreview CASCADE;
DROP TABLE IF EXISTS purchaseproduct CASCADE;
DROP TABLE IF EXISTS purchase CASCADE;
DROP TABLE IF EXISTS clientaddress CASCADE;
DROP TABLE IF EXISTS country CASCADE;
DROP TABLE IF EXISTS city CASCADE;
DROP TABLE IF EXISTS address CASCADE;
DROP TABLE IF EXISTS cartproduct CASCADE;
DROP TABLE IF EXISTS productwishlist CASCADE;
DROP TABLE IF EXISTS product CASCADE;
DROP TABLE IF EXISTS productcategory CASCADE;
DROP TABLE IF EXISTS ban CASCADE;
DROP TABLE IF EXISTS admin CASCADE;
DROP TABLE IF EXISTS brandManager CASCADE;
DROP TABLE IF EXISTS brand CASCADE;
DROP TABLE IF EXISTS message CASCADE;
DROP TABLE IF EXISTS client CASCADE;
DROP TABLE IF EXISTS chatSupport CASCADE;
DROP TABLE IF EXISTS users CASCADE;



CREATE TABLE users (
  id SERIAL PRIMARY KEY,
  firstName TEXT NOT NULL,
  lastName TEXT NOT NULL,
  username TEXT NOT NULL UNIQUE,
  email TEXT NOT NULL UNIQUE,
  password TEXT NOT NULL,
  imageURL TEXT NOT NULL,
  dateCreated TIMESTAMP DEFAULT now() NOT NULL,
  dateModified TIMESTAMP DEFAULT now() NOT NULL,
  active BOOLEAN NOT NULL
);

CREATE TABLE chatSupport (
  id_chatSupport INTEGER PRIMARY KEY REFERENCES users
);

CREATE TABLE client (
  id_client INTEGER PRIMARY KEY REFERENCES users,
  cellphone INTEGER
);

CREATE TABLE message (
  id SERIAL PRIMARY KEY,
  message TEXT NOT NULL,
  dateSent TIMESTAMP DEFAULT now() NOT NULL,
 sender TEXT NOT NULL CHECK((sender = ANY (ARRAY['Client'::text, 'ChatSupport'::text]))),
  id_chatSupport INTEGER NOT NULL REFERENCES chatSupport,
  id_client INTEGER NOT NULL REFERENCES client
);

CREATE TABLE brand (
  id SERIAL PRIMARY KEY,
  name TEXT NOT NULL UNIQUE,
  contact INTEGER NOT NULL
);
```

```
CREATE TABLE brandManager (
  id_brandManager INTEGER PRIMARY KEY REFERENCES users
);

CREATE TABLE admin (
  id INTEGER PRIMARY KEY REFERENCES brandManager
);

CREATE TABLE ban (
  id_user INTEGER PRIMARY KEY REFERENCES users,
  id_admin INTEGER REFERENCES admin NOT NULL,
  banDate TIMESTAMP DEFAULT now() NOT NULL
);

CREATE TABLE productcategory (
  id SERIAL PRIMARY KEY,
  categoryName TEXT NOT NULL UNIQUE
);

CREATE TABLE product (
  id SERIAL PRIMARY KEY,
  name TEXT UNIQUE NOT NULL,
  quantityInStock INTEGER NOT NULL DEFAULT 0,
  dateCreated TIMESTAMP DEFAULT now() NOT NULL,
  modelNumber INTEGER NOT NULL,
  weight DECIMAL NOT NULL,
  price MONEY NOT NULL,
  imageURL TEXT NOT NULL UNIQUE,
  bigDescription TEXT NOT NULL,
  shortDescription TEXT NOT NULL,
  id_brand INTEGER NOT NULL REFERENCES brand,
  id_category INTEGER NOT NULL REFERENCES productcategory
);

CREATE TABLE productwishlist (
  id_product INTEGER REFERENCES product,
  id_client INTEGER REFERENCES client,
  PRIMARY KEY(id_product, id_client)
);

CREATE TABLE cartproduct (
  id_client INTEGER REFERENCES client,
  id_product INTEGER REFERENCES product,
  quantity INTEGER NOT NULL CHECK (quantity > 0),
  PRIMARY KEY(id_client, id_product)
);

CREATE TABLE country (
  id SERIAL PRIMARY KEY,
  country TEXT NOT NULL UNIQUE
);

CREATE TABLE city (
  id SERIAL PRIMARY KEY,
  city TEXT NOT NULL,
  id_country INTEGER NOT NULL REFERENCES country
);

CREATE TABLE address (
  id SERIAL PRIMARY KEY,
  address TEXT NOT NULL,
  zipcode TEXT NOT NULL,
```

```
  id_city INTEGER NOT NULL REFERENCES city
);

CREATE TABLE clientaddress (
  id_client INTEGER NOT NULL REFERENCES client,
  id_address INTEGER NOT NULL REFERENCES address,
  PRIMARY KEY(id_client, id_address)
);

CREATE TABLE purchase (
  id SERIAL PRIMARY KEY,
  id_client INTEGER REFERENCES client NOT NULL,
  id_address INTEGER REFERENCES address NOT NULL,
  purchaseDate TIMESTAMP DEFAULT now() NOT NULL,
  purchaseState TEXT NOT NULL,
  cost MONEY NOT NULL CHECK (cost > CAST ( 0 AS MONEY )),
  paymentType TEXT NOT NULL,
  cardNumber TEXT NOT NULL,
  cardName TEXT NOT NULL,
  cardExpirationDate TIMESTAMP NOT NULL,
  nif INTEGER NOT NULL,
  CHECK (cardExpirationDate > purchaseDate)
);

CREATE TABLE purchaseproduct (
  id_purchase INTEGER REFERENCES purchase,
  id_product INTEGER NOT NULL REFERENCES product,
  quantity INTEGER NOT NULL CHECK (quantity > 0),
  cost MONEY NOT NULL CHECK (cost > CAST ( 0 AS MONEY )),
  PRIMARY KEY(id_purchase, id_product)
);

CREATE TABLE productreview (
  id_product INTEGER NOT NULL REFERENCES product,
  id_purchase INTEGER NOT NULL REFERENCES purchase,
  reviewDate TIMESTAMP DEFAULT now() NOT NULL,
  textReview TEXT NOT NULL,
  rating INTEGER NOT NULL CHECK (((rating >= 0) AND (rating <= 5))),
  PRIMARY KEY(id_product, id_purchase)
);

CREATE TABLE brandBrandManager (
  idBrand INTEGER NOT NULL REFERENCES brand,
  idBrandManager INTEGER NOT NULL REFERENCES brandManager,
  PRIMARY KEY(idBrand, idBrandManager)
);
```

Link: https://beatrizhenriquesmartins.github.io/lbaw1736/src/sql/createDB.sql

Revision history
Changes made to the first submission:

- Separação de *addresses* em *countries*, *cities* and *addresses*
- Removed *client* reference in *productreview*
- Removed *chat* class and created references of *chatSupport* and *client* in *message*
- Removed *cart* and reformed *cartproduct* in *Relational Schema*
- Removed *wishlist* and reformed *productwishlist* in *Relational Schema*
- Removed *nif* in *client* and added in *purchase*
- Changed all *1...* relations to NOT NULL
- Added Functional Dependency in *productreview*
- Added the justification for the Normal Form of the relational schema
- Created domain for *message* Table
- Added github pages links

GROUP1736, 18/03/2018

- Group member 1 Beatriz de Henriques Martins, up201502858@fe.up.pt
- Group member 2 Francisco Tuna Andrade, up201503481@fe.up.pt
- Group member 3 Luís Miguel Santos Monteiro Saraiva, up201404302@fe.up.pt
- Group member 4 Ricardo Filipe Amaro Saleiro Abreu, up201304450@fe.up.pt