

A6: Integrity constraints. Indexes, triggers, user functions and database populated with data

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 its Relational Schema which was obtained by mapping its UML Conceptual Data Model.

1 Database Workload

1.1 Estimate of Tuples

Relation Reference	Relation Name	Order of Magnitude	Estimated growth
R01	user	tens	units per day
R02	client	tens	units per day
R03	brandManager	units	units per month
R04	chatSupport	units	units per month
R05	admin	units	no growth
R06	product	hundreds	units per day
R07	cartProduct	hundreds	dozens per day
R08	productReview	tens	units per day
R09	productCategory	tens	no growth
R10	productWishlist	tens	units per day
R11	address	tens	units per day
R12	country	tens	no growth
R13	city	tens	no growth
R14	clientAddress	tens	units per day
R15	purchase	tens	units per day
R16	purchaseProduct	tens	units per day
R17	brand	tens	units per month
R18	brandBrandManager	tens	units per month
R19	ban	units	units per year
R20	message	thousands	dozens per day

1.2 Most frequent queries

Query Reference	SELECT01
Query description	List of Products
Query frequency	hundreds per day
<pre>CREATE VIEW "ListProducts" AS SELECT p.id AS "ID",p.name AS "Name", p.quantityInStock AS "Quantity In Stock", p.dateCreated AS "Date Created", p.modelNumber AS "Model Number", p.weight AS "Weight", p.price AS "Price", p.imageURL AS "Image URL", p.bigDescription AS "Big Description", p.shortDescription AS "Short Description", (array_agg(pc.categoryName ORDER BY p.id DESC))[1] AS "Category", (array_agg(b.name ORDER BY p.id DESC))[1] AS "Brand", AVG(pr.rating) AS "Rating" FROM product p, productcategory pc, brand b, productreview pr WHERE p.id_brand = b.id AND p.id_category = pc.id AND pr.id_product = p.id GROUP BY p.id;</pre>	

Query Reference	SELECT02
Query description	View Product
Query frequency	hundreds per day
<pre>CREATE VIEW "ViewProduct" AS SELECT p.id AS "ID",p.name AS "Name", p.quantityInStock AS "Quantity In Stock", p.dateCreated AS "Date Created", p.modelNumber AS "Model Number", p.weight AS "Weight", p.price AS "Price", p.imageURL AS "Image URL", p.bigDescription AS "Big Description", p.shortDescription AS "Short Description", (array_agg(pc.categoryName ORDER BY p.id DESC))[1] AS "Category", (array_agg(b.name ORDER BY p.id DESC))[1] AS "Brand", AVG(pr.rating) AS "Rating" FROM product p, productcategory pc, brand b, productreview pr WHERE p.id_brand = b.id AND p.id_category = pc.id AND pr.id_product = p.id AND p.id = \$productID GROUP BY p.id;</pre>	

Query Reference	SELECT03
Query description	List Products By Category
Query frequency	hundreds per day
<pre>CREATE VIEW "ListProductsByCategory" AS SELECT p.id AS "ID",p.name AS "Name", p.quantityInStock AS "Quantity In Stock", p.dateCreated AS "Date Created", p.modelNumber AS "Model Number", p.weight AS "Weight", p.price AS "Price", p.imageURL AS "Image URL", p.bigDescription AS "Big Description", p.shortDescription AS "Short Description", (array_agg(pc.categoryName ORDER BY p.id DESC))[1] AS "Category", (array_agg(b.name ORDER BY p.id DESC))[1] AS "Brand", AVG(pr.rating) AS "Rating" FROM product p, productcategory pc, brand b, productreview pr WHERE p.id_brand = b.id AND p.id_category = pc.id AND pr.id_product = p.id AND pc.categoryName LIKE '%Fashion%' GROUP BY p.id;</pre>	

Query Reference	SELECT04
Query description	Reviews of Purchase
Query frequency	hundreds per day
<pre> CREATE VIEW "ReviewsByPurchaseId" AS SELECT * FROM productreview WHERE id_purchase = \$purchaseId; </pre>	

Query Reference	SELECT05
Query description	User's Profile
Query frequency	dozens per day
<pre> CREATE VIEW "Profile" AS SELECT * FROM users WHERE id = \$userId; </pre>	

Query Reference	SELECT06
Query description	Client's Wishlist
Query frequency	dozens per month
<pre> CREATE VIEW "WishlistByClientId" AS SELECT product.name, product.price, product.imageURL, client.id_client FROM product, client, productwishlist WHERE client.id_client = productwishlist.id_client AND product.id = productwishlist.id_product AND client.id_client = \$clientId; </pre>	

Query Reference	SELECT07
Query description	Products in Purchase
Query frequency	dozens per day
<pre> CREATE VIEW "ProductsInPurchase" AS SELECT product.name, product.price, product.imageURL, purchase.id_client, purchaseproduct.quantity, purchaseproduct.cost FROM purchase, product, purchaseproduct WHERE purchase.id = purchaseproduct.id_purchase AND product.id = purchaseproduct.id_product AND purchase.id_client = \$id_client; </pre>	

Query Reference	SELECT08
Query description	Products in Cart
Query frequency	dozens per day
<pre> CREATE VIEW "ProductsInCart" AS SELECT product.name, product.price, product.imageUrl, client.id_client FROM product, client, cartproduct WHERE client.id_client = cartproduct.id_client AND product.id = cartproduct.id_product AND client.id_client = \$clientId; </pre>	

Query Reference	SELECT09
Query description	Messages of a Client
Query frequency	hundreds per month
<pre> CREATE VIEW "ClientMessages" AS SELECT * FROM message WHERE id_client = \$id_client; </pre>	

1.3 Most frequent modifications

Query Reference	UPDATE01
Query description	Update user information
Query frequency	hundreds per month
<pre>UPDATE users SET firstName = \$firstName, lastName = \$lastName, username = \$username, email = \$email, password = \$password, imageURL = \$imageURL, dateModified = now() WHERE id = \$id;</pre>	

Query Reference	UPDATE02
Query description	Update product information
Query frequency	hundreds per month
<pre>UPDATE product SET quantityInStock=\$quantityInStock, modelNumber=\$modelNumber, weight=\$weight, price=\$price, imageURL=\$imageURL, bigDescription=\$bigDescription, shortDescription=\$shortDescription, id_category=\$id_category WHERE id_brand=\$id_brand AND name=\$name;</pre>	

Query Reference	UPDATE03
Query description	Update products in cart
Query frequency	hundreds per day
<pre>UPDATE cartproduct SET quantity=\$quantity WHERE id_product=\$id_product AND id_client=\$id_client;</pre>	

Query Reference	INSERT01
Query description	Create new user
Query frequency	units per day
<pre> INSERT INTO users(id, firstName, lastName, username, email, password, imageURL, dateCreated, dateModified, active) VALUES (\$id, \$firstName, \$lastName, \$username, \$email, \$password, \$imageURL, DEFAULT, DEFAULT, true); </pre>	

Query Reference	INSERT02
Query description	Create new product
Query frequency	units per day
<pre> INSERT INTO product(id, name, quantityInStock, modelNumber, weight, price, imageURL, bigDescription, shortDescription, id_brand, id_category) VALUES(\$id, \$name, \$quantityInStock, \$modelNumber, \$weight, \$price, \$imageURL, \$bigDescription, \$shortDescription, \$id_brand, \$id_category); </pre>	

Query Reference	INSERT03
Query description	Add to cart
Query frequency	hundreds per day
<pre> INSERT INTO cartproduct (id_client, id_product, quantity) VALUES (\$id_client, \$id_product, \$quantity); </pre>	

Query Reference	INSERT04
Query description	Add to wishlist
Query frequency	units per day
<pre> INSERT INTO productwishlist (id_product, id_client) VALUES (\$id_product, \$id_client); </pre>	

Query Reference	INSERT05
Query description	Add to purchase list
Query frequency	units per day
<pre> INSERT INTO purchase(id, id_client, id_address, purchaseDate, purchaseState, cost, paymentType, cardNumber, cardName, cardExpirationDate, nif) VALUES (\$id_client,\$id_address, DEFAULT, \$purchaseState, \$cost, \$paymentType, \$cardNumber, \$cardName, \$cardExpirationDate, \$nif); </pre>	

Query Reference	INSERT06
Query description	Add purchase to product
Query frequency	units per day
<pre>INSERT INTO purchaseproduct (id_purchase, id_product, quantity, cost) VALUES (\$id_purchase, \$id_product, \$quantity, \$cost);</pre>	

Query Reference	INSERT07
Query description	Add message
Query frequency	dozens per day
<pre>INSERT INTO message (id, message, dateSent, sender, id_chatSupport, id_client) VALUES(\$id, \$message, DEFAULT, \$sender, \$id_chatSupport, \$id_client);</pre>	

Query Reference	DELETE01
Query description	Delete from cart
Query frequency	dozens per day
<pre>DELETE FROM cartproduct WHERE id_product=\$id_product AND id_client=\$id_client;</pre>	

Query Reference	DELETE02
Query description	Delete from wishlist
Query frequency	dozens per month
<pre>DELETE FROM productwishlist WHERE id_product=\$id_product AND id_client=\$id_client;</pre>	

Query Reference	DELETE03
Query description	Delete product
Query frequency	units per month
<pre>DELETE FROM product WHERE id_product=\$id_product;</pre>	

2 Proposed Indexes

2.1 Performance Indexes

Index Reference	Index01
Related Queries	SELECT09
Index Relation	message
Index Attribute	id_client
Index Type	hash
Cardinality	medium
Clustering	Yes
Justification	Query SELECT09, used to search the messages of an Client to a SupportChat, has to be fast because it's executed several times; doesn't need range query support; cardinality is medium so it's a good candidate for clustering.
<pre>CREATE INDEX idx_message ON message USING hash (id_client);</pre>	

Index Reference	Index02
Related Queries	SELECT02
Index Relation	product
Index Attribute	name
Index Type	hash
Cardinality	medium
Clustering	Yes
Justification	Query SELECT02, used to search the products, has to be fast because it's executed several times; doesn't need range query support; cardinality is medium so it's a good candidate for clustering.
<pre>CREATE INDEX idx_product ON product USING hash (name);</pre>	

Index Reference	Index03
Related Queries	SELECT07
Index Relation	purchase
Index Attribute	id_client
Index Type	btree
Cardinality	medium
Clustering	Yes
Justification	Query SELECT07, used to search the products in all Client Purchases, has to be fast because it's executed several times; doesn't need range query support; cardinality is medium so it's a good candidate for clustering.
<pre>CREATE INDEX idx_purchase ON purchase USING btree (id_client);</pre>	

2.2 Full-text Search indexes

Index Reference	Index04
Related Queries	SELECT03
Index Relation	productCategory
Index Attribute	categoryName
Index Type	GiST
Clustering	No
Justification	Query SELECT03, used to search the products by category name, has to be fast because it's executed several times; GiST because it's better for dynamic data.
<pre>CREATE INDEX search_idx ON productcategory USING GIST (to_tsvector('english', categoryName));</pre>	

3 Triggers

Trigger reference	Trigger01
Trigger description	An admin can't be banned
<pre>CREATE FUNCTION ban_admin() RETURNS TRIGGER AS \$BODY\$ BEGIN IF EXISTS (SELECT * FROM admin WHERE NEW.id_user = admin.id) THEN RAISE EXCEPTION 'An Admin can not be banned'; END IF; RETURN NEW; END \$BODY\$ LANGUAGE plpgsql; CREATE TRIGGER ban_admin BEFORE INSERT OR UPDATE ON ban FOR EACH ROW EXECUTE PROCEDURE ban_admin();</pre>	

Trigger reference	Trigger02
Trigger description	The cost of a purchase must be the sum of the current prices of the products
<pre>CREATE FUNCTION purchase_cost() RETURNS TRIGGER AS \$BODY\$ BEGIN UPDATE purchaseProduct SET cost = (SELECT (SELECT price FROM product WHERE id=NEW.id_product) * NEW.quantity) WHERE purchaseProduct.id_product = NEW.id_product AND purchaseProduct.id_purchase = NEW.id_purchase; UPDATE purchase SET cost = (SELECT SUM(cost2) FROM (SELECT purchaseProduct.cost AS cost2 FROM purchaseProduct WHERE (purchaseProduct.id_purchase = NEW.id_purchase)) AS derived_table) WHERE purchase.id = NEW.id_purchase; RETURN NEW; END \$BODY\$ LANGUAGE plpgsql; CREATE TRIGGER purchase_cost AFTER INSERT ON purchaseProduct FOR EACH ROW EXECUTE PROCEDURE purchase_cost();</pre>	

4 Complete SQL Code

The following are github links to the sql code:

- https://beatrizhenriquesmartins.github.io/lbaw1736/src/sql/lbaw1736_create.sql
- https://beatrizhenriquesmartins.github.io/lbaw1736/src/sql/lbaw1736_data.sql

4.1 Create Database

```
1  -- Drops
2  DROP Table IF EXISTS brandBrandManager CASCADE;
3  DROP TABLE IF EXISTS productreview CASCADE;
4  DROP TABLE IF EXISTS purchaseproduct CASCADE;
5  DROP TABLE IF EXISTS purchase CASCADE;
6  DROP TABLE IF EXISTS clientaddress CASCADE;
7  DROP TABLE IF EXISTS country CASCADE;
8  DROP TABLE IF EXISTS city CASCADE;
9  DROP TABLE IF EXISTS address CASCADE;
10 DROP TABLE IF EXISTS cartproduct CASCADE;
11 DROP TABLE IF EXISTS productwishlist CASCADE;
12 DROP TABLE IF EXISTS product CASCADE;
13 DROP TABLE IF EXISTS productcategory CASCADE;
14 DROP TABLE IF EXISTS ban CASCADE;
15 DROP TABLE IF EXISTS admin CASCADE;
16 DROP TABLE IF EXISTS brandManager CASCADE;
17 DROP TABLE IF EXISTS brand CASCADE;
18 DROP TABLE IF EXISTS message CASCADE;
19 DROP TABLE IF EXISTS client CASCADE;
20 DROP TABLE IF EXISTS chatSupport CASCADE;
21 DROP TABLE IF EXISTS users CASCADE;
22
23 DROP FUNCTION IF EXISTS ban_admin() CASCADE;
24 DROP TRIGGER IF EXISTS ban_admin ON ban CASCADE;
25 DROP FUNCTION IF EXISTS purchase_cost() CASCADE;
26 DROP TRIGGER IF EXISTS purchase_cost ON purchase CASCADE;
27
28 -- Tables
29
30 CREATE TABLE users (
31     id SERIAL PRIMARY KEY,
32     firstName TEXT NOT NULL,
33     lastName TEXT NOT NULL,
34     username TEXT NOT NULL UNIQUE,
35     email TEXT NOT NULL UNIQUE,
36     password TEXT NOT NULL,
37     imageURL TEXT NOT NULL,
38     dateCreated TIMESTAMP DEFAULT now() NOT NULL,
39     dateModified TIMESTAMP DEFAULT now() NOT NULL,
40     active BOOLEAN NOT NULL
41 );
42
43 CREATE TABLE chatSupport (
44     id_chatSupport INTEGER PRIMARY KEY REFERENCES users
45 );
46
47 CREATE TABLE client (
48     id_client INTEGER PRIMARY KEY REFERENCES users,
49     cellphone INTEGER
50 );
51
52 CREATE TABLE message (
53     id SERIAL PRIMARY KEY,
```

```

54     message TEXT NOT NULL,
55     dateSent TIMESTAMP DEFAULT now() NOT NULL,
56     sender TEXT NOT NULL CHECK((sender = ANY (ARRAY['Client'::text,
    ↪     'ChatSupport'::text]))),
57     id_chatSupport INTEGER NOT NULL REFERENCES chatSupport,
58     id_client INTEGER NOT NULL REFERENCES client
59 );
60
61 CREATE TABLE brand (
62     id SERIAL PRIMARY KEY,
63     name TEXT NOT NULL UNIQUE,
64     contact INTEGER NOT NULL
65 );
66
67 CREATE TABLE brandManager (
68     id_brandManager INTEGER PRIMARY KEY REFERENCES users
69 );
70
71 CREATE TABLE admin (
72     id INTEGER PRIMARY KEY REFERENCES brandManager
73 );
74
75 CREATE TABLE ban (
76     id_user INTEGER PRIMARY KEY REFERENCES users,
77     id_admin INTEGER REFERENCES admin NOT NULL,
78     banDate TIMESTAMP DEFAULT now() NOT NULL
79 );
80
81 CREATE TABLE productcategory (
82     id SERIAL PRIMARY KEY,
83     categoryName TEXT NOT NULL UNIQUE
84 );
85
86 CREATE TABLE product (
87     id SERIAL PRIMARY KEY,
88     name TEXT UNIQUE NOT NULL,
89     quantityInStock INTEGER NOT NULL DEFAULT 0,
90     dateCreated TIMESTAMP DEFAULT now() NOT NULL,
91     modelNumber INTEGER NOT NULL,
92     weight DECIMAL NOT NULL,
93     price MONEY NOT NULL,
94     imageURL TEXT NOT NULL UNIQUE,
95     bigDescription TEXT NOT NULL,
96     shortDescription TEXT NOT NULL,
97     id_brand INTEGER NOT NULL REFERENCES brand,
98     id_category INTEGER NOT NULL REFERENCES productcategory
99 );
100
101 CREATE TABLE productwishlist (
102     id_product INTEGER REFERENCES product,
103     id_client INTEGER REFERENCES client,
104     PRIMARY KEY(id_product, id_client)
105 );
106
107 CREATE TABLE cartproduct (
108     id_client INTEGER REFERENCES client,
109     id_product INTEGER REFERENCES product,
110     quantity INTEGER NOT NULL CHECK (quantity > 0),
111     PRIMARY KEY(id_client, id_product)
112 );
113
114 CREATE TABLE country (

```

```

115     id SERIAL PRIMARY KEY,
116     country TEXT NOT NULL UNIQUE
117 );
118
119 CREATE TABLE city (
120     id SERIAL PRIMARY KEY,
121     city TEXT NOT NULL,
122     id_country INTEGER NOT NULL REFERENCES country
123 );
124
125 CREATE TABLE address (
126     id SERIAL PRIMARY KEY,
127     address TEXT NOT NULL,
128     zipcode TEXT NOT NULL,
129     id_city INTEGER NOT NULL REFERENCES city
130 );
131
132 CREATE TABLE clientaddress (
133     id_client INTEGER NOT NULL REFERENCES client,
134     id_address INTEGER NOT NULL REFERENCES address,
135     PRIMARY KEY(id_client, id_address)
136 );
137
138 CREATE TABLE purchase (
139     id SERIAL PRIMARY KEY,
140     id_client INTEGER REFERENCES client NOT NULL,
141     id_address INTEGER REFERENCES address NOT NULL,
142     purchaseDate TIMESTAMP DEFAULT now() NOT NULL,
143     purchaseState TEXT NOT NULL,
144     cost MONEY NOT NULL CHECK (cost > CAST ( 0 AS MONEY )),
145     paymentType TEXT NOT NULL,
146     cardNumber TEXT NOT NULL,
147     cardName TEXT NOT NULL,
148     cardExpirationDate TIMESTAMP NOT NULL,
149     nif INTEGER NOT NULL,
150     CHECK (cardExpirationDate > purchaseDate)
151 );
152
153 CREATE TABLE purchaseproduct (
154     id_purchase INTEGER REFERENCES purchase,
155     id_product INTEGER NOT NULL REFERENCES product,
156     quantity INTEGER NOT NULL CHECK (quantity > 0),
157     cost MONEY NOT NULL CHECK (cost > CAST ( 0 AS MONEY )),
158     PRIMARY KEY(id_purchase, id_product)
159 );
160
161 CREATE TABLE productreview (
162     id_product INTEGER NOT NULL REFERENCES product,
163     id_purchase INTEGER NOT NULL REFERENCES purchase,
164     reviewDate TIMESTAMP DEFAULT now() NOT NULL,
165     textReview TEXT NOT NULL,
166     rating INTEGER NOT NULL CHECK (((rating >= 0) AND (rating <= 5))),
167     PRIMARY KEY(id_product, id_purchase)
168 );
169
170 CREATE TABLE brandBrandManager (
171     idBrand INTEGER NOT NULL REFERENCES brand,
172     idBrandManager INTEGER NOT NULL REFERENCES brandManager,
173     PRIMARY KEY(idBrand, idBrandManager)
174 );
175
176 -- Indexes

```

```

177
178 CREATE INDEX idx_message ON message USING hash (id_client); /*cardinalidade
    ↳ media-bom candidato para cluster*/
179
180 CREATE INDEX idx_product ON product USING hash (name); /*cardinalidade media-bom
    ↳ candidato para cluster*/
181
182 CREATE INDEX idx_purchase ON purchase USING btree (id_client); /*cardinalidade
    ↳ media-bom candidato para cluster*/
183
184 CREATE INDEX search_idx ON productcategory USING GIST (to_tsvector('english',
    ↳ categoryName));
185
186 -- Triggers
187
188 --Trigger that prevents an admin from being banned
189 CREATE FUNCTION ban_admin() RETURNS TRIGGER AS
190 $BODY$
191 BEGIN
192     IF EXISTS (SELECT * FROM admin WHERE NEW.id_user = admin.id) THEN
193         RAISE EXCEPTION 'An Admin can not be banned';
194     END IF;
195     RETURN NEW;
196 END
197 $BODY$
198 LANGUAGE plpgsql;
199
200 --Trigger that actualizes the of a product in a purchase according to its actual
    ↳ value
201 CREATE TRIGGER ban_admin
202     BEFORE INSERT OR UPDATE ON ban
203     FOR EACH ROW
204         EXECUTE PROCEDURE ban_admin();
205
206
207
208 CREATE FUNCTION purchase_cost() RETURNS TRIGGER AS
209 $BODY$
210 BEGIN
211     UPDATE purchaseProduct
212     SET cost = (SELECT (SELECT price FROM product WHERE id=NEW.id_product) *
        ↳ NEW.quantity) WHERE purchaseProduct.id_product = NEW.id_product AND
        ↳ purchaseProduct.id_purchase = NEW.id_purchase;
213     UPDATE purchase
214     SET cost = (SELECT SUM(cost2) FROM (SELECT purchaseProduct.cost AS cost2
        ↳ FROM purchaseProduct WHERE ( purchaseProduct.id_purchase =
        ↳ NEW.id_purchase) ) AS derived_table) WHERE purchase.id =
        ↳ NEW.id_purchase;
215     RETURN NEW;
216 END
217 $BODY$
218 LANGUAGE plpgsql;
219
220
221
222 CREATE TRIGGER purchase_cost
223     AFTER INSERT ON purchaseProduct
224     FOR EACH ROW
225         EXECUTE PROCEDURE purchase_cost();

```

4.2 Data

```
1  INSERT INTO users VALUES (1, 'Luis', 'Saraiva', 'admin_luissaraiva',
   → 'a_luissaraiva@gmail.com', '1234', '', DEFAULT, DEFAULT, true);
2  INSERT INTO users VALUES (2, 'Beatriz', 'Henriques', 'admin_beatriz',
   → 'a_beatriz@gmail.com', '1234', '', DEFAULT, DEFAULT, true);
3  INSERT INTO users VALUES (3, 'Francisco', 'Andrade', 'admin_francisco',
   → 'a_francisco@gmail.com', '1234', '', DEFAULT, DEFAULT, true);
4  INSERT INTO users VALUES (4, 'Ricardo', 'Abreu', 'admin_ricardo',
   → 'a_ricardo@gmail.com', '1234', '', DEFAULT, DEFAULT, true);
5  INSERT INTO users VALUES (5, 'Bernardo', 'Leite', 'sprt_bernardoleite',
   → 'sprt_bernardo@gmail.com', '1234', '', DEFAULT, DEFAULT, true);
6  INSERT INTO users VALUES (6, 'Antonio', 'Pires', 'sprt_antoniopires',
   → 'sprt_antonio@gmail.com', '1234', '', DEFAULT, DEFAULT, true);
7  INSERT INTO users VALUES (7, 'Jose', 'Cunha', 'sprt_josecunha',
   → 'sprt_jose@gmail.com', '1234', '', DEFAULT, DEFAULT, true);
8  INSERT INTO users VALUES (8, 'Tomas', 'Costa', 'bm_tomascosta',
   → 'bm_tomas@gmail.com', '1234', '', DEFAULT, DEFAULT, true);
9  INSERT INTO users VALUES (9, 'Diogo', 'Silva', 'bm_diogosilva',
   → 'bm_diogo@gmail.com', '1234', '', DEFAULT, DEFAULT, true);
10 INSERT INTO users VALUES (10, 'Joao', 'Monteiro', 'bm_joamonteiro',
   → 'bm_joao@gmail.com', '1234', '', DEFAULT, DEFAULT, true);
11 INSERT INTO users VALUES (11, 'Pedro', 'Goncalves', 'pedrogoncalves',
   → 'pedro@gmail.com', '1234', '', DEFAULT, DEFAULT, true);
12 INSERT INTO users VALUES (12, 'Mariana', 'Oliveira', 'marianaoliveira',
   → 'mariana@gmail.com', '1234', '', DEFAULT, DEFAULT, true);
13 INSERT INTO users VALUES (13, 'Francisca', 'Rodrigues', 'francisarodrigues',
   → 'francisca@gmail.com', '1234', '', DEFAULT, DEFAULT, true);
14 INSERT INTO users VALUES (14, 'Ines', 'Pinto', 'inespinto', 'ines@gmail.com',
   → '1234', '', DEFAULT, DEFAULT, true);
15 INSERT INTO users VALUES (15, 'Raquel', 'Pereira', 'raquelpereira',
   → 'raquel@gmail.com', '1234', '', DEFAULT, DEFAULT, true);
16 INSERT INTO users VALUES (16, 'Margarida', 'Santos', 'margaridasantos',
   → 'margarida@gmail.com', '1234', '', DEFAULT, DEFAULT, true);
17 INSERT INTO users VALUES (17, 'Teresa', 'Brito', 'teresabrito',
   → 'teresa@gmail.com', '1234', '', DEFAULT, DEFAULT, true);
18 INSERT INTO users VALUES (18, 'Tiago', 'Carvalho', 'tiagocarvalho',
   → 'tiago@gmail.com', '1234', '', DEFAULT, DEFAULT, true);
19 INSERT INTO users VALUES (19, 'Maria', 'Coelho', 'mariacoelho', 'maria@gmail.com',
   → '1234', '', DEFAULT, DEFAULT, true);
20 INSERT INTO users VALUES (20, 'Ana', 'Ferreira', 'anaferreira', 'ana@gmail.com',
   → '1234', '', DEFAULT, DEFAULT, true);
21
22
23 INSERT INTO chatSupport VALUES (5);
24 INSERT INTO chatSupport VALUES (6);
25 INSERT INTO chatSupport VALUES (7);
26
27
28 INSERT INTO client VALUES (11, 923132456);
29 INSERT INTO client VALUES (12, 929432849);
30 INSERT INTO client VALUES (13, 921232133);
31 INSERT INTO client VALUES (14, 912375785);
32 INSERT INTO client VALUES (15, 918478539);
33 INSERT INTO client VALUES (16, 919453854);
34 INSERT INTO client VALUES (17, 919648954);
35 INSERT INTO client VALUES (18, 914834304);
36 INSERT INTO client VALUES (19, 923456536);
37 INSERT INTO client VALUES (20, 924254676);
38
39 INSERT INTO message VALUES (1, 'teste msg', DEFAULT, 'Client', 5, 11);
40 INSERT INTO message VALUES (2, 'teste msg', DEFAULT, 'ChatSupport', 5, 11);
```

```

41 INSERT INTO message VALUES (3, 'teste msg', DEFAULT, 'Client', 5, 11);
42 INSERT INTO message VALUES (4, 'teste msg', DEFAULT, 'ChatSupport', 6, 12);
43 INSERT INTO message VALUES (5, 'teste msg', DEFAULT, 'Client', 6, 12);
44 INSERT INTO message VALUES (6, 'teste msg', DEFAULT, 'ChatSupport', 6, 12);
45 INSERT INTO message VALUES (7, 'teste msg', DEFAULT, 'Client', 6, 12);
46 INSERT INTO message VALUES (8, 'teste msg', DEFAULT, 'ChatSupport', 7, 15);
47 INSERT INTO message VALUES (9, 'teste msg', DEFAULT, 'Client', 7, 15);
48 INSERT INTO message VALUES (10, 'teste msg', DEFAULT, 'ChatSupport', 5, 18);
49 INSERT INTO message VALUES (11, 'teste msg', DEFAULT, 'Client', 5, 18);
50 INSERT INTO message VALUES (12, 'teste msg', DEFAULT, 'ChatSupport', 5, 18);
51 INSERT INTO message VALUES (13, 'teste msg', DEFAULT, 'Client', 5, 18);
52 INSERT INTO message VALUES (14, 'teste msg', DEFAULT, 'Client', 6, 20);
53 INSERT INTO message VALUES (15, 'teste msg', DEFAULT, 'ChatSupport', 6, 20);
54
55
56 INSERT INTO brand VALUES (1, 'Alameda Turquesa', 223143243);
57 INSERT INTO brand VALUES (2, 'Aldeia da Roupa Branca', 223143243);
58 INSERT INTO brand VALUES (3, 'Alma de Luce', 223143243);
59 INSERT INTO brand VALUES (4, 'Ame Moi', 223143243);
60 INSERT INTO brand VALUES (5, 'Ana Leite', 223143243);
61 INSERT INTO brand VALUES (6, 'Anita Picnic', 223143243);
62 INSERT INTO brand VALUES (7, 'Antiflop', 223143243);
63 INSERT INTO brand VALUES (8, 'Aparattus', 223143243);
64 INSERT INTO brand VALUES (9, 'Babash Design', 223143243);
65 INSERT INTO brand VALUES (10, 'Bateye', 223143243);
66 INSERT INTO brand VALUES (11, 'Bluf', 223143243);
67 INSERT INTO brand VALUES (12, 'Boca do Lobo', 223143243);
68 INSERT INTO brand VALUES (13, 'Bordallo Pinheiro', 223143243);
69 INSERT INTO brand VALUES (14, 'Briel', 223143243);
70 INSERT INTO brand VALUES (15, 'Cabo d Mar', 223143243);
71 INSERT INTO brand VALUES (16, 'Candle In', 223143243);
72 INSERT INTO brand VALUES (17, 'Cante', 223143243);
73 INSERT INTO brand VALUES (18, 'Castelbel', 223143243);
74 INSERT INTO brand VALUES (19, 'Cavalinho', 223143243);
75 INSERT INTO brand VALUES (20, 'Chicos', 223143243);
76 INSERT INTO brand VALUES (21, 'Claus Porto', 223143243);
77 INSERT INTO brand VALUES (22, 'Coloradd', 223143243);
78 INSERT INTO brand VALUES (23, 'Deamor', 223143243);
79 INSERT INTO brand VALUES (24, 'Decenio', 223143243);
80 INSERT INTO brand VALUES (25, 'Design Flops', 223143243);
81 INSERT INTO brand VALUES (26, 'Doodles', 223143243);
82 INSERT INTO brand VALUES (27, 'Dub Dressed', 223143243);
83 INSERT INTO brand VALUES (28, 'Ecola', 223143243);
84 INSERT INTO brand VALUES (29, 'Enamorata', 223143243);
85 INSERT INTO brand VALUES (30, 'Eureka Shoes', 223143243);
86 INSERT INTO brand VALUES (31, 'Fasm', 223143243);
87 INSERT INTO brand VALUES (32, 'Fio Rosa', 223143243);
88 INSERT INTO brand VALUES (33, 'Science4You', 223143243);
89
90 INSERT INTO brandManager VALUES (1);
91 INSERT INTO brandManager VALUES (2);
92 INSERT INTO brandManager VALUES (3);
93 INSERT INTO brandManager VALUES (4);
94 INSERT INTO brandManager VALUES (8);
95 INSERT INTO brandManager VALUES (9);
96 INSERT INTO brandManager VALUES (10);
97
98 INSERT INTO admin VALUES (1);
99 INSERT INTO admin VALUES (2);
100 INSERT INTO admin VALUES (3);
101 INSERT INTO admin VALUES (4);
102

```



```

103 INSERT INTO ban VALUES (20, 1, DEFAULT);
104
105 INSERT INTO productcategory VALUES (1, 'Fashion');
106 INSERT INTO productcategory VALUES (2, 'Beauty');
107 INSERT INTO productcategory VALUES (3, 'Technology');
108 INSERT INTO productcategory VALUES (4, 'Food');
109 INSERT INTO productcategory VALUES (5, 'Culture');
110 INSERT INTO productcategory VALUES (6, 'Home');
111 INSERT INTO productcategory VALUES (7, 'Sports');
112
113 INSERT INTO product VALUES (1, 'Gelato', 10, now(), 1, 3.0, 30.00,
    → './src/images/brands/alameda_turquesa/01-holi-400x400.jpg', '', '', 1, 1);
114 INSERT INTO product VALUES (2, 'Holi', 10, now(), 1, 3.0, 30.00,
    → './src/images/brands/alameda_turquesa/2-alameda-turquesa-gelato-400x400.jpg',
    → '', '', 1, 1);
115 INSERT INTO product VALUES (3, 'Chizela', 10, now(), 1, 3.0, 30.00,
    → './src/images/brands/alameda_turquesa/03-chizela-black-alameda-turquesa.png',
    → '', '', 1, 1);
116 INSERT INTO product VALUES (4, 'Frozen', 10, now(), 1, 3.0, 30.00,
    → './src/images/brands/alameda_turquesa/3-frozen-sneakers-alamedaturquesa-400x400.jpg',
    → '', '', 1, 1);
117 INSERT INTO product VALUES (6, 'Cardosas', 10, now(), 1, 3.0, 300.00,
    → './src/images/brands/alma_de_luce/1.jpg', '', '', 3, 6);
118 INSERT INTO product VALUES (5, 'Aparador Multi-Gavetas', 10, now(), 1, 3.0,
    → 300.00, './src/images/brands/alma_de_luce/3.jpg', '', '', 3, 6);
119
120
121 INSERT INTO productwishlist VALUES (1, 11);
122 INSERT INTO productwishlist VALUES (2, 12);
123 INSERT INTO productwishlist VALUES (3, 12);
124 INSERT INTO productwishlist VALUES (4, 14);
125 INSERT INTO productwishlist VALUES (5, 15);
126 INSERT INTO productwishlist VALUES (1, 16);
127 INSERT INTO productwishlist VALUES (4, 16);
128 INSERT INTO productwishlist VALUES (1, 19);
129 INSERT INTO productwishlist VALUES (2, 19);
130 INSERT INTO productwishlist VALUES (4, 19);
131
132
133 INSERT INTO cartproduct VALUES (11, 1, 2);
134 INSERT INTO cartproduct VALUES (11, 3, 2);
135 INSERT INTO cartproduct VALUES (11, 2, 2);
136 INSERT INTO cartproduct VALUES (12, 1, 2);
137 INSERT INTO cartproduct VALUES (13, 5, 1);
138 INSERT INTO cartproduct VALUES (13, 2, 2);
139 INSERT INTO cartproduct VALUES (13, 3, 2);
140 INSERT INTO cartproduct VALUES (13, 1, 1);
141 INSERT INTO cartproduct VALUES (14, 2, 1);
142 INSERT INTO cartproduct VALUES (14, 1, 1);
143 INSERT INTO cartproduct VALUES (16, 4, 1);
144 INSERT INTO cartproduct VALUES (16, 2, 2);
145 INSERT INTO cartproduct VALUES (17, 3, 2);
146 INSERT INTO cartproduct VALUES (17, 1, 2);
147 INSERT INTO cartproduct VALUES (18, 4, 1);
148 INSERT INTO cartproduct VALUES (19, 5, 3);
149
150
151 INSERT INTO country VALUES (1, 'Portugal');
152 INSERT INTO country VALUES (2, 'Finland');
153 INSERT INTO country VALUES (3, 'Denmark');
154 INSERT INTO country VALUES (4, 'Canada');
155 INSERT INTO country VALUES (5, 'Brazil');

```

```

156 INSERT INTO country VALUES (6, 'Australia');
157 INSERT INTO country VALUES (7, 'Austria');
158 INSERT INTO country VALUES (8, 'Poland');
159 INSERT INTO country VALUES (9, 'Belgium');
160 INSERT INTO country VALUES (10, 'Switzerland');
161 INSERT INTO country VALUES (11, 'Italy');
162 INSERT INTO country VALUES (12, 'Netherlands');
163 INSERT INTO country VALUES (13, 'Germany');
164 INSERT INTO country VALUES (14, 'France');
165 INSERT INTO country VALUES (15, 'Spain');
166 INSERT INTO country VALUES (16, 'United Kingdom');
167 INSERT INTO country VALUES (17, 'USA');
168 INSERT INTO country VALUES (18, 'Greece');
169 INSERT INTO country VALUES (19, 'Norway');
170 INSERT INTO country VALUES (20, 'Mozambique');
171 INSERT INTO country VALUES (21, 'Angola');
172 INSERT INTO country VALUES (22, 'Ireland');
173 INSERT INTO country VALUES (23, 'Hungary');
174 INSERT INTO country VALUES (24, 'Romania');
175 INSERT INTO country VALUES (25, 'Sweden');
176 INSERT INTO country VALUES (26, 'Slovenia');
177
178 INSERT INTO city VALUES (1, 'Porto', 1);
179 INSERT INTO city VALUES (2, 'Lamego', 1);
180 INSERT INTO city VALUES (3, 'Lisboa', 1);
181 INSERT INTO city VALUES (4, 'Coimbra', 1);
182 INSERT INTO city VALUES (5, 'Faro', 1);
183 INSERT INTO city VALUES (6, 'Bragança', 1);
184 INSERT INTO city VALUES (7, 'Leiria', 1);
185 INSERT INTO city VALUES (8, 'Viana do Castelo', 1);
186 INSERT INTO city VALUES (9, 'Braga', 1);
187 INSERT INTO city VALUES (10, 'Guarda', 1);
188
189
190
191 INSERT INTO address VALUES (1, 'Rua Marco de Canaveses 19', '4100-123', 1);
192 INSERT INTO address VALUES (2, 'Avenida 5 de Outubro 52', '4312-234', 2);
193 INSERT INTO address VALUES (3, 'Rua de Moçambique 138', '4152-164', 4);
194 INSERT INTO address VALUES (4, 'Rua Ferreira Lapa 49', '4291-134', 3);
195 INSERT INTO address VALUES (5, 'Rua General Alberto Delgado 534', '3125-343', 5);
196 INSERT INTO address VALUES (6, 'Rua Professor Egas Moniz 243', '3322-421', 6);
197 INSERT INTO address VALUES (7, 'Rua Manuel Simões Maia 123', '2314-432', 7);
198 INSERT INTO address VALUES (8, 'Rua dos Poveiros 77', '2151-542', 8);
199 INSERT INTO address VALUES (9, 'Rua Damião de Góis 211', '1312-321', 9);
200 INSERT INTO address VALUES (10, 'Rua Serpa Pinto 15', '1532-253', 10);
201
202
203 INSERT INTO purchase VALUES (1, 11, 1, DEFAULT, 'verificacao', 300, 'cartao
→ credito', '5400 4102 4021 7362', 'Pedro Gonçalves', '2018-07-23', 235123482);
204 INSERT INTO purchase VALUES (2, 11, 2, DEFAULT, 'verificacao', 300, 'cartao
→ credito', '5400 4102 4021 7362', 'Pedro Gonçalves', '2018-07-23', 235123482);
205 INSERT INTO purchase VALUES (3, 12, 3, DEFAULT, 'verificacao', 300, 'cartao
→ credito', '5400 4102 4021 7362', 'Pedro Gonçalves', '2018-07-23', 212393204);
206 INSERT INTO purchase VALUES (4, 12, 4, DEFAULT, 'verificacao', 300, 'cartao
→ credito', '5400 4102 4021 7362', 'Pedro Gonçalves', '2018-07-23', 235123482);
207 INSERT INTO purchase VALUES (5, 12, 5, DEFAULT, 'verificacao', 300, 'cartao
→ credito', '5400 4102 4021 7362', 'Pedro Gonçalves', '2018-07-23', 235123482);
208 INSERT INTO purchase VALUES (6, 15, 6, DEFAULT, 'verificacao', 300, 'cartao
→ credito', '5400 4102 4021 7362', 'Pedro Gonçalves', '2018-07-23', 235123482);
209 INSERT INTO purchase VALUES (7, 17, 7, DEFAULT, 'verificacao', 300, 'cartao
→ credito', '5400 4102 4021 7362', 'Pedro Gonçalves', '2018-07-23', 235123482);

```

```

210 INSERT INTO purchase VALUES (8, 17, 8, DEFAULT, 'verificacao', 300, 'cartao
    ↳ credito', '5400 4102 4021 7362', 'Pedro Gonçalves', '2018-07-23', 235123482);
211 INSERT INTO purchase VALUES (9, 19, 9, DEFAULT, 'verificacao', 300, 'cartao
    ↳ credito', '5400 4102 4021 7362', 'Pedro Gonçalves', '2018-07-23', 235123482);
212 INSERT INTO purchase VALUES (10, 19, 10, DEFAULT, 'verificacao', 300, 'cartao
    ↳ credito', '5400 4102 4021 7362', 'Pedro Gonçalves', '2018-07-23', 235123482);
213
214
215 INSERT INTO purchaseproduct VALUES (1, 5, 1, 300);
216 INSERT INTO purchaseproduct VALUES (2, 2, 1, 300);
217 INSERT INTO purchaseproduct VALUES (2, 3, 1, 300);
218 INSERT INTO purchaseproduct VALUES (3, 1, 1, 300);
219 INSERT INTO purchaseproduct VALUES (4, 4, 1, 300);
220 INSERT INTO purchaseproduct VALUES (5, 1, 1, 300);
221 INSERT INTO purchaseproduct VALUES (6, 3, 1, 300);
222 INSERT INTO purchaseproduct VALUES (7, 3, 1, 300);
223 INSERT INTO purchaseproduct VALUES (8, 2, 1, 300);
224 INSERT INTO purchaseproduct VALUES (9, 2, 1, 300);
225 INSERT INTO purchaseproduct VALUES (10, 4, 1, 300);
226 INSERT INTO purchaseproduct VALUES (3, 3, 1, 300);
227 INSERT INTO purchaseproduct VALUES (4, 1, 1, 300);
228 INSERT INTO purchaseproduct VALUES (5, 5, 1, 300);
229 INSERT INTO purchaseproduct VALUES (6, 1, 1, 300);
230 INSERT INTO purchaseproduct VALUES (7, 2, 1, 300);
231 INSERT INTO purchaseproduct VALUES (8, 3, 1, 300);
232 INSERT INTO purchaseproduct VALUES (9, 4, 1, 300);
233 INSERT INTO purchaseproduct VALUES (10, 2, 1, 300);
234
235 INSERT INTO productreview VALUES (1, 5, DEFAULT, '', 3);
236 INSERT INTO productreview VALUES (2, 2, DEFAULT, '', 5);
237 INSERT INTO productreview VALUES (3, 2, DEFAULT, '', 5);
238 INSERT INTO productreview VALUES (4, 10, DEFAULT, '', 4);
239 INSERT INTO productreview VALUES (2, 8, DEFAULT, '', 2);
240 INSERT INTO productreview VALUES (1, 3, DEFAULT, '', 3);
241 INSERT INTO productreview VALUES (3, 6, DEFAULT, '', 5);
242 INSERT INTO productreview VALUES (5, 1, DEFAULT, '', 4);
243 INSERT INTO productreview VALUES (4, 4, DEFAULT, '', 4);
244
245 INSERT INTO brandBrandManager VALUES (1, 8);
246 INSERT INTO brandBrandManager VALUES (2, 9);
247 INSERT INTO brandBrandManager VALUES (3, 10);
248 INSERT INTO brandBrandManager VALUES (4, 8);
249 INSERT INTO brandBrandManager VALUES (5, 9);
250 INSERT INTO brandBrandManager VALUES (6, 10);
251 INSERT INTO brandBrandManager VALUES (7, 8);
252 INSERT INTO brandBrandManager VALUES (8, 9);
253 INSERT INTO brandBrandManager VALUES (9, 10);
254 INSERT INTO brandBrandManager VALUES (10, 8);
255 INSERT INTO brandBrandManager VALUES (11, 9);
256 INSERT INTO brandBrandManager VALUES (12, 10);
257 INSERT INTO brandBrandManager VALUES (13, 8);
258 INSERT INTO brandBrandManager VALUES (14, 9);
259 INSERT INTO brandBrandManager VALUES (15, 10);
260 INSERT INTO brandBrandManager VALUES (16, 8);
261 INSERT INTO brandBrandManager VALUES (17, 9);
262 INSERT INTO brandBrandManager VALUES (18, 10);

```

Revision history

Changes made to the first submission:

- The queries SELECT01 e SELECT02 were modified so that it is now possible to consult the name of a product category, the name of a product brand and a product rating
- Added query SELECT03 so that it is now possible to search a category name and consult its products
- Removed indexes Index01, Index04, Index05, Index07, Index08 as these were indexes for the primary keys of some relations
- Updated indexes Index02 and Index03 (indexes Index01 and Index02 after this revision) so that they now use hash instead of btree
- Added Index04, making this index an example of a full-text search index
- Updated SQL Code

GROUP1736, 04/04/2018

- 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