A9: Main accesses to the database and transactions

This artefact shows the main accesses to the database, including the transactions.

For each transaction, the isolation level is explicitly stated and read-only transactions are identified to improve global performance. For each identified access, the SQL code and the reference of web resources (A7) are provided.

1. Main Accesses

Main accesses to the database.

M01: Authentication and Profile

SQL101	Login a user.
Web Resource	R102

```
SELECT id
FROM users
WHERE email = $email AND password = $hashedPassword;
```

M02: Products and Categories

SQL201	Get products from a category.
Web Resource	R202

```
SELECT id
FROM categories
WHERE name = $categoryName;

SELECT products.id,products.name,products.price
FROM products
WHERE category_id = $categoryId AND products.id
NOT IN(SELECT * FROM archived_products);

SELECT path
FROM photos
WHERE product_id = $prod_id;
```

M03: Management Area

SQL301	Update purchases state.
Web Resource	R302

```
UPDATE purchases
SET status = $status
WHERE purchase_id = $purchaseId
```

M04: Product and Reviews

SQL401	Retrieve a products information.
Web Resource	R401

```
SELECT name, price, score, brand, quantity
FROM products
WHERE id = $prod_id;
```

M05: Static Pages

SQL501	Retrieve all frequently asked questions.
Web Resource	R501

```
SELECT *
FROM faqs;
```

2. Transactions

Transactions needed to assure the integrity of the data, with a proper justification.

T01	Retrieve a product's reviews.
Web Resource	R401
Isolation level	SERIALIZABLE READ ONLY
Justification	In the middle of the transaction, the insertion of new rows in the review can occur, which implies that the information retrieved in the query that selects the reviews counter and the query that selects the reviews is different, consequently resulting in a Phantom Read. It's READ ONLY because it only uses Selects.

```
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE READ ONLY

    SELECT COUNT(*)
```

```
    FROM reviews AS R
    WHERE R.product_id = $prod_id

    SELECT R.title, R.content, R.date, R.score, U.name
    FROM reviews AS R
    WHERE R.product_id = $prod_id

COMMIT;
```

T02	Retrieve a product, its information and properties.
Web Resource	R602
Isolation level	READ COMMITTED
Justification	The isolation level Read Committed ensures that an operation will never read data another application has changed but not yet committed. This means that when retrieving a product's information, all data should be equally updated and committed.

```
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL READ COMMITTED

    SELECT name, price, score, brand, quantity
    FROM products
    WHERE id = $prod_id

    SELECT path
    FROM photos
    WHERE product_id = $prod_id

    SELECT V.name, P.name
    FROM values AS V, values_lists AS VL, category_properties AS CP,
properties AS P
    WHERE VL.product_id = $prod_id
    AND V.values_list_id = VL.id
    AND VL.category_property_id = CP.id AND CP.property_id = P.id

COMMIT;
```

T03	Complete a new purchase
Web Resource	R602
Isolation level	REPEATABLE READ
Justification	In order to maintain consistency, it's necessary to use a transaction to ensure that the all the

code executes without errors. If an error occurs, a ROLLBACK is issued (when the available quantity is not enough nor the insertion of a product_purchase fails, per example). The isolation level is Repeatable Read, because otherwise, an update of product.quantity could happen, due to an update in the table products committed by a concurrent transaction, and as a result, there could be not enough product quantity.

```
BEGIN TRANSACTION;
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ

    IF EXISTS (SELECT quantity_available FROM products
               WHERE id = $proId
               AND quantity_available >= $quantity)
    BEGIN
        INSERT INTO purchases (total, user_id, address_id)
        VALUES ($total, $userid, $address_id)

        INSERT INTO product_purchases (product_id, purchase_id, quantity,
price)
        VALUES ($proId, $purchId, $quantity, $price)

        UPDATE products
        SET quantity = quantity - $quantity
        WHERE product_id = $proId
    END

COMMIT;
```

Revision history

Changes made to the first submission:

1. Added and changed transactions.

GROUP1761, 01/05/2018

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