

A9: Acessos principais à base de dados e transações

Este artefacto tem dois elementos principais: o primeiro é a identificação de, para cada módulo definido no artefacto A7, quais os acessos principais à base de dados realizados em cada módulo. Cada acesso foi documentado com o respectivo código SQL e a referência para o(s) recursos Web a que está associado. O segundo elemento é a definição de transações para acessos à base de dados que delas necessitem, tendo-se justificado, para cada transação, a razão pela qual é necessária e o motivo da escolha do seu nível de isolamento. É incluído também o código SQL necessário para a implementação de cada transação.

1. Acessos principais

M01: Autentificação e perfil individual

SQL101	Criar um novo utilizador
Recurso Web	R105
<pre>-- Insert user INSERT INTO member (address, age, email, name, password, phone, postalCode, username, idCountry) VALUES (\$address, \$age, \$email, \$name, \$password, \$phone, \$postalCode, \$username, \$idCountry); -- Insert image INSERT INTO image (source, idusers) VALUES (\$source, \$idusers);</pre>	
SQL102	Obter perfil de um utilizador
Recurso Web	R106
<pre>--select user and image SELECT * FROM member, image WHERE member.idImage = image.id AND member.id = \$userID; --select comments SELECT datePosted, liked, text, idParent, idSender, idReceiver, username FROM comment, member WHERE comment.idParent = \$profileID AND is_removed = false AND comment.idSender = member.id;</pre>	

M02: Leilões

SQL201	Obter leilões usando múltiplos parâmetros
Recurso Web	R203
<pre>--Search by language SELECT auction.*, image.source FROM auction, language, image WHERE auction.idLanguage = language.id AND language.language = \$langName AND image.idAuction = auction.idAuction; --OR --Search by fields SELECT auction.&#42;, image.source FROM auction, image WHERE isbn = \$isbn OR author = \$author OR title = \$title;</pre>	

SQL202	Obter a informação completa de um leilão
Recurso Web	R204
<pre>SELECT auction.*, member.username FROM auction, member WHERE auction.id = \$auctionID AND idSeller = member.id; SELECT source FROM image WHERE idAuction = \$auctionID;</pre>	

SQL203	Obter o valor atual das licitações num leilão
Recurso Web	R210
<pre>SELECT max(bidValue) FROM bid WHERE idAuction = \$auctionID;</pre>	

SQL204	Licitar num leilão
Recurso Web	R209
<pre>UPDATE bid SET bidValue = \$newValue, bidDate = now() WHERE idBuyer = \$userID AND idAuction = \$auctionID;</pre>	

M04: Comunicação

SQL401	Obter notificações não lidas
Recurso Web	R402
SELECT notifications.* FROM notifications WHERE idMember = \$userID AND is_seen = false;	

SQL402	Marcar uma notificação como lida
Recurso Web	R403
UPDATE notification SET is_seen = TRUE, dateSeen = now() WHERE idMember = \$userID;	

2. Transações

T01	Criar um novo utilizador
Recurso Web	R105
Justificação	De modo a manter a consistência da base de dados, é necessário que todo o código execute sem erros e que ambos os tuplos sejam inseridos, sendo que se provoca um <i>Rollback</i> caso a inserção em pelo menos uma das tabelas falhe (falhas essas que podem acontecer se, por exemplo, os atributos inseridos violarem uma ou mais restrições). O nível de isolamento é <i>READ COMMITTED</i> visto que não queremos que outra transação leia o tuplo inserido em <i>users</i> sem este ter o tuplo em <i>image</i> a ele associado também inserido e <i>committed</i> (ou seja, não queremos <i>dirty reads</i>).
Nível de isolamento	READ COMMITTED

```
BEGIN TRANSACTION; SET TRANSACTION ISOLATION LEVEL READ COMMITTED
-- Insert users INSERT INTO users (address, age, email, name, password, phone, postalCode, username, idCountry)
VALUES ($address, $age, $email, $name, $password, $phone, $postalCode, $username, $idCountry);

-- Insert image INSERT INTO image (source, idusers) VALUES ($source, $idusers);

COMMIT;
```

T02	Terminar uma conta
Recurso Web	R504
Justificação	De modo a manter a consistência da base de dados, é necessário que todo o código execute sem erros e que ambos os tuplos sejam inseridos, sendo que se provoca um <i>Rollback</i> caso a inserção em pelo menos uma das tabelas falhe. O nível de isolamento é <i>READ UNCOMMITTED</i> visto que a camada lógica de negócio garante que não há problemas caso sejam lidos dados inconsistentes enquanto a transação decorre. Esta garantia é dada pelo facto de existir apenas um único administrador.
Nível de isolamento	READ UNCOMMITTED

```
BEGIN TRANSACTION; SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED
-- Set user as terminated UPDATE users SET users_status = 'terminated' WHERE id = $userID;

-- Delete user profile image DELETE FROM requested_termination WHERE idusers = $idRequest;

COMMIT;
```

T03	Criar um novo leilão
Recurso Web	R206

Justificação	De modo a manter a consistência da base de dados, é necessário que todo o código execute sem erros e que ambos os tuplos sejam inseridos, sendo que se provoca um <i>Rollback</i> caso a inserção em pelo menos uma das tabelas falhe. O nível de isolamento é <i>SERIALIZABLE</i> porque não queremos permitir <i>queries</i> do tipo SELECT...WHERE (tais como a <i>query</i> SQL201, referente à pesquisa de leilões usando vários parâmetros) enquanto esta transação não estiver totalmente concluída. Isto evita a ocorrência de <i>phantom reads</i> durante essas pesquisas.
Nível de isolamento	SERIALIZABLE
<pre> BEGIN TRANSACTION; SET TRANSACTION ISOLATION LEVEL SERIALIZABLE -- Insert auction INSERT INTO auction (author, description, duration, ISBN, title, idPublisher, idLanguage, idSeller) VALUES (\$author, \$description, \$duration, \$ISBN, \$title, \$idPublisher, \$idLanguage, \$userID); -- Insert image INSERT INTO image (source, idAuction) VALUES (\$source, \$idAuction); COMMIT;</pre>	

T04	Criar um pedido de edição de leilão
Recurso Web	R208
Justificação	De modo a manter a consistência da base de dados, é necessário que todo o código execute sem erros e que ambos os tuplos sejam inseridos, sendo que se provoca um <i>Rollback</i> caso a inserção em pelo menos uma das tabelas falhe. O nível de isolamento é <i>READ COMMITTED</i> visto que não queremos que outra transação, como por exemplo a listagem de pedidos por parte de um moderador, leia o tuplo inserido em <i>auction_modification</i> sem este ter o tuplo em <i>image</i> a ele associado também inserido e <i>committed</i> (ou seja, não queremos <i>dirty reads</i>).
Nível de isolamento	READ COMMITTED
<pre> BEGIN TRANSACTION; SET TRANSACTION ISOLATION LEVEL READ COMMITTED -- Insert auction INSERT INTO auction_modification (newDescription, idApprovedAuction) VALUES (\$newDescription, \$idApprovedAuction); -- Insert image INSERT INTO image (source, idAuctionModification) VALUES (\$source, \$idAuctionModification); COMMIT;</pre>	

Histórico de revisões:

- Foi corrigido o SQL do SQL101;
- Removidos os acessos triviais dos módulos 3, 5 e 6;
- Adicionadas referências para os recursos Web que usam cada transação;
- Justificações dos níveis de isolamento melhoradas usando exemplos concretos das situações em que problemas de concorrência podem acontecer

GROUP1726, 18/4/2018

- Daniel Vieira Azevedo, up201000307@fe.up.pt
- Nelson André Garrido da Costa, up201403128@fe.up.pt
- Rúben José da Silva Torres, up201405612@fe.up.pt
- Tiago Lascasas dos Santos, up201503616@fe.up.pt