

A11: Transações e acessos principais à base de dados

Neste artefacto serão apresentados os principais acessos à base de dados, incluindo as transações. Apesar de existirem outros acessos, os que se apresentam neste artefacto são aqueles distinguidos como importantes e usados com muita frequência. Para cada acesso SQLxxx identificado no A9 será apresentado o respetivo código SQL utilizado para o efeito. Os acessos que estão relacionados são incluídos todos na mesma secção.

Módulo M01: Autenticação, Pesquisa e Notificações

SQL102

Autenticação no sistema dado o nome de utilizador e a *password*.

```
SELECT *  
FROM Membro  
WHERE nomeUtilizador = :username  
AND password = :password;
```

SQL103

Registo no sistema através de *input* de informação da parte do utilizador.

```
INSERT INTO Membro  
(nomeUtilizador, email, password, dataNascimento, tipoMembro)  
VALUES  
(:username, :e-mail, :password, :birthday, 'Cliente');
```

SQL106 e SQL107

Pesquisa em leilões através do *input* dado pelo utilizador. Faz uso da capacidade de *full-text search* do *PostgreSQL* para fazer pesquisas rápidas. O SQL106 e o SQL107 são idênticos, sendo aplicados em sítios diferentes (o SQL106 na página principal a partir da barra superior e o SQL107 na página de pesquisa).

```
SELECT *  
FROM Leilao  
WHERE to_tsvector('portuguese', descricao) @@ to_tsquery('portuguese',  
:INPUT) OR nome LIKE ':input';
```

SQL108

Selecciona todas as notificações do utilizador.

```
SELECT texto, datanotificacao
FROM notificacao
WHERE idutilizador = :idutilizador;
```

Módulo M02: Membros

SQL201

Selecciona todas as preferências do utilizador.

```
SELECT Marca.nome
FROM Preferencia, Marca
WHERE Preferencia.idMarca = Marca.idMarca
AND Preferencia.idCliente = :idCliente;
```

SQL202

Altera as preferências do utilizador. É usado uma transação porque são substituídas as preferências antigas do Utilizador (apagando-as) pelas novas (inserindo-as). O facto de existir um write e um read a uma tabela e de forma a impedir todos os problemas de concorrência é usado o melhor isolamento (*SERIALIZABLE*).

```
BEGIN;
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
DELETE
FROM Preferencia
WHERE idCliente = :idCliente;

// para cada Marca

INSERT INTO Preferencia(idCliente,idMarca)
VALUES(:idCliente, (SELECT idMarca FROM Marca WHERE nome = :nome));
COMMIT;
```

SQL203

Pesquisa os leilões criados pelo utilizador usando essa informação para ir buscar uma das imagens de cada um. Não foi associada uma transição devido ao facto de existir a possibilidade do leilão não ter imagens associadas.

```
SELECT nome, licitacaoAtual, dataColocacao, duracao
```

```
FROM Leilao
WHERE idLeiloeiro = :idLeiloeiro;

SELECT link
FROM Imagem
WHERE idLeilao = :idLeilao;
```

SQL204

Selecciona todas as licitações que o utilizador realizou pelos diversos leilões.

```
SELECT Leilao.nome, Licitacao.valor, Leilao.licitacaoAtual
FROM Leilao, Licitacao
WHERE Licitacao.idCliente = :idClientte
AND Licitacao.idLeilao = Leilao.idLeilao;
```

SQL206

Selecciona todas as mensagens que o utilizador recebeu e enviou. É usada uma transição *READ ONLY* para aumentar o desempenho global do sistema.

```
BEGIN;
SET TRANSACTION READ ONLY
SELECT Membro.nomeUtilizador, Mensagem.texto, Mensagem.dataMensagem
FROM Membro, Mensagem
WHERE Mensagem.idEmissor = :idEmissor
AND Membro.idUtilizador = Mensagem.idRecetor;

SELECT Membro.nomeUtilizador, Mensagem.texto, Mensagem.dataMensagem
FROM Membro, Mensagem
WHERE Mensagem.idReceptor = :idRecetor
AND Membro.idUtilizador = Mensagem.idEmissor;
COMMIT;
```

SQL208

Insere uma mensagem criada pelo utilizador na base de dados.

```
INSERT INTO Mensagem
(idEmissor, idRecetor, texto, dataMensagem)
VALUES
(:idEmissor, :idRecetor, :texto, :dataMensagem);
```

SQL209

Apaga uma mensagem a partir do seu id.

```
DELETE
FROM Mensagem
WHERE Mensagem.idMensagem = :idMensagem;
```

Módulo M03: Leilões

SQL302

Esta transação é utilizada para criar um leilão. A criação de um leilão envolve, além da criação do leilão em si, as imagens que definem esse leilão (pelo menos uma, num máximo de 10). De forma a garantir a integridade na base de dados de todos os itens adicionados pelo utilizador, é usado uma transação *READ COMMITTED* que é o isolamento *DEFAULT* do *POSTGRESQL*. O isolamento escolhido baseou-se no facto de poderem ocorrer várias instruções deste género, ao mesmo tempo, sem afectar a integridade da base de dados no momento do *COMMIT*.

```
BEGIN;
INSERT INTO Leilao(idLeiloeiro, nome, descricao, duracao, licitacaoBase)
VALUES(:idUtilizador, :nome, :descricao, :duracao, :licitacaoBase);

// para cada imagem

INSERT INTO Imagem(idLeilao, linkImagem)
VALUES(:idLeilao, :linkImagem);
COMMIT;
```

SQL303

Carrega as informações de um leilão para que este possa ser visualizado e editado.

```
SELECT Leilao.nome, Leilao.descricao, Leilao.licitacaoBase, Leilao.idLeilao
FROM Leilao
WHERE Leilao.idLeilao = :idLeilao;
```

SQL305

Faz *update* na descrição de um leilão.

```
UPDATE Leilao
SET descricao = :descricao
WHERE (idleilao = :idLeilao);
```

SQL306

Adiciona uma imagem na base de dados para um certo leilão (este *insert* distingue-se do processo de criação uma vez que um utilizador pode inserir imagens após a criação do leilão, desde que o limite de 10 imagens seja respeitado).

```
INSERT INTO Imagem
(idLeilao, link)
VALUES
(:idLeilao, :link);
```

SQL307

Apaga uma imagem da base de dados. Esta imagem deixa de estar disponível para o leilão.

```
DELETE FROM Imagem
WHERE Imagem.idLeilao = :idLeilao
AND Imagem.link = :link;
```

SQL308

Adiciona uma licitação para um determinado leilão.

```
INSERT INTO Licitacao(idLeilao, idCliente, valor)
VALUES (:idLeilao, :idUtilizador, :licitacao);
```

Módulo M04: Feedback

SQL401 e SQL402 e SQL403

Dado um *idUtilizador* é seleccionado o seu nome de Utilizador no caso do primeiro *select*. No segundo e o terceiro seleccionam, respectivamente, o Feedback que o Utilizador deu enquanto Comprador e o Feedback que o Utilizador deu enquanto Leiloeiro. Os SQL referidos são idênticos sendo que é usado, respetivamente, o primeiro *select* por todos, o segundo pelo SQL401 e SQL403 e o terceiro pelo SQL401 e SQL402.

```
SELECT nomeUtilizador
FROM Membro
WHERE idUtilizador = :idUtilizador;

SELECT Feedback.texto, Feedback.dataFeedback, Feedback.valor
FROM Feedback, FeedbackComprador
WHERE FeedbackComprador.idComprador = :idComprador
AND Feedback.idFeedback = FeedbackComprador.idFeedback;
```

```
SELECT Feedback.texto, Feedback.dataFeedback, Feedback.valor
FROM Feedback, FeedbackLeiloeiro
WHERE FeedbackLeiloeiro.idLeiloeiro = :idLeiloeiro
AND Feedback.idFeedback = FeedbackLeiloeiro.idFeedback;
```

SQL404

Dado um *idUtilizador* é seleccionado o seu nome de Utilizador no caso do primeiro *select*. No segundo é seleccionado o nome dos leilões com o *feedback* respetivo que o utilizador deu.

```
SELECT nomeUtilizador
FROM Membro
WHERE idUtilizador = :idUtilizador;

SELECT Feedback.texto, Feedback.dataFeedback, Feedback.valor
FROM Leilao, Feedback
WHERE (Leilao.idFeedbackLeiloeiro = :idFeedback
OR Leilao.idFeedbackCliente = :idFeedback)
AND Feedback.idFeedback = :idFeedback;
```

SQL406

Esta transação existe para determinar a tabela onde o *feedback* irá ser colocado, dependendo do *role* atual do utilizador (se é leiloeiro e portanto dono de um leilão, ou comprador). É necessário uma transação para garantir que a base de dados só é alterada caso ao inserir o texto e o valor na tabela geral de *feedback*, fazer *update* no leilão dos respetivos *ids* do leiloeiro e do comprador, e adicionar a relação da tabela de *feedback* e do Utilizador, não apresentar nenhum erro durante a sua execução. O isolamento desta transição é o mais restrito possível para impedir todos os problemas de concorrência.

```
BEGIN;
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
INSERT INTO Feedback
(texto, dataFeedback, valor)
VALUES
(:texto, :dataFeedback, :valor);

// IF ROLE IS 'Leiloeiro'

UPDATE Leilao
SET idFeedbackLeiloeiro =
(SELECT idFeedback
FROM Feedback
WHERE texto = :texto
AND dataFeedback = :dataFeedback
AND valor = :valor
)
WHERE idLeilao = :idLeilao;
```

```
INSERT INTO FeedbackLeiloeiro
(idFeedback, idLeiloeiro)
VALUES
((
    SELECT idFeedback
    FROM Feedback
    WHERE texto = :texto
    AND dataFeedback = :dataFeedback
    AND valor = :valor
), (
    SELECT idLeiloeiro
    FROM Leilao
    WHERE idLeilao = :idLeilao
));

// IF ROLE IS 'Comprador'

UPDATE Leilao
SET idFeedbackCliente =
    (SELECT idFeedback
    FROM Feedback
    WHERE texto = :texto
    AND dataFeedback = :dataFeedback
    AND valor = :valor
    )
WHERE idLeilao = :idLeilao;

INSERT INTO FeedbackComprador
(idFeedback, idComprador)
VALUES
((
    SELECT idFeedback
    FROM Feedback
    WHERE texto = :texto
    AND dataFeedback = :dataFeedback
    AND valor = :valor
), (
    SELECT Licitacao.idCliente
    FROM Licitacao, Leilao
    WHERE Leilao.idLeilao = :idLeilao
    AND Leilao.idLeilao = Licitacao.idLeilao
    AND Licitacao.valor = Leilao.LicitacaoAtual
    ));
COMMIT;
```

Módulo M05: Administração

SQL501

Um *admin* tem acesso a uma lista de todos os utilizadores do *site* pelo seu nome.

```
SELECT nomeUtilizador  
FROM Membro;
```

SQL502

Insere um utilizador banido na tabela *MembroBanido* sem o retirar da tabela *Membro*.

```
INSERT INTO MembroBanido  
(idMembroBanido, dataBanido, duracao, motivo)  
VALUES  
(:idMembroBanido, :dataBanido, :duracao, :motivo);
```

SQL503

Torna um utilizador como *admin* fazendo *update* à sua condição.

```
UPDATE Membro  
SET tipomembro = 'Admin'  
WHERE idUtilizador = :idUtilizador;
```

SQL505

Apaga um leilão da base de dados fazendo *update* ao seu estado.

```
UPDATE Leilao  
SET estadoleilao = 'delete'  
WHERE idLeilao = :idLeilao;
```

SQL506

Congela um leilão na base de dados (impedindo novas licitações e paralisando o tempo restante) fazendo *update* ao seu estado.

```
UPDATE Leilao  
SET estadoleilao = 'freeze'  
WHERE idLeilao = :idLeilao;
```

SQL507

Pesquisa de utilizadores de um *admin* a partir do seu nome.

```
SELECT nomeUtilizador  
FROM Membro  
WHERE nomeUtilizador = :nomeUtilizador;
```

— *AutoLeilões, Lda.*

[\[Voltar à página principal\]](#)

From:

<http://lbaw.fe.up.pt/201516/> - **L B A W :: WORK**

Permanent link:

<http://lbaw.fe.up.pt/201516/doku.php/lbaw1512/proj/a11>

Last update: **2016/05/19 18:51**

