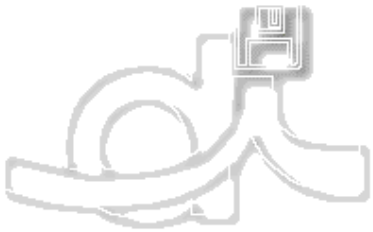


Programação com ODBC 3

Ambientes de Desenvolvimento Avançados

Engenharia Informática
Instituto Superior de Engenharia do Porto

Alexandre Bragança
1998/99

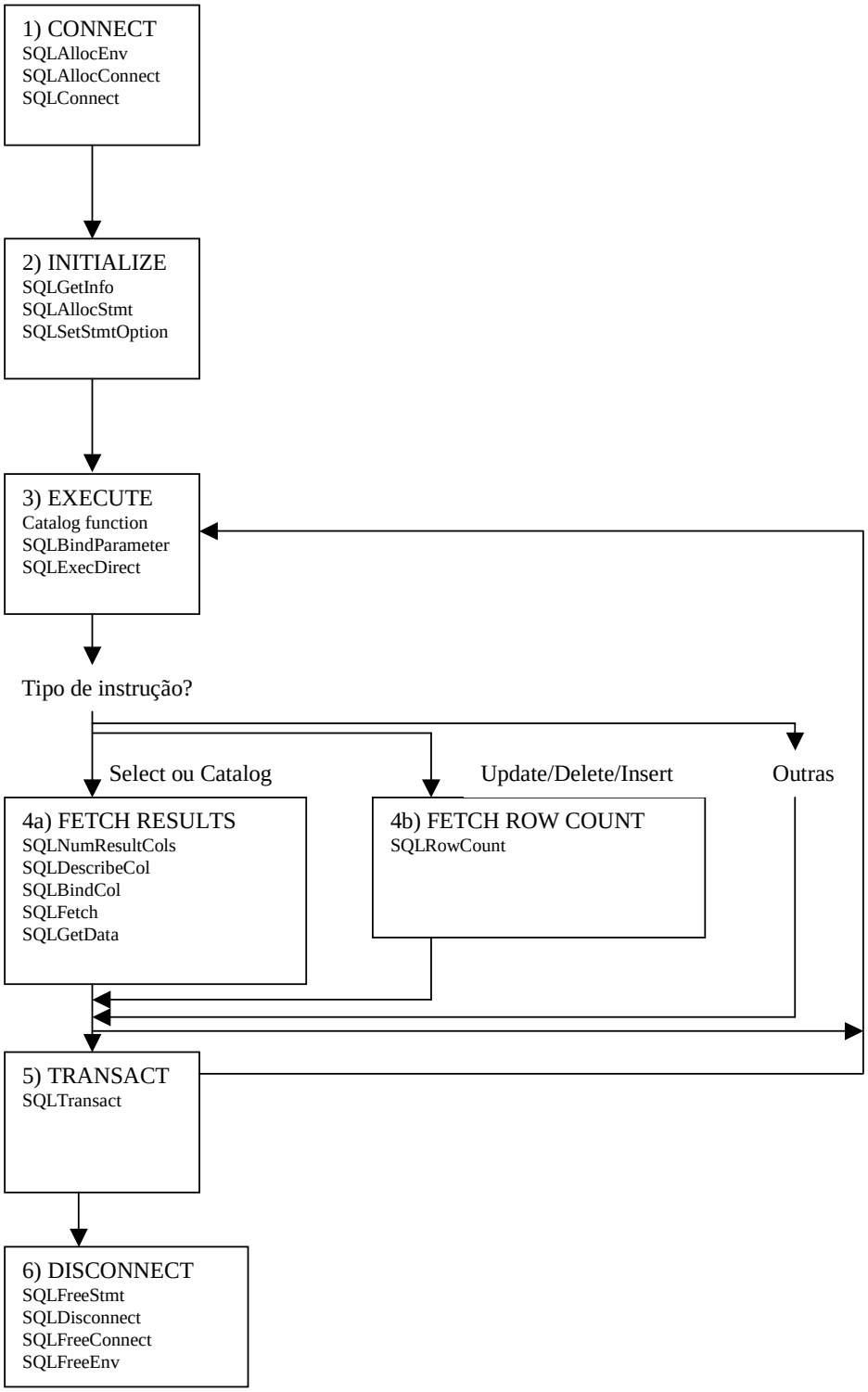
3 Programação com ODBC

3.1 Estrutura de uma aplicação que utilize ODBC

3.2 Níveis de conformidade do ODBC

3.3 Classes para Manipular BD através do ODBC

3.1 Estrutura de uma aplicação que utilize ODBC



- **Primeiro passo**
  - Inclui a ligação à fonte dos dados (data source).
  - A função **SQLAllocEnv** permite carregar o drive manager assim como dar inicio à utilização das funções do ODBC.
  - A função **SQLSetConnectOption** permite inicializar algumas opções globais para a connection, tais como a forma de commit. Se o commit for automatico então o ODBC confirma uma transacção automaticamente após a execução de cada statement. Se o commit for manual é necessário chamar a função SQLTransact para fazer commit ou rool back.

- **Segundo passo**
  - Neste passo normalemente as aplicações chamam o **SQLGetInfo** para obterem informações sobre as capacidades do driver que estão a utilizar.
  - Para executar uma instrução SQL é necessário antes efectuar a reserva de um statement através da função **SQLAllocStmt**.
  - A instrução **SQLSetStmtOption** permite especificar configurações possíveis para o statement, como por exemplo o tipo de cursor ou a forma de gestão de concorrência (locks…).

- **Terceiro passo**
  - Normalmente inclui a 'construção' e execução de um comando SQL.
  - Se o comando SQL tiver parametros estes podem ser ligados a variáveis do programa através da função **SQLBindParameter**.
  - A instrução SQL é executada com a função **SQLExecDirect**.
  - Se desejarmos que a instrução possa ser executada várias vezes é necessário prepará-la com **SQLPrepare** e só depois executá-la com o **SQLExecute**.

- **Quarto passo (a)**
  - Se a instrução é do tipo SELECT ou é uma função de Catalog então a função **SQLNumResultCols** permite obter o número de colunas resultantes no result set.
  - O **SQLDescribeCol** permite obter a descrição de cada coluna resultante: nome, tipo de dado, precisão(número máximo de digitos) e escala(número máximo de digitos à direita do ponto decimal).
  - O **SQLBindCol** permite ligar uma coluna a uma variável do programa.
  - O **SQLFetch** retorna o valor de um registo para as variáveis do programa indicadas no SQLBindCol.
  - O **SQLGetData** permite ir buscar valores para os quais não foi efectuado o bind ou valores de colunas com dimensão possivelmente elevada (ex: campos binários…).

- **Quarto passo (b)**
  - No caso do comando de SQL for do tipo INSERT, DELETE ou INSERT então como o resultado não é um result set o que se deve fazer é verificar o número de registos afectado com **SQLRowCount**.

Nesta altura a aplicação volta ao passo 3 para executar outra instrução na mesma transacção ou vai para o passo 5 confirmar ou fazer desfazer (rool back) a transacção.

- **Quinto passo**
  - A aplicação confirma (ou desfaz) a transacção com **SQLTransact**. Só necessita de fazer isso no caso de ter optado por confirmação de transacção manual no primeiro passo.
  - A aplicação pode então seguir para o passo 3 de forma a executar outro comando SQL ou seguir para o passo 6 e terminar o processamento.

- **Sexto passo**
  - Finalmente a aplicação termina o processamento do ODBC.
  - Primeiro liberta-se o statement com **SQLFreeStmt**.
  - De seguida 'desliga-se' da data source com **SQLDisconnect**.
  - Finalmente liberta a connection e o environment com **SQLFreeConnect** e **SQLFreeEnv**.

3.2 Níveis de conformidade do ODBC

3.2.1 Conformidade da API

- A API do ODBC está dividida em 3 níveis. As funções nucleares incluiem todas as funções definidas pela especificação X/Open e SQL Access Group. Para além destas funções o ODBC define ainda dois níveis superiores.

- **Core API**
  - Reservar e libertar handles (connection, environment, statement)
  - Ligação a fontes de dados. Utilizar vários statements numa ligação.
  - Preparar e executar instuções SQL. Executar de forma imediata instruções SQL.
  - Atribuir áreas de armazenamento para parametros e colunas de resultado para as instruções SQL.
  - Obter os dados de um result set. Obter informação sobre um result set.
  - Commit, roll back sobre transacções.
  - Obter informação sobre possíveis erros.

- **Level 1 API**
  - Toda a funcionalidade da core API.
  - Ligação a fontes de dados utilizando ecrãs de diálogo especificos do driver.
  - Obter e alterar valores de opções para statement e connection.
  - Enviar uma parte ou todo o valor associado a um parametro (útil para tipos de dados longos).
  - Obter parte ou o total de um valor de uma coluna resultante (útil para tipos de dados longos).
  - Obter informação de catalogo (colunas, colunas especiais, tabelas, etc.).
  - Obter informação acerca das características e capacidades do driver e da fonte de dados (tipos de dados suportados, funções SQL, funções do ODBC, etc.).

- **Level 2 API**
  - Toda a funcionalidade do nivel 1 e core.
  - Poder mostrar a informação da ligação e listar as fontes de dados.
  - Enviar vectores de valores de parametros. Obter vectores de valores de colunas resultantes.
  - Obter o número de parametros e descrever cada parametro.
  - Usar cursores do tipo 'scrollable'.
  - Obter a forma nativa de uma instrução SQL.
  - Obter informação de catalogo (privilégios, chaves, procedimentos, etc.).
  - Chamar uma Dynamic Link Library de tradução.

3.2.2 Conformidade do SQL

- **Minimum SQL Grammar**
  - Data Definition Language (DDL): CREATE TABLE e DROP TABLE.
  - Data Manipulation Language (DML): SELECT simples, INSERT, UPDATE SEARCHED e DELETE SEARCHED.
  - Expressões: simples (ex: A > A + C).
  - Tipos de dados: CHAR, VARCHAR ou LONG VARCHAR.

- **Core SQL Grammar**
  - Gramática e tipos de dados do SQL minimos.
  - DDL: ALTER TABLE, CREATE INDEX, DROP INDEX, CREATE VIEW, DROP VIEW, GRANT e REVOKE.
  - DML: SELECT completo.
  - Expressões: subquery, funções de conjunto como SUM e MIN.
  - Tipos de dados: DECIMAL, NUMERIC, SMALLINT, INTEGER, REAL, FLOAT, DOUBLE PRECISION.

- **Extended SQL Grammar**
  - Gramática e tipos de dados do SQL minimo e core.
  - DML: outer joins, positioned UPDATE, positioned DELETE, SELECT FOR UPDATE e uniões.
  - Expressões: funções escalares como SUBSTRING E ABS, literais de tempo, datas e timestamp.
  - Tipos de dados: BIT, TINYINT, BIGINT, BINARY, VARBINARY, LONG VARBINARY, DATE, TIME, TIMESTAMP.
  - Instruções de SQL em lote.
  - Chamadas a procedimentos.