



embarcadero®

embarcadero
conference

FireDAC

Guinther Pauli - MVP

embarcadero®
MVP

#delphi20anos



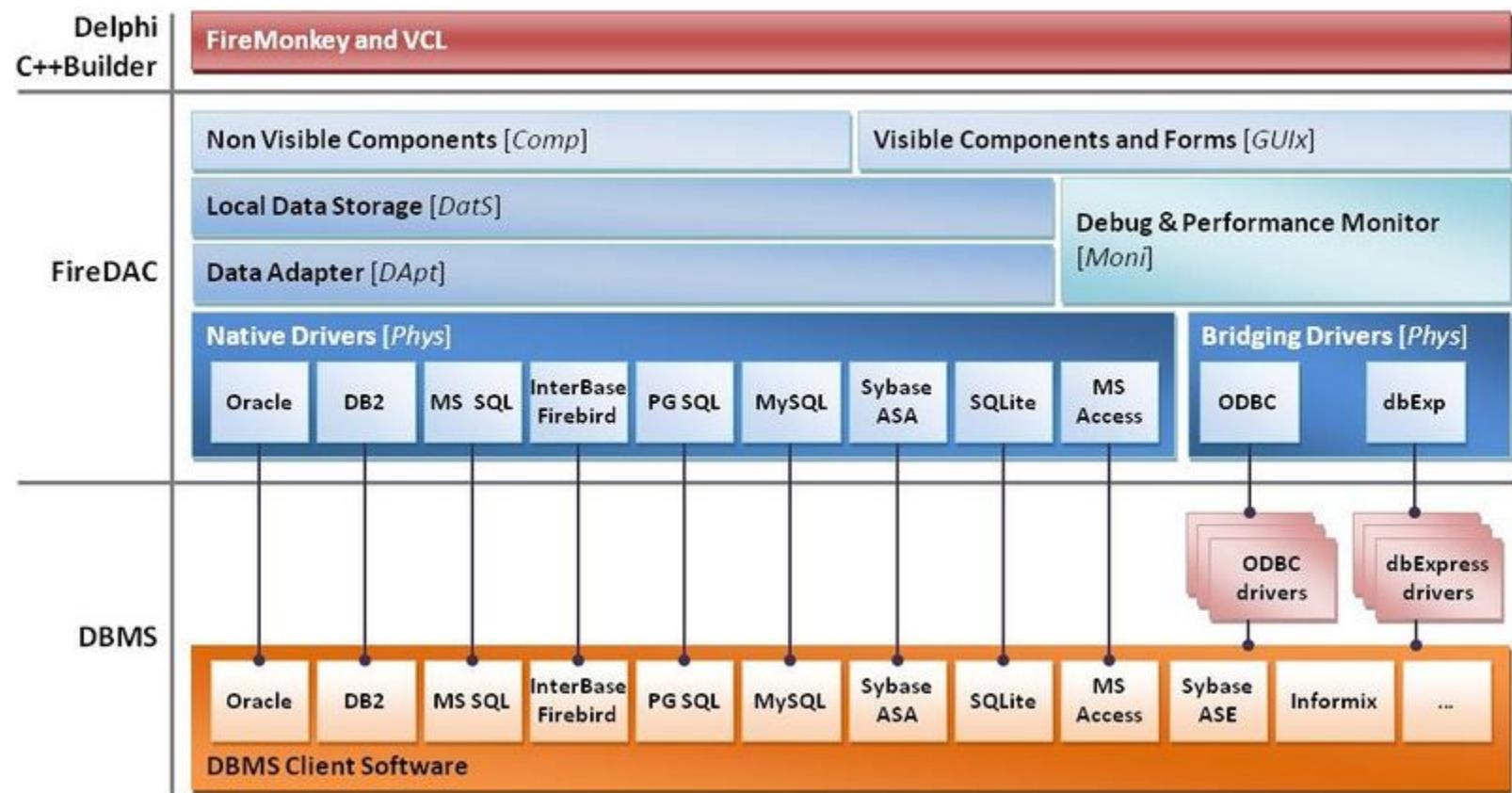
Agenda

- Recursos intermediários, avançados e segredos do FireDAC
- Recursos semelhantes do FireDAC com DBX + CDS + DSP
(como fazer no FireDAC o que já sei fazer no CDS)
- Recursos do FireDAC que não possuem equivalência no DBX + CDS + DSP
- Recursos onde o FireDAC é melhor que o DBX + CDS + DSP
- ETL



FireDAC

Arquitetura



FireDAC

- Um conjunto de componentes universais de acesso a dados
- Desenvolvimento para qualquer banco de dados
- Delphi e C++Builder
- Alta performance, fácil de usar, conexão com bancos locais e corporativos
- Acesso a dados universal, porém com características específicas de cada BD
- Baseado em classes + componentes TDataSet

Recovering Connection

- Ambiente instável, rede ou DBMS pode cair
- Aplicativo precisa recuperar a conexão e continuar a comunicação
- FireDAC pode recuperar automaticamente a conexão
- Detecta quando uma conexão foi perdida, permite dar uma resposta
- FDConnection.ResourceOptions.AutoReconnect := true;
- Erro só ocorre ao executar um Open, ExecSQL, Ping por ex., então a exceção EFDDDBEngineException é levantada com tipo ekServerGone

```
procedure TFrmRecoveringConnection.conRecover(
  ASender, AInitiator: TObject; AException: Exception;
  var AAction: TFDPhysConnectionRecoverAction);
var
  res: Integer;
begin
  res := MessageDlg(
    'Conexão foi perdida. Offline = Yes, Retry = Ok, Fail = Cancel',
    mtConfirmation, [mbYes, mbOK, mbCancel], 0);
  case res of
    mrYes:   AAction := faOfflineAbort;
    mrOk:    AAction := faRetry;
    mrCancel: AAction := faFail;
  end;
end;
```

Offlining Connection

- Lembra por que muitos usavam BDE sem Data Controls?
- Semelhante a um cliente multi-camadas (CDS, KeepConnection etc)
- Cliente está desconectado, ativa só quando precisa trocar dados
- Útil em ambiente instável, preservar recursos do BD
- Conexão está fechada, mas os datasets ficam abertos
- FDConnection.Offline();
- Recuperar conexão com AutoReconnect ou OnLine();

```
qry.SQL.Text := 'select * from EMPLOYEE';
qry.Open();
qry.FetchAll();
DM.con.Offline();
DM.con.Close();
// Dados ainda podem ser visualizados e alterados
```

SQL Dialect Abstraction

- Se o aplicativo precisa suportar múltiplos SGBDs, deve estar ciente de que seus dialetos SQL podem ser diferentes
- FireDAC permite que você escreva os comandos SQL independente de dialeto
- Semelhante a diretivas de compilação IFs do Pascal
- Mais fácil criar aplicações multi-banco, sem perder recursos específicos de cada um (usando SQL “genérico”) ou uma API, componente de terceiro etc.

```
{ IF FIREBIRD}  
SELECT RDB$RELATION_NAME FROM RDB$RELATIONS  
{ fi }  
{ IF MSSQL}  
SELECT NAME FROM SYS.OBJECTS  
{ fi }
```

Substitution Variables e Macros

- Qualquer coisa pode ser variável (parâmetro) no seu SQL, estamos acostumados a usar parâmetros em where por exemplo
- Pense em qualquer coisa como um parâmetro, de strings, comandos a objetos, como por exemplo, o próprio nome da tabela
- Ex.: Isso não pode ser feito com DBX “select * from :TABELA”
- No FireDAC, você pode usar variáveis de substituição (MACROS), como um processador de templates, e depois substituir o texto da variável, usando os caracteres “!” (string) e “&” (SQL)
- Lembra dos HTMLTags dos PageProducers do WebBroker?

```
var
  tabela: string;
begin
  tabela := boxTableName.Text; // ex. EMPLOYEE
  qry.SQL.Text := 'SELECT * FROM &TABELA';
  qry.MacroByName('TABELA').AsRaw := tabela;
  qry.Open();
```

Array DML

- Permite enviar um único comando para o DBMS com um conjunto de parâmetros (array)
- Cada parâmetro de comando tem uma matriz de valores
- Todos os parâmetros têm matrizes do mesmo comprimento
- FireDAC solicita o DBMS para executar um comando uma vez para cada linha do array
- Reduz a quantidade de comunicação entre DBMS e cliente
- Acelera o tempo de execução



```
var
  i: integer;
begin
  qry.Params.ArraySize := 100;
  qry.SQL.Text := 'insert into TESTE_ARRAY values (:ID,:TEXTO)';
  // configurar parâmetros aqui ou no DFM / wizard
  for i := 0 to 99 do
    begin
      qry.ParamByName('ID').AsIntegers[i] := i;
      qry.ParamByName('TEXTO').AsStrings[i] := 'TEXTO' + i.ToString();
    end;
  qry.Execute(100);
```

Executing SQL Scripts

- Script = conjunto de comandos SQL executados separadamente
- Manutenção, criação, versionamento, carga inicial de BD
- TFDScript vs. TFDQuery.ExecSQL
- Pode ser dividida em várias transações (lote é uma única)
- Totalmente controlado pelo cliente (lote controlado pelo BD)
- A execução do script pode ser colocada em log
- o script fornece feedback progresso execução
- FireDAC tem componente pronto para feedback (TFDGUIxScriptDialog)



```
var
  SQL: TFDSQLScript;
begin
  Script.SQLScripts.Clear();
  SQL := Script.SQLScripts.Add();
  SQL.SQL.Add('create table CLIENTES( ' +
    ' ID integer not null primary key, NOME varchar(100));');
  SQL.SQL.Add('insert into CLIENTES values(1, ''GUINTHER'');');
  SQL.SQL.Add('insert into CLIENTES values(2, ''RUDOLFO'');');
  SQL.SQL.Add('insert into CLIENTES values(3, ''FERNANDO'');');
  SQL.SQL.Add('insert into CLIENTES values(4, ''FABIO'');');
  Script.ValidateAll();
  Script.ExecuteAll();
```



FireDAC Processing



Please wait, application is processing SQL script

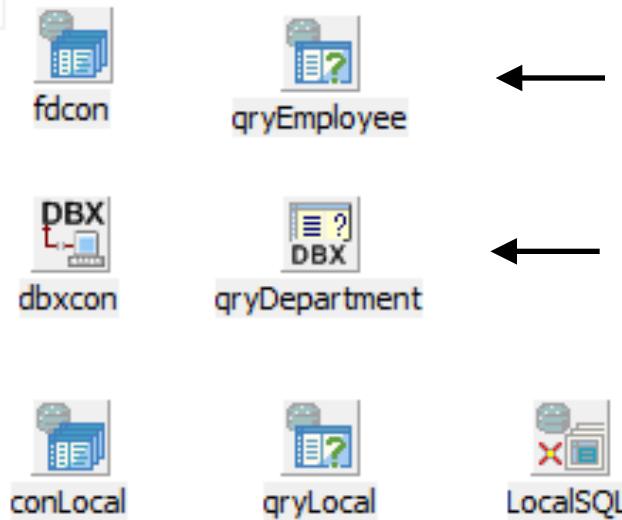
(line 6)

Total script size: 271 bytes Total % done: 84,8 %
Total processed: 230 bytes Total errors: 0

```
drop table CLIENTES;
Ok [00:00:00.015].
create table CLIENTES( ID integer not null primar ...
Ok [00:00:00.031].
insert into CLIENTES values(1,'GUINTHER');
Ok [00:00:00.016].
insert into CLIENTES values(2,'RUDOLFO');
Ok [00:00:00.016].
insert into CLIENTES values(3,'FERNANDO');
Ok [00:00:00.016].
```

Local SQL

- Permite a execução de comandos SQL, onde os descendentes de TDataSet são utilizadas em vez das tabelas de BD
- Usa driver do SQLite por baixo
- TFDLocalSQL contém a coleção de todos os TDatasets
- Dados de diferentes BDs usando diferentes engines (BDE, DBX, ADO)
- Consultas heterogêneas, in-memory database, modo off-line avançado
- Pode ser usado para migração de BD
- Pode ser usado de FireDAC para FireDAC, esqueça o “Filter”!



```

SELECT E.dept_no,
       E.first_name,
       E.last_name,
       D.department
  FROM employee E
 INNER JOIN department D
    ON E.dept_no = D.dept_no
 WHERE first_name LIKE '%A%'
 ORDER BY E.first_name
  
```

The screenshot shows the "Local SQL" window displaying a grid of data from a local dataset. The columns are labeled: DEPT_NO, FIRST_NAME, LAST_NAME, and DEPARTMENT. The data includes 14 rows of employee information from the Northwind database.

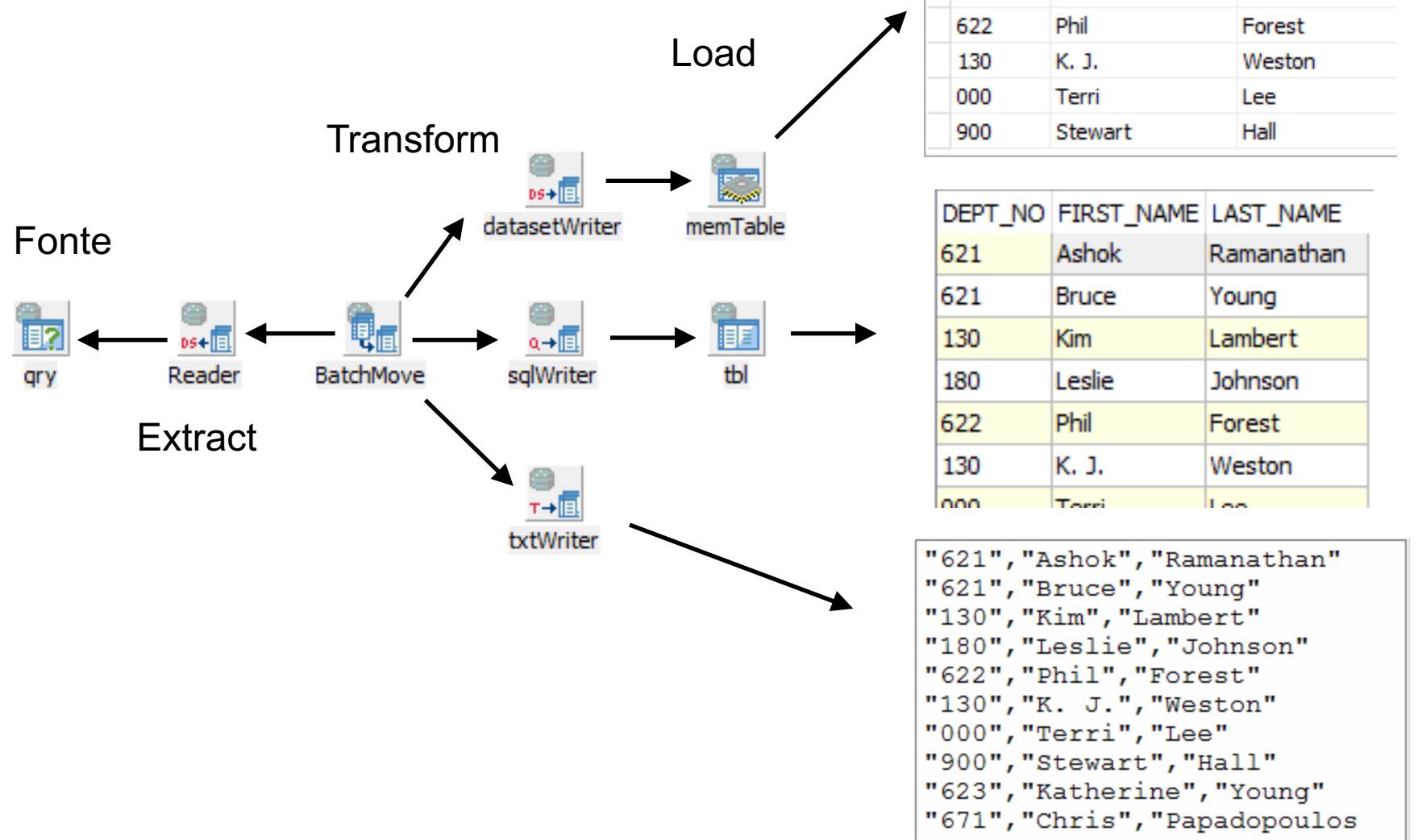
DEPT_NO	FIRST_NAME	LAST_NAME	DEPARTMENT
120	Ann	Bennet	European Headquarters
621	Ashok	Ramanathan	Software Development
180	Carol	Nordstrom	Marketing
140	Claudia	Sutherland	Field Office: Canada
621	Dana	Bishop	Software Development
123	Jacques	Glon	Field Office: France
110	Janet	Baldwin	Pacific Rim Headquarters
623	Katherine	Young	Customer Support
671	Mary	Page	Research and Development
100	Mary S.	MacDonald	Sales and Marketing
100	Michael	Yanowski	Sales and Marketing
672	Randy	Williams	Customer Services
900	Stewart	Hall	Finance

Query local, trazendo dados de “tabelas” (TDataSets) locais, fazendo joins, filtros, order, where etc, tudo localmente em cache, usando *diferentes engines de acesso como se fossem fontes de dados*

ETL – Extract, Transform, Load

- Extract – extrair dados de fontes homogêneas ou heterogêneas
- Transform – transformar os dados para armazenar em um formato ou estrutura específica para propósitos de análise
- Load – armazenar no BD final (BD relacional, data warehouse)
- FONTE / DESTINO: TXT, DataSet (qualquer engine / BD), FireDAC (SQL)
- Exemplos de uso: extrair dados de um BD e exportar para TXT, ou ler de um arquivo TXT e armazenar em uma tabela do BD (carga), transferir dados de um dataset para outro (engines diferentes)
- Lembra do Batch Move do BDE?





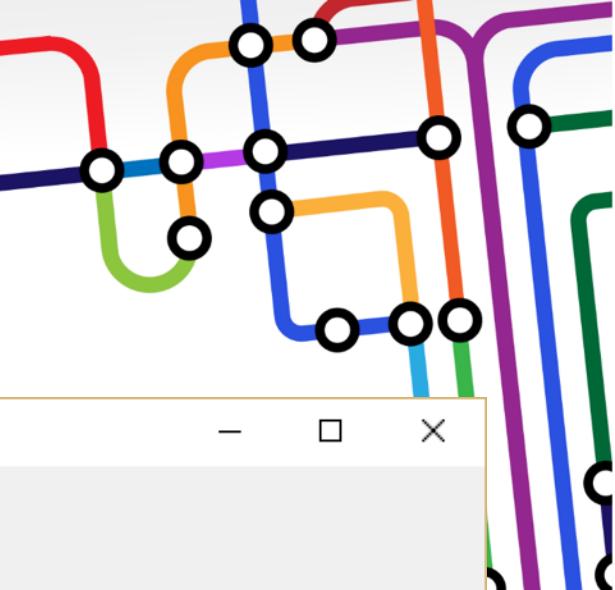
TDataSet,
qualquer
engine e
BD

Criação direta
da tabela
(metadados +
dados) no BD
via FireDAC

Arquivo texto

Metadata

- API comum para retornar metadados de qualquer BD, usando mesmo código / componentes;
- TFDConnection - métodos para retornar nomes de objetos do BD:
GetCatalogNames, GetSchemaNames, GetTableNames, GetFieldNames,
GetKeyFieldNames, GetGeneratorNames, GetPackageNames,
GetStoredProcNames etc.
- TFDMetaInfoQuery (TDataSet) - propriedade MetaInfoKind (mkCatalogs,
mkSchemas, mkTables, mkTableFields, mkIndexes, mkIndexFields,
mkPrimaryKey, mkPrimaryKeyFields, mkForeignKeys, mkForeignKeyFields,
mkPackages, mkProcs, mkProcArgs, mkGenerators, mkResultSetFields,
mkTableTypeFields)



Querying Metadata

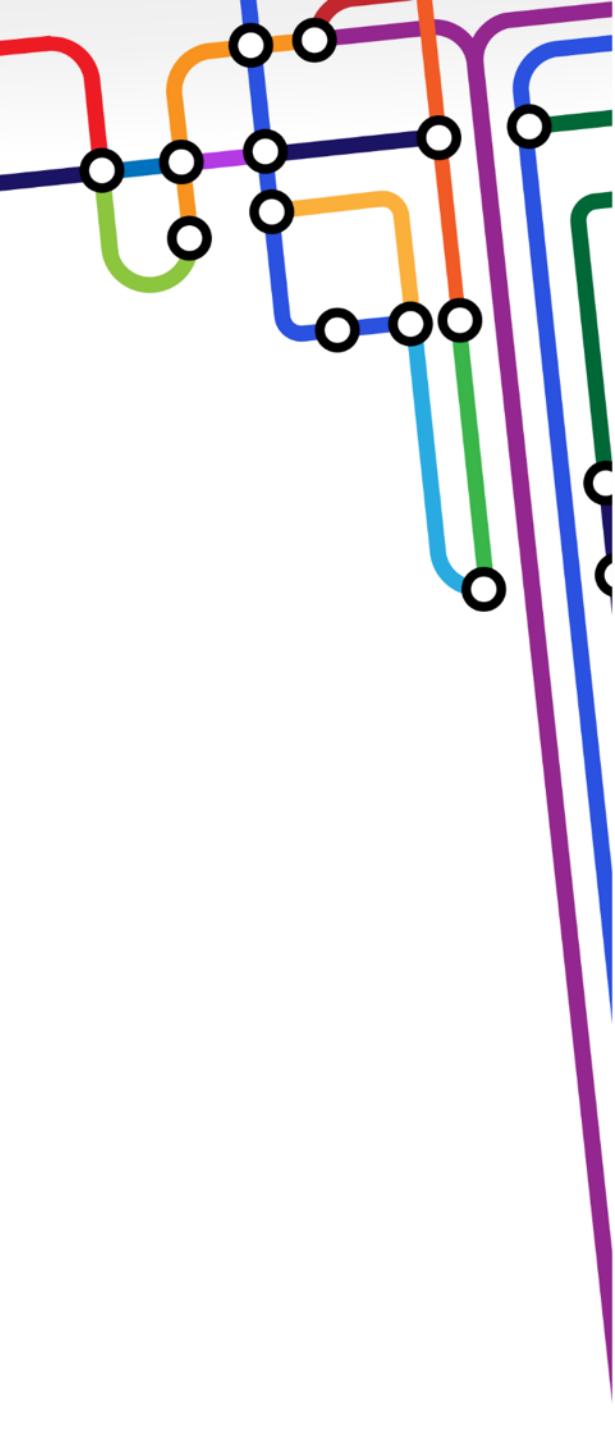
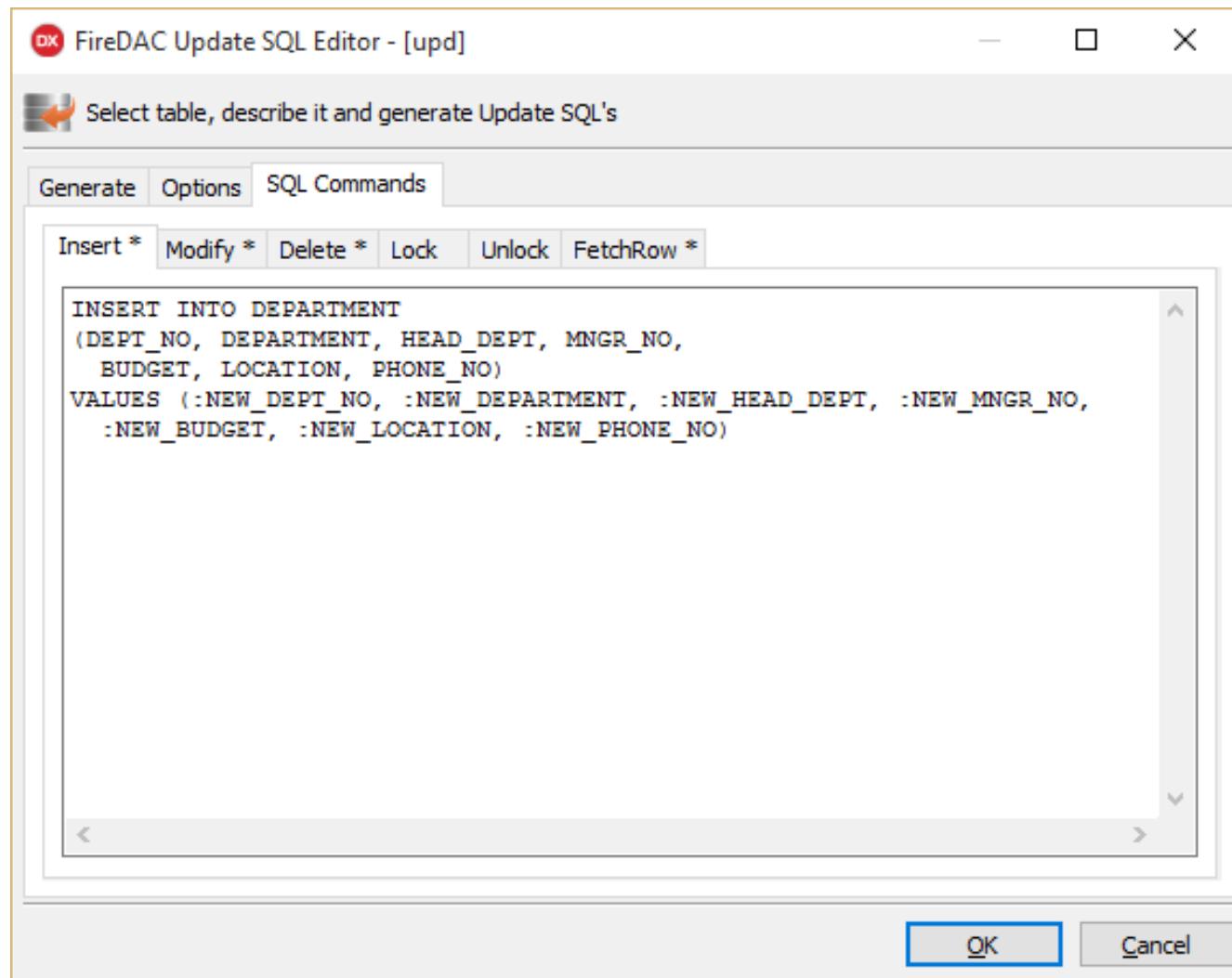
Escolha tabela MetaInfoKind

RECNO	TABLE_NAME	COLUMN_NAME	COLUMN_POSITION	COLUMN_DATATYPE	COLUMN_TYPENAME	COLUMN_ATTRIBUTES	COLUMN_PRECISION	COLUMN_SCALE	COLUMN_LENGTH
1	EMPLOYEE	EMP_NO	1	3	SMALLINT	16385	0	0	
2	EMPLOYEE	FIRST_NAME	2	23	VARCHAR	16385		0	15
3	EMPLOYEE	LAST_NAME	3	23	VARCHAR	16385		0	20
4	EMPLOYEE	PHONE_EXT	4	23	VARCHAR	16387		0	4
5	EMPLOYEE	HIRE_DATE	5	19	TIMESTAMP	16513		0	
6	EMPLOYEE	DEPT_NO	6	23	CHAR	16389		0	3
7	EMPLOYEE	JOB_CODE	7	23	VARCHAR	16385		0	5
8	EMPLOYEE	JOB_GRADE	8	3	SMALLINT	16385	0	0	
9	EMPLOYEE	JOB_COUNTRY	9	23	VARCHAR	16385		0	15
10	EMPLOYEE	SALARY	10	15	NUMERIC	16385	10	2	
11	EMPLOYEE	FULL_NAME	11	23	VARCHAR	17427		0	0

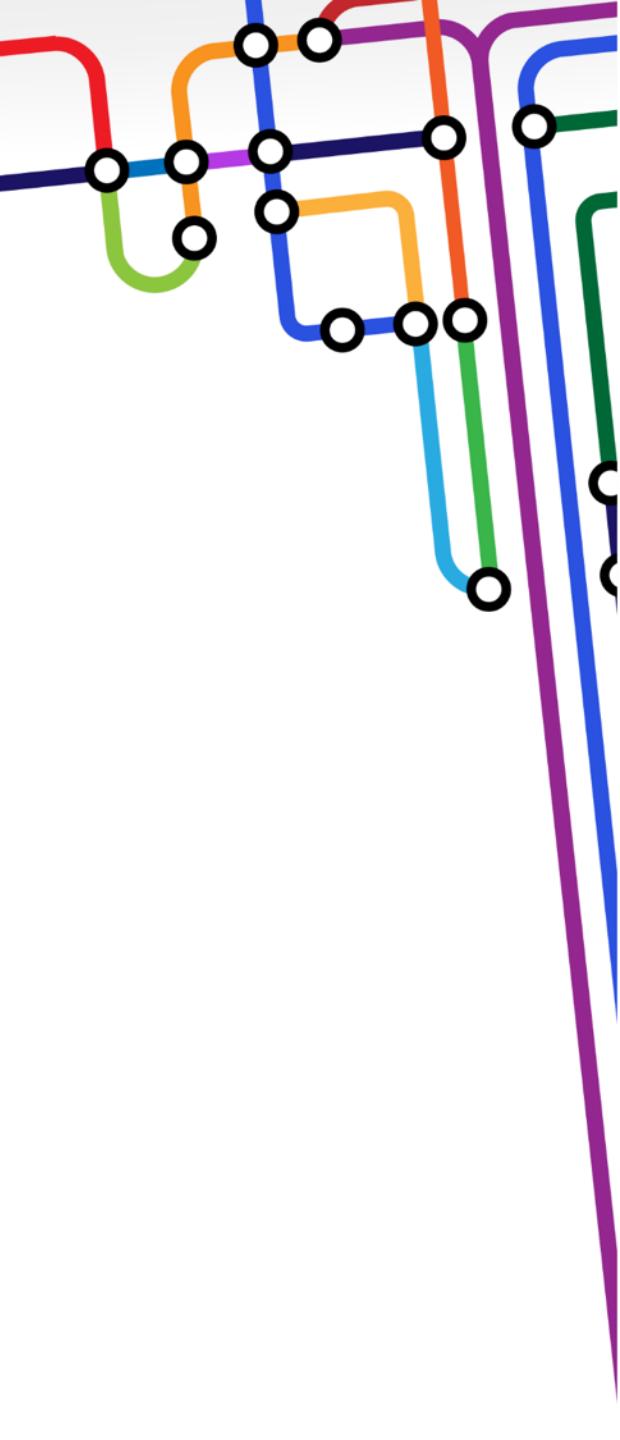
Cache e Update

- Cache Updates em memória, estilo antigo Cache Updates do BDE e Data do ClientDataSet
- ApplyUpdates permite enviar as atualizações para o BD, em um único lote (Delta, estilo ChangeLog do CDS)
- Cache é feita em memória, como CDS
- Componente específico separado (opcional) para geração de comandos de atualização (update, delete, insert, lock), estilo BDE / IBX
- Suporte a SavePoint (snapshot), UndoLastChange
- Suporte a Data e Delta (IFDDDataSetReference)

embarcadero conference



```
FDQuery1.CachedUpdates := True;  
iSavePoint := FDQuery1.SavePoint;  
try  
  FDQuery1.Append;  
  ...  
  FDQuery1.Post;  
  FDQuery1.Append;  
  ...  
  FDQuery1.Post;  
  FDQuery1.Append;  
  ...  
  FDQuery1.Post;  
except  
  FDQuery1.SavePoint := iSavePoint;  
end;
```



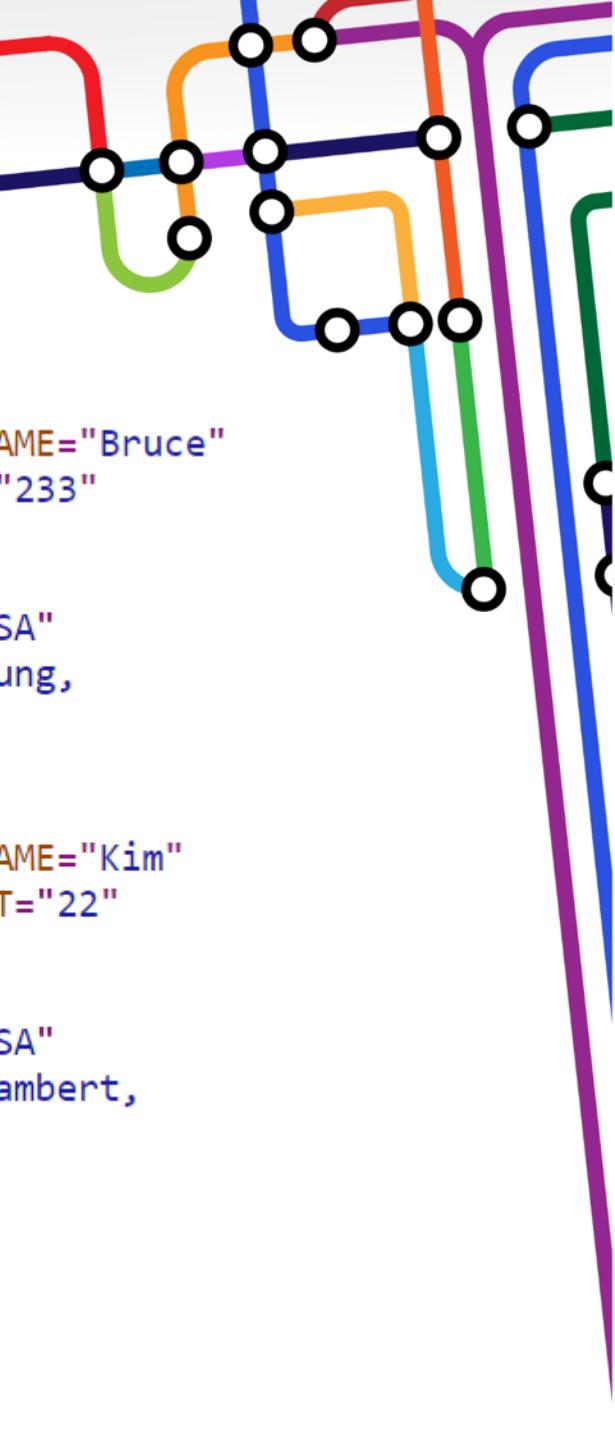
Save to File e Load from File

- Persistência em disco, estilo CDS
- Suporte aos formatos XML, JSON e binário
- Necessário registrar os componentes TFDStanStorageBinLink, TFDStanStorageXMLLink e TFDStanStorageJSONLink
- Também pode-se ler os dados de XML / JSON semelhante ao CDS

```
qry.SaveToFile ('\dados.json', sfJSON);  
qry.SaveToFile ('\dados.xml', sfXML);  
qry.SaveToFile ('\dados.bin', sfBinary);
```

embarcadero conference

```
"RowList": [ {  
    "RowID": 0,  
    "Original": {  
        "EMP_NO": 2,  
        "FIRST_NAME": "Robert",  
        "LAST_NAME": "Nelson",  
        "PHONE_EXT": "250",  
        "HIRE_DATE": "19881228T000000",  
        "DEPT_NO": "600",  
        "JOB_CODE": "VP",  
        "JOB_GRADE": 2,  
        "JOB_COUNTRY": "USA",  
        "SALARY": 104000,  
        "FULL_NAME": "Nelson, Robert"  
    }  
},
```



```
▼<Row RowID="1">  
    <Original EMP_NO="4" FIRST_NAME="Bruce"  
    LAST_NAME="Young" PHONE_EXT="233"  
    HIRE_DATE="19881228T000000"  
    DEPT_NO="621" JOB_CODE="Eng"  
    JOB_GRADE="2" JOB_COUNTRY="USA"  
    SALARY="97500" FULL_NAME="Young,  
    Bruce"/>  
    </Row>  
▼<Row RowID="2">  
    <Original EMP_NO="5" FIRST_NAME="Kim"  
    LAST_NAME="Lambert" PHONE_EXT="22"  
    HIRE_DATE="19890206T000000"  
    DEPT_NO="130" JOB_CODE="Eng"  
    JOB_GRADE="2" JOB_COUNTRY="USA"  
    SALARY="102750" FULL_NAME="Lambert,  
    Kim"/>  
    </Row>
```

Fetch

- Total controle sobre como os dados são obtidos do BD
- Propriedade FetchOptions.Mode e RowsetSize
- Mode = fmManual, fmOnDemand, fmAll, fmExactRecsMax
- RowsetSize = tamanho do pacote
- Semelhante ao uso de Packet Records do CDS
- Pode-se trazer os dados por demanda
- Cuidado com o cursor

```
qry.FetchOptions.RowsetSize := 3;  
qry.FetchOptions.Mode := fmManual;  
qry.Open(); // dataset abre vazio  
qry.GetNextPacket(); // traz 3 registros  
qry.GetNextPacket(); // traz +3 registros
```

obrigado

 guintherpauli.blogspot.com

 <http://www.facebook.com/guintherpauli>

 <http://www.twitter.com/guintherpauli>

 <http://br.linkedin.com/in/guintherpauli>

 <http://www.gpauli.com>

 guintherpauli@gmail.com

 [guinther.pauli](skype:guinther.pauli)

<http://fb.com/DelphiBrasil>

<http://fb.com/EmbarcaderoBR>

<http://www.embarcadero.com/mvp-directory>

<http://www.embarcaderobr.com.br/treinamentos/>

Download

Code Central

<http://bit.ly/fontesfiredac>

