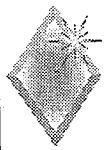


Clarion For Windows

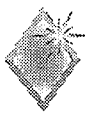
The Preferred Client/Server Development environment

Presentation by
Erik Pepping
Technical director Extensie B.V
The Netherlands

Topspeed Developers Conference
Ft. Lauderdale, September 1997

Did you ever wish for:

- Unlimited sort orders in Browses ?
- Unlimited sort orders in Reports ?
- Scalability of Database from laptop to Mainframe ?
- Security at the database level ?
- Easy to update/maintain database (without rebuild) ?
- Easy User definable Queries for browse/reports



Well ?????

It's Here !!!!

Get connected to RDBMS

A Small historical Overview

- Clarion for DOS could be linked to Oracle, Sybase via Third party Linked libraries, using library calls
- However, communication and DOS memory possibilities where very limited
- Windows changed all this and made it possible to get connected to the midrange and Mainframe world from the PC
- Clarion/Topspeed has long denied SQL existence
- Clarion got late on the train, VB, Powerbuilder Etc. where there first

Historical Overview 2

- Clarion has always been Record oriented:
  - Get (file,key)
  - Set (Key,Key)
  - Next(file)
  - etc.
- While it is more elegant to be set oriented
  - Select x from y where g = 'g' order by z

Differences Record<-> Set oriented

- Points to be made:
  - Retrieval of data
  - Performance
  - Scalibility / Hardware platforms
  - Security
  - Robustness / Integrity / Business Rules

Difference Record<-> SET oriented Retrieval of data (1)

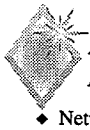
- *Purpose : Get the client with number 1.*
- Record oriented:
  - Cust_no = 1
  - Get(Customer,cus:key_cus)
- Set oriented
  - SELECT * from Customer where cust_no = 1
- So far no much difference

Difference Record<-> SET oriented Retrieval of data(2)

- *Purpose : Update all product prices in 1997 with +10%*
- Record oriented:
  - Clear(art:record); Art:PriceYear = 1997; Set(art:key_date,art:key_date)
  - Logout
  - Loop
  - next(Article); if errorcode() then break.
  - if Art:priceyear > 1997 then break.
  - Art:Price = Art:Price + Art:Price * 0.1
  - put(Article)
  - end
  - Commit
- Set oriented:
  - Update article set price=price*1.1 where Priceyear = 1997;
- Lots of difference !!!

Difference Record<-> SET oriented Retrieval of data (3)

- Purpose: Make a report with the best selling list of customers for each department in the company, ascending on department, best selling customers first, for 1997
- Record oriented oriented:
  - Sorry this would take to much space
- Set Oriented:
  - Select Depname ,Cust_name ,Sum(Ordertotal) from department, customers,orders where order.depno = dep.depno and order.client_no = Customers.clientno where orders.Year = 1997 group by depname,custname
  - loop
  - print
  - end



Difference Record<-> SET oriented Performance

- Network performance of RDBMS is better. A Select Custno From Customer where, will only send Custno over the network, not the other 250 fields in the Customer record.
- There is a misconception that a Select * from Customer where Name > 'AAAA' will wait to deliver records until all records are retrieved. This is not so.
- He ! Buy bigger hardware (Multi processor if you want).
- When joining data, RDBMS Server is always faster then Record oriented solution (joining is done on the server)

Difference Record<-> SET oriented Scalability / Hardware platforms

- What better than your customer choosing the target platform (for the server). Oracle runs on 70+ Operating systems
- Scalable from Portable PC to IBM Mainframe (No all RDBMS) SQL Server only runs on Windows NT, Oracle, Sybase and Ingres are better scaleable in that sense.

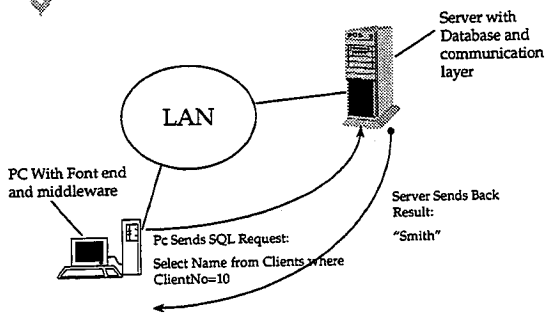
Difference Record<-> SET oriented Security

- Existing PC databases are hard to secure. Somebody takes all you PC-files in a minute.
- RDBMS is secured via Grant options on Users/Roles/Views and other objects
- Execute only permissions on stored procedures, gives controlled access to tables users normally have no access to
- It is very hard to take a complete database, because you do not have access to the Physical files, other then via the SQL Engine

Difference Record<-> SET oriented Robustness / Integrity / business rules

- Record oriented solutions code all rules in application.
- Rules are almost always never enforced everywhere
- Looking at data consistency/Integrity we have noticed that 90 % + off data files are no longer consistent with the rules.
- When ODBC, or another way is used to change files/fields, the rules are not applied.
- Triggers are always performed, no matter which procedure updates what, and this is done transaction wise.
- Read consistency / Commit and Rollback are important and are performed right on RDBMS

How does it work ?



What do you need ? (for oracle)

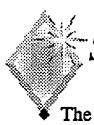
- The ORACLE Database driver (License policy is one license per server)
  - This will give you:
    - CW2OR16.DLL and CW2OR16.LIB
    - CW2OR32.DLL and CW2OR32.LIB
    - A Help file
    - A Manual
- ORACLE
  - You can start as small as Personal ORACLE (500$)
- If you want to connect to a server you need a communication protocol that is supported by Oracle and SQL NET from ORACLE

What does it do ?

- The driver translates as following:
  - Get(file,key) into Select * from <table> where field=value and field2= value, depending on the number of key fields in the key
  - SET(key,key) into select * from <table> where field > Startvalue

Which Extra's do you get ?

- You can directly access the database with:
  - Filename{prop:sql}=
- You can add whereclauses to a view with
  - Viewname{Prop:whereclause} =
- The filename{prop:sql} can start any valid SQL statement like:
  - Create table abc (a char(5))
  - Or start a stored procedure
    - filename{prop:sql} = 'begin sp_name(); end'
- you can include hints (ORACLE only) for example to use a certain index



Some experiences (1)

- The standard Clarion Browse is slow, since they keep access files over and over again
- The driver is very fast, native ORACLE CLI is used, so no extra translation (esp. when using {prop:sql}
- Reporting becomes very very simple. almost all reports are only 1/2 page coding. It consists of an SQL-select, a loop and a print (rpt:detail)
- Dictionary synchronizing is promised in the enterprise edition. (Currently this is a pain)
- A SQL-statement builder (like powerbuilder has) is lacking (3-party opportunity)
- SQL Templates are recommended

Some Experiences (2)

- Error message reporting is good (fileerrorcode() gives the Oracle Error message).
- 32-bit works better than 16 bit.
- Be careful which oracle version you are using. It is made for version 7.2, and needs ORO72.DLL. ORA73.DLL on W95 will work (when you rename it), but on NT it will not work (here you need to install Oracle 7.2 required support files) This is fixed in the latest driver !, But how about future Oracle Releases?
- How will Clarion keep up with Oracle changes (Oracle 8 has object oriented features
- It is not possible to work directly with a oracle cursor returned by a stored procedure

Some experiences (3)

- You can not do select field1,field2 from file, when the files has more fields then field1 and field 2. You have to define a view
- The driver automatically includes fields on which there are keys sometimes which can result into a message that the field is not included in the select statement
- Prop:sql has to be in the order of the dictionary fields, if you add a field, you will have to change all your {Prop:sql}. (That's why I made a SetSQL Function)

Some experiences (4) Add On's

- It is easy to create extra Add-on's like:
  - Creating user / Roles from with CW app
  - Security on insert/change/delete buttons depending on user grants in database
  - Make a SQL select with columns from a table (This will be demonstrated)
  - Dictionary wizards for creation of Grant, column comment, Sequence generation scripts, and more (this will be demonstrated)

A Pratical example of an APP

- We have build a large CW/Oracle application for Cervix Cancer Screening project in the Netherlands
- Database contains hundredths of thousands of records. Largest table contains 1.5 Million Records
- Build in 3 months
- Contains 80 Tables, and about 300 Procedures/Functions, many triggers, stored procedures and other Oracle objects
- Build with DLL's
- Ver y stable, with a good performance

SQL Templates

- CCS SQL templates are a must !!!!
- It gives you:
  - Great browse. Generates LIKE statement for locators
  - Gives you locator on every column of the table
  - Sort on every column of the table with click on the header
  - SQL Combo box
  - SETNULL template
  - SQL explorer Tree
  - QBE window
  - Great performance
  - Oracle, SqlServer and others create scripts objects

Why better than VB?

- VB uses ODBC. It is hard to get the most out of you database / ODBC
- You always need to perform a select. CW still has the simple GET, which make coding short and understandable
- VB requires a lot of add-on utilities to perform database programming (data aware controls)
- VB links into a lot of standard window DLL's. With Clarion you only install 2-5 DLL's and that it to connect to RDBMS
- VB is unmaintainable, and without any structure (not comparable with a CW APP file
- CW is faster (See the Why clarion Brochure)

Why better than Powerbuilder

- Powerbuilder is very hard to learn (or maybe I'm Stupid). Steep Learning curve
- You need a lot of add on utilities (commercial libraries) to make a simple reusable Master-Detail screen
- You need to distribute a lot with you application
- CW is Faster
- The datawindow concept is it's strength, but also it's weakness.
- Although it is supposed to be object oriented, it is almost never used that way.

DEMO

- Demonstration Dictionary differences.
- Connecting to an Example application (Oracle tools)
- Demo of the SQL template browse
- View the code of a report.
- Dictionary Template extensions demo