UN DRONE TUTTO DA PROGRAMMARE!

Vola a "bordo" di un sofisticato quadricottero e sviluppa tu le app per estenderne le funzioni

VERSIONE PLUS RIVISTA+LIBRO+CD € 9,99

VERSIONE STANDARD RIVISTA+CD € 6,99

ioPROGRAMMO

PER ESPERTI E PRINCIPIANTI

Poste Italiane S.p.A Spedizione in A.P. • D.L. 353/2003 (ONV.IN L.27/02/2004 N.46) ART.1 COMMA1 DCB ROMA. • Periodicità bimestrale • Settembre/Ottobre 2013 • ANNO XVII, N.6 (187)

iOS 7

I nuovi iDevice si programmano così!

PRONTO PER I NUOVI IPHONE 5S E IPAD 5

Cosa aspetti? Non farti trovare impreparato...
Con la nostra guida impari fin da subito a sfruttare le **nuove feature** del sistema mobile più rivoluzionario di sempre

- ✓ **Multitasking**
- ✓ **AirDrop**
- ✓ **Gestione testo**
- ✓ **Sviluppo giochi**
- ✓ **Nuove mappe**
- ✓ **Connessioni P2P**

Il tuo VLC in HTML5

Sviluppa un player multimediale per le tue pagine Web

SUL CD IL CODICE PRONTO ALL'USO

Cross-platform e compatibile con i codec e i formati audio/video di ultima generazione

Il trova-amici con Google Maps

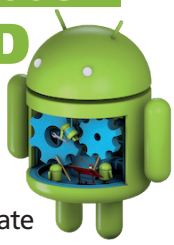
Realizza insieme a noi l'app social che ti permette di chattare e interagire con gli amici intorno a te

Salta la fila grazie al wireless

Progetta un efficiente sistema di check-in sfruttando le connessioni senza fili NFC dei dispositivi mobili

PROVATO PER VOI L'IDE UFFICIALE DI GOOGLE

ANDROID STUDIO



Tool integrati e tempi di sviluppo più rapidi. Usalo per creare app più affidabili e sofisticate

Nuova versione sul CD-Rom

WEB 2.0

UN MOTORE DI RICERCA CREATO "INTORNO A TE"

Sfrutta le funzioni dei dispositivi mobile per sviluppare un sistema più preciso di Google & Co.

UN SITO INTERNET PERFETTO IN UNA SOLA PAGINA!

Ecco come creare un portale "one page navigation" browsabile da PC, smartphone e tablet

ANIMAZIONI WEB "SOLO" CON I FOGLI DI STILE

Utilizza CSS3 per creare transizioni e trasformazioni grafiche, anche senza JavaScript

COBOL: LA RINASCITA

La soluzione "indolore" per modernizzare tutto il software scritto in questo linguaggio

ELETTRONICA

UN TELECOMANDO UNIVERSALE FAI DA TE

Utilizza Arduino per comandare a distanza TV, player multimediali, decoder, PVR...

9 771128 594009 30187

EDIZIONI MASTER
www.edmaster.it

APPLICAZIONI COBOL: UNA SECONDA GIOVINEZZA!

LA SOLUZIONE INTELLIGENTE E "INDOLORE" PER MODERNIZZARE E RENDERE RIUTILIZZABILI GLI STRUMENTI IMPLEMENTATI CON QUESTO STORICO LINGUAGGIO DI PROGRAMMAZIONE. PERCHÉ EVOLVERSI NON SIGNIFICA RINUNCIARE ALL'AFFIDABILITÀ E ALL'EFFICIENZA DI SISTEMI COLLAUDATI IN ANNI DI SVILUPPO

Oggi sono numerosissime le realtà aziendali attive nel campo del software che si affidano ad applicativi "legacy" in termini di linguaggio di programmazione e non dispongono di interfacce con i nuovi paradigmi che il mercato in continua evoluzione impone di usare. La necessità di modernizzare è diventata sempre più pressante, sopratutto a causa della forte "spinta" esercitata dagli utenti finali che, oltre a richiedere interfacce più ergonomiche, necessitano, sempre più spesso, di strumenti di integrazione, reporting, business intelligence e altro moderni. Il valore degli applicativi esistenti non è comunque assolutamente trascurabile. Esso, infatti, consiste nella maturità derivante dalla continua evoluzione e dalla stabilità raggiunta in anni e anni di sviluppo e riscrivere tali software non è una strada percorribile, né tecnicamente, né economicamente. Cosa bisogna fare allora? Il trucco vincente consiste nel coniugare l'affidabilità degli applicativi esistenti con le nuove opportunità offerte dagli strumenti di ultima generazione. In sintesi, occorre una soluzione rapida, "indolore" in grado di garantire continuità e linearità con il passato, la stessa implementata con successo dalla società *Metodo S.r.l.* di Modena che è significativa per vari aspetti e utilizza un approccio a volte non considerato, ma che può essere "L'uovo di Colombo". In pratica, si tratta di modernizzare attraverso il database (o file system come spesso viene definito nel mondo *COBOL*), mantenendo inalterata l'infrastruttura e gli applicativi ma, soprattutto, senza modificare una sola riga di codice.

Conoscenze richieste

Linguaggio di programmazione COBOL

Software

Microsoft Visual Studio, c-treeACE for COBOL, versione di valutazione ottenibile da FairCom (www.faircom.com)

Impegno

Tempo di realizzazione

UNA SOLUZIONE BRILLANTE

Durante il processo di modernizzazione, Metodo S.r.l. ha dovuto confrontarsi con vari problemi oggettivi:

- Esistenza di applicativi scritti in COBOL (decine di migliaia di linee di codice);
- Manutenzione giornaliera degli applicativi (obiettivo in movimento);
- Sviluppatori esperti in materia di flussi aziendali e COBOL, ma non di nuovi linguaggi di programmazione;
- Necessità di mantenere l'intero impianto funzionante al 100%;
- Evolvere le interfacce utente, comunicare con il Web, scambiare dati con altri applicativi e linguaggi.

La soluzione scelta da Metodo S.r.l. per risolvere tali problemi è stata tanto semplice quanto efficace. Innanzitutto, bisognava mantenere il progetto COBOL inalterato, inclusa la sua evoluzione e manutenzione. Inoltre, era necessario integrare gradualmente nuovi paradigmi di programmazione. Scopriamo come sono riusciti a raggiungere tali obiettivi.

CONVIVENZA POSSIBILE

Il linguaggio COBOL ha un accesso di tipo *ISAM* e scrive direttamente un buffer contenente i dati in formato proprietario. I dati, ovviamente, sono interpretati correttamente dall'applicativo COBOL, ma per gli altri linguaggi non sono altro che stringhe di byte. Inoltre, bisogna considerare che i COBOL vendor, ovviamente, non hanno mai spinto l'integrazione con altri linguaggi per proteggere il proprio mercato, quindi, si tratta di una realtà chiusa. Tenendo conto di tutto questo e delle nuove esigenze, per poter effettuare la transizione in modo indolore, Metodo S.r.l. ha sostituito il file system nativo di COBOL con il *c-treeACE* (www.faircom.com/ace/ace_for_cobol_t.php, **Fig. 1**) sviluppato da *FairCom* (www.faircom.com). Questa operazione, assolutamente indolore dal punto di vista del codice, ha permesso di mantenere l'accesso *ISAM* del COBOL invariato e, di conseguenza, non ha richiesto nessuna modifica ai sorgenti e non ha causato alcun degrado in termini di performance, anzi, il risultato è stato un miglioramento evidente di velocità e stabilità. Il file system FairCom una volta linkato al runtime, ha immediatamente permesso di condividere i dati scritti da COBOL con altri linguaggi di programmazione e la scelta di Metodo è stata quella di utilizzare il linguaggio di scripting *Tcl* (www.tcl.tk), migrando modulo per modulo l'intera applicazione e permettendo, allo stesso tempo,

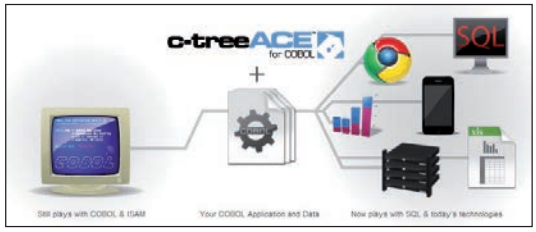


Fig. 1: Il processo di "modernizzazione" di un'applicazione COBOL

l'accesso contemporaneo e la condivisione dei dati dalle due architetture.

LIMITI DELLE SOLUZIONI ALTERNATIVE

Sul mercato esistono varie soluzioni che permettono di "modernizzare" un applicativo COBOL, ma spesso comportano degli svantaggi. Ad esempio, il passaggio di una soluzione COBOL ad un sistema completamente relazionale (*SQL Only*), come quello fornito da Oracle o Microsoft SQL Server, comporta diversi svantaggi: utilizzo di componenti bridge costosi; traduzione di ogni richiesta del COBOL in SQL statement (perdita di performance da parte dell'applicazione esistente); modifica del codice esistente a causa delle diversità di gestione dei dati; non permette di fare coesistere le diverse applicazioni migrando modulo a modulo ma, necessariamente, il programma comleto viste le modifiche necessarie ai sorgenti; si tratta di una tecnologia sicuramente valida ma lontana dalla gestione "file by file" che FairCom mantiene e che risulta più naturale per la gestione e la manutenzione a cui i "cobolisti" sono abituati (**Fig. 2**). Altre soluzioni cercano di ovviare al problema di condivisione dei dati con applicazioni più moderne, attraverso l'uso di procedure batch che mantengono copie non sincronizzate del database accessibili dal Web o con strumenti di reporting. Questo tipo di soluzione non permette di aggiornare ma, sopratutto, di accedere ai dati in tempo reale con tutti i limiti che ne derivano.

UNO PER TUTTI!

Integrare la tecnologia FairCom nei propri applicativi è estremamente semplice. Tipicamente chi ha utilizzato *file-handler* di terze parti o scritto delle routine di gestione per file ad indici ha spesso utilizzato un approccio programmatico orientato ai record, il cosiddetto ISAM. Oggi, molto spesso, queste realtà si trovano a fare i conti con i limiti dei sistemi sviluppati in casa o con le politiche tecnico/commerciali dei vendor, che non sempre sono in linea con le reali necessità delle software house. FairCom offre per tutti questi casi una valida alternativa con una tecnologia ISAM estremamente affidabile e portabile interfacciata direttamente con il mondo SQL e con un elevato numero di API. Per il mondo *Btrieve* esiste già un driver diretto che permette di sostituire direttamente quest'ultimo con c-treeACE, mantenendo le stesse chiamate nel codice e garantendo così una serie di vantaggi sia commerciali sia tecnici.

DALLA TEORIA ALLA PRATICA

Adesso che conosciamo quali sono i vantaggi offerti dalla tecnologia FairCom e abbiamo le prove che è stata utilizzata con successo dalla società Metodo S.r.l., di seguito riportiamo una breve guida per gli utilizzatori di *Acucobol* tratta dal

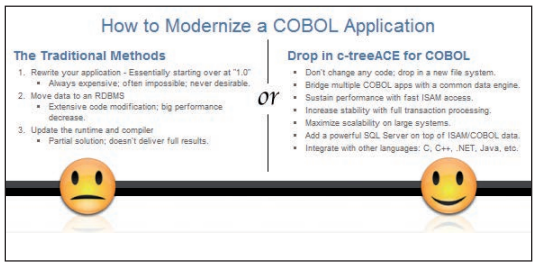


Fig. 2: Confronto tra il nuovo metodo c-treeACE for COBOL di FairCom e quelli tradizionali

manuale di c-treeACE. Le operazioni da eseguire sono più o meno simili a quelle degli altri COBOL, fatta eccezione per *IsCobol* di Veryant, che non necessita di alcuna operazione, visto il supporto integrato per c-treeACE con estensione SQL. Iniziamo subito! Per aggiungere il supporto per c-treeACE for COBOL ad *ACUCOBOL-GT* su Windows, è necessario disporre dell'ambiente di sviluppo *Microsoft Visual Studio*. Se vogliamo aggiungere il supporto a ACUCOBOL-GT versione 9, è necessario utilizzare *Visual Studio 2008*. Passiamo ora alle modifiche da implementare per compilare correttamente il supporto. Per prima cosa, bisogna copiare il file *ctreeacu.c* nella directory lib di *ACUCOBOL-GT*:

```
copy win32\Driver\ctree.cobol\Acucobol\ctreeacu.c
                                        C:\AcuGT\lib
```

Subito dopo, è necessario copiare l'utility *ctutil.exe* e la DLL *mtclient.dll* nella directory *bin* di ACUCOBOL-GT mediante i comandi seguenti:

```
copy win32\Driver\ctree.cobol\Acucobol\mtclient.dll
                                        C:\AcuGT\bin
copy win32\Tools\cmdline\ctutil.exe C:\AcuGT\bin
```

A questo punto, apriamo la soluzione Visual Studio *wrun32.sln* e aggiungiamo al progetto *wrundll* il file *ctreeacu.c* copiato con l'operazione svolta in precedenza. Per aggiungere il file, basta cliccare sul menu *Progetto*, selezionare *Aggiungi elemento esistente* e scegliere *ctreeacu.c* dalla directory *lib* di ACUCOBOL-GT. Subito dopo, apriamo e modifichiamo il file di gestione dei file system di ACUCOBOL-GT, chiamato *filetbl.c* (il file fa parte del progetto *wrundll* creato precedentemente con Visual Studio). Il file *filetbl.c* va modificato in tre punti particolari. Vediamo come procedere. Per prima cosa, individuiamo le tre righe seguenti:

```
#ifndef USE_VISION
#define USE_VISION 1
#endif
```

Trovate le righe di nostro interesse, subito dopo aggiungiamo la seguente dichiarazione:

```
#ifndef USE_CTREE
#define USE_CTREE 1
#endif
```

NOTA

UNA BASE SOLIDA

Metodo S.r.l. (www.metodo.net) è la software house di *LAPAM* Federimpresa (www.lapam.mo.it/lapam), che dal 1995 ha iniziato un progetto di modernizzazione della propria infrastruttura e del software. È stata tra le prime realtà ad utilizzare GNU/Linux come piattaforma software, sia per quanto riguarda i server, sia i client, il tutto creando una propria distribuzione del potente sistema operativo Open Source. Lapam Federimpresa è una Federazione di Associazioni, della provincia di Modena, che rappresenta tutto il mondo imprenditoriale. Il sistema informatico di Lapam, sviluppato e gestito da Metodo S.r.l., comprende 800 postazioni Unix distribuite su 45 sedi e accede ad un archivio di 750 GB contenente i dati di circa 12.000 aziende per le quali elabora 9.000 contabilità, 37.000 cedolini paga mensili e 45.000 dichiarazioni dei redditi.

NOTA

ANCORA PIÙ SEMPLICE

Partendo dall'esempio di Metodo S.r.l., FairCom ha sviluppato una soluzione ancora più lineare ed estesa mantenendo la medesima semplicità di approccio. Il supporto è stato esteso da FairCom a vari COBOL (*ACUCOBOL*, *MicroFocus*, *IsCobol*, *NetCOBOL*, *COBOLIT* etc), che oggi possono, grazie al file system di FairCom, condividere i dati utilizzando *SQL* tramite *JDBC*, *ODBC*, *Ado.NET*, *PHP*, *Python* e con linguaggi di programmazione classici, come *C*, *C++*, *JAVA* etc. Il tutto, utilizzando il livello *ISAM*.

Per procedere con la seconda modifica, invece, individuiamo la riga seguente: *extern DISPATCH_TBL v5_dispatch,...;*. A questo punto, immediatamente dopo, aggiungiamo la riga riportata di seguito: *extern DISPATCH_TBL ct_dispatch;*. Infine, apportiamo la terza modifica necessaria. Per farlo, individuiamo il blocco seguente:

```
TABLE_ENTRY file_table[] = {
#if USE_VISION
{ &v5_dispatch, "VISIO" },
#endif /* USE_VISION */
```

Subito dopo, aggiungiamo la seguente definizione:

```
#if USE_CTREE
{ &ct_dispatch, "CTREE" },
#endif /* USE_CTREE */
```

A questo punto, è necessario ricompilare il runtime ACUCOBOL-GT contenuto nella DLL *wrun32.dll*. Per farlo, selezioniamo *Compila soluzione* dal menu *Compila* di Visual Studio. Fatto questo, copiamo la nuova *wrun32.dll* nella directory *bin* di ACUCOBOL-GT: *copy C:\AcuGT\lib\wrun32.dll C:\AcuGT\bin*. Ora, verifichiamo che il supporto per c-treeACE for COBOL sia compilato correttamente eseguendo il comando *wrun32 –vv*. Questo comando mostrerà i vari file system supportati dalla versione di ACUCBOL-GT in uso. Se c-treeACE for COBOL è correttamente implementato, vedremo tra i vari file system anche il seguente:

```
FairCom c-treeACE version 10.1.0.xxxxx-yymmdd file
                                system (interface v10.1.0.xxxxx-yymmdd)
```

ESECUZIONE DI UN PROGRAMMA COBOL

Ora che il runtime *ACUCOBOL-GT* supporta c-treeACE for COBOL, vediamo subito come eseguire un programma scritto in COBOL sfruttando questo efficiente strumento. Per prima cosa, copiamo il sorgente, nel nostro caso *cobol_Tutorial1.cbl*, nella directory *bin* di ACUCOBOL-GT:

```
copy \FairCom\V10.x.x.COBOL\win32\ctree.cobol\
                        tutorials\cobol_Tutorial1.cbl C:\AcuGT\bin
```

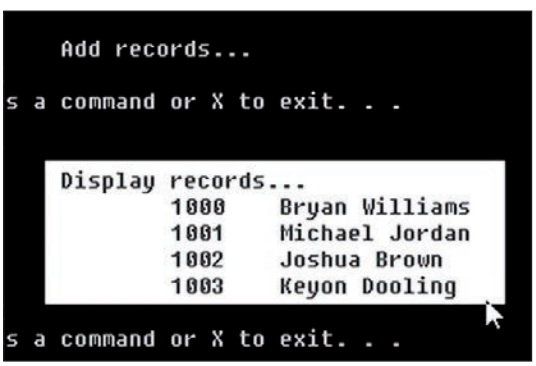


Fig. 3: Il programma COBOL in esecuzione

Avviamo il compilatore ACUCBOL-GT come mostrato di seguito:

```
cd C:\AcuGT\bin
ccbl32.exe -fx cobol_Tutorial1.cbl
```

Nello specifico, l'opzione *-fx* forza il compilatore a generare un file *XFD* che definisce la struttura dati utilizzata da *cobol_Tutorial1*. Successivamente, utilizzeremo il file *XFD* appena generato per mostrare le caratteristiche del supporto SQL di c-treeACE for COBOL. A questo punto, prima di eseguire *cobol_Tutorial1*, è necessario definire la variabile d'ambiente *DEFAULT_HOST*. Questa variabile seleziona quale file system sarà utilizzato dal runtime ACUCOBOL-GT: *set DEFAULT_HOST=CTREE*. Infine, possiamo eseguire il programma *cobol_Tutorial1* invocando il runtime ACUCOBOL-GT nel modo seguente (**Fig. 3**): *wrun32.exe cobol_Tutorial1.acu*.

ACCESO AI DATI TRAMITE SQL

Il programma *cobol_Tutorial1* crea un file di dati indicizzato chiamato *CUSTMAST*. Tale file può essere aperto tramite l'utilizzo di semplici query utilizzando ancora una volta c-treeACE SQL. Per aprire *CUSTMAST* tramite c-treeACE SQL seguiamo la procedura riportata di seguito. Questo metodo utilizza un tool interattivo a riga di comando, nello specifico *isql*, che permette l'esecuzione di query sui dati gestiti da c-treeACE SQL. Vediamo come procedere. Innanzitutto, importiamo direttamente il file di dati *custmast.dat* mediante il tool ctutil servendoci anche del file XFD generato in precedenza. In particolare, invocando *ctutil* con il parametro *-sqlize, questo* convertirà il file *XFD* in formato *XDD* e andrà ad inserirlo nell'intestazione del file di dati, quindi, quest'ultimo verrà automaticamente importato nel database specificato tra i parametri. Ecco il comando da eseguire: *ctuil.exe -sqlize custmast custmast.xfd ADMIN ctreeSQL*. Arrivati a questo punto, dobbiamo avviare ISQL:

```
\FairCom\V10.x.x.COBOL\win32\tools\cmdline\isql.
                                exe -u admin -a ADMIN ctreeSQL
```

Infine, dall'interfaccia di ISQL visualizziamo i dati contenuti nella tabella *custmast table* eseguendo una semplice istruzione *SQL SELECT* (**Fig. 4**):

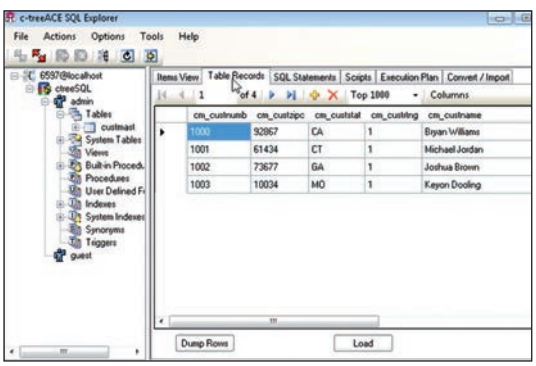


Fig. 4: Visualizzazione dei dati della tabella tramite c-treeACE Explorer

NOTA

LINEARITÀ PRIMA DI TUTTO

Metodo S.r.l., che ha deciso per la gestione degli stipendi di avvalersi di una procedura paghe fornita da una software house leader del mercato, ma non ha voluto rinunciare alla scelta del file system FairCom, ha semplicemente sostituito direttamente il file system COBOL con *FairCom c-treeACE* a runtime, garantendo l'uniformità e la stabilità dei dati. Questo conferma ulteriormente la linearità della soluzione al punto che un utente finale esperto è in grado di implementarla senza avere di fatto accesso ai sorgenti.

NOTA

COBOL ANCORA IN AUGE!

Un altro esempio di utilizzo di c-treeACE specifico nel mondo COBOL, riguarda l'implementazione del file system FairCom nella soluzione *isCobol di Veryant* (www.veryant.com). Una tecnologia tutta italiana che propone una valida alternativa ai classici COBOL (Acucobol, MicroFocus etc) attraverso un compilatore e un runtime COBOL entrambi sviluppati in JAVA, aspetto che ne garantisce la portabilità e con un ottimo successo all'estero, come spesso accade per le aziende italiane, sopratutto negli Stati Uniti, dove conta ormai centinaia di clienti tra cui molti ISV le cui applicazioni vengono distribuite a migliaia di end user o come in Olanda, dove ANVA (gestionale in ambito assicurativo) ha già distribuito 12.000 posti di lavoro con la nuova tecnologia.

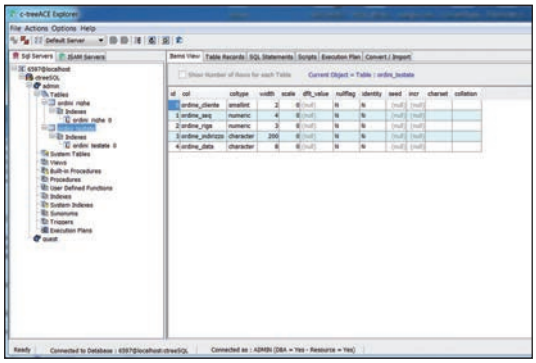


Fig. 5: Il contenuto della tabella ORDINI_TESTATE

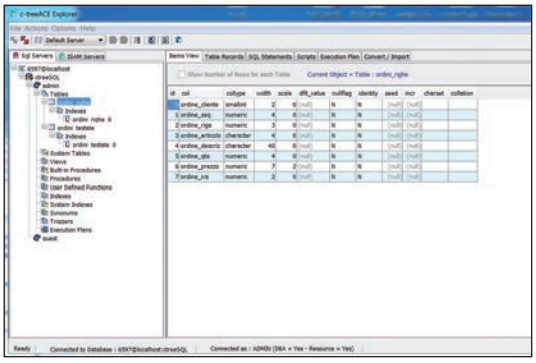


Fig. 6: Il contenuto della tabella ORDINI_RIGHE

```
ISQL> SELECT * FROM CUSTMAST;
...
  <field name="CM_CUSTNAME" size="47"
                  type="Alphanum" digits="47" scale="0"/>
  <field name="CM_CUSTADDR" size="47"
                  type="Alphanum" digits="47" scale="0"/>
  <field name="CM_CUSTCITY" size="47"
                  type="Alphanum" digits="47" scale="0"/>
 </schema>
</table>
```

UNA TECNICA PARTICOLARE

Di seguito riportiamo un esempio di *file descriptor*, tipicamente contenuto nel sorgente di un programma COBOL, che utilizza la tecnica delle *redefines* per gestire nella stessa tabella sia la testata sia le righe di un ordine:

```
      FD  ORDINI
          LABEL RECORD IS STANDARD.
      01 REC-ORDINI.
          02 ORDINE-CHIAVE.
              03 ORDINE-CLIENTE   PIC  9(04) USAGE IS
                                              COMP-1.
...
              03 ORDINE-QTA          PIC  9(4) USAGE IS
                                              COMP-6.
              03 ORDINE-PREZZO       PIC  9(5)V9(2)
                                      USAGE IS COMP-3.
              03 ORDINE-IVA          PIC  9(02) USAGE
                                              IS COMP-6.
```

Dopo la compilazione mediante l'utilizzo dell'opzione *–fx* vista in precedenza e dopo aver eseguito l'utility contenuta nella distribuzione di c-treeACE for COBOL (*ctutil* con il parametro *–xfd2xdd*), otteniamo un file con estensione *XDD* che definisce per SQL la struttura del record dividendolo su due tabelle distinte (*ORDINI_TESTATE* e *ORDINI_RIGHE*), entrambe gestibili da SQL:

```
<?xml version="1.0" encoding="US-ASCII"?>
<table name="ORDINI" minRecLen="215" maxRe
                                      cLen="215">
 <key duplicate="false" primary="true">
  <segment offset="0" size="7"/>
  <part name="ORDINE_CLIENTE" offset="0" size="2" />
  <part name="ORDINE_SEQ" offset="2" size="3" />
  <part name="ORDINE_RIGA" offset="5" size="2" />
  ...
size="2" type="BinarySigned" digits="4" scale="0" />
  <field name="ORDINE_SEQ" indexed="true" size="3"
                  type="PackedPositive" digits="4" scale="0" />
  <field name="ORDINE_RIGA" indexed="true" size="2"
                  type="PackedPositive" digits="3" scale="0" />
  <field name="ORDINE_INDIRIZZO" size="200"
                    type="Alphanum" digits="200" scale="0" />
  <field name="ORDINE_DATA" size="8"
                     type="Alphanum" digits="8" scale="0" />
 </schema>
 <schema name="ORDINI_RIGHE" size="215" filter="2">
  <field name="ORDINE_CLIENTE" indexed="true"
     size="2" type="BinarySigned" digits="4" scale="0" />
  <field name="ORDINE_SEQ" indexed="true" size="3"
                  type="PackedPositive" digits="4" scale="0" />
  <field name="ORDINE_RIGA" indexed="true" size="2"
                  type="PackedPositive" digits="3" scale="0" />
  <field name="ORDINE_ARTICOLO" size="4"
                    type="Alphanum" digits="4" scale="0" />
  <field name="ORDINE_DESCRIZ" size="40"
                  type="Alphanum" digits="40" scale="0" />
  <field name="ORDINE_QTA" size="2"
                type="PackedUnsigned" digits="4" scale="0" />
  <field name="ORDINE_PREZZO" size="4"
                  type="PackedPositive" digits="7" scale="2" />
  <field name="ORDINE_IVA" size="1"
                type="PackedUnsigned" digits="2" scale="0" />
  <field name="FILLER58" hidden="true" size="157"
                    type="Alphanum" digits="157" scale="0" />
 </schema>
</table>
```

Questa singola tabella COBOL, nel *c-treeACE* SQL Explorer verrà vista come due tabelle separate (**Fig.5** e **Fig. 6**). La modernizzazione del programma COBOL termina qui! All'indirizzo www.faircom.com/ace/ace_for_cobol_modernize_file_system_t.php è disponibile la documentazione completa per chi desidera approfondire l'argomento.

NOTA

UNA TECNOLOGIA COLLAUDATA

Per molti utenti finali e per alcuni degli addetti ai lavori FairCom non rappresenta un nome conosciuto come Oracle o Microsoft, sopratutto perché la tecnologia di FairCom è stata da sempre utilizzata da ISV e produttori di software come componente interno e non come un prodotto commerciale da acquistare separatamente. FairCom fornisce dal 1979 un database noto con il nome *c-tree*, poi evoluto da semplice file-handler ad una tecnologia clent/server in ambiente transazionale con un layer relazionale SQL, con il nome di *c-treeACE* (*Advanced Core Engine*). Le implementazioni della tecnologia FairCom sono innumerevoli e vanno da installazioni embedded a sistemi finanziari mission-critical come, ad esempio, *VISA Europe*,nel cui sistema VDPS di autorizzazione delle catre di credito basato su c-treeACE server, sono transitati e autorizzati nel 2012, 1,23 trilioni di Euro di acquisti con carte di credito VISA in tutta Europa, con un obbiettivo a 2 trilioni di Euro nel 2015 e un numero totale di 19 miliardi di transazioni tutte gestite da FairCom c-treeACE.