

Consiglio Nazionale delle Ricerche

SQL/DS:

Introduzione all'utilizzo

del sistema SQL/DS

1^a PARTE

R. Bartoli - A. Ceccarelli - O. Signore

202

GNUCE

A cura di : R. Bartoli
A. Ceccarelli
O. Signore

Copyright - Ottobre 1986

by - CNUCE - Pisa

Istituto del Consiglio Nazionale delle Ricerche

INDICE

| | |
|---|-----------|
| Lista delle figure | v |
| 1.0 Introduzione | 1 |
| 2.0 Architettura SQL/DS | 3 |
| 3.0 Definizione di DBSPACE | 7 |
| 4.0 Accesso al DB/DC | 8 |
| 5.0 Interrogazione dei dati di una tabella | 10 |
| 5.1 Visualizzazione dei risultati con dimensioni superiori allo schermo | 10 |
| 5.2 Stampa di un prospetto | 11 |
| 5.3 Selezione di particolari colonne da una tabella | 11 |
| 5.4 Selezione di tutte le colonne da una tabella | 13 |
| 5.5 Selezione di particolari righe da una tabella | 14 |
| 5.6 Come evitare di selezionare righe duplicate | 15 |
| 5.7 Selezione di righe che soddisfano condizioni multiple | 17 |
| 5.8 Selezione di righe che soddisfano una di condizioni diverse | 18 |
| 5.9 Selezione di righe contenenti valori entro certi limiti | 20 |
| 5.10 Selezione di righe che contengono valori compresi in un elenco | 21 |
| 5.11 Selezione di righe contenenti una combinazione di caratteri | 21 |
| 5.12 Selezione di righe che soddisfano ad una condizione calcolata | 23 |
| 5.13 Selezione di informazioni calcolate dai dati della tabella | 24 |
| 5.14 Selezione di informazioni con l'uso di funzioni incorporate | 25 |
| 5.15 Selezione di costanti | 27 |
| 5.16 Ordinamento delle righe di una interrogazione | 28 |
| 5.17 Ordinamento delle righe in base ad un risultato calcolato | 29 |
| 5.18 Aggiornamento dei dati di una tabella | 30 |
| 5.18.1 Aggiornamento di righe e colonne di una tabella | 31 |
| 5.18.2 Cancellazione di righe da una tabella | 32 |
| 5.18.3 Aggiunta di una singola riga ad una tabella | 33 |
| 5.18.4 Aggiunta di righe ad una tabella | 33 |
| 6.0 Altri comandi ISQL | 36 |
| 6.1 Riemissione del comando corrente SQL | 36 |
| 6.2 Cambio del comando corrente SQL | 36 |
| 6.3 Uso di variabili posizionali nei comandi SQL | 37 |
| 6.4 Come memorizzare ed eseguire un comando SQL | 37 |

| | | |
|------------|---|-----------|
| 6.5 | Creazione di un prospetto dei risultati di una interrogazione | 39 |
| 6.6 | Modifica il titolo di una colonna visualizzata | 42 |
| 6.7 | Altre informazioni sulla formattazione dei prospetti | 44 |
| 6.8 | Creazione di titoli per i prospetti stampati | 47 |
| 6.9 | Controllo delle caratteristiche operative | 48 |
| 6.10 | Comandi per la ricerca di informazioni on-line | 50 |
| 7.0 | Bibliografia | 51 |

| | | |
|------|--|----|
| 1.0 | Definizione di database | 1 |
| 2.0 | Accesso al database | 4 |
| 3.0 | Interrogazione dei dati di una tabella | 12 |
| 4.0 | Visualizzazione dei risultati con diverse opzioni | 13 |
| 5.0 | Stampa di un prospetto | 14 |
| 6.0 | Selezione di particolari colonne di una tabella | 16 |
| 7.0 | Selezione di tutte le colonne di una tabella | 16 |
| 8.0 | Selezione di particolari righe di una tabella | 17 |
| 9.0 | Come evitare di selezionare righe duplicate | 17 |
| 10.0 | Selezione di righe che soddisfano piu' condizioni | 18 |
| 11.0 | Selezione di righe che soddisfano una condizione | 19 |
| 12.0 | Selezione di righe che non soddisfano a una condizione | 19 |
| 13.0 | Selezione di righe con valori compresi in un intervallo | 20 |
| 14.0 | Selezione di righe con valori compresi in una lista | 21 |
| 15.0 | Esempio di uso della funzione LIKE con il carattere '%' | 22 |
| 16.0 | Esempio di uso della funzione LIKE con il carattere '_' | 23 |
| 17.0 | Selezione di righe che soddisfano a una condizione calcolata | 24 |
| 18.0 | Visualizzazione di un valore calcolato | 25 |
| 19.0 | Uso delle funzioni AVG e SUM | 26 |
| 20.0 | Uso della funzione COUNT | 27 |
| 21.0 | Uso delle funzioni DISTINCT e COUNT | 27 |
| 22.0 | Come evidenziare una costante | 28 |
| 23.0 | Uso della funzione ORDER BY | 28 |
| 24.0 | Ordinamento di righe in base ad un risultato calcolato | 29 |
| 25.0 | Prospetto usato per operazioni di modifica | 40 |
| 26.0 | Prima modifica al prospetto | 41 |
| 27.0 | Seconda modifica al prospetto | 42 |
| 28.0 | Terza modifica al prospetto | 43 |
| 29.0 | Quarta modifica al prospetto | 43 |
| 30.0 | Prospetto usato per altri tipi di formattazione | 45 |
| 31.0 | Utilizzo della funzione GROUP | 45 |
| 32.0 | Utilizzo della funzione TOTAL | 46 |
| 33.0 | Utilizzo della funzione SUBTOTAL | 47 |
| 34.0 | Utilizzo del comando Set con l'opzione Decimal | 49 |

LISTA DELLE FIGURE

| | | |
|------------|--|----|
| Figura 1. | Una tabella SQL/DS | 1 |
| Figura 2. | Una macchina DB che lavora su un singolo data base | 4 |
| Figura 3. | Una macchina DB che possiede piu' di un data base | 5 |
| Figura 4. | Piu' macchine data base che accedono piu' data base | 6 |
| Figura 5. | Accesso a uno specifico database | 8 |
| Figura 6. | Sessione ISQL | 9 |
| Figura 7. | Selezione di colonne | 12 |
| Figura 8. | Selezione di tutte le colonne | 13 |
| Figura 9. | Selezione di righe | 14 |
| Figura 10. | Selezione di righe con valori duplicati | 16 |
| Figura 11. | Come evitare righe duplicate | 17 |
| Figura 12. | Selezione di righe che soddisfano piu' condizioni | 18 |
| Figura 13. | Selezione di righe che soddisfano una condizione | 19 |
| Figura 14. | Selezione di righe che non soddisfano a una condizione | 19 |
| Figura 15. | Selezione di righe con valori compresi in un intervallo | 20 |
| Figura 16. | Selezione di righe con valori compresi in una lista | 21 |
| Figura 17. | Esempio di uso della funzione LIKE con il carattere '%' | 22 |
| Figura 18. | Esempio di uso della funzione LIKE con il carattere '_' | 23 |
| Figura 19. | Selezione di righe che soddisfano a una condizione calcolata | 24 |
| Figura 20. | Visualizzazione di un valore calcolato | 25 |
| Figura 21. | Uso delle funzioni AVG e SUM | 26 |
| Figura 22. | Uso della funzione COUNT | 27 |
| Figura 23. | Uso delle funzioni DISTINCT e COUNT | 27 |
| Figura 24. | Come evidenziare una costante | 28 |
| Figura 25. | Uso della funzione ORDER BY | 28 |
| Figura 26. | Ordinamento di righe in base ad un risultato calcolato | 29 |
| Figura 27. | Prospetto usato per operazioni di modifica | 40 |
| Figura 28. | Prima modifica al prospetto | 41 |
| Figura 29. | Seconda modifica al prospetto | 42 |
| Figura 30. | Terza modifica al prospetto | 43 |
| Figura 31. | Quarta modifica al prospetto | 43 |
| Figura 32. | Prospetto usato per altri tipi di formattazione | 45 |
| Figura 33. | Utilizzo della funzione GROUP | 45 |
| Figura 34. | Utilizzo della funzione TOTAL | 46 |
| Figura 35. | Utilizzo della funzione SUBTOTAL | 47 |
| Figura 36. | Utilizzo del comando Set con l'opzione Decimal | 49 |

1.0 INTRODUZIONE

Structured Query Language/Data System (SQL/DS) e' un sistema di gestione di Basi di Dati (DBMS) di tipo relazionale.

Nel modello relazionale, i dati sono visti dall' utente sotto forma di tabelle a due dimensioni, che hanno un certo numero di colonne e un certo numero di righe. Ogni riga rappresenta un entry nella relazione e contiene un valore per ogni colonna. In questo modo, e lo vedremo ampiamente nel seguito, i dati memorizzati nel data base sono definiti ed acceduti in termini di tabelle e operazioni sulle tabelle.

L' utente puo' reperire, aggiungere, cancellare o aggiornare i dati contenuti nelle tabelle.

I dati, in SQL/DS, possono essere acceduti in modo:

- interattivo;
- batch, tramite il programma di utilita' Data Base Services (DBS);
- interattivo o batch, tramite programmi applicativi scritti in FORTRAN, PL/I, COBOL, ASSEMBLER.

Questa guida descrive alcuni dei comandi necessari per usare SQL/DS in modo interattivo da un terminale video. Per una descrizione completa dei comandi disponibili e delle possibilita' offerte, si rimanda al manuale SQL/Data System Terminal Users' Reference.

SQL/DS presenta tutti i dati sotto forma di tabelle. I dati appaiono percio' nel formato di righe e colonne. Per esempio:

| MODEL | YEAR | COLOR |
|--------|------|--------|
| Alfa | 1970 | verde |
| Lancia | 1973 | rosso |
| Fiat | 1980 | bianca |

Figura 1. Una tabella SQL/DS

La tabella e' costituita da 3 righe e 3 colonne. Il nome della tabella 'CARS' viene definito al momento della sua creazione insieme ai nomi delle colonne di cui la tabella sara' composta. In genere per memorizzare tutte le informazioni a disposizione sara' necessario definire piu' di una tabella. Nella descrizione dei comandi SQL faremo rife-

rimento a tre tabelle che potrebbero essere usate per il controllo dell'inventario di una piccola azienda manifatturiera.

- **INVENTORY:** che indica il numero di inventario del pezzo acquistato o immagazzinato (PARTNO), la descrizione (DESCRIPTION) e la quantita' disponibile (QONHAND);
- **SUPPLIERS:** dove sono riportati il numero del fornitore (SUPPNO), il nome (NAME) e l'indirizzo delle ditte fornitrici (ADDRESS);
- **QUOTATIONS:** descrive il numero dei fornitori (SUPPNO), il codice dei pezzi approvvigionati per ogni fornitore (PARTNO), il prezzo corrente (PRICE), il tempo di consegna in giorni (DELIVERY_TIME) e la quantita' ordinata (QONORDER).

Le tabelle sono state create nel data base di sistema e quindi risultano accessibili a tutti gli users per operazioni di query.

Come abbiamo gia' detto, l'SQL fornito con SQL/DS e' usato per interrogare, aggiungere, cancellare o aggiornare informazioni memorizzate nel Data Base. L'emissione dei comandi da un terminale video e' resa possibile dalla funzione ISQL che abilita il terminale dello user a lavorare direttamente con i dati SQL/DS. Riportiamo una lista delle possibilita' offerte dai comandi ISQL.

- Controllare le visualizzazioni dei dati che risultano da una interrogazione;
- Stampare prospetti;
- Introdurre i dati;
- Ottenere informazioni da un terminale (Help);
- Memorizzare comandi SQL;
- Listare le caratteristiche operative;
- Usare routines.

2.0 ARCHITETTURA SQL/DS

Prima di descrivere le possibili configurazioni di un sistema SQL/DS, e' necessario ricordare i modi di impiego del sistema. Esso puo' operare in tre differenti modi:

- Multiple User Mode;
- Single User Mode;
- Multiple Data Base Mode.

Il sistema lavora in Multiple User Mode allorché uno o piu' users o applicazioni accedono in modo concorrente allo stesso data base. Quindi mentre SQL/DS sta lavorando sulla propria macchina virtuale le applicazioni sono eseguite sulle varie macchine virtuali di utente. Le applicazioni comunicano con SQL/DS attraverso una communication facility (link) trasparente alle applicazioni stesse. Questo e' il modo standard di lavorare di SQL/DS in quanto oltre ad avere il sistema isolato sulla propria macchina virtuale, con vantaggi sulla sicurezza ed integrita', consente a piu' utenti di accedere ai dati e risorse in modo simultaneo.

Nel caso in cui si ha una singola applicazione o utility che viene eseguita sulla stessa macchina virtuale di SQL/DS si dice che il sistema lavora in Single User Mode. Come si puo' ben capire questo e' un modo particolare di lavorare per SQL/DS e viene eseguito solo in relazione ad una precisa richiesta dell'applicazione o utility.

Infine il terzo modo si ha quando alcuni SQL/DS data bases operano nello stesso tempo. Lo scopo di questo e' consentire a differenti gruppi di users di avere il proprio data base. Per esempio un gruppo puo' avere un data base di test mentre un altro ha un data base di produzione.

In base alle caratteristiche di funzionamento di SQL/DS si possono individuare altrettanti environment di data base dove il sistema puo' essere operativo. Vediamo un esempio di configurazione con SQL/DS operante in Multiple User Mode.

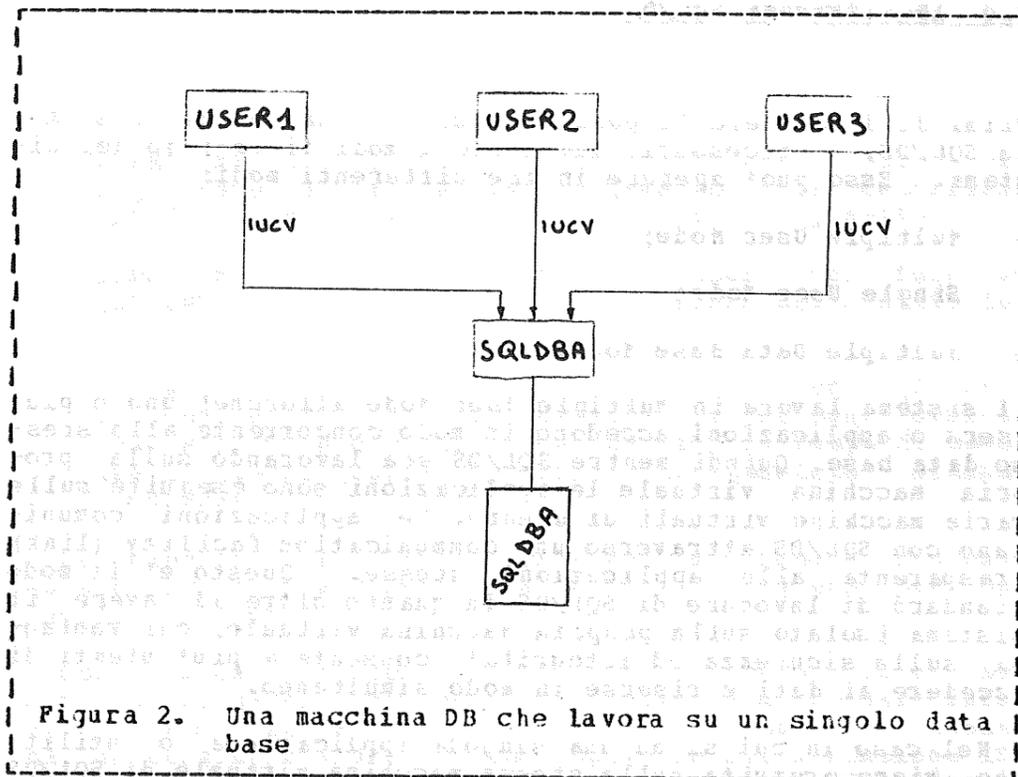


Figura 2. Una macchina DB che lavora su un singolo data base

In questo esempio sono riportate tre macchine virtuali di utenti (USER1, USER2 e USER3) che comunicano con la macchina data base SQLDBA. Sulle macchine users e' stato creato precedentemente una parte di codice SQL/DS che va sotto il nome di 'Resource Manager' il quale usa la VM/SP IUCV per comunicare con il codice base che e' attivo sulla macchina virtuale SQLDBA. Il collegamento logico e' creato eseguendo la procedura SQLINIT che inizializza la macchina virtuale.

Ogni utente prima di accedere a SQL/DS, come descritto nel paragrafo 'Accesso al DB/DC' dovra' eseguire il comando SQLINIT specificando il nome del data base a cui vuole accedere durante la sessione di lavoro. Stabilito il collegamento viene inviato sul proprio terminale il messaggio che identifica la macchina data base proprietaria del data base specificato. A questo punto, allorché lo user lancia il primo comando SQL il Resource Manager stabilisce la connessione IUCV con la macchina data base. E' importante notare che la procedura avra' un risultato positivo se alla macchina data base e' acceduto il data base al quale ci si vuol collegare. In caso contrario lo user riceverà il codice di errore SQL 943. Nel nostro caso essendo l'environment costituito da un singolo data base questo problema non dovrebbe verificarsi.

La semplice configurazione permette allo user di eseguire una sola volta la procedura SQLINIT la quale crea i moduli di inizializzazione che saranno mantenuti sul minidisco. Solamente nel caso in cui la configurazione e' Multiple Data Base o Multiple Data Base Machines lo user dovrà rieseguire la procedura SQLINIT specificando quale data base vorrà accedere.

Una seconda possibile configurazione in cui può operare SQL/DS e' rappresentata dalla fig. seguente:

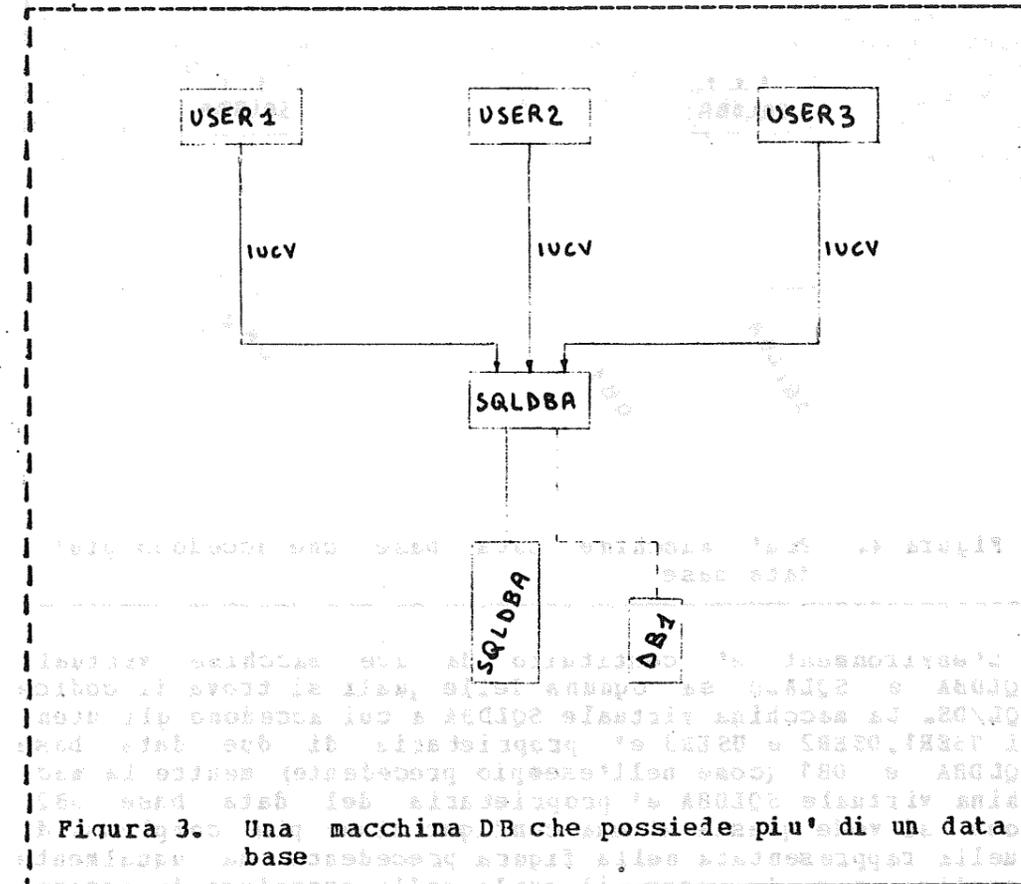


Figura 3. Una macchina DB che possiede più di un data base

L'environment di questo esempio differisce dal precedente per l'aggiunta di un data base DB1 alla macchina virtuale proprietaria del data base SQLDBA.

In questo caso lo user dovrà inizializzare la sua macchina virtuale a seconda di quale data base vorrà accedere. E' evidente che nel caso in cui lo user chieda di accedere a quel data base che non e' stato linkato dalla macchina virtuale SQLDBA riceverà dal Resource Manager il codice di errore SQL 943.

Una ulteriore configurazione, abbastanza comune, puo' essere cosi' rappresentata:

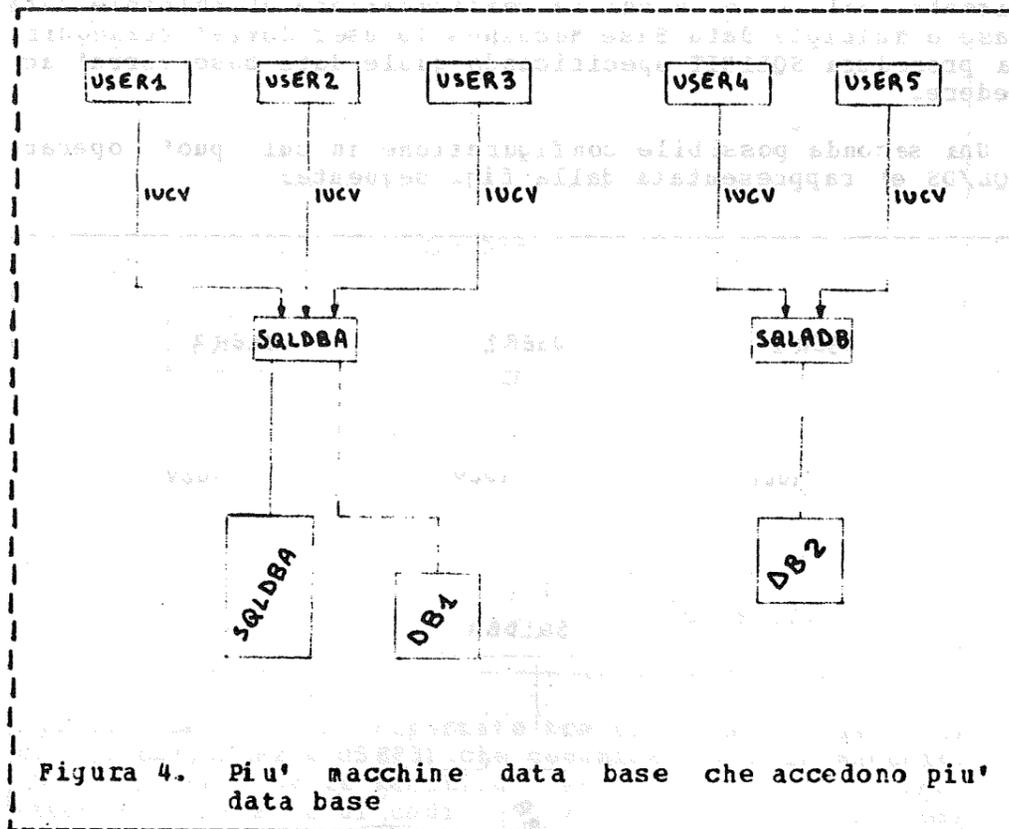


Figura 4. Più macchine data base che accedono più data base

L'environment e' costituito da due macchine virtuali SQLDBA e SQLADB su ognuna delle quali si trova il codice SQL/DS. La macchina virtuale SQLDBA a cui accedono gli utenti USER1, USER2 e USER3 e' proprietaria di due data base SQLDBA e DB1 (come nell'esempio precedente) mentre la macchina virtuale SQLADB e' proprietaria del data base DB2. Come si vede questa e' una configurazione più complessa di quella rappresentata nella figura precedente ma ugualmente semplice per lo user il quale nella procedura di accesso dovrà ricordarsi solamente il nome del DB con il quale collegarsi. Vale anche in questo caso la condizione che se viene richiesto di accedere un DB che non e' stato linkato verterà inviato il codice di errore SQL 943.

3.0 DEFINIZIONE DI DBSPACE

A completamento delle caratteristiche generali di un sistema SQL/DS vediamo un aspetto molto importante dal punto di vista utente del concetto più generale di data base, e rimandiamo alla II parte per maggiori dettagli.

Abbiamo detto che un data base SQL/DS e' un insieme di dati definiti dallo user che vanno sotto il nome di tabelle. Or bene, questi dati sono raggruppati in insiemi logici chiamati DBSPACE con la condizione che ogni tabella e i suoi eventuali indici debbono appartenere allo stesso DBSPACE. I vari DBSPACE vengono quindi definiti durante la fase di generazione del sistema SQL/DS e messi a disposizione degli utenti dove potranno memorizzare i loro dati al momento di creazione delle relazioni.

In definitiva un DBSPACE puo' essere visto come una parte del data base che puo' contenere una o più tabelle con i relativi indici.

9.0 ACCESSO AL DB/DC

Prima di poter usare SQL tramite un terminale video tipo 327x e' necessario rivolgersi alle appropriate strutture per ottenere:

- una **userid** e relativa **password**: per accedere al sistema VM e successivamente a SQL/DS;
- una **autorizzazione**: per avere libero accesso al DB/DC.

Definita la macchina virtuale e' possibile eseguire una sessione a terminale; la quale consiste nel collegarsi al sistema VM, avviare ISQL, eseguire una o piu' elaborazioni e chiudere la sessione ISQL.

Infine per accedere al DB/DC occorre:

1. effettuare il LOGIN della macchina virtuale autorizzata;
2. eseguire la procedura SQLDS.

Per accedere a uno specifico Database, e' necessario inizializzare la macchina virtuale con il comando:

```
SQLINIT DB(<databasename>)
```

Figura 5. Accesso a uno specifico database

Poiche' le tabelle di esempio sono nel database SQLDBA, per effettuare una sessione di prova occorrera' digitare:

```
SQLINIT DB(SQLDBA).
```

Il comando SQLINIT crea sul disco A (normalmente il 191) della macchina virtuale due moduli (ARISISBT e ARISRMBT) usati per caricare ISQL e il codice del Resource Manager.

A questo punto la M.V. e' pronta per comunicare con SQL/DS. Per accedere ai dati e' necessario dare il comando ISQL; per terminare la sessione e' necessario il comando EXIT. Con questo comando l'utente ritorna in ambiente CMS.

```
ISQL
```

```
comandi SQL
```

```
EXIT
```

Figura 6. Sessione ISQL

5.0 INTERROGAZIONE DEI DATI DI UNA TABELLA

L'interrogazione e' l'attivita' piu' comune che viene eseguita sui dati memorizzati in un data base. Essa consente di accedere ai dati contenuti in una o piu' tabelle del data base. I punti seguenti descrivono come andare a leggere i dati memorizzati in SQL/DS usando le tabelle riportate in appendice.

Il comando Select ha la seguente struttura formale:

```
SELECT (ALL | DISTINCT) {lista delle colonne | *}  
FROM (nome user.)nome della tabella  
{WHERE condizione di ricerca}  
{GROUP BY nome di colonna (HAVING condizione di ricerca)}  
{ORDER BY nome di colonna {ASC | DESC} (, colspec {ASC |  
DESC})
```

5.1 VISUALIZZAZIONE DEI RISULTATI CON DIMENSIONI SUPERIORI ALLO SCHERMO

Accade spesso che i risultati di una interrogazione sono superiori alle capacita' del video. Sono disponibili alcuni comandi per muoversi attraverso i risultati.

- FORWARD 10 (PF8) per spostare l'osservazione dei risultati in avanti di 10 righe;
- FORWARD MAX per vedere le ultime 20 righe del risultato;
- BACKWARD 10 (PF7) per spostare l'osservazione dei risultati indietro di 10 righe;
- BACKWARD MAX per andare a vedere le prime 20 righe;
- RIGHT 10 (PF11) per spostare la finestra di 10 colonne a destra del risultato;
- LEFT 10 (PF10) per spostare la finestra di 10 colonne a sinistra del risultato;
- COLUMN 1 sposta il risultato a partire dalla prima colonna.

5.2 STAMPA DI UN PROSPETTO

Per stampare il risultato di una interrogazione e' sufficiente dopo aver fatto la query, dare il comando

```
print
```

oppure premere il tasto PF4.

Per ottenere piu' copie del prospetto usare la parola chiave COPIES mentre per inviare l'uscita in una particolare classe di stampa usare la parola chiave CLASS del comando PRINT. Cioe':

```
print copies 3 class s
```

per stampare 3 copie del prospetto ed indirizzare l'uscita sulla stampante alla quale e' associata la classe S. E' importante notare che il comando Print agisce su tutto il risultato dell'interrogazione e non solo su cio' che si vede a terminale. Inoltre il prospetto di stampa inizia con la colonna che si trova correntemente al margine sinistro dello schermo.

5.3 SELEZIONE DI PARTICOLARI COLONNE DA UNA TABELLA

Il comando usato per ritrovare i dati memorizzati in una tabella e' il comando Select. Esso consiste di due parti:

- La clausola SELECT che elenca le colonne da selezionare;
- La clausola FROM che indica la tabella dalla quale selezionare le colonne.

Se vogliamo selezionare tutte le colonne della tabella Inventory e' necessario dare il comando:

```
select partno, qonhand, description from inventory
```

Premendo il tasto ENTER compare il seguente schermo:

Interrogazione dei dati di una tabella

| PARTNO | QONHAND | DESCRIPTION |
|--------|---------|-------------|
| 207 | 75 | GEAR |
| 209 | 50 | CAM |
| 221 | 650 | BOLT |
| 222 | 1250 | BOLT |
| 231 | 700 | NUT |
| 232 | 1100 | NUT |
| 241 | 6000 | WASHER |
| 285 | 350 | WHEEL |
| 295 | 85 | BELT |
| 301 | 300 | FUSE |
| 302 | 367 | FUSE |
| 310 | 150 | PIN |
| 311 | 292 | PIN |
| 315 | 421 | SWITCH |
| 316 | 342 | SWITCH |
| 320 | 0 | RELAY |
| 321 | 0 | RELAY |
| 322 | 0 | RELAY |
| 323 | 0 | RELAY |

Figura 7. Selezione di colonne

In questo caso sono state selezionate le tre colonne di cui e' costituita la tabella; ma si possono selezionare qualsiasi numero di colonne desiderato. Dobbiamo notare che la sequenza in cui le colonne vengono visualizzate e' la stessa in cui i nomi di colonna compaiono nella clausola Select. E' possibile scegliere l'ordine sinistra-destra delle colonne nei risultati della interrogazione tramite la posizione nella quale viene disposto il nome della colonna nella clausola Select.

Terminata la consultazione della tabella; per rimuovere dal video il risultato della interrogazione battere END e premere il tasto ENTER oppure premere il tasto PF3.

Puo' capitare che alcuni comandi sono troppo lunghi per poter essere scritti su un'unica riga. In questo caso si puo' continuare il comando usando il carattere di continuazione rappresentato da un trattino (-) dopo aver lasciato uno spazio al termine della riga.

Nell'esempio riportato avremo:

```
select partno,qonhand,description -
from inventory
```

Il risultato della interrogazione sara' identico a quello precedente.

5.4 SELEZIONE DI TUTTE LE COLONNE DA UNA TABELLA

Nell'esempio precedente le informazioni sono state selezionate elencando i nomi delle colonne desiderate. Se devono essere consultate tutte le colonne di una tabella e' sufficiente sostituire un asterisco ai nomi delle colonne.

```
select * from inventory
```

Il comando fornisce il seguente risultato:

| PARTNO | DESCRIPTION | QONHAND |
|--------|-------------|---------|
| 207 | GEAR | 75 |
| 209 | CAM | 50 |
| 221 | BOLT | 650 |
| 222 | BOLT | 1250 |
| 231 | NUT | 700 |
| 232 | NUT | 1100 |
| 241 | WASHER | 6000 |
| 285 | WHEEL | 350 |
| 295 | BELT | 85 |
| 301 | FUSE | 300 |
| 302 | FUSE | 367 |
| 310 | PIN | 150 |
| 311 | PIN | 292 |
| 315 | SWITCH | 421 |
| 316 | SWITCH | 342 |
| 320 | RELAY | 0 |
| 321 | RELAY | 0 |
| 322 | RELAY | 0 |
| 323 | RELAY | 0 |

Figura 8. Selezione di tutte le colonne

Dalla figura si vede che l'ordine in cui le colonne sono visualizzate e' lo stesso in cui compaiono nella tabella. Invece nell'esempio precedente esse comparivano nell'ordine in cui erano specificate nella clausola Select.

5.5 SELEZIONE DI PARTICOLARI RIGHE DA UNA TABELLA

Nell'esempio precedente sono state selezionate tutte le righe della tabella Inventory. Supponiamo di voler selezionare soltanto le righe con valore della descrizione uguale a BOLT. Per far cio' basta aggiungere la clausola WHERE al comando Select.

La clausola Where se usata segue sempre la clausola Select. La parola Bolt deve essere racchiusa tra apici per indicare al sistema che si riferisce a dati di tipo carattere. I riferimenti a dati di tipo alfanumerico usati nei comandi SQL devono essere racchiusi tra apici, mentre i dati di tipo numerico non devono essere racchiusi tra apici.

La sintassi formale della clausola WHERE e' la seguente:

WHERE nome-colonna operatore stringa alfanum.ica/val.numerico

Quindi la clausola rappresenta la condizione di ricerca in quanto fa si che vengano selezionate soltanto quelle righe che soddisfano la condizione specificata.

Nella condizione di ricerca puo' essere usato uno qualsiasi dei seguenti operatori:

- = (uguale a)
- ≠ (diverso da)
- > (maggiore di)
- >= (maggiore di o uguale a)
- < (minore di)
- <= (minore di o uguale a)

In questo caso il comando sara':

```
select * from inventory where description='bolt'
```

Il risultato sara':

| PARTNO | DESCRIPTION | QONHAND |
|--------|-------------|---------|
| 221 | BOLT | 650 |
| 222 | BOLT | 1250 |

Figura 9. Selezione di righe

Interrogazione dei dati di una tabella

5.6 COME EVITARE DI SELEZIONARE RIGHE DUPLICATE

Abbiamo visto il comando per selezionare una o piu' colonne di una tabella. Per esempio impostando il comando:

```
select partno from quotations
```

si ha il seguente risultato:

Interrogazione dei dati di una tabella

Il risultato e' espresso dalla figura:

| SUPPNO | PARTNO | PRICE | DELIVERY_TIME | QONORDER |
|--------|--------|-------|---------------|----------|
| 51 | 221 | 0.30 | 10 | 50 |
| 54 | 221 | 0.10 | 30 | 150 |

Figura 12. Selezione di righe che soddisfano piu' condizioni

Cioe' le condizioni di ricerca sono collegate con la parola AND e questo significa che nella selezione delle righe debbono essere soddisfatte entrambe le condizioni. E' possibile aggiungere qualsiasi numero di condizioni a patto di collegare queste con la parola AND.

5.8 SELEZIONE DI RIGHE CHE SODDISFANO UNA DI CONDIZIONI DIVERSE

Allorché si desidera selezionare righe di una tabella che soddisfano ad almeno una di un insieme di condizioni l'operatore logico da usare e' OR. Esempio per selezionare il fornitore numero 54 o il fornitore numero 64 il comando e':

```
select * from quotations -
where suppno=54 or suppno=64
```

Il risultato e' il seguente:

| SUPPNO | PARTNO | PRICE | DELIVERY_TIME | QONORDER |
|--------|--------|-------|---------------|----------|
| 54 | 209 | 18.00 | 21 | 0 |
| 54 | 221 | 0.10 | 30 | 150 |
| 54 | 231 | 0.04 | 30 | 200 |
| 54 | 241 | 0.02 | 30 | 200 |
| 64 | 207 | 29.00 | 14 | 20 |
| 64 | 209 | 19.50 | 7 | 7 |

Figura 13. Selezione di righe che soddisfano una condizione

Se invece e' necessario identificare condizioni di ricerca che non devono essere soddisfatte l'operatore logico da usare e' NOT. Per selezionare le righe della tabella Quotations relative al fornitore 64 e le righe non relative al pezzo 209 che riguardano il fornitore 54, il comando e':

```
select * from quotations -
where suppno=64 or suppno=54 and not partno=209
```

Il risultato e':

| SUPPNO | PARTNO | PRICE | DELIVERY_TIME | QONORDER |
|--------|--------|-------|---------------|----------|
| 54 | 221 | 0.10 | 30 | 150 |
| 54 | 231 | 0.04 | 30 | 200 |
| 54 | 241 | 0.02 | 30 | 200 |
| 64 | 207 | 29.00 | 14 | 20 |
| 64 | 209 | 19.50 | 7 | 7 |

Figura 14. Selezione di righe che non soddisfano a una condizione

5.9 SELEZIONE DI RIGHE CONTENENTI VALORI ENTRO CERTI LIMITI

Vogliamo adesso selezionare dalla tabella Quotations le righe il cui prezzo e' compreso tra 0.30 \$ e 10 \$. Il comando e':

```
select * from quotations -
where price between .30 and 10.00
```

Il risultato e':

| SUPPNO | PARTNO | PRICE | DELIVERY_TIME | QONORDER |
|--------|--------|-------|---------------|----------|
| 51 | 221 | 0.30 | 10 | 50 |
| 57 | 295 | 8.50 | 21 | 24 |
| 70 | 315 | 1.00 | 8 | 0 |
| 70 | 316 | 1.01 | 7 | 0 |
| 70 | 320 | 3.00 | 18 | 0 |
| 70 | 321 | 3.10 | 17 | 0 |
| 70 | 322 | 3.15 | 16 | 0 |
| 71 | 315 | 1.10 | 3 | 65 |
| 71 | 316 | 1.11 | 16 | 0 |
| 71 | 320 | 3.20 | 18 | 0 |
| 71 | 321 | 3.19 | 19 | 0 |
| 71 | 322 | 3.21 | 20 | 0 |
| 72 | 315 | 1.49 | 15 | 0 |
| 72 | 316 | 1.50 | 14 | 850 |
| 72 | 320 | 3.50 | 11 | 0 |
| 72 | 321 | 3.51 | 12 | 0 |
| 72 | 322 | 3.49 | 13 | 0 |
| 72 | 323 | 3.52 | 14 | 0 |

Figura 15. Selezione di righe con valori compresi in un intervallo

Percio' con la clausola Between vengono selezionate le righe che cadono in un certo intervallo. Se invece si vogliono selezionare le righe i cui valori di prezzo non cadono nell'intervallo specificato la clausola sara':

```
where price not between .30 and 10.00
```

5.10 SELEZIONE DI RIGHE CHE CONTENGONO VALORI COMPRESI IN UN ELENCO

Tramite la funzione IN e' possibile selezionare le righe che contengono un particolare valore (o stringa di caratteri) in corrispondenza di un particolare attributo (colonna) confrontandolo con una lista di valori (o stringa di caratteri).

Così' per selezionare dalla tabella Inventory le righe che riguardano dadi (nut), bulloni (bolt) e rondelle (washer) dare il comando:

```
select * from inventory -
where description in ('nut','bolt','washer')
```

Il risultato e':

| PARTNO | DESCRIPTION | QONHAND |
|--------|-------------|---------|
| 221 | BOLT | 650 |
| 222 | BOLT | 1250 |
| 231 | NUT | 700 |
| 232 | NUT | 1100 |
| 241 | WASHER | 6000 |

Figura 16. Selezione di righe con valori compresi in una lista

La funzione In puo' essere modificata da NOT per selezionare le righe che contengono un valore che non e' nella lista.

5.11 SELEZIONE DI RIGHE CONTENENTI UNA COMBINAZIONE DI CARATTERI

La funzione LIKE consente di selezionare le righe che contengono una combinazione di caratteri in una certa colonna.

Per selezionare dalla tabella Suppliers il nome o l'indirizzo dei fornitori con sede a New York (NY), il comando e':

```
select name,address from suppliers -
where address like '%NY'
```

Il risultato sara':

| NAME | ADDRESS |
|----------------|-------------------------------|
| VESUVIUS, INC. | 512 ANCIENT BLVD., POMPEII NY |

Figura 17. Esempio di uso della funzione LIKE con il carattere '%'

La figura, in relazione alla query effettuata, mostra un risultato parziale in quanto la dimensione della colonna Address e' maggiore del campo visualizzato. Per avere l'intera informazione e' necessario dare il comando:

```
set varchar n
```

dove n indica la larghezza desiderata del display delle colonne definite VARCHAR nella tabella (default 20).

Nella condizione espressa dalla funzione Like il carattere % rappresenta qualsiasi stringa di zero o piu' caratteri. In questo caso la funzione specifica di selezionare quelle righe che contengono i caratteri NY al termine della rispettiva colonna Address. Volendo invece scegliere quelle righe che contengono i caratteri NY ovunque nella colonna Address la codifica sarebbe '%NY%'.

Un altro carattere che ha uno speciale significato nella funzione LIKE e' la sottolineatura. Esso rappresenta un singolo carattere posizionale e dice al sistema di non analizzare il carattere in quella posizione.

Volendo selezionare dalla tabella Inventory in corrispondenza della colonna Description le righe contenenti una stringa di caratteri che inizia con la lettera 'b' ed e' lunga quattro caratteri inviare il comando:

```
select * from inventory where description like 'b____'
```

Interrogazione dei dati di una tabella

22

Il risultato e' il seguente:

| PARTNO | DESCRIPTION | QONHAND |
|--------|-------------|---------|
| 221 | BOLT | 650 |
| 222 | BOLT | 1250 |
| 295 | BELT | 85 |

Figura 18. Esempio di uso della funzione LIKE con il carattere '_'

La funzione puo' essere modificata da NOT; in questo caso saranno selezionate le righe in cui sono presente caratteri diversi da quelli specificati.

5.12 SELEZIONE DI RIGHE CHE SODDISFANO AD UNA CONDIZIONE CALCOLATA

La condizione di ricerca espressa dalla clausola Where, puo' contenere una espressione che consente di selezionare righe basate su valori calcolati.

L'espressione nella clausola Where si compone della combinazione di nomi di colonna e di costanti (valori numerici) collegati da operatori aritmetici. Le operazioni aritmetiche che si possono eseguire sono:

- + (addizione)
- (sottrazione)
- * (moltiplicazione)
- / (divisione)

Per sapere quali ordini nella tabella Quotations avranno un valore superiore a 100 dollari il comando e':

```
select * from quotations -
where price*qonorder > 100.00
```

Interrogazione dei dati di una tabella

23

Il risultato e':

| SUPPNO | PARTNO | PRICE | DELIVERY_TIME | QONORDER |
|--------|--------|-------|---------------|----------|
| 57 | 295 | 8.50 | 21 | 24 |
| 64 | 207 | 29.00 | 14 | 20 |
| 64 | 209 | 19.50 | 7 | 7 |
| 70 | 311 | 41.00 | 31 | 100 |
| 72 | 310 | 43.10 | 23 | 300 |
| 72 | 316 | 1.50 | 14 | 850 |

Figura 19. Selezione di righe che soddisfano a una condizione calcolata

5.13 SELEZIONE DI INFORMAZIONI CALCOLATE DAI DATI DELLA TABELLA

Se vogliamo evidenziare il risultato della condizione calcolata nel precedente esempio sara' sufficiente specificare la condizione stessa nella clausola Select. Cioe':

```
select price*qonorder from quotations -
where price*qonorder > 100.00
```

Il risultato vale:

| EXPRESSION 1 |
|--------------|
| 204.00 |
| 580.00 |
| 136.50 |
| 4100.00 |
| 12930.00 |
| 1275.00 |

Figura 20. Visualizzazione di un valore calcolato

Nella figura compare sotto la colonna 'EXPRESSION 1' il valore che risulta dalla moltiplicazione di price e qonorder. Per cambiare il titolo vedere il comando Format.

5.14 SELEZIONE DI INFORMAZIONI CON L'USO DI FUNZIONI INCORPORATE

Ai valori dei vari attributi di una tabella e' possibile applicare le seguenti funzioni di calcolo AVG,SUM,MIN,MAX e COUNT.

- **AVG:** la funzione calcola il valore medio della colonna che e' stata selezionata. La funzione puo' quindi essere applicata solamente a colonne di tipo numerico.
- **SUM:** questa funzione calcola il totale della colonna selezionata e puo' quindi essere applicata soltanto alle colonne di tipo numerico.
- **MIN:** trova il piu' piccolo valore della colonna selezionata. Questa funzione puo' essere applicata a tutte le colonne di una tabella qualunque sia il tipo di definizione.
- **MAX:** trova il piu' grande valore della colonna selezionata. Anche questa funzione puo' essere applicata alle colonne di qualsiasi tipo e come per la funzione MIN se viene applicata alle colonne del tipo CHAR o VARCHAR per la ricerca del valore max viene seguito l'ordinamento alfanumerico.
- **COUNT:** questa funzione puo' essere usata in due modi diversi:

1. usando un asterisco in luogo del nome di colonna, per contare il numero di righe che soddisfano ad una particolare condizione di ricerca;
2. oppure specificando il nome della colonna preceduto dalla clausola Distinct per contare il numero di righe diverse che soddisfano una condizione di ricerca.

E' necessario notare che se una funzione e' applicata ad una colonna nella clausola Select e' necessario specificare una funzione per tutte le colonne nella Select in quanto la funzione incorporata estrae per le colonne a cui si riferisce un unico valore. (eccezione Cap. 4 informazioni riassuntive per gruppi)

Supponiamo di voler conoscere il prezzo medio e la quantita' totale in ordine per il pezzo numero 221. Il comando relativo e':

```
select avg(price),sum(qonorder) from quotations -
where partno=221
```

Il risultato sara':

| AVG(PRICE) | SUM(QONORDER) |
|----------------|---------------|
| 0.200000000000 | 200 |

Figura 21. Uso delle funzioni AVG e SUM

Per trovare quante righe ci sono nella tabella Quotations per il pezzo 201 il comando e':

```
select count(*) from quotations where partno=221
```

con il risultato:

| COUNT(EXPRESSION 1) |
|---------------------|
| 3 |

Figura 22. Uso della funzione COUNT

Se invece vogliamo trovare quanti numeri di pezzi diversi ci sono nella tabella Quotations allora:

```
select count (distinct partno) from quotations
```

Il risultato sara':

| COUNT(DISTINCT PARTNO) |
|------------------------|
| 19 |

Figura 23. Uso delle funzioni DISTINCT e COUNT

5.15 SELEZIONE DI COSTANTI

Il comando Select puo' essere usato anche per fornire una colonna contenente una certa costante.

```
select avg(price),sum(qonorder), 'partno221' -
from quotations where partno=221
```

Il risultato e':

| AVG (PRICE) | SUM (QONORDER) | EXPRESSION 1 |
|----------------|----------------|--------------|
| 0.200000000000 | 200 | PARTNO221 |

Figura 24. Come evidenziare una costante

Cioe' viene fornita una colonna che indica il numero di pezzo a cui la select si riferisce.

5.16 ORDINAMENTO DELLE RIGHE DI UNA INTERROGAZIONE

Volendo ordinare le righe ottenute come risultato di una operazione di interrogazione e' necessario aggiungere la clausola ORDER BY al comando Select. La clausola e' posizionale, deve essere specificata come ultimo valore nel comando.

Ad esempio tramite il comando:

```
select suppno,partno,price from quotations -
where partno=221 order by partno,price
```

vengono selezionate dalla tabella Quotations il numero di fornitori, il numero dei pezzi ed il prezzo per i pezzi 221 ordinati secondo il numero del pezzo e all'interno di questi ordinati per valore del prezzo. Il risultato sara':

| SUPPNO | PARTNO | PRICE |
|--------|--------|-------|
| 54 | 221 | 0.10 |
| 61 | 221 | 0.20 |
| 51 | 221 | 0.30 |

Figura 25. Uso della funzione ORDER BY

Come si puo' vedere il valore di default dell'ordinamento e' ascendente. Per avere l'ordinamento in modo discendente deve essere specificato DESC nella clausola ORDER BY.

5.17 ORDINAMENTO DELLE RIGHE IN BASE AD UN RISULTATO CALCOLATO

Supponiamo di voler conoscere nella tabella Quotations tutti gli ordini di valore maggiore di 100 dollari ordinati in modo discendente per valore degli ordini. Il comando e':

```
select suppno,partno,price*qonorder,qonorder -
from quotations -
where price*qonorder > 100.00 order by 3 desc
```

Nella Select e' stata specificata una colonna price*qonorder che non esiste nella tabella Quotations ma viene calcolata come prodotto delle colonne Price e Qonorder. Percio' volendo la stampa ordinata secondo questo prodotto si dovra' specificare la posizione dell'attributo nella lista della Select. La clausola Order By contiene infine l'opzione Desc per avere l'ordinamento in modo discendente.

Il risultato e' il seguente:

| SUPPNO | PARTNO | EXPRESSION 1 | QONORDER |
|--------|--------|--------------|----------|
| 72 | 310 | 12930.00 | 300 |
| 70 | 311 | 4100.00 | 100 |
| 72 | 316 | 1275.00 | 850 |
| 64 | 207 | 580.00 | 20 |
| 57 | 295 | 204.00 | 24 |
| 64 | 209 | 136.50 | 7 |

Figura 26. Ordinamento di righe in base ad un risultato calcolato

5.18. AGGIORNAMENTO DEI DATI DI UNA TABELLA.

Prima di vedere come e' possibile modificare (o cancellare) i dati all'interno di una tabella e' necessario avere chiaro quando tali aggiornamenti devono esser fatti.

Con il comando:

```
set autocommit off
```

viene aperta una "unita' logica di lavoro" tramite la quale e' possibile controllare le modifiche prima di renderle definitive. Le modifiche fatte nell'ambito dell'unita' logica saranno percio' provvisorie.

Terminato il lavoro di aggiornamento il comando:

```
commit work
```

chiude l'unita' logica rendendo definitive le modifiche effettuate.

Tramite il comando:

```
rollback work
```

e' possibile invece annullare tutte le modifiche fatte all'inizio allorché si verifica un errore di aggiornamento nell'ambito della unita' logica di lavoro.

E' opportuno mettere in evidenza che questa operazione e' valida all'interno di una unita' logica e una volta emesso il comando "commit work" l'unico modo per ripristinare la situazione di partenza e' di ricorrere ai nastri di Backup del Data Base.

Infine nel caso in cui si desidera che le modifiche fatte abbiano effetto immediato non appena viene introdotto il comando e' necessario lavorare con autocommit on. Il comando relativo sara':

```
set autocommit on
```

Interrogazione dei dati di una tabella

30

5.18.1. Aggiornamento di righe e colonne di una tabella

Vogliamo innanzitutto fare in modo da controllare gli aggiornamenti che andremo a fare sulle tabelle. Dovremo percio' dare il comando:

```
set autocommit off
```

che apre una unita' logica di lavoro.

Il comando per eseguire l'aggiornamento delle righe (e colonne) di una tabella e' UPDATE. Per spiegare la sintassi del comando facciamo riferimento ad un esempio sulle solite tabelle esemplificative.

Supponiamo che sia arrivato l'ordine emesso al fornitore 61 per il pezzo 222. Per aggiornare la tabella Inventory al fine di riflettere il movimento emettere:

```
update inventory -  
set qonhand=1450 -  
where partno=222
```

Cioe' il comando si compone di 3 parti:

1. la clausola UPDATE che indica la tabella da aggiornare;
2. la clausola SET che specifica le colonne da aggiornare secondo la struttura seguente:

```
SET nome colonna=nuovo valore
```

3. la clausola WHERE che specifica la riga da aggiornare.

Adesso e' necessario aggiornare la tabella Quotations per annullare la quantita' del pezzo ordinato e ricevuto. Il comando e':

```
update quotations -  
set qonorder=0 -  
where partno=222 and suppno=61
```

A questo punto richiamando le due tabelle andiamo a verificare se gli aggiornamenti sono stati eseguiti in modo corretto. In caso affermativo per rendere definitive le correzioni dare il comando:

Interrogazione dei dati di una tabella

31

commit work

Se invece le modifiche fatte non sono corrette, per ripristinare la situazione iniziale il comando da eseguire e':

rollback work

Per uscire dallo stato di modifiche differite dare il comando

set autocommit on

Nel caso in cui sia necessario aggiornare una intera colonna bastera' omettere la clausola Where nel comando Update. Ad esempio per aumentare il tempo di consegna di 7 giorni per tutti i pezzi il comando e':

```
update quotations -  
set delivery_time=delivery_time+7
```

5.18.2 Cancellazione di righe da una tabella

Anche in questo caso, per prevenire possibili cause di errore dare il comando

set autocommit off

Vogliamo adesso cancellare le righe della tabella Inventory in cui il campo Description contiene il valore 'BOLT'. Il comando e':

```
delete from inventory -  
where description='bolt'
```

Esso si compone di due parti:

1. la clausola DELETE FROM che indica la tabella dalla quale devono essere cancellate le righe;

Interrogazione dei dati di una tabella

32

2. la clausola WHERE che specifica le righe da cancellare.

Verificato che le modifiche sono state eseguite correttamente dare il comando:

commit work

altrimenti

rollback work

Dobbiamo notare che se nel comando viene omessa la clausola Where vengono cancellate tutte le righe della tabella la quale continua ad esistere ma e' completamente vuota.

5.18.3 Aggiunta di una singola riga ad una tabella

Per aggiungere una singola riga alla tabella Inventory il comando e':

```
insert into inventory -  
values (229, 'bushing', null)
```

In questo modo alla prima colonna della tabella verra' aggiunto il valore 229 mentre nella seconda colonna il valore 'bushing'. La logica del comando e' percio' di aggiungere i dati a tutte le colonne nell'ordine in cui sono state definite nella tabella.

5.18.4 Aggiunta di righe ad una tabella

Esistono vari modi per aggiungere righe ad una tabella. Nel caso in cui il numero di aggiornamenti sia limitato, il lavoro puo' essere fatto direttamente in modo conversazionale. Facendo riferimento alle tabelle esemplificative supponiamo di dover memorizzare nuovi pezzi che non sono mai stati immagazzinati. Viene inviato il comando per andare in input alla tabella:

```
input inventory
```

Interrogazione dei dati di una tabella

33

il sistema rispondera' con le caratteristiche di definizione della tabella riportate nella figura seguente:

ARI7307A ENTER DATA VALUES, SEPARATED BY COMMAS, FOR THE FOLLOWING COLUMNS. CHARACTER DATA MUST BE ENCLOSED IN SINGLE QUOTES.

| COLUMN NAME | DATA TYPE | NULL VALUE ALLOWED |
|-------------|-----------------|--------------------|
| PARTNO | WHOLE NUMBER | NO |
| DESCRIPTION | CHARACTER (24) | YES |
| QONHAND | WHOLE NUMBER | YES |

Dalla figura si puo' vedere quanto segue:

1. Nella prima colonna compare il nome dell'attributo della tabella;
2. Nella seconda colonna compare il tipo di dato relativo all'attributo;
3. Nella terza colonna compare la specifica YES o NO a seconda che sia permesso o no l'omissione del dato per l'attributo. Nel caso in cui compare YES l'omissione del dato viene eseguita battendo NULL.

Supponiamo che siano arrivati 300 esemplari del pezzo 296 la cui descrizione e' 'belt' e 75 esemplari del pezzo 333 senza descrizione. I comandi da inviare sono:

```
296,'belt',300
333,null,75
```

Con il comando

viene terminata la sessione di input e i dati sono convalidati nella tabella (non sono convalidati se AUTOCOMMIT e' OFF).

Interrogazione dei dati di una tabella

34

Nel caso in cui e' necessario aggiungere una grande quantita' di righe viene fornito il comando

save

per convalidare nella tabella i dati che sono stati battuti fino a quel momento. Mentre il comando che consente di cancellare i dati introdotti e' il seguente:

backout

Infine se vogliamo inserire i dati in ordine di colonne diverso da quello definito nella tabella bastera' specificare nel comando input le varie colonne secondo la seguente sintassi:

```
input inventory {colonna1,colonna2,...}
```

Interrogazione dei dati di una tabella

35

6.0 ALTRI COMANDI ISQL

Vediamo in questo capitolo alcuni comandi che possono essere utilizzati per facilitare l'uso del sistema SQL.

6.1 RIEMMISSIONE DEL COMANDO CORRENTE SQL

Accade spesso dover eseguire un comando SQL precedentemente eseguito. Il sistema inserisce il comando SQL che costituisce perciò il comando 'corrente' in un'area di memoria chiamata 'buffer' il cui contenuto rimane invariato fino a quando un nuovo comando SQL viene inviato. E' così possibile eseguire un comando senza bisogno di ribattere il comando stesso ma semplicemente iniziando da terminale la parola

```
start
```

o premere il tasto PF2.

Essendo questo un comando ISQL viene memorizzato nel buffer dei comandi ISQL senza andare a sporcare il contenuto del buffer SQL.

6.2 CAMBIO DEL COMANDO CORRENTE SQL

Puo' capitare di dover correggere un errore di battitura in un comando SQL. Per evitare di ripetere l'intero comando viene fornita la facility CHANGE nell'insieme dei comandi ISQL. Supponiamo di voler eseguire la seguente query:

```
select supno,partno from quotaions
```

Nella battitura del comando e' stato fatto un errore (nome della tabella) che provoca un messaggio indicante che la tabella Quotaions non e' disponibile. Il comando Change nella specifica indicata provvede a cambiare il contenuto del buffer con il valore corretto e presentare il nuovo contenuto del buffer.

Altri comandi ISQL

```
change /quotations/quotations/
```

Il comando SQL così modificato viene mandato in esecuzione battendo a terminale il comando Start.

6.3 USO DI VARIABILI POSIZIONALI NEI COMANDI SQL

Supponiamo di voler aggiornare la tabella Inventory relativamente a nuove quantità di pezzi disponibili. Il comando:

```
hold update inventory -  
set gonhand=&1 -  
where partno=&2
```

tramite l'uso delle variabili posizionali &1 e &2 consente di eseguire l'aggiornamento della tabella fornendo dei valori (parametri) che andranno a sostituire le variabili nell'ordine in cui sono specificati.

Per eseguire il comando Update fornendo i valori da aggiornare il comando e':

```
start (gonhand+20 207)
```

Volendo aggiornare un'altra riga basterà dare un successivo comando di start:

```
start (gonhand+10 209)
```

il quale, come si ricorderà, manda in esecuzione il comando corrente che si trova nel buffer SQL. Infine la clausola Hold fa sì che il comando stesso venga memorizzato nel buffer dei comandi SQL senza venir elaborato.

6.4 COME MEMORIZZARE ED ESEGUIRE UN COMANDO SQL

Abbiamo visto come attraverso il comando Start e' possibile eseguire quello che e' stato chiamato il 'comando corrente'.

Il sistema fornisce la possibilità di memorizzare i comandi SQL in modo esplicito tramite il comando:

Altri comandi ISQL

37

store nnnnnnnn

dove nnnnnnnn e' un nome max di otto caratteri con il quale vogliamo che sia memorizzato un comando SQL e attraverso il quale vi sara' fatto riferimento .

Il comando cosi' memorizzato rimane anche dopo aver chiuso una sessione ISQL potendo in questo modo essere nuovamente utilizzato in una successiva sessione ISQL.

Volendo successivamente eseguire il comando memorizzato inviare

start nnnnnnnn

In questo modo, ISQL sposta nel buffer dei comandi SQL il comando memorizzato, lo esegue e viene mantenuto fino a quando non si vuole espressamente cancellarlo.

Supponiamo di voler memorizzare una particolare query che sara' richiamata successivamente. Il comando sia:

```
select * from quotations order by partno
```

Dopo aver inviato questo e' necessario dare uno o piu' comandi di Format relativi al comando stesso per formattare il video quando il comando viene elaborato:

```
format column 3 name cost  
format separator 4 blanks
```

quindi viene emesso il comando End per terminare la visualizzazione e il comando Store per memorizzare la Select ed i formati relativi; cioe':

```
store getquo
```

A questo punto e' utile andare a vedere se nella stesura del comando sono stati fatti errori. Tramite il comando:

```
recall getquo
```

Altri comandi ISQL

38

il comando memorizzato viene portato nel buffer dei comandi SQL per diventare il comando corrente, viene visualizzato per poter controllare che sia corretto e quindi per modificarlo e memorizzarlo di nuovo (non e' possibile verificare le informazioni di format).

Infine tramite il comando

```
start getquo
```

viene eseguito il comando memorizzato.

Allorche' viene memorizzato un comando e' necessario fare attenzione al nome specificato nel comando Store. Quando viene inviato un comando successivo il sistema provvede a memorizzare il comando precedente con il nome simbolico Previous che percio' non puo' essere usato per operazioni di memorizzazione. Nel caso in cui venisse fatta questa operazione il sistema provvede ad inviare un messaggio di errore.

Quindi per visualizzare il comando Previous basta eseguire:

```
recall previous
```

In questo modo il comando Previous diventa quello corrente mentre quello che era corrente un attimo prima viene memorizzato come Previous.

Vediamo infine alcune utility per la gestione dei comandi memorizzati:

- **list sql *:** fornisce la lista dei nomi simbolici e dei relativi comandi SQL memorizzati;
- **rename name1 name2:** cambia il nome name1 del comando memorizzato in name2;
- **erase name1:** serve per cancellare il nome simbolico name1

6.5 CREAZIONE DI UN PROSPETTO DEI RISULTATI DI UNA INTERROGAZIONE

Abbiamo visto come ottenere su video e su carta il risultato di una interrogazione. Tramite il comando Format e' possibile dare al risultato un formato che risponda alle proprie esigenze. In particolare e' possibile:

Altri comandi ISQL

39

- cambiare gli spazi che separano le colonne o convertire gli spazi in altri caratteri;
- eliminare la visualizzazione di una colonna (e inserirla successivamente);
- cambiare il titolo di una colonna visualizzata;
- specificare per le colonne numeriche il numero di cifre decimali da visualizzare;
- eliminare la visualizzazione degli zeri non significativi;
- cambiare la larghezza di visualizzazione di una colonna.

Vediamo dunque alcune di queste possibilità.

Prendiamo ad esempio il risultato fornito dal seguente comando:

```
select partno,qonorder,price,price*qonorder,suppno -
from quotations -
where partno between 209 and 222 -
order by partno,qonorder
```

| PARTNO | QONORDER | PRICE | EXPRESSION 1 | SUPPNO |
|--------|----------|-------|--------------|--------|
| 209 | 0 | 18.00 | 0.00 | 54 |
| 209 | 7 | 19.50 | 136.50 | 64 |
| 221 | 0 | 0.20 | 0.00 | 61 |
| 221 | 50 | 0.30 | 15.00 | 51 |
| 221 | 150 | 0.10 | 15.00 | 54 |
| 222 | 0 | 0.25 | 0.00 | 53 |
| 222 | 200 | 0.20 | 40.00 | 61 |

Figura 27. Prospetto usato per operazioni di modifica

Il risultato dell'interrogazione rappresentato dalla figura viene modificato dal comando:

```
format separator '|'
```

Altri comandi ISQL

nel seguente:

| PARTNO | QONORDER | PRICE | EXPRESSION 1 | SUPPNO |
|--------|----------|-------|--------------|--------|
| 209 | 0 | 18.00 | 0.00 | 54 |
| 209 | 7 | 19.50 | 136.50 | 64 |
| 221 | 0 | 0.20 | 0.00 | 61 |
| 221 | 50 | 0.30 | 15.00 | 51 |
| 221 | 150 | 0.10 | 15.00 | 54 |
| 222 | 0 | 0.25 | 0.00 | 53 |
| 222 | 200 | 0.20 | 40.00 | 61 |

Figura 28. Prima modifica al prospetto

Se invece si vogliono eliminare dal video e quindi dal formato di stampa le colonne qonorder e price, anzichè dare un nuovo comando di select basta inviare:

```
format exclude (qonorder price)
```

oppure:

```
format exclude (2 3)
```

dove 2 e 3 sono la posizione delle colonne da eliminare e specificate nella clausola Select che ha fornito il risultato della interrogazione. Il comando:

```
format exclude all but (3)
```

per escludere tutte le colonne tranne la numero 3.

Nel caso in cui si vuole vedere una colonna che prima era stata esclusa il comando è:

```
format include (qonorder)
```

oppure :

Altri comandi ISQL

format include only (2)

6.6 MODIFICA IL TITOLO DI UNA COLONNA VISUALIZZATA

Abbiamo visto che nel prospetto di stampa, in corrispondenza di una colonna non definita nella tabella compare il nome di tipo generico 'EXPRESSION 1'. E' possibile cambiare questo nome fornendo un valore piu' significativo tramite il comando:

```
format column 'espressione 1' name cost
```

Questo da' luogo a :

| PARTNO | QONORDER | PRICE | COST | SUPPNO |
|--------|----------|-------|--------|--------|
| 209 | 0 | 18.00 | 0.00 | 54 |
| 209 | 7 | 19.50 | 136.50 | 64 |
| 221 | 0 | 0.20 | 0.00 | 61 |
| 221 | 50 | 0.30 | 15.00 | 51 |
| 221 | 150 | 0.10 | 15.00 | 54 |
| 222 | 0 | 0.25 | 0.00 | 53 |
| 222 | 200 | 0.20 | 40.00 | 61 |

Figura 29. Seconda modifica al prospetto

Analogamente a quanto visto nelle precedenti espressioni del comando Format si puo' usare il numero per identificare il titolo della colonna da modificare.

```
format column 4 name cost
```

Nella figura precedente si vede che nella colonna Cost appena modificata compaiono 2 cifre decimali. Nel caso in cui si vogliono omettere nella stampa queste cifre il comando e':

```
format column cost dplaces 0
```

Altri comandi ISQL

Il risultato e':

| PARTNO | QONORDER | PRICE | COST | SUPPNO |
|--------|----------|-------|------|--------|
| 209 | 0 | 18.00 | 0. | 54 |
| 209 | 7 | 19.50 | 136. | 64 |
| 221 | 0 | 0.20 | 0. | 61 |
| 221 | 50 | 0.30 | 15. | 51 |
| 221 | 150 | 0.10 | 15. | 54 |
| 222 | 0 | 0.25 | 0. | 53 |
| 222 | 200 | 0.20 | 40. | 61 |

Figura 30. Terza modifica al prospetto

Se vogliamo evidenziare gli zeri non significativi in una colonna il comando e':

```
format column partno zeros on
```

che da luogo alla stampa:

| PARTNO | QONORDER | PRICE | COST | SUPPNO |
|--------|----------|-------|------|--------|
| 00209 | 0 | 18.00 | 0. | 54 |
| 00209 | 7 | 19.50 | 136. | 64 |
| 00221 | 0 | 0.20 | 0. | 61 |
| 00221 | 50 | 0.30 | 15. | 51 |
| 00221 | 150 | 0.10 | 15. | 54 |
| 00222 | 0 | 0.25 | 0. | 53 |
| 00222 | 200 | 0.20 | 40. | 61 |

Figura 31. Quarta modifica al prospetto

Invece il comando

Altri comandi ISQL

format column partno zeros off

toglie gli zeri non significativi da una colonna nel prospetto di stampa. E' possibile infine cambiare la larghezza visualizzata di una colonna. Il comando:

```
format column gonorder width 8
```

riduce la larghezza della colonna gonorder al valore 8.

6.7 ALTRE INFORMAZIONI SULLA FORMATTAZIONE DEI PROSPETTI

Per analizzare altre specifiche del comando Format al fine di ottenere un prospetto di stampa sempre migliore facciamo riferimento al display fornito dal comando:

```
select partno,gonorder,price,price*gonorder,supjno -  
from quotations -  
where partno between 209 and 222 -  
order by partno,gonorder
```

già utilizzato nei paragrafi precedenti.

Adesso eseguiamo alcuni comandi di Format che conosciamo:

```
format separator '|'  
format exclude 3  
format column 4 name cost  
format column cost dplaces 0
```

otteniamo così il seguente risultato:

| PARTNO | QONORDER | PRICE | COST | SUPPNO |
|--------|----------|--------|--------|--------|
| 209 | 0 | 136.00 | 0.00 | 54 |
| 209 | 7 | 136.00 | 136.00 | 64 |
| 221 | 0 | 15.00 | 0.00 | 61 |
| 221 | 50 | 15.00 | 15.00 | 51 |
| 221 | 150 | 15.00 | 15.00 | 54 |
| 222 | 0 | 40.00 | 0.00 | 53 |
| 222 | 200 | 40.00 | 40.00 | 61 |

Altri comandi ISQL

| PARTNO | QONORDER | COST | SUPPNO |
|--------|----------|------|--------|
| 209 | 0 | 0. | 54 |
| 209 | 7 | 136. | 64 |
| 221 | 0 | 0. | 61 |
| 221 | 50 | 15. | 51 |
| 221 | 150 | 15. | 54 |
| 222 | 0 | 0. | 53 |
| 222 | 200 | 40. | 61 |

Figura 32. Prospetto usato per altri tipi di formattazione

Da una prima analisi della figura si vede che nella colonna Partno compaiono valori duplicati che desideriamo cancellare e una compattazione della tabella che rende la stessa poco chiara.

Tramite il comando:

```
format group (partno)
```

viene identificata la colonna che contiene i gruppi di valore che si vogliono evidenziare. Il risultato sarà:

| PARTNO | QONORDER | COST | SUPPNO |
|--------|----------|------|--------|
| 209 | 0 | 0. | 54 |
| | 7 | 136. | 64 |
| 221 | 0 | 0. | 61 |
| | 50 | 15. | 51 |
| | 150 | 15. | 54 |
| 222 | 0 | 0. | 53 |
| | 200 | 40. | 61 |

Figura 33. Utilizzo della funzione GROUP

Altri comandi ISQL

E' naturale che l'evidenziazione ha senso solo su una colonna che sia stata precedentemente ordinata in gruppi di valori simili tramite la clausola Order By nel comando Select. E' possibile ottenere anche un prospetto che contiene i totali e i totali parziali. Il comando:

`format total (qonorder cost)`

fornisce per le colonne Qonorder e Cost il totale cioe':

| PARTNO | QONORDER | COST | SUPPNO |
|--------|----------|-------|--------|
| 209 | 0 | 0. | 54 |
| | 7 | 136. | 64 |
| 221 | 0 | 0. | 61 |
| | 50 | 15. | 51 |
| | 150 | 15. | 54 |
| 222 | 0 | 0. | 53 |
| | 200 | 40. | 61 |
| | ===== | ===== | |
| | 407 | 206. | |

Figura 34. Utilizzo della funzione TOTAL

Mentre il comando:

`format subtotal (qonorder cost)`

fornisce i totali parziali cioe' per gruppi sempre delle colonne Qonorder e Cost.

E' ovvio che nel caso in cui sono richiesti i totali parziali viene fornito anche il totale generale di ciascuna colonna specificata.

La stampa sara':

| PARTNO | QONORDER | COST | SUPPNO |
|--------|----------|-------|--------|
| 209 | 0 | 0. | 54 |
| | 7 | 136. | 64 |
| ***** | 7 | 136. | |
| 221 | 0 | 0. | 61 |
| | 50 | 15. | 51 |
| | 150 | 15. | 54 |
| ***** | 200 | 30. | |
| 222 | 0 | 0. | 53 |
| | 200 | 40. | 61 |
| ***** | 200 | 40. | |
| | ===== | ===== | |
| | 407 | 206. | |

Figura 35. Utilizzo della funzione SUBTOTAL

Infine i totali generali e parziali possono essere cancellati rispettivamente con i comandi:

`format total erase`

`format subtotal erase`

Se vengono cancellati i totali parziali automaticamente saranno cancellati anche i totali generali.

A questo punto con il comando Store possiamo memorizzare tutte le operazioni che hanno portato alla realizzazione del prospetto desiderato e successivamente tramite il semplice invio del comando Start ottenere di nuovo il prospetto.

6.8 CREAZIONE DI TITOLI PER I PROSPETTI STAMPATI

In ognuno dei prospetti di stampa ottenuti viene riportato, in modo automatico dal sistema, un titolo di testa costituito dai primi 50 caratteri del comando Select che ha dato origine al risultato. Per cambiare il titolo al prospetto dare il comando:

`format ttitle ' nuovo titolo '`

Altri comandi ISQL

mentre

```
format btitle ' titolo di fondo '
```

fa comparire nel prospetto di stampa il titolo di fondo specificato.

6.9 CONTROLLO DELLE CARATTERISTICHE OPERATIVE

E' possibile eseguire operazioni di formattazione che se specificate all'inizio di una sessione a terminale, valgono per tutta la sua durata se naturalmente non vengono modificate o escluse.

Il tipo di formattazione che adesso andiamo a descrivere si riferisce ai prospetti visualizzati e percio' non deve essere confuso con le funzioni del comando Format.

Si possono eseguire i seguenti tipi di formattazione:

- Punteggiatura visualizzata per colonne numeriche;
- Caratteri di separazione visualizzati tra le colonne;
- Caratteri visualizzati per i campi nulli;
- Larghezza visualizzata nei campi

Tramite il comando :

```
set decimal
```

vengono definiti i separatori dei decimali e delle migliaia. I separatori sono:

punto, virgola e spazio

e possono essere usati nelle seguenti combinazioni:

```
set decimal /,./ /
set decimal /./ /
set decimal / / /
set decimal //./ /
```

dove il carattere tra le prime due barre rappresenta il separatore delle migliaia e tra le ultime due il separatore dei decimali.

La punteggiatura cosi' definita sara' valida per tutte le colonne numeriche e rimarra' in vigore per tutta la durata della sessione. Supponiamo di aver dato all'inizio della sessione il comando:

Altri comandi ISQL

48

```
set decimal /,./ /
```

Il risultato dei separatori in relazione al comando:

```
select 60*price -
from quotations -
where supno=64 and partno=209
```

e' mostrato nella figura seguente:

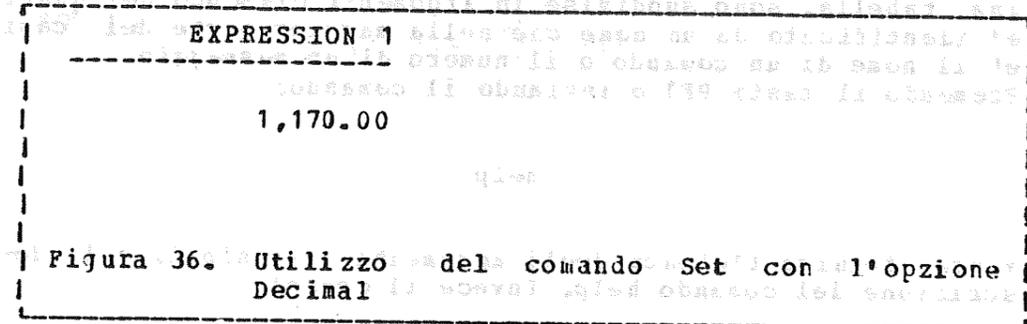


Figura 36. Utilizzo del comando Set con l'opzione Decimal

Il comando Set consente di controllare le caratteristiche di lavoro in una sessione ISQL. Riportiamo alcuni esempi:

set separator n blanks definisce il numero 'n' di blanks che debbono essere lasciati tra le colonne (default 2);

set separator string definisce una stringa di caratteri, da sostituire alla parola 'string', che dovra' essere visualizzata tra le colonne.

set null string ricopre con una stringa di caratteri eventuali campi nulli visualizzati. String indica la stringa di caratteri desiderati. La sua lunghezza massima e' 20.

set copies n consente di definire il numero di copie (n) di ogni operazione di stampa. Questa operazione puo' esser fatta anche con il comando Print il quale usera' il valore specificato nel comando Set a meno che non sia stata specificata la parola chiave Copies.

set pagesize width w lenght l il comando serve a definire i formati di pagina

- w indica la larghezza della pagine di stampa (default 132);

Altri comandi ISQL

49

- l indica la lunghezza della pagina di stampa (default 66);

list set * consente di visualizzare i valori standard delle caratteristiche operative nella sessione di lavoro.

6.10 COMANDI PER LA RICERCA DI INFORMAZIONI ON-LINE

E' disponibile il comando Help per ottenere informazioni on line sui comandi SQL e ISQL. Le informazioni, memorizzate in una tabella, sono suddivise in argomenti ciascuno dei quali e' identificato da un nome che nella maggior parte dei casi e' il nome di un comando o il numero di un messaggio. Premendo il tasto PF1 o inviando il comando:

help

viene fornito l'elenco degli argomenti disponibili e la descrizione del comando help. Invece il comando:

help contents

fornisce solamente l'elenco degli argomenti disponibili.

Altri comandi ISQL

50

7.0 BIBLIOGRAFIA

[SQLgi] SQL/DS: General Information for VM/System Product IBM GH24-5064.

[SQLtur] SQL/DS: Terminal User's Reference for VM/System Product IBM SH24-5067.

[SQLtag] SQL/DS: SQL/DS Terminal User's Guide for VM/System Product IBM SH24-5045

[SQLcf] SQL/DS: SQL/DS Concepts and Facilities for VM/System Product IBM GH24-5065.

Bibliografia

51

APPENDICE

SECRET

TABELLA INVENTORY

| PARTNO | DESCRIPTION | QONHAND |
|--------|-------------|---------|
| 207 | GEAR | 75 |
| 209 | CAM | 50 |
| 224 | BOLT | 650 |
| 222 | BOLT | 1250 |
| 231 | NUT | 700 |
| 232 | NUT | 1100 |
| 241 | WASHER | 6000 |
| 285 | WHEEL | 350 |
| 295 | BELT | 85 |
| 301 | FUSE | 300 |
| 302 | FUSE | 367 |
| 310 | PIN | 150 |
| 311 | PIN | 292 |
| 315 | SWITCH | 421 |
| 316 | SWITCH | 342 |
| 320 | RELAY | 0 |
| 321 | RELAY | 0 |
| 322 | RELAY | 0 |
| 323 | RELAY | 0 |

TABELLA SUPPLIERS

| SUPPNO | NAME | ADDRESS |
|--------|----------------|-------------------------------------|
| 51 | DEPECTO PARTS | 116 8TH ST., BROKEN HAND, WY |
| 52 | VESEVIUS, INC. | 1512 ANCIENT BLVD., POMPEII NY |
| 53 | ATLANTIS CO. | 18 OCEAN AVE., WASHINGTON DC |
| 54 | TITANIC PARTS | 132 LARGE ST., BIGTOWN, TX |
| 57 | EAGLE HARDWARE | 164 TRANQUILITY PLACE, CAPOLEON MN |
| 61 | SKY PARTS | 1128 ORBIT BLVD., SIDNEY AUSTRALIA |
| 64 | KNIGHT LTD. | 1256 ARTHUR COURT, CAMELOT ENGLAND |
| 70 | COMPANY 1 | 150 GRAND AVE, SPRINGFIELD ILL |
| 71 | COMPANY 2 | 1130 OAK HILL AVE, DETROIT MI |
| 72 | COMPANY 3 | 1241 OLD CASTLE RD, PHILADELPHIA PA |