Corso Database ed SQL

prof *Carmine Napoli*

Introduzione

Database:
Si definisce Database un insieme di dati, di solito di notevoli dimensioni, raccolti, memorizzati ed organizzai in modo opportuno tale da rendere facile una loro consultazione.

I dati possono essere del tipo **strutturati** o **non strutturati**

I **Dati Strutturati** sono quegli archivi organizzati in record, campi, sottocampi, individuabili facilmente medianti indici e chiavi, essi si indicano come **Sistema Informativo**.

I **Dati Non Strutturati** sono quei dati che non è possibile strutturare in record e campi come ad esempio possono essere una poesia e/o una sentenza, un racconto ecc, in questo caso si dice che si tratta di **Documentazione Automatica**.

Si possono avere database che riguardano sia l'uno che l'altro tipo di dati.

Ognuno dei due tipi di database ha un diversa problematica,

i **database strutturati** in genere riguardano dati dinamici, che cambiano con frequenza essendo aggiornati con continuità, un esempio l'elenco dei voti degli allievi di una scuola, che deve essere aggiornato ogni giorno;

in una qualsiasi organizzazione (azienda, comune, scuola) questo tipo di database è quello che raccoglie tutti i dati utili alla vita della organizzazione, ed è dalla sua consultazione che scaturiscono tutte le informazioni che saranno utilizzate per prendere decisioni ai vari livelli dell'organizzazione.

Ad esempio in una azienda manufatturiera si potranno gestire magazzini, ordini, vendite, paghe personale, ecc.

Nella organizzazione del l'archivio dev essere prevista la possibilità di memorizzare e di estrarre dei dati inizialmente non richiesti

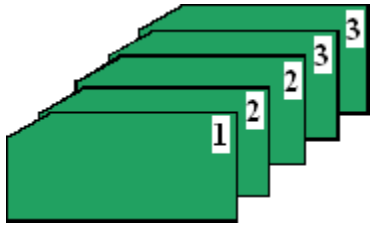
i **database informativi** trattano invece dei dati con una dinamica dei dati molto più lenta

per questo tipo di database risulta difficile determinare record, campi, sottocampi ben definiti, per cui la loro consultazione prevede che i documenti da memorizzare siano classificati in base al loro contenuto, e che da questo siano estratti tratti alcuni termini che saranno memorizzati in un vocabolario e che serviranno per la consultazione.
Il vocabolario viene chiamato THESAURUS.
Per la consultazione di questo tipo di database l'utente deve scegliere un certo numero di termini che lui ritiene più significativi

Tra i problemi legati a questo tipo di database ci sono le dimensioni del vocabolario e la presenza in esso di sinonimi che possono rendere lenta e difficile la consultazione.

Durante il corso tratteremo i Sistemi di Base Strutturati.

Analisi di un database su schede, riguardante una biblioteca.



Si dovranno memorizzare vari tipologie di dati: Libri, Lettori, Prestiti, per ognuno di essi si utilizzeranno schede diverse.

Nella prima sono memorizzati i dati relativi ai libri (Titolo, Autore, Editore, Anno Edizione ecc.)

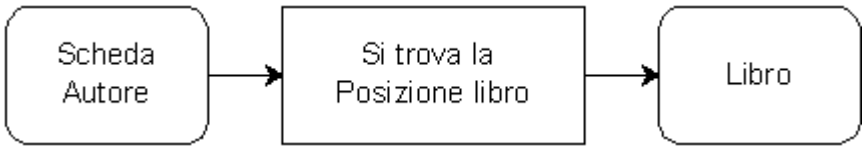
Nella seconda i dati relativi ai lettori (oltre alle generalità anche domicilio, telefono ecc.)

Infine una scheda che tiene conto dei prestiti, in essa devono comparire sia i riferimenti ai libri che ai lettori.

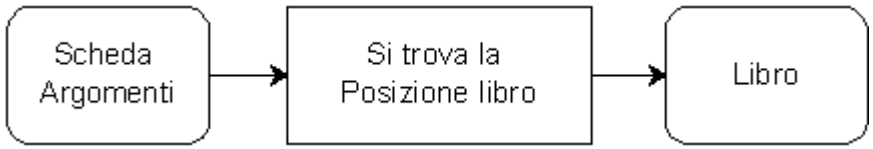
Mentre le schede dei libri e dei lettori Cambiano ma con una certa gradualità,
Le schede relative ai prestiti hanno una grande variabilità,

Per quanto riguarda quest'ultime in genere si utilizzano schede che memorizzano i prestiti di un libro, è però anche necessario sapere i vari prestiti relativi ad un lettore.

La ricerca di un libro può essere fatta partendo dalle schede degli autori



Si deve ancora far notare come per la ricerca dei libri può essere necessario un secondo schedario che tratta gli argomenti e che devono far riferimento allo schedario dei libri.



Si deve far notare come questo tipo di schedario potrebbe, analizzando i libri maggiormente letti, permettere di conoscere gli argomenti preferiti del pubblico e quindi aiutare nella scelta dei libri da comprare.

Un Database organizzato in schede risulta abbastanza rigido, tutto quello da memorizzare deve essere deciso fin dall'inizio.

Si noti la difficoltà di aggiungere nelle schede altri campi che non siano stati previsti dal principio.

Ad esempio su una scheda, impostata dieci anni fa, e relativa ai lettori sarebbe difficile memorizzare oltre all'indirizzo fisico anche la email o il numero dl telefonino, volendo aggiungere questo dato è necessario riscrivere tutti i dati delle varie schede.

Ancora tra le caratteristiche negative di questo tipo di database notiamo come alcuni dati (ad esempio il titolo del libro) sono memorizzate più volte su schede diverse (su quella del libro e su quella dei prestiti e su quella degli argomenti), questo può portare a facili errori (ad esempio lo stesso libro potrà essere scritto con nomi diversi e quindi la consultazione è difficile.)

Gli Archivi sul PC

I primi archivi scritti su Computer erano del tipo **file system**

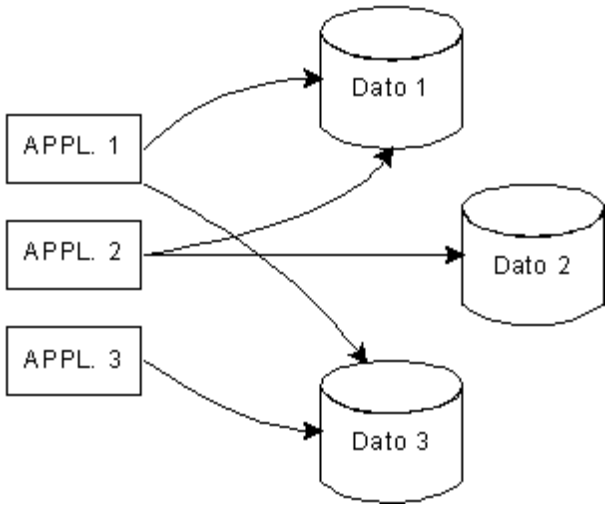


Figura 4 Approccio File System

In questo tipo di gestione degli archivi ogni programma agisce direttamente sugli archi per ne deve conoscere struttura e posizione, risulta problematico l'utilizzo contemporaneo degli archivi da parte di due utenti.

Approccio con il metodo **Database Managment System** (DBMS) - Sistemi per la gestione di base di Dati

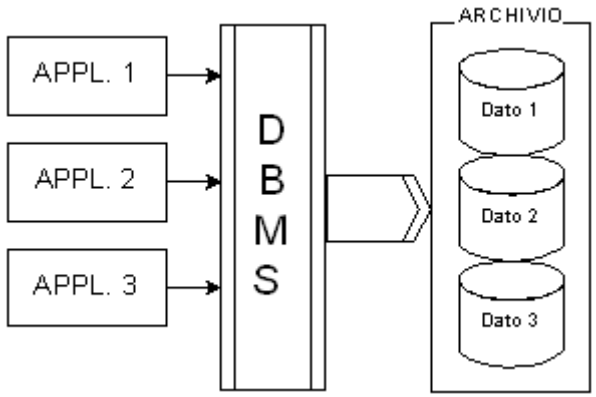


Figura 5 Approccio DBMS

In questo tipo di gestione tra gli archivi e le applicazione si interpone un componente che gestisce direttamente il database.
In questo caso l'applicazione deve solo fare la domanda al DBMS in modo corretto, sarà questo che opererà sugli archivi per dare la risposta cercata.
Si noti come in questo caso le applicazioni non devono conoscere il modo con cui le varie informazioni sono state archiviate, questo è compito del DBMS

L’uso di DBMS è possibile grazie al fatto che i programmi oggi sono progettati in modo modulare

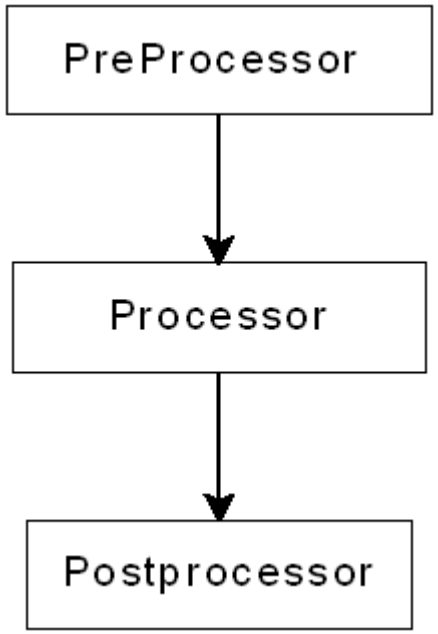


Figura 6

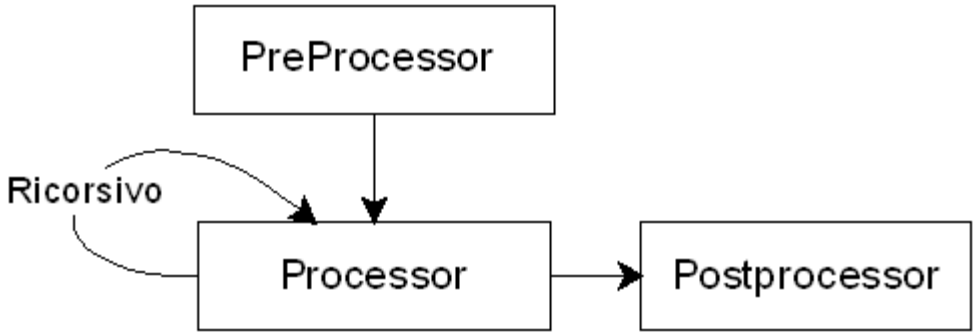


Figura 7

Ogni modulo si interessa da fare una certa operazione ottenendo dei risultati serviranno come punto di partenza per il modulo successivo, si noti come nella figura n° 7 i risultati del processor possono servire come immissione dati ancora al processor (ricorsività)

L’uso del DBMS comporta una serie di vantaggi che permettono una gestione dei dati più affidabile; in particolare si ha:

- Indipendenza dei programmi dalla organizzazione dei dati
- Non ridondanza dei dati
- Numero di Campi variabili
- Unicità degli aggiornamenti
- Possibilità di rappresentare relazioni comunque complesse tra i dati
- Protezione dei dati
- Facilità di accesso da parte di utenti diversi
- Maggior semplicità dell stesura dei programmi
- Controllo centralizzato
- Sicurezza dei dati (recupero dopo un crash)

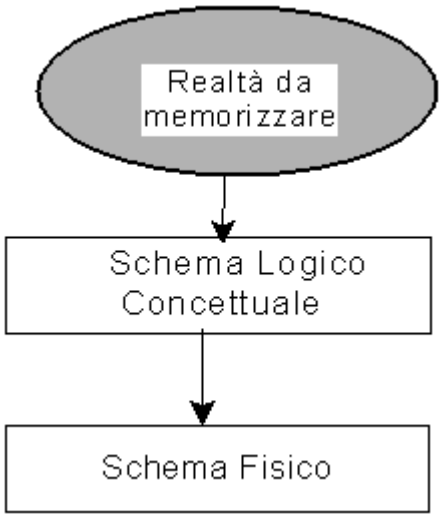
I vari termini sono di ovvia comprensione

Passi per lo sviluppo di un database

La fase iniziale consiste nell'analisi della realtà quotidiana.

Si passa poi allo schema concettuale con il quale si decide il tipo di database da utilizzare e si dividono i dati complessi in in parti semplici definendo le relazioni che le uniscono.

Dopo avere deciso il tipo di DBMS da Utilizzare si passa allo schema fisico con il quale si decide il modo con cui devono essere organizzati gli archi, che tipo di linguaggio utilizzare ecc..



Utenti di un Database

Si possono individuare tre tipologie di utenti

- **Utenti Finali**

  Sono quelli che utilizzano il database dopo aver ricevuto, per database di notevoli dimensioni, l'autorizzazione all'accesso ai dati dall'amministratore del database
- **Programmatori**

  Sono quelli che producono il database
- **Amministratori del Database**

  Sono le persone che possiedono una buona conoscenza dell'archivio, lo mantiene in esercizio provvedendo alla sua manutenzione e ad assegnare ai vari utenti le autorizzazioni all'utilizzo dei dati

Tipologia dei database

- **modello gerarchico (ad albero)** i dati sono distribuiti in modo da formare un albero, per individuare un dato si deve percorrere un tratto prestabilito
- **modello reticolare** i dati sono disposti a reticolo, l'individuazione di un dato l'individuazione di un dato può avvenire da un qualsiasi parte del reticolare il cammino però deve essere deciso al momento della progettazione
- **modello relazionale** si basano su tabelle composta da righe e colonne correlate tra di loro mediante delle associazioni (relazioni)
- **ad oggetti** sono l'ultima frontiera dei database, definiscono delle classi che definiscono oltre alle caratteristiche dei dati anche il loro comportamento. In questo tipo di database i dati non si comportano in modo passivo, ma essi contengono anche le modalità con cui essi operano.

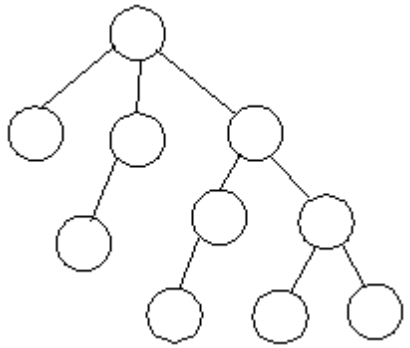


Figura 9 Database ad albero

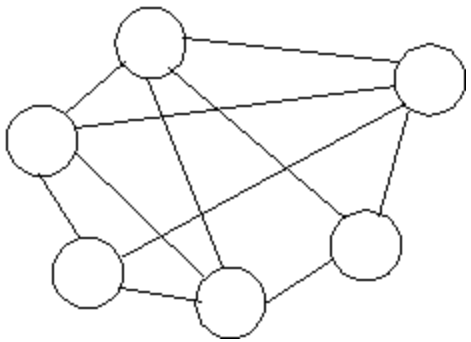


Figura 10 Database reticolare

DATABASE RELAZIONALE

Il database relazionale fu concepito nel 1969 da E. F. Codd un ricercatore dell' IBM. che utilizzò la teoria degli insiemi e le basi della logica.

Secondo il modello relazionale i dati sono immagazzinati in **relazioni** che appaiono come **tabelle**. In ogni tabella si individuano delle colonne che sono dette **campi** (Fields) e righe detti record (attributi)

TABELLA ANAGRAFE

CodicePersona	Cognome	Nome	Città	Stato	Telefono
001	Rossi	Antonio	Salerno	I	12345567
002	Verde	Francesco	Napoli	I	23455678
003	Giallo	Angelo	Avellino	I	23533535
004	Bianchi	Nicola	Salerno	I	35354353

} Record

Campi

Tabelle

Le tabelle (relazioni) sono le strutture principali di un database relazionale.
I dati che in essa sono memorizzati devono riguardare un unico, specifico soggetto.
Essi possono definire una o più **caratteristica** di qualcosa di **fisico**, ad esempio il Nome, il Cognome ecc. di una persona, oppure possono riguardare le **caratteristiche** di un **evento** che capita in un preciso momento come ad esempio il prestito di un libro.

TABELLA PRESTITI

CodicePrestito	Data Prestito	Libro	Nome	Data Restituzione
001	01/01/02	Libro di Storia	Bianchi	
002	01/21/02	Libro geografia	Verde	

In ogni tabella sarà sempre scelto un **campo** (colonna) che individua in modo univoco il **record** (la riga), esso viene detto **CHIAVE PRIMARIA**, nelle due tabelle precedenti la chiave primaria è il codice

ELEMENTI DATABASE RELAZIONALE

Campo

Il campo è la struttura più piccola di un database e rappresenta una delle caratteristiche del soggetto della tabella, in esso sono effettivamente memorizzati i dati, dopodiché essi possono essere richiamati, accorpati nei modi che si ritiene utile per ottenere le informazioni richieste.

Ai vari campi viene assegnato un nome, che deve essere scelto in modo da poter individuare immediatamente il tipo di dato memorizzato. Ad esempio un campo cognome o un campo nome immediatamente rendono conto del tipo di dati memorizzati

Record

Il Record è un insieme di campi che individuano una caratteristica particolare del soggetto.
La tabella risulta formata da un insieme di record.
Si noti come i campi che formano un record possono essere anche vuoti (in esso non è memorizzato alcun valore)
Ogni record nella tabella è indicato dal valore di un particole campo che è detto CHIAVE PRIMARIA

Chiavi
Le chiavi sono dei campi speciali che servono al collegamento tra le tabelle che compongono il database, vi sono vari tipi di chiavi, ne considereremo solo due:

- CHIAVE PRIMARIA
- CHIAVE ESTERNA

La chiave primaria è un campo che serve ad individuare in modo univoco un record.
Per questo motivo il campo non potrà mai essere vuoto e conterrà dati diversi per ogni record.
Essa potrà essere formata anche da due o più campi, in questo caso è detta CHIAVE PRIMARIA COMPOSITA. Nelle tabelle precedenti le chiavi primarie contengono la parola codice.

Se consideriamo le tabella precedenti notiamo come nella tabella prestiti compare anche un campo **Nome** che deve essere uguale al **Nome** contenuto nella tabella **anagrafe**. Tutto ciò comporta una duplicazione dei dati presenti negli archivi, con buone possibilità di fare errori.
È quindi conveniente cambiare latabella prestiti eliminando il campo Nome ed inserendo al suo posto un campo CodicePersona nel quale andremo a memorizzare il CodicePersona che troveremo nella tabella anagrafe, che quindi potrà contenere solo un tipo di valori.

La **tabella prestiti** diventa

CodicePrestito	Data Prestito	Libro	CodicePersona	Data Restituzione
001	01/01/02	Libro di Storia	004	09/09/02
002	01/21/02	Libro geografia	003	07/09/02

CodicePersona	Cognome	Nome	Città	Stato
001	Rossi	Antonio	Salerno	I
002	Verde	Francesco	Napoli	I
003	Giallo	Angelo	Avellino	I
004	Bianchi	Nicola	Salerno	I

Il campo CodicePersona della tabella prestiti viene definito chiave esterna, esso risulta strettamente collegato alla chiave primaria della tabella anagrafe, infatti in esso potranno essere memorizzati solo dati che trovano una corrispondenza nella chiave primaria della tabella correlata, in questo modo si elimina la possibilità di errori di memorizzazione e di non avere corrispondenza tra le persone che hanno preso un prestito e le persone memorizzate in anagrafe (RECORD ORFANI)

View (Query)

La tabella prestiti cambiata nel modo che si è detto risulta di difficle lettura per cui sono state definite delle particolare *Tabelle Virtuali* dette appunto View (anche se attualmente son in genere definite QUERY) che derivano dalla unione opportuna di due tabelle.

Nel nostro caso è possibile ricavare la Query

CodicePrestito	DataPrestito	Libro	CodicePersona	DataRestituzione	Cognome	Nome	Telefono
001	01/01/02	Libro di Storia	004	09/09/02	Bianchi	Nicola	35354353
002	01/21/02	Libro geografia	003	07/09/02	Giallo	Angelo	23533535

Si come le prime 5 colonne si ricavano dalla tabella prestiti, mentre le ultime 3 si ricavano dalla tabella prestiti.

È facile capire che la tabella virtuale risulta notevolmente più facile da consultare rispetto alle tabella di partenza

Nella costruzione di una Query è possibile fare anche dei calcoli numerici, come fare una somma o ricavare un totale ecc.

Una caratteristica molto importante delle Query è la possibilità di memorizzarle e richiamarle nel momento desiderato diminuendo anche notevolmente il tempo necessario.

Relationship

Da quanto scritto in precedenza si ricava che tra due tabelle è possibile definire dei collegamenti che in seguito chiameremo Relazioni.
Esistono tre tipologie di relazioni:

- **Relazione UNO a UNO**
- **Relazione UNO a MOLTI**
- **Relazione MOLTI a MOLTI**

La definizione delle relazioni tra due tabelle risulta fondamentale, ad essa risulta collegata un'altra caratteristica che deve avere quella di integrità referenziale.

Relazione UNO a UNO

Tra due tabelle esiste una RELAZIONE UNO A UNO se ad un record della prima tabella è possibile associare un solo record della seconda tabella.

Come esempio del nostro database si può inserire una tabella, collegata alla tabella anagrafe che gestisca le autorizzazioni all'accesso ad un parcheggio per dipendenti.
Se ad ogni Persona sarà consentito l'ingrasso con una sola automobile, non tutti però ne usufruiranno, per cui se fosse inserito un campo "Tipo Auto" ed un campo "Targa" si potrebbero avere dei record non pieni con spreco di spazio per cui si preferisce inserire una nuova tabella nella quale inserire tutti i dati relativi all'auto.

Si noti come per ogni Persona si potrà inserire un solo Record

CodicePersona	Auto	Targa
001	Punto	SA00000
003	Uno	SA00001
004	Panda	SA00001

CodicePersona	Cognome	Nome	Città	Stato
001	Rossi	Antonio	Salerno	I
002	Verde	Francesco	Napoli	I
003	Giallo	Angelo	Avellino	I
004	Bianchi	Nicola	Salerno	I

Sarà comunque possibile definire una Query dove visualizzare i dati delle due tabelle

Relazione UNO a MOLTI

Si ha una relazione uno a molti quando ad un record di una tabella si possono associare uno o più record di un'altra tabella.

In questo caso in ogni tabella si dovrà definire una chiave primaria che servirà ad individuare il record. Nella tabella collegata dovrà essere definito anche una ulteriore chiave detta chiave esterna.

Questo tipo di relazione è quella che esiste tra la tabella anagrafica e la tabella prestiti, di cui si è detto in precedenza.

Relazione MOLTI a MOLTI

Si ha una relazione molti a molti quando ad un record della prima tabella sono collegati molti record della seconda tabella e ad un record della prima tabella più record di una prima tabella sono collegati più record di una seconda.
In questo caso il campo che definisce la chiave primaria della tabella definisce anche la chiave esterna.

È utilizzata per mantenere operazioni storiche su più tabelle, ad esempio le operazioni di vendita dei prodotti di un'azienda hanno molteplici corrispondenze nelle operazioni di acquisto degli stessi prodotti.

Integrità refenziale

Nelle relazioni UNO a UNO e UNO a MOLTI è possibile definire un vincolo detto **Integrità Referenziale** che assicura la correttezza dei dati inseriti nelle tabelle.

Con la integrità referenziale si fa in modo che nel campo definito come chiave esterna nella tabella correlata, potranno essere inseriti solo valori che sono presenti nella chiave primaria della tabella primaria correlata, inoltre non sarà possibile eliminare dalla chiave primaria della tabella principale un valore presente nella chiave esterna della tabella secondaria.

Tipologia di Campo

Esistono varie tipologie di campo

1. **Numerici**: contengono solo numeri interi o numeri decimali
2. **Carattere**: Contengono testo a larghezza fissa o variabile
3. **Data/Ora**: contengono date , ore od entrambe le cose
4. **Binari**: contengono altri dati quali immagini, oggetti o testi di grandi dimensioni
5. **Particolari**: contengono dati di altro tipo, ad esempio numeri complessi, o altro
6.

Query di selezione

SELECT elenco dei campi
FROM nomi delle tabelle IN nome del database
WHERE condizione di ricerca
GROUP BY elenco dei campi
HAVING criteri di ricerca
ORDER BY elenco dei campi

Query di eliminazione

DELETE (tabella.*)
FROM espressione tabella
WHERE criteri

Query di accodamento
INSERT INTO destinazione [IN database esterno]
SELECT [origine.]campo1 [,campo2]
FROM espresione tabella

Query di aggiornamento
UPDATE tabella
SET nuovo valore
WHERE criteri

Join

INNER JOIN

LEFT OUTER JOIN

RIGHT OUTER JOIN

Select .....
FROM tabella1 INNER JOIN tabella2 ON tabella1.campo1=tabella.campo2

```
SELECT DISTINCTROW  nome dei campi
FROM tabella1 [LEFT |RIGHT| JOIN tabella2
ON tabella1.campo1=tabella.campo2
```

Query di unione

query1 UNION quer2