



Università degli Studi di Cagliari
Corso di Laurea in Ingegneria Elettronica



Diee

ELEMENTI DI INFORMATICA

<http://agilegroup.eu>

A.A. 2015/2016

Docente: **Michele Marchesi**

BASI DI DATI

Sommario

- **Basi di Dati, file, DBMS**
- **Il modello relazionale**
- **Il linguaggio SQL**



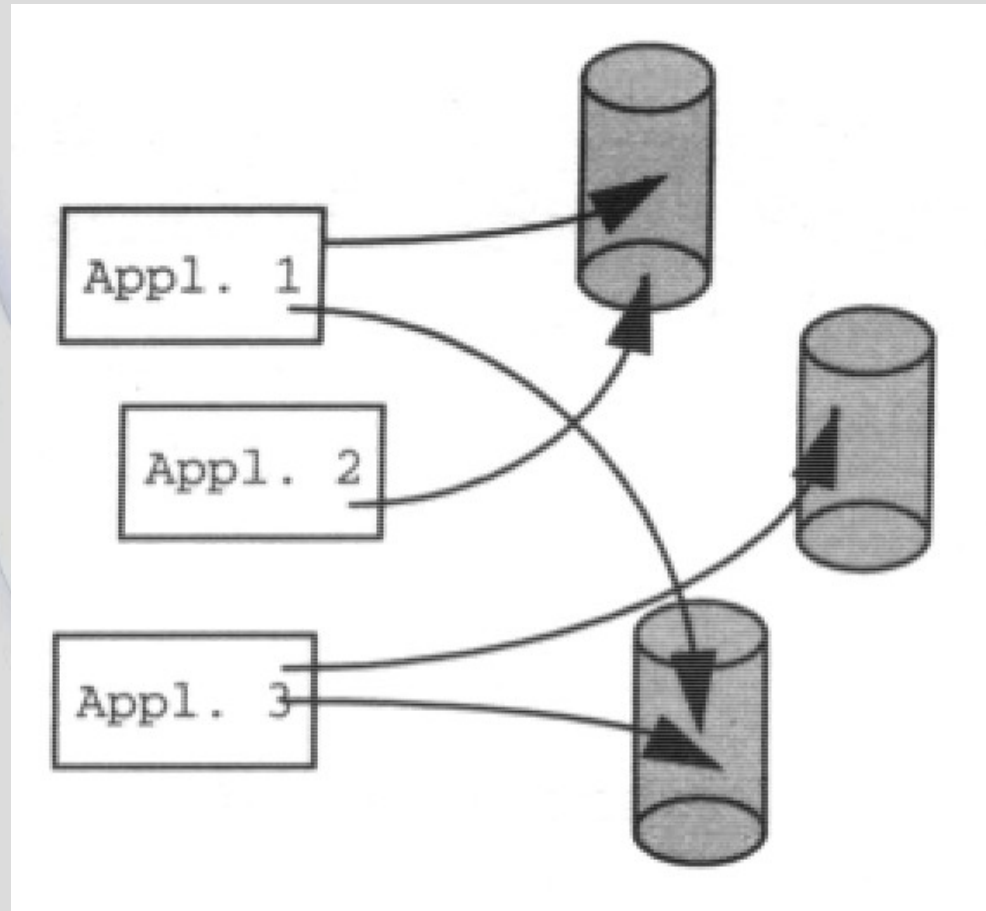
Basi di Dati

Basi di Dati

- **Base di dati (o database): insieme di dati permanenti, raccolti in base a precise regole, e organizzati in strutture ben definite**
- **DBMS (Data Base Management System): sistema software che permette di creare, organizzare e gestire basi di dati**
 - **Opera al di sopra del sistema operativo**
 - **Offre linguaggi specifici per l'organizzazione e la gestione dei dati**
 - **Offre meccanismi efficienti per l'accesso ai dati:**
 - **concorrenza**
 - **protezione e salvataggio dati (backup)**
 - **gestione di grandi masse di dati**

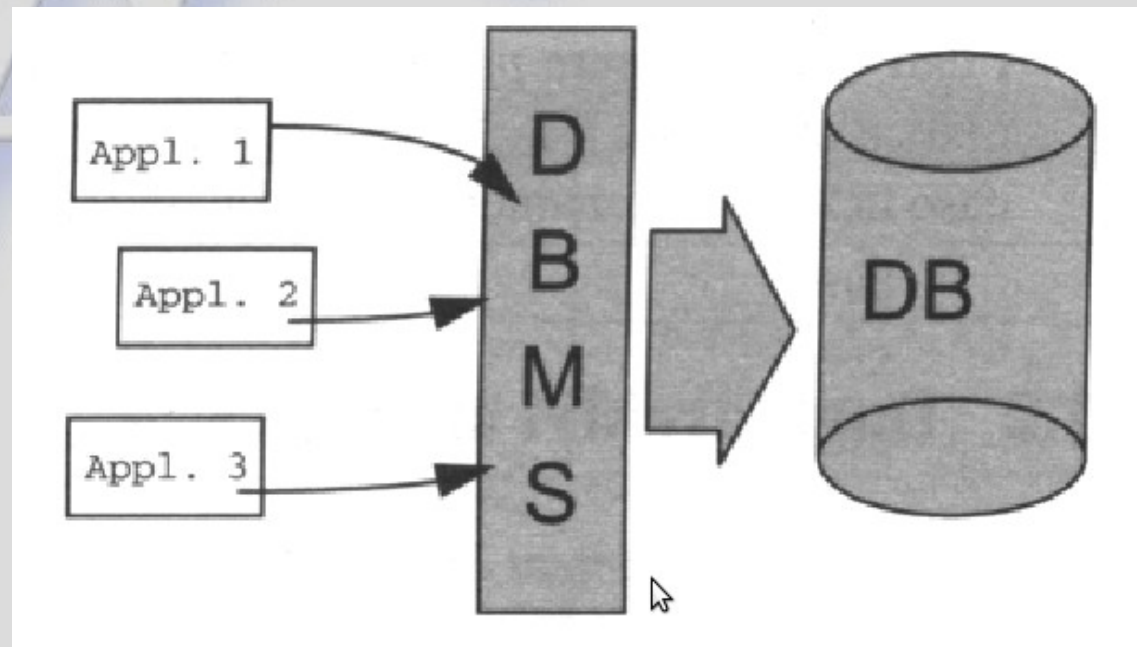
Basi di Dati e archivi su file

- **Dati gestiti tramite memorizzazione su file**
- **Più applicazioni accedono ai file**
- **Problemi:**
 - **Ridondanza e inconsistenza dei dati**
 - **Riservatezza dei dati**
 - **Necessità di regole di integrità dei dati**
 - **Accesso concorrente**
 - **Applicazioni complesse**



Basi di Dati e archivi su file

- **Dati gestiti da un DBMS**
- **Vantaggi:**
 - **I dati hanno una sola rappresentazione**
 - **L'accesso alla base di dati viene disciplinato dal DBMS**
 - **Alcuni vincoli di integrità possono essere definiti e verificati in modo automatico**
 - **Accesso regolato sino al singolo dato**
 - **Gestione della concorrenza**
 - **Transazioni**



Modelli per la gestione dei dati

- Modelli dei dati:
 - Gerarchico: basato su strutture ad albero
 - Reticolare: basato su strutture a grafo
 - **Relazionale: basato sul concetto di insieme e sulla strutturazione dei dati tramite tabelle**
 - **Ad oggetti: estende alle basi di dati l'approccio dei linguaggi di programmazione orientati agli oggetti**
- *Schema* di una base di dati: descrizione dei dati di uno specifico dominio applicativo
 - Definita tramite un'attività detta *database design*
 - *Istanza* (od *occorrenza*) di una base di dati: valore assunto dalla base di dati in un particolare istante di tempo

Categorie di linguaggi per DB

- Data Definition Language (DDL):
 - Utilizzato per definire lo schema della base di dati
 - Lo schema viene salvato nel *dizionario dei dati*
- Data Manipulation Language (DML):
 - Per formulare interrogazioni (o *query*) sulla base di dati
 - Per modificare il contenuto della base di dati
- Il linguaggio SQL supporta entrambi questi aspetti

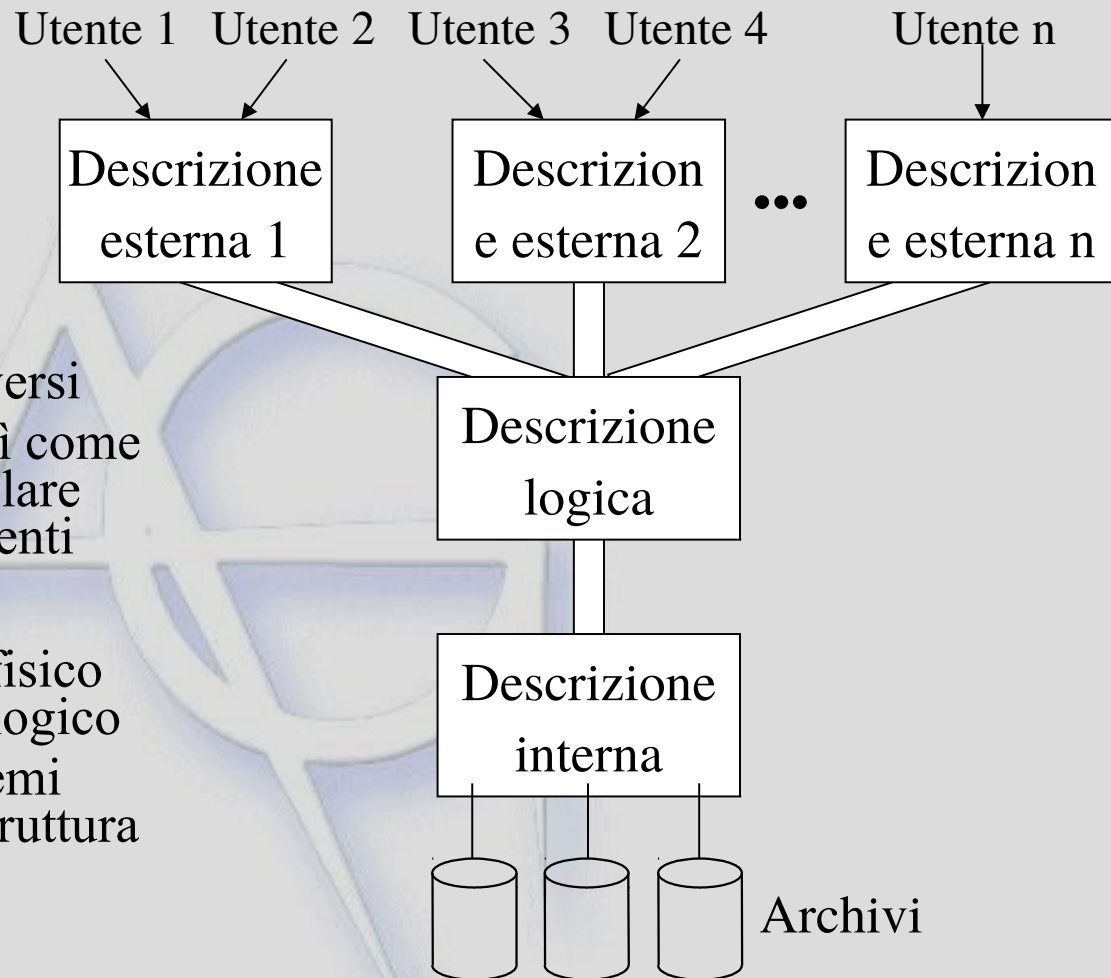
Livelli di astrazione di un DB

- Livelli di astrazione:

- *Fisico*: descrive la base di dati come un insieme di record in memoria di massa
- *Logico*: descrive la struttura di ciascun dato e i collegamenti tra dati diversi
- *Esterno*: presenta i dati così come vengono visti da un particolare utente o da una classe di utenti

- Livelli di indipendenza:

- *Fisica*: ridefinire il livello fisico senza modificare il livello logico
- *Logica*: definire nuovi schemi esterni senza modifica la struttura logica



Transazioni

- *Transazioni*: operazioni sui dati che non lasciano mai la base di dati in uno stato inconsistente
- Proprietà “acide” delle transazioni (ACID):
 - Atomicità: vi sono solo due possibili terminazioni
 - Successo: *tutte* le attività della transazione sono andate a buon fine, la base di dati è aggiornata (*commit*)
 - Fallimento: qualche attività della transazione non va a buon fine, la transazione fallisce *tutta*, la base di dati *non viene modificata* (*rollback*)
 - Consistenza: i vincoli di integrità dei dati non sono mai violati
 - Isolamento: transazioni concorrenti sono isolate le une dalle altre
 - Durabilità: una volta completata con successo, l’effetto della transazione sulla base di dati è permanente

Operatori e utenti di una base di dati

- Amministratore o Data Base Administrator (DBA):
 - Definisce gli schemi esterni, logico e fisico
 - Definisce i vincoli di integrità e i requisiti di riservatezza
 - A tal fine, il DBA utilizza il DDL
- Operatore o Terminalista:
 - Interagisce con un'applicazione che a sua volta si interfaccia con la base di dati
- Programmatore:
 - Scrive le applicazioni e le rende disponibili agli utenti
 - Utilizza il DML per interfacciare applicazione e base di dati
- Utente occasionale:
 - Formula query non previste dalle applicazioni, usando il DML o un'interfaccia apposita

Basi di dati relazionali



Il modello relazionale

- Proposto da E. F. Codd dell'IBM nel 1970 per favorire l'indipendenza dei dati dai programmi
- Reso disponibile come modello logico in DBMS reali nel 1981
- Oggi è il modello più diffuso ed è adottato dalla larga maggioranza dei DBMS disponibili a livello commerciale

Il modello relazionale

- Caratteristiche:
 - È basato su una semplice struttura dati – la *relazione o tabella*
 - Ha precise basi matematiche
- Vantaggi:
 - Semplice rappresentazione dei dati
 - Linguaggio (SQL – Structured Query Language) *dichiarativo* e non *imperativo* o *procedurale* (come è il C)
 - Facilità con cui possono essere espresse interrogazioni anche complesse

Relazione

- E' una *tabella* caratterizzata da:
 - Un numero fisso di colonne (dette *attributi*); ciascuna colonna assume valori estratti da uno stesso *dominio*
 - Esempi di domini: i numeri interi, le stringhe di caratteri, l'insieme $\{0, 1\}$, le date, ecc.
 - Un numero variabile di righe (dette *tuple*)
 - Ogni tupla è composta da una sequenza di valori, uno per ogni colonna – in pratica, equivale a un *record*
 - I campi delle tuple non possono avere *valori multipli*, ma un solo valore
 - Una relazione è un *insieme*, quindi:
 - non è definito alcun ordinamento fra le tuple
 - non esistono due tuple uguali tra loro

Relazioni

- *Istanza di una relazione*: insieme delle tuple presenti nella base di dati in un determinato istante (quindi, varia nel tempo)
- Caratteristiche di una relazione (o tabella):
 - **Grado**: il numero di colonne (attributi, campi del record)
 - **Cardinalità**: il numero di righe (numero delle tuple, dimensione della relazione)
 - **Schema di una relazione** (sua definizione): il nome della relazione seguito dalla lista, per ogni attributo:
 - nome dell'attributo (unico entro la relazione)
 - dominio dell'attributo
 - eventuali vincoli

Base di dati relazionale

- E' un insieme di relazioni:
 - *Schema logico di una base di dati*: l'elenco delle relazioni
 - *Istanza di una base di dati*: insieme delle istanze delle relazioni in un dato momento
- Ogni relazione ha un nome, univoco entro il DB
- *Schema fisico*: come lo schema logico è mappato su file e sulla memoria di massa
- *Schema esterno* o *Vista*: insieme di relazioni “estratte” dal DB per determinati utenti o applicazioni:
 - alcune sono identiche alle tabelle del DB
 - altre sono generate a partire dalle tabelle del DB, ma sono viste come se fossere tabelle vere e proprie

Esempio con 3 relazioni

infrazioni	<u>Codice</u>	Data	Vigile	Prov	Numero
	65524	3/9/1997	343	MI	3K9886
	87635	4/12/1997	476	MI	6D5563
	82236	4/12/1997	343	RM	7C5567
	35632	6/1/1998	476	RM	7C5567
	76543	5/3/1998	548	MI	6D5563

vigili	<u>Matricola</u>	Cognome	Nome
	343	Rossi	Luca
	476	Neri	Pino
	548	Nicolosi	Gino

automobili	<u>Prov</u>	<u>Numero</u>	Proprietario	...
	MI	3K9886	Nestore	...
	MI	6D5563	Nestore	...
	RM	7C5567	Menconi	...
	RM	1A6673	Mussone	...
	MI	5E7653	Marchi	...

Esempio: schema base di dati

```
CREATE TABLE CONTO-CORRENTE
```

```
(NUMERO-CC:      integer,  
 NOME:           char (20),  
 INDIRIZZO:     char (20),  
 SALDO:         decimal (14, 2)).
```

```
CREATE TABLE MOVIMENTO
```

```
(NUMERO-CC:      integer,  
 DATA-MOV:     date,  
 NUMERO-MOV:    integer,  
 IMPORTO:       decimal (14, 2),  
 CAUSALE:       char (1)).
```

Esempio: istanza base di dati

CONTO-CORRENTE

NUMERO-CC	NOME	INDIRIZZO	SALDO
1	Rossi	v. Anemoni 5	3,678.00
2	Bianchi	v. Bolla 64	664.00
3	Brunelli	v. Po 41	6,777.50
4	Grandi	v. Romolo 3	3,400.00

MOVIMENTO

NUMERO-CC	DATA-MOV	NUMERO-MOV	IMPORTO	CAUSALE
1	14-01-03	1	+200.00	V
1	14-01-03	2	-500.00	P
1	27-01-03	1	+2,700.00	S
4	27-01-03	1	+1,850.40	S
3	25-01-03	1	-650.00	A

Restrizione

- Sia R l'insieme degli attributi di una relazione
- R è un insieme di *tipi*, non dei valori specifici assunti dai campi delle tuple (o record)
- **Restrizione**: di una tupla t sugli attributi A di R , indicata con $t[A]$, è data dalla lista dei valori assunti da t sugli attributi A di R :
 - A è un insieme di attributi, e quindi un sottoinsieme di R
 - $t[A]$, è invece un insieme di valori
 - ovviamente, $t[R]$ coincide con la tupla stessa

Chiave di una relazione

- **Chiave di una relazione R** : un sottoinsieme K degli attributi di R , tale che valgano le proprietà:
 - *Unicità*: in una qualunque istanza di R , non possono esistere due tuple distinte di R la cui restrizione su K sia uguale
 - in altre parole, non ci possono essere due tuple aventi gli stessi valori di tutti i campi della chiave
 - *Minimalità*: non è possibile sottrarre a K un attributo senza che la condizione di unicità cessi di valere
- Le chiavi delle relazioni vengono individuate mediante esame del dominio applicativo e dei relativi vincoli
- Una chiave può essere composta da un solo attributo, da più attributi e, in certi casi, anche da tutti gli attributi

Chiave primaria

- Ogni relazione *deve avere almeno una chiave*
- Se la relazione ha più chiavi (*chiavi candidate*):
 - una di esse è scelta come *chiave primaria*
 - le altre sono dette *chiavi alternative*
- La chiave primaria viene utilizzata dal DBMS per ottimizzare le operazioni
- Criteri di scelta della chiave primaria:
 - Chiave candidata contenente il minor numero di attributi
 - Chiave candidata più frequentemente utilizzata nelle interrogazioni
- Le chiavi primarie non possono assumere valori nulli
- Le chiavi alternative possono assumere valori nulli

Valori nulli

- Non sempre sono disponibili informazioni complete sui valori di tutti gli attributi (campi) delle tuple
- Si introduce un valore speciale (valore nullo) che denota la mancanza di valore
 - dovuta a mancanza di informazione oppure al fatto che il valore non è ancora stato generato (ad es. la data di un evento non ancora accaduto)
- Denotiamo il valore nullo con il simbolo ‘?’, o con uno spazio vuoto, o con NULL (SQL)

Valori nulli

- Usare un valore x legale entro il dominio dell'attributo per denotare un valore nullo non è una soluzione
 - il suo uso non permetterebbe di distinguere il caso in cui x sia effettivamente il valore dell'attributo dal caso in cui x indichi il valore nullo
- Il valore nullo è un valore ammissibile per ogni dominio
- I linguaggi come SQL permettono di specificare nella definizione di una relazione quali attributi non possono mai assumere valori nulli

Associazioni

- Molto spesso occorre *associare* le tuple di due relazioni
 - Ad es., il C/C di un movimento, il vigile che ha comminato una multa, un dipartimento e i suoi impiegati, ecc.
- Tale associazione è realizzata inserendo in una relazione la chiave primaria della relazione associata
- Tali chiavi “inserite” sono dette *chiavi esterne*
- Es:
 - una tabella Dipartimenti ha per chiave primaria il *nome*
 - una tabella Impiegati ha come chiave esterna il *nome Dipartimento* (la chiave del Dipartimento dell'impiegato)
- La relazione con la chiave esterna è detta *referente* (nell'esempio, Impiegati)
- L'altra relazione è detta *riferita* (Dipartimenti)

Associazione: esempio

- Un movimento bancario si riferisce sempre a uno e un solo conto corrente
- Un C/C può avere molti movimenti associati
- L'associazione è rappresentata inserendo nella tabella MOVIMENTO un attributo (NUMERO-CC) contenente il codice (chiave primaria) del C/C relativo

NUMERO-CC	NOME	INDIRIZZO	SALDO
1	Rossi	v. Anemoni 5	3,678.00
2	Bianchi	v. Bolla 64	664.00
3	Brunelli	v. Po 41	6,777.50
4	Grandi	v. Romolo 3	3,400.00

NUMERO-CC	DATA-MOV	NUMERO-MOV	IMPORTO	CAUSALE
1	14-01-03	1	+200.00	V
1	14-01-03	2	-500.00	P
1	27-01-03	1	+2,700.00	S
4	27-01-03	1	+1,850.40	S
3	25-01-03	1	-650.00	A

Cardinalità delle associazioni

- La maggior parte delle associazioni sono da 1 a n
 - in pratica, associa le tuple di una tabella a **zero, una o più tuple** di una seconda tabella;
 - le tuple della 2^a tabella sono associate a **una e una sola tupla** della prima tabella (**o eventualmente a nessuna**)
 - Ad es., un dipartimento contiene più impiegati, un impiegato appartiene a uno e un solo dipartimento
- Tale associazione è realizzata avendo la seconda tabella come referente, e la prima come riferita
- Le associazioni da n a m mettono in relazione 0, 1 o più tuple di una tabella con 0, 1 o + tuple di una seconda tab.
- Queste associazioni si implementano con una tabella avente per colonne: le chiavi primarie delle due tabelle coinvolte

Integrità referenziale

- In caso di associazioni, è importante che tutte le tuple della relazione referente abbiano chiavi esterne che corrispondano a tuple della relazione riferita che abbiano effettivamente tale chiave primaria
- Tale condizione è detta *integrità referenziale (IR)*
 - Ad es., se in una tupla di Impiegati è presente una chiave esterna col nome di un Dipartimento che non corrisponde a un nome di una tupla della relazione Dipartimenti, l'IR è violata
- L'IR può essere violata:
 - nella relazione referente da inserimenti e modifiche del valore della chiave esterna
 - nella relazione riferita da cancellazioni e modifiche del valore della chiave primaria



Il linguaggio SQL

Il linguaggio SQL

- E' presente in tutti i DBMS relazionali
- Standard ANSI e ISO
- Mette a disposizione sia un DDL, sia un DML
- **Data Definition Language:** permette di creare lo schema della base di dati
 - creazione di tabelle
 - modifica/aggiunta di colonne, indici, vincoli
 - cancellazione di colonne, indici, vincoli, tabelle
- **Data Manipulation Language:** permette di scrivere le *query*
 - Basato sull'*algebra relazionale*
 - Operatori: selezione, proiezione, join, unione e differenza

Sintassi base SQL

- Le istruzioni non sono *case sensitive*: maiuscole e minuscole sono la stessa cosa
 - es. **SELECT**, **Select**, **select** sono la stessa parola chiave
- Le istruzioni terminano con punto e virgola
- **NULL** rappresenta il valore nullo
- Alcune istruzioni (ad es. **CREATE TABLE**) hanno un corpo tra parentesi tonde, con clausole separate da virgola

Il linguaggio SQL come DDL

- **CREATE TABLE:** crea una relazione a livello logico
 - definisce le colonne, dandone nome, tipo, vincoli:
 - **PRIMARY KEY:** fa parte della chiave primaria
 - **FOREIGN KEY:** fa parte di una chiave esterna
 - **NOT NULL:** il campo non può essere vuoto
- **CREATE INDEX ON:** crea indici per una certa relazione; gli indici permettono di accedere ai dati con maggiore efficienza
- **CREATE UNIQUE INDEX ON:** i valori dei campi componenti l'indice devono essere unici
- **DROP TABLE:** cancella una relazione
- **DROP INDEX:** cancella un indice

Esempio DDL

```
CREATE TABLE CANZONI
(CANTANTE CHAR(30) NOT NULL,
 TITOLO CHAR(40) NOT NULL,
 GENERE CHAR(12),
 ANNO NUMERIC(4),
 PRIMARY KEY (CANTANTE, TITOLO) );
CREATE UNIQUE INDEX CANZ-KEY ON
CANZONI (CANTANTE, TITOLO) ;
```

Un'istanza della tabella CANZONI

Tabella: Canzoni

Chiave Primaria			
Cantante	Titolo	Genere	Anno
Fabrizio De Andrè	Bocca di rosa	Cantautori	1967
Fabrizio De Andrè	Creuza de ma	Cantautori	1984
Fabrizio De Andrè	Dolcenera	Cantautori	1996
Vasco Rossi	Colpa di Alfredo	Cantautori	1980
Vasco Rossi	Stammi vicino	Rock	2011
Vasco Rossi	Buoni o cattivi	Rock	
Rihanna	Umbrella	Pop	2007
Rihanna	We found love	Pop	2011
Madonna	Love profusion	Folk	2003
Madonna	Like a virgin	Pop	1984
Madonna	True Blue	Rock	1986
Tazenda	Madre Terra	Rock	2008
Tazenda	Domo mia	Pop	2007
Tazenda	Fortza paris		1995
Tazenda	Mamoiada	Rock	1993

Esempio DDL

```
CREATE TABLE DISCHI
(CODICE          NUMERIC(4) PRIMARY KEY NOT NULL,
 CANTANTE       CHAR(30)  NOT NULL,
 TITOLO         CHAR(40)  NOT NULL,
 TIPO           CHAR(2)   NOT NULL,
 SCAFFALE       CHAR(3) ,
 FOREIGN KEY (CANTANTE, TITOLO) REFERENCES
 CANZONI (CANTANTE, TITOLO) ) ;
```

Un'istanza della tabella DISCHI

Tipo:

DI: disco vinile

CD: CD ROM

CA: cassetta

MP: MP3

Tabella: Dischi				
<i>C. Prim.</i>	<i>Chiave Esterna</i>			
Codice	Cantante	Titolo	Tipo	Scaffale
1005	Fabrizio De Andrè	Bocca di rosa	DI	A2
1006	Fabrizio De Andrè	Bocca di rosa	CD	A1
1011	Fabrizio De Andrè	Creuza de ma	CA	C3
1032	Fabrizio De Andrè	Dolcenera	MP	
1601	Vasco Rossi	Colpa di Alfredo	CD	A1
1604	Vasco Rossi	Stammi vicino	CD	A3
1609	Vasco Rossi	Stammi vicino	MP	
1616	Vasco Rossi	Buoni o cattivi	CD	C3
1400	Rihanna	Umbrella	CD	C3
1401	Rihanna	We found love	MP	
2003	Madonna	Love profusion	CD	C2
2020	Madonna	Like a virgin	DI	A2
2022	Madonna	True Blue	MP	
2103	Tazenda	Madre Terra	MP	
2130	Tazenda	Domo mia	CD	C3
2104	Tazenda	Domo mia	MP	
2120	Tazenda	Fortza paris	DI	A3
2108	Tazenda	Mamoiada	DI	B1
2121	Tazenda	Mamoiada	MP	

Esempio DDL

```
CREATE TABLE PRESTITI
(COD_DISCO    NUMERIC(4) NOT NULL,
 DATA        DATE       NOT NULL,
 PERSONA      CHAR(40)   NOT NULL,
 STATO        CHAR(5)    NOT NULL,
 PRIMARY KEY (COD_DISCO, DATA, PERSONA) ,
 FOREIGN KEY (COD_DISCO) REFERENCES
 DISCHI (CODICE) ) ;
```

Un'istanza della tabella PRESTITI

Stato:

RESO: restituito

COPIA: data copia
MP3

FUORI: in prestito

Ch. Esterna			
Chiave Primaria			
CodDisco	Data	Persona	Stato
1006	2012-1-12	Carlo Rossi	RESO
1401	2012-2-11	Gianna Verdi	COPIA
2020	2012-2-20	Carlo Rossi	FUORI
2130	2012-4-30	Pino Bianchi	FUORI
1006	2012-1-18	Pino Bianchi	FUORI
2104	2012-5-11	Gianna Verdi	COPIA

Linguaggio SQL: tipi di dati

- Tipi numerici: interi, reali, reali a virgola fissa, ecc.
- Tipi carattere: stringhe di lunghezza fissa, stringhe variabili ma di lunghezza massima fissata
- Tipi temporali (date, ore, timestamp, intervalli temporali...)
- Valori booleani
- Testi di grandi dimensioni (CLOB)
- Oggetti qualsiasi, ovvero sequenze di bit, di grandi dimensioni (BLOB)
 - per rappresentare immagini, audio, video, porzioni di memoria, ecc.

Tipi numerici e logici

- **INTEGER**: numeri interi (come **int** del C)
- **SMALLINT**, **BIGINT**: valori interi di precisione non maggiore e non minore di **INTEGER**
- **NUMERIC**, **DECIMAL**: valori decimali a virgola fissa con nr. max. di cifre e posizioni dopo la virgola dati:
 - **DECIMAL (p, s)** : numero con **p** cifre decimali complessive, e **s** a destra della virgola
- **REAL**: numero reale singola precisione (**float** del C)
- **DOUBLE PRECISION**: reale a doppia precisione
- **FLOAT (p)** : reale con precisione **p** dichiarata
- **BOOLEAN** : boolean (**TRUE** , **FALSE**)

Tipi carattere

- **CHAR**: stringhe di lunghezza fissata:
 - **CHAR (p)** : stringa con *p* caratteri – la stringa può essere più corta, ma in tal caso è completata con degli spazi
 - **CHAR** : nel caso *p* sia omesso, è una stringa di un carattere
- **VARCHAR**: (character varying) stringhe di lunghezza massima predefinita:
 - **VARCHAR (p)** : stringa con al massimo *p* caratteri – se la stringa è più corta, sono usati solo i caratteri necessari per rappresentarla (nessun spreco di spazio)
- Esempi di costanti stringa: `'Ciao'`, `'O'Brien'`, `''` (la stringa nulla)
- Si possono includere `\n` `\t` `\xxx` ... (come in C)

Tipi temporali

- **DATE** : date (anno, mese e giorno)
- **TIME** : ore (ora, minuto, secondo)
- **TIMESTAMP** : date e ore insieme (anno, mese, giorno, ora, minuto, secondo)
- **INTERVAL YEAR TO MONTH** : un intervallo temporale (anni e mesi); ad es. il valore:
interval '1-3' year to month vale 1 anno e 3 mesi (ricordiamo che SQL non è *case-sensitive*).
- **INTERVAL DAY TO TIME** : un intervallo temporale (giorni e ora); ad es. il valore:
interval '36 22:30' day to minute vale 36 giorni, 22 ore e 30'

Esempi di costanti letterali SQL

- 0 -1 345 45.6 2.6e-4 2.6E-4
- 'Ciao!' 'Questa e' ' una stringa'
- TRUE FALSE
- '2012-05-16' (formato ANSI per le date)
- '17:20:04' '2012-05-16 17:20:04'
- interval '1-3' year to month
- interval '36 22:30' day to minute

Le operazioni fondamentali

- **Proiezione:** genera una nuova tabella con un sottoinsieme delle colonne della tabella originale
 - le eventuali tuple duplicate sono eliminate
- **Selezione:** genera una nuova tabella con le stesse colonne della tabella originale, contenente solo le tuple che soddisfano una determinata condizione
- **Join:** a partire da due tabelle, si selezionano due sottoinsiemi dei loro attributi e con essi si genera una nuova tabella coi dati di tuple logicamente collegate

Il linguaggio SQL: select

- L'istruzione che permette di formulare le query, incluse proiezioni, selezioni e join
- Sintassi:
 - SELECT** nomi di attributi
 - FROM** nomi di relazioni
 - WHERE** condizioni di ricerca sui dati
- Rende come risultato una tabella, al limite di una sola riga e/o di una sola colonna
- **SELECT DISTINCT** elimina le righe duplicate nel risultato: se è omessa, eventuali duplicati nel risultato non sono eliminati e la tabella risultante non è una vera e propria relazione

Il linguaggio SQL: `select`

- Le tre clausole individuano rispettivamente:
 - **SELECT** : gli attributi da includere nel risultato; devono essere separati da virgole. L'asterisco * indica “tutti gli attributi” di una relazione
 - se le relazioni da cui estrarre i dati hanno attributi con lo stesso nome, questi vanno qualificati con:
NOME_RELAZIONE.NOME_ATTRIBUTO
 - **FROM** : le relazioni da cui estrarre i dati (una o più tabelle);
 - **WHERE** la condizione di ricerca sui dati. Questa condizione può essere molto complessa e coinvolgere nel suo interno altri blocchi SQL
 - se omessa, la tabella risultante contiene tutte le tuple delle tabelle di partenza

Esempio: istanza base di dati

Tabella: Canzoni

<i>Chiave Primaria</i>			
Cantante	Titolo	Genere	Anno
Fabrizio De Andrè	Bocca di rosa	Cantautori	1967
Fabrizio De Andrè	Creuza de ma	Cantautori	1984
Fabrizio De Andrè	Dolcenera	Cantautori	1996
Vasco Rossi	Colpa di Alfredo	Cantautori	1980
Vasco Rossi	Stammi vicino	Rock	2011
Vasco Rossi	Buoni o cattivi	Rock	
Rihanna	Umbrella	Pop	2007
Rihanna	We found love	Pop	2011
Madonna	Love profusion	Folk	2003
Madonna	Like a virgin	Pop	1984
Madonna	True Blue	Rock	1986
Tazenda	Madre Terra	Rock	2008
Tazenda	Domo mia	Pop	2007
Tazenda	Fortza paris		1995
Tazenda	Mamoiada	Rock	1993

Tabella: Dischi

<i>C. Prim.</i>	<i>Chiave Esterna</i>			
Codice	Cantante	Titolo	Tipo	Scaffale
1005	Fabrizio De Andrè	Bocca di rosa	DI	A2
1006	Fabrizio De Andrè	Bocca di rosa	CD	A1
1011	Fabrizio De Andrè	Creuza de ma	CA	C3
1032	Fabrizio De Andrè	Dolcenera	MP	
1601	Vasco Rossi	Colpa di Alfredo	CD	A1
1604	Vasco Rossi	Stammi vicino	CD	A3
1609	Vasco Rossi	Stammi vicino	MP	
1616	Vasco Rossi	Buoni o cattivi	CD	C3
1400	Rihanna	Umbrella	CD	C3
1401	Rihanna	We found love	MP	
2003	Madonna	Love profusion	CD	C2
2020	Madonna	Like a virgin	DI	A2
2022	Madonna	True Blue	MP	
2103	Tazenda	Madre Terra	MP	
2130	Tazenda	Domo mia	CD	C3
2104	Tazenda	Domo mia	MP	
2120	Tazenda	Fortza paris	DI	A3
2108	Tazenda	Mamoiada	DI	B1
2121	Tazenda	Mamoiada	MP	

Ch. Esterna

<i>Chiave Primaria</i>			
CodDisco	Data	Persona	Stato
1006	2012-1-12	Carlo Rossi	RESO
1401	2012-2-11	Gianna Verdi	COPIA
2020	2012-2-20	Carlo Rossi	FUORI
2130	2012-4-30	Pino Bianchi	FUORI
1006	2012-1-18	Pino Bianchi	FUORI
2104	2012-5-11	Gianna Verdi	COPIA

Esempio di selezione

Codice	Cantante	Titolo	Tipo	Scaffale
1005	Fabrizio De Andrè	Bocca di rosa	DI	A2
1006	Fabrizio De Andrè	Bocca di rosa	CD	A1
1011	Fabrizio De Andrè	Creuza de ma	CA	C3
1032	Fabrizio De Andrè	Dolcenera	MP	
1601	Vasco Rossi	Colpa di Alfredo	CD	A1
1604	Vasco Rossi	Stammi vicino	CD	A3
1609	Vasco Rossi	Stammi vicino	MP	
1616	Vasco Rossi	Buoni o cattivi	CD	C3
1400	Rihanna	Umbrella	CD	C3
1401	Rihanna	We found love	MP	
2003	Madonna	Love profusion	CD	C2
2020	Madonna	Like a virgin	DI	A2
2022	Madonna	True Blue	MP	
2103	Tazenda	Madre Terra	MP	
2130	Tazenda	Domo mia	CD	C3
2104	Tazenda	Domo mia	MP	
2120	Tazenda	Fortza paris	DI	A3
2108	Tazenda	Mamoiada	DI	B1
2121	Tazenda	Mamoiada	MP	

SELECT TITOLO

FROM DISCHI

WHERE TIPO = 'DI' **AND**
CANTANTE = 'Madonna';

Titolo

Like a virgin

Esempio di proiezione

Cantante	Titolo	Genere	Anno
Fabrizio De Andrè	Bocca di rosa	Cantautori	1967
Fabrizio De Andrè	Creuza de ma	Cantautori	1984
Fabrizio De Andrè	Dolcenera	Cantautori	1996
Vasco Rossi	Colpa di Alfredo	Cantautori	1980
Vasco Rossi	Stammi vicino	Rock	2011
Vasco Rossi	Buoni o cattivi	Rock	
Rihanna	Umbrella	Pop	2007
Rihanna	We found love	Pop	2011
Madonna	Love profusion	Folk	2003
Madonna	Like a virgin	Pop	1984
Madonna	True Blue	Rock	1986
Tazenda	Madre Terra	Rock	2008
Tazenda	Domo mia	Pop	2007
Tazenda	Fortza paris		1995
Tazenda	Mamoiada	Rock	1993

```
SELECT DISTINCT
      CANTANTE, GENERE
FROM   CANZONI;
```

Cantante	Genere
Fabrizio De Andrè	Cantautori
Vasco Rossi	Cantautori
Vasco Rossi	Rock
Rihanna	Pop
Madonna	Folk
Madonna	Pop
Madonna	Rock
Tazenda	Rock
Tazenda	Pop
Tazenda	

Esempio di join (dischi in prestito e persone che li hanno)

```
SELECT    CANTANTE, TITOLO, PERSONA
FROM      DISCHI, PRESTITI
WHERE     STATO = 'FUORI'
AND      CODDISCO = CODICE;
```

Cantante	Titolo	Persona
Fabrizio De Andrè	Bocca di rosa	Pino Bianchi
Madonna	Like a virgin	Carlo Rossi
Tazenda	Domo mia	Pino Bianchi