



## Linguaggio SQL: costrutti avanzati

### Gestione degli indici

# Gestione degli indici

- Introduzione
- Strutture fisiche di accesso
- Definizione di indici in SQL
- Progettazione fisica



# Gestione degli indici

## Introduction

# Organizzazione fisica dei dati

- All'interno di un DBMS relazionale, i dati sono rappresentati come collezioni di record memorizzati in uno o più file
  - l'organizzazione fisica dei dati all'interno di un file influenza il tempo di accesso alle informazioni
  - ogni organizzazione fisica dei dati rende alcune operazioni efficienti e altre onerose
- Non esiste un'organizzazione fisica dei dati che sia efficiente per qualunque tipo di lettura e scrittura dei dati

# Base di dati di esempio

Dipendente (CodD, Nome, Cognome, DataNascita, Residenza, SalarioMensile)

## Dipendente

<u>CodD</u>	Nome	Cognome	DataNascita	Residenza	SalarioMensile
D1	Elena	Rossi	02/01/1967	Torino	2.200,00
D2	Andrea	Verdi	04/05/1973	Como	1.100,00
D3	Giulia	Neri	14/04/1975	Roma	2.200,00
D4	Paolo	Bianchi	12/08/1970	Milano	3.000,00
D5	Daniele	Bruno	13/02/1968	Como	1.900,00
D6	Antonio	Bianco	25/11/1964	Venezia	1.700,00
D7	Lucia	Carta	09/04/1971	Alessandria	2.500,00
D8	Luca	Draghi	03/08/1973	Roma	2.400,00
D9	Tania	Bravo	11/06/1976	Asti	1.800,00
D10	Irene	Massa	28/04/1979	Torino	2.600,00
D11	Lia	Massa	15/05/1965	Milano	3.500,00
D12	Alessio	Morra	19/06/1969	Como	1.200,00

## ➤ Tabella DIPENDENTE

- è memorizzata dal DBMS relazionale in un file

## ➤ Interrogazione

- visualizzare le informazioni sui dipendenti residenti a Como

```
SELECT *  
FROM DIPENDENTE  
WHERE Residenza='Como';
```

# Esempio: tabella DIPENDENTE

## Dipendente

<b>CodD</b>	<b>Nome</b>	<b>Cognome</b>	<b>DataNascita</b>	<b>Residenza</b>	<b>SalarioMensile</b>
D1	Elena	Rossi	02/01/1967	Torino	2.200,00
D2	Andrea	Verdi	04/05/1973	Como	1.100,00
D3	Giulia	Neri	14/04/1975	Roma	2.200,00
D4	Paolo	Bianchi	12/08/1970	Milano	3.000,00
D5	Daniele	Bruno	13/02/1968	Como	1.900,00
D6	Antonio	Bianco	25/11/1964	Venezia	1.700,00
D7	Lucia	Carta	09/04/1971	Alessandria	2.500,00
D8	Luca	Draghi	03/08/1973	Roma	2.400,00
D9	Tania	Bravo	11/06/1976	Asti	1.800,00
D10	Irene	Massa	28/04/1979	Torino	2.600,00
D11	Lia	Massa	15/05/1965	Milano	3.500,00
D12	Alessio	Morra	19/06/1969	Como	1.200,00

# Esempio: risultato dell'interrogazione

## Dipendente

<b>CodD</b>	<b>Nome</b>	<b>Cognome</b>	<b>DataNascita</b>	<b>Residenza</b>	<b>SalarioMensile</b>
D1	Elena	Rossi	02/01/1967	Torino	2.200,00
D2	Andrea	Verdi	04/05/1973	Como	1.100,00
D3	Giulia	Neri	14/04/1975	Roma	2.200,00
D4	Paolo	Bianchi	12/08/1970	Milano	3.000,00
D5	Daniele	Bruno	13/02/1968	Como	1.900,00
D6	Antonio	Bianco	25/11/1964	Venezia	1.700,00
D7	Lucia	Carta	09/04/1971	Alessandria	2.500,00
D8	Luca	Draghi	03/08/1973	Roma	2.400,00
D9	Tania	Bravo	11/06/1976	Asti	1.800,00
D10	Irene	Massa	28/04/1979	Torino	2.600,00
D11	Lia	Massa	15/05/1965	Milano	3.500,00
D12	Alessio	Morra	19/06/1969	Como	1.200,00



## Esempio: esecuzione dell'interrogazione

- Operazioni effettuate dal DBMS per eseguire l'interrogazione SQL
  - lettura sequenziale dell'intero file
  - durante la lettura, selezione dei record dei dipendenti con residenza a Como
  - visualizzazione dei record
- Esistono organizzazioni fisiche dei dati su file che permettano di evitare la scansione completa del file?

## Esempio: struttura fisica 1

➤ I record della tabella DIPENDENTE sono memorizzati in ordine alfabetico di Residenza

# Esempio: struttura fisica 1

## Dipendente

<b>CodD</b>	<b>Nome</b>	<b>Cognome</b>	<b>DataNascita</b>	<b>Residenza</b>	<b>SalarioMensile</b>
D7	Lucia	Carta	09/04/1971	Alessandria	2.500,00
D9	Tania	Bravo	11/06/1976	Asti	1.800,00
D2	Andrea	Verdi	04/05/1973	Como	1.100,00
D5	Daniele	Bruno	13/02/1968	Como	1.900,00
D12	Alessio	Morra	19/06/1969	Como	1.200,00
D4	Paolo	Bianchi	12/08/1970	Milano	3.000,00
D11	Lia	Massa	15/05/1965	Milano	3.500,00
D3	Giulia	Neri	14/04/1975	Roma	2.200,00
D8	Luca	Draghi	03/08/1973	Roma	2.400,00
D1	Elena	Rossi	02/01/1967	Torino	2.200,00
D10	Irene	Massa	28/04/1979	Torino	2.600,00
D6	Antonio	Bianco	25/11/1964	Venezia	1.700,00

# Esempio: struttura fisica 1

## Dipendente

<b>CodD</b>	<b>Nome</b>	<b>Cognome</b>	<b>DataNascita</b>	<b>Residenza</b>	<b>SalarioMensile</b>
D7	Lucia	Carta	09/04/1971	Alessandria	2.500,00
D9	Tania	Bravo	11/06/1976	Asti	1.800,00
D2	Andrea	Verdi	04/05/1973	Como	1.100,00
D5	Daniele	Bruno	13/02/1968	Como	1.900,00
D12	Alessio	Morra	19/06/1969	Como	1.200,00
D4	Paolo	Bianchi	12/08/1970	Milano	3.000,00
D11	Lia	Massa	15/05/1965	Milano	3.500,00
D3	Giulia	Neri	14/04/1975	Roma	2.200,00
D8	Luca	Draghi	03/08/1973	Roma	2.400,00
D1	Elena	Rossi	02/01/1967	Torino	2.200,00
D10	Irene	Massa	28/04/1979	Torino	2.600,00
D6	Antonio	Bianco	25/11/1964	Venezia	1.700,00

## Esempio: struttura fisica 1

- Operazioni da effettuare per eseguire l'interrogazione (versione semplice)
  - lettura sequenziale del file fino al primo record con Residenza uguale a Como
  - lettura sequenziale di tutti i record con Residenza uguale a Como, fino al primo record con Residenza diversa da Como
  - visualizzazione dei record dei dipendenti con Residenza a Como
- Esistono metodi più efficienti per consultare la stessa struttura dati

## Esempio: struttura fisica 1

- Organizzazione fisica progettata specificamente per l'interrogazione proposta
  - memorizzazione dei record nel file in ordine alfabetico di Residenza
- File con *struttura sequenziale ordinata*

# Esempio: struttura fisica 1

## ➤ Vantaggi

- si evita la lettura sequenziale dell'intero file

## ➤ Svantaggi

- mantenere l'ordinamento è computazionalmente oneroso
  - occorre riorganizzare il file quando sono inseriti, aggiornati o cancellati i record
- l'organizzazione fisica proposta non è efficiente quando occorre eseguire altre tipologie di interrogazioni
  - esempio: selezionare i dipendenti che percepiscono un salario mensile superiore a 2000 euro

## Esempio: struttura fisica 2

- È possibile definire strutture fisiche accessorie che permettano di facilitare l'accesso ai dati



## Esempio: struttura fisica 2

Struttura fisica  
accessoria

Residenza	Locazione fisica
Alessandria	
Asti	
Como	
Milano	
...	...
Venezia	

Dipendente

<u>CodD</u>	...	Residenza	...
D1	...	Torino	...
D2	...	Como	...
D3	...	Roma	...
D4	...	Milano	...
D5	...	Como	...
D6	...	Venezia	...
D7	...	Alessandria	...
D8	...	Roma	...
D9	...	Asti	...
D10	...	Torino	...
D11	...	Milano	...
D12	..	Como	...

## Esempio: struttura fisica 2

Struttura fisica  
accessoria

Residenza	Locazione fisica
Alessandria	
Asti	
Como	
Milano	
...	...
Venezia	

Dipendente

<u>CodD</u>	...	Residenza	...
D1	...	Torino	...
D2	...	Como	...
D3	...	Roma	...
D4	...	Milano	...
D5	...	Como	...
D6	...	Venezia	...
D7	...	Alessandria	...
D8	...	Roma	...
D9	...	Asti	...
D10	...	Torino	...
D11	...	Milano	...
D12	..	Como	...

## Esempio: struttura fisica 2

- Struttura fisica accessoria con accesso associativo ai dati
  - realizzata sull'attributo Residenza
- L'attributo Residenza è il campo chiave della struttura
  - per ogni valore assunto dall'attributo Residenza si memorizzano
    - tutte le locazioni fisiche dei record corrispondenti al valore del campo chiave
  - la locazione fisica
    - indica la posizione di un record all'interno del file
    - permette di accedere direttamente al record d'interesse (alla pagina fisica che lo contiene)

## Esempio: struttura fisica 2

- Operazioni da effettuare per eseguire l'interrogazione
  - lettura della struttura fisica accessoria per recuperare le locazioni fisiche dei record corrispondenti a Residenza=Como
  - accesso diretto solo ai record del file associati alla Residenza Como
  - visualizzazione dei record di interesse
- Esistono metodi diversi per reperire rapidamente le informazioni nelle strutture fisiche accessorie

## Esempio: struttura fisica 2

### ➤ Vantaggi

- si evita la lettura completa e sequenziale del file
  - come nel caso della struttura fisica 1
- accesso diretto *solo* ai record di interesse
- il costo di mantenimento della struttura accessoria è meno oneroso rispetto al costo di mantenimento del file con struttura ordinata

## Esempio: struttura fisica 2

### ➤ Svantaggi

- occupazione di spazio maggiore
  - è necessario spazio supplementare per memorizzare la struttura fisica accessoria
- la struttura accessoria può essere utilizzata solo quando nell'interrogazione compare l'attributo Residenza
  - possono essere necessarie strutture accessorie per più attributi o combinazioni di attributi

- Gli *indici* sono le strutture fisiche accessorie offerte dai DBMS relazionali per migliorare l'efficienza delle operazioni di accesso ai dati
  - sono realizzati mediante strutture fisiche di tipo diverso
    - alberi
    - hash table
- Le istruzioni per la gestione degli indici non fanno parte dello standard SQL



## Gestione degli indici

### Strutture fisiche di accesso



# Strutture fisiche di accesso

- Le strutture fisiche di accesso descrivono il modo in cui i dati sono organizzati in memoria secondaria per garantire operazioni di ricerca e modifica dei dati *efficienti*
- Sono classificabili in
  - strutture sequenziali
  - strutture ad albero
  - strutture ad accesso calcolato

# Strutture fisiche di accesso

- Ogni DBMS relazionale dispone di diverse varianti delle strutture fisiche di base
  - la descrizione delle strutture interne di memorizzazione dei dati non è pubblicamente disponibile
  - le strutture fisiche sono diverse per DBMS diversi

# Struttura sequenziale

➤ È caratterizzata da

- una disposizione sequenziale dei record in memoria secondaria
- blocchi di memoria consecutivi nel file

➤ Ordinamento della sequenza

- la sequenza dei record dipende dal valore assunto da un campo di ordinamento, detto *chiave di ordinamento*
  - composto da uno o più attributi

# Struttura ad albero

- Accesso associativo efficiente ai dati, basato sul valore di un campo chiave
  - la chiave può essere composta da uno o più attributi
- Una *struttura ad albero* permette di raggiungere l'insieme delle locazioni fisiche dei record corrispondenti al valore prescelto del campo chiave
  - la locazione fisica di un record indica la posizione fisica del record all'interno del file in memoria secondaria

➤ Esempi: B-tree, B<sup>+</sup>-tree

# Struttura ad accesso calcolato

- Accesso associativo efficiente ai dati, basato sul valore di un campo chiave
  - la chiave può essere composta da uno o più attributi
- Richiede un *algoritmo di calcolo* per localizzare il blocco fisico del file contenente i record corrispondenti al valore del campo chiave
- Non richiede un ordinamento specifico dei record in memoria secondaria
- Esempio: struttura hash



## Gestione degli indici

### Definizione di indici in SQL

# Definizione di indici in SQL

- Il linguaggio SQL offre le seguenti istruzioni per la definizione degli indici
  - creazione di un indice
    - CREATE INDEX
  - cancellazione di un indice
    - DROP INDEX
- Le istruzioni per la gestione degli indici non fanno parte dello standard SQL

# Creazione di un indice

```
CREATE INDEX NomeIndice  
ON NomeTabella (ElencoAttributi)
```

➤ Crea un indice

- con nome *NomeIndice*
- sulla tabella *NomeTabella*
- definito sugli attributi in *ElencoAttributi*



# Creazione di un indice

```
CREATE INDEX NomeIndice  
ON NomeTabella (ElencoAttributi)
```

➤ L'ordine in cui compaiono gli attributi in *ElencoAttributi* è *importante*

- le chiavi dell'indice sono ordinate
  - prima in base al primo attributo in *ElencoAttributi*
  - a pari valore del primo attributo sui valori del secondo attributo
  - e così via, in ordine, fino all'ultimo attributo

# Base di dati di esempio

## Dipendente

<b>CodD</b>	<b>Nome</b>	<b>Cognome</b>	<b>DataNascita</b>	<b>Residenza</b>	<b>SalarioMensile</b>
D1	Elena	Rossi	02/01/1967	Torino	2.200,00
D2	Andrea	Verdi	04/05/1973	Como	1.100,00
D3	Giulia	Neri	14/04/1975	Roma	2.200,00
D4	Paolo	Bianchi	12/08/1970	Milano	3.000,00
D5	Daniele	Bruno	13/02/1968	Como	1.900,00
D6	Antonio	Bianco	25/11/1964	Venezia	1.700,00
D7	Lucia	Carta	09/04/1971	Alessandria	2.500,00
D8	Luca	Draghi	03/08/1973	Roma	2.400,00
D9	Tania	Bravo	11/06/1976	Asti	1.800,00
D10	Irene	Massa	28/04/1979	Torino	2.600,00
D11	Lia	Massa	15/05/1965	Milano	3.500,00
D12	Alessio	Morra	19/06/1969	Como	1.200,00

## Esempio n.1

- Creazione di un indice sull'attributo Residenza della tabella DIPENDENTE

```
CREATE INDEX IndiceResidenza  
ON DIPENDENTE (Residenza)
```

- Creazione di un indice sulla combinazione di attributi Cognome e Nome della tabella DIPENDENTE

```
CREATE INDEX IndiceCognomeNome  
ON DIPENDENTE(Cognome, Nome)
```

- L'indice è definito congiuntamente sui due attributi
- Le chiavi dell'indice sono ordinate
  - prima in base al valore dell'attributo Cognome
  - a pari valore dell'attributo Cognome, sul valore dell'attributo Nome

# Cancellazione di un indice

`DROP INDEX` *NomeIndice*

➤ Elimina l'indice con nome *NomeIndice*

➤ Il comando è utilizzato quando

- l'indice non è più utilizzato
- il miglioramento delle prestazioni non è sufficiente
  - ridotta riduzione del tempo di risposta per le interrogazioni
  - rallentamento degli aggiornamenti causato dal mantenimento dell'indice

➤ Cancellare l'indice IndiceResidenza

```
DROP INDEX IndiceResidenza
```



## Gestione degli indici

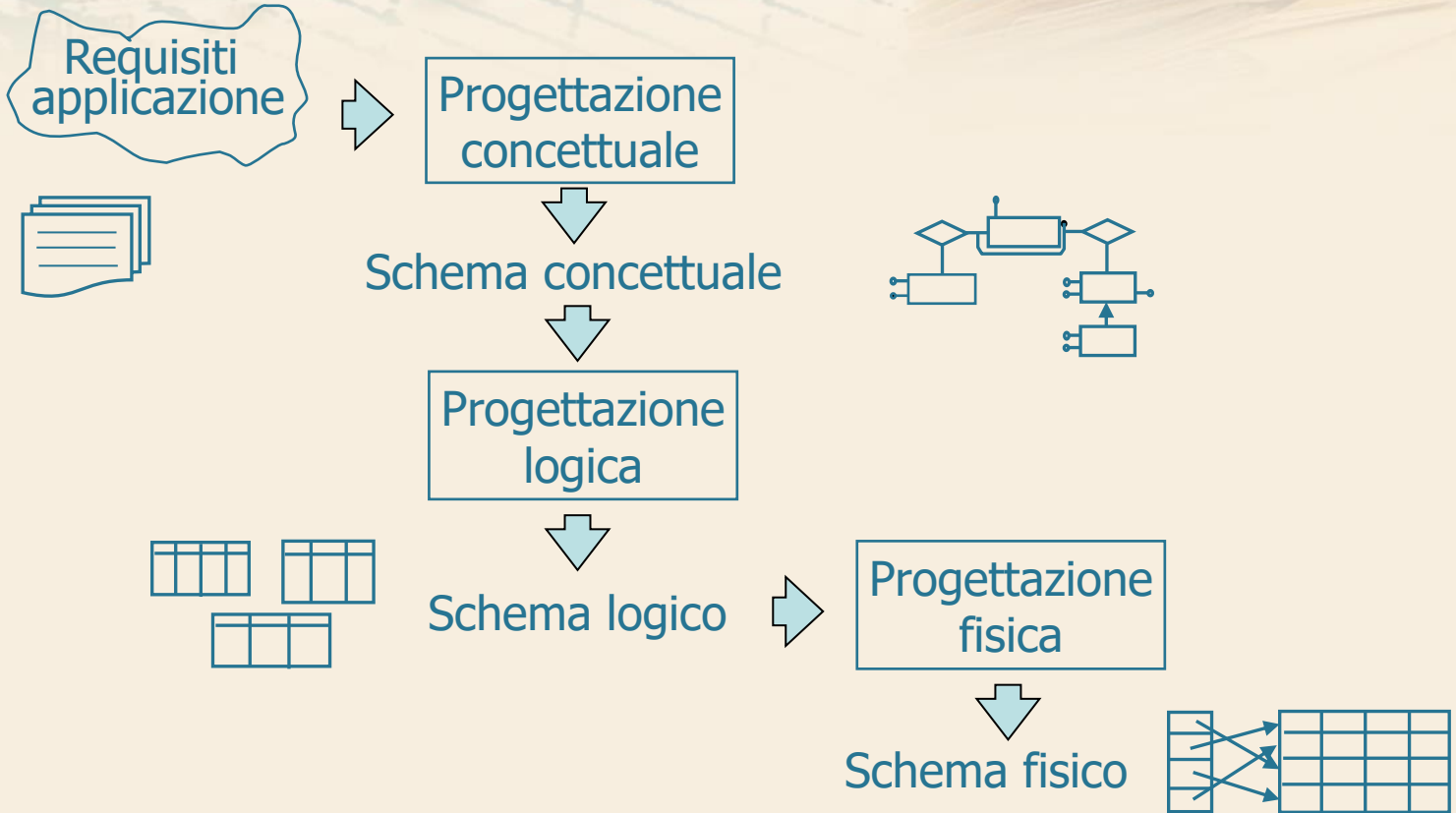
### Progettazione fisica

# Progettazione fisica

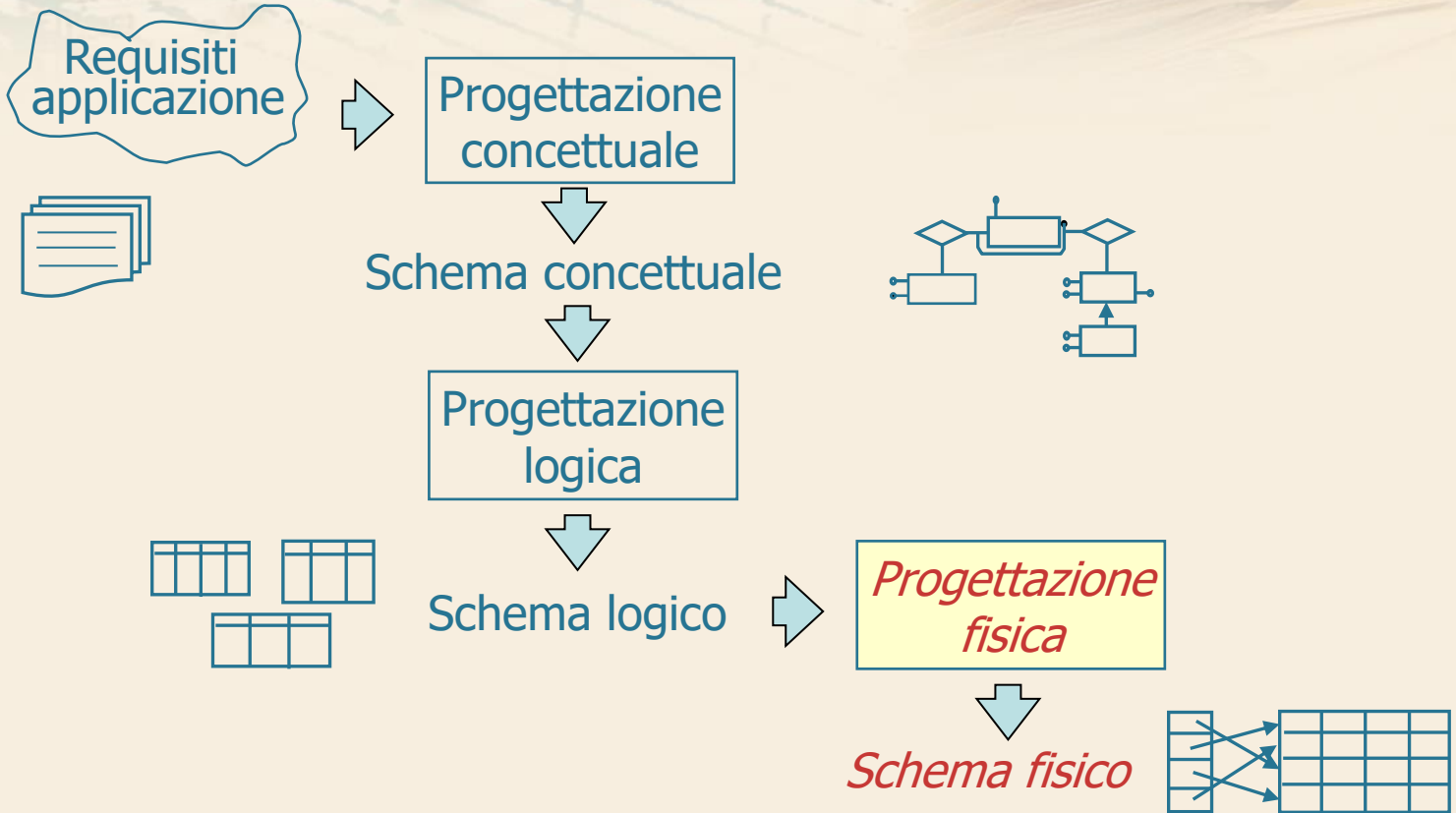
- È la fase finale della progettazione di una base di dati
- richiede la scelta del DBMS utilizzato
  - è legata alle caratteristiche del prodotto prescelto



# Fasi della progettazione di basi di dati



# Fasi della progettazione di basi di dati



# Progettazione fisica: dati di ingresso

- Schema logico della base di dati
- Caratteristiche del DBMS prescelto
  - opzioni disponibili a livello fisico
    - strutture fisiche di memorizzazione
    - indici
- Volume dei dati
  - cardinalità delle tabelle
  - cardinalità e distribuzione dei valori del dominio degli attributi

# Progettazione fisica: dati di ingresso

## ➤ Stima del carico applicativo

- interrogazioni più importanti e loro frequenza
- operazioni di aggiornamento più importanti e loro frequenza
- requisiti sul tempo di risposta per interrogazioni/aggiornamenti importanti

# Progettazione fisica: risultato

- Schema fisico della base di dati
  - organizzazione fisica delle tabelle
  - indici
- Parametri di memorizzazione e funzionamento
  - dimensioni iniziali dei file, possibilità di espansione, spazio iniziale libero, ...

# Procedimento

- La progettazione fisica è svolta in modo empirico, con un approccio per tentativi
  - non esistono metodologie di riferimento

## ➤ Caratterizzazione del carico applicativo

- per ogni interrogazione rilevante è necessario definire
  - relazioni a cui accede
  - attributi da visualizzare
  - attributi coinvolti in selezioni/join
  - grado di selettività delle condizioni di selezione
- per ogni aggiornamento rilevante è necessario definire
  - tipo di aggiornamento
    - inserimento, cancellazione, modifica
  - relazione ed eventuali attributi coinvolti
  - grado di selettività delle condizioni di selezione

## ➤ Scelte da operare

- strutturazione fisica dei file che contengono le tabelle
  - ordinati, non ordinati
- scelta degli attributi da indicizzare
  - pilotata dalla stima del carico applicativo e dal volume dei dati
- per ogni indice definizione del tipo
  - per esempio, hash oppure B-tree
- eventuali variazioni dello schema
  - partizionamenti orizzontali in memoria secondaria
  - denormalizzazione di tabelle
    - utilizzata nei data warehouse



- Se il risultato non è soddisfacente
  - *Tuning*, aggiungendo e togliendo indici
- È un procedimento guidato dalla disponibilità di strumenti che permettano di
  - verificare il piano di esecuzione adottato dal DBMS prescelto
    - il piano di esecuzione definisce la sequenza di attività svolte dal DBMS per eseguire un'interrogazione
      - metodi di accesso ai dati
      - metodi di join
  - valutare il costo di esecuzione di alternative diverse