



---

Università degli Studi di Parma  
Facoltà di Scienze MM. FF. NN.  
Corso di Laurea in Informatica  
Ingegneria del Software

**La fase di Progettazione:  
le basi di dati**  
**Giulio Destri**



Ing. del Sw: Base dati - 1

Giulio Destri - © AreaSP for Univ. Parma, 2006

**Scopo del modulo**

---

**Definire**

**I concetti fondamentali delle  
Basi di Dati Relazionali  
e delle operazioni  
ad esse associate**

Ing. del Sw: Base dati - 2

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Argomenti

---

- I database
- I DBMS
- Definizione e progettazione
- Il linguaggio SQL

## Definizione di Database

---

***Un insieme strutturato di  
Dati  
che - in un dato momento  $n$  –  
fornisce una  
rappresentazione  
semplificata di una realtà in  
evoluzione.***

## Definizione di Database

---

- *Insieme*
- *Strutturato*
- *Dati*
- *Momento  $n$*
- *Rappresentazione semplificata*
- *Realtà in evoluzione*

## I dati

---

- Sono una rappresentazione della realtà di interesse
- Devono essere organizzati, classificati e archiviati in modo da poter essere facilmente reperiti e trasformati da fatti grezzi in informazioni utili a significative decisioni
- Rendono persistenti gli oggetti

## Dati e Applicazioni

---

- Le Applicazioni elaborano dati
- Le Applicazioni trasformano le informazioni
- Le informazioni in ingresso sono dette dati di Input
- Il prodotto dell'elaborazione sono i dati di Output
- I dati durano oltre le applicazioni

## Il Database

---

- Un database è un insieme di informazioni relative ad un particolare scopo
- Un DBMS (database management system) consente di gestire l'accesso alle informazioni e la loro memorizzazione

## Caratteristiche di un DBMS

---

- Possibilità di definire la struttura dei dati, accedere ed elaborarli tramite linguaggi di alto livello
- Capacità di accedere in maniera efficiente a grandi quantità di dati
- Gestire l'accesso controllato ai dati e garantire la concorrenza nell'accesso alle informazioni preservando l'integrità dei dati

---

## VANTAGGI DEI DATA BASE RISPETTO ALLE APPLICAZIONI

## FACILITA' DI PROGETTAZIONE

---

- Permette diverse "visibilità" dei dati adatte a ciascuna applicazione
- Permette la gestione automatica e controllata dei dati
  - Es.: se viene cancellato un record da cui ne dipendono logicamente degli altri, anche questi vengono cancellati
- Fornisce estensioni dei linguaggi esistenti o linguaggi ad alto livello, orientati a facilitare la manipolazione dei dati

Ing. del Sw: Base dati - 11

Giulio Destri - © AreaSP for Univ. Parma, 2006

## FACILITA' DI GESTIONE

---

- Permette di modificare e arricchire le strutture di dati nel tempo senza modificare le applicazioni esistenti
- Permette di aggiungere nuovi campi all'interno di tracciati record già esistenti senza modificare le applicazioni che già li utilizzavano
- Rafforza l'uso di standard di organizzazione e di programmazione (organizzazione e la struttura dei dati è centralizzata a cura del System Administrator)
- Rende esplicite le relazioni logiche tra i dati, che altrimenti restano nascoste all'interno delle elaborazioni delle varie applicazioni

Ing. del Sw: Base dati - 12

Giulio Destri - © AreaSP for Univ. Parma, 2006

## EFFICACIA DELLE IMPLEMENTAZIONI

---

- Riduce la ridondanza dei dati, quindi:
  - riduce l'occupazione fisica di memoria
  - garantisce la congruenza dei dati
  - Migliora l'utilizzo di memoria RAM gestendo i buffers di I/O all'interno di pool.

## PUNTO DI VISTA DELL'UTENTE

---

- la disponibilità di un linguaggio per definire le caratteristiche dei dati e le strutture logiche che li legano
- la disponibilità di estensioni dei linguaggi di programmazione che semplificano le operazioni sui dati
- la disponibilità di accedere direttamente ai dati

## PUNTO DI VISTA DEL SISTEMA

---

- un insieme di codice che soddisfa le richieste di operazioni sui files da parte dei programmi applicativi
- un insieme di utility per:
  - creare/modificare la struttura dei DB
  - in alcuni casi per operare direttamente sui dati
  - per effettuare operazioni di manutenzione, ristrutturazione, recovery

Ing. del Sw: Base dati - 15 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)

---

## Terminologia dei database

Ing. del Sw: Base dati - 16 [Giulio Destri - © AreaSP for Univ. Parma, 2006](#)



## La terminologia dei database relazionali

---

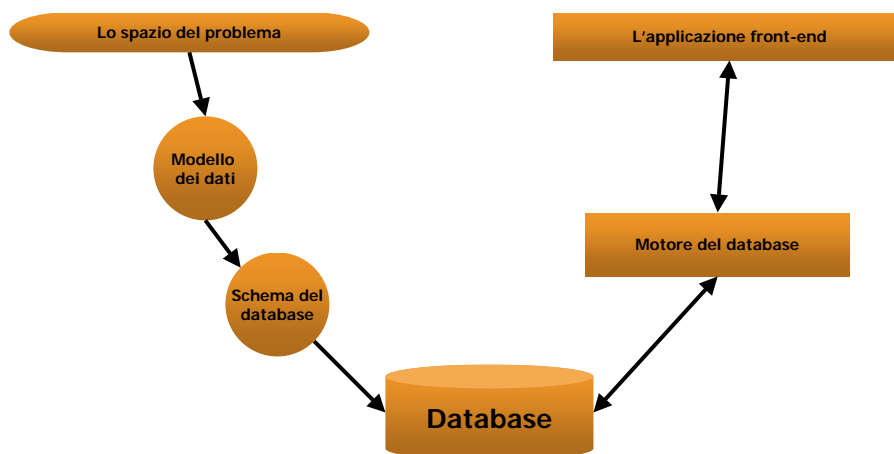
- Lo spazio del problema
  - Il modello dei dati
  - Lo schema del database
  - Il motore del database
  - L'applicazione front-end
- } Il database

Ing. del Sw: Base dati - 17

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Rappresentazione grafica della terminologia

---



Ing. del Sw: Base dati - 18

Giulio Destri - © AreaSP for Univ. Parma, 2006

## **Il modello relazionale**

---

**Il modello relazionale nella modellazione dei dati è stato applicato inizialmente alla fine degli anni 60' da E.F. Codd**

**Questo modello definisce:**

**il modo in cui i dati possono essere rappresentati (struttura dei dati)**

**il modo in cui i dati possono essere protetti (integrità dei dati)**

**Le operazioni che si possono eseguire sui dati (manipolazione dei dati)**

## **Il modello relazionale**

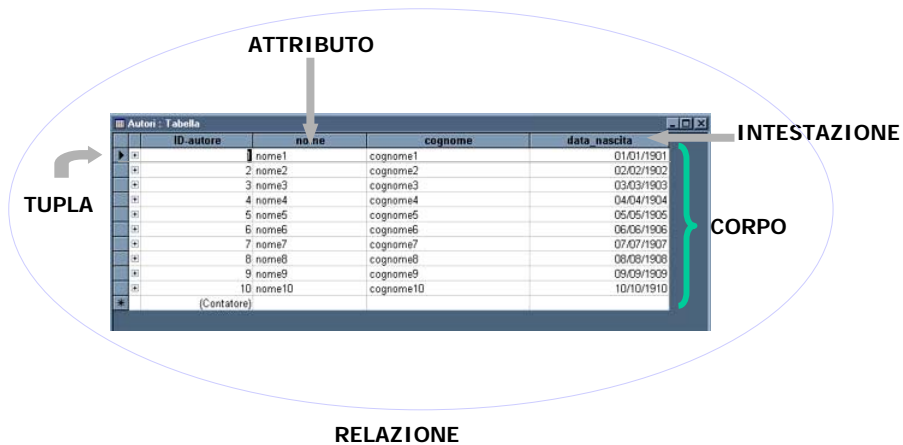
---

**I sistemi di database relazionali devono possedere determinate e precise caratteristiche:**

- **Tutti i dati sono rappresentati in strutture ordinate in righe e colonne (Relazioni)**
- **Tutti i valori sono scalari cioè in ogni cella formata dall'incontro di una riga con una colonna deve esserci uno ed un solo valore**
- **Ogni operazione è effettuata su un'intera relazione ed ha come risultato una relazione intera**

# Terminologia relazionale

---



Ing. del Sw: Base dati - 21

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Database Relazionale

---

- Un database relazionale è un insieme di tabelle, collegate tramite relazioni di norma corrispondenti alle entità definite nello schema concettuale
- Il formato e il contenuto delle tabelle è indipendente dalle modalità di registrazione fisica su disco
- L'accesso avviene tramite SQL

Ing. del Sw: Base dati - 22

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Integrità

---

Un database deve garantire l'integrità dei dati

- **Integrità semantica**, la modifica dei dati deve avvenire all'interno del dominio previsto per l'attributo
- **Integrità di contesa**, il valore dei dati va protetto da situazioni conflittuali di modifica contemporanea (lock)

## Integrità

---

- **Integrità di entità**, nessun attributo che fa parte di una chiave primaria può essere nullo
- **Integrità referenziale**, per ogni chiave esterna non nulla deve esistere la chiave primaria a cui fare riferimento
- **Integrità definita dall'utente**, controlli di dominio per garantire la correttezza formale dei dati

## Integrità

---

In caso di cancellazione della chiave primaria è possibile:

- impedire la cancellazione (RESTRICT)
- cancellare tutte le istanze collegate (CASCADE)
- rendere nulle le chiavi esterne (NULLIFY)

## Strutture dati

---

- **Tabella**, è identificata da un nome e normalmente corrisponde ad una entità
- Ha una testata di **attributi** (colonne) che identificano le istanze contenute al suo interno

## Strutture dati

---

- I nomi di colonna sono univoci per tabella
- Una colonna con identico nome in più tabelle dovrebbe avere lo stesso significato in entrambe
- L'univocità di ogni riga è garantito dall'esistenza di una chiave
- Una chiave può essere composta da più attributi

Ing. del Sw: Base dati - 27

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Relazioni

---

- Il collegamento tra tabelle avviene tramite la definizione di **chiavi esterne**
- Una chiave esterna è una **chiave primaria** in una altra tabella

Ing. del Sw: Base dati - 28

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Null

---

- Il null rappresenta la non valorizzazione di un attributo (non vi sono valori significativi per l'attributo)
- Il valore può essere sconosciuto, inesistente perchè legato ad uno stato non ancora verificato per l'istanza o non applicabile (is-a)
- Una chiave primaria non può mai assumere valore null

Ing. del Sw: Base dati - 29

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Viste

---

- Una vista o view è la definizione di una tabella virtuale i cui dati non sono memorizzati ma acquisiti da una o più tabelle
- I dati sono ricavati ogni volta che viene eseguito l'accesso tramite vista

Ing. del Sw: Base dati - 30

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Viste - 2

---

- Consentono di accedere ai dati senza specificare ogni volta i parametri di selezione
- Consentono di restringere l'ambito di consultazione per degli specifici utenti
- Non tutte le view sono aggiornabili

## Normalizzazione dei dati

---

- Una tabella è costituita da una chiave primaria e zero o più campi aggiuntivi
- Ogni tabella deve contenere le informazioni su un solo tipo di entità
- La normalizzazione ha l'obiettivo di minimizzare la ridondanza dei dati e pulire il disegno



## Prima forma normale

---

- La relazione ha una chiave primaria
- Ogni attributo fornisce una informazione sulla chiave
- Ogni campo di una tabella contiene un solo attributo semplice (nella tabella non esistono attributi multivalore)

## Seconda forma normale

---

La relazione è in seconda forma normale se:

- Rispetta la prima forma normale
- Ogni attributo fornisce una informazione sulla intera chiave
- Ogni attributo non chiave dipende esclusivamente dalla chiave primaria della relazione

## Terza forma normale

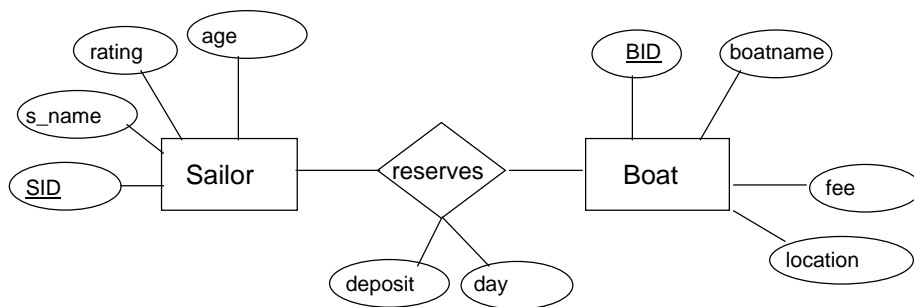
---

La relazione è in terza forma normale se:

- Rispetta la seconda forma normale
- Ogni attributo fornisce una informazione esclusivamente sulla chiave
- Ogni attributo non chiave non deve avere dipendenze funzionali su un altro attributo non chiave

## Esempio della barca a vela

---



## Insiemi di entità

---

- Un'entità è una cosa od un oggetto nel mondo reale (nel contesto del dominio applicativo), ossia corrisponde ad una classe
- Un'entità ha un insieme di proprietà che la identificano univocamente (attributi)
- Viene rappresentata con I rettangolo nei diagrammi E-R



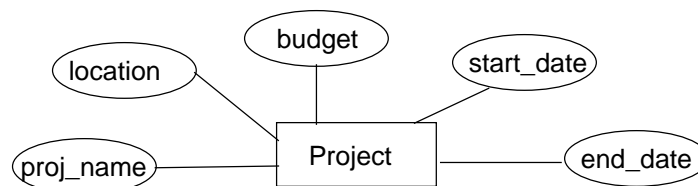
Ing. del Sw: Base dati - 37

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Attributes

---

- Proprietà descrittive delle entità
- Esempio:  $Project = (proj\_name, location, budget, start\_date, end\_date)$
- Dominio: l'insieme dei valori ammessi per ogni attributo
- Rappresentati come ellissi nei diagrammi E-R



Ing. del Sw: Base dati - 38

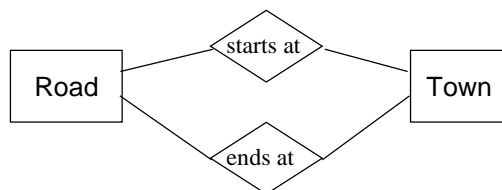
Giulio Destri - © AreaSP for Univ. Parma, 2006

## Tipi di attributi

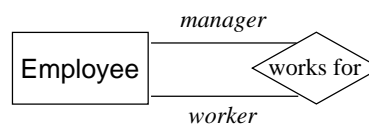
- **Semplice/Atomico** o **composito** (numero di sotto-parti):
    - Semplice: una parte
    - Composito: diviso in sotto-parti
      - Es. Nome = cognome + nome\_battesimo
  - **Singolo-** o **multi-valore** (numero di valori per entità):
    - Singolo-valore: un valore per una entità particolare
    - Multi-valore: un insieme di valori
      - es. Studente ha un numero di corsi
  - attributi **Null** : sconosciuti, non applicabili o mancanti
  - **Derivati**: valori di attributi che possono essere derivati dai valori di altri attributi (non immagazzinati ma calcolati su richiesta)
    - Es. Anni\_assunzione = data\_corrente – data\_assunzione
- Ing. del Sw: Base dati - 39      Giulio Destri - © AreaSP for Univ. Parma, 2006

## Relazioni

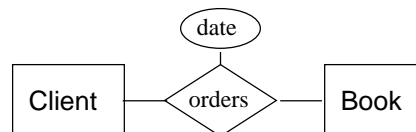
- Può esistere più di una relazione fra le entità.



- Si possono avere relazioni ricorsive ed indicarne i nomi

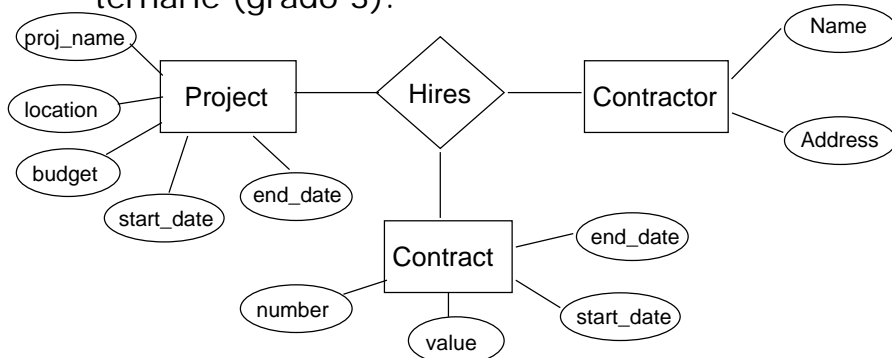


- Anche una relazione può avere attributi descrittivi



## Grado delle relazioni

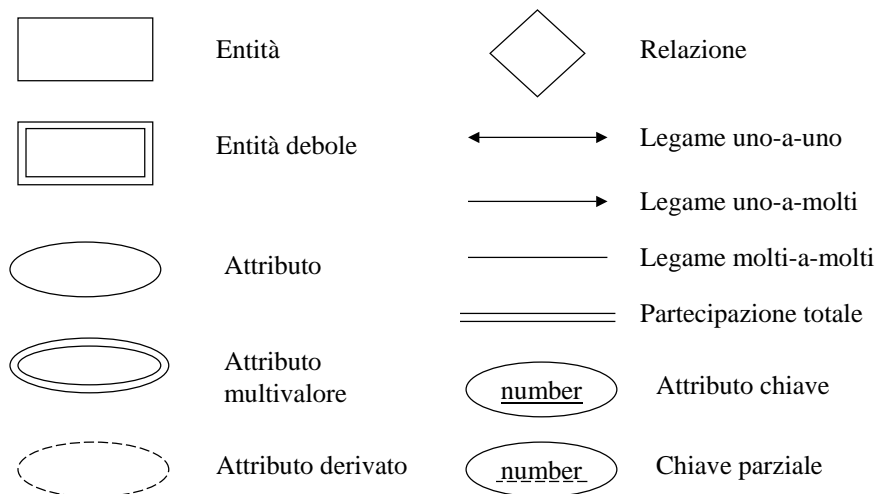
- Si riferisce al numero delle entità che partecipano alla relazione
- La maggioranza delle relazioni sono binarie (grado 2) ma si possono avere anche relazioni ternarie (grado 3).



Ing. del Sw: Base dati - 41

Giulio Destri - © AreaSP for Univ. Parma, 2006

## E-R Diagram Symbols



Ing. del Sw: Base dati - 42

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Caratteristiche Extended E-R

---

### Specializzazione & Generalizzazione

- Specializzazione: un modo per identificare sottogruppi entro un insieme di entità che hanno attributi non condivisi da tutte le entità (top-down)
- Generalizzazione: Più entità vengono sintetizzate entro una basata su caratteristiche comuni a tutti (bottom-up)
- Simbolo E-R :

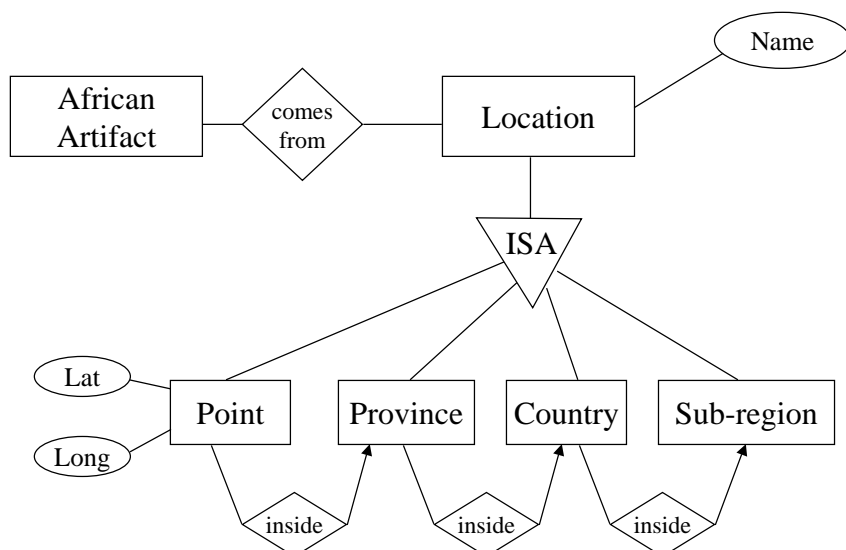


Ing. del Sw: Base dati - 43

Giulio Destri - © AreaSP for Univ. Parma, 2006

### Specializzazione

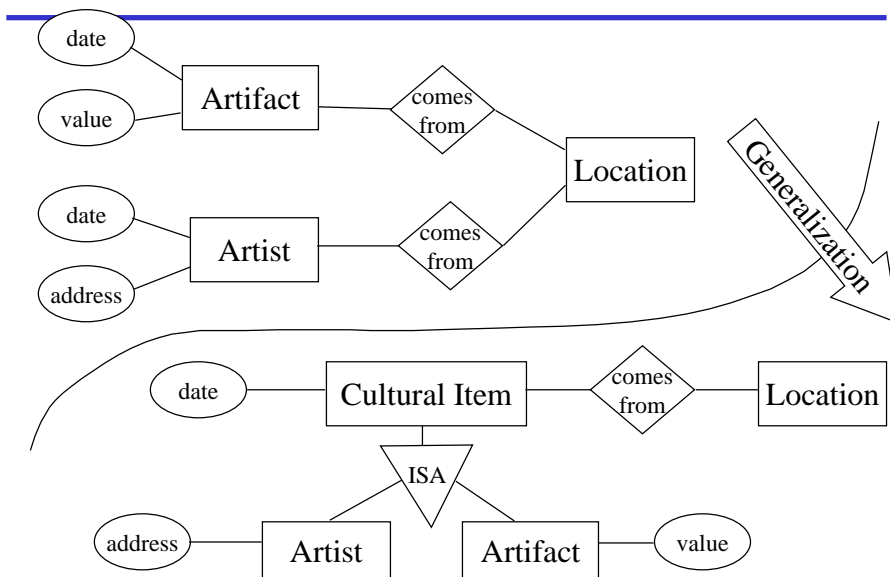
---



Ing. del Sw: Base dati - 44

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Generalizzazione



Ing. del Sw: Base dati - 45

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Utente e Database

- L'utente può accedere al database in due modalità :
  - direttamente tramite utility per l'interrogazione (SQLplus in Oracle, Enterprise Manager in SQL Server)
  - tramite interrogazioni eseguite dai programmi applicativi

Ing. del Sw: Base dati - 46

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Interrogare un Database: le query

---

- Una query è una operazione di accesso al database per la manipolazione dei dati
  - Query di interrogazione
  - Query di inserimento
  - Query di aggiornamento
  - Query di cancellazione

## Le operazioni in un database

---

- Una transazione è definita come
  - un'unità logica di elaborazione,
  - cioè una sequenza di operazioni che hanno un effetto globale sul database



## Le proprietà ACID delle transazioni

---

- **Atomicity (Atomicità)**
- **Consistency (Consistenza)**
- **Isolation (Isolamento)**
- **Durability (Durabilità)**

## Le proprietà ACID delle transazioni

---

- **Atomicity (Atomicità):** tutte le operazioni della sequenza terminano con successo (commit) oppure, se anche una sola di esse fallisce, l'intera transazione viene abortita (abort).

## Le proprietà ACID delle transazioni

---

- **Consistency (Consistenza):**  
una transazione è una trasformazione corretta dello stato del database, vale a dire, al termine di ogni transazione il DB deve trovarsi in uno stato consistente.

## Le proprietà ACID delle transazioni

---

- **Isolation (Isolamento):**  
l'effetto di esecuzioni concorrenti di più transazioni deve essere equivalente ad una esecuzione seriale delle stesse. Quindi, transazioni concorrenti non devono influenzarsi reciprocamente.

## Le proprietà ACID delle transazioni

---

- **Durability (Durabilità):** gli effetti sulla base di dati prodotti da una transazione terminata con successo sono permanenti, cioè non sono compromessi da eventuali malfunzionamenti

## Le proprietà ACID delle transazioni

---

- Lo stato di un database viene fatto evolvere per transazioni
- una transazione parte sempre da uno stato consistente e deve comunque terminare lasciando il DB in uno stato consistente.

## Risultati delle transazioni

---

- **committed transaction:** tutte le operazioni che compongono la transazione sono state eseguite con successo ed il database si trova in un nuovo stato consistente (**commit**);

## Risultati delle transazioni

---

- **aborted transaction:** alcune operazioni della transazione non possono essere portate a termine correttamente, quindi il DB viene riportato nello stato consistente in cui si trovava prima dell'inizio della transazione (**rollback**).

---

## Il linguaggio SQL

Ing. del Sw: Base dati - 57

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Il linguaggio SQL

---

- **Linguaggio standard per l'interrogazione dei database**
- **Una istruzione SQL contiene i dati su cui si vuole agire, le condizioni per le quali si vuole agire sui dati, la modalità di ordinamento dei dati**

Ing. del Sw: Base dati - 58

Giulio Destri - © AreaSP for Univ. Parma, 2006

## Il linguaggio SQL

---

- DML (Data Manipulation Language)  
istruzioni per la manipolazione dei contenuti del database
- DDL ( Data Definition Language)  
istruzioni per la gestione delle strutture dati

## Gestione Tabella

---

```
CREATE TABLE impiegati
( impiegato_id smallint PRIMARY KEY,
  impiegato_telefono varchar(50) NOT NULL
  DEFAULT 'ND',
  eta_min int NOT NULL CHECK (eta_min >= 18),
  lavoro_id smallint REFERENCES lavoro(lavoro_id)
)
DROP TABLE impiegati
ALTER TABLE impiegati ADD livello INT NULL
```

## Tipi di dati

---

INT compresi tra  $2^{31}$  (-2.147.483.648) e  $2^{31} - 1$  (2.147.483.647).

DECIMAL/NUMERIC  $10^{38} - 1$  a  $10^{38} + 1$

FLOAT  $-1,79E + 308$  e  $1,79E + 308$ .

REAL  $-3,40E + 38$  e  $3,40E + 38$ .

DATETIME/DATE

CHAR alfanumerici lunghezza fissa

VARCHAR alfanumerici lunghezza variabile

TIMESTAMP ora di sistema

## Indici

---

- Un indice è un puntatore ai record del database basato su uno o più campi
- Un indice consente la riduzione dei tempi di accesso al database, il campi dell'indice sono ordinati
- CREATE UNIQUE INDEX  
ordini\_dettaglio\_ind ON  
ordini\_dettaglio (ordine\_ID,  
rigaordine\_ID)

## Selezionare dei dati

---

```
SELECT negozio, count(orderid)
FROM ordine
WHERE importo > 5000
GROUP BY negozio
HAVING count(orderid) > 500
ORDER BY negozio
```

## Predicati

---

```
WHERE a = 5
WHERE a <> 5
WHERE a < 5
WHERE a > 5
WHERE a <= 5
WHERE a >= 5
WHERE a BETWEEN 4095 AND 12000
WHERE a IN (4,100,200)
WHERE a IS NULL
WHERE a LIKE 'X%'
```



## Espressioni Numeriche

---

```
SELECT cognome, nome,  
stipendio + premi / 14 as 'importo',  
eta - 65 as 'pensione',  
FROM impiegati  
ORDER BY importo
```

## Funzioni di insieme

---

```
SELECT COUNT(DISTINCT citta) FROM  
impiegati  
SELECT AVG(stipendio) FROM impiegati  
SELECT MAX(stipendio) FROM impiegati  
SELECT MIN(stipendio) FROM impiegati  
SELECT SUM(stipendio) FROM impiegati
```

## Inserire dei dati

---

```
INSERT impiegati (cognome, nome)
VALUES ('rossi','mario')
```

```
INSERT impiegati
SELECT impiegatiid, cognome, nome
FROM impiegatitemp
```

## Aggiornare e cancellare i dati

---

```
UPDATE impiegati
SET stipendio = 2500
WHERE livello = 1
```

```
DELETE impiegati
WHERE livello = 1
```

## Operatori Relazionali

---

```
SELECT cognome, nome, posizione  
FROM impiegati i INNER JOIN posizione p  
WHERE i.posizioneid = p.posizioneid
```

Left Join  
Right Join  
Full Join

## Operatori Relazionali

---

```
SELECT cognome, nome, posizione  
FROM impiegatiItalia  
UNION  
SELECT cognome, nome, posizione  
FROM impiegatiEsteri  
UNION  
SELECT cognome, nome, posizione  
FROM impiegatiEx
```

## Diritti di accesso azioni

---

GRANT CREATE DATABASE,CREATE  
TABLE TO oscar

GRANT INSERT, UPDATE, DELETE ON  
authors TO oscar

REVOKE revoca le autorizzazioni

## Sommario

---

- I database
- I DBMS
- Definizione e progettazione
- Il linguaggio SQL