



# Laboratorio d'Informatica

Corso di Laurea Magistrale in Ingegneria per  
l'Ambiente e il Territorio

A.A. 2021/2022

Docente: Lorenzo Putzu

## Lezione 10

# Basi Di Dati

E' vietata la copia, la rielaborazione, la riproduzione in qualsiasi forma dei contenuti e immagini presenti nelle lezioni. E' inoltre vietata la diffusione, la redistribuzione e la pubblicazione dei contenuti e immagini, incluse le registrazioni delle videolezioni con qualsiasi modalità e mezzo non autorizzate espressamente dall'autore o da Unica

# Argomenti

- Progettazione concettuale
  - il modello *Entità-Relazione*
  - definizione dello schema Entità-Relazione
- Progettazione logica
  - il modello logico *relazionale*
  - definizione dello schema logico relazionale
- Il linguaggio SQL
  - creazione dello schema di una base di dati
  - definizione di interrogazioni su una base di dati

# Sistemi informativi e basi di dati

- Sistemi informativi
  - dati (testuali/numerici), strumenti di archiviazione e gestione, utenti, procedure di accesso, ...
  - esempi: università, aziende, enti pubblici, **sistemi informativi geografici (GIS)**
- Basi di dati (BD)
  - strumenti informatici per la gestione di sistemi informativi

# Basi di dati

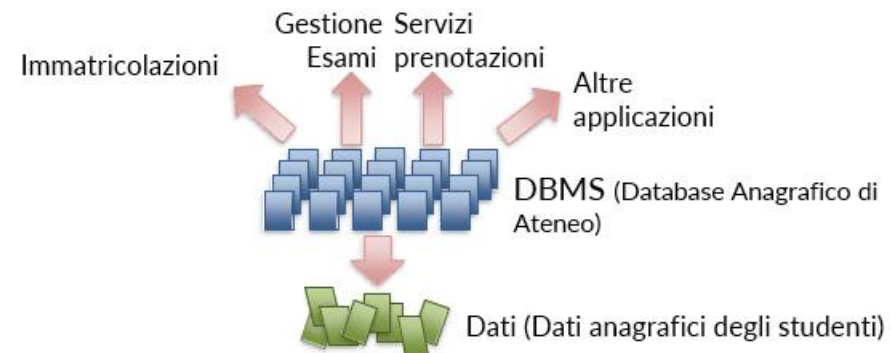
- Una Base di Dati (**Database**) è un insieme organizzato di dati.
- Una base di dati è un insieme di dati che può essere gestito da un sistema **DBMS** (Data Base Management System).
- Un DBMS è un sistema software composto da sequenze di dati interconnessi tra di loro, e da programmi applicativi che consentono di effettuare su tali dati operazioni di:
  - accesso
  - modifica
  - interrogazione (query di ricerca).

# Basi di dati

- Problemi principali:
  - quantità dei dati (risorse *hardware*, efficienza, ...)
  - molti utenti di diverse tipologie
  - procedure di accesso complesse
  - requisiti critici: accesso in tempo reale, accesso concorrente, ecc.
  - sicurezza dei dati (limitazioni all'accesso)
  - ridondanza
  - ripristino da *crash* e guasti

# Basi di dati

- Ogni SI (Sistema Informativo) è suddiviso generalmente in settori o sotto-sistemi, ognuno dei quali svolge varie funzionalità ed esegue vari processi su una base di dati spesso condivisa.
- Possono essere molteplici le applicazioni che utilizzano gli stessi dati o stesse porzioni della base di dati condivisa dal SI.

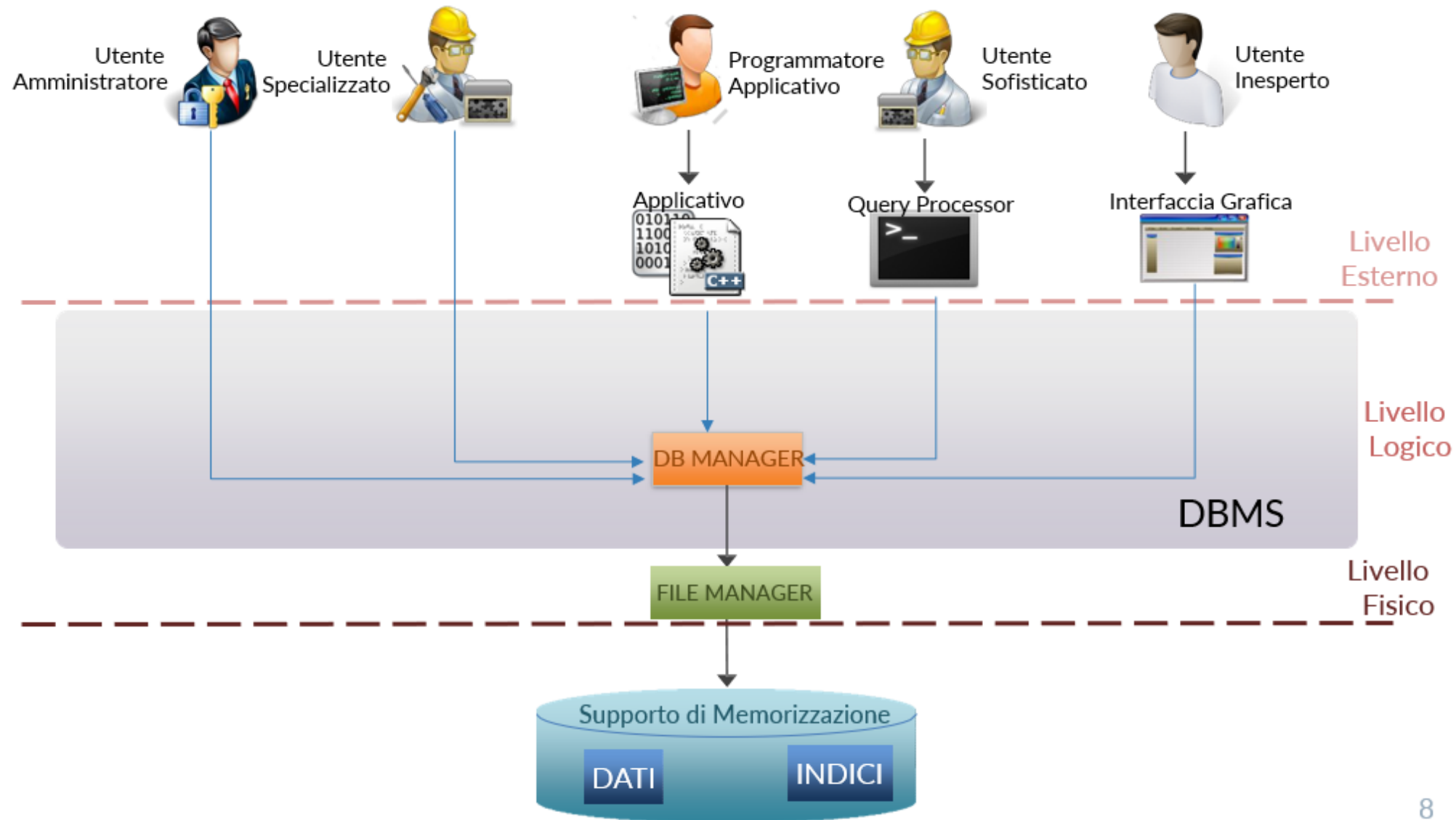


- Se la base di dati non è ben progettata, possono verificarsi problemi di integrità, ridondanza o incoerenza, a cui è necessario provvedere attraverso appositi metodi di riconciliazione dei dati:
  - Integrità: di solito dovuti ad errore umano o incompletezza nell'immissione di dati (referenziati tra di essi) all'interno del database.
  - Ridondanza: informazioni ripetute all'interno del database.
  - Ambiguità: incoerenza; differenti descrizioni di uno stesso dato o informazione non coincidono.

# Basi di dati

- Un DBMS supporta diversi livelli di astrazione sui dati, a differenza di altri sistemi informativi analoghi non informatizzati (elenco telefonico, registro anagrafico cartaceo ecc.).
  - Livello Fisico
  - Livello Logico/Concettuale
  - Livello Esterno
- Un DBMS deve essere in grado di gestire basi di dati di grandi dimensioni (Big Data), persistenti (cioè con un ciclo di vita indipendente dal programma che le utilizza) e condivise da più utenti/sistemi/applicazioni.
- Un DBMS deve inoltre prevedere modalità di controllo differenziato di accesso ai dati, per garantire la privacy e la sicurezza di eventuali dati sensibili, riservati e/o personali.

# Architettura di un DBMS



# Utenti di una base di dati

- **DBA - Database Administrator.** Utente amministratore che ha il controllo completo del sistema.

Le altre tipologie di utenti possono essere classificate in base alla loro esperienza nell'uso delle varie funzionalità del sistema.

- **Utente Inesperto.** Utente finale, non ha conoscenza della struttura dei modelli e/o dei linguaggi del sistema, quindi vi accede ed interagisce solo mediante opportune interfacce grafiche fornite dal DBMS
- **Programmatore Applicativo.** Progettano software al cui interno viene integrato un DBMS per la gestione dei dati
- **Utente Sofisticato.** Interviene direttamente sul DBMS, formulando query oppure agendo sui dati
- **Utente Specializzato.** E' un utente sofisticato che utilizza il sistema per scopi di test (prestazioni del DBMS, controllo di applicazioni ecc.)

# DBMS: vantaggi e limiti

- I vantaggi nell'uso di un DBMS sono molteplici:
  - Accesso ai dati indipendente dalla loro rappresentazione e memorizzazione.
  - Tecniche di accesso ai dati ottimizzate (anche attraverso l'uso di strutture aggiuntive, come gli indici), per migliorare le performance delle interrogazioni.
  - Possibilità di controllo dell'integrità e della completezza dei dati.
  - Possibilità di controllo dell'accesso ai dati e modalità di accesso multi-utente.
- Alcuni limiti:
  - Non permettono visualizzazioni grafiche dei dati o delle strutture dati.
  - Implementano funzionalità matematiche e statistiche elementari di base.

# Linguaggi dei DMBS

- Due funzionalità:
  - *Data Definition Language* (DDL)
  - *Data Manipulation Language* (DML)
- Linguaggio standard (ANSI) dal 1999:  
*Structured Query Language* (SQL)

# Obbiettivi del corso

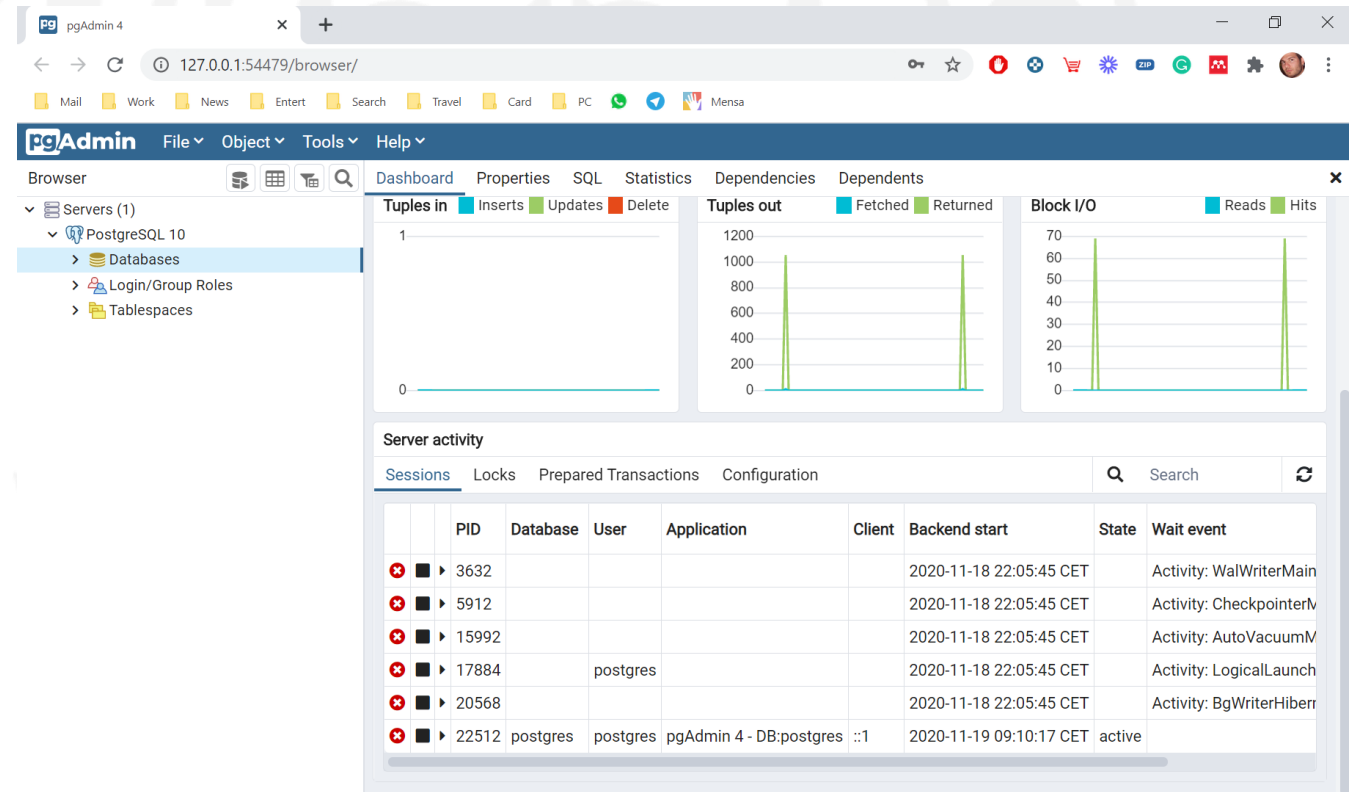
- Punto di vista
  - utente avanzato di una BD (interazione con i progettisti)
  - progettista/amministratore di piccole BD
- Conoscenze di base
  - progettazione concettuale: modello Entità-Relazione
  - progettazione logica: modello relazionale
  - principali comandi del linguaggio SQL per la definizione (DDL) e l'interrogazione (DML) di BD

# DBMS esistenti

- Commerciali
  - IBM
  - Oracle
  - Microsoft
  - ...
- *Open-source*
  - PostgreSQL
  - MySQL
  - ...

# Ambiente di lavoro

- Useremo il sistema open-source PostgreSQL
  - consiglio la versione 10.18, disponibile per ogni sistema operativo  
<https://www.postgresql.org/>
- Durante l'installazione dovrete inserire una *password* della quale dovette tener traccia in quanto verrà richiesta in seguito
- Il programma pgAdmin fornisce l'interfaccia grafica al DBMS

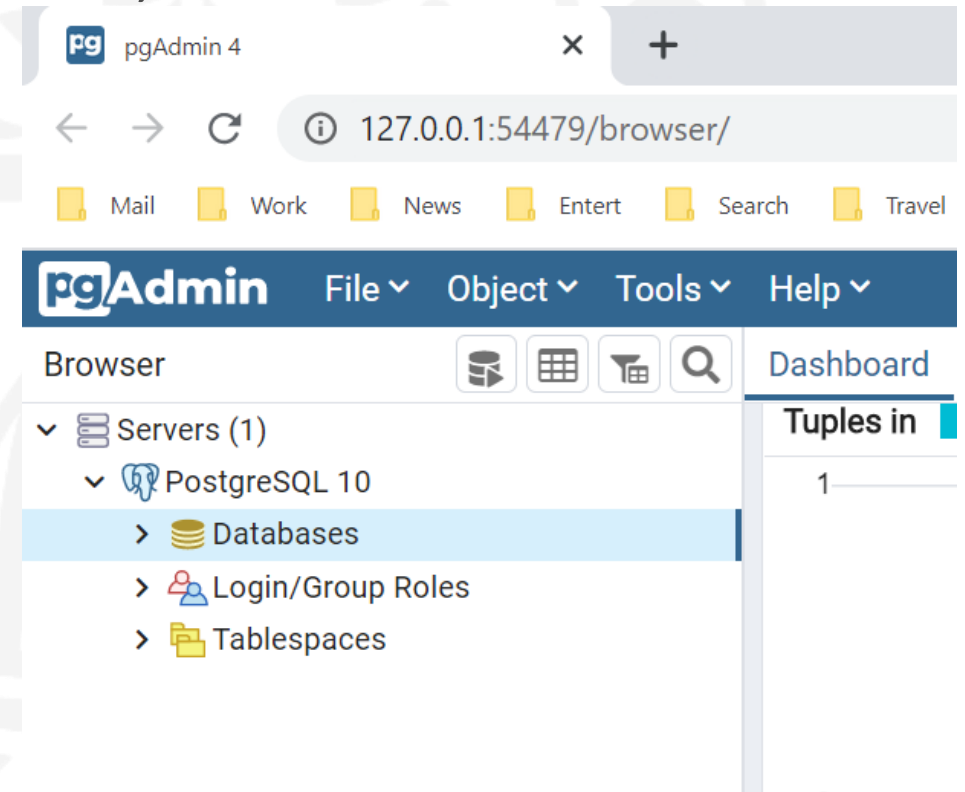


# Ambiente di lavoro

- L'interfaccia è composta da tre pannelli. In particolare, il pannello a sinistra mostra l'organizzazione gerarchica del DBMS PostgreSQL (il contenuto di ogni elemento può essere mostrato o nascosto selezionando il simbolo > alla sua sinistra):
  - un gruppo di *server* predefinito di nome 'Servers'
  - un *server* predefinito nel gruppo 'Servers', di nome 'PostgreSQL'
- Per poter accedere al contenuto di 'PostgreSQL' è necessario inserire la *password* scelta durante l'installazione se sceglierete di memorizzare la *password* non verrà più richiesta in futuro.

# Ambiente di lavoro

- Dopo l'avvio del *server* 'PostgreSQL', al suo interno (nel pannello a sinistra) verrà mostrato l'elemento 'Databases', che contiene le basi di dati da esso gestite.



# Ambiente di lavoro

- Dopo l'avvio del *server* 'PostgreSQL', al suo interno (nel pannello a sinistra) verrà mostrato l'elemento 'Databases', che contiene le basi di dati da esso gestite.
- Espandendo l'elemento 'Databases' verrà mostrata una base di dati predefinita, di nome 'postgres'. Espandendo i suoi elementi si potrà vedere che una base di dati in PostgreSQL è composta da un insieme di schemi ('Schemas'), di cui uno predefinito di nome 'public', che può contenere la definizione di un insieme di tabelle ('Tables'). Dopo l'installazione, la base di dati 'postgres' è vuota (non contiene nessuna tabella).

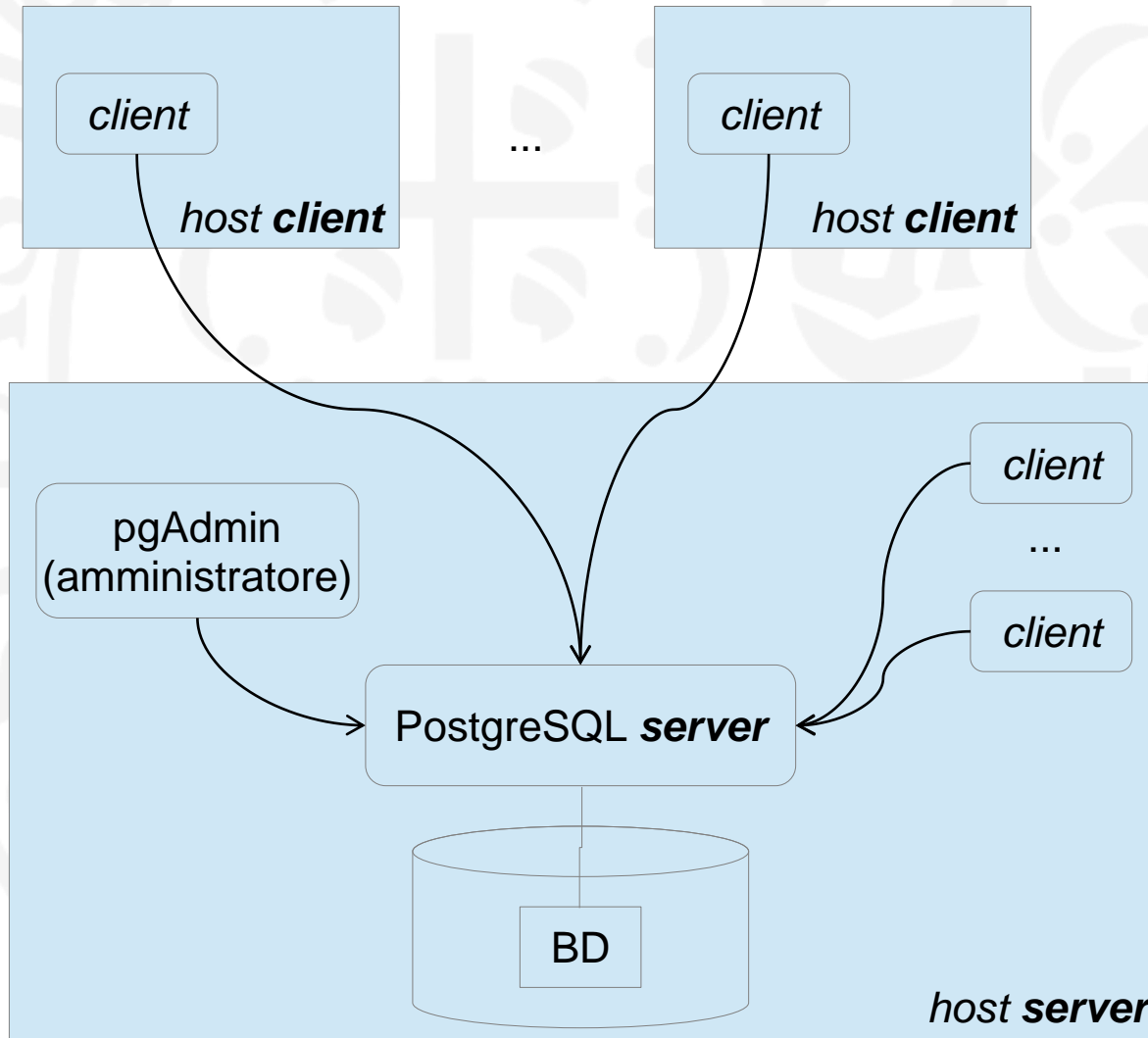
# Ambiente di lavoro

- Per creare una nuova base di dati, fare *click* con il tasto destro del mouse sull'elemento 'Databases' e scegliere la voce 'Create' dal menu a tendina e poi 'Database...'.  
• Nella nuova finestra che compare sarà sufficiente inserire il nome del database che si vuole creare.
- Anche questo nuovo database comparirà nell'elenco dei database nel pannello a sinistra

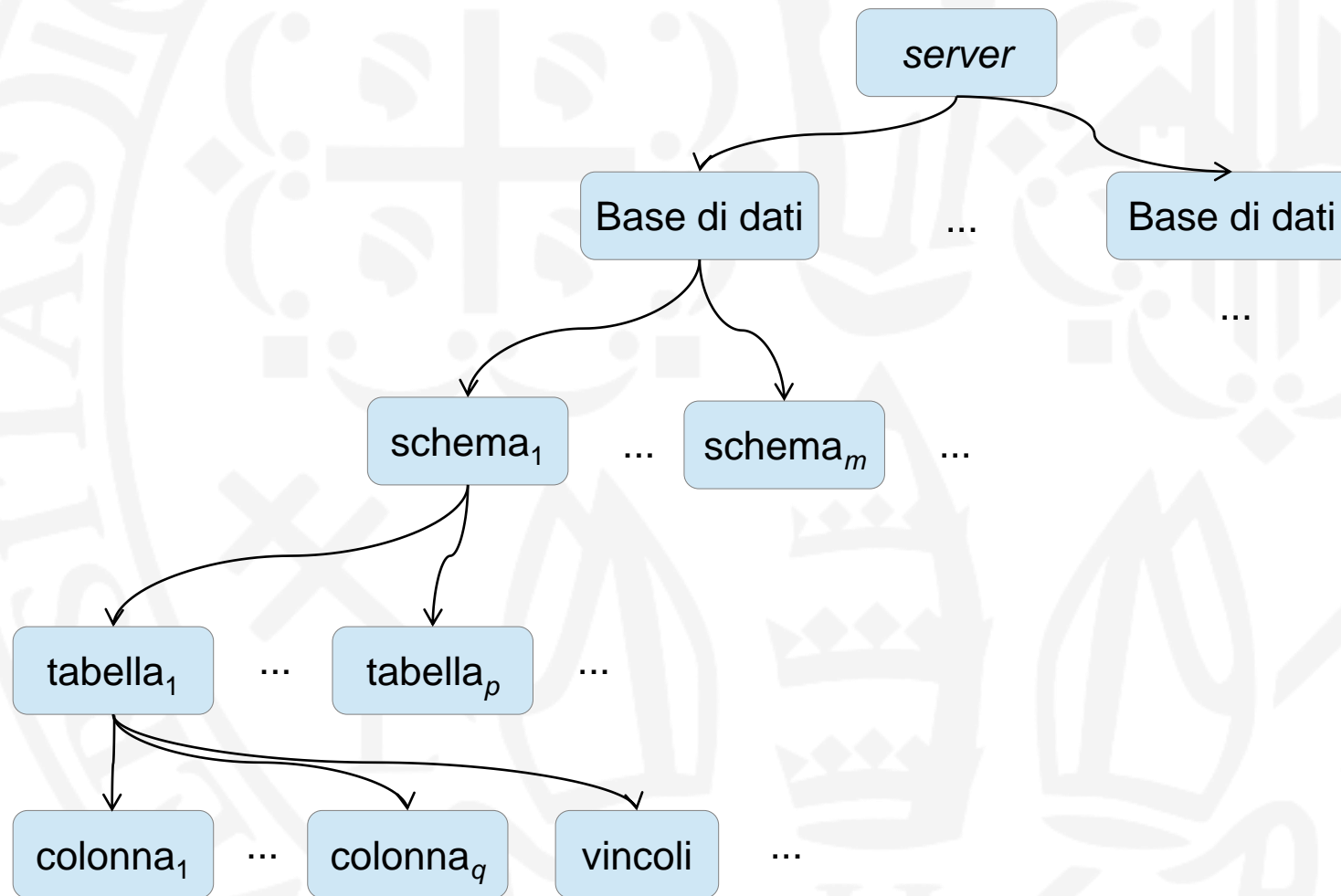
# PostgreSQL

- Il nome:
  - origine: University **I**nteractive **G**raphics **R**etrieval **S**ystem (**INGRES**)
  - University of California, Berkeley, 1982: sviluppo di un DBMS *open-source* "post-Ingres": **Postgres**
  - Introduzione del supporto per SQL: **PostgreSQL**
- *Object-relational* DBMS
- Architettura *client-server*
- Componenti principali:
  - *server* (gestione DB, utenti)
  - strumento di amministrazione: pgAdmin

# PostgreSQL: architettura *client / server*



# Architettura di un DB PostgreSQL





# IL MODELLO LOGICO RELAZIONALE

# Esempio

- L'azienda XYZ è organizzata in dipartimenti: ognuno ha un nome e un codice numerico univoci, una o più sedi (ognuna in una diversa città), ed è diretto da uno dei dipendenti a partire da una certa data
- Ogni dipartimento controlla un insieme di progetti: ogni progetto ha un nome e un codice numerico univoci e viene gestito da una delle sedi di un singolo dipartimento
- Ogni impiegato ha nome, cognome, sesso, data di nascita, CF, indirizzo (città, via, numero), stipendio annuale, supervisore (un altro impiegato), e un dipartimento di appartenenza; può lavorare su uno o più progetti (anche gestiti da altri dipartimenti), a ciascuno dei quali dedica un certo numero di ore settimanali
- Per scopi assicurativi ogni impiegato può avere familiari a carico, dei quali si memorizza nome e cognome, sesso, data di nascita, e rapporto di parentela con l'impiegato (si assume che non possano esistere due familiari omonimi di uno stesso impiegato, con la stessa data di nascita)

# Esempio

- Il corso di laurea in ABC eroga diversi insegnamenti che hanno un nome (es.: Analisi Matematica 1), un codice univoco di cinque caratteri assegnato dalla segreteria studenti, e un numero di crediti
- Al corso di laurea possono essere iscritti diversi studenti. Per ogni iscritto si vogliono memorizzare nome, cognome, codice fiscale, data di nascita e numero di matricola (una sequenza di cinque caratteri che identifica univocamente ogni studente)
- Ogni studente ha un proprio piano di studi
- Per ogni esame superato da ogni studente si vogliono conoscere il voto (un intero tra 18 e 30, oppure 30 e lode) e la data; un esame superato non può essere ripetuto

# Il modello logico relazionale

- Ideato negli anni '70 è tuttora il **modello** più utilizzato per rappresentare i database
- Fondamenti: teoria degli insiemi.
- Componenti:
  - **base di dati relazionale**: insieme di tabelle (*relazioni*) identificate da un nome univoco
  - **tabella**: insieme di **righe** (*record*, "*tuple*"), suddivise in un insieme di **colonne** (*campi*, *attributi*) identificate da un nome univoco; ogni riga contiene dati su un elemento di un dato insieme; l'ordine delle righe e delle colonne non è definito; non possono esistere righe identiche
  - **valori** di ogni colonna: appartengono a un **dominio**

# Esempio

## Studenti

<i>Matricola</i>	<i>Codice Fiscale</i>	<i>Cognome</i>	<i>Nome</i>	<i>Data di nascita</i>
00001	RSSMRA...	Rossi	Maria	1/1/1990
00002	VRDFRA...	Verdi	Francesca	3/2/1991
00003	NRIGVI...	Neri	Giovanni	5/7/1990
00004	BNCMRA...	Bianchi	Maria	10/11/1990
00005	SMTJOH...	Smith	John	12/12/1991

## Insegnamenti

<i>Nome</i>	<i>Codice</i>	<i>Crediti</i>
Analisi I	AMI01	9
Geometria	GEO01	6
Fisica I	FISI1	6
Chimica	CHI01	6
Informatica	INF01	6

## Esami

<i>Studente</i>	<i>Esame</i>	<i>Data</i>	<i>Voto</i>	<i>Lode</i>
00003	GEO01	10/9/2013	28	falso
00001	AMI01	15/9/2013	24	falso
00003	AMI01	30/09/2013	27	falso
00002	GEO01	28/9/2013	30	vero
00004	CHI01	19/9/2013	27	falso