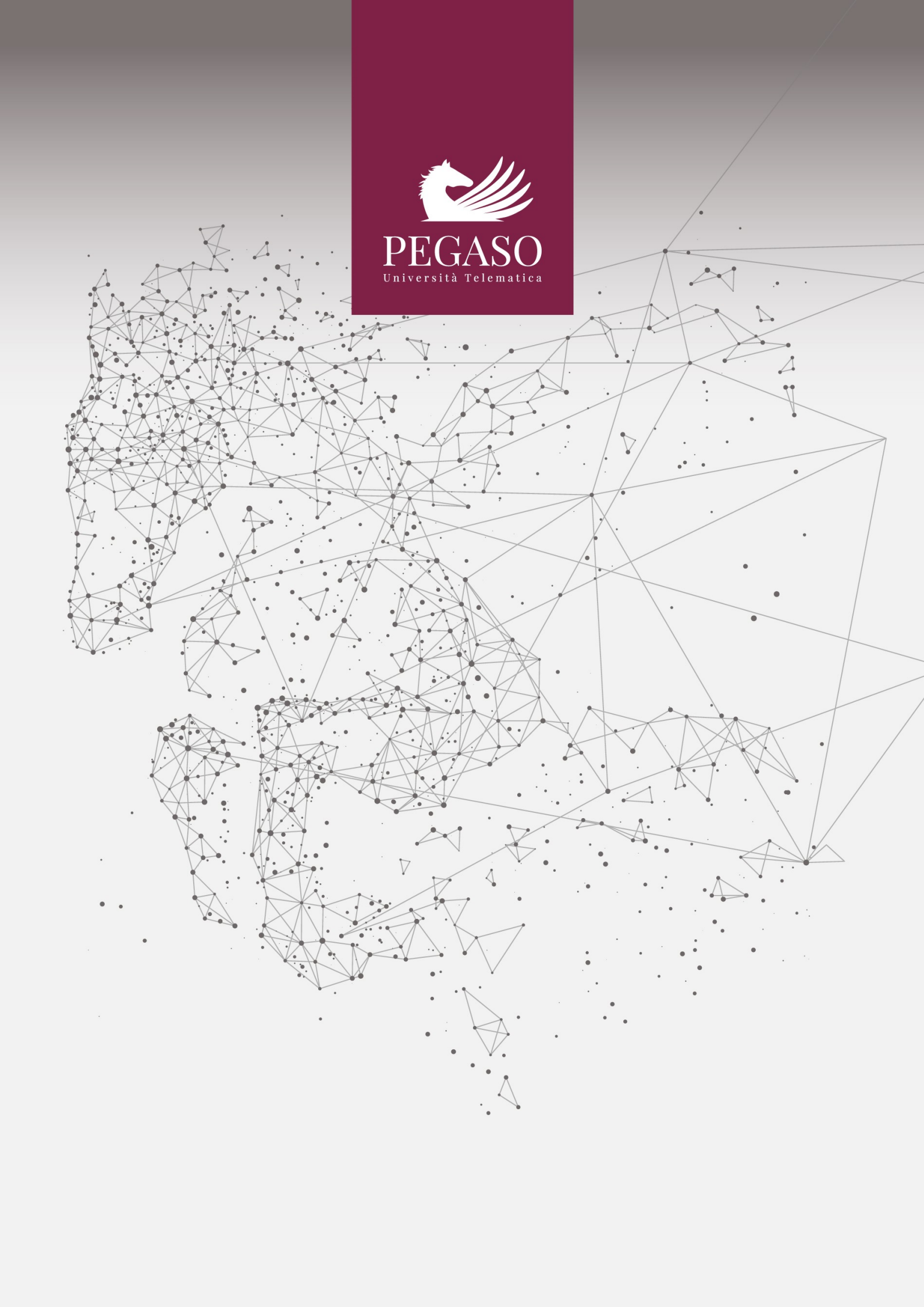
PEGASO
Università Telematica

Indice

1. Introduzione

In questa unità didattica vengono trattati i linguaggi con i quali è possibile gestire i dati memorizzati all'interno di una base di dati. Il linguaggio oramai divenuto uno standard è lo *Structured Query Language* (SQL) attraverso il quale si può interagire in forma interattiva con i dati. SQL non è un linguaggio di programmazione ma un linguaggio di definizione degli schemi dei dati e di manipolazione dei dati.

2. Il linguaggio SQL

Structured Query Language (SQL) è un linguaggio standardizzato che viene utilizzato per gestire i database relazionali ed eseguire operazioni sui dati al loro interno.

Inizialmente creato negli anni '70, SQL oggigiorno viene regolarmente utilizzato non solo dagli amministratori di database, ma anche dagli sviluppatori che scrivono script di integrazione dati e analisti di dati.

Gli usi di SQL includono:

- La modifica della tabella del database e delle strutture di directory;
- L'aggiunta, l'inserimento e la cancellazione di record, ovvero di righe di dati dalle tabelle;
- Il recupero di sottoinsiemi di informazioni da un database per l'elaborazione delle transazioni e le applicazioni di analisi dati.

Le interrogazioni e altre operazioni SQL assumono la forma di comandi: i comandi SQL comunemente utilizzati comprendono: selezionare, aggiungere, inserire, aggiornare, eliminare, creare, modificare e troncare dati.

SQL è diventato il linguaggio di programmazione standard per i database relazionali dopo la loro comparsa alla fine degli anni '70 e all'inizio degli anni '80. Conosciuto anche come database SQL, i sistemi relazionali includono set di tabelle contenenti dati in righe e colonne. Una colonna per una categoria di dati, ad esempio un docente o un'aula, mentre ogni riga contiene un valore di dati per la colonna intersecante, come l'esempio delle due tabelle di Figura 1.

Corsi

Corso	Docente	Aula
Basi di dati	Rossi	DS3
Sistemi	Neri	N3
Reti	Bruni	N3
Controlli	Bruni	G

Aule

Nome	Edificio	Piano
DS1	OMI	Terra
N3	OMI	Terra
G	Pincherle	Primo

Figura 1: *Esempio di Tabelle.*

Uno standard ufficiale SQL è stato adottato dall'American National Standards Institute (ANSI) nel 1986 e poi dall'Organizzazione internazionale per la standardizzazione, noto come ISO, nel 1987. Più di una mezza dozzina di aggiornamenti congiunti allo standard sono stati rilasciati dai due standard di sviluppo dagli organismi da allora.

Entrambi i sistemi di gestione dei database relazionali proprietari e aperti basati su SQL sono disponibili per l'uso da parte delle organizzazioni. Tra i DBMS più utilizzati abbiamo: Microsoft SQL Server, Oracle Database, IBM DB2, SAP HANA, SAP Adaptive Server, MySQL (ora di proprietà di Oracle) e PostgreSQL. Tuttavia, molti di questi prodotti di database supportano SQL con estensioni proprietarie al linguaggio standard per la programmazione procedurale insieme ad altre funzioni. Ad esempio, Microsoft offre una serie di estensioni chiamate Transact-SQL (T-SQL), mentre la versione estesa di Oracle dello standard è PL/SQL. Di conseguenza, le diverse varianti di SQL offerte dai fornitori non sono completamente compatibili l'una con l'altra.

In conclusione, l'SQL è un linguaggio specializzato per l'aggiornamento, l'eliminazione e la richiesta di informazioni dai database. Una varietà di prodotti consolidati sul mercato supportano SQL, compresi i prodotti Oracle e Microsoft SQL Server. È ampiamente utilizzato sia nell'industria che nel mondo accademico, spesso per database enormi e complessi. Nelle prossime unità didattiche questo linguaggio sarà trattato in modo approfondito.

Figura 2: *Esempio di istruzione DML.*

3. DDL e DML

I comandi SQL sono suddivisi in diversi tipi, tra cui DML (Data Manipulation Language) e DDL (Data Definition Language), controlli delle transazioni e misure di sicurezza. Il vocabolario DML viene utilizzato per recuperare e manipolare i dati, mentre le istruzioni DDL servono a definire e modificare le strutture del database.

I controlli sulle transazioni aiutano a gestirne l'elaborazione, garantendo che le transazioni siano completate o annullate se si verificano errori o problemi. Le istruzioni di sicurezza vengono utilizzate per controllare l'accesso al database e per creare ruoli utente e autorizzazioni.

Da quanto detto quindi, DDL e DML insieme formano un linguaggio per il database, con la differenza che DDL viene utilizzato per specificare la struttura e lo schema del database mentre DML viene utilizzato per accedere, modificare o recuperare i dati.

3.1 DDL (Data Definition Language)

Questo linguaggio di definizione dei dati definisce la struttura e lo schema del database ed anche proprietà aggiuntive dei dati definiti nel database, come il dominio degli attributi.

Il linguaggio di definizione dei dati fornisce anche la possibilità di specificare alcuni vincoli che manterrebbero la coerenza dei dati.

Anticipiamo qui alcuni dei comandi più importanti di DDL, anche se alcuni concetti base come la struttura di una tabella, ma in generale, i database relazionali, verranno affrontati in dettaglio nelle prossime unità didattiche. Quando si parla di tabella in questo conteso, il lettore può, per adesso, immaginarla come una matrice organizzata in righe (record o tupla) e colonne (attributi). Come vedremo, ciascuna riga corrisponde ad una istanza del dominio di conoscenza del database.

- CREATE è un comando utilizzato per creare un nuovo database o tabella;
- ALTER viene utilizzato per modificare il contenuto nella tabella;
- DROP viene utilizzato per eliminare alcuni contenuti nel database o nella tabella;
- TRUNCATE viene utilizzato per eliminare tutto il contenuto dalla tabella;
- RENAME è usato per rinominare il contenuto nel database.

I suddetti cinque comandi rappresentano le istruzioni per lavorare sullo schema di un database di tipo relazionale. In Figura 3 è illustrato un esempio di utilizzo dell'istruzione CREATE TABLE per la creazione di una tabella.

```
CREATE TABLE orario (
  insegnamento      CHAR(20) ,
  docente                CHAR(20) ,
  aula              CHAR(4)  ,
  ora               CHAR(5)  )
```

Figura 3: *Esempio di comando DDL: creazione di una tabella.*

3.2 DML (Data Manipulation Language)

Uno schema (tabella) creato attraverso un comando DDL, viene successivamente popolato o riempito utilizzando proprio il linguaggio DDL. DDL riempie le righe della tabella e ogni riga è denominata Tupla. Usando DML, si possono inserire, modificare, eliminare e recuperare le informazioni dalla tabella. In Figura 4 un esempio di istruzione DML per la costruzione di un insieme di dati estratti dal database.

```
SELECT Corso, Aula, Piano
FROM Aule, Corsi
WHERE Nome = Aula
    AND Piano = 'Terra'
```

Figura 4: *Esempio di istruzione DML.*

I comandi utilizzati più frequentemente in DML sono i seguenti:

- SELECT usato per recuperare i dati da una tabella;
- INSERT utilizzato per inserire i dati in una tabella;
- UPDATE utilizzato per aggiornare i dati di una tabella;
- DELETE utilizzato per cancellare i dati di una tabella.

In conclusione, per la costruzione e la gestione di un database, sono necessari sia DDL che DML.

4. Personaggi e interpreti

Oggi, la maggior parte delle organizzazioni dipende addirittura da un DBMS a causa di accumulo di dati strategici, come quelli collezionati attraverso le interazioni dei clienti online. I dati potrebbero essere utilizzati per l'analisi dei clienti, i servizi, il business, i vantaggi e gli svantaggi e così via. In breve, l'utilità di un DBMS varia in base al dominio di business delle organizzazioni e necessitano di servizi gestiti a livello centrale e distribuiti su LAN, reti, e con sistemi di ordini basati su Internet, servizi e così via.

4.1 DBA (Data Base Administrator)

L'amministratore del database svolge un ruolo chiave nella gestione dei dati aziendali, risorse estremamente importanti, attraverso i dipartimenti / le zone di un'organizzazione. Il principale compito di lavoro dell'Amministratore di database è quello di garantire la creazione, la disponibilità, l'archiviazione, la protezione dalla corruzione o la perdita dei dati, facilitare il facile recupero di una parte o del tutto come richiesto dagli utenti finali. Analizziamo brevemente i ruoli principali di un DBA. Questa figura, supporta i vari utenti a CREARE, GESTIRE, RECUPERARE, AGGIORNARE e memorizzare le informazioni. Ad esempio: una piccola azienda, un settore industriale o un'impresa dispone di un'enorme quantità di dati relativi a venditori, clienti, team interno, core business e così via. La gestione manuale di questa mole enorme di dati in genere risulta davvero onerosa. Con un DBMS, è possibile per i vari utenti recuperare i dati su richiesta con l'aiuto delle applicazioni e dell'interfaccia fornite dal DBMS. Un DBA partecipa attivamente alla pianificazione dell'installazione di nuovi sistemi e applicazioni a livello di organizzazione. Implementa il piano di lavoro per il Dipartimento, incontra lo staff per identificare eventuali problemi, prendere misure e risolverlo. Monitora l'efficienza e l'efficacia di tutte le risorse del database e, quindi, mantiene il flusso di lavoro ininterrotto grazie alla piattaforma tecnologica che ospita il DBMS. Inoltre il DBA garantisce il massimo servizio attraverso l'identificazione di opportunità di miglioramento e nuove best practices.

Revisiona e valuta il software, l'hardware, la fornitura del servizio e gli aggiornamenti come e quando richiesto.

4.2 Vantaggi di un DBMS

Il sistema di gestione del database presenta almeno i seguenti vantaggi:

1. Controllo della ridondanza: nel file system, ogni applicazione ha i propri file privati, che non possono essere condivisi tra più applicazioni. Ciò può spesso portare a una notevole ridondanza nei dati memorizzati, il che si traduce in uno spreco di spazio di archiviazione. Avendo un database centralizzato, la maggior parte di questo può essere evitato, anche se non risulta possibile eliminare tutta la ridondanza. A volte ci sono buone ragioni commerciali e tecniche per mantenere più copie degli stessi dati. In un sistema di database, tuttavia, è possibile controllare questa ridondanza.
2. Controllo dei vincoli di integrità: l'integrità dei dati significa che i dati nel database sono sempre accurati, in modo tale che informazioni errate non possano essere archiviate nel database. Al fine di mantenere l'integrità dei dati, alcuni vincoli di integrità vengono applicati al database. Un DBMS dovrebbe fornire funzionalità per la definizione e l'applicazione dei vincoli.
3. Controllo della coerenza dei dati: quando gli stessi dati vengono duplicati e le modifiche vengono apportate in un sito, che non viene propagato adm un altro sito correlato, si genera incoerenza e le due voci relative agli stessi dati non saranno sincronizzate. In tali momenti i dati sono considerati incoerenti. Quindi, se viene rimossa la ridondanza, vengono rimosse anche le possibilità di avere dati incoerenti.
4. Un database incoerente è in grado di fornire informazioni errate o in conflitto. Quindi non ci dovrebbero essere incoerenze nel database. Si può dimostrare chiaramente che l'inconsistenza può essere evitata nel sistema centralizzato molto bene rispetto al file system.
5. Possiamo dire che la ridondanza dei dati influisce notevolmente sulla coerenza dei dati. Se la ridondanza è bassa, è facile implementare la coerenza dei dati. Pertanto, il sistema DBMS può evitare incoerenza in larga misura.
6. Condivisione: i dati possono essere condivisi da più applicazioni in un DBMS centralizzato rispetto al file system, con la possibilità di sviluppare applicazioni per operare con gli stessi dati memorizzati. Le applicazioni possono essere quindi sviluppate senza dover creare nuovi file memorizzati.

Bibliografia

- Curtin, P., D., Foley, K., Sen, K., Morin, C. (2016). Informatica di base. McGraw-Hill Education.
- Mezzalama, M. and Piccolo, E. (2010). Capire l'informatica. CittàStudi Edizioni.
- Atzeni, P., Ceri, S., Fraternali. P., Paraboschi, S., Torlone, R. (2018). Basi di Dati. McGraw-Hill Education.
- Batini, C., Lenzerini, M. (1988). Basi di Dati. In Cioffi, G. and Falzone, V. (Eds). Calderini. Seconda Edizione.
- ANSI/X3/SPARC Study Group on Data Base Management Systems: (1975), Interim Report. FDT, ACM SIGMOD bulletin. Volume 7, No. 2.