Riassunto basi di dati

Dimix

2025-03-13

Indice

Lezione 1 – Introduzione . . . 2
- **Definizione e contesto** . . . 2
- **Considerazioni generali sull'uso delle basi di dati** . . . 2
- **I sistemi di gestione delle basi di dati** . . . 2
- **Architettura di un DBMS** . . . 3
- **Le categorie di linguaggi nei DBMS** . . . 3
- **Utenze di un DBMS** . . . 3

Lezione 2 – Il modello relazionale . . . 4
- **Tabella e domini** . . . 4
- **Vincoli di integrità** . . . 5
- **L'integrità referenziale** . . . 6
- **Le forme di relazione** . . . 6

Lezione 4 – L'algebra relazionale . . . 7
- **Operatori dell'algebra relazione** . . . 7
- **Operatori insiemistici** . . . 7
- **Operatori unari** . . . 7
- **Operatori binari** . . . 7
- **Operatori Join** . . . 7
- **Divisione** . . . 8
- **Equivalenza di espressioni algebriche** . . . 8

Lezione 5 – SQL . . . 9
- **Introduzione** . . . 9
- **Interrogazioni** . . . 10

Lezione 1 – Introduzione

Definizione e contesto

Una **base di dati** è una collezione di dati correlati. Per **dato** intendiamo la registrazione simbolica di un elemento della realtà.

Una base di dati possiede alcune caratteristiche distintive:

- Rappresenta un aspetto limitato della realtà, riferito a un sottoinsieme dei dati disponibili denominato **universo del discorso**.
- Costituisce un insieme di dati logicamente coerenti e dotati di un significato intrinseco.
- È progettata, costruita e popolata per uno scopo specifico, rivolto a un gruppo di utenti determinato e ad alcune specifiche applicazioni.

In una base di dati i dati sono correlati per fornire un'interpretazione e trasformarli in **informazione**.

Esempi:

- Dati non correlati: `Roma, 41.89, 12.48`
- Dati correlati: `Roma → Posizione: Latitudine 41.89, Longitudine 12.48`

Esistono tre tipologie di dati

1. **Dati non strutturati**: tipicamente disponibili in formato testuale (memorizzati su file o sistemi di gestione documentale)
2. **Dati semi-strutturati**: tipicamente disponibili in formati come XML, HTML, JSON, RDF, presentano una informazione struttura ma con pochi vincoli di formato e di coerenza dei dati, insieme ad una frequente ridondanza.
3. **Dati strutturati**: tipicamente in formato tabellare, memorizzati in DBMS relazionali o fogli di calcolo come Excel, la loro struttura è rigida, definita da un insieme di campi che si applicano a tutti gli oggetti descritti ma non è detto che siano rispettati vincoli di formato, coerenza dei dati e che non vi sia ridondanza.

Considerazioni generali sull'uso delle basi di dati

Le basi di dati in confronto alla gestione applicativa

Ogni applicazione possiede un sistema di gestione dei dati, tipicamente in memoria e su file andando a definire di volta in volta le strutture dati necessarie e l'organizzazione dei file, utilizzare invece una base di dati per la gestione dei dati in ausilio delle applicazioni è una scelta che implica alcune caratteristiche specifiche nel modo in cui i dati vengono trattati rispetto alle applicazioni.

Le caratteristiche delle basi di dati

- **Autodescrizione**: Una base di dati contiene sia i dati che una descrizione completa della sua struttura e dei suoi vincoli, attraverso un catalogo.
- **Indipendenza tra programmi e dati**: La struttura di memorizzazione dei dati è astratta e coerente al modello dei dati usato dalla base di dati, in tutti i casi essa è indipendente dalle applicazioni
- **Viste multiple dei dati**: Una base di dati può fornire un accesso a sottoinsiemi e a forme di organizzazione diversi dei dati a diversi utenti e applicazioni per mezzo di viste virtuali
- **Condivisione e controllo della concorrenza**: una base di dati garantisce che utenti e applicazioni diverse possano gestire in modo concorrente gli stessi dati, garantendo al tempo stesso la loro coerenza e correttezza per metto di transazioni.

I sistemi di gestione delle basi di dati

Modelli dei dati

L'organizzare i dati in una base di dati richiede la definizione di un modello logico che detti le regole di organizzazione dei dati stessi.

Il modello dei dati è un insieme di costrutti, formalismi di rappresentazione e operazioni di base necessari alla realizzazione della descrizione astratta e indipendente dalla memorizzazione fisica dei dati, ai vincoli che ne garantiscono la coerenza e correttezza e alla loro gestione.

Il modello relazione è il modello dei dati più diffuso, esso verrà trattato all'interno del nostro corso, esisto anche altri modelli come possono essere quelli *reticolari*, *gerarchici* e ad *oggetti*.

Le nozioni di schema e istanza

- **Schema** (Intensione): è un modello logico che definisce la struttura e i vincoli dei dati da rispettare (fisso nel tempo).
- **Istanza** (Estensione): insieme dei dati organizzati rispettando lo schema (varia nel tempo).

  L'istanza della base di dati in un particolare istante di tempo è detta **stato della base di dati**.

Architettura di un DBMS

Un **DBMS (Database Management System)** è un insieme di programmi che consente la creazione e manutenzione di basi di dati, in particolare consente di definire, costruire, manipolare e condividere basi di dati per vari utenti e applicazioni.

> **Obiettivo del DBMS** è gestire i dati in modo efficiente evitando duplicazioni, errori, e garantendo integrità e sicurezza.

Architettura a tre livelli

L'architettura concettuale di una base di dati è caratterizzata da tre livelli: 1. **Livello interno** (fisico): memorizzazione fisica dei dati. 2. **Livello logico**: rappresentazione logica della base di dati. 3. **Livello esterno**: viste personalizzate per gli utenti.

Le categorie di linguaggi nei DBMS

- **DDL (Data Definition Language)**: istruzioni per la creazione e manutenzione dello schema.
- **DML (Data Manipulation Language)**: istruzioni per la manutenzione e interrogazione dei dati.
- **DQL (Data Query Language)**: istruzioni per l'interrogazione dei dati di una base di dati.
- **DCL (Data Control Language)**: istruzioni per gestire permessi e i controlli di accesso alla base di dati.

Utenze di un DBMS

in una base di dati esistono due macrocategorie di utenti: - **DBA User**: Utenti amministratori dotati di privilegi massimi rispetto ad una o più basi di dati, possono essere in possesso delle funzionalità DDL, DML, DQL, DCL, gestire i privilegi e la sicurezza, gestiscono l'ottimizzazione delle basi di dati e gestiscono lo spazio di memorizzazione fisica. - **DB User**: utenti con privilegi specifici e limitati su una o più basi di dati, essi sono suddivisi in: - Developer user - Occasional user - Casual user

I DBMS adottano un'architettura di comunicazione client/server, l'accesso può avvenire da postazioni locali tramite pipe del sistema operativo o remote tramite socket TCP/IP.

Gli utenti del DMBS utilizzano un software client per comunicare con il server, questo software può essere basato su: - Riga di comando - Interfaccia grafica - Interfaccia web

Lezione 2 – Il modello relazionale

Il **modello relazionale** fu proposto da Edgar Codd nel 1970 e introdotto nei DBMS nel 1981.

- Basato sulla nozione di **relazione matematica**, intesa come sottoinsieme del prodotto cartesiano tra insiemi di dati, detti domini.
- Definisce un insieme di vincoli sui dati.
- Associato al linguaggio dell'**algebra relazionale** per eseguire interrogazioni.

Tabella e domini

La rappresentazione più intuitiva di una relazione è la tabella, una base di dati relazionale è rappresentata da una collezione di tabelle.

Ogni tabella ha un nome unico nella base di dati, con le seguenti proprietà: - Ogni **riga** rappresenta un record. - Ogni **colonna** ha un nome distinto (**attributo** (A_k)), associato a un insieme di valori detto **dominio** (D_k).

Per dominio viene intesa intesa una collezione di valori atomici, in termini pratici, i domini a partire dai quali sono costruite le relazioni nei DBMS sono definiti a partire da tipi di dati, come ad esempio stringhe, interi e date.

La tabella

Dati n domini (D1, D2 ,..., Dn), ogni riga di una tabella è una ennupla ordinata di valori (d1, ..., dn) con dk appartenente al dominio Dk del corrispondente attributo Ak Una tabella contiene un sottoinsieme di tutte le righe possibili, cioè un sottoinsieme del prodotto cartesiano D1 x D2 x … x Dn.

La relazione matematica

Siano D1, D2 ,..., Dn n insieme di valori anche non distinti, il prodotto cartesiano D1 x D2 x … x Dn. è definito cosi: D1 x D2 x … x Dn = {(d1, d2, ..., dn) tale che d1, d2, ..., dn appartengono a D1 , D2, ..., Dn} Una relazione matematica R su D1, D2 ,..., Dn è cosi definita: R sottoinsieme di D1 x D2 x … x Dn D1, D2 ,..., Dn sono domini di R, una relazione su n domini è detta avere grado n Il numero di ennuple di una relazione è detto cardinalità della relazione, nelle applicazioni reali la cardinalità di una relazione è sempre finita

Proprietà delle relazioni

La relazione con attributi, nella quale un attributo che descrive il ruolo del dominio nella relazione viene associato ad ogni occorrenza di dominio nella relazione. Una ennupla su un insieme di attributi X è una funzione t[Ak] –> Dk che associa a ciascun attributo un valor del dominio Dk di Ak, quindi t[Ak] denota il valore della ennupla t sull'attributo Ak

La rappresentazione tabellare

I valori di ciascuna colonna sono fra loro omogenei, i valori di un attributo appartengono allo stesso dominio. Le righe sono diverse fra loro, una relazione non contiene mai ennuple identiche per via dell'orientamento ai valori. l'ordinamento delle colonne è irrilevante poiché esse sono sempre identificate per nome e non per posizione, lo stesso vale per le righe in quanto identificate per contenuto

L'orientamento ai valori

In un modello relazione, per via dell'orientamento ai valori, le righe di una tabella non hanno un identificativo fisso, l'unico modo per distinguere le righe tra di loro è guardare i valori che contiene.

Per fare riferimento ad una riga specifica non possiamo fare uso del numero della riga, in quanto queste non hanno un posizione fissa ma variabile nel tempo, quindi per andare a identificare una riga dobbiamo fare riferimento ai suoi valori interni. Nel modello relazionale sono presenti due livelli principali per descrivere una relazione, cioè la tabella, il livello intensionale e il livello estensionale.

Il livello intensionale rappresenta la struttura della tabella ed include il nome della relazione, più l'insieme degli attributi e i domini dei valori che ogni attributo può assumere.

Il livello estensionale rappresenta l'insieme di ennuple che la tabella contiene in un dato momento, a differenza del livello intensionale questo può variare nel tempo.

L'informazione incompleta

Il modello relazione impone ai dati una struttura rigida, le informazioni sono rappresentate tramite ennuple con formati predefiniti, esso impone anche che tutti i valori vengano specificati ma è possibile che non tutti i valori siano noti o disponibili per cui è necessario trovare un modo per rappresentarli.

Possiamo farlo in due modi tramite l'utilizzo di valori del dominio "non utilizzati", come può essere la stringa vuota o il valore zero, però può capitare che non esistano valori non utilizzati o non significativi, per cui è stato introdotto il valore NULL, questo denota l'assenza di un valore del dominio.

Bisogna stare attenti perché il valore NULL non ha una semantica definita, a seconda dei casi può rappresentare un valore sconosciuto, un valore inesistente o un valore senza informazione.

Vincoli di integrità

Nelle basi di dati esistono istanze che pur sintatticamente corrette non rappresentano informazioni possibili per l'applicazione di interesse, come può essere un valore negativo associato alla popolazione di un paese.

Il vincolo di integrità indica delle proprietà che devono essere soddisfatte dalle istanze per far si che le informazioni siano corrette per l'applicazione, un vincolo quindi è una funzione booleana che associa ad ogni istanza i valori vero o falso.

Esistono due tipologie principali di vincoli: - I vincoli intrarelazionali: vincoli sui valori o domini, vincoli di ennupla o vincoli di chiave - I vincoli interrelazionali: vincoli di integrità referenziale, vincoli di chiave esterna

I vincoli di dominio

Un vincolo di dominio definisce l'insieme di valori validi che un attributo può assumere, la loro verifica si basa su predicati booleani.

Tali predicati sono valutati al momento dell'inserimento o modifica di un valore nella base di dati andando a garantire che vengano memorizzati solo se il predicato valutato è vero.

I vincoli di ennupla

I vincoli di ennupla sono condizioni da soddisfare sui valori di ciascuna ennupla, indipendentemente dalle altre ennuple, una possibile sintassi è un'espressione booleana che confronta i valori di un attributo con determinati valori scelti oppure tramite espressioni aritmetiche su di essi, anche essi sono verificati al momento dell'inserimento o della modificata.

I vincoli di dominio sono un caso particola di vincoli di ennupla che coinvolgono un solo attributo.

I vincoli di chiave

Una chiave è un insieme di attributi che identificano univocamente le ennuple di una relazione.

Una superchiave di una relazione è un insieme di attributi, può essere anche uno singolo, che permette di identificare in modo univoco ogni riga di una tabella, cioè se consideriamo una superchiave di una riga e la andiamo a confrontare con ogni riga della nostra tabella non ne possiamo trovare una uguale.

Quindi possiamo dire che una chiave è una superchiave con il minor numero possibile di attributi, tale che rimuovendo anche solo un attributo si perderebbe la proprietà di unicità, questo viene detto minimalità.

Trovare una superchiave minimale è il nostro obbiettivo all'interno delle basi di dati.

Nelle basi di dati ogni relazione ha almeno una chiave, per cui una relazione non può contenere ennuple uguali ed ogni relazione ha come superchiave l'insieme degli attributi su cui è definita.

L'esistenza delle chiavi garantisce l'accessibilità a ciascun dato della base di dati e tramite l'utilizzo delle chiavi riusciamo ad organizzare i dati in relazioni diverse andando ad usare le chiavi primarie ed esterne.

Nelle chiavi la presenza di valori nulli deve essere limitata, altrimenti non è possibile identificare le ennuple e realizzare facilmente i riferimenti da altre relazioni, quindi vogliamo imporre a ogni relazione la presenza di almeno una chiave priva di valori nulli, in modo che ci sia sicuramente una combinazione di dati che consenta l'accesso univoco alle ennuple della base di dati.

La chiave primaria è una chiave su cui non sono ammessi i valori nulli, solitamente è evidenziata tramite una sottolineatura, in SQL vengono definite tramite il vincolo PRIMARY KEY.

L'integrità referenziale

Vincoli di integrità referenziale

Solitamente informazioni in relazioni diverse sono correlate attraverso valori comuni, per via dell'orientamento ai valori, in particolare tramite i valori delle chiavi, solitamente primarie.

Un vincolo di integrità relazione impedisce l'inserimento di un valore in una chiave esterna se non esiste nella tabella referenziata, una chiave esterna, detta foreign key, indica un valore di una tabella che esiste sicuramente in una tabella correlata, solitamente è associato alla chiave primaria ma basta che sia associato ad una chiave unica, in SQL vengono definite tramite il vincolo FOREIGN KEY.

Le forme di relazione

Lezione 4 – L'algebra relazionale

L'**algebra relazionale** è il linguaggio procedurale per interrogare basi di dati relazionali. Il risultato di un'interrogazione è una relazione che può essere manipolata utilizzando gli operatori dell'algebra stessa.

Operatori dell'algebra relazione

- Gli operatori si suddividono in: *operatori insiemistici e operatori su relazioni*
- Operatori fondamentali: Unari(σ, π) e Binari($\cup$, $\times$, $-$)
- Operatori derivati: Binari($\bowtie$, $\bowtie_\theta$, $\cap$, $\div$)

Operatori insiemistici

- Due relazioni si dicono compatibili all'unione se sono dello stesso grado e sono definite sugli stessi attributi

Date due relazioni r e s compatibili all'unione, è possibile operare su esse con i seguenti operatori insiemistici:

- ***Unione***: l'unione $r \cup s$ è costituita dalle ennuple appartenenti a r o a s o ad entrambe
- ***Differenza***: la differenza $r - s$ è costituita dalle ennuple apparteneti a r e non a s
- ***Intersezione***: l'intersezione $r \cap s$ è costituita dalle ennuple appartenenti sia a r e sia a s

Compatibilità all'unione

$R(A_1, A_2, \dots, A_n)$ *e* $S(B_1, B_2, \dots, B_n)$ sono compatibili all'unione se:

1. R e S hanno lo stesso grado, cioè stesso numero di attributi
2. Domini corrispondenti degli attributi, gli attributi corrispondenti nelle due relazioni devono appartenere allo stesso dominio ($Dom(A_i) = Dom(B_i) \quad \forall i = 1, \dots, n$)

Operatori unari

Selezione (σ): Restituisce una relazione contenente l'insieme delle sole ennuple della relazione operando che soddisfano particolari condizioni, espresse mediante una formula proposizionale.

Proiezione (π): Restituisce come risultato una relazione contenente l'insieme delle ennuple della relazione operando limitate su particolari attributi.

Nella selezione si tratta di una **decomposizione orizzontale** in quanto va a selezionare alcune ennuple della relazione originale Nella proiezione si tratta di una **decomposizione verticale** in quanto va a selezionare alcuni insiemi di ennuple della relazione originale

Operatori binari

Prodotto cartesiano ($\times$)

Date due relazioni $r = \mathrm{R(X)}$ e $s = \mathrm{S(Y)}$ con $X \cap Y = \emptyset$ il prodotto cartesiano $r \times s$ è una nuova relazione definita sull'insieme di attributi XY contenente tutte le possibili combinazioni delle tuple di r e s.

Operatori Join

Theta-Join ($\bowtie_\theta$)

Date due relazioni $r = \mathrm{R(X)}$ e $s = \mathrm{S(Y)}$ con X e Y disgiunti, il Theta-Join $r \bowtie_\theta s$ è una relazione definita su XY composta dalle ennuple risultato della concatenazione di ennuple di r con ennuple di s che soddisfano le condizioni di confronto $A\theta B$ con $\mathrm{A} \in \mathrm{X}$ e $\mathrm{B} \in \mathrm{Y}$

- θ è una espressione composta per mezzo di operatori di confronto ($=, \neq, >, <, \geq, \leq$)
- Se il confronto è di uguaglianza, l'operazione è detta **Equijoin**
- Possiamo notare che il join $\mathrm{R} \bowtie_{A=B} \mathrm{S}$ è del tutto equivalente all'operazione $\sigma_{A=B}(\mathrm{R} \times \mathrm{S})$

Join naturale ($\bowtie$)

Date due relazioni r(YX) e s(XZ), il join naturale di r e s è l'operazione che combina le tuple delle due relazioni basandosi sull'uguaglianza dei valori dell'attributo X, il quale è comune ad entrambe.

Possiamo dare le seguenti definizioni:

- Due ennuple sono *joinabili* se è possibile effettuare l'operazione di join su di esse
- Una ennupla è detta *dangling* quando non contribuisce all'operazione di join
- Una operazione di join viene detta *completa* quando non contiene ennuple dangling
- Tramite una *ridenominazione*(ρ) possiamo rendere compatibili all'operazione di join due attribuiti uguali concentualmente ma con sintassi differente

Proprietà del join naturale

- Il join naturale è commutativo e associativo
- Può essere espresso in termini di:
  - Prodotto cartesiano
  - Selezione ennuple con valori uguali per attributi in comune
  - Proiezione di r e s eliminando duplicazioni
- Se r(X), s(Y) con X $\cap$ Y = $\emptyset$, allora $r \bowtie s = r \times s$
- Se r(X), s(Y) con X = Y, allora $r \bowtie s = r \cap s$

Outer Join

L'outer join è una variante del join che conserva anche le ennuple dangling (quelle che non trovano corrispondenza), inserendo valori NULL nei campi mancanti delle relazioni. Questo permette di evitare la perdita di dati che si verifica nei join naturali e theta join.

Tipologie di Outer Join:

1) Left outer join (Left Join)

- Mantiene tutte le ennuple della relazione di sinistra

2) Right Outer Join (Right Join)

- Mantiene tutte le ennuple della relazione di destra

3) Full Outer Join (Full Join)

- Mantiene tutte le ennuple di entrambi le relazioni

Divisione

La divisione si applica solo a relazioni r(Z) e s(X), dove X $\subseteq$ Z. La divisione va a restituire una relazione T(Y), dove Y = Z - X, contenente tutte le ennuple t tali che:

- Esistano ennuple tr in r con tr[Y] = t
- Per ogni ennupla ts in s, esista una ennupla tr in r con tr[X] = ts

Intuitivamente la divisione verifica una condizione o corrispondenza fra una o più ennuple di r e tutte le tuple di s.

Equivalenza di espressioni algebriche

Caratteristiche generali

-
-
-
-

regole di trasformazione

-
-
-
-
-

Lezione 5 – SQL

Introduzione

Le operazioni basilari per i database sono le DML, Data Manipulation Language, queste permettono di modificare i dati all'interno di una tabella.

L'operazione di **inserimento** di valori in una tabella di un database avviene tramite il seguente schema di comandi:

INSERT INTO *NomeTabella* (*ListaAttributi*) VALUES (*ListaDiValori*);

Un esempio pratico è il seguente:

```
INSERT INTO studenti (matricola, nome, età) VALUES (345216, 'Luigi', 21);
```

Da tenere presente che è anche possibile andare ad importare insiemi di righe di dati da altre tabelle del nostro database o anche in maniera estenstiva scrivendo le varie ennuple separate da parantesi rotonde all'interno del comando citato in precedenza.

L'operazione di **modifica** dei valori di una tabella di un database avviene tramite il seguente schema di comandi:

UPDATE *NomeTabella* SET *Attributo* = (*Valore*)

Un esempio pratico è il seguente:

```
UPDATE studenti SET età = 20;
```

Prestiamo attenzione che il seguente comando lavora sull'intera tabella, se vogliamo andare a modificare in modo più selettivo dobbiamo fare l'utilizzo dell'opzione WHERE, per esempio:

```
UPDATE studenti SET età = 20 WHERE matricola = 345216;
```

L'operazione di **Cancellazione** di valori in una tabella di un database avviene tramite il seguente schema di comandi:

DELETE FROM *NomeTabella*

Un esempio pratico è il seguente:

```
DELETE FROM studenti;
```

Anche in questo caso il comando precedente lavora sull'intera tabella, quindi se vogliamo una cancellazione selettiva dobbiamo fare utilizzo del comando aggiuntivo WHERE, per esempio:

```
DELETE FROM studenti WHERE matricola = 345216;
```

Interrogazioni

Introduzione alle interrogazioni

Per effettuare le interrogazioni sui dati salvati nei nostri database avremo un linguaggio SQL dedicato detto DQL, Data Query Languange. il DML, come SQL, è un linguaggio dichiarativo cioè l'utente descrive l'obbiettivo dell'operazione ma non specifica i dettagli su come il database deve eseguire l'operazione. l'interprete del DBMS analizza l'istruzione della query e genera il corrispettivo codice in linguaggio procedurale, il tutto in modo nascosto dall'utente.

Il Query Optimizer è un modulo del DBMS che ottimizza la query trovando il modo più efficiente per eseguire l'interrogazione.

il linguaggio SQL può essere incluso anche in altri programmi/applicazioni, scritti anche con linguaggi differenti, per interagire con i nostri database.

L'istruzione SELECT

l'istruzione SELECT è il comando fondamentale del DQL, questo viene usato per recuperare dei dati da una o più tabelle.

La sintassi di base del SELECT è la seguente:

SELECT *ListaAttributi* FROM *ListaTabelle* (WHERE *Condizione*)

- SELECT: Indica gli attributi, colonne, dalle quali recuperare i dati cercati.
- FROM: Nome delle tabelle sulle quali la query verrà valutata.
- WHERE: Espressione booleana di ricerca che filtra le ennuple in base ad una condizione.

Esempi di utilizzo:

Selezione di una intera tabella:

```
SELECT * FROM clienti;
```

Selezione di colonne specifiche:

```
SELECT nome, cognome FROM clienti;
```

Selezione di una intera tabella o di colonne specifiche con filtrazione sui dati:

```
SELECT * FROM clienti WHERE città = 'roma'; SELECT nome, cognome FROM clienti WHERE città = 'roma';
```

La clausola WHERE

La clausola WHERE necessità di espressioni booleane costruite tramite l'utilizzo di operatori di confronto, che possono essere combinati tra di loro tramite gli operatori logici AND, OR, NOT.

Gli operatori di confronto basici disponibili in SQL sono i seguenti: $=, <>, <, >, <=, >=, IS NULL, IS NOT NULL. Inoltre sono presenti anche operatori di confronto avanzanti come possono essere BETWEEN, IN, EXISTS e LIKE.