

RIASSUNTO SPERIMENTAZIONE

Gerarchie

- Totale
 - Tutti gli individui dell'entità padre sono individui di **almeno una** delle entità figlie
- Parziale
- Esclusiva
 - Tutti gli individui dell'entità padre sono individui di **al massimo una** delle entità figlie
- Sovrapposta

Pattern di progettazione

- Reificazione di attributo di entità
 - Un attributo è rilevante, creo un'entità aggiuntiva
- Part-of
 - L'esistenza della parte dipende dall'esistenza di ciò che la contiene
- Instance-of
 - Un'entità è istanza di un'entità astratta, o modello
- Reificazione di associazione binaria
 - L'associazione diventa un'entità che collega le altre entità
- Reificazione di associazione ricorsiva
 - Da 1 associazione e 1 entità a 2 associazioni e 2 entità
- Reificazione di attributo di associazione
 - Se l'attributo dell'associazione è importante creo un'entità
- Caso particolare di entità
 - Creo generalizzazione
- Storicizzazione di entità
 - Creo generalizzazione
- Storicizzazione di associazione
 - Associazione in entità con relativa generalizzazione
- Evoluzione di concetto
 - Aggiungo attributi
- Reificazione di associazione ternaria
 - Aggiungo delle associazioni e faccio una part-of di tutte e tre

Strategie di progetto

- Top-down
 - Individuo i concetti cardine e creo una struttura scheletro dello schema
 - Mediante trasformazioni lo raffino per descrivere i vari concetti con maggiore dettaglio
 - **PRO:** permette di inizialmente trascurare alcuni dettagli che possono essere specificati successivamente
 - **CONTRO:** possibile solo quando si possiede una visione globale di tutte le componenti
- Bottom-up
 - Specifiche suddivise in parti elementari, che vengono poi tradotte in semplici schemi concettuali e poi fusi (integrazione)

- **PRO:** adatta a una progettazione di gruppo
- **CONTRO:** L'integrazione di schemi concettuali diversi può comportare difficoltà notevoli
- Inside-out
 - Evitare la fase di integrazione
 - Si individuano alcuni concetti importanti e poi da questi ci si muove a macchia d'olio
 - **PRO:** non richiede passi di integrazione
 - **CONTRO:** è necessario continuamente riesaminare tutte le specifiche per individuare concetti non ancora rappresentati e descriverli nel dettaglio
- Mista
 - Si individuano i concetti principali e si realizza uno schema scheletro
 - Si decompone
 - Poi si raffina, si espande e si integra

1. Analisi delle ridondanze

Attributi derivabili:

- Da altri attributi della stessa entità (o associazioni)
- Da attributi di altre entità (o associazioni)
- Dal conteggio delle partecipazioni ad una associazione

2. Eliminazione delle generalizzazioni

- Accorpamento dei figli nel genitore
 - Poca differenza tra le istanze delle varie entità
- Accorpamento del genitore nei figli
 - Solo se la generalizzazione è TOTALE
 - Pochi accessi alle istanze di entrambe le entità
- Sostituzione della generalizzazione con associazioni
 - Esclusiva: vincolo E0 non può partecipare contemporaneamente a R1 e a R2
 - Totale: vincolo E0 deve partecipare a R1 o a R2
 - Le operazioni accedono solo a istanze delle figlie o del padre

3. Partizionamento/accorpamento di concetti

Gli accessi si riducono:

- Separando attributi di uno stesso concetto ai quali si accede in operazioni diverse
- Accorpendo attributi di concetti diversi a cui si accede con le medesime operazioni
- Partizionamento Verticale di entità
- Partizionamento Orizzontale di entità
 - Operazioni che accedono solo a determinate occorrenze di una entità
- Partizionamento di associazione
 - Utile quando le operazioni fanno distinzione tra i 2 tipi di associazione
- Accorpamento di entità
 - Utile quando le transazioni che accedono ad un'istanza di Persona necessitano sempre di informazioni relative anche all'appartamento
- Eliminazione di attributi multivalore

- Gli attributi multivalore non sono direttamente rappresentabili nel modello relazionale

4. Scelta di identificatori principali

- Pochi attributi
- Identificatori interni

Business Rules

- **Vincolo di integrità**
 - <concetto> deve/non deve <espressione>
- **Derivazione**
 - <concetto> si ottiene <operazione>

Qualità di uno schema concettuale

- Correttezza
- Completezza
- Leggibilità
- Minimalità

Operatori insiemistici

- Union $A \cup B$
 - Insieme delle tuple x tali che x appartiene ad A o appartiene a B (o entrambi)
- Intersect $A \cap B$
 - Insieme delle tuple x tali che x appartiene ad A e appartiene anche a B
- Except A / B
 - Insieme delle tuple x tali che x appartiene ad A ma non appartiene a B

Viste

Rappresentazioni diverse dello stesso insieme di dati

- Relazione virtuale
 - Tabelle definite per mezzo di query SQL. Non sono tabelle effettivamente memorizzate nello schema del DB ma possono venire utilizzate come se lo fossero.
- Viste materializzate
 - Tabelle, derivate da espressioni SQL, effettivamente memorizzate come tabelle nel DB e tenute automaticamente sincronizzate con le tabelle di base

Tipi di join

- Inner join
- Join
- Left/right join
- Full join
- Natural join
- Self join

Sottoquery

- **Semplici** (o stratificate): è possibile valutare prima l'interrogazione più interna, poi, sulla base del suo risultato, valutare l'interrogazione più esterna
- **Correlate** (o incrociate): l'interrogazione più interna fa riferimento a una delle tabelle appartenenti all'interrogazione più esterna. Per ciascuna riga candidata alla selezione nell'interrogazione più esterna, è necessario valutare nuovamente la sottoquery.

Algebra relazionale

- **SQL** è un linguaggio **dichiarativo**: l'utente definisce cosa vuole ottenere e il DBMS determina come ottenerlo.
 - La query SQL viene analizzata dal query optimizer per essere eseguita in modo efficiente
- **L'algebra relazionale** è **procedurale**: l'interrogazione specifica i passi da compiere per estrarre le informazioni

SQL	ALGEBRA RELAZIONALE
Select * from S;	S
Select SName, City from S;	$\pi_{SName, City}(S)$
Select SName, City from S where Status > 20	$\pi_{SName, City}(\sigma_{Status > 20}(S))$
Select S.SName, S.City, SP.Qty, P.PName, P.Color From S, SP, P	$\pi_{SName, S.City, Qty, PName, Color}(S \times SP \times P)$

Differenze tra Algebra Relazionale e SQL:

- In **Algebra relazionale** una tabella viene vista come una relazione/insieme dal punto di vista matematico
 - Insieme di tuple necessariamente diverse tra loro
- In **SQL** si possono avere nelle tabelle risultanti righe uguali, ovvero righe con gli stessi valori per tutti gli attributi

Modello 3-tire

- User interface
- Business
- Data → DBMS

Cos'è una base di dati:

Un insieme di dati **atomici**, **strutturati** e **permanenti**, raggruppati in **insiemi omogenei** in relazione tra loro, organizzati con la **minima ridondanza** per essere utilizzati da **applicazioni diverse** in **modo controllato**.

Ciclo di vita di un sistema informativo

- Studio di fattibilità
- Raccolta e analisi dei requisiti
- Progettazione
 - Dati
 - Funzionalità
- Implementazione
- Validazione e collaudo
- Funzionamento

Modello concettuale

Permettono di descrivere i dati in maniera indipendente dal DBMS.

Si usano per analizzare in modo rigoroso e astratto la realtà di interesse.

Si rappresenta il contenuto informativo del DB.

Modello logico

Usa un modello dei dati, che è l'insieme dei costrutti utilizzati per organizzare i dati di interesse e descriverne la dinamica

DBMS

- **DDL** (Data Definition Language)
- **DML** (Data Manipulation Language)

Funzioni condizionali

- Coalesce
 - Ammette una sequenza di espressioni, e restituisce il primo risultato non nullo
- Nullif
 - Ammette un'espressione ed un valore costante, se il risultato dell'operazione è uguale alla costante, restituisce null

Ordine di valutazione postgresql

1. From
2. Where
3. Group by
4. Having
5. Select
6. Distinct
7. Order by
8. Limit