

Esercitazione di Basi di Dati

Corso di Fondamenti di Informatica

29 Aprile 2004

Da Access a Protégé

Marco Pennacchiotti

pennacchiotti@info.uniroma2.it

Tel. 0672597334

Ing.dell'Informazione, stanza 1035 (primo piano)

Sommario Database

Progettazione di un Database:

1. Diagramma Entità-Relazioni (ER)

1. Modellazione entità e attributi
2. Modellazioni relazioni (1-molti, molti-molti)

2. Da ER a Access

1. Implementazione delle entità e delle relazioni in tabelle
2. Query
3. Maschere
4. Implementazione di relazioni IS_A

ER e DB

Relazione IS_A

La **relazione IS_A** esprime un legame di generalizzazione/ specializzazione tra entità.

L'entità **superclasse** generalizza le sottoclassi.

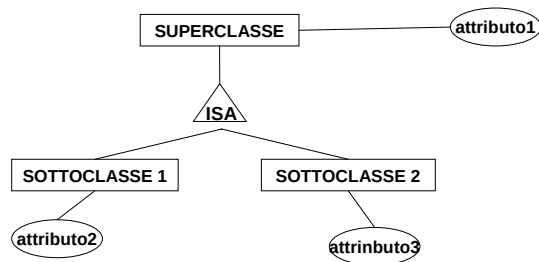
Le entità **sottoclassi** sono specializzazioni della superclasse.

Le sottoclassi **ereditano** gli attributi della superclasse

Attributi di SOTTOCLASSE1

attributo2

attributo1 → ereditato da SUPERCLASSE



IS_A

Esempio...

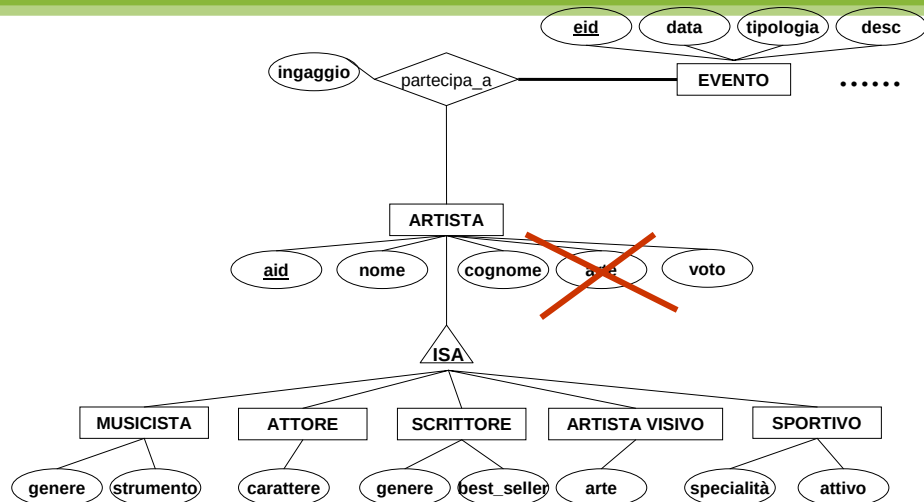


Diagramma ER con IS_A

... quando usare la IS_A in un ER ?

- ✗ Alcune volte è naturale classificare concettualmente entità in una *gerarchia*
- ✗ IS_A o attributo?
Se il concetto in questione che *specifica* l'entità (es. *arte* in *artista*) è semplice e non strutturato si può usare un attributo.
Se il concetto è invece strutturato si preferisce una IS_A.
- ✗ Non esiste una regola generale... dipende dalle esigenze dell'utente finale.

IS_A

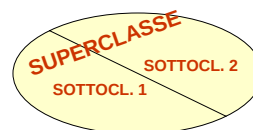
IS_A in Access

Due modalità per implementare la IS_A:

1. **Creare una tabella per la superclasse ed una per ogni sottoclasse:**
nella tabella della superclasse sono rappresentati gli attributi comuni a tutte le sottoclassi; nelle tabelle delle sottoclassi sono rappresentati solo gli attributi specifici della sottoclasse.
2. **Creare solo tabelle per le sottoclassi:**
in nella tabella di una sottoclasse sono rappresentati tutti gli attributi della sottoclasse, sia quelli specifici che quelli ereditati dalla superclasse.

Il primo metodo è sempre utilizzabile.

Il secondo metodo è utilizzabile solo nel caso in cui le sottoclassi *partizionano* completamente le superclassi.



IS_A

Esercizio 1: creare una IS_A

1. Creare le tabelle Access che implementino la relazione IS_A dell'ER precedente (*artista* → *musicista*...) utilizzando entrambe le metodologie.
2. Implementare le seguenti queries:
 1. *A quali eventi partecipa un dato musicista ?*
 2. *Quali sono i nomi dei musicisti ?*

IS_A in Access ... troppi problemi ...

SVANTAGGI:

Primo Metodo

- Difficoltà nell'inserimento delle entità: una entità va inserita sia nella tabella della superclasse che in quella della sottoclasse!
- Difficoltà per eseguire particolari queries: ad esempio, come sapere se l'artista è un musicista?

Secondo metodo

- Se esisto entità della superclasse che non appartengono a nessuna sottoclasse (ad es. un artista che non è né musicista, né artista visivo, né scrittore, né sportivo né attore) il metodo non è utilizzabile



CHE FARE ? ...

IS_A

Protégé

CHE FARE ?

Utilizzare un tool che gestisca in maniera migliore le relazioni IS_A e l'ereditarietà



PROTEGE

Protégé è un tool che consente di implementare un diagramma ER, gestendo in maniera efficace ed intuitiva le relazioni IS_A e l'ereditarietà



Protégé

Protégé... molto di più di un database

ATTENZIONE!

Protégé non è una Database, ma un tool per sviluppare
ontologie

— **ONTOLOGIA:** una definizione operativa —

Una *ontologia* è una descrizione formale esplicita dei concetti di un dominio.

Una ontologia rappresenta cioè il *modello concettuale* di un mondo. Il modello concettuale è la struttura formale di (un pezzo di) realtà percepita ed organizzata da chi modella.

ontologie

Ontologie e Database

Sia le ontologie che i Database sono *metodologie di rappresentazione della conoscenza*. Si differenziano però in molti aspetti.

DIFFERENZE PRINCIPALI

1. Approccio alla realtà da modellare: con un Database si ha una visione più concreta e specifica del mondo; con una ontologia si cerca di modellare concettualmente il mondo (il DB si concentra più sulle istanze, l'ontologia più sulle entità).
2. Le Ontologie consentono di *ragionare* sul mondo, ovvero di espandere la conoscenza sul mondo, utilizzando *regole di inferenza*, espresse in uno specifico *linguaggio*.
3. Le ontologie permettono una implementazione più naturale della relazione IS_A

ontologie

Ontologia... cosa contiene

Dovendo modellare un dominio, una ontologia deve essere in grado di esprimere:

1. **Concetti** (o classi) del dominio
2. **Proprietà** (o slot) dei concetti
3. **Restrizioni** sulle proprietà

CORRISPONDENZE CON I DB

ENTITA'

ATTRIBUTI + RELAZIONI

RESTRIZIONI

Una ontologia popolata di **istanze** e completata con delle *regole di inferenza* viene detta **Base di Conoscenza**.

ontologie

Ontologia... come ragionare

Il ragionamento su una ontologia viene implementato *attraverso regole di inferenza*.

Un ESEMPIO di 2000 anni fa...

SILLOGISMO

“Tutti gli uomini sono mortali. Socrate è un uomo. Quindi, Socrate è mortale.”

(Artistotele)

La relativa regola di inferenza è:

Uomo(Socrate)



NELL'ONTOLOGIA E' PRESENTE LA CLASSE
UOMO CHE HA COME ISTANZA SOCRATE

$\forall \text{Uomo}(x) \Rightarrow \text{Mortale}(x)$



NELL'ONTOLOGIA OGNI UOMO IS_A MORTALE

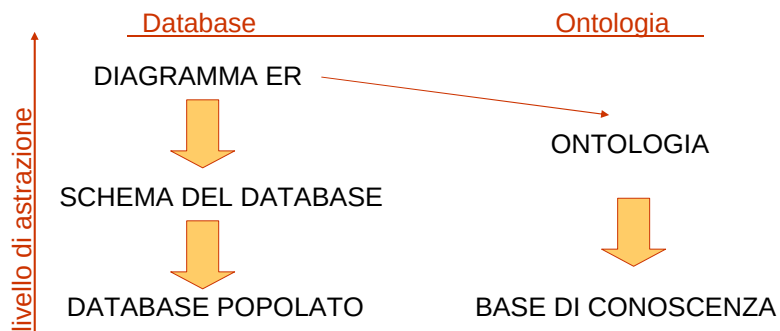
Mortale(Socrate)



DEDUZIONE E AUMENTO DELL'ONTOLOGIA

ontologie

Riassumendo: un parallelo...



Un ER può essere *tradotto* in una ontologia.

Inoltre, una ontologia permette di rappresentare più efficacemente di un ER domini particolarmente complessi.

ontologie

Nella prossima puntata ...

“.... Come si costruisce una ontologia ? ”

