



La metodologia WebML

La progettazione dei dati

Prof. Devis Bianchini

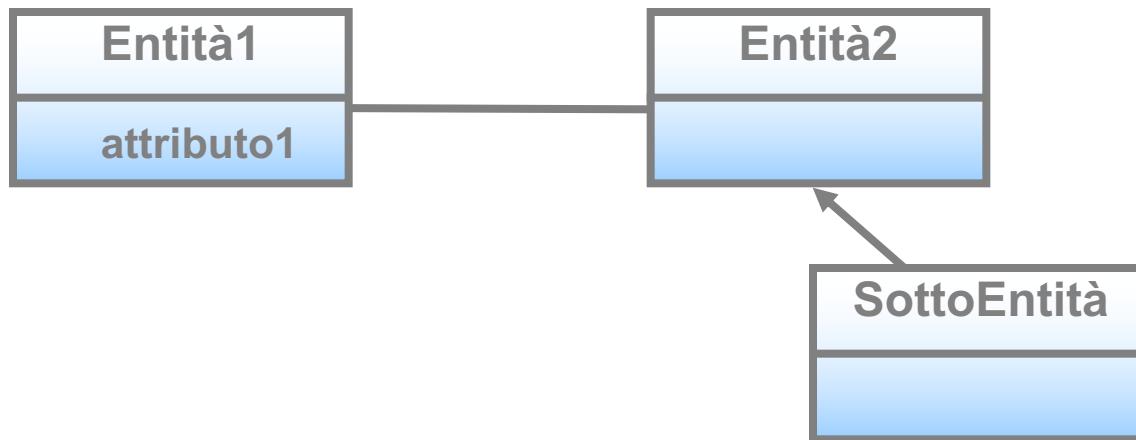
Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Brescia



Progettazione dei dati

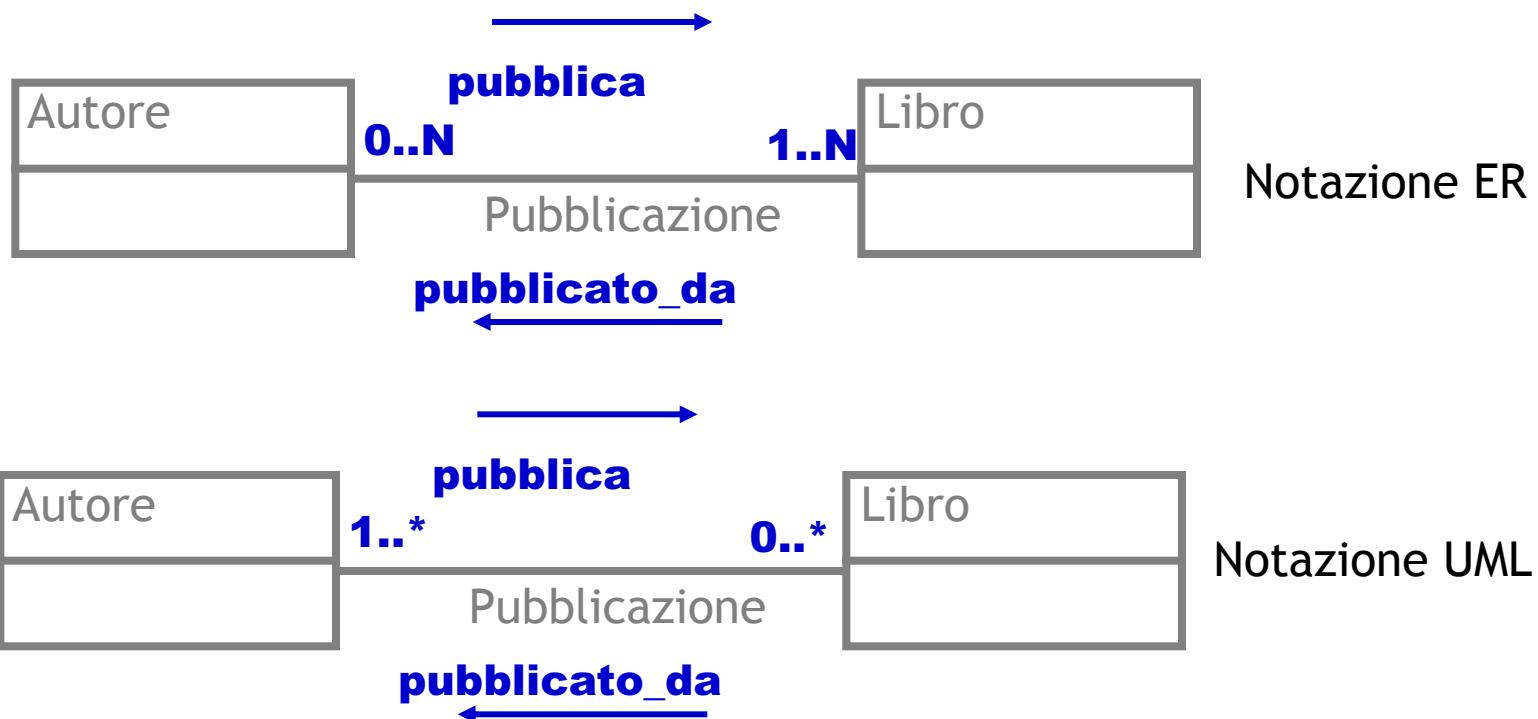
- **IN:** dizionario dei dati, mappa del sito, requisiti funzionali, requisiti degli utenti
 - **OUT:** schema formale dei dati (schema E-R/ UML, sotto forma di entità, attributi, relazioni e gerarchie IS-A)
-
- Fondamentale anche in caso di contenuti pre-esistenti
 - Trae vantaggio da una classificazione degli oggetti informativi da rappresentare nello schema

Notazione grafica (ER/UML)



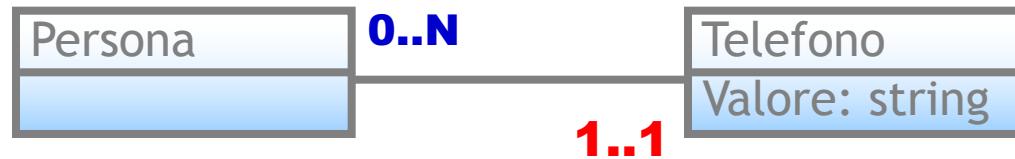
Ruoli e cardinalità delle relazioni

- Il *ruolo di una relazione* è una delle due “direzioni” in cui una relazione può essere interpretata



Attributi multi-valore

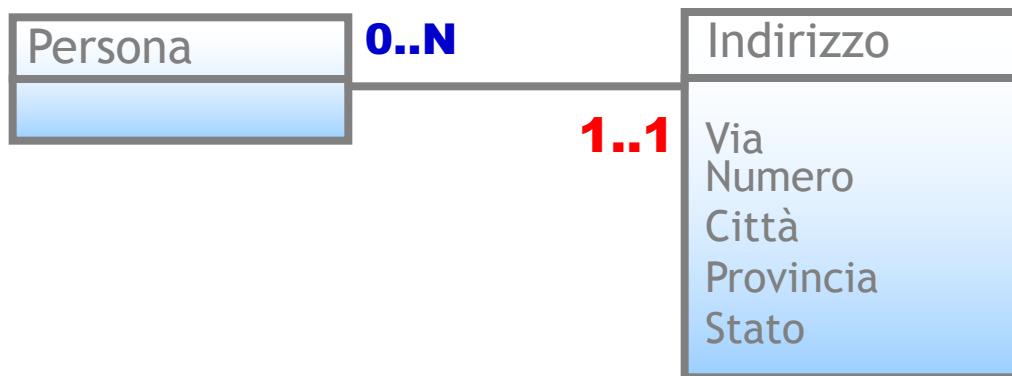
- Attributi di un oggetto che possono assumere un insieme di valori
- Rappresentati per mezzo di una entità più una relazione
 - Es.: una persona ha più numeri telefonici



NOTA: L'entità **Telefono** è **debole**

Attributi composti

- Attributi con una struttura interna (per es., un indirizzo può includere diversi campi)
- Rappresentati per mezzo di una entità e una relazione
 - Es.: una persona ha uno o più indirizzi, ognuno formato da numero, via, città, provincia, stato



NOTA: L'entità
Indirizzo è **debole**

Relazioni N-arie

- Relazioni che coinvolgono N entità ($N > 2$)
 - Es.: la fornitura di componenti ad un dipartimento da parte di una ditta
- Rappresentabili da una combinazione di entità e N relazioni binarie
 - Es magazzino acquisti



NOTA: L'entità Fornitura è debole

Relazioni con attributi

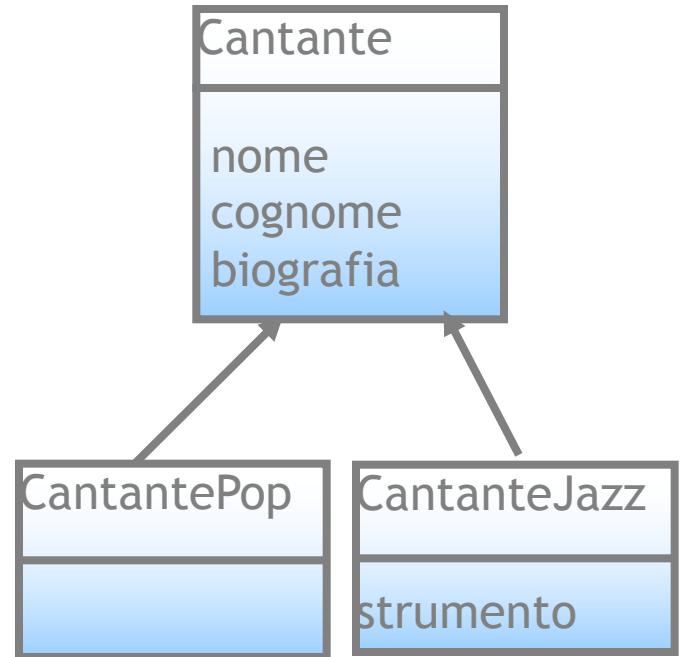
- Rappresentate da una entità centrale e due relazioni
 - Es.: il voto preso da uno studente nell'esame di un certo corso



NOTA: l'entità Voto è debole

Gerarchie IS-A

- Le sotto-entità ereditano gli attributi e le relazioni di una super entità
- Le sotto-entità possono anche avere proprietà localmente definite
- Alcune ipotesi restrittive in WebML:
 - non è permessa l'ereditarietà multipla
 - è possibile definire solo gerarchie esclusive
 - ogni entità può apparire al massimo in una gerarchia



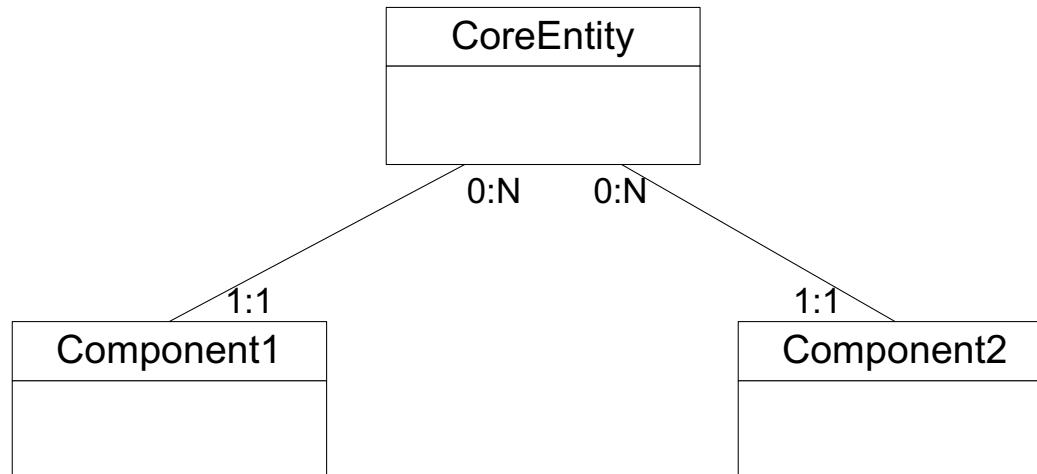
Classificazione degli oggetti informativi



- Gli elementi dello schema ER possono essere diversamente classificati rispetto al loro utilizzo:
 - **core**: i contenuti informativi principali del sito
 - **interconnessione**: per passare da un contenuto informativo core all'altro
 - **accesso**: per permettere all'utente di raggiungere il più velocemente possibile i contenuti informativi core
 - **personalizzazione**: per adattare parti diverse del sito web alle esigenze di attori diversi

Oggetti «core»

- Oggetti informativi principali gestiti dall'applicazione
 - Entità centrali attorno alle quali si sviluppa l'intero schema
 - Più di una singola entità – rappresentazione di proprietà complesse e di componenti





Sottoschema «core» (I)

- **Sotto-schema core:** insieme di entità connesse da relazioni, che collettivamente rappresentano un singolo concetto core
- Partendo da un concetto core nel dizionario dati:
 1. il concetto core è rappresentato da un'entità, detta **entità core**
 2. le proprietà con valore singolo e atomico diventano **attributi** dell'entità core; le proprietà chiave diventano **chiavi** dell'entità core
 3. le proprietà con valori multipli o strutturati diventano **componenti** interni dell'entità core

Sottoschema «core» (II)

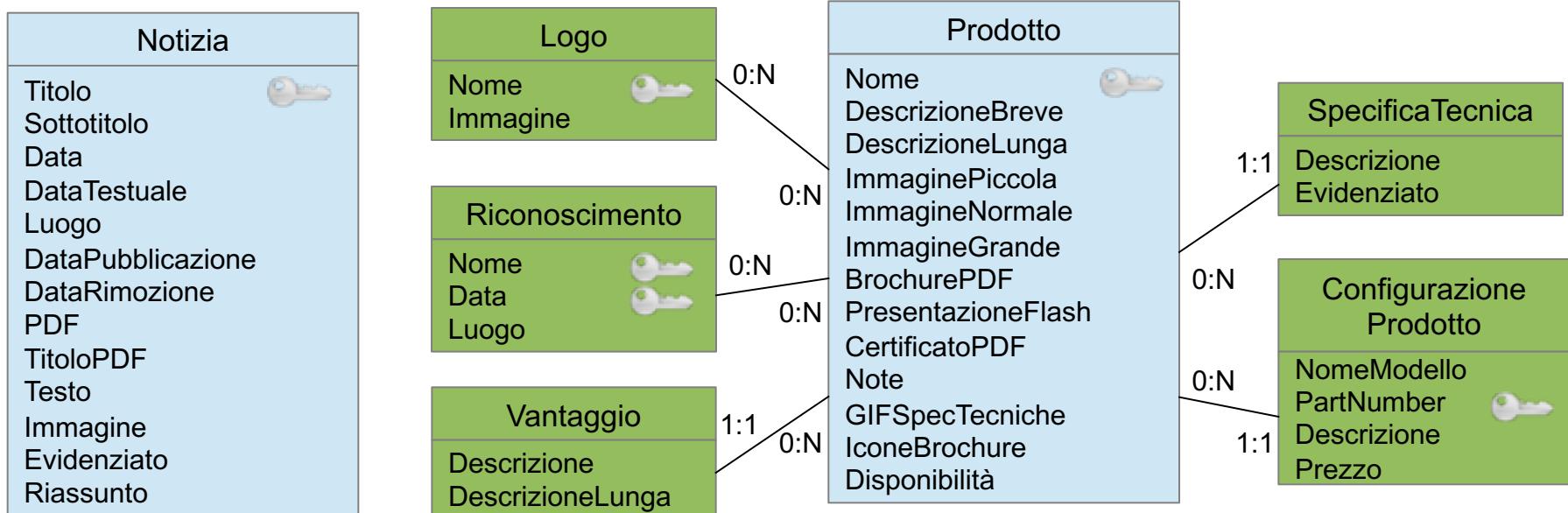
- I componenti interni sono rappresentati come entità associate all'entità core attraverso una relazione. I vincoli di cardinalità delle relazioni che associano il componente all'entità core possono creare due casi:
 1. relazione con vincolo di cardinalità 1:1 dalla parte del componente; il componente è una sotto-parte propria del concetto core, nessuna istanza del componente interno può esistere in assenza dell'entità core a cui è associata; inoltre, la stessa istanza di un componente interno non può essere condivisa da più oggetti core; nella terminologia ER i componenti interni sono chiamati **entità deboli**, mentre nella terminologia del modello ad oggetti sono chiamati **componenti part-of**

Sottoschema «core» (III)

2. relazione con vincolo di cardinalità 0:N. dalla parte del componente; il componente interno è considerato una parte del concetto core; una sua istanza può sia esistere indipendentemente dall'associazione ad una istanza dell'entità core, sia essere condivisa tra oggetti core diversi; il componente interno non può tuttavia essere considerato un elemento centrale e a sé stante e quindi non può essere elevato al ruolo di concetto core

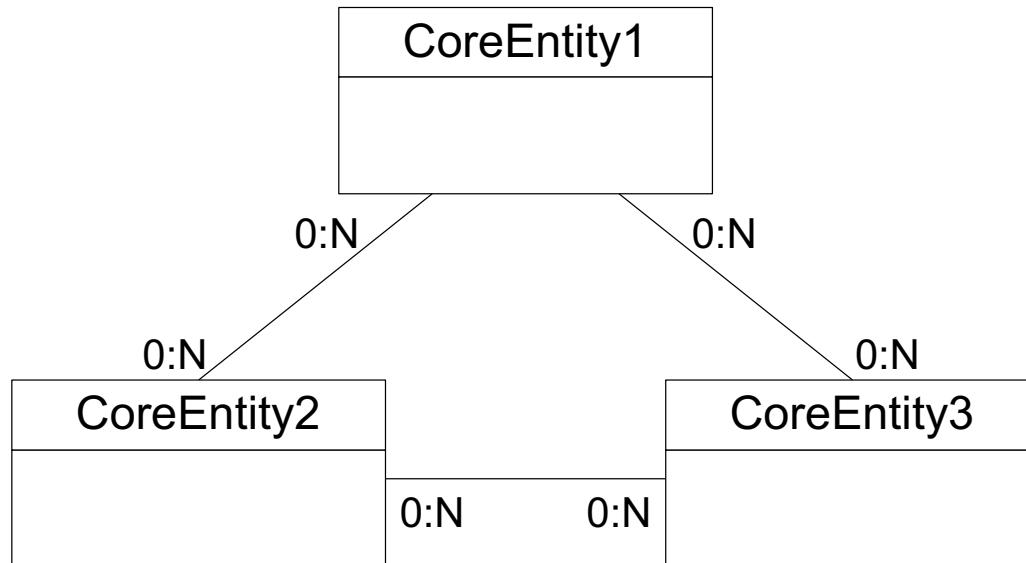
Sottoschema «core» (IV)

- Progetto dettagliato del sotto-schema core:* attributi, componenti e relazioni



Oggetti di «interconnessione»

- Associazioni semantiche tra oggetti core, usati per definire link e indici per la navigazione tra oggetti core



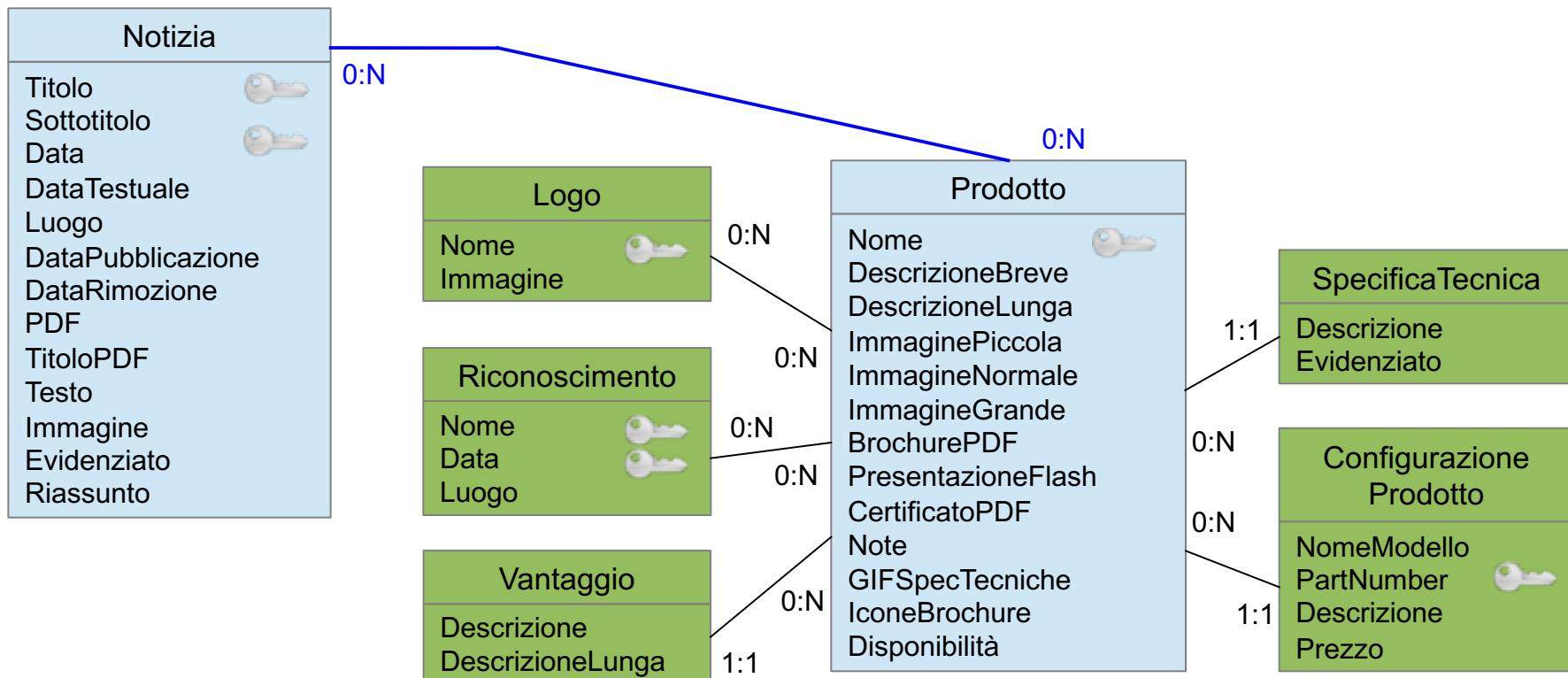


Sottoschema di «interconnessione» (I)

- I **sottoschemi di interconnessione** prevedono pattern di relazioni, introdotti nello schema dei dati per esprimere associazioni semantiche tra oggetti core
- Derivano direttamente dalle associazioni tra oggetti core dichiarate nel dizionario dei dati
- Ogni associazione semantica nel dizionario dei dati induce una relazione tra le entità core coinvolte che ha cardinalità molti a molti (a meno di vincoli specifici dichiarati nel dizionario dei dati)

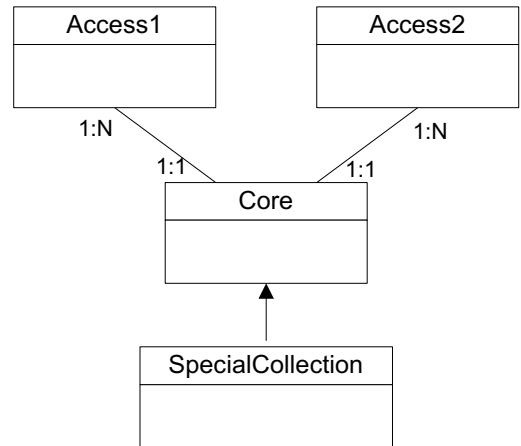
Sottoschema di «interconnessione» (II)

- Definizione di connessioni semantiche tra sottoschema core



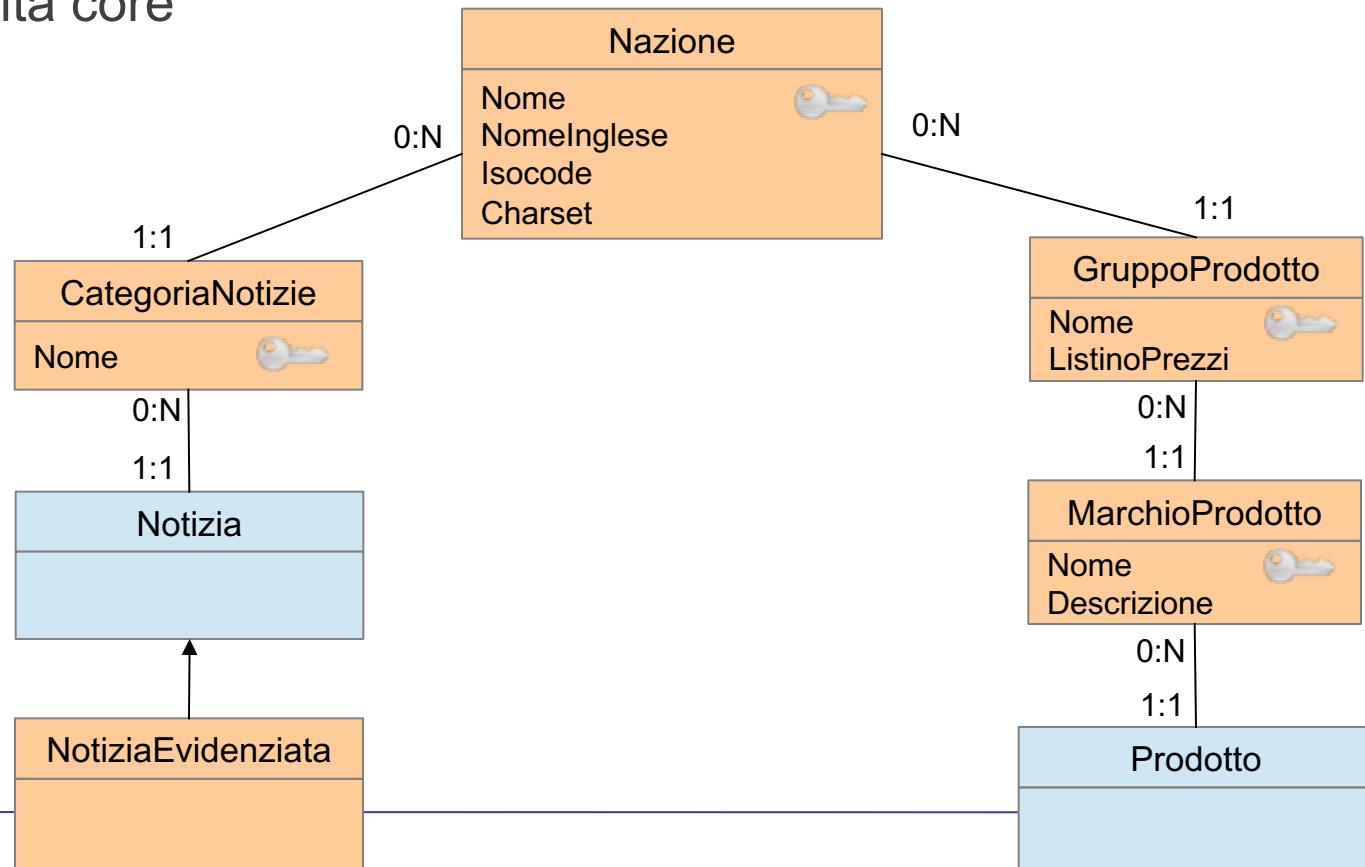
Oggetti di «accesso»

- Oggetti ausiliari, usati per classificare o specializzare gli oggetti core, allo scopo di facilitarne l'accesso in modi diversi:
 - categorizzazioni per costruire gerarchie di indici (categorie-prodotti)
 - meccanismi di ricerca efficienti
 - collezioni di selezioni di oggetti core (le offerte del giorno)
- Uno stesso oggetto core può essere categorizzato o specializzato in diversi modi



Sottoschema di «accesso»

- Sotto-schema di accesso:** Insieme di entità nello schema E-R, connesse tramite relazioni di categorizzazione o generalizzazioni sulle entità core



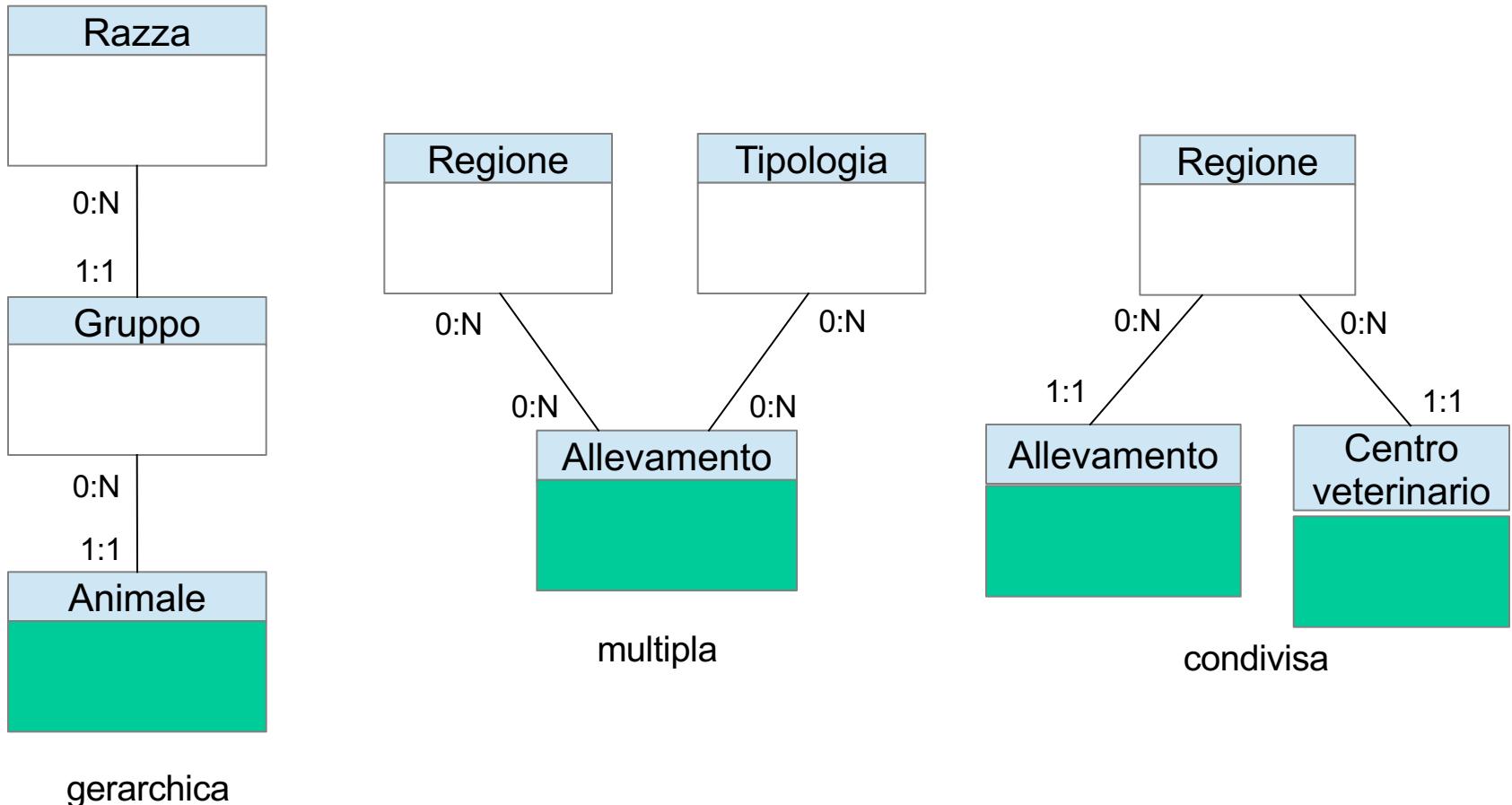


Categorizzazione (I)

Gli oggetti di categorizzazione possono essere organizzati in tre modi:

- I concetti di categorizzazione possono essere a loro volta categorizzati in modo da formare una ***gerarchia di categorizzazione***
- Lo stesso concetto può essere soggetto a più categorizzazioni originando ***categorizzazioni multiple***
- Lo stesso oggetto di categorizzazione può essere utilizzato per classificare più di un concetto ottenendo una ***categorizzazione condivisa***

Categorizzazione (II)



Oggetti di «personalizzazione» (I)

- Oggetti che rappresentano proprietà dell'utente, necessarie per definire politiche di personalizzazione



- L'entità **Utente** specifica i dati degli utenti individuali dell'applicazione
- L'entità **Gruppo** raccoglie informazioni che caratterizzano le classi di utenti con requisiti omogenei, sono proprietà collettive
- La relazione molti-a-molti **Appartenenza** associa Utente a Gruppo stabilendo che un utente possa appartenere ad uno o più gruppi e che un gruppo comprenda più utenti

Oggetti di «personalizzazione» (II)

- Oggetti che rappresentano proprietà dell'utente, necessarie per definire politiche di personalizzazione



- La relazione uno-a-molti **GruppoDefault** associa Utente a Gruppo per indicare che, tra tutti i gruppi di appartenenza, solo uno è il gruppo di default
- La relazione uno-a-molti **Accesso** associa Gruppo a SiteView per rappresentare l'assegnazione delle site view ai vari gruppi. Le relazioni GruppoDefault e Accesso assegnano un'unica site view all'utente



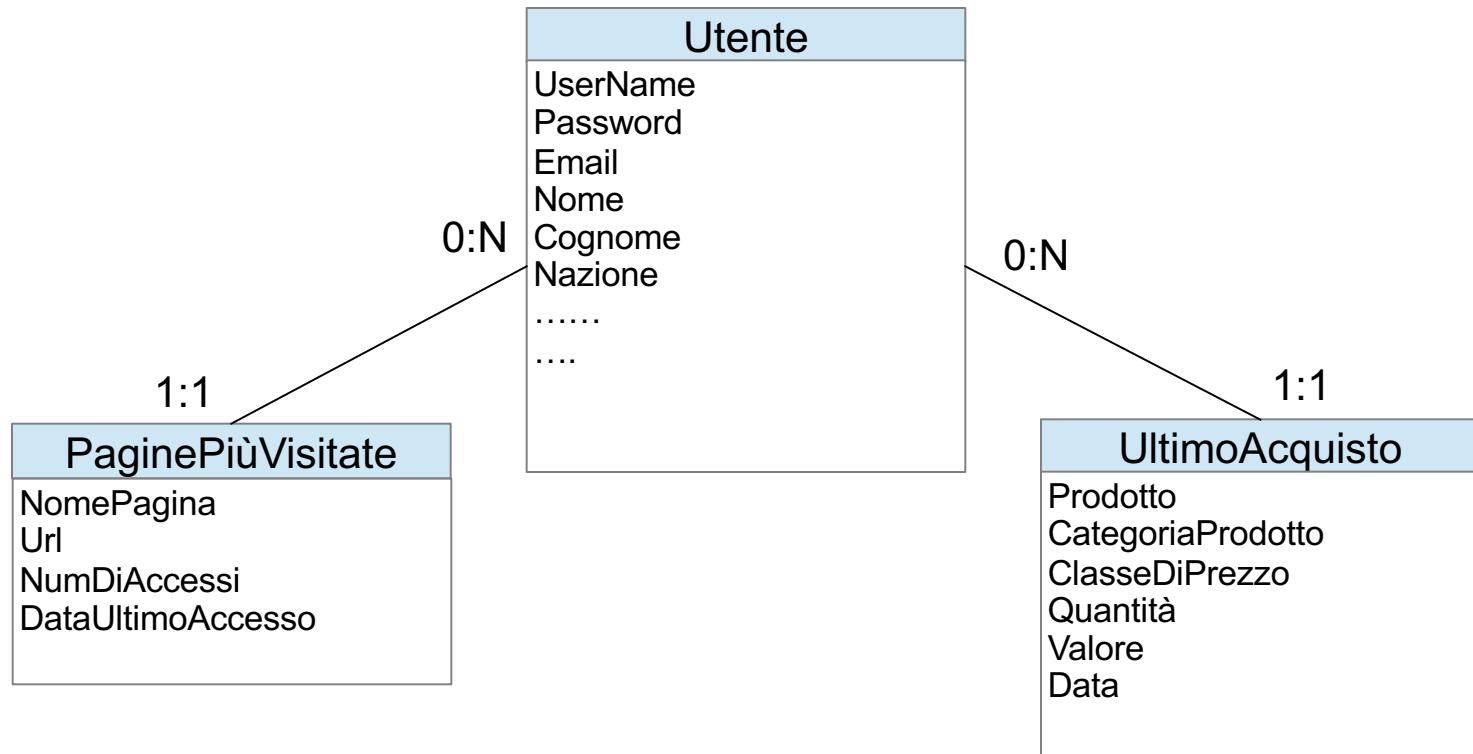
Sottoschema di «personalizzazione»

Un **sottoschema di personalizzazione** è costituito da entità e relazioni che descrivono proprietà degli utenti rilevanti per la personalizzazione dell'applicazione web. Le informazioni del sottoschema di personalizzazione comprendono:

- **dati dei profili utente**: attributi, in alcuni casi complessi, che caratterizzano utenti individuali
- **dati sui gruppi di utenti**: rappresentano classi di utenti con requisiti omogenei
- **relazioni di personalizzazione**: associazioni semantiche tra gli oggetti core/accesso e gli utenti o i gruppi, che denotano i diritti di accesso di utenti o gruppi sugli oggetti core, l'appartenenza di oggetti core a utenti o gruppi, o anche la preferenza di utenti o gruppi verso specifici oggetti core o di accesso

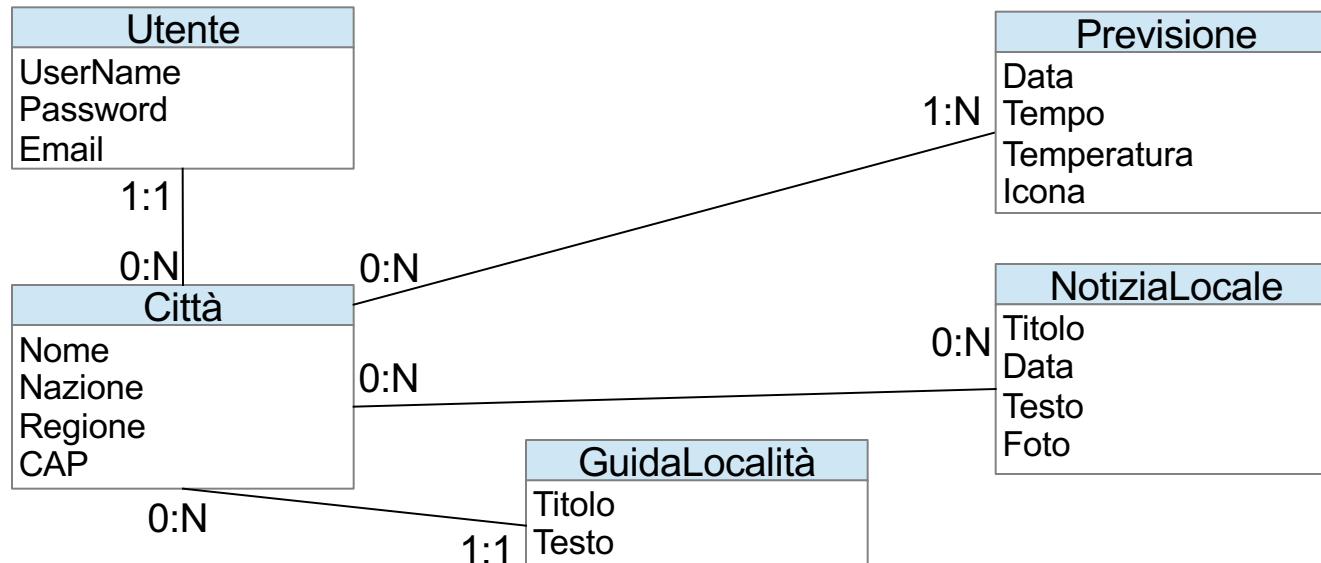
Dati di profilo dell'utente

- Lo schema precedente può essere arricchito con ulteriori elementi che rappresentano informazioni specifiche richieste da un dominio applicativo
- La figura mostra uno schema dei dati



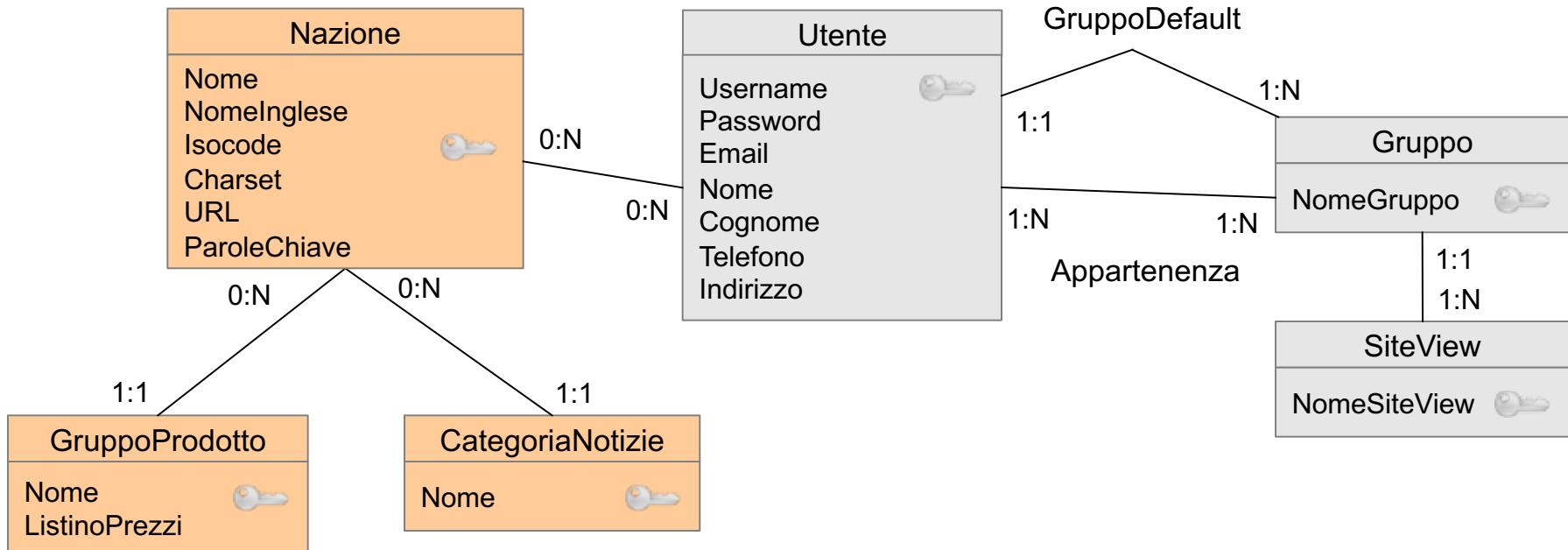
Personalizzazione con relazioni (I)

- Le **relazioni di personalizzazione** permettono di esprimere a livello dei dati le preferenze degli utenti memorizzate come valori di attributi nell'entità Utente o come istanze di una relazione tra l'entità Utente e un'entità core/accesso
- Nell'esempio vengono gestite le informazioni sulle città preferite dall'utente, le previsioni del tempo, le notizie e le descrizioni delle località da visitare

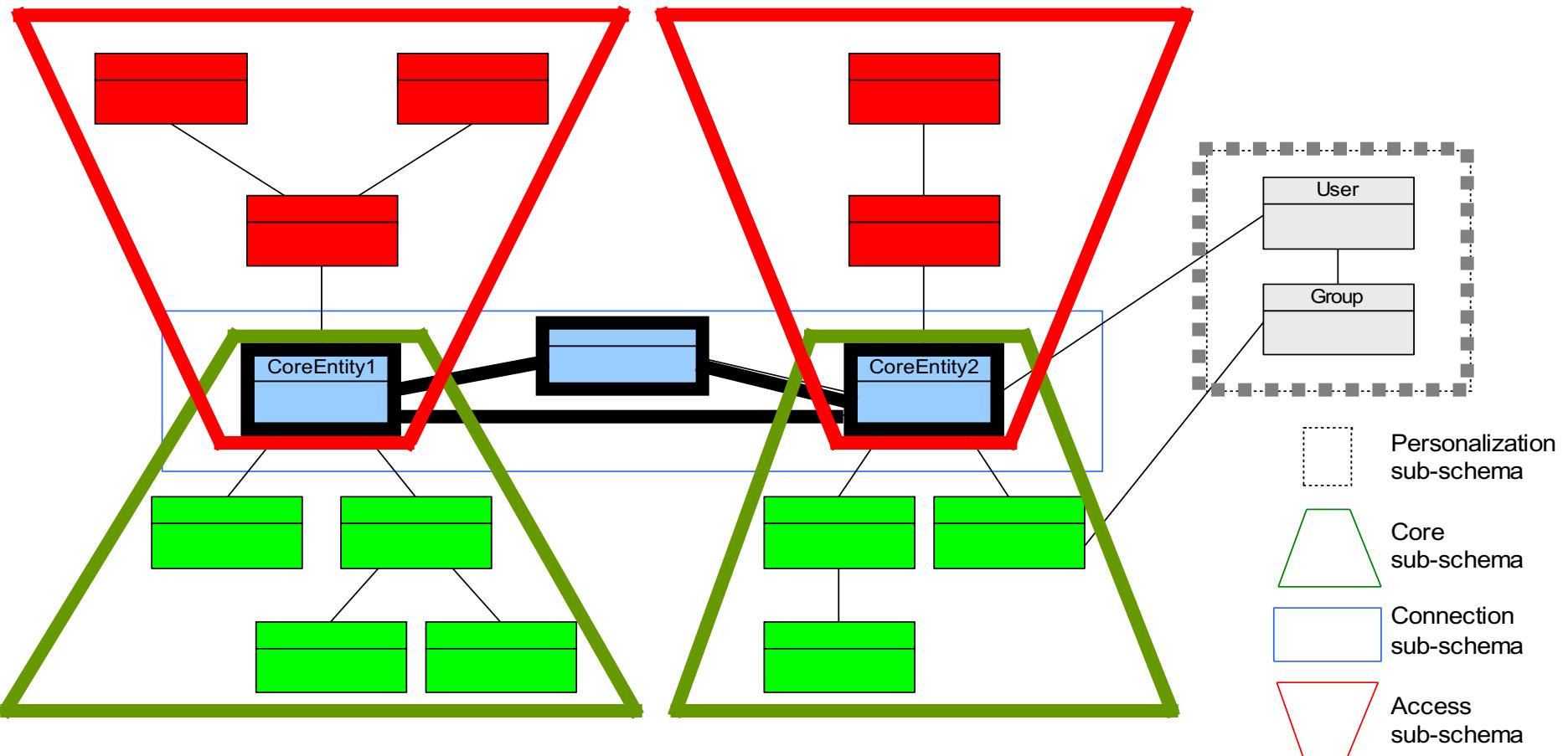


Personalizzazione con relazioni (II)

- Profili di utenti registrati e informazioni per il controllo dell'accesso



Composizione dei sottoschemi



Progettare i dati in WebML (I)

- La progettazione dei dati è un'attività incrementale e iterativa. Partendo da un nucleo iniziale, costituito dai concetti core più importanti, lo schema ER viene esteso progressivamente attraverso i seguenti passi di raffinamento:
 1. **aggiungere un nuovo sottoschema core**, o arricchire un sottoschema già esistente con la specifica dettagliata delle proprietà interne o dei componenti di un oggetto core
 2. **aggiungere un sottoschema di interconnessione**, tramite la definizione di relazioni tra entità core, per rappresentare associazioni semantiche tra concetti core



Progettare i dati in WebML (II)

3. **aggiungere un sottoschema accesso**, con l'introduzione di entità di categorizzazione e la loro connessione a entità core, oppure attraverso la specializzazione di entità core utilizzando sotto-entità che denotano collezioni speciali
4. **aggiungere un sottoschema di personalizzazione**, con l'introduzione delle entità **Utente** e **Gruppo**, la definizione delle loro proprietà e la loro connessione agli oggetti core per esprimere preferenze e appartenenze di oggetti per singoli utenti o per gruppi di utenti

Esempio di riferimento (I)

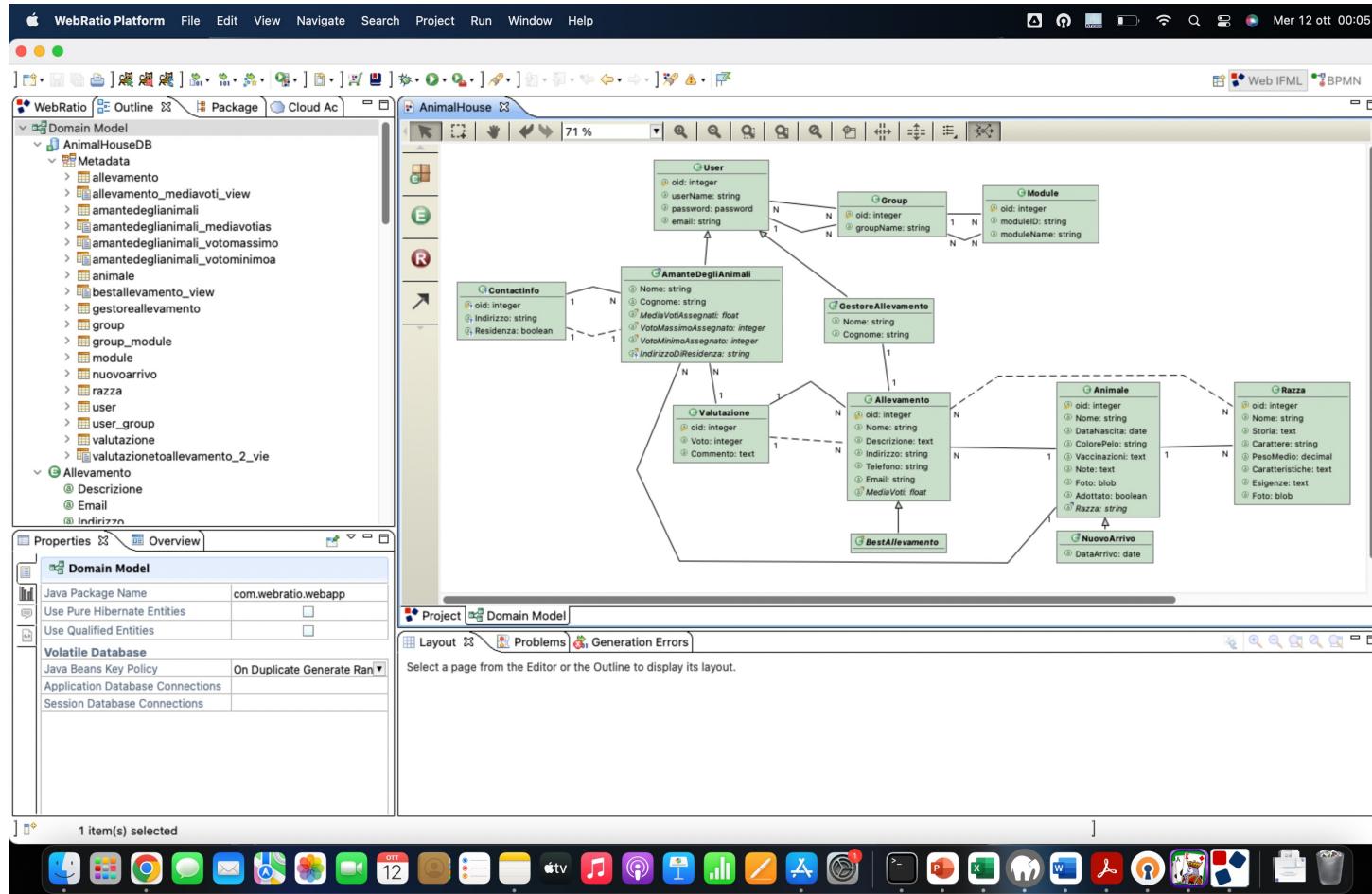
- Facendo uso di WebML, progetteremo lo schema dei dati che sottende un gestore di allevamenti per animali di razza
 - Ogni animale è descritto da un nome, una data di nascita, il colore del pelo, un testo che elenca le vaccinazioni sostenute, una breve descrizione, una foto, un flag che dica se l'animale è stato adottato o meno e una razza, utilizzata per cercare l'animale sul sito; per ciascuna razza viene registrato il nome, una breve storia, una descrizione del carattere e delle caratteristiche, il peso medio degli esemplari di quella razza, una descrizione delle esigenze tipiche della razza e una foto caratteristica
 - Tra gli animali, vengono evidenziati i nuovi arrivi, di cui si specifica anche la data di arrivo

Esempio di riferimento (II)

- Gli animali vengono custoditi in un allevamento; ogni allevamento ha un nome, una breve descrizione, un indirizzo, un numero di telefono e una email
- Gli utenti del sito, per poter adottare un animale, devono essere registrati sul sito stesso tramite nome, cognome, indirizzo, email e contatto telefonico; quando un utente adotta un animale, si tiene traccia nel database sia del fatto che l'animale è adottato, sia del riferimento alle caratteristiche dell'utente che lo ha adottato
- Gli utenti registrati (e solo quelli) possono esprimere delle valutazioni non anonime su un allevamento; ogni valutazione consta di un voto tra 1 e 10 e di un commento da parte dell'utente; la valutazione media costituisce un ulteriore attributo di ciascun allevamento

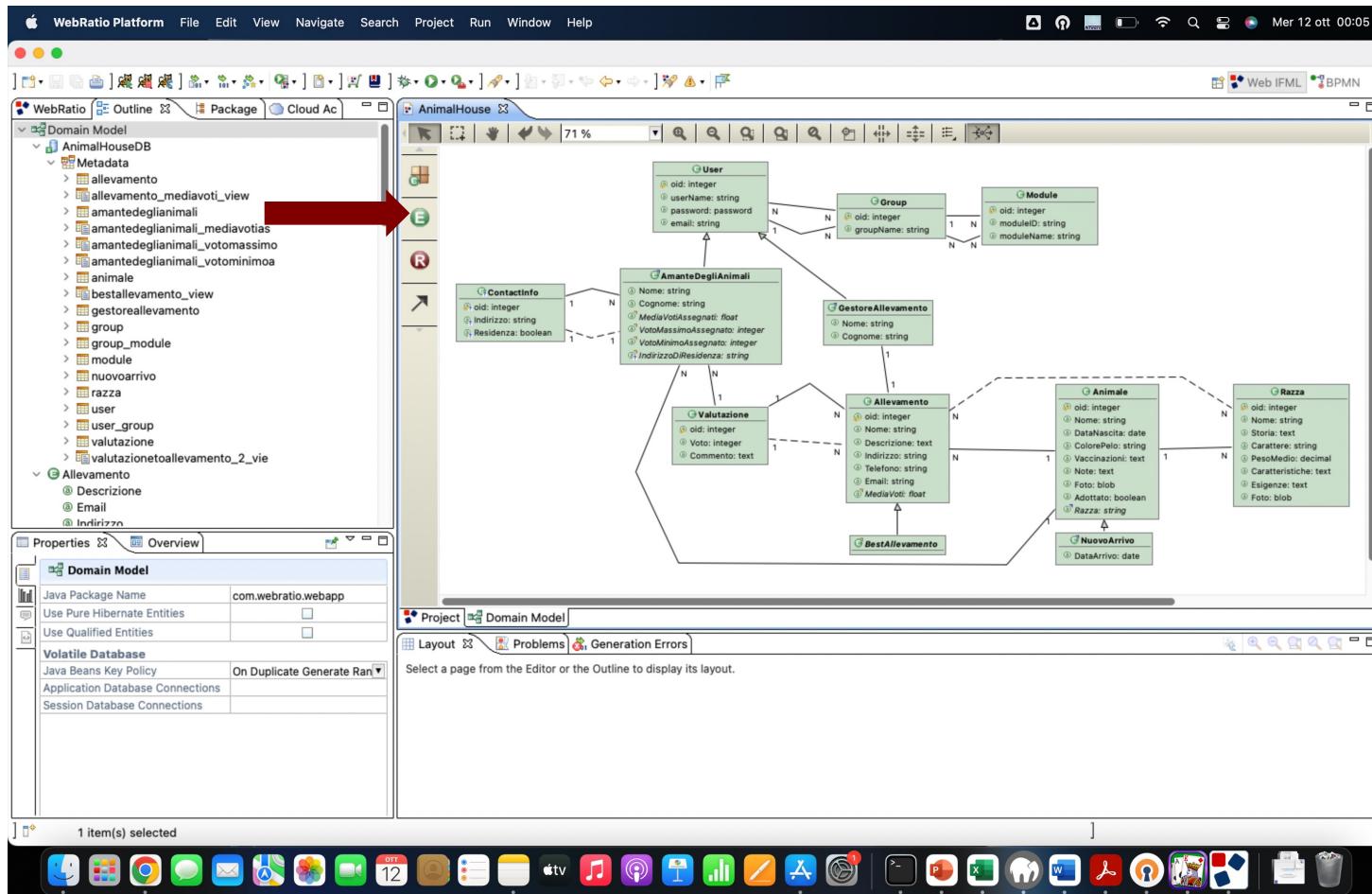


Lo strumento WebRatio





Creazione di una nuova entità



Durata di un'entità



Employee	
②	oid: integer
③	name: string
③	surname: string
③	email: string
③	role: string
③	company: string

Cart	
②	oid: integer
③	date: timestamp
③	total amount: decimal

Country	
②	oid: integer
③	name: string
③	phone prefix: string

Duration specifies the scope in which the entities instances are stored.

Duration is represented with a background color.

persistent: green background. It means that the entity's instances are stored on a persistent database.

volatile (session scope) – gray background. It means that the entity instances are stored in main memory having as scope the user session.

An example is the user shopping bag.

volatile (application scope) – violet background. It means that the entity instances are stored in main memory having as scope the application. This duration is used to represent all the information that rarely change. This information is shared between user sessions.

An example is the country list.



Aggiunta di un attributo ad un'entità

Screenshot of the WebRatio Platform interface showing the addition of an attribute to an entity in a Domain Model.

The main window displays a Domain Model diagram for "AnimalHouse". The entities shown include:

- User
- Group
- Module
- ContactInfo
- Allevamento
- Attribute
- Relationship
- Property
- Animale
- Razza
- NuovoArrivo
- DataArrivo

A context menu is open over the "ContactInfo" entity, with the "Add" option selected. A submenu shows "Attribute", "Relationship", and "Property". The "Attribute" option is highlighted.

The left sidebar shows a tree view of the project structure under "WebRatio". The "AmanteDegliAnimali" entity is selected, and its properties are displayed in the "Properties" panel:

Name	Type	Description
Id	ent7	
Name	AmanteDegliAnimali	
Super Entity	User	
Duration	persistent	
Derivation		
Attribute Order	Nome Cognome MediaVotiAssegnati Voto	

The status bar at the bottom indicates "Mer 12 ott 00:06".

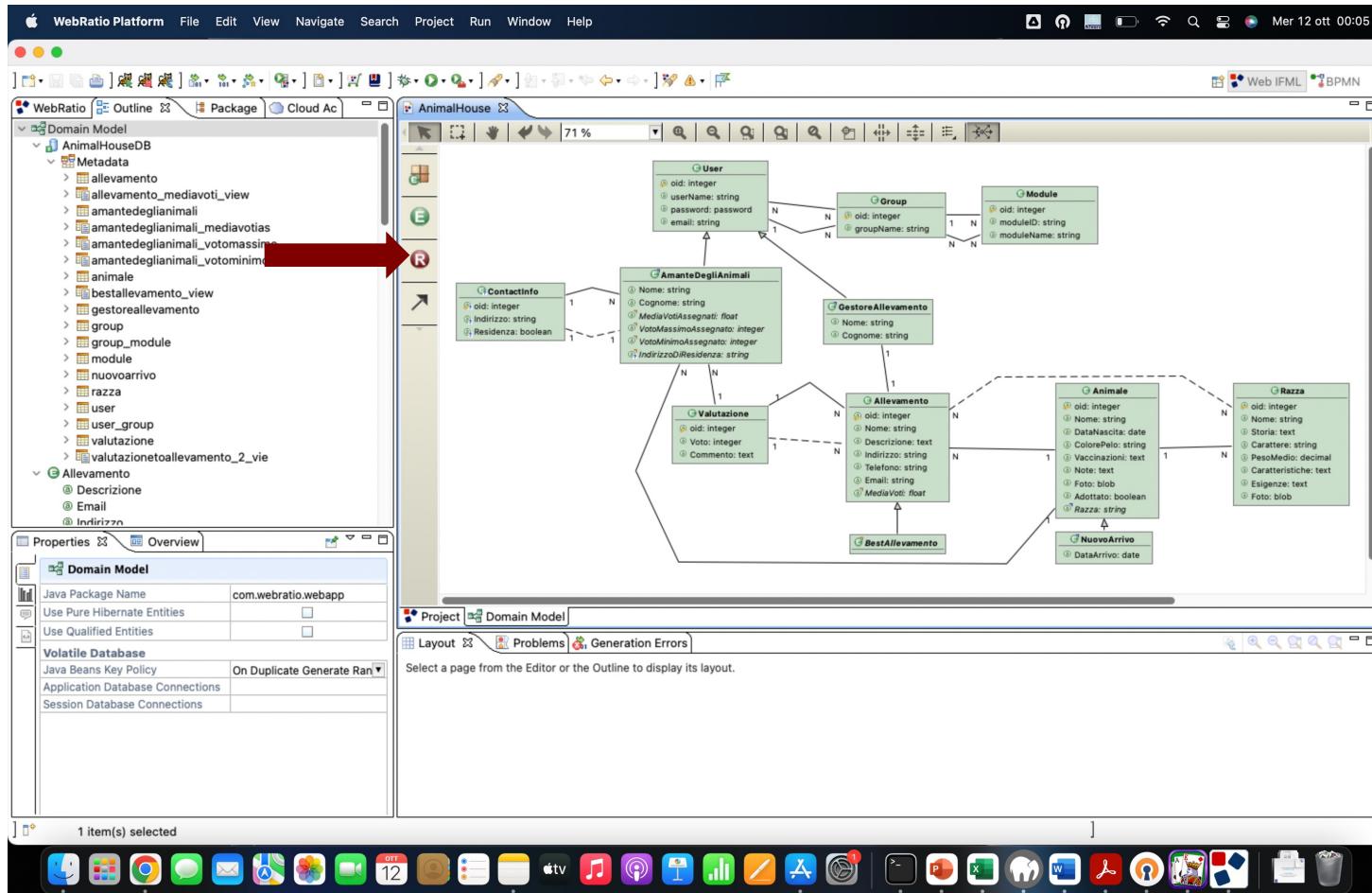
Tipi di attributo



Data Type	Description	Content Types	Pattern
blob	Represents a file stored either on the file system or on the database.	application/doc application/pdf application/x-shockwave-flash application/zip audio/mpeg image image/gif image/jpg image/png video/mpeg video/quicktime	
boolean	Represents a boolean choice.		True Value: yes False Value: no
date	Represents a date.		Pattern: M/d/yy Example: 1/15/2013
decimal	Represents a floating-point number.		Pattern: #,##0.### Example: 12,345.678
float	Represents a fixed-point number.		Pattern: #,##0.### Example: 12,345.678
integer	Represents an integer number.		Pattern: #,##0 Example: 12,345
password	Represents a string containing a password.		
string	Represents a string less than 255 characters.	text/plain text/html	
text	Represents a string longer than 255 characters.	text/plain text/html	

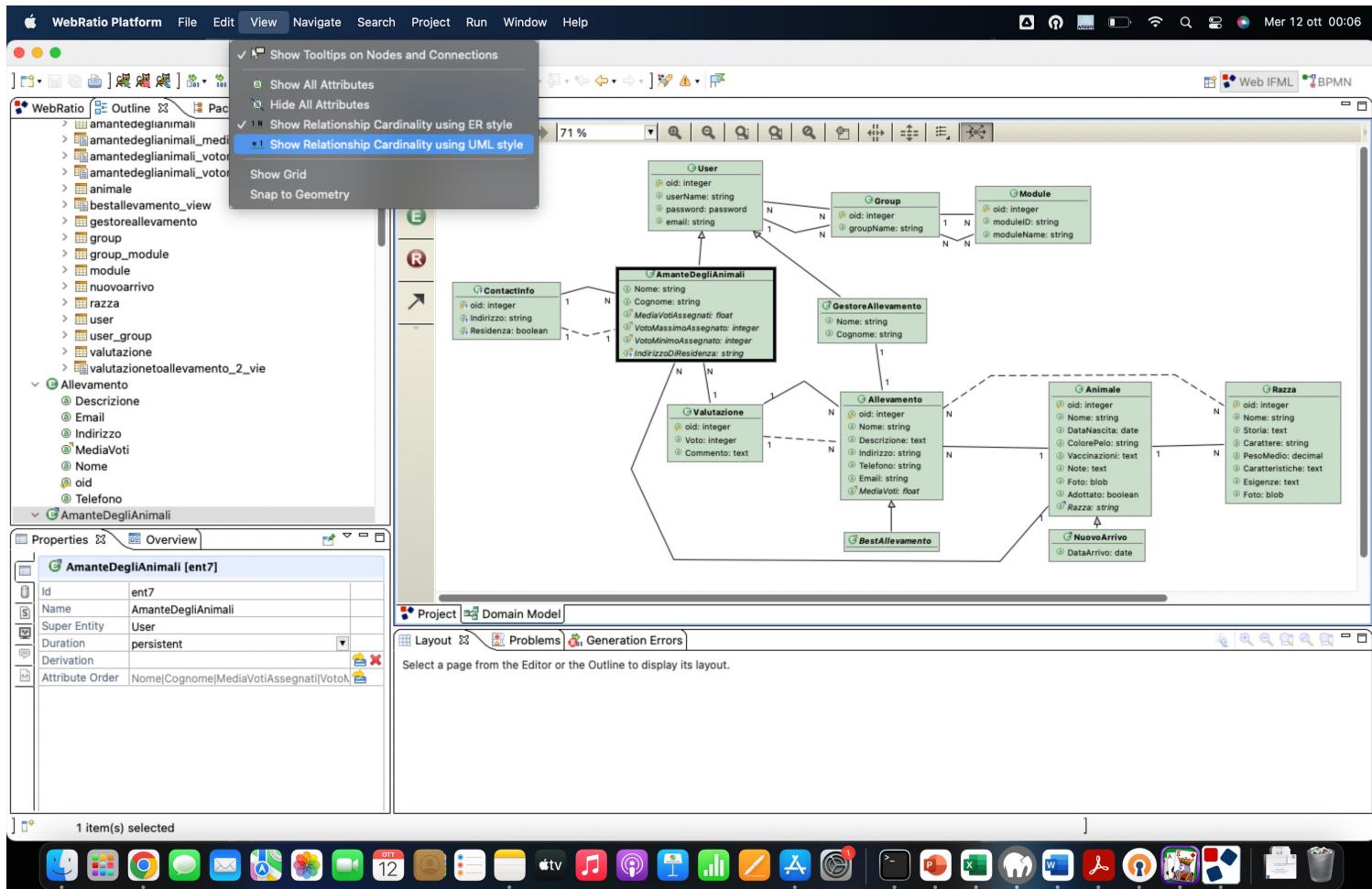


Creare una nuova relazione





ER style vs UML style



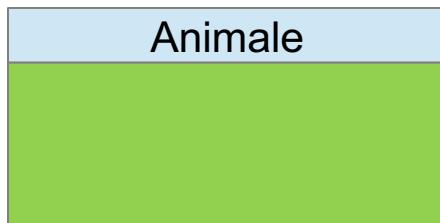
Entità «core» (esempio)

L'applicazione web Animal-House gestisce due oggetti informativi principali:

- gli animali
- gli allevamenti

Passo 1: identificazione delle entità core

- Dal dizionario dati definito durante la specifica dei requisiti estraiamo i due concetti core **Animale** e **Allevamento** che vengono inseriti nello schema dei dati come entità.



Sottoschemi «core» (esempio)

Passo 2: progettazione dei sotto-schemi core

- L'entità **Animale** è caratterizzata da proprietà semplici (Nome, DataNascita, ColorePelo, etc) tutte aventi un singolo valore. Tutte le proprietà vengono rappresentate come **attributi** dell'entità Animale
- L'entità **Allevamento** possiede proprietà semplici e complesse. Le proprietà semplici (Nome, Descrizione, etc) vengono modellate come attributi. Le proprietà multi-valore sono modellate come **componenti interni**:
 - **Valutazione**: valutazione assegnata da un utente all'allevamento, a sua volta descritta da un voto e un commento
- I componenti interni sono connessi ad Allevamento con relazioni uno-a-molti in quanto sotto-parti del concetto Allevamento. Una Valutazione non può esistere indipendentemente dall'Allevamento e non può essere associata a più di un Allevamento



Sottoschemi «core» (esempio)





Sottoschemi di «interconnessione» (esempio)

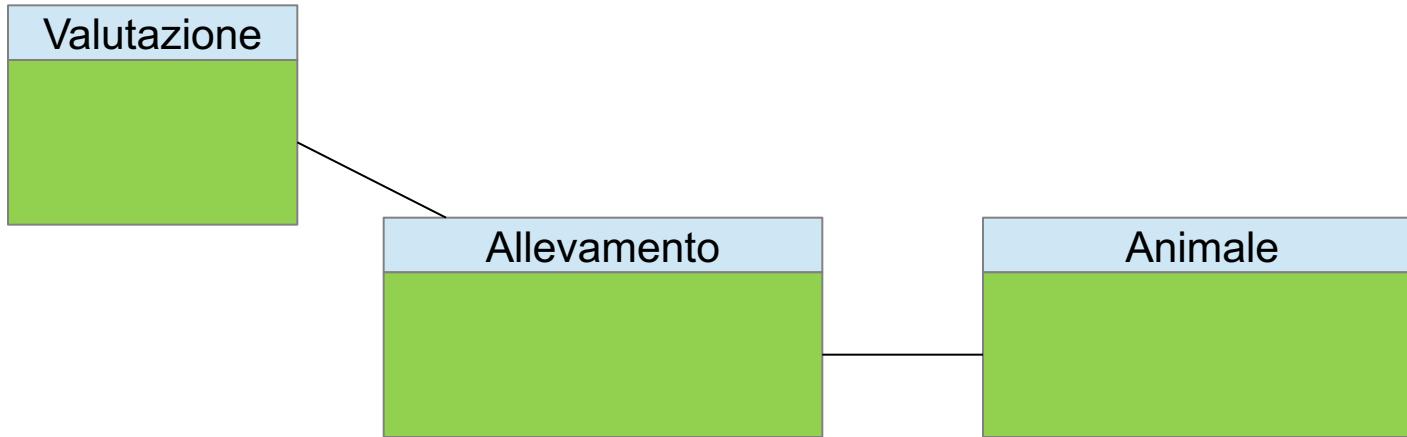
Passo 4: progettazione dei sottoschemi di interconnessione

Vengono introdotte le connessioni tra i concetti core:

- Gli Animali vengono collegata agli Allevamenti in cui sono custoditi
- Un Allevamento è associato ad uno o più Animali, in esso custoditi
- Viene quindi introdotta una relazione uno-a-molti tra Animale e Allevamento



Sottoschemi di «interconnessione» (esempio)



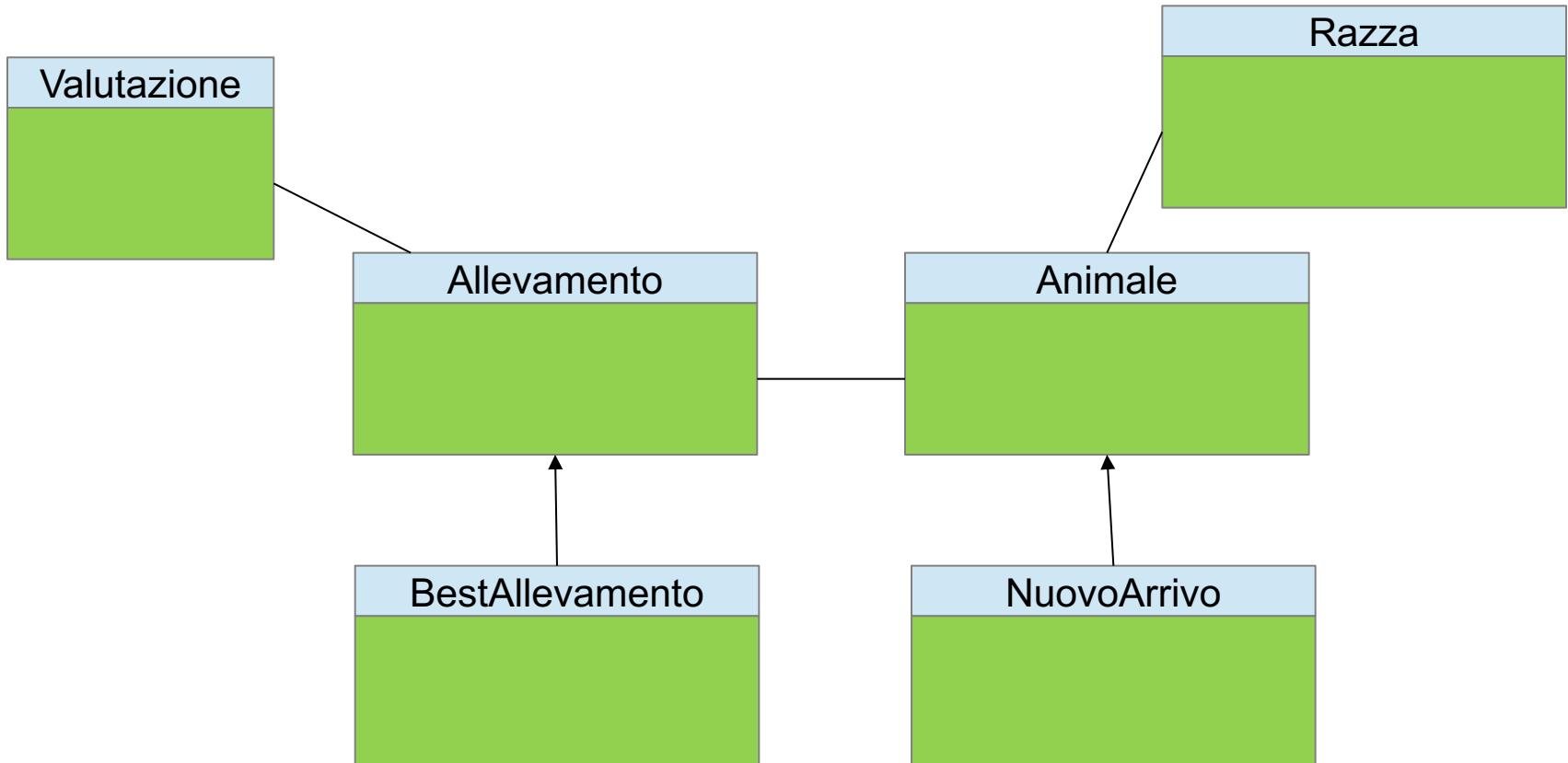


Sottoschemi di «accesso» (esempio)

Passo 3: progettazione dei sottoschemi di accesso

- A partire dai casi d'uso per gli utenti esterni si deduce che l'accesso alle informazioni sugli animali deve essere supportato da una categorizzazione, si introduce quindi l'entità **Razza** che viene associata ad Animale tramite una relazione.
- Si introduce poi la sotto-entità **NuovoArrivo** che specializza l'entità Animale per mettere in evidenza i nuovi arrivi, per esempio da presentare nella Home page.
- Similmente, la sotto-entità **BestAllevamento**, ottenuta come entità derivata per mettere in evidenza i migliori allevamenti, è introdotta come sotto-schema di accesso per Allevamento

Sottoschemi di «accesso» (esempio)



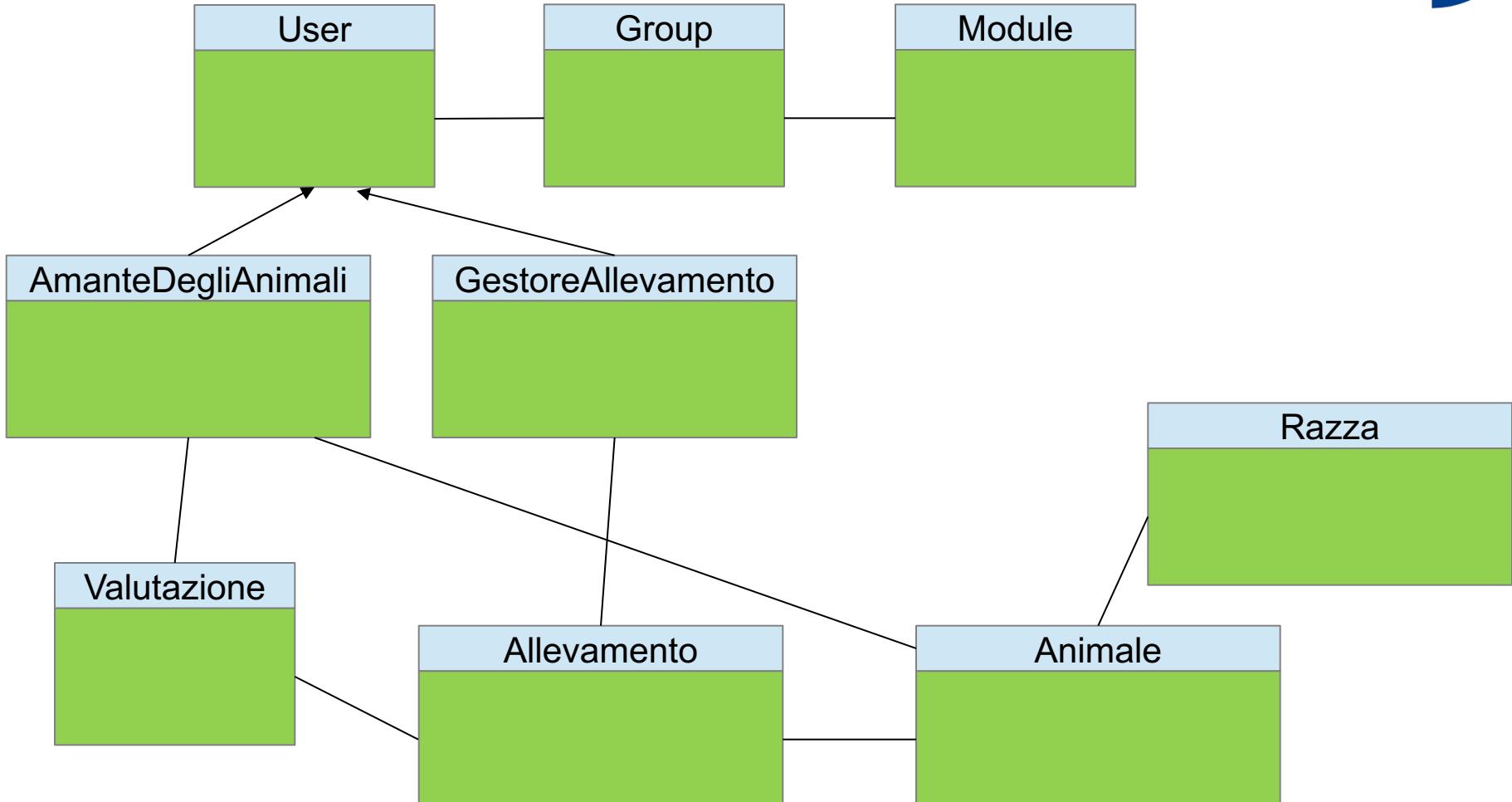


Sottoschemi di «personalizzazione» (esempio)

Passo 5: progettazione del sottoschema di personalizzazione

- La personalizzazione dell'applicazione deve essere fatta solo per gli utenti registrati
- Si aggiungono le entità:
 - **Utente**: sono gli utenti registrati
 - **Gruppo**: sono i gruppi di utenti identificati
 - **SiteView**: sono le site view definite per gli utenti più quella definita per gli utenti esterni

Sottoschemi di «personalizzazione» (esempio)



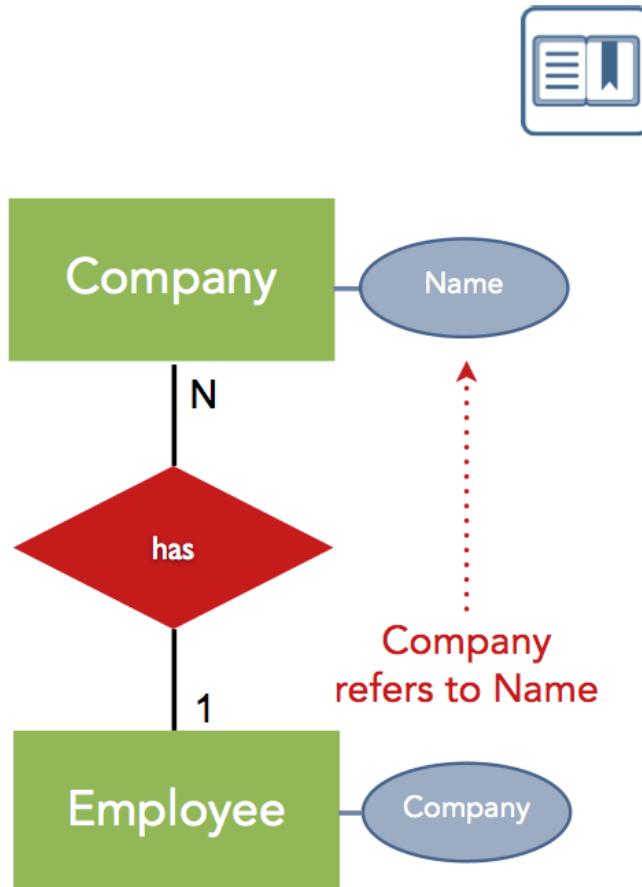


Estendere lo schema dei dati

- **Derivazione:** creazione di un nuovo elemento a partire da altri elementi del Domain Model
- WebRatio mette a disposizione un **Derivation Wizard**
- È possibile derivare entità, attributi e relazioni
 - attributi: *simple imported, complex imported, calculated*
 - entità: per *specializzazione*
 - relazioni: per *restrizione, concatenazione*



Attributi derivati: simple imported

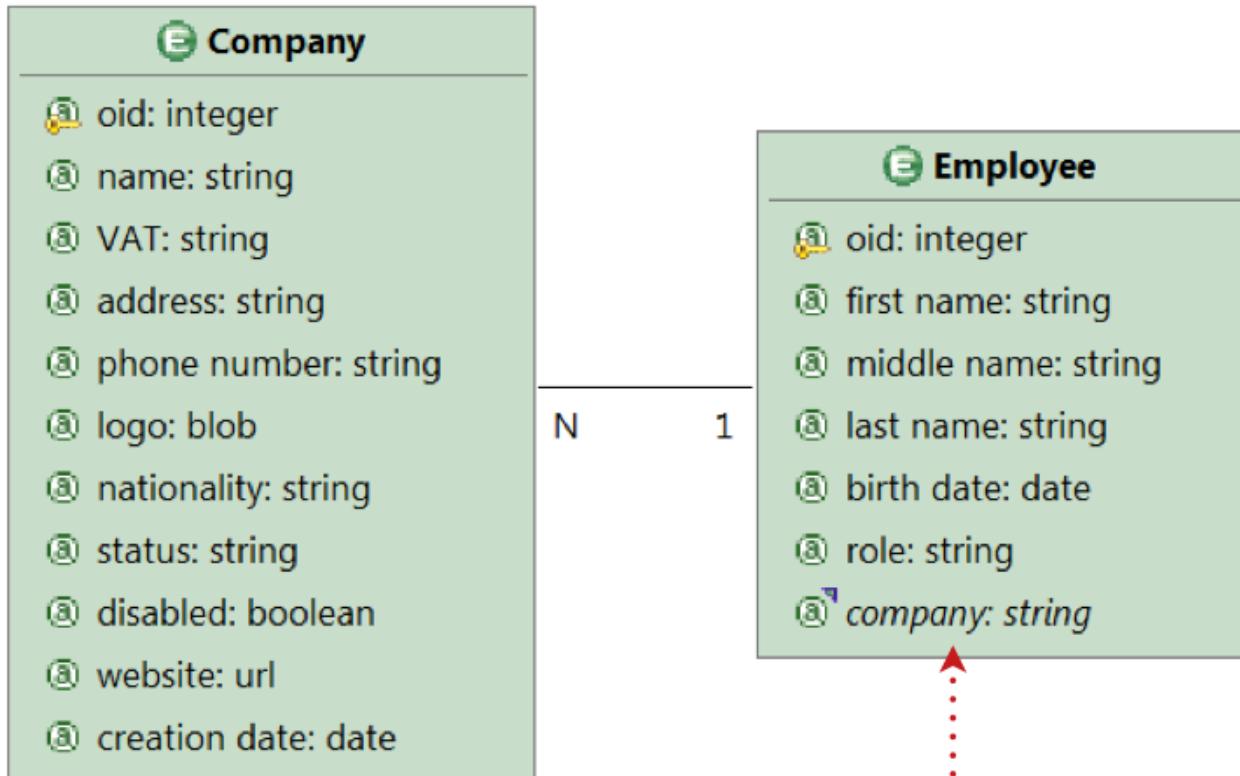


The Simple Imported Attribute defines an entity property referring to the value of another attribute of a different entity.

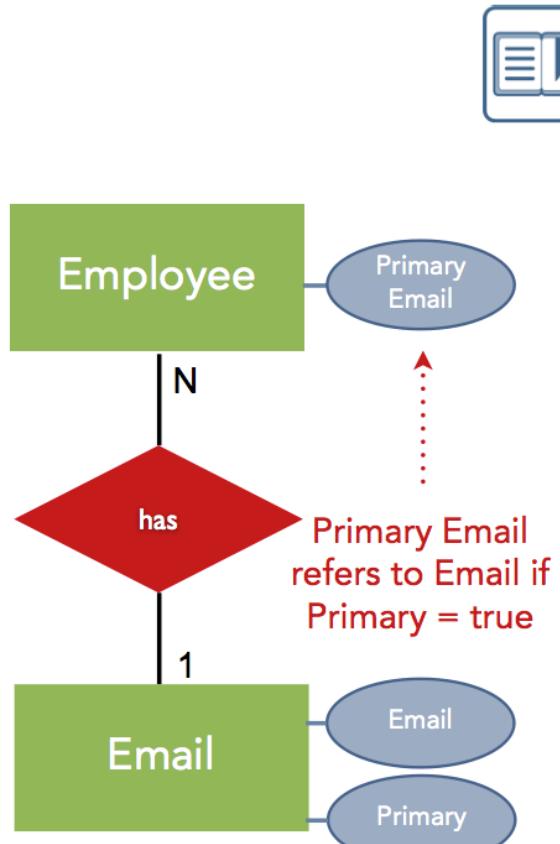
You can refer values only to attributes belonging to entities connected to the interested one with a 1-N relationship.

In particular, the 1 cardinality has to belong to the entity on which you are specifying the derived attribute.

Esempio



Attributi derivati: complex imported

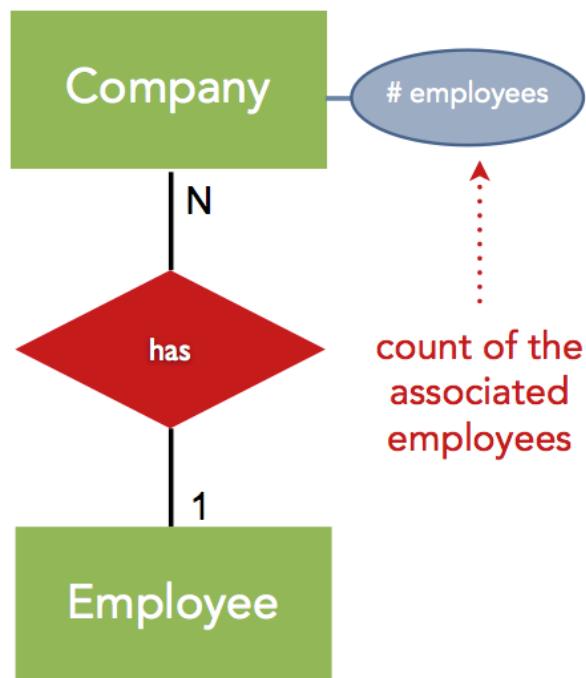


 The Complex Imported Attribute defines an entity property referring to the value of another attribute of a different entity only when a condition is satisfied.

In a CRM Web application an example is the employee '*primary email*'.

This attribute is imported from the connected Email entity only when its '*primary*' attribute is "true".

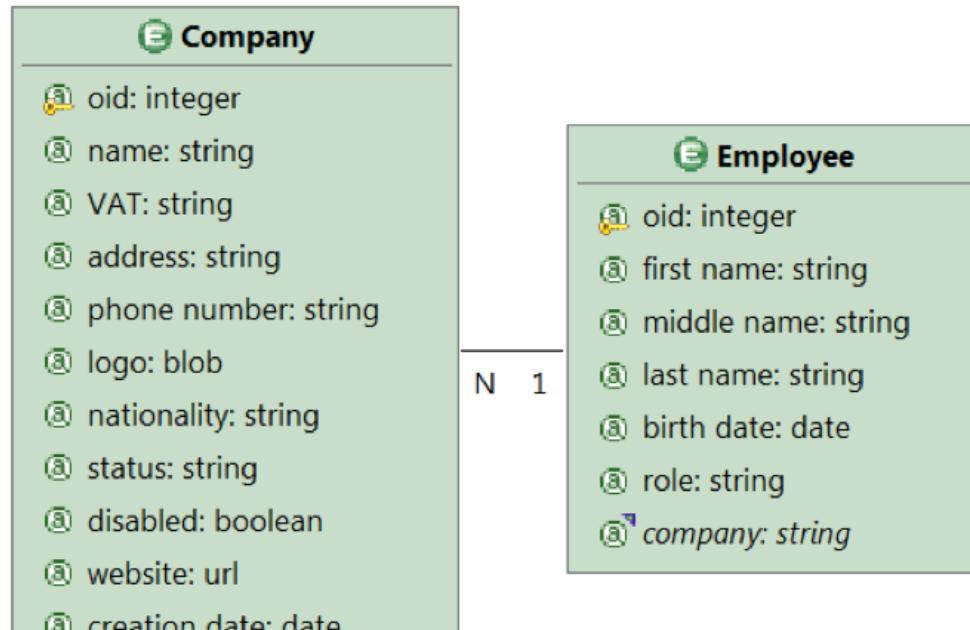
Attributi derivati: calculated



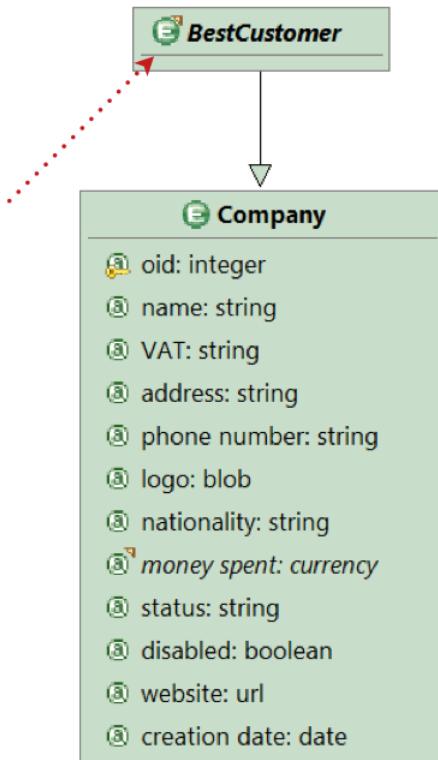
A calculated attribute defines an entity property through a calculation using other attributes or through an aggregation of other properties.

In a CRM Web application an example is the '*number of employees*' registered for a company.

Esempio



Entità derivate

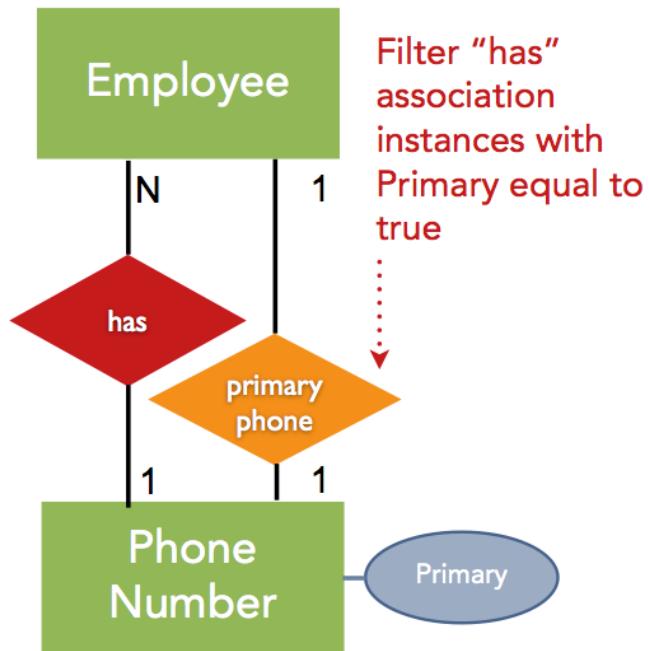


A derived entity is an abstract concepts that represents a subclass of objects having a specific property.

Usually a derived entity filters the records of another entity (named master entity) using some **criteria**.

In a CRM web application, an example of derived entity is '*Best Customer*' which contains companies who spent a lot of money purchasing your products..

Relazioni derivate: per restrizione

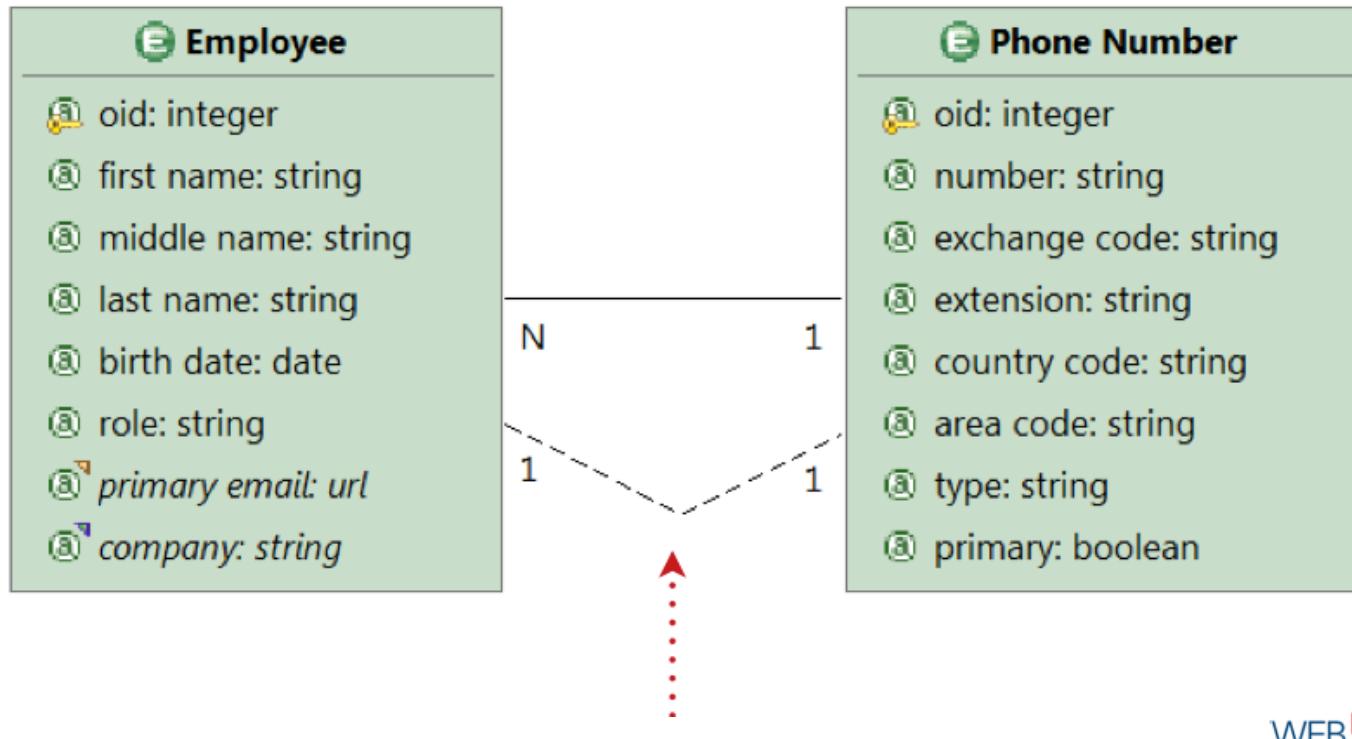


The derived relationship is defined specifying some **conditions** of an existing relationship. The connected records are a subset of the master relationship.

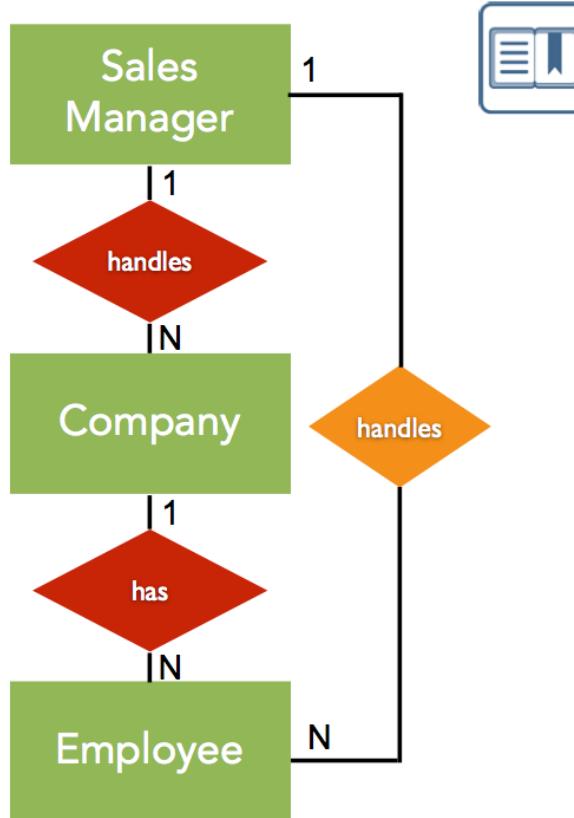
In a CRM web application, an example of derived relationship by restriction is '*Primary Phone Number*' which connects the employee with his primary phone number.

The idea is to have a shortcut to the primary phone number.

Esempio



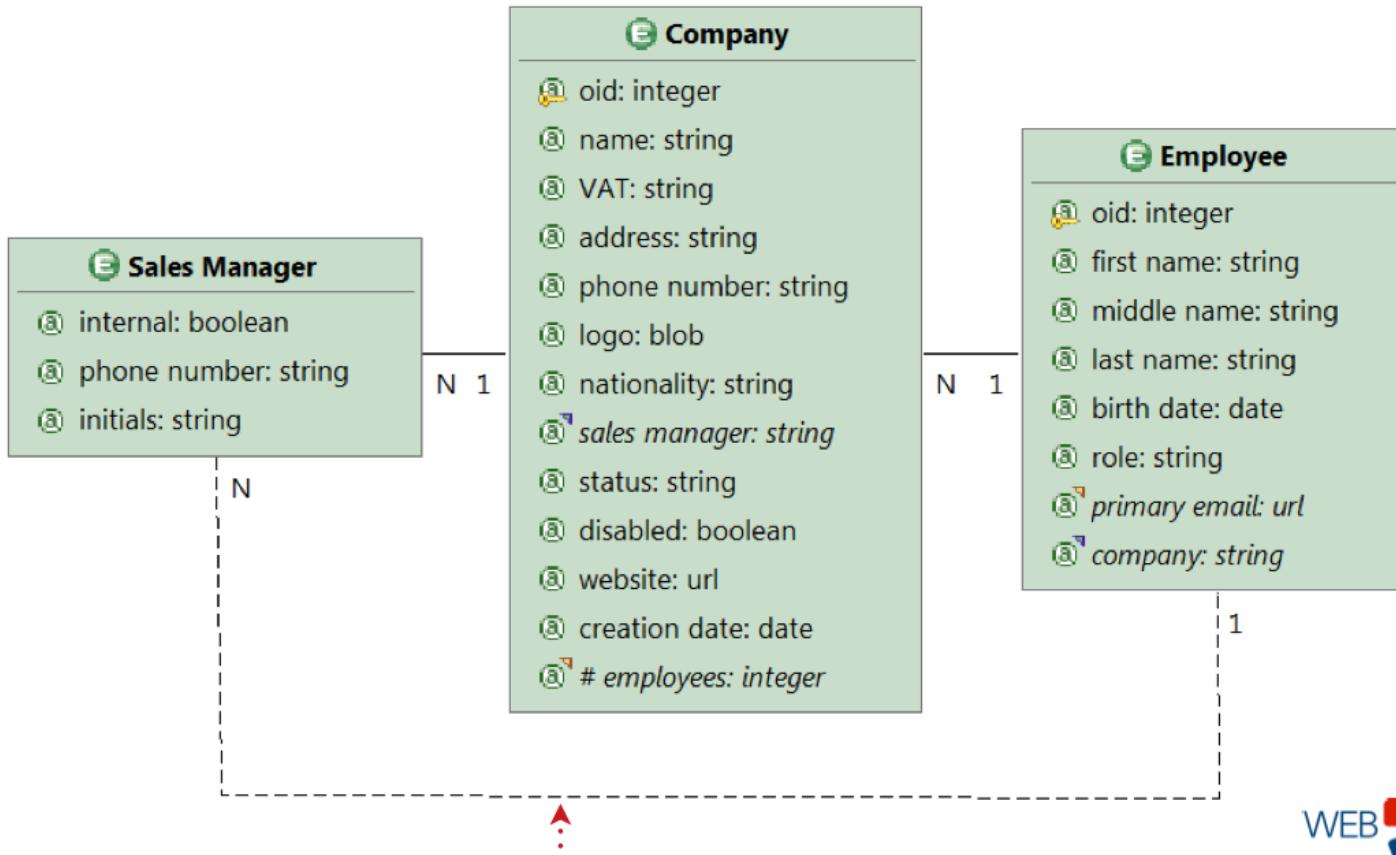
Relazioni derivate: per concatenazione



The derived relationship is defined as **composing** existing relationships.

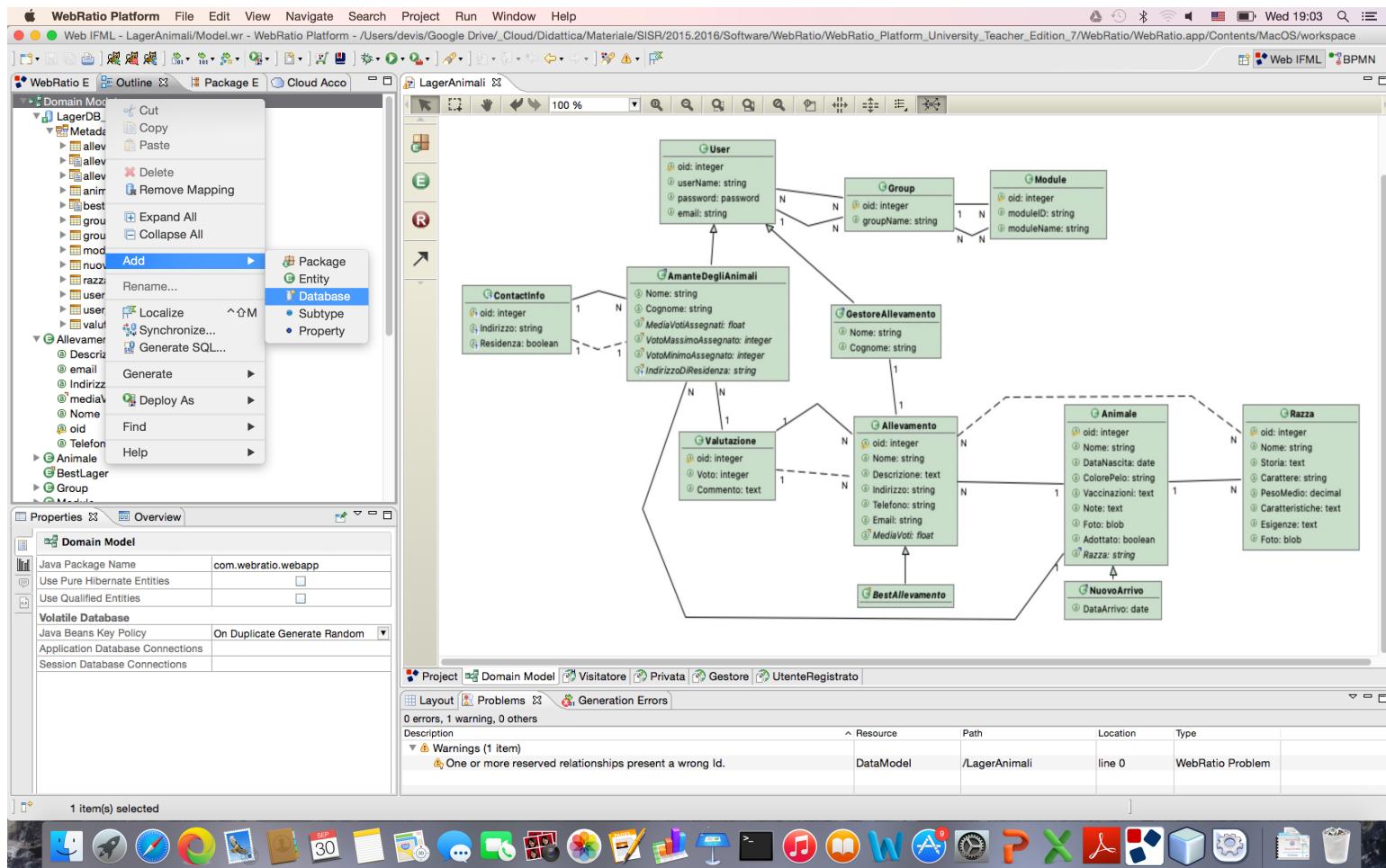
In a CRM web application, an example of derived relationship by concatenation is '*Employee to Sales Manager*' which connects the employee with the sales manager of his company.

Esempio

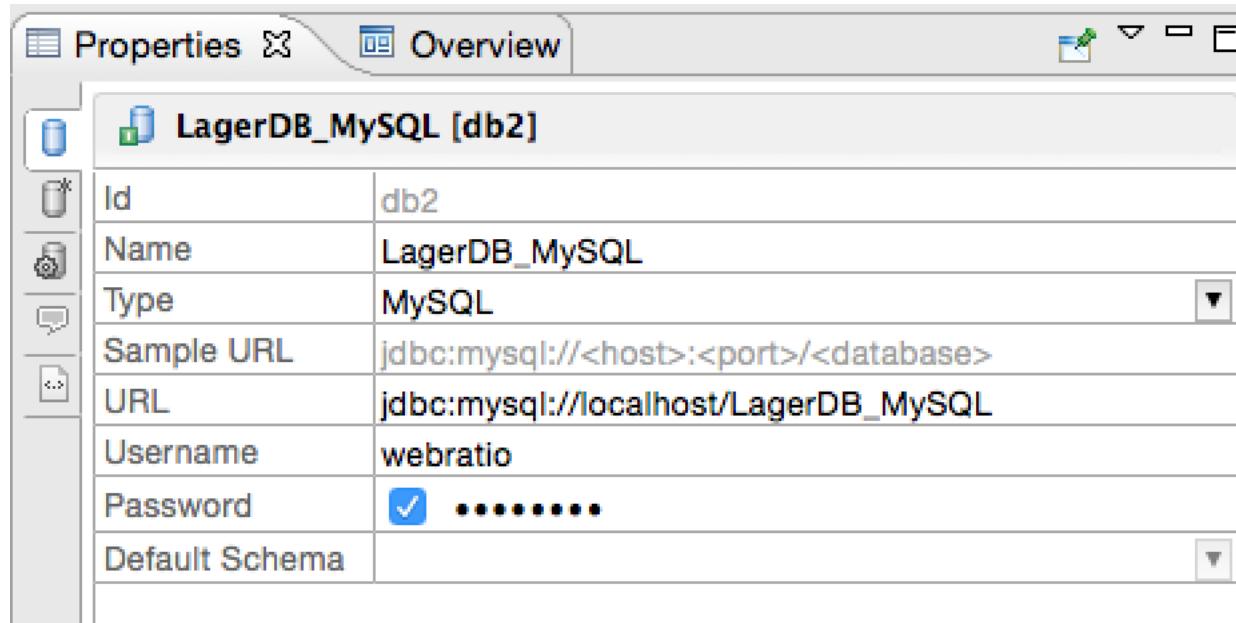




Setup del DBMS (I)



Setup del DBMS (II)

A screenshot of a software interface for managing database connections. The title bar shows "Properties" and "Overview". The main area displays a connection named "LagerDB(MySQL) [db2]". The properties listed are: Id (db2), Name (LagerDB(MySQL)), Type (MySQL), Sample URL (jdbc:mysql://<host>:<port>/<database>), URL (jdbc:mysql://localhost/LagerDB(MySQL)), Username (webratio), Password (checkmark and masked dots), and Default Schema (empty). On the left, there is a vertical toolbar with icons for creating, deleting, and modifying connections.