Analisi e riconciliazione delle fonti dati e Strumenti ETL

DWDM – 2004/2005
A.GORI

Libri di Testo

- "DATA WAREHOUSE", *Golfarelli e Rizzi*, Mc Graw-Hill, 2002

Gli strumenti ETL

- Il processo di alimentazione del DW avviene:
  - All'inizializzazione del sistema
  - All'aggiornamento del sistema

  Consiste di tre fasi:
  1. Estrazione
  2. Pulitura / Trasformazione
  3. Caricamento

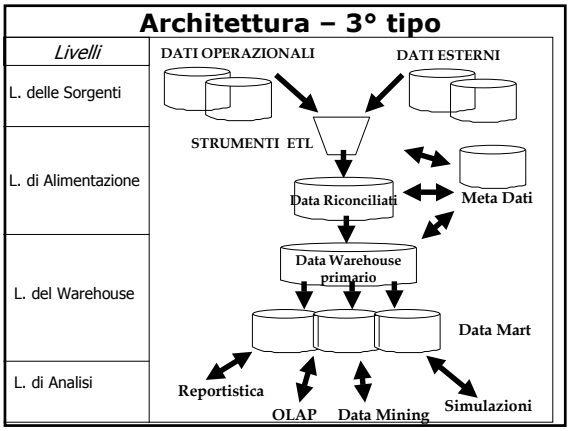
PRODUZIONE DI DATI RICONCILIATI

O

RICONCILIAZIONE

- In caso di architettura a tre livelli questi strumenti alimentano il livello dei dati riconciliati

Architettura – 3° tipo



Soluzione a tre livelli

- Preferibile perché l'alimentazione diretta del DW è compito troppo complesso per essere eseguito in modo atomico
- Presenza di uno stadio intermedio
  - Facilita la progettazione
  - Semplifica l'alimentazione e la pulizia dei dati
  - Schema concettuale integrato ben gestibile

Architettura – Modello dei dati

- **Esterno**:
  rappresenta come gli utenti e/o le applicazioni vedono il database; l'insieme delle viste utente…
- **Concettuale**:
  rappresenta la definizione degli oggetti del DB (tabelle, viste, indici,…); è una definizione del database (dati, relazioni) che non si preoccupa dei requisiti delle singole applicazioni o dei limiti fisici dei dispositivi di memorizzazione
- **Interno**:
  rappresenta la definizione della struttura fisica del DB (campi, record); tratta l'allocazione dei dati, i meccanismi di accesso per ricercare e manipolare i dati.

Fonti dati

- Insiemi di sorgenti possono differire per:
  - Tecnologia
    - DBMS, Fogli elettronici, File system, ecc.
  - Modello di rappresentazione
    - Relazionale, Ad Oggetti, …
  - Semantica

Fonti dati

- Dall'analisi delle fonti operazionali possono scaturire:
  - Inconsistenze
  - Errori
  - Domini Sovrapposti
- ESIGENZA: alimentare il dw con dati consistenti e privi di errori

Principio base - Obiettivo

- Uno dei principi fondamentali del data warehousing è il concetto di DATO INTEGRATO che permette di derivare informazioni consistenti e prive di errori.
- Necessità di una fase di "Analisi delle fonti operazionali"

Proposta

- Necessità di un processo di riconciliazione che comporta:
  - Pulizia
  - Trasformazione dei dati } → dati veri e propri
  - Integrazione → schemi
- Processo che può richiedere un elevato dispendio di tempo e di risorse

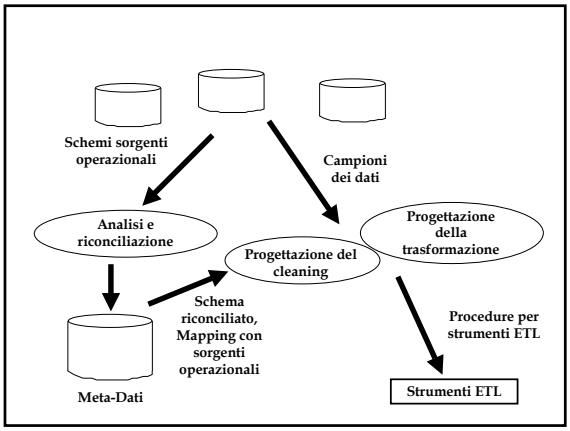
Due ritmi diversi …su un DW

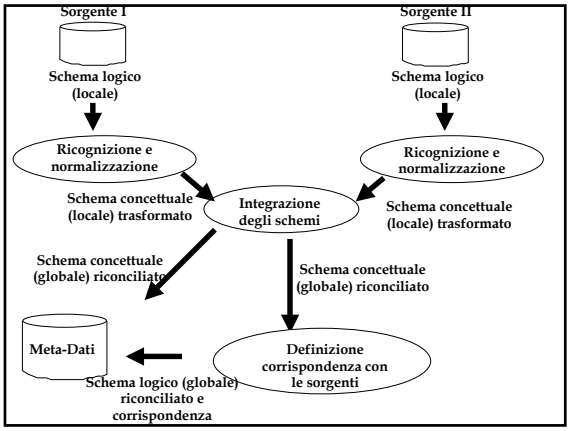
- Uso bimodale:
  - 16-22 ore al giorno usati per attività di interrogazione
    - funzionalità front room
  - 2-8 ore al giorno per caricamento, indicizzazione, controllo qualità e pubblicazione
    - funzionalità back room

Fasi

- Ricognizione
- Normalizzazione

- Integrazione
- Pulizia
- Trasformazione





- Ricognizione:
  - Esame approfondito degli schemi locali
- Normalizzazione:
  - Correzione/standardizzazione degli schemi locali

Il problema della eterogeneità

Eterogeneità

- L'eterogeneità occorre spesso quando vi è la necessità di integrare e rendere interoperabili sistemi che sono preesistenti e che sono nati per vivere autonomamente
- Esempi:
  - Uso di DBMS differenti (spesso diversi anche come modello dei dati)
  - Ogni DBMS ha le proprie caratteristiche, il proprio schema locale
  - .....

Eterogeneità

- L'eterogeneità quindi può manifestarsi a vari livelli:
  - Piattaforme hardware diverse
  - Sistemi operativi diversi
  - Protocolli di rete diversi
  - Applicazioni scritte in diversi linguaggi
    - il software che crea e manipola i dati nei vari siti è diverso
  - Diversi linguaggi di interrogazione
  - Diversi DBMS – diversi modelli dei dati
    - il formato e la struttura dei dati nei vari siti sono diversi
  - Diversità semantiche

INTERESSATI A

Heterogeneity

Database

DBMSs
Semantic Heterogeneity

Operating System

File Systems
Interprocess communication
Level of user access
Distributed processing capabilities

Communication

Network topology
Networking protocols
Physical layer differences

Hardware/System

Instruction Set
Data formats and representation
…

Eterogeneità Semantica

Esempi

Parentesi

Esempio:

- Uso dell'operatore di **UNION**
- Creazione di una tabella a partire da una vista

Select * into tabellanuova
from tabellavecchia ……

Semantic Heterogeneity Omonimi

Esempio:

- Supponiamo di avere due sorgenti dati:
  - DB-A e DB-B
- Nei Db esiste una relazione RESTAURANT.
- L'attributo MEAL_COST in DB-A descrive il costo medio di un pasto per persona *without* tax.
- L'attributo MEAL_COST in DB-B descrive il costo medio di un pasto per persona *including* tax.

Semantic Heterogeneity Omonimi

- Trovare tutti i ristoranti con un "price range" $10-$20.

```
SELECT NAME
FROM RESTAURANTS
WHERE MEAL_COST>=10
AND MEAL_COST<=20
```

⇨

La query produce risultati di significato diversi.

Semantic Heterogeneity Sinonimi

ESEMPIO:

- I ristoranti sono classificati in base al tipo di piatti che servono.
- Nel DB-A questa categoria è gestita nell'attributo TYPE.
- Nel DB-B questa categoria è gestita nell'attributo CLASSIFICATION.
- Trovare tutti i ristoranti grechi:

```
SELECT NAME                SELECT NAME
FROM RESTAURANTS           FROM RESTAURANTS
WHERE TYPE="Greek"         WHERE CLASSIFICATION="Greek"
```

Semantic Heterogeneity
Granularità

- I ristoranti sono classificati in base alla qualità dei vini serviti.
- In entrambi i DB la categoria è memorizzata nell'attributo QUALITY.
- In DB-A, la qualità è data su una scala di tre valori:

  eccellente, buono, scadente.
- IN DB-B, la qualità è data su una scala di 5 valori: *1, 2, 3, 4, 5*.

Semantic Heterogeneity
Granularità

- Ristora1(ID, Nome, Classe)
- Ristora2(ID, Ristorante, Classificazione)
- Dom(Classe): { 1,2,3,4,5 }
- Dom(Classificazione): { *eccellente, buono, scadente* }

- *Ristoranti (ID, NomeRistorante, Valore)*
- *DOM(Valore)={1,2,3,4,5}*
  - Classificazione={
    - ECCELLENTE → 5
    - BUONO → 3
    - SCADENTE → 1}

+ **Il problema dell'ID**

Semantic Heterogeneity
Granularità

ESERCIZIO

- Produrre la lista completa dei ristoranti
- Produrre la classifica dei ristoranti in base alla qualità del vino.
- Possiamo fare una classifica separata per DB e poi produrne una unica?

Heterogeneity
Differenze Strutturali

- L'indirizzo di ogni ristorante è memorizzato nel DB.
- In DB-A, l'indirizzo completo è memorizzato nell' attributo ADDRESS della relazione RESTAURANTS.
- In DB-B, l'indirizzo appare in una relazione separata: ADDRESS (NAME, ADDRESS, STATE, ZIPCODE).
- Esempio:
  Trovare tutti i ristoranti della zona con zipcode 08851

Heterogeneity
Domini differenti

- Nel DB-A lo zipcode può essere letto come una sottostringa nell'attributo ADDRESS.
- Nel DB-B lo zipcode è memorizzato come dato autonomo.

Eterogeneità di rappresentazione

- Date due Rappresentazioni R1 e R2 di uno stesso concetto, sono quattro le possibili relazioni esistenti tra di esse:

1. IDENTITA'
2. EQUIVALENZA → conflitto
3. COMPARABILITA' → conflitto
4. INCOMPATIBILITA' → conflitto

Identità

- Sono utilizzati gli stessi costrutti
- Concetti modellati dal medesimo punto di vista
- No errori di specifica

→ R1 e R2 coincidono

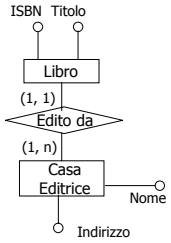
Equivalenza

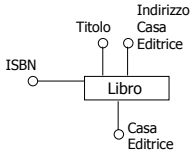
- Non sono utilizzati gli stessi costrutti (ma equivalenti)
- Due insiemi di dati diversi ma equivalenti che memorizzano le stesse informazioni

DEF. Equivalenza tra schemi:

Due schemi R1 e R2 sono equivalenti se le loro istanze possono essere messe in corrispondenza uno a uno

Rappresentazioni equivalenti





Legame Libri e Case editrici

LIBRO

ISBN	Titolo	Casa Edit.
123445	XXXX	McGraw
4354543	Yyyyy	Hoepli
123456	Zzzzzz	Fabbri
.....	.....	.....

CASA EDITRICE

Casa Edit.	Indirizzo C.E.
McGraw	Indiri1
Hoepli	Indiri2
Fabbri	Indiri3
.....	.....

LIBRO

ISBN	Titolo	Casa Edit.	Indirizzo C.E.
123445	XXXX	McGraw	Indiri1
4354543	Yyyyy	Hoepli	Indiri2
123456	Zzzzzz	Fabbri	Indiri3
.....	.....	.....	.....

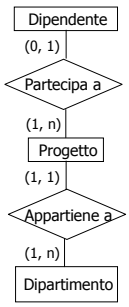
Equivalenza tra schemi

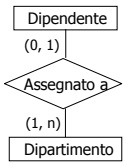
- Due schemi R1 e R2 sono equivalenti se le loro istanze possono essere messe in corrispondenza uno-a-uno

Comparabilita'

- R1 e R2 non sono né identici né equivalenti
- I costrutti non sono in contrasto tra loro

Rappresentazioni non equivalenti



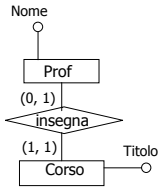


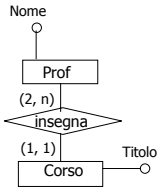
Legame Dipendenti e Dipartimenti

Incompatibilita'

- R1 e R2 sono in contrasto tra loro a causa dell'incoerenza nelle specifiche

Rappresentazioni incompatibili





Legame Professori e Corsi

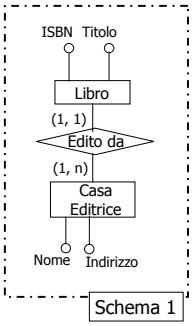
CONFLITTO

- Si verifica un conflitto tra due rappresentazioni di uno stesso concetto ogni volta che le due rappresentazioni non sono identiche.

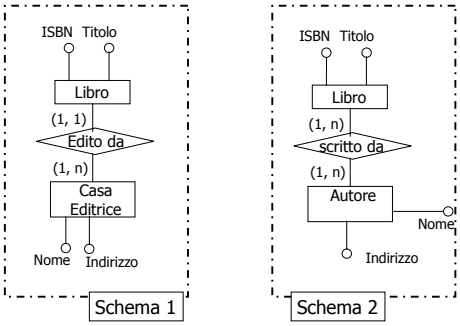
Concetti correlati

- A seguito dell'integrazione tra schemi, può accadere che concetti diversi ma correlati diano vita a nuove relazioni.
- Proprietà INTER-SCHEMA

Concetti correlati

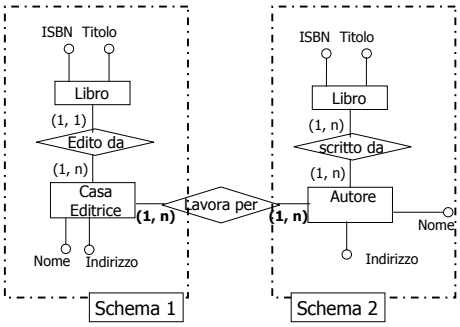


Concetti correlati



Proprietà inter-schema tra Casa editrice e Autore

Concetti correlati



Proprietà inter-schema tra Casa editrice e Autore

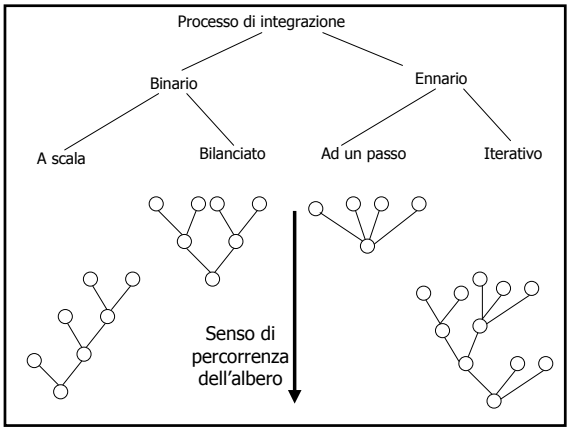
Il processo di integrazione

Metodologia (Batini et al.)

- Il processo di integrazione prevede i seguenti passi:
  - Preintegrazione
  - Comparazione degli schemi
  - Allineamento degli schemi
  - Fusione e ristrutturazione degli schemi

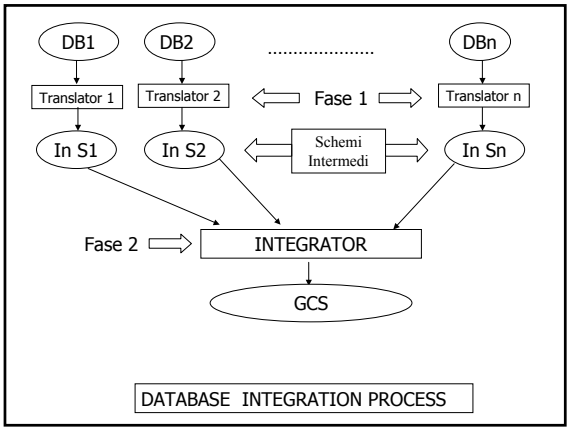
Preintegrazione

- Definizione delle porzioni di schemi che dovranno essere integrate
  - Non tutti i dati sono utili ai fini decisionali
- Definizione della strategia di integrazione
  - In quale ordine si procede nell'integrazione degli schemi
    - Tecniche Binarie
      - Semplici ma non danno una visione completa di un concetto
    - Tecniche Ennarie:
      - Complesse ma "complete"



Database integration: 2 passi

- Il processo di integrazione avviene in due passi:
  1. Schema Translation: gli schemi locali dei db partecipanti sono tradotti in una rappresentazione intermedia canonica.
  2. Schema Integration: ogni schema intermedio viene integrato in un global conceptual schema.



Schema Translation

- Si riconducono i singoli schemi ad uno schema comune in modo da facilitare la comparazione tra gli stessi.
- Il modello Relazionale e l'E-R sono I modelli più utilizzati per svolgere il ruolo di "modello comune".
- Ovviamente questo passo è necessario solo se gli schemi locali dei singoli DBMS sono di un modello diverso.

Schema Integration

Schema Integration

- Fase che segue il processo di traduzione e genera il GCS a partire dagli schemi intermedi.
- In alcune metodologie, l'integrazione viene fatta sui Local External Schemas più che sui Local Conceptual Schemas

Schema Integration

- Sottofasi:
  - Omogeneizzazione
    - Soluzione dei conflitti strutturali e semantici
  - Integrazione
    - Merging e ristrutturazione degli schemi "omogeneizzati".

⇩

Global Conceptual Schema

Omogeneizzazione

- Vengono risolti problemi di eterogeneità semantica e strutturale.
- Eterogeneità semantica:
  - NAMING CONFLICTS:
  - Sinonimi: una stessa entità individuata con nomi diversi.
    - Esempio: Employee ed E usati per il nome del dipendente.
  - Omonimi: due entità diverse con lo stesso nome.
    - Esempio: l'attributo Address in un'entità rappresenta l'indirizzo (nome strada e numero civico) e l'altro indica l'indirizzo completo di cap e città.

Omogeneizzazione

- NAMING CONFLICTS:
- Ci sono vari metodi alternativi per trattare questo problema
- Esempio:
  - I sinonimi sono risolti semplicemente rinominando.
  - Gli omonimi si risolvono:
    - Prefissando ogni attributo col nome dell'entità a cui appartiene
    - Prefissando ogni entità col nome dello schema cui appartiene.

Omogeneizzazione

- Conflitti strutturali:

1. Type conflict:
   - quando lo stesso oggetto è rappresentato da un attributo in uno schema e da un'entità in un altro
2. Dependency conflict:
   - Quando sono usate relationship diverse (es. one-to-one vs. many-to-many) per rappresentare la stessa cosa in schemi differenti
3. Key conflict:
   - Quando ci sono chiavi diverse nei diversi schemi
4. Behavioral conflict:
   - Insiti nel meccanismo di modellazione ad esempio nei vincoli di integrità

Integrazione

- Fusione degli schemi intermedi in un unico "database schema" e successiva ristrutturazione per ottenere il "migliore" schema integrato.
- La fusione richiede che l'informazione contenuta negli schemi partecipanti siano mantenute nello schema integrato.

Integrazione

- Merging e Ristrutturazione

1. Tutti gli schemi vengono fusi in un unico schema e poi ristrutturati per creare lo schema integrato migliore.
2. Requisiti:
   - Completezza
   - Minimalità
   - Comprensibilità

Requisiti

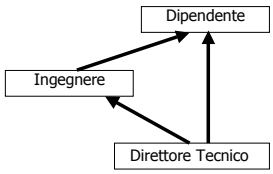
- Completezza:
  - tutte le informazioni provenienti dagli schemi sono integrate nello schema comune
  - dopo la sovrapposizione degli schemi sorgenti risulteranno visibili ulteriori proprietà inter-schema non evidenziate in precedenza (nuove associazioni, gerarchie di generalizzazione,…)

Requisiti

- Minimalità:
  - La sovrapposizione di schemi può generare forti ridondanze di concetti che, sebbene espressi in modo univoco negli schemi sorgenti, risultano duplicati o derivabili in quello riconciliato

Requisiti

- Minimalità: (esempio)



Schema con relazione di inclusione ridondante

Requisiti

- Comprensibilità:
  - dopo che si è effettuato il merge di tutti gli elementi, la ristrutturazione deve facilitare la creazione di uno schema comprensibile e sveltire le successive fasi della progettazione

Annotazione

- Non sempre i conflitti possono essere risolti (ad es. per inconsistenze di base del sistema informativo):
  - Si devono fare scelte e si devono dare priorità (es. si preferiscono le trasformazioni che avvantaggiano gli schemi ritenuti centrali)
    →
    la strategia che risulta più adatta è quella "binaria a scala" (si comincia l'integrazione a partire dagli schemi ritenuti più importanti)

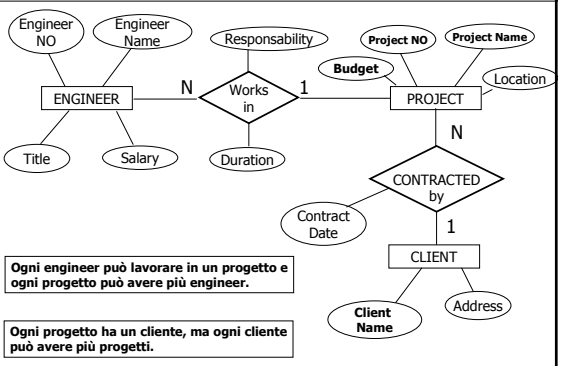
Esempio

Database Relazionale

(S2) Sia dato il seguente database relazionale:

EMP (ENO, ENAME, TITLE)
PROJ (PNO, PNAME, BUDGET, *LOC, CNAME*)
ASG (ENO, PNO, RESP, DUR)
PAY (TITLE, SAL)

(S1) Sia dato il seguente ENTITY-RELATIONSHIP DATABASE



Processo

1. Fase: SCHEMA TRANSLATION
   - Traduzione

2. FASE: SCHEMA INTEGRATION
   1. Omogeneizzazione
   2. Integrazione

Schema Translation

Traduzione dello schema relazionale nel modello E-R

- Determinare le relazioni che rappresentano entità
- Determinare le relazioni che rappresentano relationships
- Identificare la cardinalità delle relationship
- Identificare i vincoli sulle relationship
- .....

EMP (ENO, ENAME, TITLE)
PROJ (PNO, PNAME, BUDGET, *LOC, CNAME*)
ASG (ENO, PNO, RESP, DUR)
PAY (TITLE, SAL)

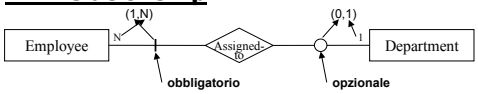
ENTITA'

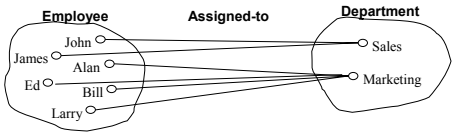
RELATIONSHIP

ER Model – Esempio Relationships

NOTAZIONI

1:N relationship

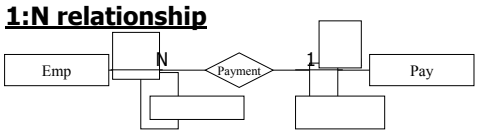


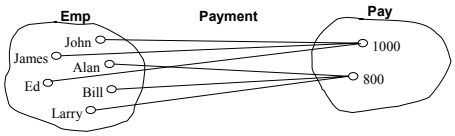


ER Model – Esempio Relationships

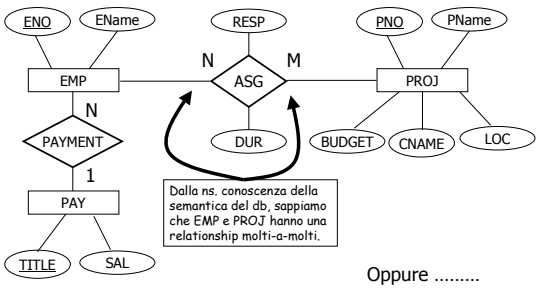
NOTAZIONI

1:N relationship



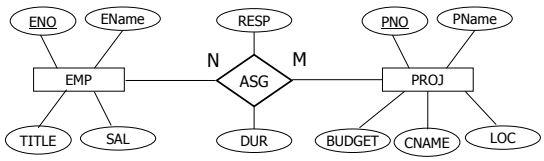


E-R equivalente al Relazionale



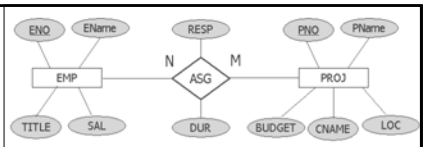
Oppure .....

E-R equivalente al Relazionale

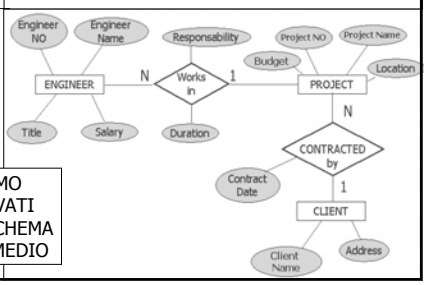


Assumiamo di considerare questo modello

Modello S2



Modello S1



SIAMO ARRIVATI ALLO SCHEMA INTERMEDIO

Schema Integration:
omogeneizzazione ed integrazione

Omogeneizzazione

- RICORDIAMO:
  - Conflitti semantici
  - Conflitti strutturali
    1. Type conflict
    2. Dependency conflict
    3. Key conflict
    4. Behavioral conflict

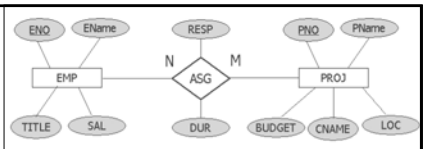
OMOGENEIZZAZIONE
Conflitti semantici

Conflitti semantici

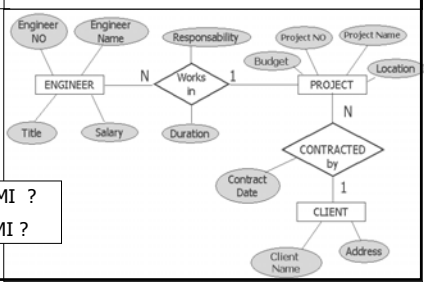
- I <u>naming conflicts</u> dell'esempio possono essere risolti rinominando:
  - Entità
  - Attributi
  - Relationships

    (ad es., potremmo rinominare lo schema del modello S2)

Modello S2



Modello S1



SINONIMI ?

OMONIMI ?

Sinonimi

nello SCHEMA INTERMEDIO

InS_1	InS_2
ENGINEER	EMP
Engineer No	ENO
Engineer Name	ENAME
Salary	SAL
WORKS IN	ASG
Responsability	RESP
Duration	DUR
PROJECTS	PROJ
Project No	PNO
Project Name	PNAME
Location	LOC

Omonimi

nello SCHEMA INTERMEDIO

InS_1	InS_2
ENGINEER	EMP
Title	Title

In InS1, Title si riferisce al titolo degli Engineers mentre in InS2 si riferisce al titolo di tutti gli impegati per cui ha un dominio più ampio
➔ caso di omonimia: lo stesso nome di attributo si riferisce a due cose diverse.

domain (EMPLOYEE.Title) >> domain (ENGINEER.Title)

- Una possibile soluzione alle omonimie è:

Se l'omonimia è a livello di attributo ↔ prefissare ogni attributo col nome dell'entità a cui appartiene

Se l'omonimia è a livello di entità ↔ prefissare ogni entità col nome dello schema cui appartiene.

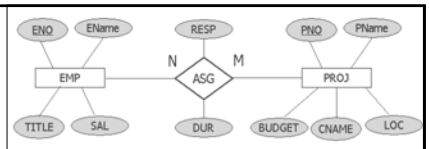
Nel ns. esempio potremmo rinominare lo schema relativo al db relazionale (S2) per renderlo conforme alla nomenclatura dello schema del db E-R (S1)

OMOGENEIZZAZIONE: Conflitti strutturali

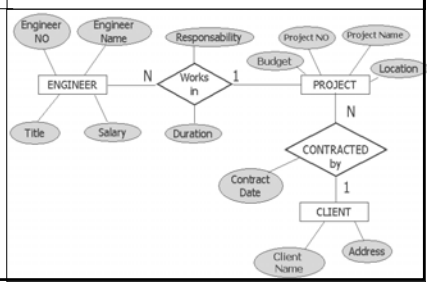
Conflitti strutturali

- Un modo di risolverli è tramite la trasformazione di entità/attributi/relationships

Modello S2



Modello S1



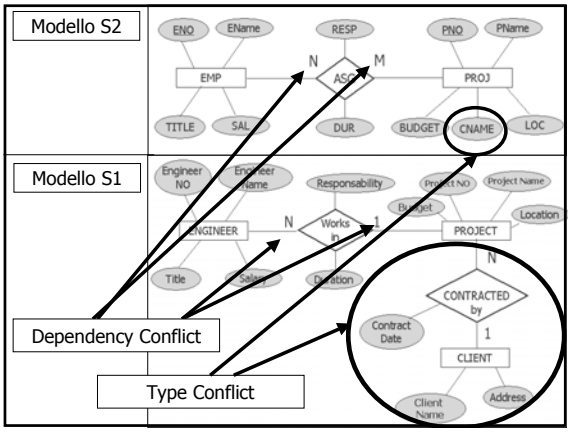
Conflitti strutturali (esempio)

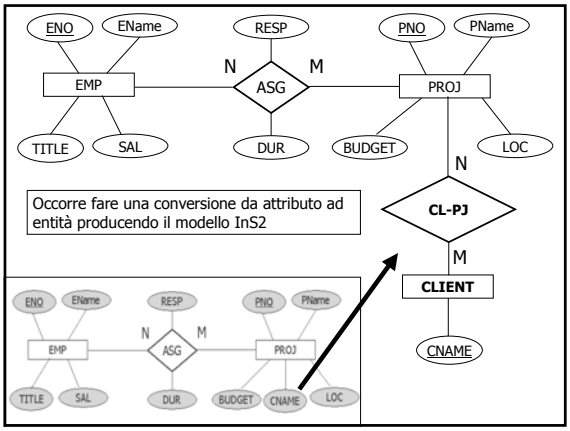
1. <u>TYPE CONFLICT</u>:
   Nell'esempio, nel modello E-R, il Cliente di un progetto è modellato come una entità mentre, nel modello relazionale, è un attributo dell'entità PROJ.

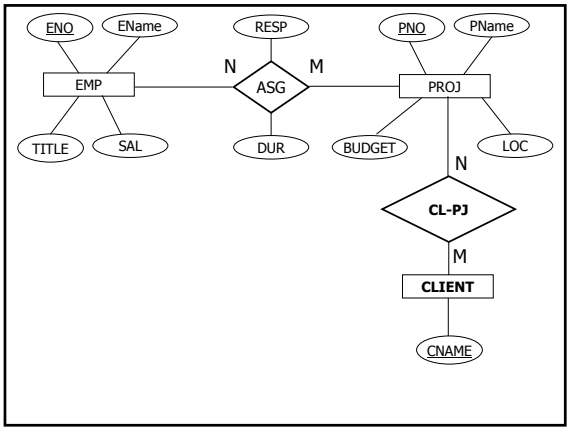
Conflitti strutturali (esempio)

2. <u>DEPENDENCY CONFLICT:</u>
   Nell'esempio, la relationship del modello E-R è molti-a-uno mentre nel relazionale è molti-a-molti.
   - *Decisione di progetto:*

si sceglie di accettare la relationship più generale molti-a-molti.

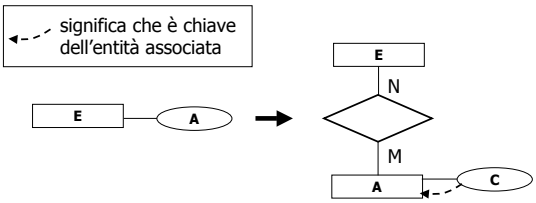

Modello S2
ENO
EName
RESP
PNO
PName
EMP
N
ASG
M
PROJ
TITLE
SAL
DUR
BUDGET
CNAME
LOC
Modello S1
Engineer NO
Engineer Name
Responsability
Project NO
Project Name
Budget
Location
ENGINEER
N
Works in
1
PROJECT
Title
Salary
Duration
N
CONTRACTED by
Contract Date
1
CLIENT
Client Name
Address
Dependency Conflict
Type Conflict


ENO
EName
RESP
PNO
PName
EMP
N
ASG
M
PROJ
TITLE
SAL
DUR
BUDGET
LOC
N
CL-PJ
M
CLIENT
CNAME
Occorre fare una conversione da attributo ad entità producendo il modello InS2


ENO
EName
RESP
PNO
PName
EMP
N
ASG
M
PROJ
TITLE
SAL
DUR
BUDGET
LOC
N
CL-PJ
M
CLIENT
CNAME

Conversione da attributo ad entità

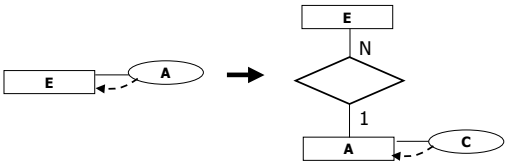
Modello di conversione da attributo ad entità (1)



Un attributo A, non chiave, può essere trasformato in una nuova entità.

Tra la nuova entità e quella di partenza esiste una relationship molti-a-molti su di un nuovo attributo (C).

Modello di conversione da attributo ad entità (2)



Un attributo chiave può essere trasformato in un nuova entità; tra la nuova e l'originale entità esiste una relationship molti-a-uno tramite un nuovo attributo (C).

LIBRO

ISBN	Titolo	Casa Edit.
123445	XXXX	McGraw
4354543	Yyyyy	Hoepli
123456	Zzzzzz	Fabbri
.....	.....	.....

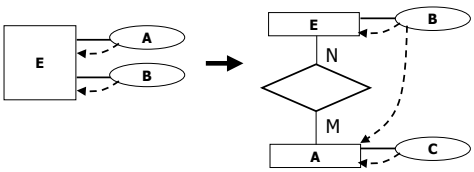
LIBRO

ISBN	Titolo	IDCE
123445	XXXX	001
4354543	Yyyyy	002
123456	Zzzzzz	001
.....	.....	.....

CASA EDITRICE

IDCE	Descrizione
001	McGraw
002	Hoepli
003	Fabbri
.....	.....

Modello di conversione da attributo ad entità (3)



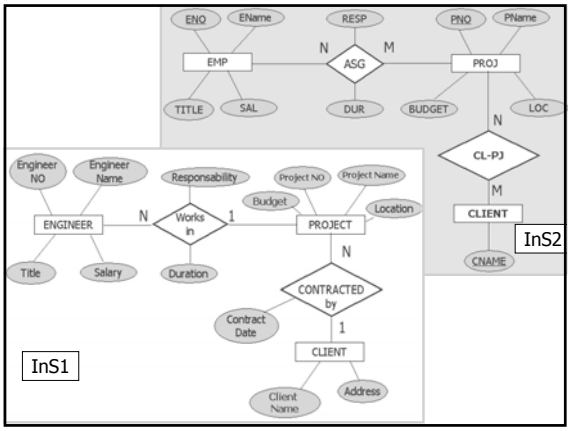
Questo modello mostra il caso in cui A è parte di una chiave.

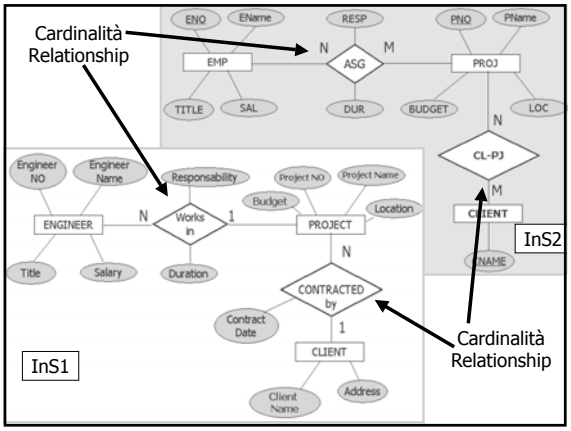
Come correlare i due schemi

- L'omogeneizzazione richiede una buona dose di intervento umano dato che è richiesta la conoscenza semantica di tutti gli schemi intermedi.

Integrazione

- Fusione degli schemi intermedi in un unico "database schema" e successiva ristrutturazione per ottenere il "migliore" schema integrato.
- La fusione richiede che l'informazione contenuta negli schemi partecipanti siano mantenute nello schema integrato.

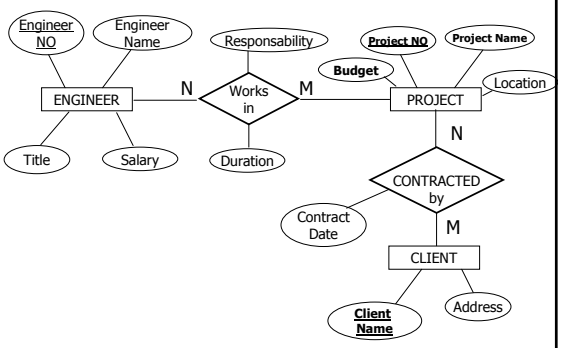




Esempio

- Nel ns. esempio, essendo InS2 subset di InS1
  ➔ InS1 diviene lo schema integrato.
- Complicazione: le diverse caratteristiche della relationship CONTRACTED-BY (in InS1 è di tipo molti-a-uno; in InS2, è di tipo molti-a-molti che è più generale).
  ➔La fusione può compiersi accettando InS1 come risultato MA disponendo che la relationship CONTRACTED-BY sia convertita in una di tipo molti-a-molti.

GLOBAL CONCEPTUAL SCHEMA INTEGRATO



Merging e Ristrutturazione

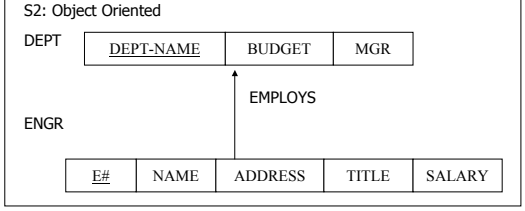
- Requisiti:
  - **Completezza**: tutte le informazioni provenienti dagli schemi sono integrate nello schema comune.
  - **Minimalità**: nessuna ridondanza viene prodotta nello schema integrato.
  - **Comprensibilità**: dopo che si è effettuato il merge di tutti gli elementi, la ristrutturazione deve facilitare la creazione di uno schema comprensibile.

Esercizio

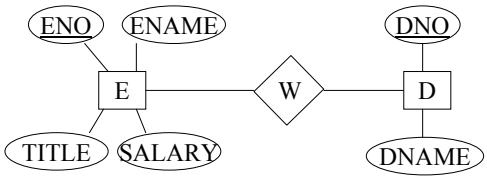
Esempio
siano dati i seguenti schemi:

S1: Relational
E (ENO, ENAME, TITLE)
D (DNO, DNAME)
W (ENO, DNO)
S (TITLE, SALARY)

S2: Object Oriented

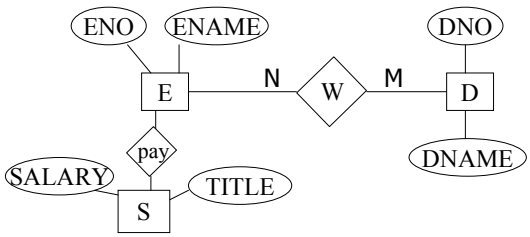


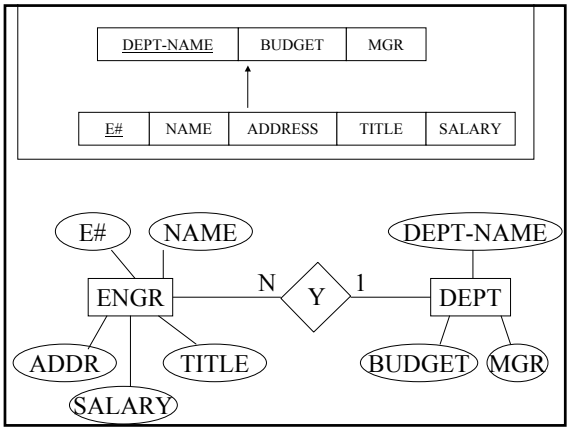
Modello ER per S1



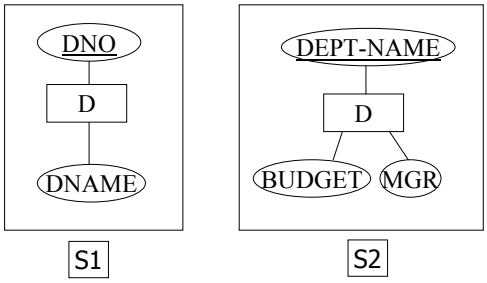
Oppure…

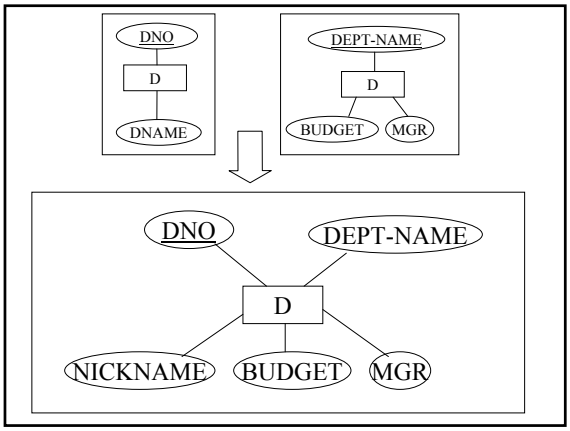
S1: Relational
E (ENO, ENAME, TITLE)
D (DNO, DNAME)
W (ENO, DNO)
S (TITLE, SALARY)

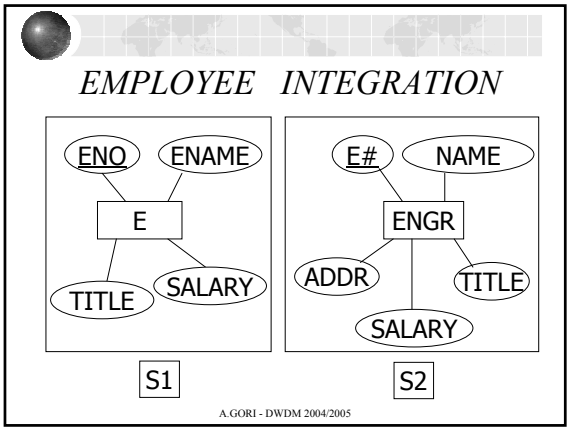


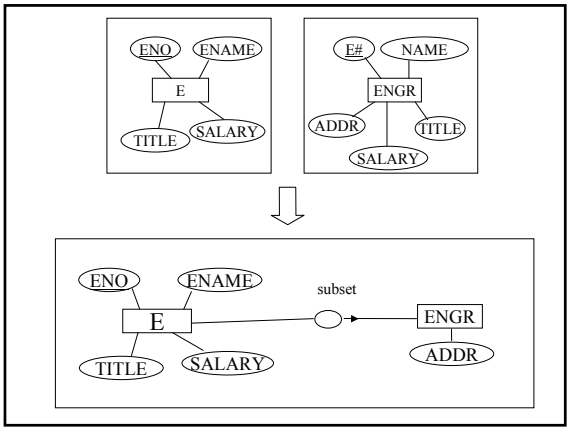


DEPT INTEGRATION

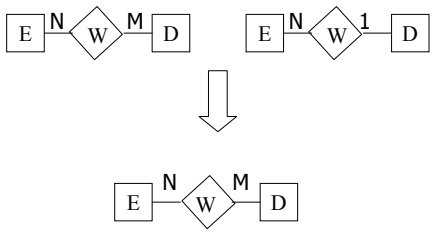



DNO
D
DNAME
DEPT-NAME
D
BUDGET
MGR
DNO
DEPT-NAME
D
NICKNAME
BUDGET
MGR


EMPLOYEE INTEGRATION
ENO
ENAME
E
TITLE
SALARY
E#
NAME
ENGR
ADDR
TITLE
SALARY
S1
S2
A.GORI - DWDM 2004/2005


ENO
ENAME
E
TITLE
SALARY
E#
NAME
ENGR
ADDR
TITLE
SALARY
ENO
ENAME
E
TITLE
SALARY
subset
ENGR
ADDR

Works-in relationship



Esempio

Definire lo schema esterno globale significa definire l'insieme delle viste esterne che consentono di accedere ai dati dei singoli sistemi.

Questo può essere fatto tramite istruzioni del tipo:

CREATE VIEW view-name (attr1, attr2, ... attrn) AS SELECT statement

La creazione della vista dovrebbe essere l'unione tra le relazioni corrispondenti dei vari siti.

Esempio:

CREATE VIEW new_view (attr1, … attrn) AS

(SELECT …… FROM rel1@db1)

UNION

(SELECT …… FROM rel2@db2);

Esempio

Schema for db1:
scientist [name, age, state-birth]
publication [title, keyword, year]
book [num, title, year]
series [name, editor]
belongs-to [book-nr, series-name]
publisher-of [series-name, pub-name]
written-by [publ-title, auth-name]
author [name, address]
of-interest [auth-name, topic]

Schema for db2:
scientist [name, position, degree, cityof-birth]
borrower [pub-title, sci-name]
dealt-with [paper-title, topic-code]
paper [title, author]
journal [title, stack]
proceedings [title, stack]
book [title, author, stack]
topic [code, name]
concerns [pub-title, topic-code]
belongs-to-proc [paper-title, proc-title]

Creare la vista "written-by"

Schema for db1:
scientist [name, age, state-birth]
publication [title, keyword, year]
book [num, title, year]
series [name, editor]
belongs-to [book-nr, series-name]
publisher-of [series-name, pub-name]
written-by [publ-title, auth-name]
author [name, address]
of-interest [auth-name, topic]

Schema for db2:
scientist [name, position, degree, cityof-birth]
borrower [pub-title, sci-name]
dealt-with [paper-title, topic-code]
paper [title, author]
journal [title, stack]
proceedings [title, stack]
book [title, author, stack]
topic [code, name]
concerns [pub-title, topic-code]
belongs-to-proc [paper-title, proc-title]

Esempio

Schema for db1:
written-by [publ-title, auth-name]

Schema for db2:
paper [title, author]
book [title, author, stack]

```
CREATE VIEW written-by (title, author) AS
( SELECT publ-title, auth-name FROM written-by@db1 )
UNION
( SELECT title, author FROM paper@db2 )
UNION
( SELECT title, author FROM book@db2 );
```

Esercizio

Esercizio: risolvere conflitti di struttura

A

sid	cid	grade
s1234	c1	85
s1234	c2	90
...	...	...
s1234	cn	75
...	...	...

B

sid	c1	c2	...	cn
s1234	85	90	...	75
...	...	...	...	...

Esercizio: risolvere conflitti di struttura

A

sid	cid	grade
s1234	c1	85
s1234	c2	90
...	...	...
s1234	cn	75
...	...	...

B

sid	c1	c2	...	cn
s1234	85	90	...	75
...	...	...	...	...

B → A

Select sid, "c1", c1 from B
Union
.....

Esercizio: risolvere conflitti di struttura

A

sid	cid	grade
s1234	c1	85
s1234	c2	90
…	…	…
s1234	cn	75
…	…	…

B

sid	c1	c2	…	cn
s1234	85	90	…	75
…	…	…	…	…

A → B

Create view V_c1(sid, c1) as

Select sid, grade from A where cid='c1';

……

V_c1 join V_c2 … join V_cn

Fine