

Lezione 1

▼ Corso	Basi di Dati per Bioinformatica
📅 Data	@October 4, 2021 8:30 AM
☑ Rifacimento	☑
▼ Status	Completed
▼ Tipo	Lezione

Sistema Informativo

Componente di una organizzazione che gestisce le informazioni di interesse - non tutte quelle disponibili, ma quelle di interesse! Cosa fa?

Gestisce la informazioni:

- acquisire → memorizzare → analizzare (interrogare, per dire che mi servono per fare qualcosa)
- elaborare
- produrre
- conservare
- condividere
- analizzare

Rientra tutto nel verbo **gestire**.

Il nostro sistema è la **BASE DI DATI** (ma in realtà io faccio riferimento a tutti i sistemi nella gestione di dati).

Ogni organizzazione ha un sistema informativo. Un'organizzazione è un termine generale, possiamo rappresentare qualsiasi insieme organizzato che ha necessità di gestire informazioni. Non per forza questo è esplicitato, soprattutto se l'organizzazione è piccola. Ma quasi sempre è di supporto ad altri sottosistemi: alla parte operativa, decisionale, di produzione, ecc.

Il *sistema organizzativo* è l'insieme di risorse e regole per lo svolgimento coordinato delle attività: se sono un'organizzazione avrò sicuramente un **obiettivo**. Le attività

che mi permettono di raggiungere degli obiettivi fanno parte dei processi. Come organizzare le risorse? Se ne occupa il sistema organizzativo, non sempre identificabile.

Tra le risorse di un'azienda ci sono le informazioni. È per questo che le organizzazioni hanno un *sistema informativo*, ergo **imparare a gestire l'informazione è importante**.

Il concetto di "Sistema Informativo" non dipende per forza dall'automazione. Nella nostra accezione è sinonimo di **Sistema Informatico**, ma in realtà la gestione dell'informazione può essere indipendente dall'informatica e dall'automazione.



Il Sistema Informatico è la porzione automatizzata del Sistema Informativo: gestisce informazioni con tecnologia informatica.

Gestione delle informazioni

Gestione: quello che posso fare con l'informazione (posso e devo non sono la stessa cosa):

- raccogliere
- acquisire
- conservare
- archiviare
- elaborare
- trasformare
- produrre
- distribuire
- comunicare
- scambiare
- proteggere

Cose che mi interessa sapere: Informazioni. Perché? Sono una risorsa.

Le basi di dati sono il sistema informativo che ci permette di immagazzinare i dati.

Nelle attività umane ci sono per forza delle informazioni, in forme diverse, sono qualsiasi cosa: idee, cose scritte, cose raccontate, è tutta informazione. Formati

diversi. Ha vari supporti: mente, carta, dispositivi.

I dati

Data la diversità dell'informazione, via via si è avuta la necessità di avere metodi di organizzazione e codificazione dell'informazione. Come tenerne traccia?

Estrapolare i dati di interesse e archiviare.

Sottile differenza tra informazioni e dati: il dato è una rappresentazione dell'informazione. Per rappresentare un'informazione uso un certo insieme di dati: devono essere interpretati per fornire una determinata informazione. L'intuizione che mi porta al significato, ciò non vuol dire però che il dato sia completo.



I dati diventano informazione solo una volta che sono stati interpretati, altrimenti mi servono a poco.

Per com'è costruito un modello relazionale avremo la spiegazione dei dati inseriti.

Più dati ho, più risorse ho: rappresentarli bene e viceversa estrapolarli non è banale, quindi impareremo, data una descrizione delle informazioni di interesse, come posso rappresentare i dati in un certo contesto, a interrogare la base di dati.

Base di dati

Una base di dati è:

- Accezione generica, metodologica:
 - Insieme organizzato di dati, utilizzati per il supporto allo svolgimento delle attività di un ente.
- Accezione specifica, metodologica e tecnologica:
 - Insieme di dati gestito da un Sistema di Gestione di Basi di Dati (DBMS)

Cos'è un sistema di Gestione dei Basi di Dati?

Gestisce dati grandi, persistenti e condivise (collezione di dati). Garantisce privacy, affidabilità, efficienza ed efficacia.

Cosa vuol dire che le collezioni sono grandi? Le dimensioni dei dati memorizzate è molto maggiore della memoria centrale dei sistemi di calcoli utilizzati. Il limite alla dimensione dei dati è solo quello fisico dei dispositivi: ho tanti dati, con l'evoluzione dei sistemi siamo arrivati a parlare di BigData, più passa il tempo più aumentano i dati da gestire.

Cosa vuol dire che devono essere persistenti? Hanno un tempo di vita indipendente dalle singole esecuzioni dei programmi che le utilizzano (le devo immagazzinare)

Cosa vuol dire che sono condivise? (spesso usiamo Basi di Dati come sinonimo dei dati stessi) spesso allo stesso insieme di dati accedono persone e settori diversi. Alla stessa base di dati accedono utenti diversi, con funzionalità diverse. Ognuno accede alla porzione che gli è di competenza (privatezza).

Ho bisogno di meccanismi di autorizzazione: sembra ovvio ma devo controllare e il sistema deve sapere chi sta accedendo e con quali diritti, dal momento che ho più utenti che accedono alle stesse sezioni devo avere un controllo della concorrenza. Il DBMS mi permette di fare queste cose.

Potrei avere delle parti a cui determinate persone non possono accedere.

Cosa vuol dire che sono affidabili? Anche in caso di malfunzionamenti io non devo perdere dati, né essi devono essere danneggiati: è garantita la persistenza dei dati. È importante garantire affidabilità? Non mi posso permettere che vengano modificati o persi.

Cos'è una transazione? È un insieme di operazioni da considerare indivisibile, corretto, anche in presenza di concorrenza o mal funzionamento e con effetti definitivi. Grazie all'utilizzo delle transazioni ho il concetto di indivisibili: o arrivo a fare tutto o non posso fare nulla. Il concetto di insieme di operazioni indivisibili e atomiche vuol dire che devo per forza farle tutte insieme o tornare al punto di partenza. Le transazioni sono concorrenti: devo gestire le cose che concorrono senza perderne alcuna. Permanenti: la conclusione positiva di una transazione corrisponde ad un impegno, è definitivo. E il **DBMS** deve fare in modo che sia così.

Cosa vuol dire che garantisce efficienza? Cerca di utilizzare al meglio le risorse: spazi di memoria e tempo.

Cosa vuol dire che è efficace? Offre funzionalità articolate, potenti e flessibili.

Descrizione dei dati nei DBSM

Due **modelli**: dei dati **relazionale** e **entità-relazione**. Si inseriscono a due diversi livelli di astrazione.

I modelli dei dati sono il modo in cui rappresento i dati. È un insieme di concetti che utilizzo per organizzare i dati di interesse e descriverne la struttura.

Lo uso per astrarre e concettualizzare i dati.

Modello relazionale

Es. tabella relazione: il modello relazionale dei dati prevede il costruttore relazione, che permette di definire insiemi di record omogenei.

Ogni modello di dati fornisce meccanismi di strutturazione (o costruttori tipo) analoghi a quelli dei linguaggi.

Costrutto = **RELAZIONE** (dentro la tabella è una rappresentazione tabellare. Cosa decido delle tabelle? Cosa mi interessa)

Decido la struttura del record in base ai dati di interesse. Una volta decisa la struttura, è quella. Per questo è importante pensare prima la struttura. A priori devo sapere cosa mi interessa memorizzare e una volta che lo so definisco la relazione. Dentro le relazioni ho **attributi**, che rappresentano la struttura dell'informazione.



Il **modello relazionale** è un modello logico, che usa il costrutto per rappresentare l'informazione. Il modello logico utilizza delle strutture che riflettono una particolare organizzazione.

Modello entità-relazione

È un modello concettuale, ovvero astrae e si occupa dei concetti che voglio rappresentare.



I **modelli concettuali** descrivono i concetti del mondo reale piuttosto che i dati utili a rappresentarlo. Sono il primo passo per costruire il grafico per avere ben presente l'astrazione grafica delle informazioni.

Usiamo la parola relazione per rappresentare cose diverse: nel modello logico è la tabella, in quello concettuale è l'approccio di un'entità a un'altra.

La potenza di questo modello sta nell'essere sconnesso da ogni forma logica.



Dato il modello logico relazionale dei dati: a cosa si riferisce la relazione? dare una definizione di relazione.

Cos'è un modello logico e cos'è un modello concettuale?

Architettura di base

Noi interagiamo con uno schema logico, lo schema logico interagisce con uno schema interno.

Indipendenza dei dati: il livello logico è indipendente da quello fisico, che il sistema usi un modello o un altro non ho bisogno di sapere cosa c'è sotto. Il modello concettuale è usato per costruire quello logico (può sempre essere utile come documentazione, ma l'utente non ci interagisce quando usa la base di dati relazionale).

Schema e Istanza

In ogni base di dati esistono uno schema e un'istanza.

Lo **schema** descrive la struttura della base di dati, sostanzialmente è invariante nel tempo. Modificare lo schema di una base di dati può causare problemi.

L'**istanza** sono i valori attuali di una base di dati: tutto ciò che ci inserisco, cambia continuamente nel tempo, anche molto rapidamente.

Linguaggi per basi di dati

DDL (linguaggi per definizione di dati) e DML (Linguaggi di manipolazione dei dati)

La prima categoria definisce la tabella per com'è fatta: da una definizione dell'organizzazione dei dati. Permette la definizione di schemi e altre operazioni generali.

Il secondo permette di interrogare la base di dati e aggiornare un'istanza. SQL fa entrambe le cose ed è usato dalla maggior parte delle basi di dati.

Il modello relazionale

È diventato il modello standard di riferimento, è disponibile dall'81 (quindi in dieci anni si è affermato come standard). Si basa sul concetto matematico di relazione con una variante. È basato sui valori: quindi anche i riferimenti fra i dati contenuti nelle relazioni sono rappresentato per mezzo dei valori stessi.

Le relazioni si rappresentano naturalmente come tabelle.

Duplicata caratteristica: parte formale delle relazioni e rappresentazioni intuitive.

Relazione

3 accezioni diverse.

1. **Relazione matematica**, come da teoria degli insiemi

Cos'è la relazione matematica?

Dati n insiemi D_1, D_2, D_n

- il prodotto cartesiano
- relazione matematica è un sottoinsieme del prodotto cartesiano
- Una relazione di n domini ha grado n

Una relazione è un insieme e quindi: non è definito un ordinamento, sono distinte, ciascuna n -upla è ordinata. Sono definiti grado e cardinalità.

Ciascuno dei due domini ha un ruolo diverso in base alla posizione in cui si trova: la struttura è posizionale - questo è un po' limitante: per cui Cod associa ad ogni dominio un nome unico (chiamato attributo) che descrive il ruolo del dominio. L'ordine tra gli attributi è ora irrilevante e la struttura non è più posizionale. Ad ogni dominio quindi associo un nome dell'attributo che descrive il significato del dominio, che definisce la struttura della tabella.

2. **Relazione** (relationship) **che rappresenta una classe di fatti**, nel modello entity-relationship; tradotto come associazione o correlazione

Una tabella rappresenta una relazione quando i valori di ogni colonna sono omogenei tra di loro e i domini sono unici (le righe sono diverse tra di loro) e l'ordinamento tra righe e colonne è irrilevante.

Tabelle diverse nella stessa base di dati possono avere attributi uguali, ma fanno riferimento a tabelle diverse - creo legami tra le tabelle basati sul contenuto degli attributi (SUL VALORE NON SUL NOME)

Il modello relazionale è basato sui valori.

3. **Relazione secondo il modello relazionale dei dati**



Quando una tabella rappresenta una relazione?

Lezione 2

▼ Corso	Basi di Dati per Bioinformatica
📅 Data	@October 5, 2021 9:30 AM
☑ Rifacimento	<input type="checkbox"/>
▼ Status	Completed
▼ Tipo	Lezione

Modello Relazionale

Schema Relazione

$R(X)$ $X = \{A1, A2, A3, ..., An\}$ è una relazione R su un insieme X di attributi

anche indicato come **$R(A1, A2, ..., An)$** dove R è il nome della mia relazione e $A1, A2, ..., An$ sono un insieme di attributi.

Es. tabella → STUDENTE(matriola, cognome, nome) e questo è lo schema della mia relazione. Non devo definire altro perché i valori di una determinata colonna sono omogenei rispetto al tipo.

Una base di dati è formata da più relazioni, quindi lo schema di una base di dati sarà:

$R = \{R1(X1), R2(X2), ..., Rn(Xn)\}$

dove R è il nome della base di dati e ogni $Ri(Xi)$ è una relazione. Lo schema di una base di dati è formato da un insieme di schemi di relazioni.

Istanza

Ogni riga dell'istanza si chiama **tupla**, una tupla è una **funzione**, definita su un insieme di attributi X, che mi permette di associare a ciascun attributo A, un valore del dominio di A.

$t[\text{matr}] = \text{VR01}$

Ogni riga della tabella si chiama tupla.

L'istanza della mia base di dati:

(ISTANZA di) BD (uso solo basi di dati ma faccio riferimento alla stessa cosa) è l'insieme dei valori che fanno riferimento alla mia base di dati.

$r = \{(VR01, Rossi, Paolo), (VR02, Verdi, Giorgia)\} \{(BDBIO, 6, Basi\ di\ dati, 30), (SIA, 6, SisInformativi\ aziendali, 10)\} \{(VR01, BDBIO), (VR01, SIA), (VR02, BDBIO)\}$ questo è lo schema di tutti i valori della mia base di dati, la mia **istanza**.

$R = \{STUDENTE(matricola, nome, cognome), CORS(cod, CFU, titolo, StudFreq), FREQ(codStud, CodCorso)\}$ e questo è lo **schema** della mia base di dati

Posso avere relazioni sui singoli attributi:

es. RAPPRESENTANTE(MatStud) ad esempio rappresentante è sulla matricola dello studente. La relazione di questa tabella è:

RAPPRESENTANTE(MatStud)

VALORE NULLO

Attenzione!

Posso avere una situazione in cui non tutte le tuple hanno tutti i valori per tutti gli attributi. Io impongo la struttura che devono avere i dati che devono essere inseriti, però non sempre tutti i dati sono disponibili.

Cosa si fa in questo caso?

Uso un valore nullo, perché non posso pensare di avere un valore di dominio. Deve essere riconoscibile il fatto che non ho il valore, non posso usare una stringa qualsiasi.



Il valore nullo denota l'assenza di un valore nel dominio e non è un valore del dominio.

Posso rappresentare 3 casi possibili:

1. Il valore è sconosciuto

→ il valore esiste ma io non lo conosco

2. Il valore è inesistente

→ il valore non esiste nella realtà modellata

3. Il valore è senza informazione

→ il valore può esistere o non esistere e se esiste io non lo conosco (è un or dei due casi precedenti)

I DBMS fanno sempre riferimento al 3° caso, non fanno distinzione tra i diversi casi (vuol dire che non so niente!)

Alcuni valori nulli non mi creano alcun tipo di problema, ma altri possono creare vari problemi:

- Ad esempio può succedere che una tupla non possa mai essere correlata ad altre tuple (es. medico/paziente)
- oppure risultino del tutto inutili, non fornendo alcuna informazione
- oppure creino confusione tra le tuple

C'È LA NECESSITÀ DI IMPORRE RESTRIZIONI SULL'USO DEI VALORI NULLI!

Potrei anche avere dei dati sintatticamente corretti ma non ammissibili per l'applicazione (es. pazienti con lo stesso codice - identificazione univoca, oppure inesistente tra le possibilità che ho)

Per ovviare a questi problemi:

Vincoli di integrità (referenziale)

Sono proprietà che devono essere soddisfatte dalle istanze che rappresentano informazioni corrette per l'applicazione.

Fanno riferimento a un certo dominio dell'applicazione, non sono universali, io devo sapere le regole da rispettare perché i dati siano inseriti nel mio DB.

A cosa servono i vincoli? Ci permettono di avere a che fare con dati di qualità!

Quali sono i tipi di vincoli di integrità?

Due categorie:

1. INTRARELAZIONALI

Riguardano una singola relazione

- a. vincoli di tupla

la valutazione del vincolo avviene sulla singola tupla

b. vincoli su un singolo valore o di dominio

es. il voto non può essere maggiore di 30 (sul singolo valore)

c. **vincoli di chiave**

specifico qual è l'attributo o l'insieme di attributi, che mi permettono di identificare in maniera univoca una tupla, **molto importanti**

Quindi ci interessa riflettere su cos'è la chiave di una relazione!

È un insieme di attributi, che può essere anche uno solo, utilizzato per identificare in maniera univoca le tuple di una relazione.

2. INTERRELAZIONALI

Sono definiti tra più relazioni

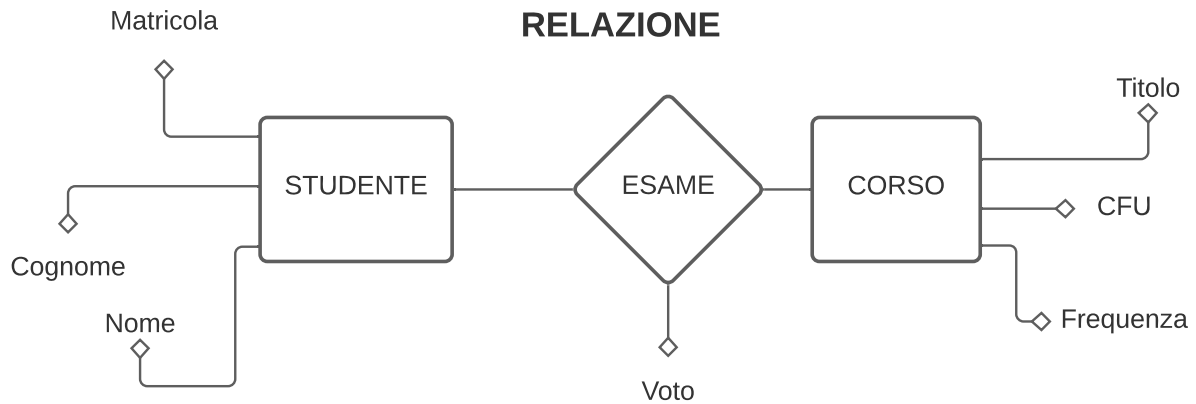
Esempi DBSM

Chiara S | October 10, 2021

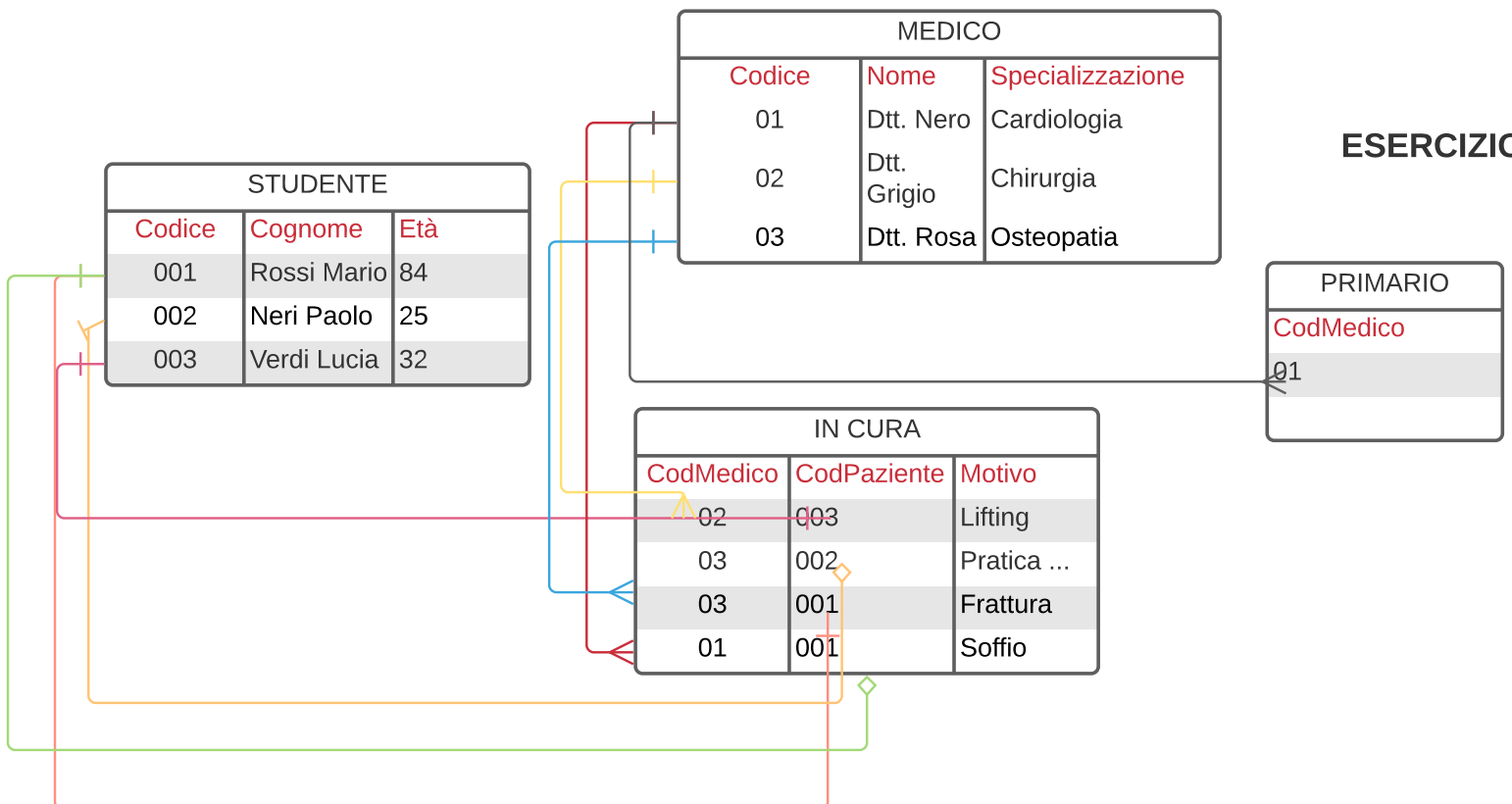
MODELLO RELAZIONALE



MODELLO ENTITÀ RELAZIONE



ESERCIZIO



Lezione 3

▼ Corso	Basi di Dati per Bioinformatica
📅 Data	@October 11, 2021 9:00 AM
☑ Rifacimento	☑
▼ Status	Completed
▼ Tipo	Lezione

Vincoli di Chiave

Chiave = insieme degli attributi che mi permette di individuare in maniera univoca ogni tupla della relazione e deve garantire due proprietà.

1. UNICITÀ

In una qualunque istanza di una relazione, non possono esistere due tuple distinte la cui restrizione alla chiave sia uguale → due tuple diverse non possono avere la stessa chiave.

$$t1[k] = t2[k] \leftrightarrow t1[x] = t2[x]$$

2. MINIMALITÀ

Non deve essere possibile sottrarre un attributo alla chiave senza che la proprietà di unicità cessi → la chiave è formata dall'insieme minimo di attributi che identificano la tupla e se ne tolgo uno non identifica più la chiave

Chiave e Superchiave

Un insieme k di attributi è **superchiave** per una relazione r , se r non contiene due tuple distinte $t1$ e $t2$, con $t1[k] = t2[k]$

Questo insieme di attributi k diventa una **chiave** per la mia relazione r , se è una **superchiave minimale** per r .

Se so che un attributo è chiave allora qualsiasi insieme di attributi che lo contengono non sarà sicuramente chiave (poiché è superchiave, ma sicuramente non minimale).

Individuare una chiave

Quando definisco uno schema, devo ricordare che sto definendo la struttura dei dati che devo inserire, e vorrò definire uno schema generale che non sia vincolato dalle informazioni che so per certo ci finiranno dentro → la scelta a priori è una scelta generale (es. scegliere cognome e nome come chiave di una tabella non è opportuno).

Nella definizione dello schema della mia relazione, per individuare una chiave (specificare quali sono i vincoli di chiave che voglio soddisfare), devo:

- considerare le **proprietà dei dati**
 - frammento del mondo reale di interesse → tra tutti gli attributi che caratterizzano l'informazione prendo solo quelli importanti per ciò che sto gestendo.
 - Cerco di stabilire quali sono i dati che mi interessano e poi cerco ciò che può individuare univocamente una tupla (in generale, non basandosi sulla propria istanza) → quali insiemi di attributi → non esistendo due tuple uguali, di base so per certo che l'insieme di attributi che ho definito è una superchiave.



Cose da tenere presente:

Perché è così importante avere una chiave?

- Permette di accedere alle tuple → garantisce l'accessibilità a ciascun dato della mia relazione e quindi a ciascun dato della BD

- Ogni valore della base di dati è accessibile tramite il nome della relazione, il valore della chiave e il nome dell'attributo
- Le chiavi ci permettono di correlare i dati in relazioni diverse, poiché il modello relazionale dei dati è basato sui valori.

Chiavi e Valori Nulli

In presenza di valori nulli, i valori della chiave non ci permettono di identificare le n-uple e non posso realizzare i riferimenti.

Quindi la presenza di valori nulli nelle chiavi deve essere limitata.

Si chiama **Chiave Primaria** la chiave su cui non sono ammessi valori nulli.



notazione:

La chiave primaria ovviamente cambia a seconda della realtà che sto rappresentando, così come il concetto di ciò che sto rappresentando.

Vincoli di integrità referenziale

("Foreign key" o "vincoli di chiave esterna")

Le informazioni in relazioni diverse della mia base di dati sono correlate attraverso valori comuni, in modo coerente (rappresentano la stessa informazione).

In particolare i valori comuni a cui faccio riferimento sono i valori delle chiavi primarie ma non solo.

Si chiama **Vincolo di Integrità Referenziale** fra gli attributi X di una relazione *r1* ed *r2*, un vincolo che impone ai valori su X di *r1* di comparire come valori di chiave primaria di *r2*.

→ es. esiste un vincolo di integrità referenziale tra l'attributo codice medico della tabella paziente e la tabella medico:

Paziente(Codice, Cognome, Nome, dataNascita, *codMedico*)

Medico(*Codice*, Cognome, Nome, Telefono, Specializzazione, *Reparto*)

Reparto(*Codice*, Descrizione)

ed esiste quindi un vincolo d'integrità referenziale tra l'attributo Reparto della tabella medico e la tabella Reparto. ***r1* è la tabella interna, *r2* è la tabella esterna.**

È importante proprio perché il modello relazionale è basato sui valori. L'importante è controllare che quando inserisco qualcosa nella tabella interna, esiste nella tabella esterna, non il contrario. L'insieme da cui traggio (la tabella esterna) può essere più grande, ma mai più piccolo.

Importanza:

1. Giocano un ruolo fondamentale nel concetto di modello basato sui valori
2. I riferimenti sono diretti verso le chiavi primarie
3. In presenza di valori nulli, possono essere resi meno restrittivi → potrei avere dei passaggi caricando
4. Vincoli su più attributi, conta l'ordine (primo con primo, secondo su secondo, ...)
5. Si possono definire delle azioni compensative a seguito della loro violazione → devo aver definito dei meccanismi che mi dicano cosa fare in funzione della violazione del vincolo.

Violazione del vincolo di integrità referenziale

Si possono introdurre violazioni al vincolo di integrità referenziale, **modificando** la tabella interna o la tabella esterna (intendo che modifico l'istanza)

/aggiungere tabelle per vincoli integrità referenziale

1. Violazioni a fronte di una modifica della tabella **interna**
 - a. modificando attributo **referente**

b. inserimento di una **nuova** riga che viola il vincolo

- queste operazioni vengono **impedite**

2. Violazioni a fronte di una modifica della tabella **esterna**

a. modifica attributo **riferito**

b. **cancellazione** di una tupla

- queste operazioni hanno bisogno di **definire delle azioni compensative**

Azioni compensative

a. Modifica attributo riferito

- Modifica a cascata
- Introduzione di NULL
- Introduzione di un valore di default

b. Cancellazione di una tupla

- Cancellazione a cascata
- Introduzione di NULL (meno pericoloso)
- Introduzione di un valore di default

Esercizio

Definire uno schema di basi di dati (con il modello relazionale dei dati) per organizzare le informazioni di una clinica privata:

- infermieri(*codiceFiscale*, nome, cognome, dataNascita)
- medici(*codiceFiscale*, nome, cognome, dataNascita, specializzazione)
- reparti(codicePiano, *caporeparto* (infermiere), *primario* (medico))

Ogni infermiere e ogni medico lavorano in un reparto. Indicare le chiavi e i vincoli di integrità referenziali dello schema e costruire una istanza della base di dati.

Vincoli di integrità:

C'è un vincolo di integrità referenziale tra l'attributo Codice Fiscale (riferito) della tabella Infermiere e l'attributo caporeparto (riferente)

C'è un vincolo di integrità referenziale tra l'attributo Codice Fiscale (riferito) della tabella Medico e l'attributo primario (riferente) della tabella Reparto

C'è un vincolo di integrità referenziale tra l'attributo codicePiano (riferito) della tabella Reparto e codReparto della tabella Infermiere (idem per quel che riguarda la tabella Medico)

INFERMIERE.codReparto → Reparto (non specifico nulla perché è chiave primaria)

MEDICO.codReparto → Reparto

REPARTO.Caporeparto → Infermiere

REPARTO.Primario → Medico

Esercizio

lunedì 4 ottobre 2021 09:04

Esercizio

Definire uno schema di basi di dati (con il modello relazionale dei dati) per organizzare le informazioni di una clinica privata:

- infermieri(codiceFiscale, nome, cognome, dataNascita)
- medici(codiceFiscale, nome, cognome, dataNascita, specializzazione)
- reparti(codicePiano, caporeparto (infermiere), **primario** (medico))

Ogni infermiere e ogni medico lavorano in un reparto. Indicare le chiavi e i vincoli di integrità referenziali dello schema e costruire una istanza della base di dati

INFERMIERE

codiceFiscale	nome	cognome	dataNascita	codReparto
FEOIGF00E32364F	Marco	Rossi	12/05/78	01
IFEWET11R0234E	Sara	Bianchi	30/08/00	03
POIVER55V1234S	Giulio	Verdi	21/01/98	02

La chiave è il codice fiscale (univoco e minimale)

REPARTO

codicePiano	caporeparto	primario
01	FEOIGF00E32364F	AAABBB0909F
02	POIVER55V1234S	HHHHIE52123A
03	IFEWET11R0234E	ZZZAAAD1212E

Se codicePiano è univoco, è chiave, altrimenti è l'insieme degli attributi ad essere chiave, o anche l'insieme di caporeparto e primario (se sono univoci per reparto)

MEDICO

codiceFiscale	nome	cognome	dataNascita	specializzazione
ZZZAAAD1212E	Giacomo	Marroni	03/03/80	Odontoiatria
AAABBB0909F	Claudia	Blu	20/09/87	Cardiologia
HHHHIE52123A	Mara	Gialli	13/01/78	Chirurgia

La chiave è il codice fiscale (univoco e minimale)

Vincoli di integrità:

C'è un vincolo di integrità referenziale tra l'attributo Codice Fiscale (riferito) della tabella Infermiere e l'attributo caporeparto (riferente) della tabella Reparto

C'è un vincolo di integrità referenziale tra l'attributo Codice Fiscale (riferito) della tabella Medico e l'attributo primario (riferente) della tabella Reparto

C'è un vincolo di integrità referenziale tra l'attributo codicePiano (riferito) della tabella Reparto e codReparto della tabella Infermiere (riferente), idem per quel che riguarda la tabella Medico

INFERMIERE.codReparto -> Reparto (non specifico nulla perché è chiave primaria)

MEDICO.codReparto -> Reparto

REPARTO.Caporeparto -> Infermiere

REPARTO.Primario -> Medico

Esercizi

▼ Corso	Basi di Dati per Bioinformatica
📅 Data	@October 12, 2021 9:30 AM
☑ Rifacimento	☑
▼ Status	Completed
▼ Tipo	Esercizi

Domande di ripetizione

Per associare più cose ad una sola ho bisogno di una relazione esterna

1. **Cos'è il modello relazionale?** Un modello di dati logico, che si basa su un tipo di costrutto che è la relazione
2. **Perché è importante?** Perché ci permette di organizzare i dati in una struttura
3. **Cos'è la schema di una relazione?** È dato dal nome di una relazione e l'insieme degli attributi, con nomi tutti diversi. All'interno di una base di dati posso avere più relazioni che hanno attributi con lo stesso nome (uso anche la relazione per riconoscere l'attributo)
4. **Il modello relazionale è basato sui valori. Che vuol dire?** I collegamenti tra relazioni diverse si basano sui valori degli attributi, il valore dell'attributo è UNO SOLO, non posso avere più valori.
5. Per avere la **garanzia che i dati sono coerenti** usiamo i vincoli di integrità referenziale, specificando quando modelliamo il data base quali vincoli vogliamo che vengano rispettati. L'attributo di tot. tabella fa riferimento ad un'altra (chiave primaria) della tabella esterna. Tutto ciò che appare nell'attributo referente nella tabella interna, deve essere un valore di una chiave di una tabella interna. Per evitare i problemi, specifico delle azioni di gestione della violazione dei vincoli referenziali.

Domande da esame:

1. Dato il modello relazionale dei dati, cos'è una relazione?
2. Cos'è lo schema e cos'è l'istanza di una relazione?
3. Cos'è la chiave di una relazione?
4. Quando posso parlare di chiave? (quando ho una superchiave minimale)
5. Quando una superchiave è primaria? (quando non può avere valori nulli)



Esercizi svolti sul quaderno

Lezione 4

▼ Corso	Basi di Dati per Bioinformatica
📅 Data	@October 18, 2021 9:00 AM
☑ Rifacimento	☑
▼ Status	Completed
▼ Tipo	Lezione

Operatori Insiemistici

r_1 ed r_2 devono essere compatibili, quindi devono avere lo stesso grado (stesso numero di attributi), domini ordinatamente dello stesso tipo in modo da avere tuple compatibili.

Unione

L'unione di due relazioni r_1 ed r_2 , definite sullo stesso insieme di attributi X , $r_1 \cup r_2$, è una relazione su X . Il risultato contiene tutte le tuple che appartengono ad r_1 oppure tutte le tuple che appartengono ad r_2 oppure ad entrambe.

Ad esempio ipotizzando un'unione tra la tabella docenti e la tabella studenti, e sempre ipotizzando di avere uno studente che è anche docente, avrò una tabella che contiene tutti i docenti e tutti gli studenti e il docente-studente che appare solo una volta.

Intersezione

L'intersezione di due relazioni r_1 ed r_2 , definite sullo stesso insieme di attributi X , $r_1 \cap r_2$, è ancora una relazione su X . Il risultato contiene solo le tuple che appartengono sia ad r_1 che ad r_2 .

Rifacendoci all'esempio in cui abbiamo uno studente-docente, l'unica tupla presente nella tabella della relazione sarà quello studente-docente.

Differenza

La differenza di due relazioni r_1 ed r_2 , definite sullo stesso insieme di attributi X , $r_1 - r_2$, è ancora una relazione su X . Il risultato contiene le tuple di r_1 , che non appartengono ad r_2 .

Sempre nello stesso esempio avremo una tabella come quella studenti, ma senza lo studente-docente, poiché questo appartiene anche ai docenti.

Attributi diversi e ridenominazione

Sarebbe sensato alle volte voler costruire determinate unioni, ma sorge il problema di avere relazioni non definite sullo stesso insieme di attributi. Abbiamo quindi bisogno di quella che si chiama **Ridenominazione** è un operatore unario, cioè che lavora sulla singola relazione, che modifica lo schema lasciando inalterata l'istanza dell'operando, permettendomi di rendere compatibili gli schemi.

Sia r una relazione su un insieme di attributi X , e Y un altro insieme di attributi con la stessa cardinalità.



Siano A_1, A_2, \dots, A_k e B_1, B_2, \dots, B_k , rispettivamente un ordinamento per gli attributi di X e di Y , allora la $\rho_{B_1, B_2, \dots, B_k \leftarrow A_1, A_2, \dots, A_k}(r)$

Nell'esempio padre e madre, li ridenomino $\rho_{\text{GENITORE}} \leftarrow \text{PADRE}(\text{Paternità})$ e $\rho_{\text{GENITORE}} \leftarrow \text{MADRE}(\text{Maternità})$.

Risultato: MATERNITÀ(genitore, figlio) e PATERNITÀ(genitore, figlio).

Ora posso fare l'unione delle due tabelle.

Selezione

È un operatore unario, che produce un risultato che ha lo stesso schema della tabella di partenza, e contiene un sottoinsieme delle tuple della tabella di partenza, quindi dell'operando, che soddisfano una certa condizione.

La selezione produce quindi un risultato sullo stesso schema in cui però son state tagliate alcune tuple, perché non soddisfano una condizione.



Si dice che la selezione genera una decomposizione orizzontale.

- **Condizione** di selezione: la condizione è ciò che le tuple devono soddisfare per finire nel risultato
- L'**operando** è la relazione su cui applico la selezione

Esempio: condizione: sulla tabella studenti, solo quelli che hanno età maggiore di 25

$\sigma_{età > 25}(STUDENTI)$

Condizioni che posso specificare

- Confronti tra attributi $A_1 O A_2$, dove O può essere $=, \neq, <, >, \leq, \geq, \dots$
- confronto tra un attributo e una costante k
 - $A_1 O k$, dove O può essere $=, \neq, <, >, \leq, \geq, \dots$
- Condizioni complesse ottenute combinando condizioni semplici con connettori logici
 - $Cond_1 \wedge Cond_2$
 - $Cond_1 \vee Cond_2$
 - $\neg Cond_1$

$\sigma_{stipendio < 50}(Impiegato)$

Proiezione

Operatorie unario che produce un risultato, il quale NON ha lo stesso schema dell'operando. Lo schema del risultato è un sottoinsieme degli attributi di partenza.



Si dice che la proiezione genera una decomposizione verticale.

Il risultato di una proiezione contiene al più tante tuple quanto l'operando ma può contenerne di meno: se due tuple hanno attributi proiettati uguali, avrò una sola tupla (non ho duplicati).

Cardinalità delle Proiezioni

La cardinalità della proiezione è \leq alla cardinalità dell'operando.

Esiste un legame tra i vincoli di chiave e le proiezioni.

$\Pi_y r$ contiene lo stesso numero di tuple di r se e solo se y è una **superchiave** di r .

Se y è una superchiave allora r non contiene tuple uguali su y e quindi ogni tupla di r su y dà un contributo al risultato.

Nel momento in cui proietto la chiave sono sicura che tutte le tuple finiscano nel risultato.

Combinare selezione e proiezione

Prima vado a verificare in quali tuple applico la selezione ed elimino quelle che non rispettano la condizione. Date le tuple che rispettano la selezione, vado a prendere gli attributi che proietto.

$\Pi_{cognome, matricola} (\sigma_{stipendio > 50} (Impiegato))$

Faccio prima la selezione e poi la proiezione, altrimenti perderei le informazioni su cui operare la selezione.

Posso scriverlo anche in **formato ad albero**:

$\Pi_{matricola, cognome}$

|

$\sigma_{stipendio > 40}$

|

Impiegato

La logica dell'algebra relazionale è che sono io a stabilire quali operazioni fare sulle tabelle per trovare le cose che sto cercando.

Nell'esame ci sono 3 operazioni di algebra relazionale

Esempi Operatori Insiemistici

LAUREATI

MATRICOLA	NOME	ETA'
7274	Rossi	42
7432	Neri	54
9824	Verdi	45

QUADRI

MATRICOLA	NOME	ETA'
9297	Neri	33
7432	Neri	54
9824	Verdi	45

LAUREATI \cup QUADRI

MATRICOLA	NOME	ETA'
7274	Rossi	42
7432	Neri	54
9824	Verdi	45
9297	Neri	33

LAUREATI \cap QUADRI

MATRICOLA	NOME	ETA'
7432	Neri	54
9824	Verdi	45

LAUREATI $-$ QUADRI

MATRICOLA	NOME	ETA'
7274	Neri	42

PATERNITÀ

PADRE	FIGLIO
Adamo	Abele
Adamo	Caino
Abramo	Isacco

MATERNITÀ

MADRE	FIGLIO
Eva	Abele
Eva	Caino
Sara	Isacco

PATERNITÀ U MATERNITÀ

GENITORE	FIGLIO
Adamo	Abele
Adamo	Caino
Abramo	Isacco
Eva	Abele
Eva	Caino
Sara	Isacco

Ho rinominato (ò) PADRE e MADRE come GENITORE.

IMPIEGATI

COGNOME	UFFICIO	STIPENDIO
Rossi	Roma	55
Neri	Milano	54

OPERAI

NOME	FABBRICA	SALARIO
Bruni	Monza	45
Verdi	Latina	55

R(Cognome, Sede, Retribuzione) -> unione

COGNOME	SEDE	RETRIBUZIONE
Rossi	Roma	55
Neri	Milano	54
Bruni	Monza	45
Verdi	Latina	55

IMPIEGATO

MATRICOLA	COGNOME	FILIALE	STIPENDIO
7399	Rossi	Roma	55
5998	Neri	Milano	64
8553	Milano	Milano	44
5698	Neri	Napoli	64

Trovare gli impiegati che hanno:

Stipendio minore di 50 -> σ

MATRICOLA	COGNOME	FILIALE	STIPENDIO
8553	Milano	Milano	44

Trovare gli impiegati che hanno:

- uno stipendio maggiore di 50
- Uno stip>50 e lavorano a milano
- Lo stesso cognome della filiale in cui lavorano

STIP>50 -> $\sigma_{\text{stipendio}>50}(\text{Impiegato})$

MATRICOLA	COGNOME	FILIALE	STIPENDIO
7399	Rossi	Roma	55
5998	Neri	Milano	64
5698	Neri	Napoli	64

STIP>50 AND FILIALE=MILANO -> $\sigma_{\text{stipendio}>50 \wedge \text{filiale}=\text{milano}}(\text{Impiegato})$

MATRICOLA	COGNOME	FILIALE	STIPENDIO
5998	Neri	Milano	64

COGNOME=FILIALE -> $\sigma_{\text{cognome}=\text{filiale}}(\text{Impiegato})$

MATRICOLA	COGNOME	FILIALE	STIPENDIO
8553	Milano	Milano	44

MATRICOLA	COGNOME	FILIALE	STIPENDIO
7309	Neri	Napoli	
5998	Neri	Milano	
9553	Rossi	Roma	
5698	Rossi	Roma	

Π_{matricola,cognome}(Impiegato)

MATRICOLA	COGNOME
7309	Neri
5998	Neri
9553	Rossi
5698	Rossi

Π_{cognome,filiale}(Impiegato)

COGNOME	FILIALE
Neri	Napoli
Neri	Milano
Rossi	Roma

Rossi, Roma va una volta sola altrimenti sarebbe un duplicato.

MATRICOLA	COGNOME	FILIALE	STIPENDIO
7309	Neri	Napoli	55
5998	Neri	Milano	64
9553	Rossi	Roma	44
5668	Rossi	Roma	64

Voglio trovare: matricola e cognome, degli impiegati che guadagnano più di 50

MATRICOLA	COGNOME
7309	Neri
5998	Neri
6558	Rossi

ESERCIZI

MATRICOLA	NOME	ETÀ	STIPENDIO
7309	Rossi	34	45
5998	Bianchi	37	38
9553	Neri	42	35
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

Trovare matricola, nome, età e stipendio degli impiegati che guadagnano più di 40

$\Pi_{\text{matricola,nome,età,stipendio}}(\text{Impiegato})$

MATRICOLA	NOME	ETÀ	STIPENDIO
7309	Rossi	34	45
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

In esame non ci sono le istanze, ma direttamente lo schema.

Trovare matricola, nome ed età, degli impiegati che guadagnano più di 40

$\Pi_{\text{matricola,nome,età,stipendio}}(\sigma_{\text{stipendio}>40}(\text{Impiegato}))$

MATRICOLA	NOME	ETÀ
7309	Rossi	34
5698	Bruni	43
4076	Mori	45
8123	Lupi	46

Trovare matricola, nome ed età degli impiegati che hanno stipendio uguale alla loro età

$\Pi_{\text{matricola,nome,età}}(\sigma_{\text{età}=\text{stipendio}}(\text{Impiegato}))$

MATRICOLA	NOME	ETÀ
-----------	------	-----

$\Pi_{\text{matricola,nome,stipendio}}(\sigma_{\text{età}>40}(\text{Impiegato}))$

MATRICOLA	NOME	STIPENDIO
9553	Neri	35
5698	Bruni	42
4076	Mori	50
8123	Lupi	60

Lezione 5

▼ Corso	Basi di Dati per Bioinformatica
📅 Data	@October 19, 2021 9:30 AM
☑ Rifacimento	☑
▼ Status	Completed
▼ Tipo	Lezione

Join

simbolo: ⋈

Permette di correlare i dati che sono contenuti in relazioni diverse, confrontando i valori contenuti in esse

Sfrutta le proprietà del modello di essere basato sui valori.

Join Naturale

- operatore binario
- collega dati in relazioni diverse sulla base di valori uguali in attributi con lo stesso nome

Produce un **risultato**

- sull'unione degli attributi delle due relazioni di partenza (gli operandi)
- le tuple sono costruite ciascuna a partire da una tupla di ognuno degli operandi, combinando le tuple con valori uguali negli attributi comuni



Il grado ottenuto con un join è \leq della somma dei gradi degli operandi

Join Completo

Ciascuna tupla di ciascun operando contribuisce ad almeno una tupla del risultato

Join Non Completo

Alcune tuple (dangling) degli operandi non contribuiscono al risultato

- Perché?

Nell'altra relazione non ha il valore corrispondente nell'attributo con lo stesso nome

esempio: in una tabella è citato qualcosa che nell'altra non c'è e quindi quella tupla non partecipa al join

Join Vuoto

Nessuna tupla degli operandi è compatibile

Cardinalità

- **Join completo:** $|r_1 \bowtie r_2| \geq \max(|r_1|, |r_2|)$
- **Join non completo:** $|r_1 \bowtie r_2| < |r_1 \times r_2|$
- **Join Vuoto:** 0
- **Join Naturale Completo:** $|r_1 \bowtie r_2| < |r_1 \times r_2|$

Il join naturale di r_1 ed r_2 contiene un numero di tuple compreso tra 0 e il prodotto delle cardinalità dei due operandi

$$0 < r_1 \bowtie r_2 < r_1 \times r_2$$

- In particolare, se il join è completo, allora contiene almeno un numero di tuple pari al massimo tra r_1 ed r_2
- se dati: $r_1(A, B)r_2(B, C)$, r_2B è chiave allora
 $0 \leq |r_1 \bowtie r_2| \leq |r_1|$
- Se c'è anche il vincolo di integrità referenziale $r_1B \rightarrow r_2$ allora

$$|r_1 \bowtie r_2| = |r_1|$$

Join Esterno

Posso prevedere che tutte le tuple degli operandi finiscono nel risultato:
per farlo estende con valori nulli le tuple che vengono tagliate fuori.

3 tipi di join esterni:

- sinistro: mantiene tutte le tuple del primo operando $r_1 \bowtie_{LEFT} r_2$
- destro: mantiene tutte le tuple del secondo operando $r_1 \bowtie_{RIGHT} r_2$
- completo: mantiene tutte le tuple $r_1 \bowtie_{FULL} r_2$

Proprietà del join naturale

- è commutativo $r_1 \bowtie r_2 = r_2 \bowtie r_1$
- è associativo: $r_{\{1\}} r_1 \bowtie (r_2 \bowtie r_3) = (r_1 \bowtie r_2) \bowtie r_3 = r_1 \bowtie r_2 \bowtie r_3$

$r_1(x_1) \bowtie r_2(x_2)$ **due casi estremi:**

- $x_1 = x_2$

Quando è questo il caso il Join corrisponde con l'intersezione:

$$r_1(x_1) \bowtie r_2(x_2) = r_1(x_1) \cap r_2(x_2)$$

- $x_1 \cap x_2 \neq 0$

In questo caso, degenera in una condizione sempre vera, quindi il risultato del join contiene tutte le tuple ottenute combinando in tutti i modi possibili le tuple degli operandi.

$$x_1 \cup x_2$$

$$|r_1(x_1) \bowtie r_2(x_2)| = |r_1(x_1)| \times |r_2(x_2)|$$

diventa un join naturale sulle relazioni senza attributi in comune:

per trovare le tuple correlate da un attributo, faccio seguire una **selezione**

$$\sigma_{reparto=codice}(Impiegato \bowtie Reparto)$$

Theta Join

Definito come un prodotto cartesiano seguito da una selezione

- è un operatore derivato dagli operatori



$$r_1 \bowtie_F r_2 = \sigma_F(r_1 \bowtie r_2)$$

I DBMS utilizzano il theta join e non il join naturale!

Quando nella condizione uso l'uguaglianza parlo di **EquiJoin**

Esercizio:

date le tabelle IMPIEGATO e SUPERVISIONE, ottenere le matricole dei capi degli impiegati che guadagnano più di 40

$\Pi_{capo}(SUPERVISIONE \bowtie_{impiegato=matricola} (\sigma_{stipendio}(IMPIEGATO)))$

ALGEBRA RELAZIONALE

JOIN COMPLETO

IMPIEGATO	REPARTO
Rossi	A
Neri	B
Bianchi	B
REPARTO	CAPO
A	Mori
B	Bruni

IMPIEGATO	REPARTO	CAPO
Rossi	A	Mori
Neri	B	Bruni
Bianchi	B	Bruni

Cardinalità > del massimo tra la cardinalit' di r1 ed r2

JOIN NON COMPLETO

IMPIEGATO	REPARTO
Rossi	A
Neri	B
Bianchi	B
REPARTO	CAPO
B	Mori
C	Bruni

IMPIEGATO	REPARTO	CAPO
Neri	B	Mori
Bianchi	B	Mori

$< |r1| \times |r2|$

JOIN VUOTO

IMPIEGATO	REPARTO
Rossi	A
Neri	B
Bianchi	B
REPARTO	CAPO
D	Mori
C	Bruni

Non ho valori comuni! Quindi il risultato è 0

JOIN NATURALE COMPLETO – CASO PARTICOLARE

IMPIEGATO	REPARTO
Rossi	A
Neri	A

REPARTO	CAPO
A	Mori
A	Bruni

Ogni tupla è compatibile con ogni tupla

IMPIEGATO	REPARTO	CAPO
Rossi	A	Mori
Rossi	A	Bruni
Neri	A	Mori
Neri	A	Bruni

La cardinalit' è pari a $|r1| \times |r2|$

Partendo da:

IMPIEGATO	REPARTO
Rossi	A
Neri	B
Bianchi	B
REPARTO	CAPO
B	Mori
C	Bruni

IMPIEGATO	REPARTO	CAPO
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	NULL

JOIN ESTERNO SINISTRO

JOIN ESTERNO DESTRO

IMPIEGATO	REPARTO	CAPO
Neri	B	Mori
Bianchi	B	Mori
NULL	C	Bruni

JOIN ESTERNO COMPLETO

IMPIEGATO	REPARTO	CAPO
Neri	B	Mori
Bianchi	B	Mori
Rossi	A	NULL
NULL	C	Bruni

JOIN NATURALE QUANDO x1 intersecato x2 è 0

IMPIEGATO	REPARTO
Rossi	A
Neri	B
Bianchi	B
CODICE	CAPO
A	Mori
B	Bruni

Cardinalità: 3 e 2

Dove: Impiegato(Impiegato, Reparto)

Reparto(Codice, Capo)

IMPIEGATO	REPARTO	CODICE	CAPO
Rossi	A	A	Mori
Rossi	A	B	Bruni
Neri	B	A	Mori
Neri	B	B	Bruni
Bianchi	B	A	Mori
Bianchi	B	B	Bruni

Cardinalità: 6 (3x2)

Non utile: SELEZIONE

Theta-JOIN

IMPIEGATO	REPARTO	CODICE	CAPO
Rossi	A	A	Mori
Neri	B	B	Bruni
Bruni	B	B	Bruni

Esercizi:

Impiegato(Matricola, Nome, Età, Stipendio)

Supervisione(Impiegato, Capo)

IMPIEGATI

MATRICOLA	NOME	ETÀ	STIPENDIO
7309	Rossi	34	45
5998	Bianchi	37	38
9553	Neri	42	35
5698	Bruni	43	42
4076	Mori	45	50
8123	Lupi	46	60

SUPERVISIONE

Impiegato	
7309	5698
5998	5698
9553	4076
5698	4076
4076	8123

Trovare le matricole dei capi degli impiegati che guadagnano più di 40

$\Pi_{\text{capo}}(\text{Supervisione join}_{(\text{impiegato}=\text{matricola})} (\sigma_{\text{stipendio}>40} (\text{Impiegato})))$