

The top-left portion of the slide features a series of thin, light-brown lines that intersect to form several overlapping, irregular polygons. These lines create a complex, abstract geometric pattern that tapers towards the right.

# MODELLO RELAZIONALE E ALGEBRA RELAZIONALE

Vincenzo Calabrò

# Modello relazionale

Proposto da Codd nel 1970 come modello che favorisce l'indipendenza dei dati

Reso disponibile come modello logico nei sistemi reali nel 1981 (affermazione lenta causa l'alto livello d'astrazione)

Si basa su:

- concetto matematico di **relazione**
- rappresentazione di relazioni tramite **tabelle**

## Tre diverse accezioni

- Relazione **matematica** (secondo la teoria degli insiemi)
- Relazione secondo il modello relazionale (**tabella**)
- Relazione (**relationship**) come costrutto del modello concettuale Entità-Relazione (Entity-Relationship) utilizzato nella progettazione

# Relazione matematica (1)

Siano  $D_1, D_2, \dots, D_n$ ,  $n$  insiemi (anche non distinti)

**Prodotto cartesiano**  $D_1 \times D_2 \times \dots \times D_n$  è l'insieme di tutte le  $n$ -uple ordinate  $(d_1, d_2, \dots, d_n)$  tali che  $d_1 \in D_1$ ,  $d_2 \in D_2, \dots, d_n \in D_n$

- $D_1 \times D_2 \times \dots \times D_n = \{(d_1, d_2, \dots, d_n) \mid d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n\}$

Una **relazione matematica** su  $D_1, D_2, \dots, D_n$  è un sottoinsieme del prodotto cartesiano  $D_1 \times D_2 \times \dots \times D_n$

Può essere rappresentata in forma tabellare

# Relazione matematica (2)

Data relazione matematica su  $D_1, D_2, \dots, D_n$

- **Domini** della relazione:  $D_1, D_2, \dots, D_n$
- **Grado** della relazione: il numero di domini  $n$  su cui è definita
- **Cardinalità** della relazione: numero di  $n$ -uple che ne fanno parte (cardinalità dell'insieme relazione)
  - Cardinalità prodotto cartesiano  $|D_1| \times |D_2| \times \dots \times |D_n|$

Tipicamente: cardinalità finite su insiemi eventualmente infiniti

# Relazione matematica: esempio

$$D_1 = \{a, b\} = \begin{array}{|c|} \hline a \\ \hline b \\ \hline \end{array} \quad D_2 = \{x, y, z\} = \begin{array}{|c|} \hline x \\ \hline y \\ \hline z \\ \hline \end{array}$$

Prodotto cartesiano  $c = D_1 \times D_2 =$

- $grado(c) = 2$ ;  $card(c) = |c| = 6$

a	x
a	y
a	z
b	x
b	y
b	z

Un esempio di relazione  $r \subseteq D_1 \times D_2 =$

- $grado(r) = 2$ ;  $card(r) = |r| = 3$

a	y
a	z
b	x

# Relazione matematica: proprietà

Una relazione sui domini  $D_1, D_2, \dots, D_n$  è un **insieme** di  $n$ -uple **ordinate**  $d_1, d_2, \dots, d_n$  tali che  $d_1 \in D_1, d_2 \in D_2, \dots, d_n \in D_n$

- La relazione è un **insieme**:
  - non vi è alcun ordinamento fra le  $n$ -uple
  - le  $n$ -uple di una relazione sono distinte una dall'altra
- Le  $n$ -uple sono **ordinate**:
  - vi è un ordinamento fra domini, l' $i$ -esimo valore di ogni  $n$ -upla proviene dall' $i$ -esimo dominio

# Relazione matematica: esempio

Partite  $\subseteq$  Squadre  $\times$  Squadre  $\times$  Integer  $\times$  Integer

Juve	Lazio	3	1
Lazio	Milan	1	0
Juve	Roma	2	2
Roma	Milan	1	3
Milan	Juve	1	0

- Ciascuno dei domini ha due **ruoli** distinti, distinguibili secondo la loro **posizione**
- La struttura è **posizionale**



# Relazione nel modello relazionale

A ciascun dominio associamo un nome (**attributo**), unico nella relazione, che descrive il suo “ruolo” nella relazione

- Gli attributi possono essere usati come intestazione delle colonne nelle tabelle
- L'ordinamento fra gli attributi è irrilevante: **la struttura non è posizionale**

<b>Casa</b>	<b>Fuori</b>	<b>RetiCasa</b>	<b>RetiFuori</b>
Juve	Lazio	3	1
Lazio	Milan	1	0
Juve	Roma	2	2
Roma	Milan	1	3
Milan	Juve	1	0

# Relazione: definizione

Sia  $X$  l'insieme degli attributi e  $D$  l'insieme dei domini

Definiamo:

- funzione  $dom: X \rightarrow D$ , che associa ad ogni attributo  $A \in X$  il suo dominio  $dom(A) \in D$
- una  $n$ -upla o tupla su  $X$  è una funzione  $t$  che associa a ciascun attributo  $A$  in  $X$  un valore nel dominio  $dom(A)$

Una **relazione** su  $X$  è un insieme di tuple su  $X$

# Relazione: notazione

Sia  $t$  una tupla su  $X$ , siano  $A \in X$  e  $Y \subseteq X$ :

- $t[A]$  o  $t.A$ : valore dell'attributo  $A$  in  $t$
- $t[Y]$ : valore del sottoinsieme di attributi  $Y$  in  $t$

<b>Casa</b>	<b>Fuori</b>	<b>RetiCasa</b>	<b>RetiFuori</b>
Juve	Lazio	3	1
Lazio	Milan	1	0
Juve	Roma	2	2

Sia  $t$  la tupla della prima riga:

- $t[\text{Casa}] = t.\text{Casa} = \text{Juve}$
- $t[\text{Casa}, \text{RetiCasa}] = [\text{Juve}, 3]$

# Tabelle e relazioni

**Una tabella rappresenta una **relazione** se soddisfa le seguenti proprietà:**

- i **valori** di ogni colonna sono **omogenei** fra loro
- le **righe** sono **diverse** fra loro
- le **intestazioni** delle **colonne** sono **diverse** fra loro

**Inoltre, in una tabella che rappresenta una relazione:**

- l'ordinamento fra le righe è irrilevante
- l'ordinamento fra le colonne è irrilevante

# Modello relazionale: definizioni

## Schema di relazione $R(X)$

- $R$  nome di relazione
- $X = \{A_1, \dots, A_n\}$  insieme di attributi

## Schema di base di dati $\mathbf{R} = \{R_1(X_1), \dots, R_n(X_n)\}$

- $\{R_1(X_1), \dots, R_n(X_n)\}$  insieme di schemi di relazioni con nomi diversi

## (Istanza di) relazione $r$ su uno schema $R(X)$

- Insieme  $r$  di tuple su  $X$

## (Istanza di) base di dati $\mathbf{r}$ su uno schema $\mathbf{R}$

- Insieme di relazioni  $\mathbf{r} = \{r_1, \dots, r_n\}$   
( $r_i$  definita su  $R_i(X_i)$ )

# Notazioni

- Attributi
  - lettere iniziali dell' alfabeto  $A, B, C, A', A_1, \dots$
- Insiemi di attributi
  - lettere finali dell' alfabeto:  $X, Y, Z, X', X_1, \dots$
- Nomi di relazione (nello schema)
  - $R$  e lettere circostanti (maiuscole):  $R, S, R', R_1, \dots$
- Relazione (istanza)
  - come il nome della relazione, ma minuscolo:  $r, s, r', r_1$
- Schema di base di dati
  - $R$  e circostanti (maiuscole, grassetto):  **$R, S, R', R_1, \dots$**
- Base di dati (istanza)
  - come il nome dello schema, ma minuscolo:  **$r, s, r', r_1, \dots$**

# Modello relazionale basato su valori

I riferimenti fra dati di relazioni diverse sono rappresentati per mezzo di valori dei domini che compaiono nelle tuple

## CORSI

Corso	Docente	Aula
Architetture	Rossi	Alfa
Basi di dati	Neri	Beta
Linguaggi	Bianchi	Gamma
Sistemi	Verdi	Beta

## AULE

Nome	Piano
Alfa	Terra
Beta	Terra
Gamma	Primo
Delta	Primo

I puntatori possono esistere a livello fisico, ma non sono visibili a livello logico

# Vantaggi utilizzo valori

## Rispetto ad un modello basato su record e puntatori, il modello **basato sui valori**:

- rappresenta solo ciò che è **rilevante** dal punto di vista dell'applicazione (i puntatori sono aspetti realizzativi a basso livello)
- fornisce **indipendenza** della rappresentazione logica rispetto a quella fisica (indipendenza fisica dei dati)
- favorisce **portabilità** dei dati (tutta l'informazione è nei valori)
- le **associazioni** fra i valori sono **bidirezionali**



# Strutture nidificate

**Il modello relazionale ammette solo valori scalari**

È possibile però rappresentare informazione strutturata organizzando propriamente i dati

# Strutture nidificate: esempio (1)

<b>“DA MARIO”</b>		
<b>Ricevuta n. 2453</b>		
<b>del 6/7/2005</b>		
3	Coperti	6.00
2	Antipasti	15.00
3	Primi	27.00
2	Secondi	42.00
3	Caffè	6.00
1	Acqua	3.00
<b>Totale</b>		<b>99.00</b>

<b>“DA MARIO”</b>		
<b>Ricevuta n. 2564</b>		
<b>del 8/7/2005</b>		
2	Coperti	4.00
2	Primi	27.00
2	Secondi	42.00
2	Dessert	12.00
2	Caffè	4.00
1	Vino	25.00
1	Acqua	3.00
<b>Totale</b>		<b>117.00</b>

# Strutture nidificate: esempio (2)

## DETTAGLIO

Numero	Q.ta	Descr.	Importo
2453	3	coperti	6.00
2453	2	antipasti	15.00
2453	3	primi	27.00
2453	2	secondi	42.00
2453	3	caffè	6.00
2453	1	acqua	3.00
2564	2	coperti	4.00
2564	2	primi	27.00
2564	2	secondi	42.00
2564	2	dessert	12.00
2564	2	caffè	4.00
2564	1	vino	25.00
2564	1	acqua	3.00

## RICEVUTE

Num	Data	Totale
2453	6/7/05	99.00
2564	8/7/05	117.00

## Strutture nidificate: esempio (3)

La rappresentazione data per le ricevute è corretta se:

- l'ordine delle righe è irrilevante
- non ci possono essere due righe ripetute

Se queste condizioni non sono soddisfatte è necessario distinguere le singole righe nelle ricevute

# Strutture nidificate: esempio (4)

## DETTAGLIO2

Numero	Riga	Q.ta	Descr.	Imp.
2453	1	3	coperti	6.00
2453	2	2	antipasti	15.00
2453	3	3	primi	27.00
2453	4	2	secondi	42.00
2453	5	3	caffè	6.00
2453	6	1	acqua	3.00
2564	1	2	coperti	4.00
2564	2	2	primi	27.00
2564	3	2	secondi	42.00
2564	4	2	dessert	12.00
2564	5	2	caffè	4.00
2564	6	1	vino	25.00
2564	7	1	acqua	3.00

## RICEVUTE

Num	Data	Totale
2453	6/7/05	99.00
2564	8/7/05	117.00

# Informazione incompleta

Città	Prefettura
Roma	Via IV Novembre
Firenze	
Tivoli	
Prato	

## Diverse semantiche:

- valore **sconosciuto**: Firenze ha prefettura ma non conosciamo indirizzo
- valore **inesistente**: Tivoli non è provincia, non ha prefettura
- **senza informazione** (valore sconosciuto o inesistente): Prato è nuova provincia e non sappiamo se ha prefettura

## Nel modello relazionale

- **Valore nullo (NULL)**: denota assenza di un valore del dominio (senza informazione)
- Formalmente si estende il concetto di  $n$ -upla: per ogni attributo  $A$ ,  $t[A]$  è un valore del dominio oppure il valore nullo NULL
- Si pongono restrizioni sulla presenza di valori nulli (a livello di definizione delle relazioni)

# Valori nulli: esempi

**Troppi valori nulli!**

## STUDENTI

<b>Matr.</b>	<b>Cognome</b>	<b>Nome</b>	<b>Data-nasc</b>
276545	Rossi	Maria	NULL
NULL	Neri	Anna	23/04/1972
NULL	Verdi	Fabio	12/02/1972

## ESAMI

<b>Studente</b>	<b>Voto</b>	<b>Corso</b>
276545	28	01
NULL	27	NULL
200768	24	NULL

## CORSI

<b>Codice</b>	<b>Titolo</b>	<b>Docente</b>
01	Analisi	Giani
03	Chimica	NULL
NULL	Chimica	Belli



# Vincoli di integrità: esempi

Esistono istanze di basi di dati che, pur sintatticamente corrette, non rappresentano valori corretti per l'applicazione

## STUDENTI

Matr.	Cognome	Nome	Data-nasc
200768	Verdi	Fabio	12/02/1972
937653	Rossi	Luca	10/10/1971
937653	Bruni	Mario	11/02/1971

## ESAMI

Studente	Voto	Lode	Corso
200768	36		05
937653	28	Lode	02
276745	25		01

## CORSI

Cod	Titolo	Docente
01	Analisi	Giani
02	Chimica	Melli
03	Chimica	Belli

# Vincoli di integrità (1)

**Proprietà che deve essere soddisfatta dalle istanze che rappresentano informazioni corrette per l'applicazione**

Ogni vincolo può essere visto come un predicato che associa ad ogni istanza valori:

- **vero**: il vincolo è soddisfatto
- **falso**: il vincolo non è soddisfatto

Ogni schema di base di dati ha associato un insieme di vincoli. Sono **corrette** solo le **istanze** della base di dati che **soddisfano tutti i vincoli**

# Vincoli di integrità (2)

## Intra-relazionali (soddisfacimento definito rispetto a singole relazioni)

- di **tupla**: vincolo che può essere valutato su ciascuna tupla indipendentemente dalle altre
- su **valori** (o di **dominio**): vincolo definito con riferimento a singoli valori, è valutato sul singolo attributo
- di **chiave**: vincolo definito rispetto a un insieme di attributi che identificano univocamente le tuple all'interno di una relazione

## Inter-relazionali (coinvolge più relazioni)

# Vincolo di tupla

**Esprime condizioni sui valori di ciascuna tupla indipendentemente dalle altre**

- espressione booleana (AND, OR, NOT) di atomi che confrontano valori di attributo o espressioni aritmetiche su attributi

## Esempi

- ESAMI(Studente,Voto,Lode,Corso)  
NOT (Lode = 'lode') OR (Voto = 30)
- PAGAMENTI(Data,Importo,Ritenute,Netto)  
Netto = Importo – Ritenute

# Vincolo di dominio

**Esprime condizioni sui valori di singoli attributi all'interno di una tupla indipendentemente dagli altri**

- espressione booleana (AND, OR, NOT) di atomi che confrontano valori di attributo con costanti

## Esempi

- ESAMI(Studente, Voto, Lode, Corso)  
(Voto  $\geq$  18) AND (Voto  $\leq$  30)  
(Lode = 'lode') OR (Lode = NULL)

# Chiavi

La chiave è un **insieme di attributi** utilizzato **per identificare univocamente le tuple** all'interno di una relazione

Un insieme di attributi  $K$  è **superchiave** di una relazione  $r$  se  $r$  non contiene due tuple distinte  $t_1$  e  $t_2$  con  $t_1[K]=t_2[K]$

Un insieme di attributi  $K$  è **chiave** di una relazione  $r$  se è una **superchiave minimale** (cioè non esiste un'altra superchiave  $K' \subset K$ )

# Chiavi: esempio

<b>Matricola</b>	<b>Cognome</b>	<b>Nome</b>	<b>Nascita</b>	<b>Corso</b>
4328	Rossi	Luigi	29/04/59	Sicurezza
6328	Rossi	Dario	29/04/59	Sicurezza
4766	Rossi	Luca	01/05/61	Informatica
4846	Neri	Luca	01/05/61	Matematica
5536	Neri	Luca	05/03/58	Sicurezza

## Esempio

### chiavi:

- **Matricola**
- **Cognome, Nome, Nascita**
- **Nome, Corso**

### superchiavi:

- qualsiasi superinsieme che contiene le chiavi

# Chiavi: esistenza

Ogni relazione  $R(X)$  ha sempre **almeno una superchiave** (e quindi una chiave):

- insieme di tutti i suoi attributi  $X$

**Esistenza di una chiave per ogni relazione:**

- garantisce accessibilità a tutti i valori della base di dati e loro identificabilità
- permette di ricostruire le corrispondenze (associazioni) fra i dati contenuti in relazioni diverse



# Chiave primaria

**È necessario limitare la presenza di valori nulli nelle chiavi**

**In ogni relazione una chiave è identificata come **chiave primaria**:**

- non sono permessi valori nulli per gli attributi della chiave primaria
- può essere un attributo aggiunto apposta per l'identificazione

# Chiave primaria: esempi

Notazione:

- per distinguere la chiave primaria sottolineiamo i nomi degli attributi che ne fanno parte

## Esempi

- STUDENTI (Matricola, Nome, Cognome)
- SEMINARI (Codice, Relatore, Titolo, data)
- RICEVUTE (Num, Data, Totale)
- DETTAGLIO2(Numero, Riga, Q.ta, Descr, Importo)

# Vincoli inter-relazionali: integrità referenziale

- $R(X)$  e  $S(Y)$  due relazioni
- $X' \subset X$ ;  $X' = A_1, A_2, \dots, A_n$
- $Y' \subseteq Y$ ;  $Y' = B_1, B_2, \dots, B_n$  (chiave primaria di  $S$ )

## Vincolo di **integrità referenziale** fra $X'$ e $Y'$ :

- per ogni tupla  $t_r \in R$  deve esistere una tupla  $t_s \in S$  con  $t_r[A_i] = t_s[B_i]$ , oppure  $t_r[A_i] = \text{NULL}$ ,  $i = 1, \dots, n$

**Le tuple in  $R$  possono avere per  $X'$  (**chiave esterna**) solo  $n$ -uple di valori esistenti per  $Y'$  in  $S$**

# Integrità referenziale: esempio (1)

## RICEVUTE

<u>Num</u>	Data	Totale
2453	6/7/05	99.00
2564	8/7/05	117.00

R=DETTAGLIO

S=RICEVUTE

X'=Numero

Y'=Num

Chiave esterna:

- Numero in DETTAGLIO

Referenzia:

- Num in RICEVUTE

In **rosso** le violazioni al vincolo

## DETTAGLIO

<u>Numero</u>	<u>Riga</u>	Q.ta	Descr.	Imp.
2453	1	3	coperti	6.00
2453	2	2	antipasti	15.00
2453	3	3	primi	27.00
2453	4	2	secondi	42.00
2453	5	3	caffè	6.00
2453	6	1	acqua	3.00
2564	1	2	coperti	4.00
2564	2	2	primi	27.00
2564	3	2	secondi	42.00
2564	4	2	dessert	12.00
2564	5	2	caffè	4.00
2564	6	1	vino	25.00
2564	7	1	acqua	3.00
<b>2675</b>	1	1	acqua	3.00

# Integrità referenziale: esempio (2)

R=INFRAZIONI

S=AUTO

X'=Prov,Num

Y'=Provincia,Numero

Chiave esterna:

- Prov, Num  
in INFRAZIONI

Referenzia:

- Provincia, Numero  
in AUTO

In **rosso** le violazioni al vincolo

## INFRAZIONI

<u>Codice</u>	<u>Data</u>	<u>Prov</u>	<u>Num</u>
987554	12/10/05	FI	4E5432
630876	15/10/05	MI	2F7653
726375	15/10/05	FI	4G7686
123435	11/10/05	MO	7D6563

## AUTO

<u>Provincia</u>	<u>Numero</u>	<u>Proprietario</u>
FI	4E5432	Rossi Mario
MI	9H4467	Verdi Paolo
FI	7D6563	Gialli Sara
MO	4G7686	Bianchi Silvia

# Linguaggi per basi di dati

## Possibili classificazioni:

- formali o programmatici
- procedurali o dichiarativi

## Algebra

- formale
- procedurale

## SQL

- programmatico
- dichiarativo

# Algebra relazionale (1)

- Definita da CODD (1970)
- Utile per imparare a formulare query
- Insieme di 5 operatori di base danno l'intero potere espressivo del linguaggio

# Algebra relazionale (2)

## Operatori di base

- **selezione**
  - seleziona un insieme di tuple (righe)
- **proiezione**
  - seleziona un insieme di colonne
- **unione**
  - insieme di tuple che appartengono a due relazioni
- **differenza**
  - insieme di tuple che appartengono a una relazione e non all'altra
- **prodotto cartesiano**
  - combinazione di tuple di due relazioni



# Algebra relazionale (3)

## Operatori derivati

- **intersezione**
  - tuple che appartengono a due relazioni
- **join** (naturale, theta-join, equi-join)
  - selezione su un prodotto cartesiano

## Operatori aggiunti

- **ridenominazione**
  - ridenomina un attributo
- **assegnamento**
  - dà un nome alla relazione risultante da una operazione

# Algebra relazionale (4)

## Vedremo le operazioni nel seguente ordine

- operatori aggiunti:
  - ridenominazione
  - assegnamento
- operatori unari
  - selezione
  - proiezione
- operatori binari
  - unione
  - differenza
  - intersezione
  - prodotto cartesiano
  - join (naturale, theta-join, equi-join)

# Ridenominazione (1)

$\rho_{B1,B2,\dots,Bn} \leftarrow A1,A2,\dots,An \ r$

- $r$ : relazione
- $A1,A2,\dots,An$ : lista di attributi che appartengono a  $r$  (sottoinsieme)
- $B1,B2,\dots,Bn$ : nuovo nome da dare agli attributi  $A1,A2,\dots,An$

## Ridenominazione (2)

$\rho_{B1,B2,\dots,Bn \leftarrow A1,A2,\dots,An} r$

Il risultato è una relazione **senza nome** uguale ad  $r$  tranne per il nome degli attributi modificati

- **schema**: schema di  $r - \{A1,A2,\dots,An\} \cup \{B1,B2,\dots,Bn\}$
- **grado**:  $grado(r)$
- **cardinalità**:  $card(r)$

# Ridenominazione: esempio

**CORSI**

<b>Corso</b>	<b>Docente</b>	<b>Aula</b>
Architetture	Rossi	Alfa
Basi di dati	Neri	Beta
Linguaggi	Bianchi	Gamma
Sistemi	Verdi	Beta

$\rho_{\text{Insegnam, Prof} \leftarrow \text{Corso, Docente}}$  **CORSI**

<b>Insegnam</b>	<b>Prof</b>	<b>Aula</b>
Architetture	Rossi	Alfa
Basi di dati	Neri	Beta
Linguaggi	Bianchi	Gamma
Sistemi	Verdi	Beta

# Assegnamento

***risultato := espressione algebrica***

- *risultato*: **nome** da dare alla relazione risultato dell'espressione

Serve per **dare un nome al risultato** di una espressione algebrica

# Assegnamento: esempio

## CORSI

Corso	Docente	Aula
Architetture	Rossi	Alfa
Basi di dati	Neri	Beta
Linguaggi	Bianchi	Gamma
Sistemi	Verdi	Beta

INSEGNAMENTI :=  $\rho_{\text{Insegnam, Prof} \leftarrow \text{Corso, Docente}}$  CORSI

## INSEGNAMENTI

Insegnam	Prof	Aula
Architetture	Rossi	Alfa
Basi di dati	Neri	Beta
Linguaggi	Bianchi	Gamma
Sistemi	Verdi	Beta

# Algebra relazionale

- operatori aggiunti:
  - ridenominazione
  - assegnamento
- operatori unari
  - selezione
  - proiezione
- operatori binari
  - unione
  - differenza
  - intersezione
  - prodotto cartesiano
  - join (naturale, theta-join, equi-join)



# Selezione (1)

## $\sigma_{\text{condizione}}$ $r$

- $r$ : relazione
- $\text{condizione}$ : espressione booleana ( $\wedge, \vee, \neg$ ) di condizioni atomiche del tipo  $A \theta B$  o  $A \theta c$ 
  - $\theta$  è un operatore di confronto ( $=, \neq, <, >, \leq, \geq$ )
  - $A$  e  $B$  sono attributi di  $r$  su cui l'operatore di confronto abbia senso
  - $c$  è una costante compatibile con il dominio di  $A$

Il risultato è una relazione **senza nome** che contiene tutte le **tuple di  $r$  che soddisfano  $\text{condizione}$**

$\sigma_{\text{condizione}} r$

Il risultato è una relazione **senza nome** che contiene tutte le **tuple di  $r$  che soddisfano *condizione***

- **schema**: schema di  $r$
- **grado**:  $\text{grado}(r)$
- **cardinalità**:  $\leq \text{card}(r)$

# Selezione: esempi

**CORSI**

<b>Corso</b>	<b>Docente</b>	<b>Aula</b>
Architetture	Rossi	Alfa
Basi di dati	Neri	Beta
Linguaggi	Bianchi	Gamma
Sistemi	Verdi	Beta

$\sigma_{\text{Aula} = \text{'Beta'}}$  CORSI

<b>Corso</b>	<b>Docente</b>	<b>Aula</b>
Basi di dati	Neri	Beta
Sistemi	Verdi	Beta

$\sigma_{\text{Aula} = \text{'Gamma'}} \vee \sigma_{\text{Corso} = \text{'Sistemi'}}$   
CORSI

<b>Corso</b>	<b>Docente</b>	<b>Aula</b>
Linguaggi	Bianchi	Gamma
Sistemi	Verdi	Beta

# Proiezione (1)

$\Pi_Y r$

- $r$ : relazione
- $Y$ : sottoinsieme degli attributi di  $r$

Il risultato è una relazione **senza nome** che contiene la restrizione delle tuple di  $r$  agli attributi  $Y$  (le **colonne  $Y$** )

$\Pi_Y r$

Il risultato è una relazione **senza nome** che contiene la restrizione delle tuple di  $r$  agli attributi  $Y$  (le **colonne  $Y$** )

- **schema**:  $Y$
- **grado**:  $|Y|$
- **cardinalità**:  $\leq \text{card}(r)$ 
  - uguale se  $Y$  è superchiave
  - può essere minore altrimenti (i **duplicati** sono **eliminati**)

# Proiezione: esempi

**CORSI**

<b>Corso</b>	<b>Docente</b>	<b>Aula</b>
Architetture	Rossi	Alfa
Basi di dati	Neri	Beta
Linguaggi	Bianchi	Gamma
Sistemi	Verdi	Beta

$\Pi_{\text{Corso, Docente}}$  **CORSI**

<b>Corso</b>	<b>Docente</b>
Architetture	Rossi
Basi di dati	Neri
Linguaggi	Bianchi
Sistemi	Verdi

$\Pi_{\text{Aula}}$  **CORSI**

<b>Aula</b>
Alfa
Beta
Gamma

# Selezione, proiezione: esempio

**CORSI**

<b>Corso</b>	<b>Docente</b>	<b>Aula</b>
Architetture	Rossi	Alfa
Basi di dati	Neri	Beta
Linguaggi	Bianchi	Gamma
Sistemi	Verdi	Beta

$\Pi_{\text{Docente}} (\sigma_{\text{Aula} = \text{'Beta'}} \text{CORSI})$

<b>Docente</b>
Neri
Verdi

# Selezione, proiezione, assegnamento: es.

**CORSI**

<b>Corso</b>	<b>Docente</b>	<b>Aula</b>
Architetture	Rossi	Alfa
Basi di dati	Neri	Beta
Linguaggi	Bianchi	Gamma
Sistemi	Verdi	Beta

$BETA := \sigma_{Aula='Beta'} \text{CORSI}$

**BETA**

<b>Corso</b>	<b>Docente</b>	<b>Aula</b>
Basi di dati	Neri	Beta
Sistemi	Verdi	Beta

$\Pi_{Docente} BETA$

<b>Docente</b>
Neri
Verdi



# Algebra relazionale

- operatori aggiunti:
  - ridenominazione
  - assegnamento
- operatori unari
  - selezione
  - proiezione
- operatori binari
  - **unione**
  - **differenza**
  - **intersezione**
  - prodotto cartesiano
  - join (naturale, theta-join, equi-join)

## $r1 \cup r2$

- $r1$  e  $r2$ : relazioni sullo stesso insieme di attributi

Il risultato è una relazione senza nome che contiene tutte le tuple che appartengono a  $r1$  e/o a  $r2$

– tuple duplicate compaiono una sola volta

- **schema**: schema di  $r1$  = schema di  $r2$
- **grado**:  $grado(r1) = grado(r2)$
- **cardinalità**:
  - $\geq \max(card(r1), card(r2))$
  - $\leq card(r1) + card(r2)$

# Unione: esempio

## STUDENTI

Cognome	Nome
Bianchi	Carla
Rossi	Marta
Rosa	Antonio

## IMPIEGATI

Cognome	Nome
Bianchi	Carla
Verdi	Matteo
Gialli	Maria
Viola	Marco

$STUDENTI \cup IMPIEGATI = IMPIEGATI \cup STUDENTI$

Cognome	Nome
Bianchi	Carla
Rossi	Marta
Rosa	Antonio
Verdi	Matteo
Gialli	Maria
Viola	Marco

# Differenza

## $r1 - r2$

- $r1$  e  $r2$ : relazioni sullo stesso insieme di attributi

Il risultato è una relazione senza nome che contiene tutte le tuple che appartengono a  $r1$  ma non a  $r2$

- **schema**: schema di  $r1$  = schema di  $r2$
- **grado**:  $grado(r1) = grado(r2)$
- **cardinalità**:
  - $\geq \max(0, card(r1) - card(r2))$
  - $\leq card(r1)$

# Differenza: esempio

## IMPIEGATI

## STUDENTI

<b>Cognome</b>	<b>Nome</b>
Bianchi	Carla
Rossi	Marta
Rosa	Antonio

<b>Cognome</b>	<b>Nome</b>
Bianchi	Carla
Verdi	Matteo
Gialli	Maria
Viola	Marco

## STUDENTI — IMPIEGATI

<b>Cognome</b>	<b>Nome</b>
Rossi	Marta
Rosa	Antonio

## IMPIEGATI — STUDENTI

<b>Cognome</b>	<b>Nome</b>
Verdi	Matteo
Gialli	Maria
Viola	Marco

# Intersezione

$$r1 \cap r2 \equiv r1 - (r1 - r2)$$

- $r1$  e  $r2$ : relazioni sullo stesso insieme di attributi

Il risultato è una relazione senza nome che contiene tutte le tuple che appartengono sia a  $r1$  sia a  $r2$

- **schema**: schema di  $r1$  = schema di  $r2$

- **grado**:  $grado(r1) = grado(r2)$

- **cardinalità**:

$$\geq 0$$

$$\leq \min(card(r1), card(r2))$$

# Intersezione: esempio

## STUDENTI

<b>Cognome</b>	<b>Nome</b>
Bianchi	Carla
Rossi	Marta
Rosa	Antonio

## IMPIEGATI

<b>Cognome</b>	<b>Nome</b>
Bianchi	Carla
Verdi	Matteo
Gialli	Maria
Viola	Marco

$STUDENTI \cap IMPIEGATI = IMPIEGATI \cap STUDENTI$

<b>Cognome</b>	<b>Nome</b>
Bianchi	Carla

# Algebra relazionale

- operatori aggiunti:
  - ridenominazione
  - assegnamento
- operatori unari
  - selezione
  - proiezione
- operatori binari
  - unione
  - differenza
  - intersezione
  - prodotto cartesiano
  - join (naturale, theta-join, equi-join)



# Prodotto cartesiano

## $r1 \times r2$

- $r1$  e  $r2$ : relazioni su **insiemi di attributi disgiunti**

Il risultato è una relazione **senza nome** che contiene tutte le tuple ottenute **concatenando tuple di  $r1$  con tuple di  $r2$**

- **schema**: schema di  $r1$  concatenato a quello di  $r2$
- **grado**:  $grado(r1) + grado(r2)$
- **cardinalità**:  $card(r1) * card(r2)$

# Prodotto cartesiano: esempio (1)

## STUDENTI

Cognome	Nome
Bianchi	Carla
Rossi	Marta
Rosa	Antonio

## CORSI

Codice	Titolo
CS101	Basidati
CS102	Sistemi

STUDENTI × CORSI = CORSI × STUDENTI

Cognome	Nome	Codice	Titolo
Bianchi	Carla	CS101	Basidati
Bianchi	Carla	CS102	Sistemi
Rossi	Marta	CS101	Basidati
Rossi	Marta	CS102	Sistemi
Rosa	Antonio	CS101	Basidati
Rosa	Antonio	CS102	Sistemi

# Prodotto cartesiano: esempio (2)

## CORSI

<b>Corso</b>	<b>Docente</b>	<b>Aula</b>
Architetture	Rossi	Alfa
Basi di dati	Neri	Beta

## AULE

<b>Nome</b>	<b>Piano</b>
Alfa	Terra
Beta	Primo

$$\text{CORSI} \times \text{AULE} = \text{AULE} \times \text{CORSI}$$

<b>Corso</b>	<b>Docente</b>	<b>Aula</b>	<b>Nome</b>	<b>Piano</b>
Architetture	Rossi	Alfa	Alfa	Terra
Architetture	Rossi	Alfa	Beta	Primo
Basi di dati	Neri	Beta	Alfa	Terra
Basi di dati	Neri	Beta	Beta	Primo

# Join (theta) (1)

$r1 \bowtie_{\text{condizione}} r2$

- $r1$  e  $r2$ : relazioni su **insiemi di attributi disgiunti**
- *condizione*: **espressione booleana** ( $\wedge, \vee, \neg$ ) di condizioni atomiche del tipo  $A \theta B$ 
  - $A$  è un attributo di  $r1$
  - $B$  è un attributo di  $r2$
  - $\theta$  è un operatore di confronto ( $=, \neq, <, >, \leq, \geq$ ) che abbia senso per  $A$  e  $B$

**Equi-join**: join in cui *condizione* è una **congiunzione** ( $\wedge$ ) di **atomi di uguaglianza** ( $A=B$ )

## Join (theta) (2)

$$r1 \bowtie_{\text{condizione}} r2 \quad \equiv \quad \sigma_{\text{condizione}} (r1 \times r2)$$

Il risultato è una relazione **senza nome** che contiene le **tuple del prodotto cartesiano** fra *r1* e *r2* che **soddisfano *condizione***

- **schema**: schema di *r1* concatenato a quello di *r2*
- **grado**:  $\text{grado}(r1) + \text{grado}(r2)$
- **cardinalità**:  $\geq 0$   
 $\leq (\text{card}(r1) * \text{card}(r2))$

# Join (theta): esempio (1)

## CORSI

Corso	Docente	Aula
Architetture	Rossi	Alfa
Basi di dati	Neri	Beta

## AULE

Nome	Piano
Alfa	Terra
Beta	Primo

$\text{CORSI} \bowtie_{\text{Aula=Nome}} \text{AULE} = \text{AULE} \bowtie_{\text{Nome=Aula}} \text{CORSI}$

Corso	Docente	Aula	Nome	Piano
Architetture	Rossi	Alfa	Alfa	Terra
Basi di dati	Neri	Beta	Beta	Primo

# Join (theta): esempio (2)

**STUDENTI**

Matr	Cognome
11111	Bianchi
22222	Rossi
33333	Verdi

**ESAMI**

Stud	Esame
11111	CS101
22222	CS102
33333	CS101

**CORSI**

Codice	Titolo
CS101	Basidati
CS102	Sistemi

STUDENTI ⋈<sub>Matr=Stud</sub> ESAMI ⋈<sub>Esame=Codice</sub> CORSI

Matr	Cognome	Stud	Esame	Codice	Titolo
11111	Bianchi	11111	CS101	CS101	Basidati
22222	Rossi	22222	CS102	CS102	Sistemi
33333	Verdi	33333	CS101	CS101	Basidati

# Join naturale (1)

$r1 \bowtie r2$

- $r1$  e  $r2$ : relazioni che potrebbero avere un insieme di attributi in comune
  - $X_1$ : attributi di  $r1$
  - $X_2$ : attributi di  $r2$
  - $X_{12} = X_1 \cap X_2$

Il risultato è una relazione **senza nome** che contiene le **tuple ottenute combinando le tuple delle due relazioni sugli attributi con ugual nome** (eliminando le colonne doppie)



## Join naturale (2)

$r1 \bowtie r2$

Il risultato è una relazione **senza nome** che contiene le **tuple ottenute combinando le tuple delle due relazioni sugli attributi con ugual nome** (eliminando le colonne doppie)

- **schema**: schema di  $r1$  concatenato a quello di  $r2$ 
  - attributi con ugual nome compaiono una sola volta

- **grado**:  $grado(r1) + grado(r2) - |X_{12}|$

- **cardinalità**:  $\geq 0$

$$\leq card(r1) * card(r2)$$

$\leq max(card(r1), card(r2))$  se  $X_{12}$  è chiave per almeno una fra  $r1$  e  $r2$

## Join naturale (3)

$r1 \bowtie r2$

Più precisamente:

- Se  $X_{12}$  contiene una **chiave per  $r2$** , allora il join di  $r1$  e  $r2$  **contiene al più  $|r1|$  tuple**
- Se  $X_{12}$  contiene una **chiave per  $r2$** , e sussiste il **vincolo di integrità referenziale** fra  $X_{12}$  (o un suo sottoinsieme) in  $r1$  e la chiave di  $r2$ , allora il join di  $r1$  e  $r2$  **contiene esattamente  $|r1|$  tuple**

# Join naturale: esempio (1)

## CORSI

<b>Corso</b>	<b>Docente</b>	<b>Aula</b>
Architetture	Rossi	Alfa
Basi di dati	Neri	Beta

## AULE

<b>Aula</b>	<b>Piano</b>
Alfa	Terra
Beta	Primo

CORSI  $\bowtie$  AULE = AULE  $\bowtie$  CORSI

<b>Corso</b>	<b>Docente</b>	<b>Aula</b>	<b>Piano</b>
Architetture	Rossi	Alfa	Terra
Basi di dati	Neri	Beta	Primo

# Join naturale: esempio (2)

## STUDENTI

Matr	Cognome
11111	Bianchi
22222	Rossi
33333	Verdi

## ESAMI

Matr	Codice
11111	CS101
22222	CS102
33333	CS101

## CORSI

Codice	Titolo
CS101	Basidati
CS102	Sistemi

STUDENTI ⋈ ESAMI ⋈ CORSI

Matr	Cognome	Codice	Titolo
11111	Bianchi	CS101	Basidati
22222	Rossi	CS102	Sistemi
33333	Verdi	CS101	Basidati

# Join naturale (3)

$r1 \bowtie r2$

- esprimibile come proiezione, equi-join, e ridenominazione
  - ridenominiamo gli attributi comuni
  - equi-join chiedendo l'uguaglianza degli attributi ex-comuni
  - proiezione per eliminare le colonne "doppie"

## Esempio

$(A, B, C)$ : attributi di  $r1$

$(B, C, D)$ : attributi di  $r2$

$$r1 \bowtie r2 = \Pi_{ABCD} (r1 \bowtie_{B=B' \wedge C=C'} (\rho_{B'C' \leftarrow BC} (r2)))$$

## Join naturale (4)

$r1 \bowtie r2$

- $X_1$ : attributi di  $r1$
- $X_2$ : attributi di  $r2$
- $X_{12} = X_1 \cap X_2$

**due casi estremi:**

- $X_1 = X_2, X_{12} = X_1 = X_2 \Rightarrow r1 \bowtie r2 = r1 \cap r2$
- $X_{12} = \emptyset \Rightarrow r1 \bowtie r2 = r1 \times r2$

Esistono altre varianti del join naturale:

- **semijoin**: proietta il join su una relazione
- **outer join**: include le tuple che non hanno corrispondenza nell'altra relazione concatenandole con una tupla di valori nulli

Appartengono propriamente a SQL, li vedremo in quel contesto

# Algebra relazionale

## Linguaggio procedurale:

- l'ordine con cui vengono eseguite le operazioni ha effetto sul costo di esecuzione
- è importante scrivere le espressioni in modo da ottimizzare i costi (limitare le dimensioni dei risultati intermedi)

**Query SQL vengono tradotte dal DBMS in espressioni algebriche che vengono quindi ottimizzate (dal DBMS)**



# Equivalenza di espressioni

Una query può essere scritta secondo diverse espressioni algebriche

Due espressioni algebriche  $E1$  e  $E2$  che **producono lo stesso risultato** (indipendentemente dalle specifiche istanze delle relazioni coinvolte) si dicono **equivalenti** ( $E1 \equiv E2$ )

**Trasformazioni di equivalenza** permettono di trasformare una espressione in una espressione equivalente (meno costosa)

# Trasformazioni di equivalenza

- Atomizzazione delle selezioni
- Idempotenza delle proiezioni
- Anticipazione delle selezioni rispetto al join
- Anticipazione delle proiezioni rispetto al join
- Inglobamento della selezione in un prodotto cartesiano
- Distributività della selezione rispetto all' unione
- Distributività della selezione rispetto alla differenza
- Distributività della proiezione rispetto all' unione
- Distributività del join rispetto all' unione
- Corrispondenza fra operatori insiemistici e selezioni
- Commutatività e associatività di tutti gli operatori binari ad eccezione della differenza

# Atomizzazione delle selezioni

$$\sigma_{C1 \wedge C2}(E) \equiv \sigma_{C1}(\sigma_{C2}(E))$$

-  $E$  espressione;  $C1, C2$  condizioni

(È una trasformazione preliminare ad altre)

## Esempio

STUDENTI(Matr, Cognome, Nome)

$$\sigma_{\text{Cognome} = \text{'Rossi'} \wedge \text{Nome} = \text{'Mario'}}(\text{STUDENTI}) \equiv$$

$$\sigma_{\text{Cognome} = \text{'Rossi'}}(\sigma_{\text{Nome} = \text{'Mario'}}(\text{STUDENTI}))$$

# Idempotenza delle proiezioni

$$\Pi_X(E) \equiv \Pi_X(\Pi_{XY}(E))$$

–  $E$  espressione;  $X$  e  $Y$  insiemi di attributi

(È una trasformazione preliminare ad altre)

## Esempio

STUDENTI(Matr,Cognome,Nome)

$$\Pi_{\text{Cognome}}(\text{STUDENTI}) \equiv \Pi_{\text{Cognome}}(\Pi_{\text{Cognome, Nome}}(\text{STUDENTI}))$$

# Anticipazione della selezione sul join

$$\sigma_C(E_1 \bowtie E_2) \equiv E_1 \bowtie \sigma_C(E_2)$$

- $E_1$  e  $E_2$  espressioni;  $C$  condizione che fa riferimento solo ad attributi di  $E_2$

## Esempio

STUDENTI(Matr,Cognome,Nome)

ESAMI(Matr,Corso,Voto)

$$\sigma_{\text{Corso}='Basidati'}(\text{STUDENTI} \bowtie \text{ESAMI}) \equiv$$

$$\text{STUDENTI} \bowtie \sigma_{\text{Corso}='Basidati'}(\text{ESAMI})$$

# Anticipazione della proiezione sul join (1)

$$\Pi_{X1Y2} (E_1 \bowtie E_2) \equiv E_1 \bowtie \Pi_{Y2}(E_2)$$

- $E_1$  e  $E_2$  espressioni;  $X1$  e  $X2$  attributi di  $E_1$  e  $E_2$ ;  
 $Y2 \supseteq (X1 \cap X2)$  insieme di attributi che include *tutti* gli attributi di  $E_2$  coinvolti nel join

## Esempio

STUDENTI(Matr,Cognome,Nome)

ESAMI(Matr,Corso,Voto)

$$\Pi_{\text{Matr,Cognome,Nome,Corso}}(\text{STUDENTI} \bowtie \text{ESAMI}) \equiv$$

$$\text{STUDENTI} \bowtie \Pi_{\text{Matr,Corso}}(\text{ESAMI})$$

# Anticipazione della proiezione sul join (2)

Combinando l'idempotenza delle proiezioni e l'anticipazione della proiezione rispetto al join

$$\Pi_Y (E_1 \bowtie_C E_2) \equiv \Pi_Y(\Pi_{Y1}(E_1) \bowtie_C \Pi_{Y2}(E_2))$$

- $E_1$  e  $E_2$  espressioni;  $X1$  e  $X2$  attributi di  $E_1$  e  $E_2$ ;  $J1$  e  $J2$  attributi di  $E_1$  e  $E_2$  coinvolti nel join;
- $Y1 = (X1 \cap Y) \cup J1$ ;  $Y2 = (X2 \cap Y) \cup J2$ ;

(Possiamo eliminare subito gli attributi non coinvolti nel join)

## Esempio

STUDENTI(Matr, Cognome, Nome); ESAMI(Matr, Corso, Voto)

$$\begin{aligned} \Pi_{\text{Cognome, Corso}}(\text{STUDENTI} \bowtie \text{ESAMI}) &\equiv \\ \Pi_{\text{Cognome, Corso}}(\Pi_{\text{Matr, Cognome}}(\text{STUDENTI}) \bowtie \Pi_{\text{Matr, Corso}}(\text{ESAMI})) & \end{aligned}$$

# Inglobamento della selezione in $\times$

$$\sigma_C(E_1 \times E_2) \equiv E_1 \bowtie_C(E_2)$$

- $E_1$  e  $E_2$  espressioni;  $C$  condizione che fa riferimento ad attributi di  $E_1$  e  $E_2$

## Esempio

STUDENTI(Matr,Cognome,Nome)

ESAMI(Matr-st,Corso,Voto)

$$\sigma_{\text{Matr}=\text{Matr-st}}(\text{STUDENTI} \times \text{ESAMI}) \equiv \text{STUDENTI} \bowtie_{\text{Matr}=\text{Matr-st}}(\text{ESAMI})$$



# Distributività della selezione rispetto a $\cup$

$$\sigma_C(E_1 \cup E_2) \equiv \sigma_C(E_1) \cup \sigma_C(E_2)$$

-  $E_1$  e  $E_2$  espressioni;  $C$  condizione

## Esempio

STUDENTI(C.F., Cognome, Nome)

IMP(C.F., Cognome, Nome)

$$\sigma_{\text{Cognome}='Rossi'}(\text{STUDENTI} \cup \text{IMP}) \equiv$$

$$\sigma_{\text{Cognome}='Rossi'}(\text{STUDENTI}) \cup \sigma_{\text{Cognome}='Rossi'}(\text{IMP})$$

# Distributività della selezione rispetto a –

$$\sigma_C(E_1 - E_2) \equiv \sigma_C(E_1) - \sigma_C(E_2)$$

–  $E_1$  e  $E_2$  espressioni;  $C$  condizione

(Ne deriva anche distributività rispetto a  $\cap$ )

## Esempio

STUDENTI(C.F., Cognome, Nome)

IMP(C.F., Cognome, Nome)

$$\sigma_{\text{Cognome}='Rossi'}(\text{STUDENTI} - \text{IMP}) \equiv$$

$$\sigma_{\text{Cognome}='Rossi'}(\text{STUDENTI}) - \sigma_{\text{Cognome}='Rossi'}(\text{IMP})$$

# Distributività della proiezione rispetto a $\cup$

$$\Pi_X(E_1 \cup E_2) \equiv \Pi_X(E_1) \cup \Pi_X(E_2)$$

–  $E_1$  e  $E_2$  espressioni;  $X$  attributi in  $E_1$  e  $E_2$

(NOTA: Non vale distributività della proiezione rispetto a  $-$  e  $\cap$  )

## Esempio

STUDENTI(C.F., Cognome, Nome)

IMP(C.F., Cognome, Nome)

$$\Pi_{\text{Cognome, Nome}} (\text{STUDENTI} \cup \text{IMP}) \equiv$$

$$\Pi_{\text{Cognome, Nome}} (\text{STUDENTI}) \cup \Pi_{\text{Cognome, Nome}} (\text{IMP})$$

# Distributività del join rispetto a $\cup$

$$E \bowtie (E_1 \cup E_2) \equiv (E \bowtie E_1) \cup (E \bowtie E_2)$$

-  $E, E_1, E_2$  espressioni

## Esempio

STUDENTI(C.F., Cognome, Nome)

IMP(C.F., Cognome, Nome); TASSE(C.F., Somma)

TASSE  $\bowtie$  (STUDENTI  $\cup$  IMP)  $\equiv$

(TASSE  $\bowtie$  STUDENTI)  $\cup$  (TASSE  $\bowtie$  IMP)

# Corrispondenza fra op. insiemistici e $\sigma$

$$\sigma_{C1 \vee C2}(E) \equiv \sigma_{C1}(E) \cup \sigma_{C2}(E)$$

$$\sigma_{C1 \wedge C2}(E) \equiv \sigma_{C1}(E) \cap \sigma_{C2}(E) \equiv \sigma_{C1}(E) \bowtie \sigma_{C2}(E)$$

$$\sigma_{C1 \wedge \neg C2}(E) \equiv \sigma_{C1}(E) - \sigma_{C2}(E)$$

## Esempi

ESAMI(Matr, Corso, Voto)

$$\sigma_{\text{Corso} = \text{'Sistemi'} \vee \text{Voto} = 30}(\text{ESAMI}) \equiv$$

$$\sigma_{\text{Corso} = \text{'Sistemi'}}(\text{ESAMI}) \cup \sigma_{\text{Voto} = 30}(\text{ESAMI})$$

$$\sigma_{\text{Corso} = \text{'Sistemi'} \wedge \text{Voto} = 30}(\text{ESAMI}) \equiv$$

$$\sigma_{\text{Corso} = \text{'Sistemi'}}(\text{ESAMI}) \cap \sigma_{\text{Voto} = 30}(\text{ESAMI})$$

$$\sigma_{\text{Corso} = \text{'Sistemi'} \wedge \neg(\text{Voto} = 30)}(\text{ESAMI}) \equiv$$

$$\sigma_{\text{Corso} = \text{'Sistemi'}}(\text{ESAMI}) - \sigma_{\text{Voto} = 30}(\text{ESAMI})$$

# Associatività e commutatività op. binari

$$E_1 \text{ op } (E_2 \text{ op } E_3) \equiv (E_1 \text{ op } E_2) \text{ op } E_3$$

$$E_1 \text{ op } E_2 \equiv E_2 \text{ op } E_1$$

- $op \in \{\cup, \cap, \times, \bowtie\}$

## Esempi

STUDENTI(Matr,Cognome,Nome); ESAMI(Matr,Corso,Voto)  
CORSI(Corso,Docente)

STUDENTI  $\bowtie$  (ESAMI  $\bowtie$  CORSI)  $\equiv$

(STUDENTI  $\bowtie$  ESAMI)  $\bowtie$  CORSI

STUDENTI  $\bowtie$  ESAMI  $\equiv$  ESAMI  $\bowtie$  STUDENTI

# Alcune formule utili

- $r \bowtie r \equiv r$
- $r \cup r \equiv r$
- $r \cap r \equiv r$
- $r - r \equiv \emptyset$
- $r \bowtie \sigma_C r \equiv \sigma_C r$
- $r \cup \sigma_C r \equiv r$
- $r \cap \sigma_C r \equiv \sigma_C r$
- $r - \sigma_C r \equiv \sigma_{\neg C} r$

- $r \bowtie \emptyset \equiv \emptyset$
- $r \cup \emptyset \equiv r$
- $r \cap \emptyset \equiv \emptyset$
- $r - \emptyset \equiv r$
- $\emptyset - r \equiv \emptyset$
- $r \times \emptyset \equiv \emptyset$
- $\sigma_C \emptyset \equiv \emptyset$
- $\Pi_X \emptyset \equiv \emptyset$

# Idiomi di interrogazione

- esiste/per ogni
- minimo/massimo
  - assoluto
  - relativo
  - rispondente a
- cardinalità
- copertura



# Esiste/per ogni

$r(A, B)$

- trovare gli  $A$  per cui **esiste un**  $B$  che soddisfa la condizione  $p(B)$

$$\Pi_A(\sigma_{p(B)}(r))$$

- trovare gli  $A$  per i quali **tutti** i  $B$  soddisfano la condizione  $p(B)$

$$\Pi_A(r) - \Pi_A(\sigma_{\neg p(B)}(r))$$

# Esiste: esempio

## REGISTRI

Matr	Corso	Data	Voto
11111	Basidati	20-01-05	28
11111	Sistemi	15-06-05	30
22222	Basidati	21-11-05	27
22222	Sistemi	15-03-05	25
33333	Basidati	16-05-05	30

Trovare gli studenti che hanno preso più di 26 in almeno un esame

$\Pi_{\text{Matr}}(\sigma_{\text{Voto} > 26} \text{REGISTRI})$

Matr
11111
22222
33333

# Per ogni: esempio

## REGISTRI

Matr	Corso	Data	Voto
11111	Basidati	20-01-05	28
11111	Sistemi	15-06-05	30
22222	Basidati	21-11-05	27
22222	Sistemi	15-03-05	25
33333	Basidati	16-05-05	30

Trovare gli studenti che hanno preso più di 26 in tutti gli esami

$$\Pi_{\text{Matr}}(\text{REGISTRI}) - \Pi_{\text{Matr}}(\sigma_{\text{Voto} \leq 26} \text{REGISTRI})$$

Matr
11111
33333

# Minimo e massimo assoluto

$r(A, B)$

- trovare il **minimo** (**massimo**) valore di  $B$

**minimo:**

$$\begin{aligned} val &:= \mathbf{\Pi}_B(r) \\ val &- \mathbf{\Pi}_B(val \bowtie_{B > B'} (\rho_{B' \leftarrow B}(val))) \end{aligned}$$

**massimo:**

$$\begin{aligned} val &:= \mathbf{\Pi}_B(r) \\ val &- \mathbf{\Pi}_B(val \bowtie_{B < B'} (\rho_{B' \leftarrow B}(val))) \end{aligned}$$

# Minimo assoluto: esempio

## REGISTRI

Matr	Corso	Data	Voto
11111	Basidati	20-01-05	28
11111	Sistemi	15-06-05	30
22222	Basidati	21-11-05	27
22222	Sistemi	15-03-05	25
33333	Basidati	16-05-05	30

Trovare il voto minimo

$$\text{VOTI} := \rho_{V \leftarrow \text{Voto}}(\Pi_{\text{Voto}}(\text{REGISTRI}))$$

$$\text{VOTI} - \Pi_V(\text{VOTI} \bowtie_{V > V'} (\rho_{V' \leftarrow V} \text{VOTI}))$$

<b>v</b>
25

# Massimo assoluto: esempio

## REGISTRI

Matr	Corso	Data	Voto
11111	Basidati	20-01-05	28
11111	Sistemi	15-06-05	30
22222	Basidati	21-11-05	27
22222	Sistemi	15-03-05	25
33333	Basidati	16-05-05	30

Trovare la data più recente

$DATE := \rho_{D \leftarrow Data}(\Pi_{Data}(REGISTRI))$

$DATE - \Pi_D(DATE \bowtie_{D < D'} (\rho_{D' \leftarrow D} DATE))$

D
21-11-05

# Minimo e massimo relativo

$r(A, B)$

- trovare, per ogni  $A$ , il **minimo** (**massimo**) valore di  $B$

**minimo:**

$$r - \prod_{A,B} (r \bowtie_{A=A' \wedge B>B'} (\rho_{A',B' \leftarrow A,B}(r)))$$

**massimo:**

$$r - \prod_{A,B} (r \bowtie_{A=A' \wedge B<B'} (\rho_{A',B' \leftarrow A,B}(r)))$$

# Minimo relativo: esempio

## REGISTRI

Matr	Corso	Data	Voto
11111	Basidati	20-01-05	28
11111	Sistemi	15-06-05	30
22222	Basidati	21-11-05	27
22222	Sistemi	15-03-05	25
33333	Basidati	16-05-05	30

Trovare per ogni studente il voto minimo

$$\text{VOTI} := \rho_{M,V \leftarrow \text{Matr}, \text{Voto}} (\Pi_{\text{Matr}, \text{Voto}} (\text{REGISTRI}))$$

$$\text{VOTI} - \Pi_{M,V} (\text{VOTI} \bowtie_{M=M' \wedge V > V'} (\rho_{M',V' \leftarrow M,V} \text{VOTI}))$$

M	V
11111	28
22222	25
33333	30



# Massimo relativo: esempio

## REGISTRI

Matr	Corso	Data	Voto
11111	Basidati	20-01-05	28
11111	Sistemi	15-06-05	30
22222	Basidati	21-11-05	27
22222	Sistemi	15-03-05	25
33333	Basidati	16-05-05	30

Trovare per ogni corso la data più recente

$ESAMI := \rho_{C,D \leftarrow \text{Corso, Data}}(\Pi_{\text{Corso, Data}}(\text{REGISTRI}))$

$ESAMI - \Pi_{C,D} (ESAMI \bowtie_{C=C' \wedge D < D'} (\rho_{C', D' \leftarrow C, D} ESAMI))$

C	D
Basidati	21-11-05
Sistemi	15-06-05

# Rispondente a minimo e massimo

$r(A, B)$

- trovare gli  $A$  per cui  $B$  ha il valore **minimo** (**massimo**)

**minimo:**

$$\Pi_A (r - \Pi_{A,B}(r \triangleright_{B>B'} (\rho_{A',B' \leftarrow A,B} (r)))$$

**massimo:**

$$\Pi_A (r - \Pi_{A,B}(r \triangleleft_{B<B'} (\rho_{A',B' \leftarrow A,B} (r)))$$

# Rispondente a minimo: esempio

## REGISTRI

Matr	Corso	Data	Voto
11111	Basidati	20-01-05	28
11111	Sistemi	15-06-05	30
22222	Basidati	21-11-05	27
22222	Sistemi	15-03-05	25
33333	Basidati	16-05-05	30

Trovare gli studenti che hanno preso il voto più basso

$$\begin{aligned}
 \text{VOTI} &:= \rho_{M, V \leftarrow \text{Matr}, \text{Voto}}(\Pi_{\text{Matr}, \text{Voto}}(\text{REGISTRI})) \\
 \Pi_M(\text{VOTI} - \Pi_{M, V}(\text{VOTI} \bowtie_{V > V'}(\rho_{M', V' \leftarrow M, V} \text{VOTI})))
 \end{aligned}$$

<b>M</b>
22222

# Rispondente a massimo: esempio

## REGISTRI

Matr	Corso	Data	Voto
11111	Basidati	20-01-05	28
11111	Sistemi	15-06-05	30
22222	Basidati	21-11-05	27
22222	Sistemi	15-03-05	25
33333	Basidati	16-05-05	30

Trovare il corso con la data più recente

$$\begin{aligned}
 \text{ESAMI} &:= \rho_{C,D \leftarrow \text{Corso, Data}}(\Pi_{\text{Corso, Data}}(\text{REGISTRI})) \\
 &\Pi_C(\text{ESAMI} - \Pi_{C,D}(\text{ESAMI} \bowtie_{D < D'} (\rho_{C', D' \leftarrow C, D} \text{ESAMI})))
 \end{aligned}$$

<b>C</b>
Basidati

# Idiomi di interrogazione

- esiste/per ogni
- minimo/massimo
  - assoluto
  - relativo
  - rispondente a
- cardinalità
- copertura

# Cardinalità (1)

$r(A, B)$

- trovare gli  $A$  che sono associati a

almeno due  $B$

$$\Pi_A(r \bowtie_{A=A' \wedge B \neq B'} (\rho_{A', B' \leftarrow A, B}(r)))$$

al più uno  $B$

$$\Pi_A(r) - \Pi_A(r \bowtie_{A=A' \wedge B \neq B'} (\rho_{A', B' \leftarrow A, B}(r)))$$

# Cardinalità: esempio

## REGISTRI

Matr	Corso	Data	Voto
11111	Basidati	20-01-05	28
11111	Sistemi	15-06-05	30
22222	Basidati	21-11-05	27
22222	Sistemi	15-03-05	25
33333	Basidati	16-05-05	30

Trovare gli studenti che hanno fatto almeno due esami

$$\text{ESAMI} := \rho_{M,C \leftarrow \text{Matr, Corso}} (\Pi_{\text{Matr, Corso}} \text{REGISTRI})$$
$$\Pi_M (\text{ESAMI} \bowtie_{M=M' \wedge C \neq C'} (\rho_{M',C' \leftarrow M,C} (\text{ESAMI})))$$

M
11111
22222

# Cardinalità: esempio

## REGISTRI

Matr	Corso	Data	Voto
11111	Basidati	20-01-05	28
11111	Sistemi	15-06-05	30
22222	Basidati	21-11-05	27
22222	Sistemi	15-03-05	25
33333	Basidati	16-05-05	30

Trovare gli studenti che hanno fatto al più un esame:

$$\begin{aligned} R1 &:= \rho_{M,C \leftarrow \text{Matr, Corso}} (\Pi_{\text{Matr, Corso}} \text{REGISTRI}) \\ \text{ESAME} &:= \Pi_M (R1 \bowtie_{M=M' \wedge C \neq C'} (\rho_{M', C' \leftarrow M, C} (R1))) \\ &(\Pi_M R1) - \text{ESAME} \end{aligned}$$

M
33333



## Cardinalità (2)

$r(A, B)$

- trovare gli  $A$  che sono associati a:

- almeno tre  $B$

$$\prod_A (\sigma_{A=A' \wedge A=A'' \wedge B \neq B' \wedge B \neq B'' \wedge B' \neq B''})$$

$$(r \times \rho_{A', B' \leftarrow A, B}(r) \times \rho_{A'', B'' \leftarrow A, B}(r))$$

- almeno  $n$   $B$

(con  $n-1$  prodotti cartesiani)

- al più  $n$   $B$

$\prod_A(r)$  – “ $A$  associati ad almeno  $n+1$   $B$ ”

- esattamente  $n$   $B$

“ $A$  associati ad almeno  $n$   $B$ ” –

“ $A$  associati ad almeno  $n+1$   $B$ ”

# Copertura

$r(A, B)$  e  $s(B)$

- trovare gli  $A$  che sono associati a **tutti** i valori di  $B$  in  $s$

$$\Pi_A(r) - \Pi_A ((\Pi_A(r) \times s) - r)$$

# Copertura: esempio

## ESAMI

Matr	Esame
11111	CS101
22222	CS102
11111	CS102

## CORSI

Esame	Titolo
CS101	Basidati
CS102	Sistemi

Trovare la matricola degli studenti che hanno fatto tutti gli esami

$$CO := \Pi_{\text{Esame}}(\text{CORSI})$$

$$\Pi_{\text{Matr}}(\text{ESAMI}) - \Pi_{\text{Matr}}((\Pi_{\text{Matr}}(\text{ESAMI}) \times CO) - \text{ESAMI})$$

Matr
11111



# Algebra relazionale – Esercizi

# Interrogazione 1

CELLULARE (Codice, CFUtente, Modello, Marca, Colore)

ABBONAMENTO(Numero, CFUtente, Operatore, Tariffa)

UTENTE(CF, Nome, Cognome, Città)

- determinare il codice dei cellulari di marca Nokia di colore rosso oppure nero

$\Pi_{\text{Codice}}(\sigma_{(\text{Colore}='rosso' \vee \text{Colore}='nero') \wedge \text{Marca}='Nokia'} \text{CELLULARE})$

# Interrogazione 2

CELLULARE (Codice, CFUtente, Modello, Marca, Colore)

ABBONAMENTO(Numero, CFUtente, Operatore, Tariffa)

UTENTE(CF, Nome, Cognome, Città)

- determinare il codice dei cellulari degli utenti residenti a Milano

$UT\_MIL := \Pi_{CF}(\sigma_{Città='Milano'} UTENTE)$

$CELL := \Pi_{Codice, CFUtente}(CELLULARE)$

$\Pi_{Codice}(UT\_MIL \bowtie_{CF=CFUtente} CELL)$

# Interrogazione 3 (1)

CELLULARE (Codice, CFUtente, Modello, Marca, Colore)

ABBONAMENTO(Numero, CFUtente, Operatore, Tariffa)

UTENTE(CF, Nome, Cognome, Città)

- determinare la marca dei cellulari degli utenti serviti da Tim nella città di Milano

$UT\_TIM := \Pi_{CFUtente}(\sigma_{Operatore='TIM'} ABBONAMENTO)$

$UT\_MIL := \rho_{CFUtente \leftarrow CF}(\Pi_{CF}(\sigma_{Città='Milano'} UTENTE))$

$\Pi_{Marca}((UT\_TIM \cap UT\_MIL) \bowtie CELLULARE)$

## Interrogazione 3 (2)

CELLULARE (Codice, CFUtente, Modello, Marca, Colore)

ABBONAMENTO(Numero, CFUtente, Operatore, Tariffa)

UTENTE(CF, Nome, Cognome, Città)

- determinare la marca dei cellulari degli utenti serviti da Tim nella città di Milano

$UT\_TIM := \Pi_{CFUtente}(\sigma_{Operatore='TIM'}(ABBONAMENTO))$

$UT\_MIL := \Pi_{CF}(\sigma_{Città='Milano'}(UTENTE))$

$CELL\_TIM := \Pi_{CFUtente, Marca}(UT\_TIM \bowtie CELLULARE)$

$CELL\_MIL := \Pi_{CFUtente, Marca}(UT\_MIL \bowtie_{CF=CFUtente} CELLULARE)$

$\Pi_{Marca}(CELL\_TIM \cap CELL\_MIL)$



# Interrogazione 4

CELLULARE (Codice, CFUtente, Modello, Marca, Colore)

ABBONAMENTO(Numero, CFUtente, Operatore, Tariffa)

UTENTE(CF, Nome, Cognome, Città)

- determinare gli operatori che servono tutti gli utenti della città di Milano

$UT\_MIL := \rho_{CFUtente \leftarrow CF} ( \Pi_{CF} ( \sigma_{Città='Milano'} UTENTE ) )$

$OP := \Pi_{Operatore}(ABBONAMENTO)$

$TUTTI := UT\_MIL \times OP$

$ABB\_MIL := \Pi_{CFUtente, Operatore}(UT\_MIL \bowtie ABBONAMENTO)$

$\Pi_{Operatore}(ABB\_MIL) - \Pi_{Operatore}(TUTTI - ABB\_MIL)$

# Interrogazione 5

CELLULARE (Codice, CFUtente, Modello, Marca, Colore)

ABBONAMENTO(Numero, CFUtente, Operatore, Tariffa)

UTENTE(CF, Nome, Cognome, Città)

- determinare la tariffa minima applicata da Tim

$TAR\_TIM := \Pi_{Tariffa}(\sigma_{Operatore='Tim'} ABBONAMENTO)$

$TAR\_TIM - \Pi_{Tariffa}(TAR\_TIM \bowtie_{Tariffa > Tariffa'} (\rho_{Tariffa'} \leftarrow_{Tariffa} TAR\_TIM))$

# Interrogazione 6

CELLULARE (Codice, CFUtente, Modello, Marca, Colore)

ABBONAMENTO(Numero, CFUtente, Operatore, Tariffa)

UTENTE(CF, Nome, Cognome, Città)

- determinare gli utenti (CF, Nome, Cognome e Città) per i quali tutti i numeri di telefono loro intestati hanno tariffa maggiore di 100

$UT\_MIN := \Pi_{CFUtente}(\sigma_{Tariffa \leq 100} ABBONAMENTO)$

$UT\_RIS := \Pi_{CFUtente}(ABBONAMENTO) - UT\_MIN$

$\Pi_{CF, Nome, Cognome, Città}(UT\_RIS \bowtie_{CFUtente=CF} UTENTE)$

# Interrogazione 7

RADIO (Codice, Nome, Frequenza, Luogo)

PROGRAMMA (Codice, Nome, Conduttore, FasciaOraria, Durata, Tipo, CodiceRadio)

- determinare i nomi delle radio che trasmettono programmi non musicali nella fascia del mattino e della sera ma non nella fascia del pomeriggio

$\text{MAT} := \Pi_{\text{CodiceRadio}} (\sigma_{\text{Tipo} \neq \text{'musicale'} \wedge \text{FasciaOraria} = \text{'mattina'}} \text{PROGRAMMA})$

$\text{SER} := \Pi_{\text{CodiceRadio}} (\sigma_{\text{Tipo} \neq \text{'musicale'} \wedge \text{FasciaOraria} = \text{'sera'}} \text{PROGRAMMA})$

$\text{POM} := \Pi_{\text{CodiceRadio}} (\sigma_{\text{Tipo} \neq \text{'musicale'} \wedge \text{FasciaOraria} = \text{'pomeriggio'}} \text{PROGRAMMA})$

$\text{COD\_RIS} := (\text{MAT} \cap \text{SER}) - \text{POM}$

$\Pi_{\text{Nome}} (\text{RADIO} \bowtie_{\text{Codice} = \text{CodiceRadio}} \text{COD\_RIS})$

# Interrogazione 8

RADIO (Codice, Nome, Frequenza, Luogo)

PROGRAMMA (Codice, Nome, Conduttore, FasciaOraria, Durata, Tipo, CodiceRadio)

- determinare, per ogni radio, i codici dei programmi che hanno durata minima

$\text{PROG} := \Pi_{\text{Codice}, \text{Durata}, \text{CodiceRadio}}(\text{PROGRAMMA})$

$\text{PROG}' := \rho_{C', D', CR' \leftarrow \text{Codice}, \text{Durata}, \text{CodiceRadio}}(\text{PROG})$

$\Pi_{\text{Codice}, \text{CodiceRadio}}(\text{PROG}) -$

$\Pi_{\text{Codice}, \text{CodiceRadio}}(\text{PROG} \bowtie_{\text{CodiceRadio}=\text{CR}' \wedge \text{Durata}>D'}(\text{PROG}'))$

# Interrogazione 9

RADIO (Codice, Nome, Frequenza, Luogo)

PROGRAMMA (Codice, Nome, Conduttore, FasciaOraria, Durata, Tipo, CodiceRadio)

- determinare i nomi delle radio che trasmettono almeno due programmi non musicali

$\text{NO\_MUS} := \Pi_{\text{Codice, CodiceRadio}} (\sigma_{\text{Tipo} \neq \text{'musicale'}} \text{PROGRAMMA})$

$\text{NO\_MUS}' := \rho_{C', CR' \leftarrow \text{Codice, CodiceRadio}} (\text{NO\_MUS})$

$\text{COD\_RIS} := \Pi_{\text{CodiceRadio}} (\text{NO\_MUS} \bowtie_{\text{CodiceRadio} = CR' \wedge \text{Codice} \neq C'} \text{NO\_MUS}')$

$\Pi_{\text{Nome}} (\text{RADIO} \bowtie_{\text{Codice} = \text{CodiceRadio}} \text{COD\_RIS})$

# Interrogazione 10

RADIO (Codice, Nome, Frequenza, Luogo)

PROGRAMMA (Codice, Nome, Conduttore, FasciaOraria, Durata,  
Tipo, CodiceRadio)

- determinare i nomi delle radio che non hanno alcun programma musicale

$MUS := \Pi_{\text{CodiceRadio}} (\sigma_{\text{Tipo}=' \text{musicale}' } \text{PROGRAMMA})$

$NO\_MUS := \Pi_{\text{Codice}} (\text{RADIO}) - \rho_{\text{Codice} \leftarrow \text{CodiceRadio}} (MUS)$

$\Pi_{\text{Nome}} (\text{RADIO} \bowtie \text{NO\_MUS})$

# Interrogazione 11

RADIO (Codice, Nome, Frequenza, Luogo)

PROGRAMMA (Codice, Nome, Conduttore, FasciaOraria, Durata,  
Tipo, CodiceRadio)

- determinare i nomi delle radio che hanno programmi musicali condotti da DjX

$COD\_RIS := \Pi_{CodiceRadio} (\sigma_{Tipo='musicale' \wedge Conduttore='DjX'} PROGRAMMA)$

$\Pi_{Nome} (RADIO \bowtie_{Codice=CodiceRadio} COD\_RIS)$



# Interrogazione 12

RADIO (Codice, Nome, Frequenza, Luogo)

PROGRAMMA (Codice, Nome, Conduttore, FasciaOraria, Durata, Tipo, CodiceRadio)

- determinare i nomi delle radio che trasmettono programmi musicali o quiz

$COD\_RIS := \Pi_{CodiceRadio} (\sigma_{Tipo='musicale' \vee Tipo='quiz'} PROGRAMMA)$

$\Pi_{Nome} (RADIO \bowtie_{Codice=CodiceRadio} COD\_RIS)$

# Interrogazione 13

RADIO (Codice, Nome, Frequenza, Luogo)

PROGRAMMA (Codice, Nome, Conduttore, FasciaOraria, Durata, Tipo, CodiceRadio)

- determinare i nomi delle radio che trasmettono programmi musicali e quiz

MUS :=  $\Pi_{\text{CodiceRadio}} (\sigma_{\text{Tipo}=' \text{musicale}' } \text{PROGRAMMA})$

QUIZ :=  $\Pi_{\text{CodiceRadio}} (\sigma_{\text{Tipo}=' \text{quiz}' } \text{PROGRAMMA})$

COD\_RIS := MUS  $\cap$  QUIZ

$\Pi_{\text{Nome}} (\text{RADIO} \bowtie_{\text{Codice}=\text{CodiceRadio}} \text{COD\_RIS})$

A series of thin, light-brown lines forming an abstract, overlapping geometric pattern in the top-left corner of the page. The lines intersect to create various polygonal shapes, some of which are partially cut off by the edge of the page.

# VINCENZO CALABRÒ

LinkedIn vincenzocalabro

[www.vincenzocalabro.it](http://www.vincenzocalabro.it)