# Calcolo relazionale

# Calcolo relazionale

- **Una famiglia di linguaggi formali**
- **Due tipi principali**
  - **Calcolo delle tuple (TRC, Tuple Relational Calculus)**
  - **Calcolo dei domini (DRC, Domain Relational Calculus)**
- **TRC in due versioni:**
  - **Con tuple ristrette sul range**
  - **Con tuple arbitrarie**
- **Studieremo il TRC con tuple arbitrarie (anche se non è descritto nel libro!)**



# TRC è dichiarativo

- Esprime cosa si vuole nel risultato ma non come ottenerlo
- E' diverso dall'algebra, che è procedurale
- La dichiaratività è una caratteristica tipica dei linguaggi relazionali



# Definizione formale del TRC

- **Forma standard: { *t* | *p(t)* }**
  ***p(t)* è una *formula*, costruita tramite *atomi***
- **Atomi**
  - ***t*** $\in$ **R**
  - ***expr comp expr***
- ***comp è un comparatore: =, <>, >, >=, <, <=***
- ***expr è una espressione che usa costanti e t[A]***
  - ***t[A] è una restrizione sull'attributo A della tupla t***
- ***Esempio: {t: t*** $\in$ **R*}***



# Definizione formale del TRC

- **Regole di costruzione delle formule**
  - **un atomo è una formula**
  - **se *p* è una formula, lo sono anche** $\neg p$ **e *(p)***
  - **se *p1* e *p2* sono formule, lo sono anche** $p1 \wedge p2$, $p1 \vee p2$, $p1 \Rightarrow p2$
  - **se *p* è una formula in cui *s* è una variabile, lo sono anche** $\exists\, s \in R\ (p(s))$, $\forall\, s \in R\ (p(s))$



# Proprietà del TRC

- **Legge di De Morgan**
  $p1 \wedge p2 \equiv \neg (\neg p1 \vee \neg p2)$
- **Corrispondenze tra quantificatori**
  $\forall\, t \in R\ (p(t)) \equiv \neg \exists\, t \in R\ (\neg p(t))$
- ***Definizione di implicazione***
  $p1 \Rightarrow p2 \equiv \neg p1 \vee p2$

## Forme Normali

- ***Dalle tre leggi segue che è possibile scrivere tutte le possibili espressioni senza implicazione e con:***
  - ***Un solo quantificatore***
  - ***Un solo operatore binario***
- ***La forma normale più usata (simile ad SQL) usa quantificatore esistenziale e congiunzione***



## Esempi di TRC

- **Nome degli studenti che hanno preso 30 in "matematica"**

**{ *t* | ∃ *t1* ∈ STUDENTE,**
**∃ *t2* ∈ ESAME,**
**∃ *t3* ∈ CORSO**
**( *t*[Nome]=*t1*[Nome] ∧**
***t1*[Matr]=*t2*[Matr] ∧**
***t2*[CodCorso]=*t3*[CodCorso] ∧**
***t2*[Voto]=30 ∧**
***t3*[Titolo]='matematica') }**



## Esempi di TRC

- **Matricole degli studenti che hanno sostenuto "matematica" ma non "basi di dati"**

**{ *t* | ∃ *t1* ∈ ESAME, ∃ *t2* ∈ CORSO**
**( *t*[Matr]=*t1*[Matr] ∧**
***t1*[CodCorso]=*t2*[CodCorso] ∧**
***t2*[Titolo]='matematica') ∧**
**¬ ( ∃ *t3* ∈ ESAME, ∃ *t4* ∈ CORSO**
**(*t*[Matr]=*t3*[Matr] ∧**
***t3*[CodCorso]=*t4*[CodCorso] ∧**
***t4*[Titolo]='basi di dati'))) }**



## Correttezza

- **Si devono evitare formule *unsafe*:**
  **{ t | t ∉ R } dà un risultato infinito**
- **Si considerano corrette solo formule *indipendenti dal dominio***
  - **la soluzione non dipende dal dominio degli attributi, ma solo dall'istanza del DB**



## AR è esprimibile tramite TRC

**E' sufficiente mostrare che si possono realizzare i cinque operatori fondamentali:**

- **Selezione, $\sigma_{A=1}$ R:**
  **{ *t* | ∃ *t* ∈ R (*t*[A]=1) }**
- **Proiezione, $\Pi_{AC}$ R:**
  **{ *t* | ∃ *t1* ∈ R (*t*[A,C]=*t1*[A,C]) }**



## AR è esprimibile tramite TRC

- **Prodotto cartesiano, R(A,B,C) × S(D,E,F):**
  **{ *t* | ∃ *t1* ∈ R, ∃ *t2* ∈ S**
  **(*t*[A,B,C]=*t1*[A,B,C] ∧**
  ***t*[D,E,F]=*t2*[D,E,F] ) }**
- **Esempio di join, R(A,C) $\bowtie_{A=B}$ S(B,D):**
  **{ *t* | ∃ *t1* ∈ R, ∃ *t2* ∈ S**
  **(*t*[A,C] = *t1*[A,C] ∧**
  ***t*[B,D] = *t2*[B,D] ∧**
  ***t*[A] = *t*[B] ) }**

## AR è esprimibile tramite TRC

- **Unione, R $\cup$ S:**
  $\{ t \mid \exists\, t1 \in R, \exists\, t2 \in S\ ( t = t1 \lor t = t2 ) \}$
- **Differenza, R $-$ S:**
  $\{ t \mid \exists\, t \in R\ ( t \notin S) \}$

13

## Anche TRC è esprimibile tramite AR

- **La prova è più complicata**
- **Si devono escludere espressioni unsafe e dipendenti dal dominio**
- **Sotto queste ipotesi TRC e AR hanno lo stesso potere espressivo**

14