

OPERATORI	BASE	DERIVATI																								
<b>INSIEMISTICI</b> Su relazioni con lo stesso schema $X = \{A_1, \dots, A_n\}$	<table><tr><th colspan="2">UNIONE</th></tr><tr><td>Schema</td><td><math>X</math></td></tr><tr><td>Istanza</td><td><math>r_1 \cup r_2 = \{t \mid t \in r_1 \vee t \in r_2\}</math></td></tr><tr><td>Cardinalità</td><td><math>\max( r_1 ,  r_2 ) \leq  r_1 \cup r_2  \leq  r_1  +  r_2 </math></td></tr></table> <table><tr><th colspan="2">DIFFERENZA</th></tr><tr><td>Schema</td><td><math>X</math></td></tr><tr><td>Istanza</td><td><math>r_1 - r_2 = \{t \mid t \in r_1 \wedge t \notin r_2\}</math></td></tr><tr><td>Cardinalità</td><td><math>0 \leq  r_1 - r_2  \leq  r_1 </math></td></tr></table>	UNIONE		Schema	$X$	Istanza	$r_1 \cup r_2 = \{t \mid t \in r_1 \vee t \in r_2\}$	Cardinalità	$\max( r_1 ,  r_2 ) \leq  r_1 \cup r_2  \leq  r_1  +  r_2 $	DIFFERENZA		Schema	$X$	Istanza	$r_1 - r_2 = \{t \mid t \in r_1 \wedge t \notin r_2\}$	Cardinalità	$0 \leq  r_1 - r_2  \leq  r_1 $	<table><tr><th colspan="2">INTERSEZIONE</th></tr><tr><td>Schema</td><td><math>X</math></td></tr><tr><td>Istanza</td><td><math>r_1 \cap r_2 = \{t \mid t \in r_1 \wedge t \in r_2\} = r_1 - (r_1 - r_2)</math></td></tr><tr><td>Cardinalità</td><td><math>0 \leq  r_1 \cap r_2  \leq \min( r_1 ,  r_2 )</math></td></tr></table>	INTERSEZIONE		Schema	$X$	Istanza	$r_1 \cap r_2 = \{t \mid t \in r_1 \wedge t \in r_2\} = r_1 - (r_1 - r_2)$	Cardinalità	$0 \leq  r_1 \cap r_2  \leq \min( r_1 ,  r_2 )$
	UNIONE																									
Schema	$X$																									
Istanza	$r_1 \cup r_2 = \{t \mid t \in r_1 \vee t \in r_2\}$																									
Cardinalità	$\max( r_1 ,  r_2 ) \leq  r_1 \cup r_2  \leq  r_1  +  r_2 $																									
DIFFERENZA																										
Schema	$X$																									
Istanza	$r_1 - r_2 = \{t \mid t \in r_1 \wedge t \notin r_2\}$																									
Cardinalità	$0 \leq  r_1 - r_2  \leq  r_1 $																									
INTERSEZIONE																										
Schema	$X$																									
Istanza	$r_1 \cap r_2 = \{t \mid t \in r_1 \wedge t \in r_2\} = r_1 - (r_1 - r_2)$																									
Cardinalità	$0 \leq  r_1 \cap r_2  \leq \min( r_1 ,  r_2 )$																									
<b>SPECIFICI</b> Su singole relazioni di schema $X = \{A_1, \dots, A_n\}$	<table><tr><th colspan="2">RIDENOMINAZIONE - Cambia nome agli attributi</th></tr><tr><td>Schema</td><td><math>Y = \{B_1, \dots, B_n\}</math> con <math> Y  =  X </math></td></tr><tr><td>Istanza</td><td><math>\rho_{A_1, \dots, A_n \rightarrow B_1, \dots, B_n}(r_1) = \{t \mid \exists t' \in r_1: \forall i \in [1, \dots, n]: t[B_i] = t'[A_i]\}</math> Forma breve: <math>\rho_{A_1, \dots, A_n \rightarrow A'_1, \dots, A'_n}(r_1) = \rho_{X \rightarrow X'}(r_1)</math></td></tr></table> <table><tr><th colspan="2">SELEZIONE - Estrae delle tuple</th></tr><tr><td>Schema</td><td><math>X</math></td></tr><tr><td>Istanza</td><td><math>\sigma_F(r_1) = \{t \mid t \in r_1 \wedge F(t)\}</math> con <math>F(t)</math> formula proposizionale</td></tr><tr><td>Cardinalità</td><td><math>0 \leq  \sigma_F(r_1)  \leq  r_1 </math></td></tr></table> <table><tr><th colspan="2">PROIEZIONE - Elimina attributi</th></tr><tr><td>Schema</td><td><math>Y = \{A_1, \dots, A_m\} \subseteq X</math></td></tr><tr><td>Istanza</td><td><math>\Pi_Y(r_1) = \{t \mid \exists t' \in r_1: t = t'[Y]\}</math> con <math>t'[Y]</math> tupla <math>t_0</math> su <math>Y</math> tale che <math>\forall A_i \in Y: t_0[A_i] = t'[A_i]</math></td></tr><tr><td>Cardinalità</td><td><math>\min( r_1 , 1) \leq  \Pi_Y(r_1)  \leq  r_1 </math> dove <math> \Pi_Y(r_1)  =  r_1 </math> se <math>Y</math> è superchiave per <math>r_1</math></td></tr></table>	RIDENOMINAZIONE - Cambia nome agli attributi		Schema	$Y = \{B_1, \dots, B_n\}$ con $ Y  =  X $	Istanza	$\rho_{A_1, \dots, A_n \rightarrow B_1, \dots, B_n}(r_1) = \{t \mid \exists t' \in r_1: \forall i \in [1, \dots, n]: t[B_i] = t'[A_i]\}$ Forma breve: $\rho_{A_1, \dots, A_n \rightarrow A'_1, \dots, A'_n}(r_1) = \rho_{X \rightarrow X'}(r_1)$	SELEZIONE - Estrae delle tuple		Schema	$X$	Istanza	$\sigma_F(r_1) = \{t \mid t \in r_1 \wedge F(t)\}$ con $F(t)$ formula proposizionale	Cardinalità	$0 \leq  \sigma_F(r_1)  \leq  r_1 $	PROIEZIONE - Elimina attributi		Schema	$Y = \{A_1, \dots, A_m\} \subseteq X$	Istanza	$\Pi_Y(r_1) = \{t \mid \exists t' \in r_1: t = t'[Y]\}$ con $t'[Y]$ tupla $t_0$ su $Y$ tale che $\forall A_i \in Y: t_0[A_i] = t'[A_i]$	Cardinalità	$\min( r_1 , 1) \leq  \Pi_Y(r_1)  \leq  r_1 $ dove $ \Pi_Y(r_1)  =  r_1 $ se $Y$ è superchiave per $r_1$	<div>FORMULA PROPOSIZIONALE <math>F(t)</math>:</div> <p>Si ottiene combinando con i connettivi logici <math>\wedge, \vee</math> e <math>\neg</math> condizioni atomiche del tipo: <math>A \theta B</math> o <math>A \theta c</math>, dove:</p> <ul style="list-style-type: none"><li><math>\rightarrow A, B \in X</math></li><li><math>\rightarrow \theta \in \{&lt;, =, &gt;, \neq, \geq, \leq\}</math></li><li><math>\rightarrow c</math> è una costante appartenente a <math>DOM(A)</math> o compatibile con esso</li></ul> <ul style="list-style-type: none"><li>- Una formula <math>A \theta B</math> è vera sulla tupla <math>t</math> se: <math>t[A] \theta t[B]</math> ma se <math>t[A]</math> o <math>t[B]</math> sono <i>NULL</i> allora <math>t[A] \theta t[B]</math> è falsa.</li><li>- Una formula <math>A \theta c</math> è vera sulla tupla <math>t</math> se: <math>t[A] \theta c</math> ma se <math>t[A]</math> è <i>NULL</i> allora <math>t[A] \theta c</math> è falsa.</li><li>- Le formule <math>F_1 \wedge F_2, F_1 \vee F_2, \neg F_1</math> hanno valori booleani definiti dalle note tabelle di verità dei connettivi logici corrispondenti.</li><li>- <i>A IS NULL</i> valutato sulla tupla <math>t</math> è vera se <math>t[A]</math> contiene <i>NULL</i> altrimenti è falsa, viceversa per <i>A IS NOT NULL</i></li></ul>		
	RIDENOMINAZIONE - Cambia nome agli attributi																									
Schema	$Y = \{B_1, \dots, B_n\}$ con $ Y  =  X $																									
Istanza	$\rho_{A_1, \dots, A_n \rightarrow B_1, \dots, B_n}(r_1) = \{t \mid \exists t' \in r_1: \forall i \in [1, \dots, n]: t[B_i] = t'[A_i]\}$ Forma breve: $\rho_{A_1, \dots, A_n \rightarrow A'_1, \dots, A'_n}(r_1) = \rho_{X \rightarrow X'}(r_1)$																									
SELEZIONE - Estrae delle tuple																										
Schema	$X$																									
Istanza	$\sigma_F(r_1) = \{t \mid t \in r_1 \wedge F(t)\}$ con $F(t)$ formula proposizionale																									
Cardinalità	$0 \leq  \sigma_F(r_1)  \leq  r_1 $																									
PROIEZIONE - Elimina attributi																										
Schema	$Y = \{A_1, \dots, A_m\} \subseteq X$																									
Istanza	$\Pi_Y(r_1) = \{t \mid \exists t' \in r_1: t = t'[Y]\}$ con $t'[Y]$ tupla $t_0$ su $Y$ tale che $\forall A_i \in Y: t_0[A_i] = t'[A_i]$																									
Cardinalità	$\min( r_1 , 1) \leq  \Pi_Y(r_1)  \leq  r_1 $ dove $ \Pi_Y(r_1)  =  r_1 $ se $Y$ è superchiave per $r_1$																									
<b>GIUNZIONE</b> Coppie di relazioni di schemi $X_1$ e $X_2$	<table><tr><th colspan="2">JOIN NATURALE - Uguaglianza tra gli attributi comuni alle due relazioni</th></tr><tr><td>Schema</td><td><math>X_1 \cup X_2</math></td></tr><tr><td>Istanza</td><td><math>r_1 \bowtie r_2 = \{t \mid \exists t_1 \in r_1: \exists t_2 \in r_2: t[X_1] = t_1 \wedge t[X_2] = t_2\}</math><ul style="list-style-type: none"><li>- Se almeno uno tra <math>t_1</math> e <math>t_2</math> è <i>NULL</i>, non c'è uguaglianza.</li><li>- <i>NULL</i> = <i>NULL</i> è falsa.</li><li>- Completo se <math>\forall t_1 \in r_1: \exists t \in r_1 \bowtie r_2: t[X_1] = t_1 \wedge \forall t_2 \in r_2: \exists t \in r_1 \bowtie r_2: t[X_2] = t_2</math></li></ul>Se il join non è completo, le tuple che non sono nel risultato sono le dangling tuples<ul style="list-style-type: none"><li>- Per ottenere nel risultato del join tutte le tuple (anche dangling) di una o entrambe le relazioni: left join (<math>r_1 \bowtie_{LEFT} r_2</math>), right join (<math>r_1 \bowtie_{RIGHT} r_2</math>), full join (<math>r_1 \bowtie_{FULL} r_2</math>)</li></ul></td></tr><tr><td>Cardinalità</td><td><math>0 \leq  r_1 \bowtie r_2  \leq  r_1  \cdot  r_2 </math><ul style="list-style-type: none"><li>- Se join completo: <math>\max( r_1 ,  r_2 ) \leq  r_1 \bowtie r_2  \leq  r_1  \cdot  r_2 </math></li><li>- Se <math>X_1 \cap X_2</math> superchiave per <math>r_2</math>: <math>0 \leq  r_1 \bowtie r_2  \leq  r_1 </math></li><li>- Se <math>X_1 \cap X_2</math> superchiave per <math>r_2</math> e esiste un vincolo di integrità referenziale tra <math>X_1 \cap X_2</math> (o una parte) di <math>r_1</math> e <math>r_2</math>: <math> r_1 \bowtie r_2  =  r_1 </math></li></ul></td></tr></table> <div>Proprietà del Join naturale:<ul style="list-style-type: none"><li>- Commutatività: <math>r_1 \bowtie r_2 = r_2 \bowtie r_1</math></li><li>- Associatività: <math>r_1 \bowtie (r_2 \bowtie r_3) = (r_1 \bowtie r_2) \bowtie r_3</math></li><li>- Se <math>X_1 = X_2</math>, allora: <math>r_1 \bowtie r_2 = r_1 \cap r_2</math></li><li>- Se <math>r_1</math> e <math>r_2</math> non hanno attributi comuni: <math>r_1 \bowtie r_2 = r_1 \times r_2</math></li></ul></div>	JOIN NATURALE - Uguaglianza tra gli attributi comuni alle due relazioni		Schema	$X_1 \cup X_2$	Istanza	$r_1 \bowtie r_2 = \{t \mid \exists t_1 \in r_1: \exists t_2 \in r_2: t[X_1] = t_1 \wedge t[X_2] = t_2\}$ <ul style="list-style-type: none"><li>- Se almeno uno tra <math>t_1</math> e <math>t_2</math> è <i>NULL</i>, non c'è uguaglianza.</li><li>- <i>NULL</i> = <i>NULL</i> è falsa.</li><li>- Completo se <math>\forall t_1 \in r_1: \exists t \in r_1 \bowtie r_2: t[X_1] = t_1 \wedge \forall t_2 \in r_2: \exists t \in r_1 \bowtie r_2: t[X_2] = t_2</math></li></ul> Se il join non è completo, le tuple che non sono nel risultato sono le dangling tuples <ul style="list-style-type: none"><li>- Per ottenere nel risultato del join tutte le tuple (anche dangling) di una o entrambe le relazioni: left join (<math>r_1 \bowtie_{LEFT} r_2</math>), right join (<math>r_1 \bowtie_{RIGHT} r_2</math>), full join (<math>r_1 \bowtie_{FULL} r_2</math>)</li></ul>	Cardinalità	$0 \leq  r_1 \bowtie r_2  \leq  r_1  \cdot  r_2 $ <ul style="list-style-type: none"><li>- Se join completo: <math>\max( r_1 ,  r_2 ) \leq  r_1 \bowtie r_2  \leq  r_1  \cdot  r_2 </math></li><li>- Se <math>X_1 \cap X_2</math> superchiave per <math>r_2</math>: <math>0 \leq  r_1 \bowtie r_2  \leq  r_1 </math></li><li>- Se <math>X_1 \cap X_2</math> superchiave per <math>r_2</math> e esiste un vincolo di integrità referenziale tra <math>X_1 \cap X_2</math> (o una parte) di <math>r_1</math> e <math>r_2</math>: <math> r_1 \bowtie r_2  =  r_1 </math></li></ul>	<table><tr><th colspan="2"><math>\theta</math>-JOIN</th></tr><tr><td>Schema</td><td><math>X_1 \cup X_2</math></td></tr><tr><td>Istanza</td><td><math>r_1 \bowtie_{\theta} r_2 = \sigma_{\theta}(r_1 \bowtie r_2)</math></td></tr><tr><td>Precondizione</td><td>Schemi disgiunti: <math>X_1 \cap X_2</math></td></tr></table> <div>Teorema: Date due relazioni <math>r_1</math> e <math>r_2</math> di schema <math>X_1</math> e <math>X_2</math> con <math>X_1 \cap X_2 = \{C_1, \dots, C_m\}</math> con <math>m &gt; 0</math>, vale la seguente equivalenza:<math display="block">r_1 \bowtie r_2 = \Pi_{X_1 \cup X_2}(r_1 \bowtie_{C_1 = C'_1 \wedge \dots \wedge C_m = C'_m} \rho_{C_1, \dots, C_m \rightarrow C'_1, \dots, C'_m}(r_2))</math></div> <table><tr><th colspan="2">EQUI-JOIN</th></tr><tr><td>Schema</td><td><math>X_1 \cup X_2</math></td></tr><tr><td>Istanza</td><td><math>r_1 \bowtie_{A_1 = B_1 \wedge \dots \wedge A_n = B_n} r_2</math></td></tr><tr><td>Precondizione</td><td>La relazione <math>\theta</math> è una relazione di uguaglianza</td></tr></table>	$\theta$ -JOIN		Schema	$X_1 \cup X_2$	Istanza	$r_1 \bowtie_{\theta} r_2 = \sigma_{\theta}(r_1 \bowtie r_2)$	Precondizione	Schemi disgiunti: $X_1 \cap X_2$	EQUI-JOIN		Schema	$X_1 \cup X_2$	Istanza	$r_1 \bowtie_{A_1 = B_1 \wedge \dots \wedge A_n = B_n} r_2$	Precondizione	La relazione $\theta$ è una relazione di uguaglianza
	JOIN NATURALE - Uguaglianza tra gli attributi comuni alle due relazioni																									
Schema	$X_1 \cup X_2$																									
Istanza	$r_1 \bowtie r_2 = \{t \mid \exists t_1 \in r_1: \exists t_2 \in r_2: t[X_1] = t_1 \wedge t[X_2] = t_2\}$ <ul style="list-style-type: none"><li>- Se almeno uno tra <math>t_1</math> e <math>t_2</math> è <i>NULL</i>, non c'è uguaglianza.</li><li>- <i>NULL</i> = <i>NULL</i> è falsa.</li><li>- Completo se <math>\forall t_1 \in r_1: \exists t \in r_1 \bowtie r_2: t[X_1] = t_1 \wedge \forall t_2 \in r_2: \exists t \in r_1 \bowtie r_2: t[X_2] = t_2</math></li></ul> Se il join non è completo, le tuple che non sono nel risultato sono le dangling tuples <ul style="list-style-type: none"><li>- Per ottenere nel risultato del join tutte le tuple (anche dangling) di una o entrambe le relazioni: left join (<math>r_1 \bowtie_{LEFT} r_2</math>), right join (<math>r_1 \bowtie_{RIGHT} r_2</math>), full join (<math>r_1 \bowtie_{FULL} r_2</math>)</li></ul>																									
Cardinalità	$0 \leq  r_1 \bowtie r_2  \leq  r_1  \cdot  r_2 $ <ul style="list-style-type: none"><li>- Se join completo: <math>\max( r_1 ,  r_2 ) \leq  r_1 \bowtie r_2  \leq  r_1  \cdot  r_2 </math></li><li>- Se <math>X_1 \cap X_2</math> superchiave per <math>r_2</math>: <math>0 \leq  r_1 \bowtie r_2  \leq  r_1 </math></li><li>- Se <math>X_1 \cap X_2</math> superchiave per <math>r_2</math> e esiste un vincolo di integrità referenziale tra <math>X_1 \cap X_2</math> (o una parte) di <math>r_1</math> e <math>r_2</math>: <math> r_1 \bowtie r_2  =  r_1 </math></li></ul>																									
$\theta$ -JOIN																										
Schema	$X_1 \cup X_2$																									
Istanza	$r_1 \bowtie_{\theta} r_2 = \sigma_{\theta}(r_1 \bowtie r_2)$																									
Precondizione	Schemi disgiunti: $X_1 \cap X_2$																									
EQUI-JOIN																										
Schema	$X_1 \cup X_2$																									
Istanza	$r_1 \bowtie_{A_1 = B_1 \wedge \dots \wedge A_n = B_n} r_2$																									
Precondizione	La relazione $\theta$ è una relazione di uguaglianza																									

## OTTIMIZZAZIONE DI ESPRESSIONI

### Equivalenza

- Dipendente dallo schema: Dato uno schema  $R$ ,  $E_1 \equiv_R E_2$  se  $E_1(r) = E_2(r)$  per ogni istanza  $r$  di schema  $R$
- Assoluta:  $E_1 \equiv E_2$  se  $E_1 \equiv_R E_2$  per ogni schema  $R$  compatibile con  $E_1$  e  $E_2$

**Trasformazioni di equivalenza** Data  $E$  un'espressione di schema  $X$  e  $E_1$  e  $E_2$  espressioni di schema  $X_1$  e  $X_2$

- Atomizzazione delle selezioni:  $\sigma_{F_1 \wedge F_2}(E) \equiv \sigma_{F_1}(\sigma_{F_2}(E))$  → È propedeutica ad altre trasformazioni. Non ottimizza se non è seguita da altre trasformazioni.

- Idempotenza delle proiezioni:  $\Pi_Y(E) \equiv \Pi_Y(\Pi_{YZ}(E))$  dove  $Z \subseteq X$  → È propedeutica ad altre trasformazioni. Non ottimizza se non è seguita da altre trasformazioni.

- Anticipazione delle selezioni rispetto al join:  $\sigma_F(E_1 \bowtie E_2) \equiv E_1 \bowtie \sigma_F(E_2)$  → Applicabile solo se  $F$  si riferisce solo ad attributi di  $E_2$ .

- Anticipazione della proiezione rispetto al join:  $\Pi_{X_1 Y}(E_1 \bowtie E_2) \equiv_R E_1 \bowtie \Pi_Y(E_2)$  → Applicabile solo se  $Y \subseteq X_2$  e  $(X_2 - Y) \cap X_1 = \emptyset$

- Combinazione di anticipazione della proiezione e idempotenza delle proiezioni:

$$\Pi_Y(E_1 \bowtie_F E_2) \equiv \Pi_Y(\Pi_{Y_1}(E_1) \bowtie_F \Pi_{Y_2}(E_2)) \text{ e } \Pi_Y(E_1 \bowtie E_2) \equiv \Pi_Y(\Pi_{Y_1}(E_1) \bowtie \Pi_{Y_2}(E_2))$$

dove:

$$- Y_1 = (X_1 \cap Y) \cup J_1$$

$$- Y_2 = (X_2 \cap Y) \cup J_2$$

-  $J_{1/2}$  sono gli attributi di  $E_{1/2}$  coinvolti nel join (vale a dire presenti in  $F$  per il theta-join, mentre in caso di join naturale  $J_1 = J_2 = X_1 \cap X_2$ )

- Inglobamento di una selezione in un prodotto cartesiano:  $\sigma_F(E_1 \bowtie E_2) \equiv E_1 \bowtie_F E_2$  dove  $X_1 \cap X_2 = \emptyset$

→ Applicabile solo dopo aver verificato che non sia possibile anticipare selezioni rispetto al join

- Applicazione delle proprietà commutativa e associativa di unione, prodotto cartesiano, intersezione.

- Applicazione della proprietà distributiva di selezione, proiezione e join rispetto all'unione e alla differenza:

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

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

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

$$E_1 \bowtie (E_2 \cup E_3) \equiv (E_1 \bowtie E_2) \cup (E_1 \bowtie E_3)$$

- Altre trasformazioni:

$$\sigma_{F_1 \vee F_2}(E) \equiv \sigma_{F_1}(E) \cup \sigma_{F_2}(E)$$

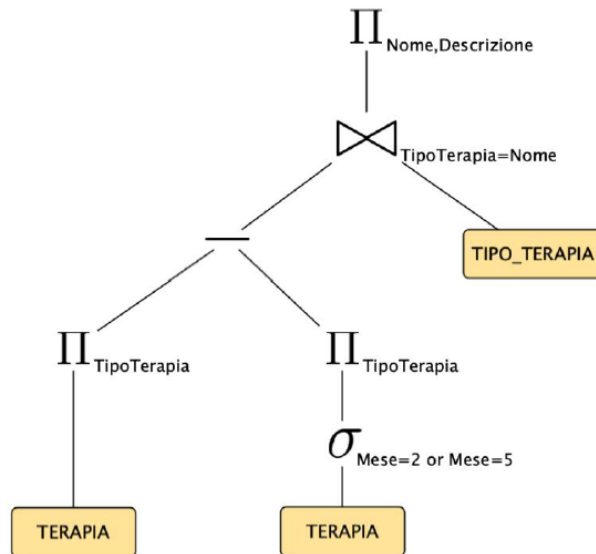
$$\sigma_{F_1 \wedge F_2}(E) \equiv \sigma_{F_1}(E) \cap \sigma_{F_2}(E) \equiv \sigma_{F_1}(E) \bowtie \sigma_{F_2}(E)$$

$$\sigma_{F_1 \wedge \neg F_2}(E) \equiv \sigma_{F_1}(E) - \sigma_{F_2}(E)$$

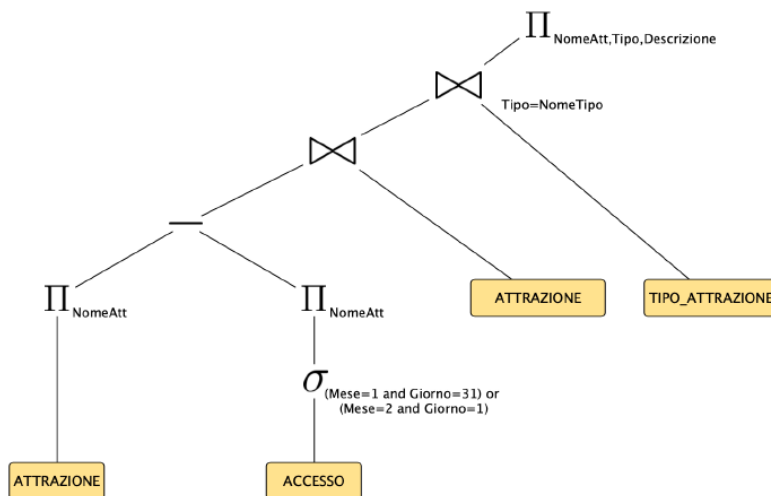
## SCHEMA ALGEBRA RELAZIONALE

IL **NON**:

**1.c (3)** Trovare i tipi di terapia che non sono state somministrate né a Febbraio né a Maggio, riportando il nome e la descrizione del tipo di terapia.

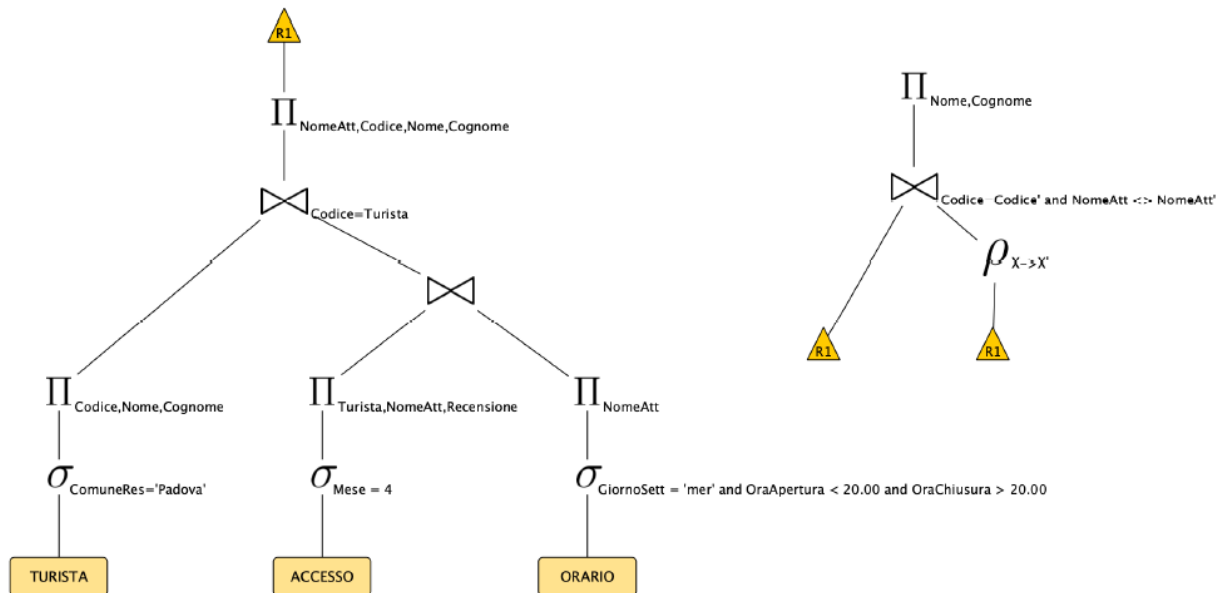


**1.c (3)** Trovare le attrazioni che non sono state visitate né il 31 gennaio né il 1 febbraio, riportando il nome dell'attrazione, il tipo e la descrizione del suo tipo.

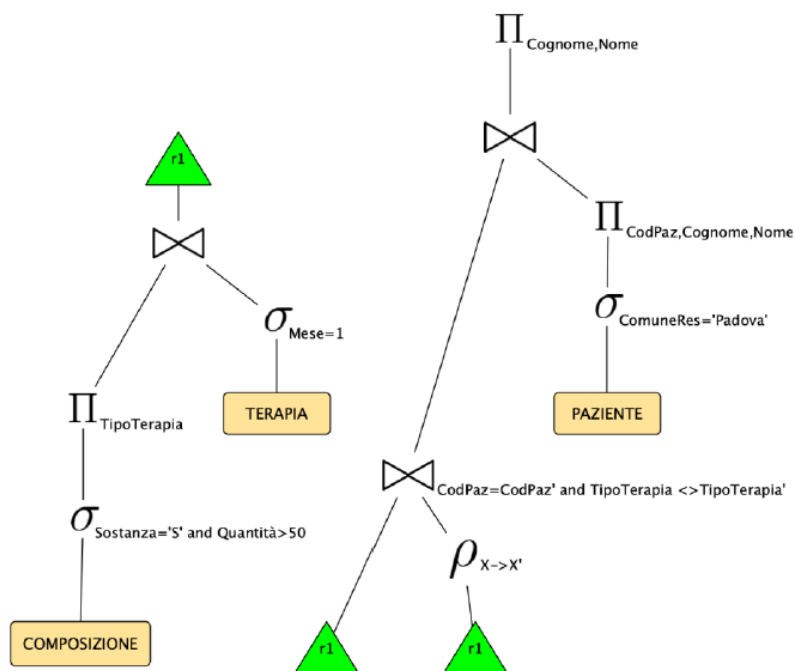


L' ALMENO DUE:

**1.b (3)** *Trovare il nome e il cognome dei turisti di Padova che hanno visitato almeno due attrazioni diverse in aprile entrambe aperte il mercoledì dopo le 20.00.*

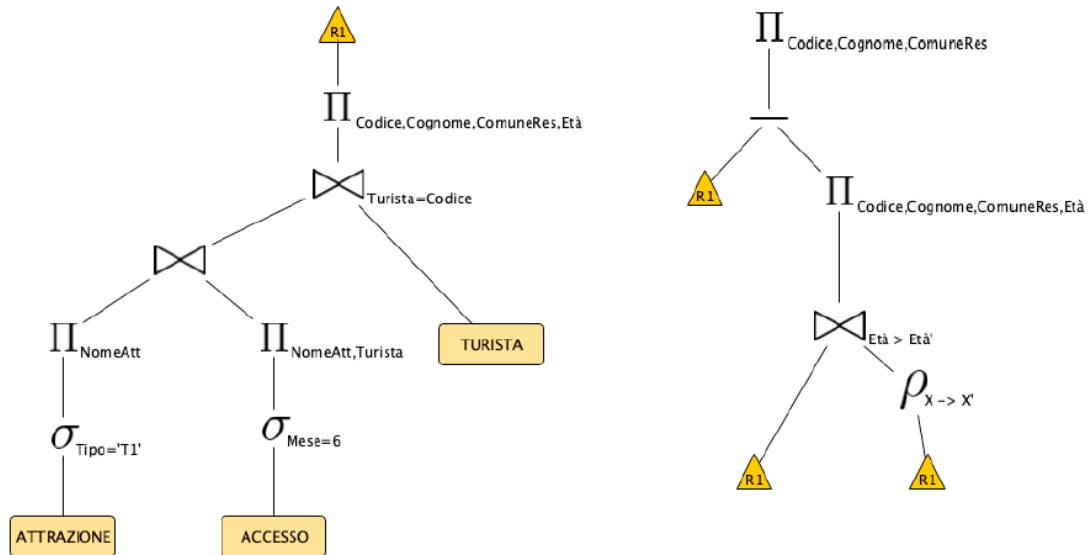


**1.b (3)** *Trovare il nome e il cognome dei pazienti di Padova che hanno eseguito almeno due terapie diverse in gennaio entrambe con una quantità di sostanza S maggiore di 50mg.*



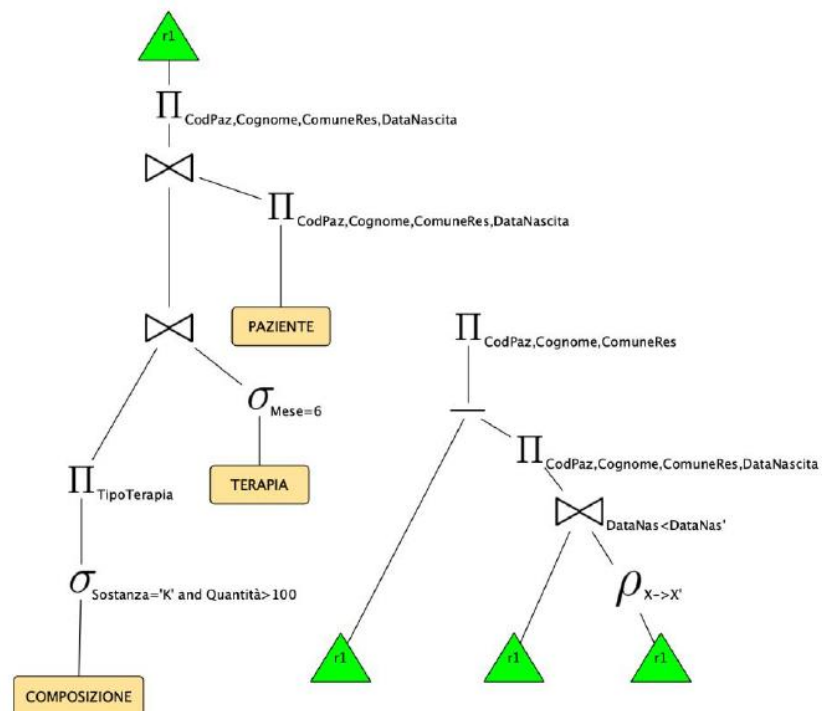
IL PIU' GIOVANE / IL PRIMO / IL MINIMO:

**2.b (3)** Trovare il codice, il cognome e il comune di residenza del turista più giovane che abbia visitato in giugno un'attrazione di tipo T1.



2. Formulare in algebra relazionale le seguenti interrogazioni:

**2.a (3)** Trovare il codice, il cognome e il comune di residenza del paziente più giovane a cui sia stata somministrata in giugno una terapia contenente più di 100mg di sostanza K.



**IL PIU' VECCHIO/ L'ULTIMO / IL MASSIMO:**

2.a (3) *Trovare la partita di durata massima che si è svolta in Francia riportando il nome del torneo, la città dove si è svolto, la data della partita e il cognome del vincitore.*

$$r = \Pi_{\{Torneo, NumeroGiornata, Città, Vincitore, Durata\}} \left( \Pi_{\{Torneo, NumeroGiornata, Vincitore, Durata\}} (PARTITA) \bowtie_{\text{Torneo} = \text{Nome}} \Pi_{\{Nome, Città\}} (\sigma_{\text{Stato} = 'Francia'} (TORNEO)) \right)$$

$$\Pi_{\{Torneo, NumeroGiornata, Città, Cognome, Data\}} \left( ((r - \Pi_{\{Torneo, NumeroGiornata, Città, Vincitore, Durata\}} (r \bowtie_{\text{durata} < \underline{\text{durata}}} \rho_{x - x} (r))) \bowtie_{\text{Vincitore} = \text{Codice}} GIORNATA) \bowtie TENNISTA) \right)$$


2.b (2) *Trovare le lezioni che si svolgono il lunedì nel 1° semestre 2016/2017 nell'aula di capienza massima riportando il nome dell'aula, la capienza, il nome dell'insegnamento, l'ora di inizio e l'ora di fine.*

$$\Pi_{\text{NomeAula, Capienza, NomeIns, OraInizio, OraFine}} (\sigma_{\text{Ora}} (LEZIONE) \bowtie (\Pi_{\text{NomeAula, Capienza}} (AULA) - \Pi_{\text{NomeAula, Capienza}} (\Pi_{\text{NomeAula, Capienza}} (AULA) \bowtie_{\text{Capienza} < \text{Capienza}'} \rho_{x \rightarrow x'} (\Pi_{\text{NomeAula, Capienza}} (AULA))))$$

con  
Q: Semestre = 1  $\wedge$  AnnoAcc = '2016/2017'  $\wedge$  Giorno = 'lun'

**IL SIA ... SIA:**

1.b (3) *Trovare il nome e il cognome dei docenti che hanno svolto lezione sia in aula A sia in aula B durante l'anno accademico 2014/2015.*

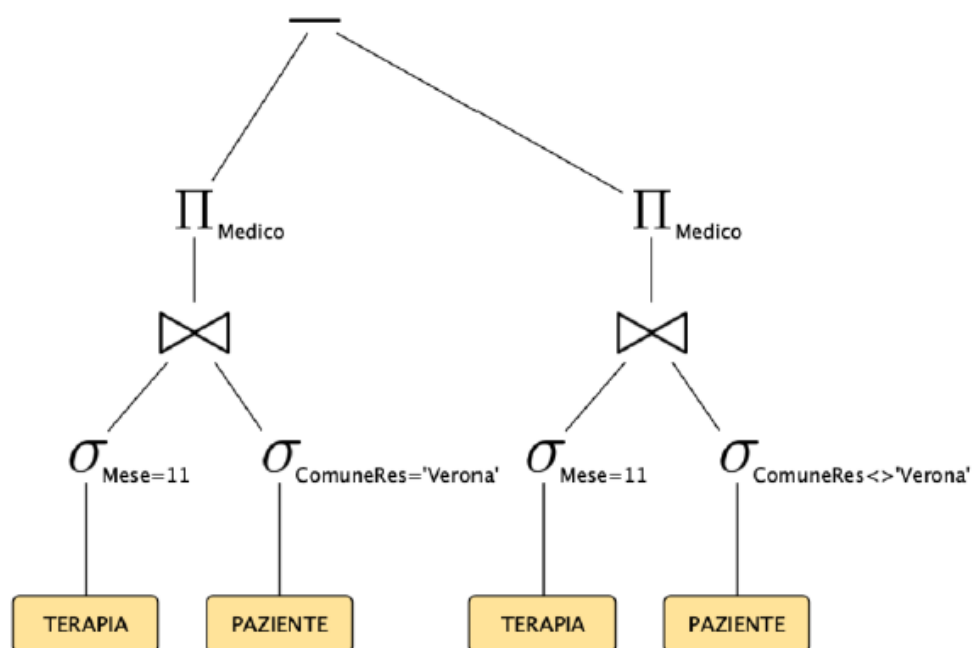
$$\Pi_{\text{Cognome, Nome}} \left( (\Pi_{\text{NomeIns, AnnoAcc}} (\sigma_{\text{Ora}} (LEZIONE)) \bowtie INSEGNAMENTO \bowtie_{\text{Docente} = \text{Matricola}} \Pi_{\text{Cognome, Nome, Matricola}} (DOCENTE)) \right)$$


$$\Pi_{\text{Cognome, Nome}} \left( (\Pi_{\text{NomeIns, AnnoAcc}} (\sigma_{\text{P}} (LEZIONE)) \bowtie INSEGNAMENTO \bowtie_{\text{Docente} = \text{Matricola}} \Pi_{\text{Cognome, Nome, Matricola}} (DOCENTE)) \right)$$

con  
Q: NomeAula = 'A'  $\wedge$  AnnoAcc = '2014/2015'  
e  
P: NomeAula = 'B'  $\wedge$  AnnoAcc = '2014/2015'

IL SOLO:

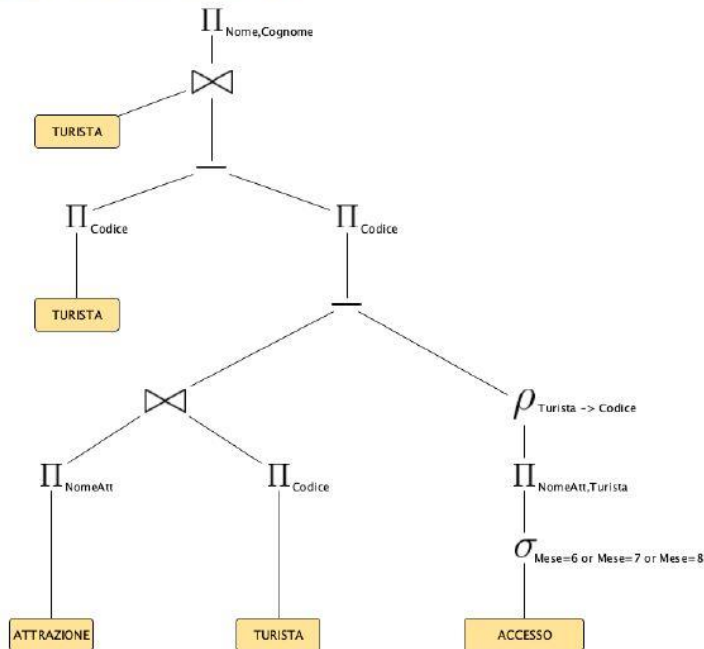
*Trovare il nome e il cognome dei medici che in Novembre hanno seguito terapie solo di clienti di Verona*



## IL TUTTI:

2. Formulare in algebra relazionale le seguenti interrogazioni:

**2.a (3)** *Trovare il nome e il cognome dei turisti che nei mesi estivi (giugno, luglio e agosto) hanno visitato tutte le attrazioni.*



**2.b (3)** *Trovare il nome e il cognome dei tennisti che hanno partecipato (sono coinvolti in almeno una partita) a tutti i tornei svoltisi in Italia.*

$r = \Pi_{\{\text{Nome}\}}(\sigma_{\text{Stato} = \text{'Italia'}}(\text{TORNEO})) \bowtie \Pi_{\{\text{Codice}\}}(\text{TENNISTA})$

$\Pi_{\{\text{Nome, Cognome}\}}(\Pi_{\{\text{Codice}\}}(\text{TENNISTA}) - \Pi_{\{\text{Codice}\}}(r - \Pi_{\{\text{Nome, Codice}\}}((\Pi_{\{\text{Codice}\}}(\text{TENNISTA}) \bowtie \text{PARTITA})$   
 $\text{Giocatore1} = \text{Codice} \vee \text{Giocatore2} = \text{Codice}$   
 $\bowtie \sigma_{\text{Stato} = \text{'Italia'}}(\text{TORNEO}))))$   
 $\bowtie \text{TENNISTA}))$

**2.a (3)** *Trovare il nome e la capienza delle aule dove nel 1° semestre 2015/2016 hanno svolto lezione tutti i docenti con ruolo='ordinario'.*

$\Pi_{\text{NomeAula, Capienza}}(\text{AULA} \bowtie$   
 $(\Pi_{\text{NomeAula}}(\text{AULA}) - \Pi_{\text{NomeAula}}($   
 $\rho_{\text{Matricola} \rightarrow \text{Docente}}(\Pi_{\text{Matricola, NomeAula}}(\text{AULA} \bowtie \sigma_P(\text{DOCENTE}))) -$   
 $\Pi_{\text{Docente, NomeAula}}(\Pi_{\text{NomeIns, AnnoAcc, NomeAula}}(\sigma_Q(\text{LEZIONE})) \bowtie \text{INSEGNAMENTO}))$   
 $))$

con  
P: Ruolo = 'ordinario'  
e  
Q: Semestre = 1  $\wedge$  AnnoAcc = '2015/2016'



CASO 7 : ... l'almeno o il trovare/ esiste (senza uso di quantificatori) :

4.b (4) *Trovare i docenti che nel 1° semestre 2016/2017 hanno svolto almeno una lezione di durata maggiore di 180 minuti, riportando nel risultato il nome e il cognome del docente insieme alla durata della lezione (durata in minuti = (OraFine – Oralnizio) ).*

```
Q = { Nome, Cognome: x(Nome,Cognome), DurataLezione: y(OraFine) – y(Oralinizio)
      | x(DOCENTE), y(LEZIONE), z(INSEGNAMENTO)
      | x.Matricola = z.Docente ∧ z.Nomelns = y.Nomelns ∧ z.AnnoAcc = y.AnnoAcc ∧
      y.Semestre = '1' ∧ y.AnnoAcc = '2016/2017' ∧ (y.OraFine – y.Oralinizio) > 180
    }
```

4.b *Trovare il nome e il cognome dei tennisti italiani che hanno giocato almeno due partite nel torneo "T3".*

```
Q = { Nome, Cognome: t(Nome,Cognome) | t(TENNISTA), y(PARTITA), z(PARTITA)
      | t.Nazionalità = 'Italia' ∧ y.Torneo = 'T3' ∧
      (y.Giocatore1 = t.Codice ∨ y.Giocatore2 = t.Codice) ∧
      z.Torneo = 'T3' ∧
      (z.Giocatore1 = t.Codice ∨ z.Giocatore2 = t.Codice) ∧
      (y.NumeroGiornata ≠ z.NumeroGiornata ∨
      y.Oralinizio ≠ z.Oralinizio) }
```

**Q<sub>5</sub>:Trovare il numero e la categoria dei treni che fermano a Vicenza e fermano a Padova dopo le 20.30**

```
Q5 = {Numero,Categoria: x.(Num,Cat)
      | x(TRENO) , y(FERMATA) , z(FERMATA) |
      x.Num=y.Treno ∧ y.Staz='Vicenza' ∧
      x.Num=z.Treno ∧ z.Staz='Padova' ∧
      z.Orario > 20.30 }
```

## CASO 1 : Né ... Non ci sono... :

4.a (4) *Trovare il nome e la capienza delle aule dove nel 2° semestre 2015/2016 il venerdì non si sono svolte lezioni.*

$Q = \{ \text{Aula, Capienza: } x.(\text{NomeAula, Capienza}) \mid x(\text{AULA}) \mid$   
 $\neg \exists y(\text{LEZIONE})(y.\text{NomeAula} = x.\text{NomeAula} \wedge y.\text{Semestre} = 2 \wedge$   
 $y.\text{AnnoAcc} = '2015/2016' \wedge y.\text{Giorno} = 'ven') \}$

3.a (2) *interrogazione 1.c (Trovare le attrazioni che non sono state visitate né il 31 gennaio né il 1 febbraio, riportando il nome dell'attrazione, il tipo e la descrizione del suo tipo)*

$Q_a = \{x.(\text{NomeAtt, Tipo}), y.(\text{Descrizione}) \mid x(\text{ATTRAZIONE}), y(\text{TIPO\_ATTRAZIONE}) \mid$   
 $x.\text{Tipo} = y.\text{NomeTipo} \wedge$   
 $\neg \exists z(\text{ACCESSO}).(z.\text{NomeAtt} = x.\text{NomeAtt} \wedge$   
 $((z.\text{Mese}=1 \wedge z.\text{Giorno}=31) \vee (z.\text{Mese}=2 \wedge z.\text{Giorno}=1)) \}$

4.a *Trovare il nome e la città dei tornei spagnoli dove nella prima e nella seconda giornata non sono previste partite dopo le 21.30*

$Q = \{ \text{Torneo, Città: } x.(\text{Nome, Città})$   
 $\mid x(\text{TORNEO})$   
 $\mid x.\text{Stato} = 'Spagna' \wedge \neg \exists y(\text{PARTITA})(y.\text{Torneo} = x.\text{Nome} \wedge$   
 $y.\text{NumeroGiornata} = '1' \wedge y.\text{Orainizio} > '21.30') \wedge$   
 $\neg \exists z(\text{PARTITA})(z.\text{Torneo} = x.\text{Nome} \wedge$   
 $z.\text{NumeroGiornata} = '2' \wedge z.\text{Orainizio} > '21.30') \}$

## CASO 2 : ...Almeno due... :

4.b Trovare il nome e il cognome dei tennisti italiani che hanno giocato almeno due partite nel torneo "T3".

$$Q = \{ \text{Nome, Cognome} : t(\text{Nome, Cognome}) \mid t(\text{TENNISTA}) \\ \mid t.\text{Nazionalità} = \text{'Italia'} \wedge \\ \exists y(\text{PARTITA})(y.\text{Torneo} = \text{'T3'} \wedge \\ (y.\text{Giocatore1} = t.\text{Codice} \vee y.\text{Giocatore2} = t.\text{Codice}) \wedge \\ \exists z(\text{PARTITA})(z.\text{Torneo} = \text{'T3'} \wedge \\ (z.\text{Giocatore1} = t.\text{Codice} \vee z.\text{Giocatore2} = t.\text{Codice}) \wedge \\ (y.\text{NumeroGiornata} \neq z.\text{NumeroGiornata} \vee \\ y.\text{Orainizio} \neq z.\text{Orainizio})) \\ \}$$

3b) Trovare la marca di auto che il 20/2/24 ha  
reportato la vendita di (almeno) due auto  
dello stesso modello, una a Milano  
(Concessionaria di MI) e una a Torino  
(Concessionaria di TO).

$$Q = \{ \text{Marca} : x.\text{Marca}, \text{Marca} : w.\text{Marca} \mid x(\text{AUTO}), w(\text{AUTO}) \\ w.\text{Modello} = x.\text{Modello} \\ \wedge$$

$$\exists y(\text{VENDITA})(y.\text{Data} = \text{'20/2/24'} \wedge y.\text{Concessionaria} = \text{'MI'} \\ \wedge y.\text{Tarpa} = x.\text{Tarpa}) \\ \wedge$$

$$\exists z(\text{VENDITA})(z.\text{Data} = \text{'20/2/24'} \wedge z.\text{Concessionaria} = \text{'TO'} \\ \wedge z.\text{Tarpa} = w.\text{Tarpa})$$

CASO 3 : in ogni ... almeno uno / trovare in ogni / tutti i ... di almeno:

4.c Trovare il nome e il cognome dei tennisti che hanno partecipato **a tutti i tornei svoltisi in USA** (partecipare ad un torneo equivale ad essere coinvolti in almeno una partita del torneo)

$$Q = \{ \text{Nome, Cognome: } t(\text{Nome, Cognome}) \mid t(\text{TENNISTA}) \mid \forall y(\text{TORNEO})(y.\text{Stato} = \text{'USA'} \Rightarrow \exists z(\text{PARTITA})(z.\text{Torneo} = y.\text{Nome} \wedge (z.\text{Giocatore1} = t.\text{Codice} \vee z.\text{Giocatore2} = t.\text{Codice})) \} \}$$

4.a (3) Trovare i tipi di terapia che nel mese di Dicembre sono state somministrate da tutti i medici, riportando nel risultato il nome e la descrizione del tipo di terapia.

$$\{ \text{nome, descrizione: } t.* \mid t(\text{TIPO\_TERAPIA}) \mid \forall x(\text{MEDICO})(\exists y(\text{TERAPIA})(y.\text{TipoTerapia} = t.\text{Nome} \wedge y.\text{Medico} = x.\text{Codice} \wedge y.\text{Mese} = 12)) \}$$

3.c (2) Trovare il nome e il cognome del turista che ha visitato (fatto almeno un accesso) tutte le attrazioni di categoria 'C1'.

$$Q_c = \{ x.(\text{Nome, Cognome}) \mid x(\text{TURISTA}) \mid \forall a(\text{ATTRAZIONE}).(\exists t(\text{TIPO\_ATTRAZIONE}).(a.\text{Tipo} = t.\text{NomeTipo} \wedge t.\text{Categoria} = \text{'C1'}) \Rightarrow \exists z(\text{ACCESSO}).(z.\text{NomeAtt} = a.\text{NomeAtt} \wedge z.\text{Turista} = x.\text{Codice})) \}$$

d. {codiceFiscale: x.Dip, cognome: y.Cognome, acronimo: x.Progetto |  
x(PARTECIPAZIONE), y(DIPENDENTE) | x.Dip = y.CF ∧  
∀t(TASK)((x.Progetto = t.Progetto) ⇒  
∃z(PARTECIPAZIONE)(z.Progetto = t.Progetto ∧ z.Numero = t.Numero ∧ z.Dip = x.Dip))}

**Q<sub>7</sub>: Trovare l'elenco di tutte le stazioni dove fermano tutti i treni con destinazione 'Venezia SL'**

$$Q_7 = \{ \text{Stazione: } x.(\text{Staz}) \mid x(\text{FERMATA}) \mid \forall y(\text{TRENO})(y.\text{Dest} = \text{'Venezia SL'} \Rightarrow \exists z(\text{FERMATA})(z.\text{Treno} = y.\text{Num} \wedge x.\text{Staz} = z.\text{Staz})) \}$$

#### CASO 4 : ... quantità massima ... /... ultimo accesso ... :

4.b (2) *Trovare il tipo di terapia che è stata somministrata nel mese di Aprile e che contiene la quantità massima della sostanza X, riportando nel risultato il tipo di terapia e la quantità.*

$\{ \text{nome: } t.\text{Nome}, \text{quantità\_X: } c.\text{Quantità} \mid t(\text{TIPO\_TERAPIA}), c(\text{COMPOSIZIONE}) \mid$   
 $t.\text{Nome} = c.\text{TipoTerapia} \wedge c.\text{Sostanza} = 'X' \wedge$   
 $\exists y(\text{TERAPIA})(y.\text{TipoTerapia} = t.\text{Nome} \wedge y.\text{Mese} = 4) \wedge$   
 $\neg \exists c1(\text{COMPOSIZIONE})(c1.\text{Sostanza} = 'X' \wedge c1.\text{Quantità} > c.\text{Quantità}) \}$

3.b (2) *Trovare l'ultimo accesso ad una attrazione di un turista di Roma il 13 dicembre, riportando nel risultato il codice del turista, l'attrazione e l'ora dell'ultimo accesso individuato.*

$Q_b = \{ x.(\text{Turista}, \text{NomeAtt}, \text{Orario}) \mid x(\text{ACCESSO}) \mid$   
 $x.\text{Mese} = 12 \wedge x.\text{Giorno} = 13 \wedge \exists t(\text{TURISTA}).(x.\text{Turista} = t.\text{Codice} \wedge t.\text{ComuneRes} = 'Roma') \wedge$   
 $\neg \exists z(\text{ACCESSO}).(z.\text{Mese} = 12 \wedge z.\text{Giorno} = 13 \wedge$   
 $\exists t1(\text{TURISTA}).(z.\text{Turista} = t1.\text{Codice} \wedge t1.\text{ComuneRes} = 'Roma') \wedge$   
 $z.\text{Orario} > x.\text{Orario}) \}$

**$Q_6$ : Trovare per ogni stazione il numero dell'ultimo treno che ferma in quella stazione**

$Q_6 = \{ \text{Stazione}, \text{Numero} : x.(\text{Staz}, \text{Treno})$   
 $\mid x(\text{FERMATA}) \mid$   
 $\neg \exists y(\text{FERMATA}) (x.\text{Staz} = y.\text{Staz} \wedge y.\text{Orario} > x.\text{Orario}) \}$



## CASO 5 : ... immediatamente successivo/a ... :

$Q_8$ : Trovare per ogni treno regionale il treno freccia rossa immediatamente successivo con la stessa destinazione, riportando nel risultato il numero e l'orario di partenza dei due treni insieme alla loro destinazione

$$Q_8 = \{ \text{Numero\_reg}, \text{Partenza\_reg} : x(\text{Num}, \text{Part}), \\ \text{Numero\_fr}, \text{Partenza\_fr}, \text{Destinazione} : y(\text{Num}, \text{Part}, \text{Dest}) \\ | x(\text{TRENO}), y(\text{TRENO}) \mid x.\text{Cat} = \text{'Reg'} \wedge y.\text{Cat} = \text{'FR'} \wedge \\ x.\text{Dest} = y.\text{Dest} \wedge x.\text{Part} < y.\text{Part} \wedge \\ \neg \exists z(\text{TRENO}) (z.\text{Cat} = \text{'FR'} \wedge x.\text{Dest} = z.\text{Dest} \wedge \\ x.\text{Part} < z.\text{Part} \wedge z.\text{Part} < y.\text{Part}) \}$$

4.c (4) *Trovare per ogni lezione del 1° semestre 2016/2017 che si svolge il martedì prima delle 17.00 in aula A, la lezione immediatamente successiva nella stessa aula, riportando nel risultato per la prima lezione il nome dell'insegnamento, l'ora di inizio e l'ora di fine e per la lezione successiva solo il nome dell'insegnamento.*

$$Q = \{ \text{Insegnamento}, \text{Inizio}, \text{Fine} : x1.(\text{NomeIns}, \text{Orainizio}, \text{OraFine}), \text{InsSuccessivo} : x2.\text{NomeIns} \\ | x1(\text{LEZIONE}), x2(\text{LEZIONE}) \\ | x1.\text{Semestre} = 1 \wedge x2.\text{Semestre} = 1 \wedge \\ x1.\text{AnnoAcc} = \text{'2016/2017'} \wedge x2.\text{AnnoAcc} = \text{'2016/2017'} \wedge \\ x1.\text{Giorno} = \text{'mar'} \wedge x2.\text{Giorno} = \text{'mar'} \wedge \\ x1.\text{NomeAula} = \text{'A'} \wedge x2.\text{NomeAula} = \text{'A'} \wedge \\ x1.\text{Orainizio} < \text{'17.00'} \wedge x1.\text{Orainizio} < x2.\text{Orainizio} \wedge \\ \neg \exists y(\text{LEZIONE}) (y.\text{Semestre} = 1 \wedge y.\text{AnnoAcc} = \text{'2016/2017'} \wedge \\ y.\text{Giorno} = \text{'mar'} \wedge y.\text{NomeAula} = \text{'A'} \wedge \\ y.\text{Orainizio} > x1.\text{Orainizio} \wedge y.\text{Orainizio} < x2.\text{Orainizio}) \}$$

## CASO 6 : ... il più giovane ... :

---

e. *Trovare il dipendente con titolo di studio 'diploma' più giovane, riportando il suo nome, cognome e data di nascita.*

e. {nome, cognome, dataNascita: x.(Nome,Cognome,DataNascita) |

x(DIPENDENTE) |

x.TitoloStudio = 'diploma'  $\wedge \neg \exists y(\text{DIPENDENTE})(y.\text{TitoloStudio} = \text{'diploma'} \wedge$

y.DataNascita > x.DataNascita) }

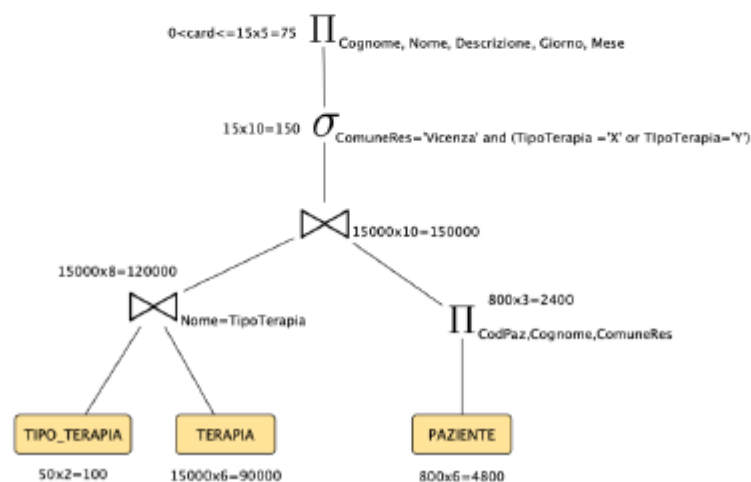
## ESEMPI ESERCIZI CARDINALITA'

3. Supponendo che le relazioni abbiano le seguenti cardinalità:

- MEDICO: 60
- TIPO\_TERAPIE: 50
- PAZIENTE: 800 (pazienti di Vicenza = 150)
- TERAPIA: 15000 (terapie di tipo X o Z = 200 e da parte di clienti vicentini = 15)

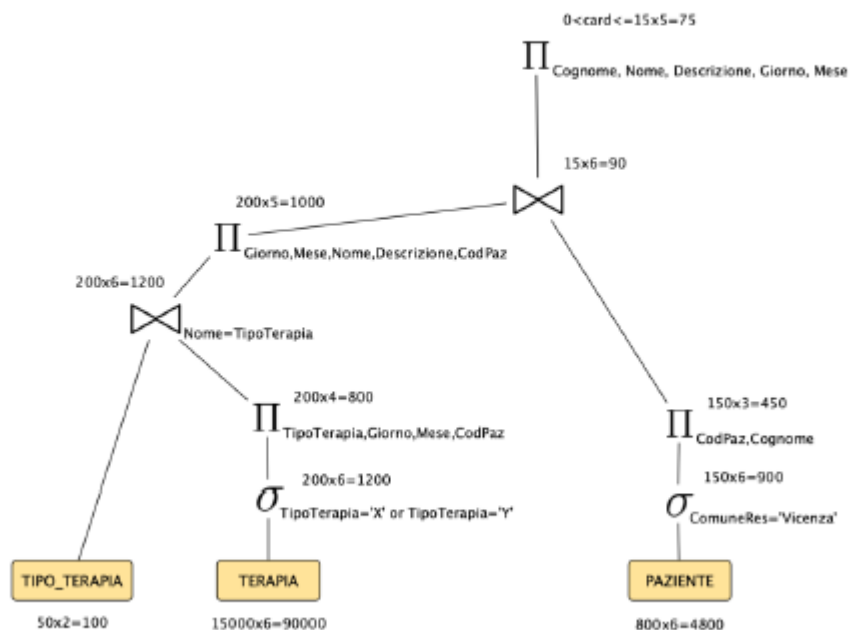
3.a (3) calcolare la dimensione dei risultati intermedi (in termini di numero di valori) in tutti i nodi dell'albero che rappresenta la seguente interrogazione:

$\Pi_{\{Cognome, Nome, Descrizione, Giorno, Mese\}}(\sigma_{ComuneRes='Vicenza' \wedge (TipoTerapia='X' \vee TipoTerapia='Z')}($   
 $(TIPO\_TERAPIA \bowtie TERAPIA) \bowtie \Pi_{CodPaz, Cognome, ComuneRes}(PAZIENTE)))$   
 $Nome = TipoTerapia$



3.b (3) produrre la versione ottimizzata della precedente interrogazione.

3.c (3) calcolare la dimensione dei risultati intermedi (in termini di numero di valori) in tutti i nodi dell'albero che rappresenta la versione ottimizzata prodotta al punto precedente.



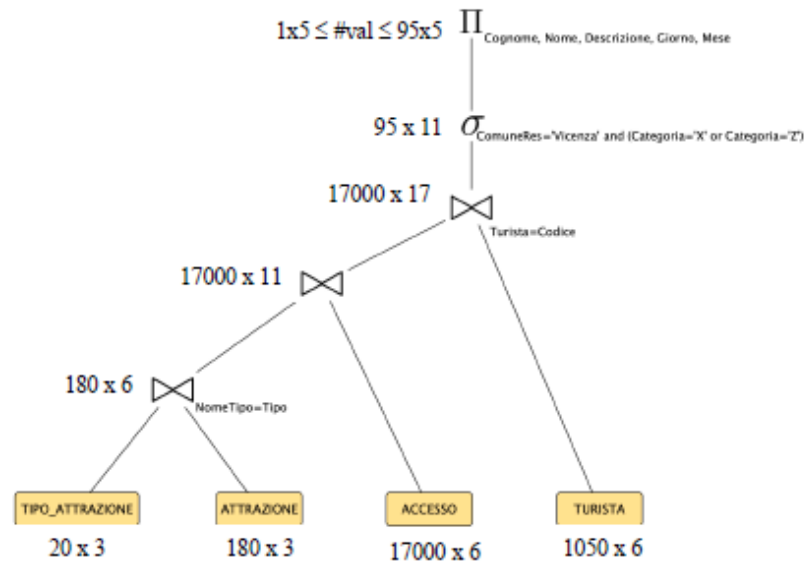


### Ottimizzazione

4. Supponendo che le relazioni abbiano le seguenti cardinalità:

- **ATTRAZIONE**: 180 (attrazioni di categoria X o Z: 37)
- **TIPO\_ATTRAZIONE**: 20 (tipi di categoria X o Z: 9)
- **TURISTA**: 1050 (turisti di Vicenza = 85)
- **ACCESSI**: 17000 (accessi con recensione: 8500 e accessi ad attrazioni di categoria X o Z: 550 e di clienti vicentini = 95)

4.a (3) calcolare la dimensione dei risultati intermedi (in termini di numero di valori) in tutti i nodi dell'albero che rappresenta la seguente interrogazione:



4.b (3) produrre la versione ottimizzata della precedente interrogazione

4.c (3) calcolare la dimensione dei risultati intermedi (in termini di numero di valori) in tutti i nodi dell'albero che rappresenta la versione ottimizzata prodotta al punto precedente.

