Università di Camerino

2. Ripasso - Algebra Relazionale e Fondamenti SQL

Piero Hierro Piermichele Rosati

Università di Camerino
Tutorato - Basi di Dati

28 aprile 2024

Indice

Algebra relazionale

È un linguaggio procedurale in cui le operazioni complesse vengono specificate descrivendo il procedimento da seguire per ottenere le soluzioni.

È uno strumento tecnico di fondamentale importanza per comprendere come si interrogano i database relazionali.

Definizioni I

Ennupla

Una n-upla (o tupla) è una collezione ordinata di n oggetti.

Relazione binaria

Una relazione r su 2 insiemi A e B è un sottoinsieme del prodotto cartesiano $A \times B$ dei due insiemi.

$$r \subseteq A \times B$$

Relazione n-aria

Una relazione su n insiemi $A_1, A_2, \ldots, A_n$ è un sottoinsieme di tutte le n-uple ordinate $a_1, a_2, \ldots, a_n$ che si possono costruire prendendo nell'ordine un elemento a_1 dal primo insieme A_1, a_2 dal scondo insiemie A_2 e così via.

Definizioni II

Dato che una relazione è un insieme:

- l'ordine delle n-uple è irrilevante;
- ogni n-upla è ordinata e distinta;
- possiamo rappresentarla sotto forma di tabella se...

Definizioni III

...vengono rispettati i requisiti fondamentali delle tabelle di un database:

- tutte le righe sono diverse tra loro;
- tutte le intestazioni delle colonne sono diverse tra loro;
- i valori di ogni colonna sono fra loro omogenei;
- l'ordine delle righe è irrilevante;
- l'ordine delle colonne è irrilevante.

Definizioni IV

- **Attributo**: nome con il quale si identifica una colonna;
- **Dominio**: insieme dei valori che possono essere assunti da un attributo (int, boolean, float, string);
- Una **ennupla** su in insieme X di attributi è una funzione che associa a ciascun attributo A in X un valore del dominio di A;
- **Schema di relazione** $R(X)$:
  - R è il nome della relazione;
  - X è un insieme di attributi $X = \{A_1, \ldots, A_n\}$
- Uno **schema di Basi di Dati** è un insieme di schemi di relazione distinti:

$$R = \{R_1(X_1), \ldots, R_k(X_k)\}$$

- L'**instanza di una relazione** su uno schema $R(X)$ è l'insieme di n-uple su X;
- L'**instanza di una base di dati** su uno schema $R = \{R_1(X_1), \ldots, R_k(X_k)\}$ è l'insieme di relazioni $r = \{r_1, \ldots, r_k\} \mid r_i$ è una relazione su R_i.

Esempio di Relazione

$$Docente = \{Re, Gagliardi, Marcantoni\}$$

$$Corso = \{Basi\ di\ Dati, Reti\}$$

$$r = \text{“}Insegna\text{”}$$

$$Insegna \subseteq Docente \times Corso$$

Esempio di Relazione

$$Docente = \{Re, Gagliardi, Marcantoni\}$$

$$Corso = \{Basi\ di\ Dati, Reti\}$$

$$r = \text{“}Insegna\text{”}$$

$$Insegna \subseteq Docente \times Corso$$

Prodotto cartesiano: $Docente \times Corso$

Docente	Corso
Re	Basi di Dati
Re	Reti
Gagliardi	Basi di Dati
Gagliardi	Reti
Marcantoni	Basi di Dati
Marcantoni	Reti

Esempio di Relazione

$$Docente = \{Re, Gagliardi, Marcantoni\}$$

$$Corso = \{Basi\ di\ Dati, Reti\}$$

$$r = \text{“}Insegna\text{”}$$

$$Insegna \subseteq Docente \times Corso$$

Prodotto cartesiano: $Docente \times Corso$

Docente	Corso
Re	Basi di Dati
Re	Reti
Gagliardi	Basi di Dati
Gagliardi	Reti
Marcantoni	Basi di Dati
Marcantoni	Reti

Insegna

Docente	Corso
Re	Basi di Dati
Gagliardi	Basi di Dati
Marcantoni	Reti

Operazioni Relazionali

- Agiscono su una o più relazioni per ottenere una nuova relazione;
- Consentono di effettuare le interrogazioni al database per ottenere le informazioni desiderate:
  - estraendo da una tabella una sottotabella;
  - combinando tra loro 2 o più tabelle;
  - generando nuove relazioni.
- Dato che le relazioni sono insiemi:
  - X insieme di attributi su cui le relazioni A e B sono definite;
  - **unione** $A \cup B$: relazione su X contente le tuple appartenenti a A o B o entrambe;
  - **intersezione** $A \cap B$: relazione su X contente le tuple appartenenti a A e B;
  - **differenza** $A - B$: relazione su X contente le tuple appartenenti a A ma non a B.

$$
\begin{aligned}
X &= \{Nome, Cognome\} \\
Docenti(X) &= \{(Barbara, Re), (Roberto, Gagliardi), (Fausto, Marcantoni), \\
&\quad (Mario, Rossi)\} \\
Studenti(X) &= \{(Mario, Rossi), (Luigi, Verdi), (Pinco, Pallino)\}
\end{aligned}
$$

$$X = \{Nome, Cognome\}$$
$$Docenti(X) = \{(Barbara, Re), (Roberto, Gagliardi), (Fausto, Marcantoni), (Mario, Rossi)\}$$
$$Studenti(X) = \{(Mario, Rossi), (Luigi, Verdi), (Pinco, Pallino)\}$$

Docenti

Nome	Cognome
Barbara	Re
Roberto	Gagliardi
Fausto	Marcantoni
Mario	Rossi

Studenti

Nome	Cognome
Mario	Rossi
Luigi	Verdi
Pinco	Pallino

Docenti $\mathbb{U}$ *Studenti* =?

$$Docenti(X) = \{(Barbara, Re), (Roberto, Gagliardi), (Fausto, Marcantoni), (Mario, Rossi)\}$$

$$Studenti(X) = \{(Mario, Rossi), (Luigi, Verdi), (Pinco, Pallino)\}$$

$$Docenti \cup Studenti = \{(Barbara, Re), (Roberto, Gagliardi), (Fausto, Marcantoni), (Mario, Rossi), (Luigi, Verdi), (Pinco, Pallino)\}$$

Docenti $\cup$ *Studenti*

Nome	Cognome
Barbara	Re
Roberto	Gagliardi
Fausto	Marcantoni
Mario	Rossi
Luigi	Verdi
Pinco	Pallino

$$
\begin{aligned}
X &= \{Nome, Cognome\} \\
Docenti(X) &= \{(Barbara, Re), (Roberto, Gagliardi), (Fausto, Marcantoni), \\
&\quad (Mario, Rossi)\} \\
Studenti(X) &= \{(Mario, Rossi), (Luigi, Verdi), (Pinco, Pallino)\}
\end{aligned}
$$

$$X = \{Nome, Cognome\}$$
$$Docenti(X) = \{(Barbara, Re), (Roberto, Gagliardi), (Fausto, Marcantoni), (Mario, Rossi)\}$$
$$Studenti(X) = \{(Mario, Rossi), (Luigi, Verdi), (Pinco, Pallino)\}$$

Docenti

Nome	Cognome
Barbara	Re
Roberto	Gagliardi
Fausto	Marcantoni
Mario	Rossi

Studenti

Nome	Cognome
Mario	Rossi
Luigi	Verdi
Pinco	Pallino

Docenti $\cap$ *Studenti* =?

$$Docenti(X) = \{(Barbara, Re), (Roberto, Gagliardi), (Fausto, Marcantoni), (Mario, Rossi)\}$$
$$Studenti(X) = \{(Mario, Rossi), (Luigi, Verdi), (Pinco, Pallino)\}$$
$$Docenti \cap Studenti = \{(Mario, Rossi)\}$$

$Docenti \cap Studenti$

Nome	Cognome
Mario	Rossi

$$X = \{Nome, Cognome\}$$
$$Docenti(X) = \{(Barbara, Re), (Roberto, Gagliardi), (Fausto, Marcantoni), (Mario, Rossi)\}$$
$$Studenti(X) = \{(Mario, Rossi), (Luigi, Verdi), (Pinco, Pallino)\}$$

$$X = \{Nome, Cognome\}$$
$$Docenti(X) = \{(Barbara, Re), (Roberto, Gagliardi), (Fausto, Marcantoni), (Mario, Rossi)\}$$
$$Studenti(X) = \{(Mario, Rossi), (Luigi, Verdi), (Pinco, Pallino)\}$$

Docenti

Nome	Cognome
Barbara	Re
Roberto	Gagliardi
Fausto	Marcantoni
Mario	Rossi

Studenti

Nome	Cognome
Mario	Rossi
Luigi	Verdi
Pinco	Pallino

$$Docenti \ - \ Studenti = ?$$

$$Docenti(X) = \{(Barbara, Re), (Roberto, Gagliardi), (Fausto, Marcantoni), (Mario, Rossi$$
$$Studenti(X) = \{(Mario, Rossi), (Luigi, Verdi), (Pinco, Pallino)\}$$
$$Docenti - Studenti = \{(Mario, Rossi)\}$$

Docenti − *Studenti*

Nome	Cognome
Barbara	Re
Roberto	Gagliardi
Fausto	Marcantoni

L'operatore ρ **Ridenominazione** viene utilizzato per rinominare uno o più attributi $A_1, \ldots, A_n$ in $A'_1, \ldots, A'_n$ di una relazione r.

$$\rho_{A'_1,\ldots,A'_n \leftarrow A_1,\ldots,A_n}(r)$$

Cani

NomeCane	RazzaCane
Alfred	Beagle
Bruce	Chihuahua
Tom	Bassotto

Gatti

NomeGatto	RazzaGatto
Virgola	Siamese
Joker	Persiano
Elvis	Siberiano

L'operatore ρ **Ridenominazione** viene utilizzato per rinominare uno o più attributi $A_1, \ldots, A_n$ in $A'_1, \ldots, A'_n$ di una relazione r.

$$\rho_{A'_1,\ldots,A'_n \leftarrow A_1,\ldots,A_n}(r)$$

Cani

NomeCane	RazzaCane
Alfred	Beagle
Bruce	Chihuahua
Tom	Bassotto

Gatti

NomeGatto	RazzaGatto
Virgola	Siamese
Joker	Persiano
Elvis	Siberiano

$$Cani \ \mathbb{U} \ Gatti = ?$$

L'operatore ρ **Ridenominazione** viene utilizzato per rinominare uno o più attributi $A_1, \ldots, A_n$ in $A'_1, \ldots, A'_n$ di una relazione r.

$$\rho_{A'_1,\ldots,A'_n \leftarrow A_1,\ldots,A_n}(r)$$

Cani

NomeCane	RazzaCane
Alfred	Beagle
Bruce	Chihuahua
Tom	Bassotto

Gatti

NomeGatto	RazzaGatto
Virgola	Siamese
Joker	Persiano
Elvis	Siberiano

$Cani \cup Gatti = ?$

Non si può fare.

È necessario usare ρ per ridenominare NomeCane e NomeGatto!

È necessario usare ρ per ridenominare RazzaCane e RazzaGatto!

$$\rho_{Nome,Razza \leftarrow NomeCane,RazzaCane}(Cani)$$

$$\rho_{Nome,Razza \leftarrow NomeGatto,RazzaGatto}(Gatti)$$

Cani

Nome	Razza
Alfred	Beagle
Bruce	Chihuahua
Tom	Bassotto

Gatti

Nome	Razza
Virgola	Siamese
Joker	Persiano
Elvis	Siberiano

$Cani \cup Gatti$

Nome	Razza
Alfred	Beagle
Bruce	Chihuahua
Tom	Bassotto
Virgola	Siamese
Joker	Persiano
Elvis	Siberiano

L'operatore σ **Selezione** viene utilizzato su una relazione r per produrre una nuova relazione in base a una condizione booleana C.
La nuova relazione conterrà tutte le n-uple che soddisfano C.

$$\sigma_C(r)$$

Studenti

Nome	Cognome	Eta
Mario	Rossi	19
Luigi	Verdi	22
Pinco	Pallino	21
Mario	Verdini	20

L'operatore σ **Selezione** viene utilizzato su una relazione r per produrre una nuova relazione in base a una condizione booleana C.
La nuova relazione conterrà tutte le n-uple che soddisfano C.

$$\sigma_C(r)$$

Studenti

Nome	Cognome	Eta
Mario	Rossi	19
Luigi	Verdi	22
Pinco	Pallino	21
Mario	Verdini	20

Studenti con più di 20 anni?

Studenti con più di 20 anni:

$$\sigma_{Eta > 20}(Studenti)$$

Studenti

Nome	Cognome	Eta
Mario	Rossi	19
Luigi	Verdi	22
Pinco	Pallino	21
Mario	Verdini	20

Studenti di nome “Mario” con più di 20 anni?

Studenti di nome “Mario” con più di 20 anni?

$$\sigma_{Nome='Mario' \ AND \ Eta > 20}(Studenti)$$

Studenti di nome “Mario” con più di 20 anni?

$$\sigma_{Nome='Mario' \ AND \ Eta > 20}(Studenti)$$

Studenti

Nome	Cognome	Eta

L'operatore π **Proiezione** viene utilizzato su una relazione r per produrre una nuova relazione in base a una lista di attributi L specificata.
La nuova relazione conterrà tutte le n-uple di r con gli attributi presenti in L.

$$\pi_L(r)$$

Studenti

Nome	Cognome	Eta
Mario	Rossi	19
Luigi	Verdi	22
Pinco	Pallino	21
Mario	Verdini	20

L'operatore π **Proiezione** viene utilizzato su una relazione r per produrre una nuova relazione in base a una lista di attributi L specificata.
La nuova relazione conterrà tutte le n-uple di r con gli attributi presenti in L.

$$\pi_L(r)$$

Studenti

Nome	Cognome	Eta
Mario	Rossi	19
Luigi	Verdi	22
Pinco	Pallino	21
Mario	Verdini	20

Nome e Cognome degli studenti?

Nome e cognome degli studenti:

$$\pi_{Nome,Cognome}(Studenti)$$

Studenti

Nome	Cognome	Eta
Mario	Rossi	19
Luigi	Verdi	22
Pinco	Pallino	21
Mario	Verdini	20

Nome ed età degli studenti con meno di 21 anni?

Nome ed età degli studenti con meno di 21 anni?

$$\pi_{Nome,Eta}(\sigma_{Eta <= 20}(Studenti))$$

Nome ed età degli studenti con meno di 21 anni?

$$\pi_{Nome,Eta}(\sigma_{Eta <= 20}(Studenti))$$

Studenti

Nome	Cognome	Eta
Mario	Rossi	19
Luigi	Verdi	22
Pinco	Pallino	21
Mario	Verdini	20

Nome ed età degli studenti con meno di 21 anni?

$$\pi Nome, Eta(\sigma_{Eta <= 20}(Studenti))$$

Studenti

Nome	Eta
Mario	19
Luigi	22
Pinco	21
Mario	20

L'operatore $\bowtie$ **Join** viene utilizzato per correlare dati in relazioni diverse basandosi sui valori degli attributi.

L'operatore $\bowtie$ **Join** viene utilizzato per correlare dati in relazioni diverse basandosi sui valori degli attributi.
Esistono diversi tipi di join:

- **Natural Join**;
- **Join completo, incompleto, vuoto**;
- **Outer Join**;
  - Left outer Join;
  - Right outer Join;
  - Full outer Join.
- **SemiJoin**;
- **Theta Join**;
- **Equi-Join**.

Il **natural join** correla i dati in 2 relazioni diverse sulla base di valori uguali negli attributi con lo stesso nome.

$$R_1 \bowtie R_2 \;=\; \{t \in \; X_1 \,\mathbb{U}\, X_2 \mid t[X_1] \in R_1 \wedge \; t[X_2] \; \in \; R_2\}$$

È una relazione definita sull'unione degli insiemi degli attributi delle due relazioni $R_1(X_1)$ e $R_2(X_2)$ ossia $X_1 \,\mathbb{U}\, X_2$.

R_1

Col1	Col2	Col3
1	a	b
2	c	d
3	e	f
4	g	h
5	g	j

R_2

Col2	Col4
a	50
g	120
m	345

Operazioni Relazionali

Natural Join

R_1

Col1	Col2	Col3
1	a	b
2	c	d
3	e	f
4	g	h
5	g	j

R_2

Col2	Col4
a	50
g	120
m	345

$R_1 \bowtie R_2$

Col1	Col2	Col3	Col4
1	a	b	50
4	g	h	50
5	g	j	120

Il Join è **completo** se ogni tupla di ciascuna relazione degli operandi contribuisce almeno a una tupla del risultato.

R_1

ID	Carattere
1	a
2	b
3	b

R_2

Carattere	Numero
a	Uno
b	Due

$R_1 \bowtie R_2$

ID	Carattere	Numero
1	a	Uno
2	b	Due
3	b	Due

Il Join è **incompleto** se almeno una tupla delle relazioni degli operandi non contribuisce al risultato.

R_1

ID	Carattere
1	a
2	b
3	b

R_2

Carattere	Numero
a	Uno
c	Due

$R_1 \bowtie R_2$

ID	Carattere	Numero
1	a	Uno

Il **join esterno** (outer join) include tutte le tuple di una relazione estese con le tuple dell'altra relazione che rispettano la condizione di join.
Gli attributi delle tuple che non rispettano la condizione di join sono riempite con i valori **NULL**.
3 tipi di join esterno:

- **sinistro**: include tutte le tuple della prima relazione

$$R_1 ⟕ R_2$$

- **destro**: include tutte le tuple della seconda relazione

$$R_1 ⟖ R_2$$

- **completo**: include tutte le tuple della prima e della seconda relazione

$$R_1 ⟗ R_2$$

Il **Left join** include tutte le tuple della prima relazione.

R_1

ID	Carattere
1	a
2	b
3	b

R_2

Carattere	Numero
a	Uno
c	Due

Il **Left join** include tutte le tuple della prima relazione.

R_1

ID	Carattere
1	a
2	b
3	b

R_2

Carattere	Numero
a	Uno
c	Due

$R_1 \ltimes R_2$

ID	Carattere	Numero
1	a	Uno
2	b	NULL
3	b	NULL

Il **Right join** include tutte le tuple della seconda relazione.

R_1

ID	Carattere
1	a
2	b
3	b

R_2

Carattere	Numero
a	Uno
c	Due

Il **Right join** include tutte le tuple della seconda relazione.

R_1

ID	Carattere
1	a
2	b
3	b

R_2

Carattere	Numero
a	Uno
c	Due

$R_1 \rtimes R_2$

ID	Carattere	Numero
1	a	Uno
NULL	c	Due

Il **Full join** include tutte le tuple della prima e della seconda relazione.

R_1

ID	Carattere
1	a
2	b
3	b

R_2

Carattere	Numero
a	Uno
c	Due

Il **Full join** include tutte le tuple della prima e della seconda relazione.

R_1

ID	Carattere
1	a
2	b
3	b

R_2

Carattere	Numero
a	Uno
c	Due

$R_1 \text{ ⟗ } R_2$

ID	Carattere	Numero
1	a	Uno
2	b	NULL
3	b	NULL
NULL	c	Due

Il Join è **vuoto** se nessuna tupla delle relazioni degli operandi contribuisce al risultato finale.

R_1

ID	Carattere
1	a
2	b
3	b

R_2

Carattere	Numero
c	Uno
d	Due

Il Join è **vuoto** se nessuna tupla delle relazioni degli operandi contribuisce al risultato finale.

R_1

ID	Carattere
1	a
2	b
3	b

R_2

Carattere	Numero
c	Uno
d	Due

$R_1 \bowtie R_2 = \emptyset$

ID	Carattere	Numero

Se nel natural join non ci sono attributi in comune, il risultato è il **prodotto cartesiano** delle relazioni $R_1 \times R_2$.

R_1

ID	Carattere
1	a
2	b
3	b

R_2

Carattere	Numero
a	Uno
c	Due

Se nel natural join non ci sono attributi in comune, il risultato è il **prodotto cartesiano** delle relazioni $R_1 \times R_2$.

R_1

ID	Carattere
1	a
2	b
3	b

R_2

Carattere	Numero
a	Uno
c	Due

$R_1 \times R_2$

ID	CarattereR_1	CarattereR_2	Numero
1	a	a	Uno
1	a	c	Due
2	b	a	Uno
2	b	c	Due
3	b	a	Uno
3	b	c	Due

Il natural join con tutti gli attributi in comune viene detto **SemiJoin** ed il risultato è l'intersezione delle relazioni $R_1 \cap R_2$ su X_1.

R_1

ID	Carattere
1	a
2	b
3	b

R_2

Carattere	Numero
a	Uno
c	Due

Il natural join con tutti gli attributi in comune viene detto **SemiJoin** ed il risultato è l'intersezione delle relazioni $R_1 \cap R_2$ su X_1.

R_1

ID	Carattere
1	a
2	b
3	b

R_2

Carattere	Numero
a	Uno
c	Due

$(R_1 \cap R_2)(X_1)$

ID	Carattere
1	a

Il **theta join** correla i dati in due relazioni diverse sulla base di una condizione booleana C (AND, OR, NOT, $>$, $<$, $<=$, $>=$).

$$R_1 \bowtie_C R_2$$

Supereroi

Nome	Film
Bruce Wayne	The Batman
Bruce Wayne	Batman Begins
Peter Parker	Spiderman
Clark Kent	Superman

Film

Titolo	AnnoUscita	Durata
The Batman	2022	176
Spiderman	2002	121
Superman	1978	151

$$Supereroi \bowtie_{Film = Titolo\ AND\ Durata < 160} Film$$

Supereroi

Nome	Film
Bruce Wayne	The Batman
Bruce Wayne	Batman Begins
Peter Parker	Spiderman
Clark Kent	Superman

Film

Titolo	AnnoUscita	Durata
The Batman	2022	176
Spiderman	2002	121
Superman	1978	151

$$Supereroi \bowtie_{Film = Titolo\ AND\ Durata < 160} Film$$

Supereroi

Nome	Film
Bruce Wayne	The Batman
Bruce Wayne	Batman Begins
Peter Parker	Spiderman
Clark Kent	Superman

Film

Titolo	AnnoUscita	Durata
The Batman	2022	176
Spiderman	2002	121
Superman	1978	151

$$Supereroi \bowtie_{Film = Titolo\ AND\ Durata < 160} Film$$

Nome	Film	Titolo	AnnoUscita	Durata
Peter Parker	Spiderman	Spiderman	2002	121
Clark Kent	Superman	Superman	1978	151

Se la condizione è composta da operatori di uguaglianza (=), eventualmente in congiunzione (AND) tra loro, il theta join è detto **equi-join**.

Supereroi

Nome	Film
Bruce Wayne	The Batman
Bruce Wayne	Batman Begins
Peter Parker	Spiderman
Clark Kent	Superman

Film

Titolo	AnnoUscita	Durata
The Batman	2022	176
Spiderman	2002	121
Superman	1978	151

$$Supereroi \bowtie_{Film\ =\ Titolo} Film$$

Supereroi

Nome	Film
Bruce Wayne	The Batman
Bruce Wayne	Batman Begins
Peter Parker	Spiderman
Clark Kent	Superman

Film

Titolo	AnnoUscita	Durata
The Batman	2022	176
Spiderman	2002	121
Superman	1978	151

$$Supereroi \bowtie_{Film\ =\ Titolo} Film$$

Supereroi

Nome	Film
Bruce Wayne	The Batman
Bruce Wayne	Batman Begins
Peter Parker	Spiderman
Clark Kent	Superman

Film

Titolo	AnnoUscita	Durata
The Batman	2022	176
Spiderman	2002	121
Superman	1978	151

$Supereroi \bowtie_{Film=Titolo} Film$

Nome	Film	Titolo	AnnoUscita	Durata
Bruce Wayne	The Batman	The Batman	2022	176
Peter Parker	Spiderman	Spiderman	2002	121
Clark Kent	Superman	Superman	1978	151

$$Supereroi \bowtie_{Film\ =\ Titolo} Film$$

$Supereroi \bowtie Film$[1]

Nome	Film	AnnoUscita	Durata
Bruce Wayne	The Batman	2022	176
Peter Parker	Spiderman	2002	121
Clark Kent	Superman	1978	151

$Supereroi \bowtie_{Film=Titolo} Film$

Nome	Film	Titolo	AnnoUscita	Durata
Bruce Wayne	The Batman	The Batman	2022	176
Peter Parker	Spiderman	Spiderman	2002	121
Clark Kent	Superman	Superman	1978	151

[1] ovviamente ridenominando prima Titolo in Film...

Indice

QUERY

Table			
Column1	Column2	Column3	Column4
Data1	Data2	Data3	Data4
Data5	Data6	Data7	Data8
Data9	Data10	Data11	Data12

Query

Temp Table		
TempColumn1	TempColumn2	TempColumn3
TempData1	TempData2	TempData3
TempData4	TempData5	TempData6
TempData7	TempData8	TempData9

SELECT

Table			
column1	column2	column3	column4
data1	data2	data3	data4
data5	data6	data7	data8
data9	data10	data11	data12

```
SELECT column1, column2
FROM TABLE
```

SELECT

Table			
column1	column2	column3	column4
data1	data2	data3	data4
data5	data6	data7	data8
data9	data10	data11	data12

```
SELECT column1, column2
FROM TABLE
```

SELECT

Table			
column1	column2	column3	column4
data1	data2	data3	data4
data5	data6	data7	data8
data9	data10	data11	data12

```
SELECT *
FROM TABLE
```

SELECT

Table			
column1	column2	column3	column4
data1	data2	data3	data4
data5	data6	data7	data8
data9	data10	data11	data12

```
SELECT *
FROM TABLE
```

SELECT

Table			
nome	column2	column3	column4
Marco	data2	data3	data4
Lucia	data6	data7	data8
Marco	data10	data11	data12

```
SELECT nome
FROM Table
```

SELECT

Table			
nome	column2	column3	column4
Marco	data2	data3	data4
Lucia	data6	data7	data8
Marco	data10	data11	data12

Query

Result
Marco
Lucia
Marco

SELECT

Result
Marco
Lucia

```
SELECT DISTINCT nome
FROM Table
```

Funzioni di aggregazione

Esami	
voti	...
18	...
30	...
25	...
27	...

```
SELECT MIN(voti)
SELECT MAX(voti)
SELECT SUM(voti)
SELECT AVG(voti)
SELECT COUNT(voti)
```

Funzioni di aggregazione

Esami	
voti	...
18	...
30	...
25	...
27	...

```
SELECT MIN(voti) 18
SELECT MAX(voti) 30
SELECT SUM(voti) 100
SELECT AVG(voti) 25
SELECT COUNT(voti) 4
```

FROM

L'istruzione SQL `FROM` viene utilizzata per specificare la tabella (o tabelle) da cui si desidera estrarre i dati.

WHERE

Table			
column1	column2	column3	column4
data1	data2	data3	data4
data5	data6	data7	data8
data9	data10	data11	data12

```
SELECT *
FROM TABLE
WHERE (condizione)
```

WHERE

Table			
column1	column2	column3	column4
data1	data2	data3	data4
data5	data6	data7	data8
data9	data10	data11	data12

```
SELECT *
FROM TABLE
WHERE (condizione)
```

WHERE

Esami	
voti	...
18	...
30	...
25	...
27	...

```
SELECT voti
FROM Esami
WHERE voti>25;
```

WHERE

Esami	
voti	...
18	...
30	...
25	...
27	...

```
SELECT voti
FROM Esami
WHERE voti>25;
```

WHERE

- **Operatori di confronto:** $=, <>, >, <, >=, <=$
- **Operazioni booleane:** AND, OR, NOT
- **BETWEEN**
- **IN**

WHERE

- **Operatori di confronto:** $=, <>, >, <, >=, <=$
- **Operazioni booleane:** AND, OR, NOT
- **BETWEEN**
- **IN**

```
WHERE Price BETWEEN 10 AND 20
WHERE Price >= 10 AND Price <= 20
```

WHERE

- **Operatori di confronto:** $=, <>, >, <, >=, <=$
- **Operazioni booleane:** AND, OR, NOT
- **BETWEEN**
- **IN**

WHERE

- **Operatori di confronto:** $=, <>, >, <, >=, <=$
- **Operazioni booleane:** AND, OR, NOT
- **BETWEEN**
- **IN**

```
WHERE Country IN ('Germany', 'France', 'UK')
WHERE Country = 'Germany' OR Country = 'France' OR Country = 'UK'
```

WHERE

- **Operatori di confronto:** $=, <>, >, <, >=, <=$
- **Operazioni booleane:** AND, OR, NOT
- **BETWEEN**
- **IN**
- **LIKE**$\{$ **%, _** $\}$

% rappresenta zero, uno o più caratteri
_ rappresenta un singolo carattere

WHERE

- **Operatori di confronto:** $=, <>, >, <, >=, <=$
- **Operazioni booleane:** AND, OR, NOT
- **BETWEEN**
- **IN**
- **LIKE**{ **%**, _ }

```
SELECT *
FROM Cliente
WHERE nome LIKE 'A%'
```

WHERE

- **Operatori di confronto:** $=, <>, >, <, >=, <=$
- **Operazioni booleane:** AND, OR, NOT
- **BETWEEN**
- **IN**
- **LIKE**{ **%**, _ }

Cliente	
nome	...
Anna	...
Fabrizio	...
Asia	...
Laura	...

```
SELECT *
FROM Cliente
WHERE nome LIKE 'A%'
```

WHERE

- **Operatori di confronto:** $=, <>, >, <, >=, <=$
- **Operazioni booleane:** AND, OR, NOT
- **BETWEEN**
- **IN**
- **LIKE**$\{$ **%, _** $\}$

Cliente	
nome	...
Anna	...
Paolo	...
Asia	...
Paola	...

```
SELECT *
FROM Cliente
WHERE nome LIKE 'A%'
```

WHERE

- **Operatori di confronto:**
  $=, <>, >, <, >=, <=$
- **Operazioni booleane:** AND, OR, NOT
- **BETWEEN**
- **IN**
- **LIKE**{ **%**, _ }

Cliente	
nome	...
Anna	...
Paolo	...
Asia	...
Paola	...

```
SELECT *
FROM Cliente
WHERE nome LIKE 'Paol_'
```

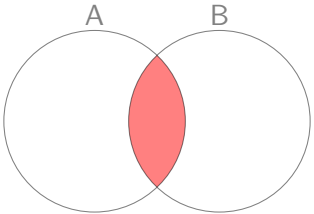
WHERE

- **Operatori di confronto:** $=, <>, >, <, >=, <=$
- **Operazioni booleane:** AND, OR, NOT
- **BETWEEN**
- **IN**
- **LIKE**{ **%**, _ }

Cliente	
nome	...
Anna	...
Paolo	...
Asia	...
Paola	...

```
SELECT *
FROM Cliente
WHERE nome LIKE 'Paol_'
```

INNER JOIN



INNER JOIN

Studente		
matricola	nome	cognome
100	Gianni	Morandi
101	Jerry	Calà
102	Pippo	Franco
103	Mara	Venier

Esame		
matricola	voto	corso
100	30	27035
100	30	27038
100	30	27010
102	21	27038
90	25	27045

```
SELECT st.matricola, cognome, voto, corso
FROM Studente AS st INNER JOIN Esame AS es ON st.matricola=es.matricola;
```

INNER JOIN

Result			
matricola	cognome	voto	corso
100	Morandi	30	27035
100	Morandi	30	27038
100	Morandi	30	27010
102	Franco	21	27038

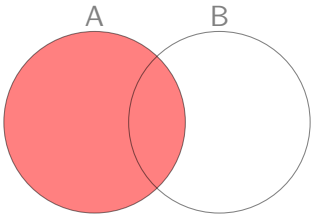
INNER JOIN

Studente		
matricola	nome	cognome
100	Gianni	Morandi
101	Jerry	Calà
102	Pippo	Franco
103	Mara	Venier

Esame		
matricola	voto	corso
100	30	27035
100	30	27038
100	30	27010
102	21	27038
90	25	27045

```
SELECT Studente.matricola, cognome, voto, corso
FROM Studente, Esame
WHERE Studente.matricola=Esame.matricola;
```

LEFT JOIN



LEFT JOIN

Studente		
matricola	nome	cognome
100	Gianni	Morandi
101	Jerry	Calà
102	Pippo	Franco
103	Mara	Venier

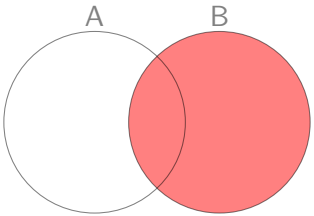
Esame		
matricola	voto	corso
100	30	27035
100	30	27038
100	30	27010
102	21	27038
90	25	27045

```
SELECT st.matricola, cognome, voto, corso
FROM Studente AS st LEFT JOIN Esame AS es ON st.matricola=es.matricola;
```

LEFT JOIN

Result			
matricola	cognome	voto	corso
100	Morandi	30	27035
100	Morandi	30	27038
100	Morandi	30	27010
101	Calà	NULL	NULL
102	Franco	21	27038
103	Venier	NULL	NULL

RIGHT JOIN



RIGHT JOIN

Studente		
matricola	nome	cognome
100	Gianni	Morandi
101	Jerry	Calà
102	Pippo	Franco
103	Mara	Venier

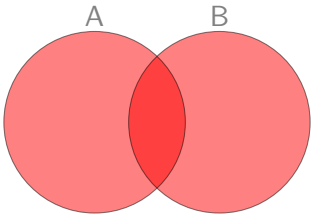
Esame		
matricola	voto	corso
100	30	27035
100	30	27038
100	30	27010
102	21	27038
90	25	27045

```
SELECT st.matricola, cognome, voto, corso
FROM Studente AS st RIGHT JOIN Esame AS es ON st.matricola=es.matricola;
```

RIGHT JOIN

Result			
matricola	cognome	voto	corso
100	Morandi	30	27035
100	Morandi	30	27038
100	Morandi	30	27010
102	Franco	21	27038
90	NULL	25	27045

FULL OUTER JOIN



FULL OUTER JOIN

Studente		
matricola	nome	cognome
100	Gianni	Morandi
101	Jerry	Calà
102	Pippo	Franco
103	Mara	Venier

Esame		
matricola	voto	corso
100	30	27035
100	30	27038
100	30	27010
102	21	27038
90	25	27045

```
SELECT st.matricola, cognome, voto, corso
FROM Studente AS st FULL OUTER JOIN Esame AS es ON
st.matricola=es.matricola;
```

FULL OUTER JOIN

Result			
matricola	cognome	voto	corso
100	Morandi	30	27035
100	Morandi	30	27038
100	Morandi	30	27010
101	Calà	NULL	NULL
102	Franco	21	27038
103	Venier	NULL	NULL
90	NULL	25	27045

Indice

Domande?

Grazie dell'attenzione!