Abilità Informatiche

Luigi Catuogno

[luigi.catuogno@uniparthenope.it]

Corso di Laurea in Economia e Commercio - Anno Accademico 2022-23

1

Libro di testo

[IdB]

Dennis P. Curtin, Kim Foley, Kunal Sen, Cathleen Morin

Informatica di base

VII edizione (2016), MacGraw Hill Education
ISBN: 978-88-386-1537-5

2

Altro materiale di utile consultazione

[Sli] Slides, appunti e altro materiale distribuito dal docente

[Misc] Altra fonte diversamente specificata di volta in volta

3

Basi di dati

Primo approccio con SQL

4

Primo approccio con SQL

5

Materiali

I database SQLite utilizzati nei seguenti esempi sono forniti insieme al materiale didattico del corso, nei seguenti file:

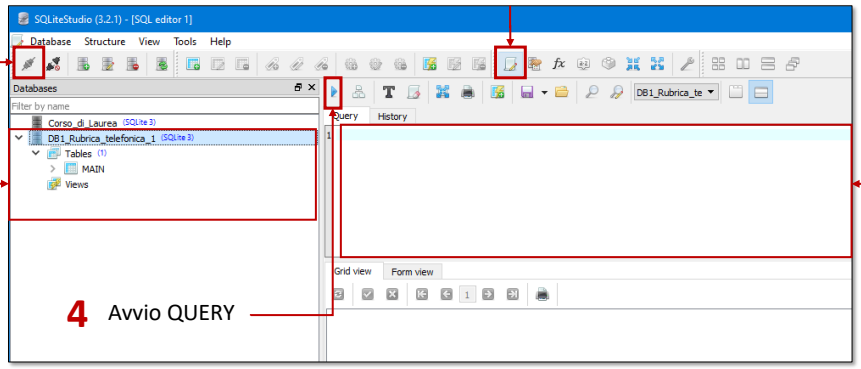
DB1_Rubrica_telefonica_1.db
DB2_Dipendenti_1.db
DB3_Creditori_1.db
DB4_Statistiche_Calciatori_1.db

6

Database: rubrica_telefonica_1

1 Aprire e connettere il database `DB1_Rubrica_telefonica_1`

2 Open SQL Editor



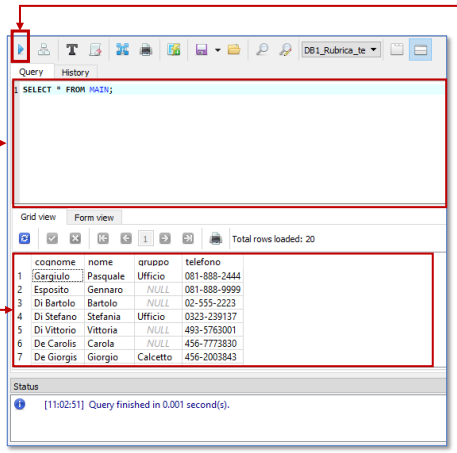
3 Comporre la query nell'editor

4 Avvio QUERY

Database: rubrica_telefonica_1

3 Comporre la Query nell'editor

4 Avvio QUERY



5 Risultati

Database: rubrica_telefonica_1

Database: **`rubrica_telefonica_1`**

Table name: MAIN ☐ WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Default value
1	cognome	STRING (20)							*NULL*
2	nome	STRING (20)							*NULL*
3	gruppo	STRING (20)							*NULL*
4	telefono	STRING (20)							*NULL*

9

Database: rubrica_telefonica_1

Database: **`rubrica_telefonica_1`** **Table**: `MAIN`

	cognome	nome	gruppo	telefono
1	Gargiulo	Pasquale	Ufficio	081-888-2444
2	Esposito	Gennaro	*NULL*	081-888-9999
3	Di Bartolo	Bartolo	*NULL*	02-555-2223
4	Di Stefano	Stefania	Ufficio	0323-239137
5	Di Vittorio	Vittoria	*NULL*	493-5763001
6	De Carolis	Carola	*NULL*	456-7773830
7	De Giorgis	Giorgio	Calcetto	456-2003843
8	Marea	Mara	Famiglia	456-7366944
9	Giannini	Giunone	Famiglia	456-8704813
10	Giannini	Gino	Famiglia	456-4026990
11	Giannini	Gianna	Famiglia	456-2321123
12	D'Alambert	Alamberto	Ufficio	02-555-23201
13	Ivanovich	Ivan	Calcetto	012-222-4837
14	Bottini	Enrico	Scuola	012-222-2232
15	Viola	Dino	Calcetto	02-555-53100
16	Rosi	Marco	Scuola	02-555-37523
17	Neri	Franco	Ufficio	02-555-49567
18	Verdi	Giuseppe	Calcetto	02-555-65047
19	Rossi	Mario	Calcetto	02-555-21582
20	Bianchi	Carlo	Ufficio	02-555-93174

10

SELECT – FROM

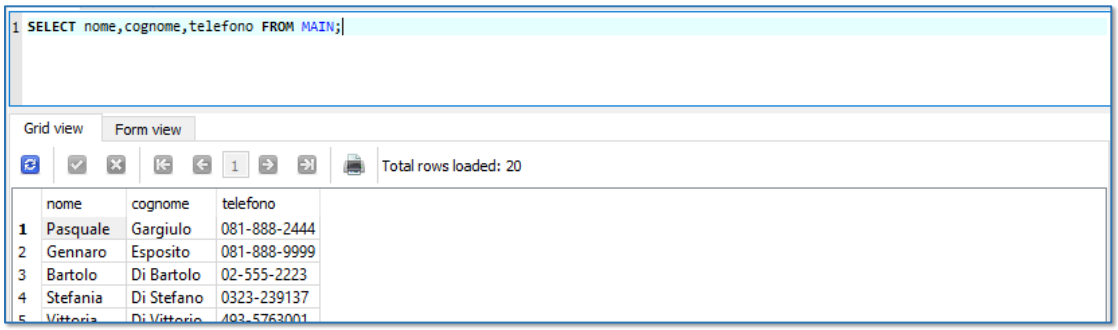
SELECT *[cols]* **FROM** *[table];*

Restituisce un elenco in cui ciascuna riga è composta dal contenuto delle colonne *[cols]* nella corrispondente riga della tabella *[table]*

11

SELECT – FROM

SELECT *nome,cognome,telefono* **FROM** *MAIN*



12

SELECT – FROM

```
SELECT nome,cognome,telefono FROM MAIN;
```

cognome	nome	gruppo	telefono
Gargiulo	Pasquale	Ufficio	081-888..
Esposito	Gennaro		081-888..
…			

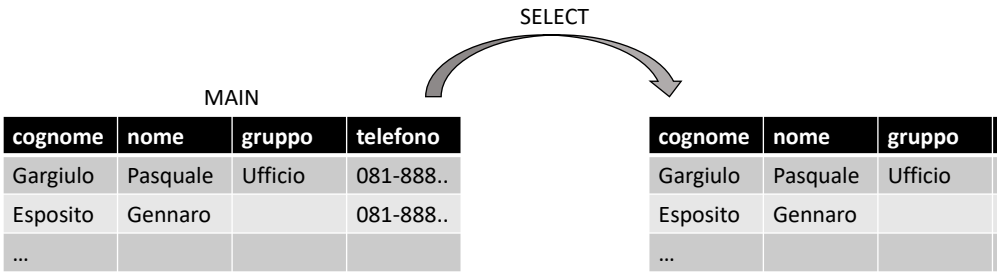
nome	cognome	telefono
Pasquale	Gargiulo	081-888..
Gennaro	Esposito	081-888..
…		

13

SELECT – FROM

Con '*' si intendono *tutte le colonne*.

```
SELECT * FROM MAIN;
```



cognome	nome	gruppo	telefono
Gargiulo	Pasquale	Ufficio	081-888..
Esposito	Gennaro		081-888..
…			

cognome	nome	gruppo	telefono
Gargiulo	Pasquale	Ufficio	081-888..
Esposito	Gennaro		081-888..
…			

14

Esercizio

1 Estrarre dalla rubrica una tabella, per ogni riga i campi nome, cognome e gruppo

2 Estrarre dalla rubrica una tabella che riporti, per ogni riga: cognome e telefono

15

Esercizio

1 Estrarre dalla rubrica una tabella che riporti, per ogni riga i campi nome, cognome e gruppo

```
SELECT nome,cognome,gruppo FROM MAIN
```

2 Estrarre dalla rubrica una tabella che riporti, per ogni riga: cognome e telefono

```
SELECT cognome,telefono FROM MAIN
```

16

SELECT – FROM – WHERE

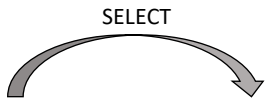
SELECT *[cols]* **FROM** *[table]* **WHERE** c1=*'v1'* …**;**

Restituisce una tabella in cui ciascuna riga è composta dal contenuto delle colonne *[cols]* in ogni riga della tabella *[table]* in cui la colonna *c1* contiene il valore *v1*.

17

SELECT – FROM

SELECT telefono **FROM** MAIN **WHERE** nome=*'Franco';*



MAIN

cognome	nome	gruppo	telefono
Gargiulo	Pasquale	Ufficio	081-888..
Esposito	Gennaro		081-888..
…			

telefono
02-555-49567

18

Esercizio

3 Estrarre dalla rubrica una tabella che riporti, nome e cognome dei componenti del gruppo 'Calcetto'

4 Estrarre dalla rubrica una tabella che riporti il nome e il numero di telefono di tutti i componenti del gruppo 'Famiglia'

19

Esercizio

3 Estrarre dalla rubrica una tabella che riporti, nome e cognome dei componenti del gruppo 'Calcetto'

```
SELECT nome,cognome FROM MAIN WHERE gruppo='Calcetto'
```

4 Estrarre dalla rubrica una tabella che riporti il nome e il numero di telefono di tutti i componenti del gruppo 'Famiglia'

```
SELECT nome,numero FROM MAIN WHERE gruppo='Famiglia'
```

20

Esercizio

5 Estrarre dalla rubrica tutte le righe in cui la colonna gruppo ha valore Ufficio

6 Estrarre dalla rubrica una tabella con cognome, nome e telefono di tutte le righe della table MAIN in cui la colonna Gruppo non contiene alcun valore.

21

Esercizio

5 Estrarre dalla rubrica tutte le righe in cui la colonna gruppo ha valore Ufficio

```
SELECT * FROM MAIN WHERE gruppo='Ufficio'
```

6 Estrarre dalla rubrica una tabella con cognome, nome e telefono di tutte le righe della table MAIN in cui la colonna Gruppo non contiene alcun valore.

```
SELECT cognome,nome,telefono FROM MAIN WHERE gruppo is NULL
```

Il modo corretto per indicare un campo che non contiene alcun dato.

22

SELECT – FROM – WHERE + AND/OR/NOT

SELECT *[cols]* **FROM** *[table]* **WHERE** c1='*v1*' **AND/OR** c2='v2'…

Restituisce una tabella in cui ciascuna riga è composta dal contenuto delle colonne *[cols]* in ogni riga della tabella *[table]* in cui la colonna *c1* contiene il valore *v1* **E/O** la colonna *c2* contiene il valore *v2*…

SELECT *[cols]* **FROM** *[table]* **WHERE** c1='*v1*' **AND/OR**
NOT c2='v2'…

Restituisce una tabella in cui ciascuna riga è composta dal contenuto delle colonne *[cols]* in ogni riga della tabella *[table]* in cui la colonna *c1* contiene il valore *v1* **E/O** la colonna *c2* **NON** contiene il valore *v2*…

23

Esercizio

7 Estrarre dalla rubrica tutte le righe in cui la colonna gruppo ha valore Famiglia oppure ha valore Scuola

8 Estrarre dalla rubrica una tabella con nome, cognome e telefono di tutte le righe della table MAIN in cui la colonna Gruppo non contiene il valore Ufficio.

24

Esercizio

7 Estrarre dalla rubrica tutte le righe in cui la colonna gruppo ha valore Famiglia oppure ha valore Scuola

```
SELECT * FROM MAIN WHERE gruppo='Famiglia' OR gruppo='Scuola'
```

8 Estrarre dalla rubrica una tabella con nome, cognome e telefono di tutte le righe della table MAIN in cui la colonna Gruppo non contiene il valore Ufficio.

```
SELECT nome, telefono FROM MAIN WHERE NOT gruppo='Ufficio'
```

25

SELECT – FROM – ORDER BY … [ASC/DESC]

```
SELECT [cols] FROM [table] ORDER BY c1 [ASC/DESC] …
```

```
SELECT [cols] FROM [table] WHERE c1='v1' ORDER BY c2 [ASC/DESC] …
```

Restituisce una tabella in cui ciascuna riga è composta dal contenuto delle colonne *[cols]* in ogni riga della tabella *[table] (che rispetti l'eventuale clausola WHERE).* Le righe della tabella sono ordinate rispetto al valore della colonna *c1*. **ASC** o **DESC** prescrivono rispettivamente il senso **crescente** o **decrescente** dell'ordinamento.

26

Esercizio

9 Visualizzare il contenuto della rubrica ordinando le righe per cognome

10 Visualizzare in ordine alfabetico rispetto al cognome i record della rubrica il cui campo gruppo è 'Calcetto'.

27

OK! Esercizio

9 Visualizzare il contenuto della rubrica ordinando le righe per cognome

```
SELECT * FROM MAIN ORDER BY cognome
```

10 Visualizzare in ordine alfabetico rispetto al cognome i record della rubrica il cui campo gruppo è 'Calcetto'.

```
SELECT * FROM MAIN WHERE gruppo='Calcetto' ORDER BY cognome
```

28

SELECT – FROM – ORDER BY … [ASC/DESC]

SELECT *[cols]* **FROM** *[table]* **ORDER BY** c1 **[ASC/DESC]** …

SELECT *[cols]* **FROM** *[table]* **WHERE** c1='*v1*' **ORDER BY** c2 **[ASC/DESC]** …

Restituisce una tabella in cui ciascuna riga è composta dal contenuto delle colonne *[cols]* in ogni riga della tabella *[table] (che rispetti l'eventuale clausola WHERE)*. Le righe della tabella sono ordinate rispetto al valore della colonna *c1*. **ASC** o **DESC** prescrivono rispettivamente il senso **crescente** o **decrescente** dell'ordinamento.

SELECT *[cols]* **FROM** *[table]* **ORDER BY** c1 **[ASC/DESC],** c2 **[ASC/DESC]** …

E' possibile ordinare le righe selezionate per chiavi multiple. Qui, c1 è il primo criterio di ordinamente selezionato. A parità di valore per c1, le righe sono ordinate secondo il criterio indicato per c2, etc.

29

Esercizio

11

Visualizzare il contenuto della rubrica ordinando le righe per cognome e nome.

30

Esercizio

11

Visualizzare il contenuto della rubrica ordinando le righe per cognome e nome.

```
SELECT * FROM MAIN ORDER BY cognome, nome
```

31

Database: Dipendenti_1

Database: **Dipendenti_1**

Table name: libro_paga WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Default value
1	matricola	STRING (5)							NULL
2	cognome	STRING (30)							NULL
3	data_assunzione	DATE							NULL
4	qualifica	STRING (2)							NULL
5	livello	INTEGER							NULL
6	ultima_retribuzione	INTEGER							NULL

32

Database: Dipendenti_1

Database: **`Dipendenti_1`** **Table**: `libro_paga`

	matricola	cognome	data_assun:	qualifica	livello	ultima_retrit
81	P0216	MONTA...	15/01/1975	OP	2	1644000
82	P0407	SQUILL...	04/06/1974	OP	1	1244000
83	P0234	ALESSIO	02/07/1973	IMP	1	1618000
84	P0217	MAROV...	07/09/1972	IMP	1	1891000
85	P0256	ANFOSSI	07/01/1972	OP	3	1771000
86	P0474	RASERO	25/03/1971	OP	5	2164000
87	P0468	RIZZITIE...	18/08/1970	IMP	2	2136000
88	P0328	FASANO	15/08/1970	OP	2	1315000
89	P0322	BUSON...	10/06/1970	OP	2	1349000
90	P0467	CORAG...	21/12/1969	OP	3	1560000
91	P0287	CASETTA	26/08/1969	OP	3	1759000
92	P0316	PERNET...	24/07/1969	OP	5	2227000
93	P0252	CARDIA	08/02/1969	OP	5	2771000
94	P0279	BUZZELLI	30/01/1969	OP	2	1409000
95	P0402	SQUIN...	06/08/1967	OP	2	1257000
96	P0215	PRENNA	05/06/1965	OP	1	1574000
97	P0334	ULLASCI	01/06/1964	IMP	2	2409000
98	P0340	CRIDA	21/05/1964	OP	1	1264000
99	P0251	BOSSO	25/12/1963	OP	1	1493000
100	P0326	NEBIOLO	15/09/1963	OP	2	1335000
101	P0224	VELARDI	07/08/1963	IMP	3	2851000
102	P0218	MONTA...	05/08/1963	OP	2	1594000
103	P0330	AMERIO	22/12/1962	OP	3	1628000

33

SELECT – FROM – WHERE – BETWEEN..AND

`SELECT` *`[cols]`* **`FROM`** *`[table]`* **`WHERE`** `c1` **`BETWEEN`** *`v1`* **`AND`** *`v2`*…

`SELECT` *`[cols]`* **`FROM`** *`[table]`* **`WHERE`** `c1` **`BETWEEN`** *`v1`* **`AND`** *`v2`* **`ORDER BY`**…

Restituisce una tabella in cui ciascuna riga è composta dal contenuto delle colonne *[cols]* di ogni riga della tabella *[table]* in cui il valore della colonna c1 è compreso tra *v1* e *v2*.

34

Esercizio

12 Visualizzare l'elenco dei dipendenti il cui stipendio è compreso tra 1500000 e 1600000. Il risultato sia ordinato rispetto all'importo e al cognome

13 Visualizzare l'elenco dei dipendenti la cui qualifica è OP, e il cui livello è compreso tra la 1 e la 2. Il risultato sia ordinato rispetto alla qualifica (decrescente) e il cognome (crescente).

35

OK! Esercizio

12 Visualizzare l'elenco dei dipendenti il cui stipendio è compreso tra 1500000 e 1600000. Il risultato sia ordinato rispetto all'importo e al cognome

```
SELECT * FROM libro_paga WHERE ultima_retribuzione
  BETWEEN 1500000 AND 1600000
  ORDER BY ultima_retribuzione, cognome
```

13 Visualizzare l'elenco dei dipendenti la cui qualifica è OP, e il cui livello è compreso tra la 1 e la 2. Il risultato sia ordinato rispetto alla qualifica (decrescente) e il cognome (crescente).

```
SELECT * FROM libro_paga WHERE qualifica = 'OP' AND
  livello BETWEEN 1 AND 2
  ORDER BY livello DESC, cognome ASC
```

36

SELECT – FROM – WHERE *op. relazionali.*

SELECT *[cols]* **FROM** *[table]* **WHERE** c1 **[op]** *v1* **AND/OR [NOT]** c2 **[op]** *v2* …

SELECT *[cols]* **FROM** *[table]* **WHERE** c1 **[op]** *v1* **AND/OR [NOT]** c2 **[op]** *v2* …

[op] può essere uno tra: **<, <=, >, >=, ==** (anche **'='**), **!=**

37

Esercizio

14 Visualizzare l'elenco dei dipendenti il cui livello è superiore al terzo, il cui stipendio è superiore a 1800000 e la cui qualifica è diversa da IMP. Il risultato sia ordinato rispetto al cognome

38

Esercizio

14 Visualizzare l'elenco dei dipendenti il cui livello è superiore al terzo, il cui stipendio è superiore a 1800000 e la cui qualifica è diversa da IMP. Il risultato sia ordinato rispetto al cognome

```
SELECT * FROM libro_paga WHERE livello > 3 AND
ultima_retribuzione > 1800000 AND qualifica != 'IMP'
ORDER BY cognome
```

39

Esercizio

15 Visualizzare l'elenco dei dipendenti di terzo livello, ordinati per ultima retribuzione (decrescente) e per cognome in ordine alfabetico.

16 Estrarre dal libro_paga una tabella contenente le colonne matricola, cognome e ultima_retribuzione di tutti i dipendenti di qualifica OP e di livello 1. Il risultato deve essere ordinato per cognome.

40

Database: Dipendenti_1

Database: **`Dipendenti_1`**

Table name: libro_paga ☐ WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Default value
1	matricola	STRING (5)							*NULL*
2	cognome	STRING (30)							*NULL*
3	data_assunzione	DATE							*NULL*
4	qualifica	STRING (2)							*NULL*
5	livello	INTEGER							*NULL*
6	ultima_retribuzione	INTEGER							*NULL*

41

Database: Dipendenti_1

Database: **`Dipendenti_1`** **Table**: `libro_paga`

	matricola	cognome	data_assun:	qualifica	livello	ultima_retrit
81	P0216	MONTA...	15/01/1975	OP	2	1644000
82	P0407	SQUILL...	04/06/1974	OP	1	1244000
83	P0234	ALESSIO	02/07/1973	IMP	1	1618000
84	P0217	MAROV...	07/09/1972	IMP	1	1891000
85	P0256	ANFOSSI	07/01/1972	OP	3	1771000
86	P0474	RASERO	25/03/1971	OP	5	2164000
87	P0468	RIZZITIE...	18/08/1970	IMP	2	2136000
88	P0328	FASANO	15/08/1970	OP	2	1315000
89	P0322	BUSON...	10/06/1970	OP	2	1349000
90	P0467	CORAG...	21/12/1969	OP	3	1560000
91	P0287	CASETTA	26/08/1969	OP	3	1759000
92	P0316	PERNET...	24/07/1969	OP	5	2227000
93	P0252	CARDIA	08/02/1969	OP	5	2771000
94	P0279	BUZZELLI	30/01/1969	OP	2	1409000
95	P0402	SQUIN...	06/08/1967	OP	2	1257000
96	P0215	PRENNA	05/06/1965	OP	1	1574000
97	P0334	ULLASCI	01/06/1964	IMP	2	2409000
98	P0340	CRIDA	21/05/1964	OP	1	1264000
99	P0251	BOSSO	25/12/1963	OP	1	1493000
100	P0326	NEBIOLO	15/09/1963	OP	2	1335000
101	P0224	VELARDI	07/08/1963	IMP	3	2851000
102	P0218	MONTA...	05/08/1963	OP	2	1594000
103	P0330	AMERIO	22/12/1962	OP	3	1628000

42

Esercizio

15 Visualizzare l'elenco dei dipendenti di terzo livello, ordinati per ultima retribuzione (decrescente) e per cognome in ordine alfabetico.

43

Esercizio

15 Visualizzare l'elenco dei dipendenti di terzo livello, ordinati per ultima retribuzione (decrescente) e per cognome in ordine alfabetico.

```
SELECT * FROM libro_paga WHERE livello = '3'
      ORDER BY ultima_retribuzione DESC, cognome ASC;
```

44

Esercizio

16 Estrarre dal libro_paga una tabella contenente le colonne matricola, cognome e ultima_retribuzione di tutti i dipendenti di qualifica OP e di livello 1. Il risultato deve essere ordinato per cognome.

45

Esercizio

16 Estrarre dal libro_paga una tabella contenente le colonne matricola, cognome e ultima_retribuzione di tutti i dipendenti di qualifica OP e di livello 1. Il risultato deve essere ordinato per cognome.

```
SELECT matricola, cognome, ultima_retribuzione
       FROM libro_paga
       WHERE qualifica = 'OP' AND livello ='1'
       ORDER BY cognome;
```

46

SELECT – FROM – WHERE + LIKE

SELECT *[cols]* **FROM** *[table]* **WHERE** c1 **LIKE** *'expr1'*…

Restituisce una tabella in cui ciascuna riga è composta dal contenuto delle colonne *[cols]* in ogni riga della tabella *[table]* in cui il valore *v1 (di tipo stringa)* riportato nella colonna *c1* corrisponde a quanto indicato nell'espressione *expr1.*

L'espressione *expr1* indica delle *caratteristiche* che il valore contenuto nella colonna deve avere affinchè il **LIKE** abbia successo. Tale espressione utilizza caratteri frammisti a *operatori* detti *whild cards:* **%**, **[ ]**, **_** *(underscore),* **^** *e* **–**

47

ESPRESSIONI REGOLARI (usate con LIKE)

L'espressione *expr1* indica delle *caratteristiche* che il valore contenuto nella colonna deve avere affinchè il **LIKE** abbia successo. Tale espressione utilizza caratteri frammisti a *operatori* detti *whild cards:* **%**, **[ ]**, **_** *(underscore),* **^** *e* **–**

	Sim.	Descrizione	Espressione	LIKE
	%	*0 o più caratteri in sequenza*	**R%; A%Z**	**'Rosa', 'REATO', 'RR223'; AZ, Ax2Z, ABCDZ, AAZ;**
	_	*Un carattere (qualsiasi)*	**Cap._**	**Cap.1, Cap.2, Cap.3, Cap.A** (e non **Cap.23, Cap.xxs**)
Non in SQLite	**[ ]**	*Uno qualsiasi dei caratteri in parentesi.*	**Cap.[123ab]**	**Cap.1, Cap.2, Cap.3, Cap.a, Cap.b** (e non **Cap.4**)
Non in SQLite	**^**	*Un qualsiasi carattere esclusi quelli in parentesi*	**Fig.[^123]**	**Fig.4, Fig.5, Fig.A**… (e non **Fig.1, Fig.2** e **Fig.3**)
Non in SQLite	**–**	*Uno qualsiasi dei caratteri compresi nel range indicato*	**Fig.[1-7]**	**Fig.1, Fig.2** … **Fig.7** (e non **Fig.8, Fig.9**…)

48

SELECT – FROM – WHERE + LIKE

```
SELECT cognome FROM libro_paga WHERE cognome
  LIKE 'C%'
```

Elenca i soli cognomi che iniziano per C

Cognome
CASUCCI
CATALANO
CASTELLANO
CATALANOTTO
…

49

SELECT – FROM – WHERE + LIKE

```
SELECT matricola,cognome FROM libro_paga WHERE matricola
  LIKE 'P___[13579]'
```

Elenca i soli cognomi che iniziano con una P + tre caratteri qualsiasi + uno qualsiasi dei caratteri indicati tra parentesi

Matricola	Cognome
P0211	Basilico
P0311	Rissone
P0223	Tripiedi
P0233	Farsetti
…	…

50

SELECT – FROM – WHERE + LIKE

```
SELECT cognome FROM libro_paga WHERE cognome
  LIKE '_a%';
```

Elenca i soli cognomi la cui seconda lettera è la a;

Cognome
Basilico
Favella
Farsetti
Laurenti
…

51

SELECT – FROM – WHERE + LIKE [+ AND/OR]

```
SELECT * FROM libro_paga WHERE data_assunzione LIKE '%1998'
  AND qualifica = 'IMP';
```

Elenca i record dei dipendenti assunti nel 1999 con la qualifica di Impiegati (IMP);

matricola	cognome	data_assunzione	qualifica	livello	ultima_retribuzione
P0412	Isabella	18/03/1998	IMP	2	2232000

52

Esercizio

17 Estrarre dal libro_paga tutti i record in cui il cognome comincia per ‘Di(spazio)’ o ‘De(spazio)’.

18 Estrarre dal libro_paga tutti i record dei dipendenti assunti nel 1999

53

Esercizio

17 Estrarre dal libro_paga tutti i record in cui il cognome comincia per ‘Di(spazio)’ o ‘De(spazio)’.

```
SELECT * FROM libro_paga WHERE cognome LIKE 'Di %' OR
  cognome LIKE 'De %'
```

18 Estrarre dal libro_paga tutti i record dei dipendenti assunti nel 1999

```
SELECT * FROM libro_paga WHERE data LIKE '%1999'
```

54

Query aggregate

A differenza delle query ordinarie, queste effettuano delle *operazioni di «sintesi»* sulle righe estratte dalla tabella e ne visualizzano il risultato.

Per esempio:

```
SELECT * FROM libro_paga WHERE qualifica = 'OP'
```

Seleziona tutti i record in cui il campo qualifica contiene il valore OP e li visualizza;

invece:

```
SELECT count(*) FROM libro_paga WHERE qualifica = 'OP'
```

Seleziona tutti i record in cui il campo qualifica contiene il valore OP, li conta e visualizza il loro numero;

55

Alcuni operatori di aggregazione

Fun.	Descrizione	Esempio	Risultato
Count()	Conta i record selezionati	**SELECT Count(***[cols]***) FROM** *[tab]*	Numero di record in *[tab]*
Sum(*cols***)**	Somma il valore dei campi cols (se numerici) selezionati	**SELECT Sum(**col_importi**) FROM** *[tab]*	Somma di tutti gli importi delle righe selezionate in *[tab]*
AVG(*cols***)**	Calcola la media tra i valori assunti dal campo cols nei record selezionati	**SELECT AVG(**col_importi**) FROM** *[tab]*	
Max(*cols***)**	Estrae il valore massimo tra quelli assunti dai campi cols selezionati	**SELECT Max(**col_importi**) FROM** *[tab]*	Visualizza il massimo valore del campo col_importi
Min(*cols***)**	Idem ma calcola il minimo	**SELECT Min(**col_importi**) FROM** *[tab]*	Idem ma calcola il minimo

56

SELECT – FROM – WHERE + LIKE [+ AND/OR]

```
SELECT MAX(ultima_retribuzione) FROM libro_paga
```

Estrae il valore più alto per il campo **`ultima_retribuzione`** tra tutti i record di **`libro_paga`**

MAX(ultima_retribuzione)
2883000

57

Esercizio

19 Estrarre dal **`libro_paga`** la somma di tutte le retribuzioni pagate

20 Estrarre dal **`libro_paga`** la retribuzione media dei dipendenti di qualifica `OP` e livello `1`

58

Esercizio

19 Estrarre dal **libro_paga** la somma di tutte le retribuzioni pagate

```
SELECT SUM(ultima_retribuzione) FROM libro_paga;
```

20 Estrarre dal **libro_paga** la retribuzione media dei dipendenti di qualifica `OP` e livello `1`

```
SELECT AVG(ultima_retribuzione) FROM libro_paga
  WHERE qualifica = 'OP' AND livello = '1'
```

59

Esercizio

21 Estrarre dal **libro_paga** la somma di tutte le retribuzioni pagate per i dipendenti di qualifica *IMP*

22 Estrarre dal **libro_paga** la retribuzione minima e massima dei dipendenti di qualifica *IMP* dei livelli tra *1* e *3*

60

Esercizio

21 Estrarre dal **`libro_paga`** la somma di tutte le retribuzioni pagate per i dipendenti di qualifica *IMP*

```
SELECT SUM(ultima_retribuzione) FROM libro_paga
  WHERE qualifica = 'IMP';
```

22 Estrarre dal **`libro_paga`** la retribuzione minima e massima dei dipendenti di qualifica *IMP* dei livelli tra *1* e *3*

```
SELECT MIN(ultima_retribuzione),MAX(ultima_retribuzione)
  FROM libro_paga WHERE qualifica = 'IMP'
  AND livello BETWEEN 1 AND 3
```

61

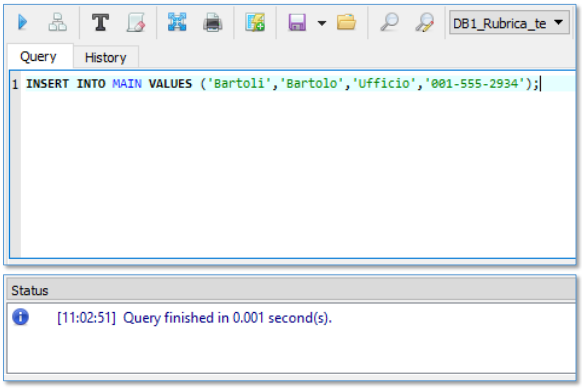
Inserimento nuovi record in SQL

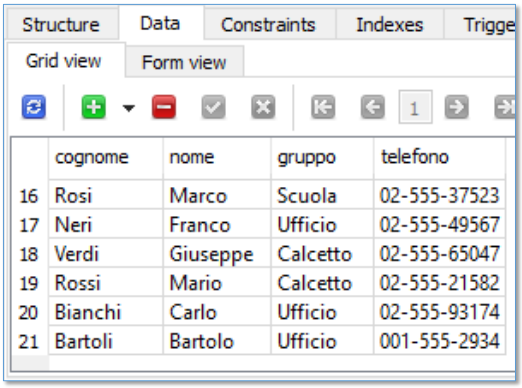
Database: **`rubrica_telefonica_1`** **Table**: `MAIN`

	cognome	nome	gruppo	telefono
1	Gargiulo	Pasquale	Ufficio	081-888-2444
2	Esposito	Gennaro	*NULL*	081-888-9999
3	Di Bartolo	Bartolo	*NULL*	02-555-2223
4	Di Stefano	Stefania	Ufficio	0323-239137
5	Di Vittorio	Vittoria	*NULL*	493-5763001
6	De Carolis	Carola	*NULL*	456-7773830
7	De Giorgis	Giorgio	Calcetto	456-2003843
8	Marea	Mara	Famiglia	456-7366944
9	Giannini	Giunone	Famiglia	456-8704813
10	Giannini	Gino	Famiglia	456-4026990

62

Inserimento nuovi record in SQL





63

INSERT INTO - VALUES

INSERT INTO *[tab]* **(**c_1**,** c_2**,** c_3**,…,** c_n**)**
VALUES (v_1**,** v_2**,** v_3**,…,** v_n**);**

Inserisce nella tabella *[tab]* una nuova riga in cui nelle colonne $c_1, \ldots, c_n$ sono posti rispettivamente i valori $v_1, \ldots, v_n$.

I due elenchi devono rispettare le seguenti regole:

1. Stesso numero di colonne e di valori
2. Stesso ordine tra le colonne e i valori e corrispondenza tra ogni colonna e il rispettivo valore
3. I valori devono essere del tipo corretto
4. Non dimenticare di inserire tra i campi quelli che devono avere un valore non nullo
5. Non inserire nei campi «unique» valori già esistenti nella tabella

64

INSERT INTO - VALUES

```
INSERT INTO MAIN
  (cognome, nome, gruppo, telefono)
  VALUES ('Bartoli','Bartolo',Ufficio,'001-555-2934')
```

Inserisce nella tabella `MAIN` una nuova riga in cui nelle colonne `nome,cognome,gruppo, telefono` sono posti rispettivamente i valori *'Bartoli','Bartolo','Ufficio', '001-555-2934'*.

65

INSERT INTO – VALUES multipli

```
INSERT INTO MAIN
  (cognome, nome, gruppo, telefono)
  VALUES ('Bartoli',Biagio','Ufficio','001-555-24493'),
         ('Bartoli','Barnaba','Calcetto','001-555-2934'),
```

Inserisce nella tabella `MAIN` due nuove righe nelle cui colonne `cognome,nome,gruppo,telefono` sono posti rispettivamente i valori indicati in ciascuna n-pla che segue VALUES;

66

INSERT INTO - VALUES

INSERT INTO *[tab]* **VALUES (**v_1, v_2, v_3,…, v_n**);**

! Se la query fornisce un valore per tutti i campi, nello stesso ordine in cui questi appaiono nella tabella, la n-pla $c_1,\ldots,c_n$ può essere omessa

```
INSERT INTO MAIN
    VALUES ('Bartoli','Bartolo',Ufficio,'001-555-2934');
```

```
INSERT INTO MAIN
    VALUES ('Bartoli',Biagio','Ufficio','001-555-24493'),
           ('Bartoli','Barnaba','Calcetto','001-555-2934');
```

67

INSERT INTO - VALUES

INSERT INTO *[tab]* **(**c_1, c_2, c_3,…, c_n**)**
VALUES (v_1, v_2, v_3,…, v_n**);**

Inserisce nella tabella *[tab]* una nuova riga in cui nelle colonne $c_1,\ldots,c_n$ sono posti rispettivamente i valori $v_1,\ldots,v_n$.

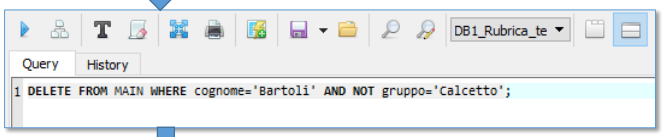
I due elenchi devono rispettare le seguenti regole:

1. Stesso numero di colonne e di valori
2. Stesso ordine tra le colonne e i valori e corrispondenza tra ogni colonna e il rispettivo valore
3. I valori devono essere del tipo corretto
4. Non dimenticare di inserire tra i campi quelli che devono avere un valore non nullo
5. Non inserire nei campi «unique» valori già esistenti nella tabella

68

Cancellare i record da una tabella

19	Rossi	Mario	Calcetto	02-555-21582
20	Bianchi	Carlo	Ufficio	02-555-93174
21	Bartoli	Bartolo	Ufficio	001-555-2934
22	Bartoli	Biagio	Ufficio	001-555-24493
23	Bartoli	Barnaba	Calcetto	081-888-1121



17	Neri	Franco	Ufficio	02-555-49567
18	Verdi	Giuseppe	Calcetto	02-555-65047
19	Rossi	Mario	Calcetto	02-555-21582
20	Bianchi	Carlo	Ufficio	02-555-93174
21	Bartoli	Barnaba	Calcetto	081-888-1121

69

DELETE FROM-WHERE

DELETE FROM *[tab]* **WHERE** $c_1=v_1$**;**

Elimina dalla tabella *[tab]* tutte le righe in la colonna *c1* riporti il valore *v1;*

DELETE FROM *[tab]* **WHERE** $c_1=v_1$ **AND/OR** $c_2=v_2$ **…;**

La clausola WHERE del comando DELETE è sostanzialmente analoga a quella del comando SELECT;

70

DELETE FROM-WHERE

```
DELETE FROM MAIN WHERE cognome='Bartoli' AND
       NOT gruppo='Calcetto';
```

Elimina dalla rubrica tutti i record in cui il campo cognome è Bartoli, purchè il campo gruppo non sia 'Calcetto'.

```
DELETE FROM MAIN;
```

Elimina tutti i record dalla tabella MAIN.

71

Esercizio

23 Inserire nella tabella **MAIN** il contatto Bartoli Barbara, gruppo Famiglia, telefono: 081-888-2232;

24 Eliminare dalla tabella **MAIN** tutti i contatti in cui il campo gruppo è **NULL**

25 Contare, nella tabella **MAIN** tutti i record in cui il campo Cognome è **Giannini**

72

Esercizio

23 Inserire nella tabella **MAIN** il contatto Bartoli Barbara, gruppo Famiglia, telefono: 081-888-2232;

```
INSERT INTO MAIN
       VALUES ('Bartoli', 'Barbara', 'Famiglia', '081-888-2232');
```

24 Eliminare dalla tabella **MAIN** tutti i contatti in cui il campo gruppo è **NULL**

25 Contare, nella tabella **MAIN** tutti i record in cui il campo Cognome è **Giannini**

73

Esercizio

23 Inserire nella tabella **MAIN** il contatto Bartoli Barbara, gruppo Famiglia, telefono: 081-888-2232;

```
INSERT INTO MAIN
       VALUES ('Bartoli', 'Barbara', 'Famiglia', '081-888-2232');
```

24 Eliminare dalla tabella **MAIN** tutti i contatti in cui il campo gruppo è **NULL**

```
DELETE FROM MAIN WHERE gruppo IS NULL;
```

25 Contare, nella tabella **MAIN** tutti i record in cui il campo Cognome è **Giannini**

74

Esercizio

23 Inserire nella tabella **MAIN** il contatto Bartoli Barbara, gruppo Famiglia, telefono: 081-888-2232;

```
INSERT INTO MAIN
       VALUES ('Bartoli', 'Barbara', 'Famiglia', '081-888-2232');
```

24 Eliminare dalla tabella **MAIN** tutti i contatti in cui il campo gruppo è **NULL**

```
DELETE FROM MAIN WHERE gruppo IS NULL;
```

25 Contare, nella tabella **MAIN** tutti i record in cui il campo Cognome è **Giannini**

```
SELECT count(cognome) FROM MAIN WHERE cognome='Giannini';
```

75

Database con più tabelle

76

Database: Creditori_1

Database: **`Creditori_1`**

Table: **`anagrafica`**

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Default value
1	codice	INTEGER							*NULL*
2	ragione_sociale	STRING (20)							*NULL*
3	totale_crediti	INTEGER							0

Table: **`operazioni`**

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Default value
1	data	DATE							*NULL*
2	codice	STRING (20)							*NULL*
3	importo	INTEGER							*NULL*
4	descrizione	STRING (40)							*NULL*

77

Database: Creditori_1

Database: **`Creditori_1`**

Table: **`anagrafica`**

	codice	ragione_sociale	totale_crediti
1	1	Bianchi & C.	0
2	2	Martini & Verdi	0
3	3	Piercer & Crumbler	0
4	4	ACME spa	0
5	5	Pannofino Editore	0
6	6	Balocchi & Profumi	0
7	7	Glauco De Giorgis	0
8	8	Mill & Moll	0
9	9	Cavolini & figli	0
10	10	First Person Inc.	0

Table: **`operazioni`**

	data	codice	importo	descrizione
1	10/10/2019	1	10000	Fattura n.33
2	14/10/2019	7	56700	Fattura n.21
3	20/10/2019	3	10000	Cambio batt.
4	20/10/2019	1	4700	Fatt. no. 11b
5	28/10/2019	3	39000	Fatt. no. 12c
6	31/10/2019	2	100	Fattura n.100
7	02/11/2019	3	20390	Cambio batt.

78

SELECT su più tabelle

Table: **anagrafica**

	codice	ragione_sociale	totale_crediti
1	1	Bianchi & C.	0
2	2	Martini & Verdi	0
3	3	Piercer & Crumbler	0
4	4	ACME spa	0
5	5	Pannofino Editore	0
6	6	Balocchi & Profumi	0
7	7	Glauco De Giorgis	0
8	8	Mill & Moll	0
9	9	Cavolini & figli	0
10	10	First Person Inc.	0

Table: **operazioni**

	data	codice	importo	descrizione
1	10/10/2019	1	10000	Fattura n.33
2	14/10/2019	7	56700	Fattura n.21
3	20/10/2019	3	10000	Cambio batt.
4	20/10/2019	1	4700	Fatt. no. 11b
5	28/10/2019	3	39000	Fatt. no. 12c
6	31/10/2019	2	100	Fattura n.100
7	02/11/2019	3	20390	Cambio batt.

Estraiamo dalle due tabelle del database un elenco delle operazioni effettuate con aggiunte delle informazioni anagrafiche del creditore coinvolto. Per ciascuna record di **operazioni**, estraiamo i campi `data` e `importo` e dalla tabella **anagrafica**, la `ragione_sociale`, corrispondente al codice riportato nel record dell'operazione

79

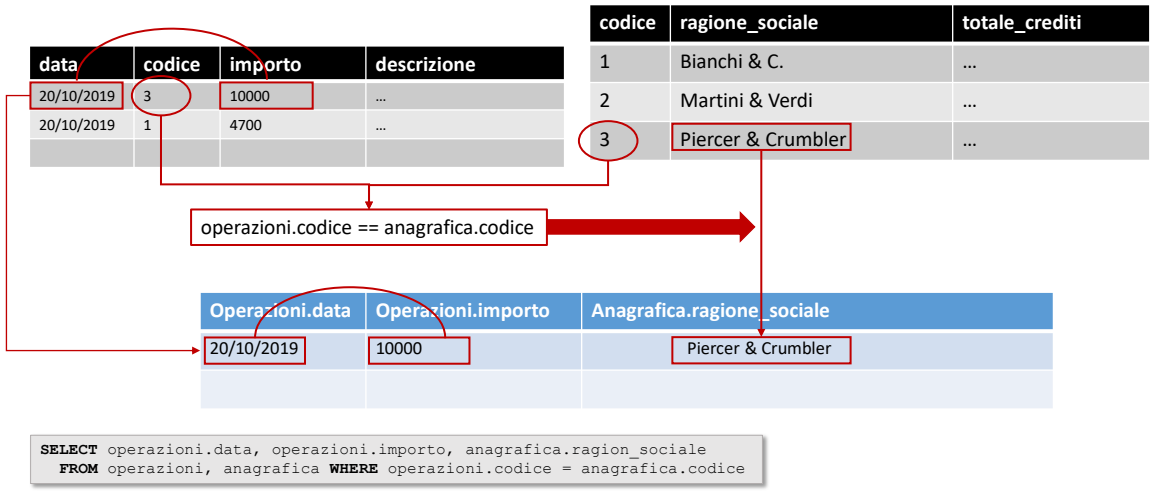
SELECT su più tabelle

Estraiamo dalle due tabelle del database un elenco delle operazioni effettuate con aggiunte delle informazioni anagrafiche del creditore coinvolto. Per ciascuna record di **operazioni**, estraiamo i campi `data` e `importo` e dalla tabella **anagrafica**, la `ragione_sociale`, corrispondente al codice riportato nel record dell'operazione

```
SELECT operazioni.data, operazioni.importo,
  anagrafica.ragion_sociale FROM operazioni, anagrafica
  WHERE operazioni.codice = anagrafica.codice
```

80

SELECT su più tabelle




```
SELECT operazioni.data, operazioni.importo, anagrafica.ragion_sociale
  FROM operazioni, anagrafica WHERE operazioni.codice = anagrafica.codice
```

81

SELECT su più tabelle

Estraiamo dalle due tabelle del database i totali delle operazioni effettuate col creditore *«Martini & Verdi»*. Dalla tabella **operazione**, estraiamo la somma del campo `importo` dei record con il codice che in **anagrafica**, è associato alla `ragione_sociale` data.

```
SELECT anagrafica.ragione_sociale,SUM(operazioni.importo)
FROM operazioni, anagrafica
  WHERE anagrafica.codice == operazioni.codice AND
  anagrafica.ragione_sociale == 'Martini & Verdi'
```

82

Esercizio

26 Data la descrizione dell'operazione *'Fattura n.33'*, nella tabella **operazioni**, selezionare il campo `importo` nella stessa tabella e il campo `ragione_sociale` della tabella **anagrafica**, del creditore coinvolto.

83

Esercizio

26 Data la descrizione dell'operazione *'Fattura n.33'*, nella tabella **operazioni**, selezionare il campo `importo` nella stessa tabella e il campo `ragione_sociale` della tabella **anagrafica**, del creditore coinvolto.

```
SELECT anagrafica.ragione_sociale, operazioni.importo
  FROM anagrafica, operazioni
  WHERE anagrafica.codice == operazioni.codice
  AND operazioni.descrizione == 'Fattura n.33'
```

84

Chiavi e relazioni…

85

Basi di Dati *(o Database)*

Per collegare tra loro le tabelle è necessario un campo ***chiave*** che identifica univocamente ogni record.

- *Carta di Identità, Codice fiscale, Matricola per lo studente, etc.*

Il collegamento tra le tabelle è realizzato mediante il cosiddetto *meccanismo della chiave esterna*.

I valori del campo chiave della tabella principale (che pertanto prende il nome di *chiave primaria*) sono duplicati in un campo apposito di ciascuna tabella dipendente

86

Basi di Dati *(o Database)*

La *chiave* è…

Unica. Non sono ammessi più record nella tabella principale che riportino il campo chiave (o la combinazione dei campi chiave) avente lo stesso valore.

Non nulla. Non sono ammessi record che contengano il campo chiave (o la combinazione dei campi chiave) vuoto

87

Basi di Dati *(o Database)*

Il campo ID è la *chiave primaria* e identifica ciascun record univocamente.

Il campo ID è aggiunto ai campi di tutte le tabelle in cui i record contengono dati *riconducibili* ai record contenuti nella tabella principale

Anagrafica clienti

ID	Cognome	Nome	Via	Città	Telefono
1001	Jones	Frank	100 Elm	Oakland	555-1789
1002	Smith	Susan	200 Main	Sarasota	555-1902
1003	Ashley	Bill	100 Woods	San Jose	555-0232

Vendite

ID	Data	Quantità
1001	1/10/99	10.00
1002	1/11/99	15.00
1002	1/12/99	20.00

88

Basi di Dati *(o Database)*

Il campo ID è la *chiave primaria* e identifica ciascun record univocamente.

Anagrafica clienti

ID	Cognome	Nome	Via	Città	Telefono
1001	Jones	Frank	100 Elm	Oakland	555-1789
1002	Smith	Susan	200 Main	Sarasota	555-1902
1003	Ashley	Bill	100 Woods	San Jose	555-0232

Vendite

ID	Data	Quantità
1001	1/10/99	10.00
1002	1/11/99	15.00
1002	1/12/99	20.00

Il campo ID è aggiunto ai campi di tutte le tabelle in cui i record contengono dati *riconducibili* ai record contenuti nella tabella principale

Il valore del campo ID del cliente è riprodotto nel campo analogo in tutti i record che descrivono i suoi acquisti

Tra le due tabelle è definita una *relazione*

89

Basi di Dati *(o Database)*

Tipi di relazioni:

Uno-a-uno: a un record della prima tabella corrisponde un solo record della seconda

Uno-a-molti: a un record della prima tabella corrispondono più record della seconda

Molti-a-molti: a più record della prima tabella corrispondono più record della seconda

90

Un esempio…

91

Basi di Dati *(o Database)*

Costruiamo un nuovo database: Corso_laurea_2, composto da due tabelle:

studenti

Matricola	Nome	Cognome
0112233001	Mario	Rossi
0112233003	Gennaro	Esposito
0112233004	Nicola	Lorusso
0112233005	Rosalia	Greco

esami

Sessione	Codice Corso	Matricola	Voto	Lode
20201012-01	0119943	0112233003	30	T
20201012-01	0119943	0112233001	18	F
20201012-01	0119934	0112233006	25	F

92

Il database Corso_di_laurea_2

Un'applicazione semplice può presentare anche una sola tabella, ma più spesso un DB è composto da più tabelle, collegate tra di loro tramite valori di campi comuni

studenti

Matricola	Nome	Cognome

esami

Sessione	Codice Corso	Matricola	Voto	Lode

Il campo matricola contiene un codice che identifica univocamente ciascuno studente e compare in qualsiasi dato riguardi quello studente. Nelle relazioni che riguardano gli studenti, esso è la chiave primaria.

93

Il database Corso_di_laurea_2

Un'applicazione semplice può presentare anche una sola tabella, ma più spesso un DB è composto da più tabelle, collegate tra di loro tramite valori di campi comuni

studenti

Matricola	Nome	Cognome

esami

Sessione	Codice Corso	Matricola	Voto	Lode

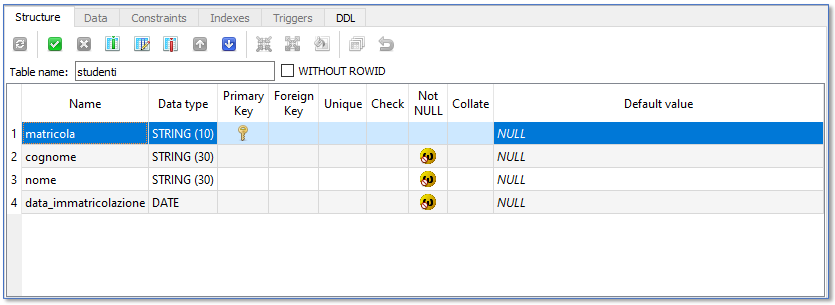
Il campo matricola contiene un codice che identifica univocamente ciascuno studente e compare in qualsiasi dato riguardi quello studente. Nelle relazioni che riguardano gli studenti, esso è la chiave primaria.

Il campo matricola, nella tabella esami è una chiave esterna. Ciò indica che questo campo è vincolato alla chiave principale (stessi valori, se contenuti nella tabella principale)

94

Il database Corso_di_laurea_2

1 Creare la tabella studenti. Il campo matricola è la chiave primaria

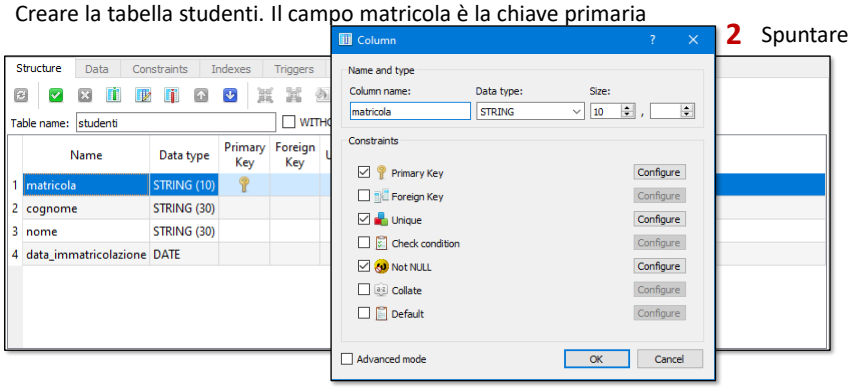


95

Il database Corso_di_laurea_2

1 Creare la tabella studenti. Il campo matricola è la chiave primaria

2 Spuntare la voce Primary Key

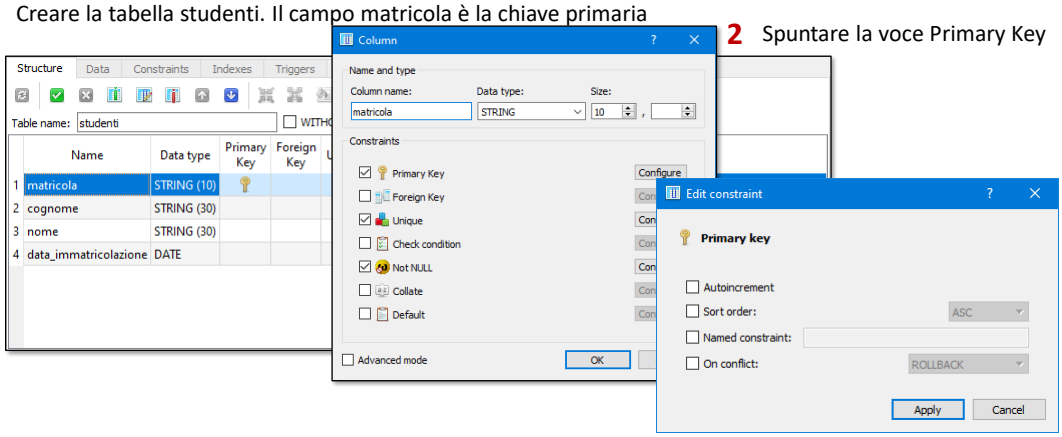


96

Il database Corso_di_laurea_2

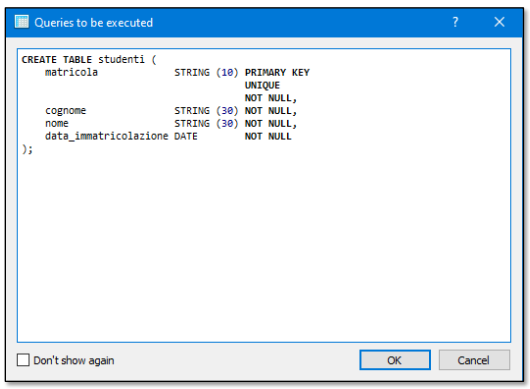
1 Creare la tabella studenti. Il campo matricola è la chiave primaria

2 Spuntare la voce Primary Key



97

Il database Corso_di_laurea_2

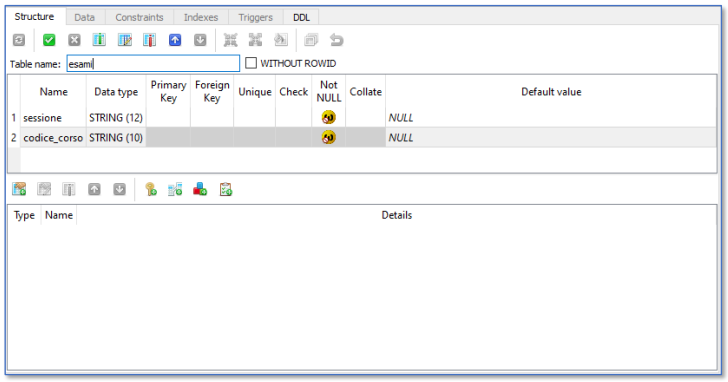


3 Comando SQL per la creazione della tabella

98

Il database Corso_di_laurea_2

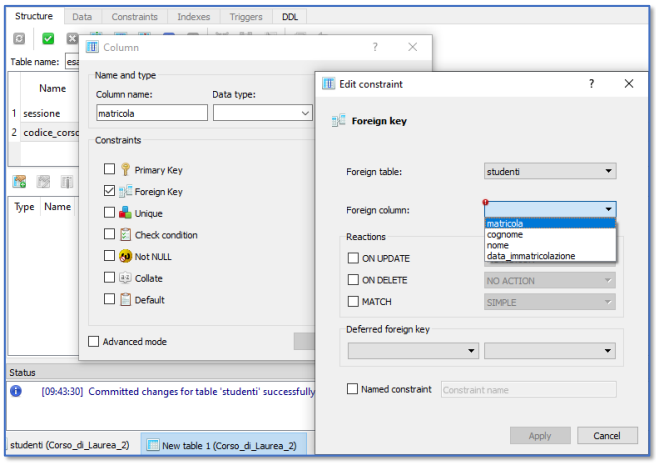
1 Creiamo la tabella esami. Il campo matricola è la chiave esterna



99

Il database Corso_di_laurea_2

1 Creiamo la tabella esami. Il campo matricola è la chiave esterna



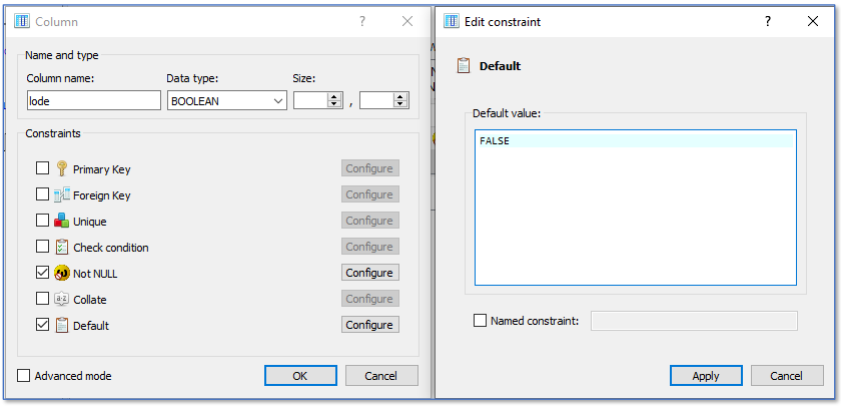
2 Spuntare la voce Foreign Key

3 Scegliere la tabella principale

4 Scegliere la chiave

100

Il database Corso_di_laurea_2

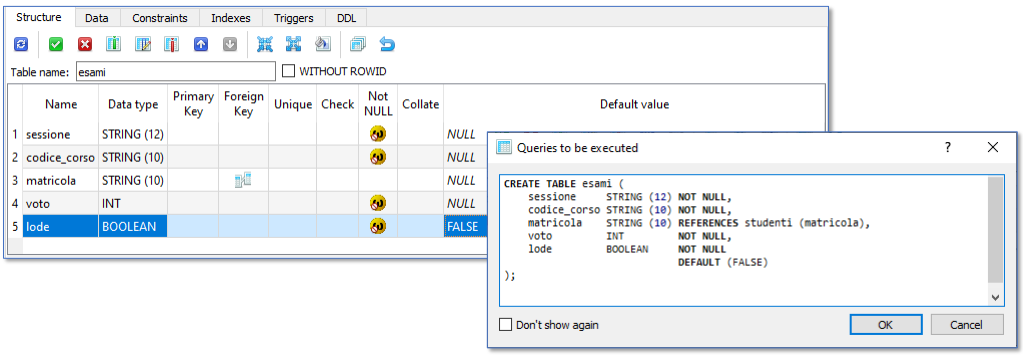


Nella tabella esami, il campo lode è di tipo booleano (TRUE, FALSE), ed è non nullo.

E' possibile impostare un valore di default (assegnato automaticamente se non indicato esplicitamente in fase di inserimento)

101

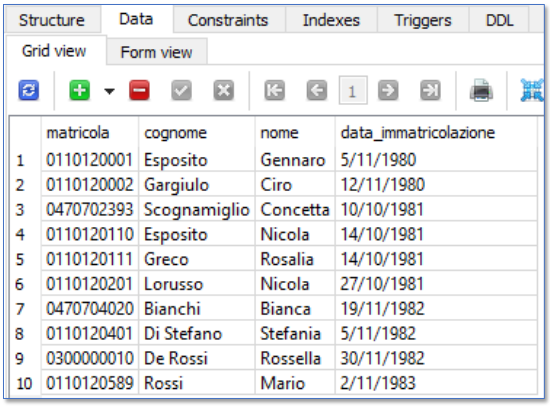
Il database Corso_di_laurea_2



La tabella esami.

102

Il database Corso_di_laurea_2



Structure | Data | Constraints | Indexes | Triggers | DDL

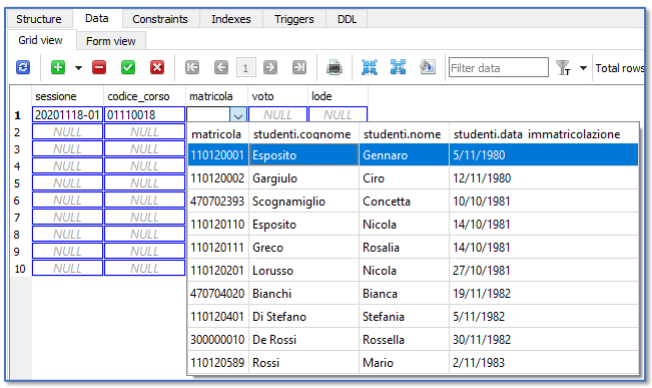
Grid view | Form view

	matricola	cognome	nome	data_immatricolazione
1	0110120001	Esposito	Gennaro	5/11/1980
2	0110120002	Gargiulo	Ciro	12/11/1980
3	0470702393	Scognamiglio	Concetta	10/10/1981
4	0110120110	Esposito	Nicola	14/10/1981
5	0110120111	Greco	Rosalia	14/10/1981
6	0110120201	Lorusso	Nicola	27/10/1981
7	0470704020	Bianchi	Bianca	19/11/1982
8	0110120401	Di Stefano	Stefania	5/11/1982
9	0300000010	De Rossi	Rossella	30/11/1982
10	0110120589	Rossi	Mario	2/11/1983

L'inserimento dei dati nella tabella studenti avviene nel modo che conosciamo

103

Il database Corso_di_laurea_2



Structure | Data | Constraints | Indexes | Triggers | DDL

Grid view | Form view

Filter data | Total rows

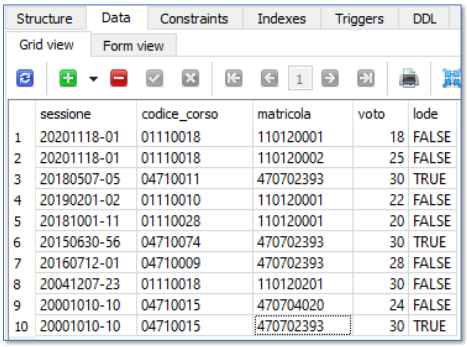
	sessione	codice_corso	matricola	voto	lode
1	20201118-01	01110018		NULL	NULL
2	NULL	NULL			
3	NULL	NULL			
4	NULL	NULL			
5	NULL	NULL			
6	NULL	NULL			
7	NULL	NULL			
8	NULL	NULL			
9	NULL	NULL			
10	NULL	NULL			

matricola	studenti.cognome	studenti.nome	studenti.data_immatricolazione
110120001	Esposito	Gennaro	5/11/1980
110120002	Gargiulo	Ciro	12/11/1980
470702393	Scognamiglio	Concetta	10/10/1981
110120110	Esposito	Nicola	14/10/1981
110120111	Greco	Rosalia	14/10/1981
110120201	Lorusso	Nicola	27/10/1981
470704020	Bianchi	Bianca	19/11/1982
110120401	Di Stefano	Stefania	5/11/1982
300000010	De Rossi	Rossella	30/11/1982
110120589	Rossi	Mario	2/11/1983

Durante l'inserimento dei dati nella tabella esami, il DBMS ci suggerisce i valori possibili per il campo matricola (la chiave esterna). Inserire valori diversi può causare un errore.

104

Il database Corso_di_laurea_2



Structure | Data | Constraints | Indexes | Triggers | DDL

Grid view | Form view

	sessione	codice_corso	matricola	voto	lode
1	20201118-01	01110018	110120001	18	FALSE
2	20201118-01	01110018	110120002	25	FALSE
3	20180507-05	04710011	470702393	30	TRUE
4	20190201-02	01110010	110120001	22	FALSE
5	20181001-11	01110028	110120001	20	FALSE
6	20150630-56	04710074	470702393	30	TRUE
7	20160712-01	04710009	470702393	28	FALSE
8	20041207-23	01110018	110120201	30	FALSE
9	20001010-10	04710015	470704020	24	FALSE
10	20001010-10	04710015	470702393	30	TRUE

I dati inseriti nella tabella esami

105

Database: Statistiche_calciatori_1

Database: **`Statistiche_calciatori_1`**

Table: **`calciatori`**

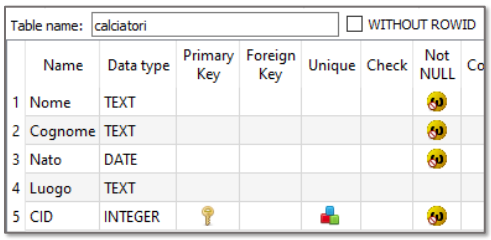


Table name: calciatori — WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Co
1	Nome	TEXT						
2	Cognome	TEXT						
3	Nato	DATE						
4	Luogo	TEXT						
5	CID	INTEGER						

Table: **`marcature`**

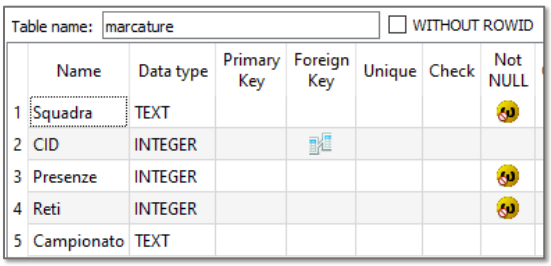


Table name: marcature — WITHOUT ROWID

	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL
1	Squadra	TEXT					
2	CID	INTEGER					
3	Presenze	INTEGER					
4	Reti	INTEGER					
5	Campionato	TEXT					

106

Esercizio

27

Inserire nella tabella **`calciatori`** i seguenti record

Nome	Cognome	Nato	Luogo	CID
Matteo	Politano	03/08/1993	Roma	1
Giacomo	Raspadori	18/02/2000	Bentivoglio	2

107

Esercizio

27

Inserire nella tabella **`calciatori`** i seguenti record

Nome	Cognome	Nato	Luogo	CID
Matteo	Politano	03/08/1993	Roma	1
Giacomo	Raspadori	18/02/2000	Bentivoglio	2

```
INSERT INTO calciatori
    …
```

108

Esercizio

27

Inserire nella tabella **`calciatori`** i seguenti record

Nome	Cognome	Nato	Luogo	CID
Matteo	Politano	03/08/1993	Roma	1
Giacomo	Raspadori	18/02/2000	Bentivoglio	2

```
INSERT INTO calciatori
    VALUES ('Matteo','Politano', 03/08/1993,'Roma',1),
           ('Giacomo','Raspadori',18/02/2000,'Bentivoglio', 2);
```

109

Esercizio

28

Inserire nella tabella **`marcature`** i seguenti record

Squadra	CID	Presenze	Reti	Campionato
Sassuolo	1	96	20	2015
Sassuolo	2	76	18	2018
Italia	1	7	3	2018
Italia	2	15	5	2021
Napoli	1	94	17	2020
Napoli	2	8	1	2022

110

Esercizio

29 Estrarre dal database *Statistiche_calciatori_1* la seguente informazione: Nome, Cognome e numero di reti del miglior marcatore nel Sassuolo

111

Esercizio

29 Estrarre dal database *Statistiche_calciatori_1* la seguente informazione: Nome, Cognome e numero di reti del miglior marcatore nel Sassuolo

```
SELECT calciatori.Nome, calciatori.Cognome,
       …
```

112

Esercizio

29 Estrarre dal database *Statistiche_calciatori_1* la seguente informazione: Nome, Cognome e numero di reti del miglior marcatore nel Sassuolo

```
SELECT calciatori.Nome, calciatori.Cognome,
       MAX(marcature.Reti)
FROM …
```

113

Esercizio

29 Estrarre dal database *Statistiche_calciatori_1* la seguente informazione: Nome, Cognome e numero di reti del miglior marcatore nel Sassuolo

```
SELECT calciatori.Nome, calciatori.Cognome,
       MAX(marcature.Reti)
FROM calciatori, marcature
…
```

114

Esercizio

29 Estrarre dal database *Statistiche_calciatori_1* la seguente informazione: Nome, Cognome e numero di reti del miglior marcatore nel Sassuolo

```
SELECT calciatori.Nome, calciatori.Cognome,
       MAX(marcature.Reti)
FROM calciatori, marcature
WHERE marcature.Squadra='Sassuolo'
       AND marcature.CID=calciatori.CID;
```

115

Esercizio

30 Estrarre dal database *Statistiche_calciatori_1* una tabella riportante squadra, presenze e delle reti del calciatore 'Politano' ordinate per anno in maniera crescente.

116

Esercizio

30 Estrarre dal database *Statistiche_calciatori_1* una tabella riportante squadra, presenze e delle reti del calciatore 'Politano' ordinate per anno in maniera crescente.

```
SELECT marcature.Squadra, marcature.Presenze, …
```

117

Esercizio

30 Estrarre dal database *Statistiche_calciatori_1* una tabella riportante squadra, presenze e delle reti del calciatore 'Politano' ordinate per anno in maniera crescente.

```
SELECT marcature.Squadra, marcature.Presenze, Marcature.Reti
FROM …
```

118

Esercizio

30 Estrarre dal database *Statistiche_calciatori_1* una tabella riportante squadra, presenze e delle reti del calciatore 'Politano' ordinate per anno in maniera crescente.

```
SELECT marcature.Squadra, marcature.Presenze, Marcature.Reti
FROM calciatori, marcature
WHERE …
```

119

Esercizio

30 Estrarre dal database *Statistiche_calciatori_1* una tabella riportante squadra, presenze e delle reti del calciatore 'Politano' ordinate per anno in maniera crescente.

```
SELECT marcature.Squadra, marcature.Presenze, Marcature.Reti
FROM calciatori, marcature
WHERE calciatori.Cognome='Politano'
      AND calciatori.CID=marcature.CID
…
```

120

Esercizio

30 Estrarre dal database *Statistiche_calciatori_1* una tabella riportante squadra, presenze e delle reti del calciatore 'Politano' ordinate per anno in maniera crescente.

```
SELECT marcature.Squadra, marcature.Presenze, Marcature.Reti
FROM calciatori, marcature
WHERE calciatori.Cognome='Politano'
      AND calciatori.CID=marcature.CID
ORDER BY marcature.Campionato;
```

121

Esercizio

31 Estrarre dal database *Statistiche_calciatori_1* la seguente informazione: In quale squadra il giocatore Raspadori ha totalizzato il maggior numero di presenze (e quante).

122

Mappe

Per lo studio e l'approfondimento degli argomenti trattati

123

Mappe

[Idb] Cap. 13: tutto, tralasciando gli esempi in Microsoft Access

[Sli] Le slides della lezione forniscono esempi dettagliati dell'utilizzo del linguaggio SQL, con riferimento a database d'esempio forniti con il materiale didattico.

[misc] Il W3 consortium (W3C), fornisce una accurata documentazione sul linguaggio SQL corredata da numerosi esempi ed esercizi online.

https://www.w3schools.com/sql/

124

Mappe

[misc] L'applicazione SQLiteStudio è disponibile al seguente URL:

https://sqlitestudio.pl/

125

Approfondimenti *(facoltativi)*

[misc] Per chi desidera approfondire l'argomento, si consiglia di fare riferimento a:

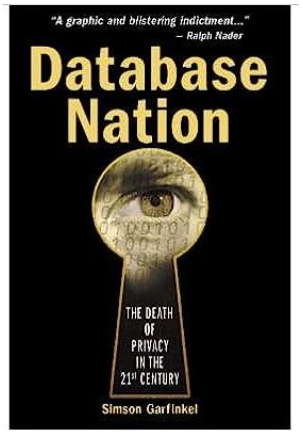
P. Atzeni, S. Ceri, P. Fraternali, S. Paraboschi, R. Torlone
Basi di Dati, V edizione (2018),
McGraw-Hill Education Italia, 766 pp.
ISBN: 8838694451

Si tratta di un testo completo ed esauriente, sebbene di livello più avanzato rispetto agli obbiettvi di questo corso.

126

Approfondimenti *(facoltativi)*

[misc]



Simson Garfinkel, **Database Nation**
O'Reilly, USA (2000), ISBN 1-56592-653-6

Lo sviluppo e la diffusione dei Database su Internet, ha amplificato in maniera non ancora compresa appieno la possibilità di interrogare le grandi quantità di dati personali (e potenzialmente sensibili) resi disponibili e i pericoli che ne derivano.

Simson Garfinkel è un autorevole ricercatore nel campo della sicurezza informatica. In questo volume anticipa, già all'inizio del secolo, i rischi connessi alla violazione della riservatezza dei dati, in termini di privacy e di libertà personale.

127