

Seconda parte

- Abbiamo visto la struttura
- Per modellare bene non basta

Vincoli di integrità

- Non sempre le nostre basi di dati sono "corrette" ..

Una base di dati "scorretta"

Esami	Studente	Voto	Lode	Corso
	276545	32	no	01
	276545	30	sì	02
	787643	27	sì	03
	739430	24	no	04

Studenti	Matricola	Cognome	Nome
	276545	Rossi	Mario
	787643	Neri	Piero
	787643	Bianchi	Luca

Una base di dati "scorretta"

Esami	Studente	Voto	Lode	Corso
	276545	32	no	01
	276545	30	sì	02
	787643	27	sì	03
	739430	24	no	04

Studenti	Matricola	Cognome	Nome
	276545	Rossi	Mario
	787643	Neri	Piero
	787643	Bianchi	Luca

Vincolo di integrità

- Proprietà che deve essere soddisfatta dalle istanze che rappresentano informazioni corrette per l'applicazione
- Un vincolo è una funzione booleana (un **predicato**):
 - associa ad ogni istanza il valore **vero** o **falso**

Vincoli di integrità, nota

- solo alcuni tipi di vincoli sono "supportati" dai DBMS:
 - ...il DBMS ne impedisce la violazione,
 - per i vincoli "non supportati", la responsabilità della verifica è dell'utente o del programmatore
- Vediamo i vincoli su un DBMS (oggi lo mostro io, prossimamente dovreste usarlo)

Verifica dei vincoli

- Vediamo fra poco con un esempio sul tipo più semplice di vincoli

Una base di dati "scorretta"

Esami	Studente	Voto	Lode	Corso
	276545	32	no	01
	276545	30	sì	02
	787643	27	sì	03
	739430	24	no	04

Studenti	Matricola	Cognome	Nome
	276545	Rossi	Mario
	787643	Neri	Piero
	787643	Bianchi	Luca

Tipi di vincoli

- vincoli **intrarelazionali**
 - vincoli su valori (o di **dominio**)
 - vincoli di **ennupla**
 - vincoli di **chiave**
- vincoli **interrelazionali**

Vincolo di dominio

- Pone condizioni sui valori ammissibili di un singolo attributo, ad esempio
 - il voto ha come tipo "intero"
 - ma deve essere compreso fra 18 e 30 (un sottoinsieme del dominio)

Vincolo di dominio, in concreto

- Una possibile sintassi:
 - combinazione booleana (AND, OR e NOT)
 - di condizioni semplici che confrontano un attributo con una costante

$(\text{Voto} \geq 18) \text{ AND } (\text{Voto} \leq 30)$

Verifica dei vincoli

- Vediamo su un sistema (è importante il concetto non i dettagli del linguaggio)
 - utilizziamo il servizio <https://sqliteonline.com/>
 - in particolare il sistema PostgreSQL offerto su di esso

Semplici passi

- <https://sqliteonline.com/>
- Prima creiamo e popoliamo le tabelle
- Poi vediamo i vincoli

SQL OnLine IDE

sqliteonline.com

File Owner DB Run Export Import Sign in

SQLite

Table

demo

```
1 SELECT * FROM demo;
```

ID	Name	Hint
1	SQL Online	for Data Science
2	Kirill N.	https://www.linkedin.com/in/sqliteonlinecom
3	Twitter	https://twitter.com/SQLiteOnlineCom
4	Chart	LINE-SELECT name, cos(id), sin(id) FR
5	Short CODE	s* tableName => SELECT * FROM table
6	SqLite 3.36.0	SQL OnLine on JavaScript
7	[RightClick] mouse	Opens many addit

SQL OnLine IDE

sqliteonline.com

File Owner DB Run Export Import Sign in

SQLite

Table

demo

SQLite

```
1 SELECT * FROM demo;
```

PostgreSQL

ID	Name	Hint
1	SQL Online	for Data Science
2	Kirill N.	https://www.linkedin.com/in/sqliteonlinecom
3	Twitter	https://twitter.com/SQLiteOnlineCom
4	Chart	LINE-SELECT name, cos(id), sin(id) FR
5	Short CODE	s* tableName => SELECT * FROM table
6	SqLite 3.36.0	SQL OnLine on JavaScript
7	[RightClick] mouse	Opens many addit

SQLite Online IDE interface showing a PostgreSQL connection attempt. The 'Click to connect' button is highlighted with a red circle.

ID	Name	Hint
1	SQL Online	for Data Science
2	Kirill N.	https://www.linkedin.com/in/sqliteonlinecom
3	Twitter	https://twitter.com/SQLiteOnlineCom
4	Chart	LINE-SELECT name, cos(id), sin(id) FR
5	Short CODE	s* tableName => SELECT * FROM table
6	SQLite 3.36.0	SQL OnLine on JavaScript
7	[RightClick] mouse	Opens many additional
8	Left-Button Table	[RightClick] mouse "Context menu"

SQLite Online IDE interface showing the PostgreSQL database selected. The main editor area contains the text: **Inseriamo qui la definizione della base di dati (lucido seguente)**

```
DROP TABLE IF EXISTS Esami;
CREATE TABLE Esami (
  Studente INTEGER,
  Voto INTEGER,
  Lode BOOLEAN,
  Corso INTEGER);
INSERT INTO Esami VALUES (276545, 30, TRUE, 02);
INSERT INTO Esami VALUES (787643, 27, TRUE, 03);
INSERT INTO Esami VALUES (739430, 24, FALSE, 04);
INSERT INTO Esami VALUES (276545, 32, FALSE, 01);
DROP TABLE IF EXISTS Studenti;
CREATE TABLE Studenti (
  Matricola INTEGER,
  Cognome TEXT,
  Nome TEXT);
INSERT INTO Studenti VALUES (276545, 'Rossi','Mario');
INSERT INTO Studenti VALUES (787643, 'Neri','Piero');
INSERT INTO Studenti VALUES (787643, 'Bianchi','Luca');
```

The screenshot shows the SQLite Online IDE interface. The browser address bar displays 'sqliteonline.com'. The main workspace contains the following SQL code:

```
1 DROP TABLE IF EXISTS Esami;
2 CREATE TABLE Esami (
3     Studente INTEGER,
4     Voto INTEGER,
5     Lode BOOLEAN,
6     Corso INTEGER);
7 INSERT INTO Esami VALUES (276545, 30, TRUE, 02);
8 INSERT INTO Esami VALUES (787643, 27, TRUE, 03);
9 INSERT INTO Esami VALUES (739430, 24, FALSE, 04);
10 INSERT INTO Esami VALUES (276545, 32, FALSE, 01);
11 DROP TABLE IF EXISTS Studenti;
12 CREATE TABLE Studenti (
13     Matricola INTEGER,
14     Cognome TEXT,
15     Nome TEXT);
16 INSERT INTO Studenti VALUES (276545, 'Rossi', 'Mario');
17 INSERT INTO Studenti VALUES (787643, 'Neri', 'Piero');
18 INSERT INTO Studenti VALUES (787643, 'Bianchi', 'Luca');
```

The interface includes a sidebar with database options (SQLite, MariaDB, PostgreSQL, MS SQL, Oracle, Docker, Syntax, Business) and a table named 'demo'. The bottom of the window shows a file explorer with files: 'SQLite (1).csv', 'SQLite.csv', 'PostgreSQL (1).csv', and 'PostgreSQL.json'. A 'Mostra tutto' button is visible in the bottom right corner.

The screenshot shows the SQLite Online IDE interface. The browser address bar displays `sqliteonline.com`. The top navigation bar includes a menu icon, `File`, `Owner DB`, a circled `Run` button, `Export`, and `Import`. The left sidebar lists database engines: SQLite, MariaDB, PostgreSQL (selected), MS SQL, Oracle, Docker, Syntax, and Business. Under PostgreSQL, a database instance `TEST - v.4.2.180` is shown with a `demo` table.

The main editor area contains the following SQL code:

```
1 DROP TABLE IF EXISTS Esami;
2 CREATE TABLE Esami (
3     Studente INTEGER,
4     Voto INTEGER,
5     Lode BOOLEAN,
6     Corso INTEGER);
7 INSERT INTO Esami VALUES (276545, 30, TRUE, 02);
8 INSERT INTO Esami VALUES (787643, 27, TRUE, 03);
9 INSERT INTO Esami VALUES (739430, 24, FALSE, 04);
10 INSERT INTO Esami VALUES (276545, 32, FALSE, 01);
11 DROP TABLE IF EXISTS Studenti;
12 CREATE TABLE Studenti (
13     Matricola INTEGER,
14     Cognome TEXT,
15     Nome TEXT);
16 INSERT INTO Studenti VALUES (276545, 'Rossi', 'Mario');
17 INSERT INTO Studenti VALUES (787643, 'Neri', 'Piero');
18 INSERT INTO Studenti VALUES (787643, 'Bianchi', 'Luca');
```

The bottom of the interface shows a file explorer with `SQLite (1).csv`, `SQLite.csv`, `PostgreSQL (1).csv`, and `PostgreSQL.json`. A `Mostra tutto` button is visible on the right.

SQL OnLine IDE

sqliteonline.com

File Owner DB Run Export Import Sign in

SQLite PostgreSQL

TEST - v.4.2.180

Table

- demo
- esami**
- studenti

```
1 DROP TABLE IF EXISTS Esami;
2 CREATE TABLE Esami (
3     Studente INTEGER,
4     Voto INTEGER,
5     Lode BOOLEAN,
6     Corso INTEGER);
7 INSERT INTO Esami VALUES (276545, 30, TRUE, 02);
8 INSERT INTO Esami VALUES (787643, 27, TRUE, 03);
9 INSERT INTO Esami VALUES (739430, 24, FALSE, 04);
10 INSERT INTO Esami VALUES (276545, 32, FALSE, 01);
11 DROP TABLE IF EXISTS Studenti;
12 CREATE TABLE Studenti (
13     Matricola INTEGER,
14     Cognome TEXT,
15     Nome TEXT);
16 INSERT INTO Studenti VALUES (276545, 'Rossi', 'Mario');
17 INSERT INTO Studenti VALUES (787643, 'Neri', 'Piero');
18 INSERT INTO Studenti VALUES (787643, 'Bianchi', 'Luca');
```

SQLite (1).csv SQLite.csv PostgreSQL (1).csv PostgreSQL.json Mostra tutto

SQL OnLine IDE | sql online ide export - Google

sqliteonline.com

File | Owner DB | Run | Export | Import | Sign in

SQLite | PostgreSQL | MariaDB

PostgreSQL

TEST - v.4.2.180

Table

- demo
- esami

Column

- studente integer
- voto integer
- lode boolean
- corso integer

1 `SELECT * FROM esami`

studente	voto	lode	corso
276545	30	true	2
787643	27	true	3
739430	24	false	4
276545	32	false	1

SQLite (1).csv | SQLite.csv | PostgreSQL (1).csv | PostgreSQL.json | Mostra tutto

Semplici passi

- <https://sqliteonline.com/>
- Prima creiamo e popoliamo le tabelle
- Poi vediamo i vincoli

Vincolo di dominio, in concreto

- Una possibile sintassi:
 - combinazione booleana (AND, OR e NOT)
 - di condizioni semplici che confrontano un attributo con una costante

$(\text{Voto} \geq 18) \text{ AND } (\text{Voto} \leq 30)$

Una sintassi reale (SQL)

```
ALTER TABLE Esami ADD CONSTRAINT IntervalloVoto  
CHECK (Voto >= 18 AND Voto <= 30);
```

- Proviamo a definire il vincolo sulla nostra base di dati

The screenshot shows the SQLiteonline.com web interface. The browser address bar displays `https://sqliteonline.com`. The interface includes a navigation menu with options like 'File', 'Owner DB', 'Run', and 'Export'. A sidebar on the left lists various database systems: SQLite, MariaDB, PostgreSQL (selected), MS SQL, Oracle, Docker, Syntax, and Business. Under PostgreSQL, a 'TEST - v.4.2.180' instance is shown with tables 'demo', 'esami', and 'studenti'. The main editor area contains the following SQL query:

```
1 ALTER TABLE Esami ADD CONSTRAINT IntervalloVoto CHECK (Voto >= 18 AND Voto <= 30);
2
```

A red error message box is overlaid on the interface, stating: "PostgreSQL 23514 check constraint "IntervalloVoto" of relation "esami" is violated by some row".

At the bottom of the interface, there is a 'History' section and a file manager showing several files: 'SQLite (1).csv', 'SQLite.csv', 'PostgreSQL (1).csv', and 'PostgreSQL.json'. A 'Mostra tutto' button is visible next to the file list.

Il vincolo funziona

- Se la base di dati non avesse avuto la ennupla incriminata, allora non avremmo avuto problemi

```
DROP TABLE IF EXISTS Esami;
```

```
CREATE TABLE Esami (
```

```
    Studente INTEGER,
```

```
    Voto INTEGER,
```

```
    Lode BOOLEAN,
```

```
    Corso INTEGER);
```

```
INSERT INTO Esami VALUES (276545, 30, TRUE, 02);
```

```
INSERT INTO Esami VALUES (787643, 27, TRUE, 03);
```

```
INSERT INTO Esami VALUES (739430, 24, FALSE, 04);
```

```
INSERT INTO Esami VALUES (276545, 32, FALSE, 01);
```

Browser window showing the SQLite online IDE interface. The URL is <https://sqliteonline.com>. The interface includes a menu bar with options like File, Owner DB, Run, Export, and Import. The main area displays SQL code for creating and inserting data into a table named 'Esami' in PostgreSQL. The code includes a CHECK constraint named 'IntervalloVoto'.

```
1 DROP TABLE IF EXISTS Esami;
2 CREATE TABLE Esami (
3     Studente INTEGER,
4     Voto INTEGER,
5     Lode BOOLEAN,
6     Corso INTEGER);
7 INSERT INTO Esami VALUES (276545, 30, TRUE, 02);
8 INSERT INTO Esami VALUES (787643, 27, TRUE, 03);
9 INSERT INTO Esami VALUES (739430, 24, FALSE, 04);
10 ALTER TABLE Esami ADD CONSTRAINT IntervalloVoto CHECK (Voto >= 18 AND Voto <= 30);
11
```

The interface also shows a sidebar with database options (SQLite, MariaDB, PostgreSQL, MS SQL, Oracle, Docker, Syntax, Business) and a table list for PostgreSQL (demo, esami, studenti). A 'History' section at the bottom shows the executed SQL command.

OK

SQLite (1).csv | SQLite.csv | PostgreSQL (1).csv | PostgreSQL.json

Verifica dei vincoli

- Per vedere il concetto abbiamo visto
 - prima i dati, poi i vincoli
 - su una base di dati esistente
 - proviamo a definire un vincolo, il sistema controlla il soddisfacimento
 - se è violato, non permette di definirlo

Verifica dei vincoli

- Per vedere il concetto abbiamo visto
 - prima i dati, poi i vincoli
 - su una base di dati esistente
 - proviamo a definire un vincolo, il sistema controlla il soddisfacimento
 - se è violato, non permette di definirlo
- In realtà, di solito
 - prima i vincoli, poi i dati
 - definiamo lo schema, con i vincoli
 - modifichiamo i dati (inserimenti, eliminazioni, ...), il sistema controlla i vincoli
 - se un vincolo è violato, la modifica viene rifiutata

Il vincolo funziona

- A partire dalla base di dati corretta

```
DROP TABLE IF EXISTS Esami;  
CREATE TABLE Esami (  
  Studente INTEGER,  
  Voto INTEGER,  
  Lode BOOLEAN,  
  Corso INTEGER);  
ALTER TABLE Esami ADD CONSTRAINT IntervalloVoto  
  CHECK (Voto >= 18 AND Voto <= 30);  
INSERT INTO Esami VALUES (276545, 30, TRUE, 02);  
INSERT INTO Esami VALUES (787643, 27, TRUE, 03);  
INSERT INTO Esami VALUES (739430, 24, FALSE, 04);
```

- Proviamo ad inserire la ennupla scorretta

```
INSERT INTO Esami VALUES (276545, 32, FALSE, 01);
```

The screenshot shows the SQLiteOnline web interface. The browser address bar displays `https://sqliteonline.com`. The interface includes a navigation menu with options like 'File', 'Owner DB', 'Run', 'Export', and 'Import'. A sidebar on the left lists database engines: SQLite, MariaDB, PostgreSQL (selected), MS SQL, Oracle, Docker, and Syntax. Under PostgreSQL, a specific instance 'TEST - v.4.2.180' is selected, and a 'Table' section lists 'demo', 'esami', and 'studenti'. The main editor area contains the following SQL code:

```
1 DROP TABLE IF EXISTS Esami;
2 CREATE TABLE Esami (
3     Studente INTEGER,
4     Voto INTEGER,
5     Lode BOOLEAN,
6     Corso INTEGER);
7 ALTER TABLE Esami ADD CONSTRAINT IntervalloVoto
8     CHECK (Voto >= 18 AND Voto <= 30);
9 INSERT INTO Esami VALUES (276545, 30, TRUE, 02);
10 INSERT INTO Esami VALUES (787643, 27, TRUE, 03);
11 INSERT INTO Esami VALUES (739430, 24, FALSE, 04);
12
13
```

Below the code editor is a 'History' section with tabs for 'Table', 'Syntax', and 'History'. A preview of the executed SQL code is shown in a light blue box. At the bottom, a file manager displays several CSV and JSON files: 'SQLite (1).csv', 'SQLite.csv', 'PostgreSQL (1).csv', and 'PostgreSQL.json'. A 'Mostra tutto' button is visible on the right side of the file manager.

OK

The screenshot shows the SQLiteonline.com web interface. The browser address bar displays `https://sqliteonline.com`. The interface includes a top navigation bar with options like 'File', 'Owner DB', 'Run', and 'Export'. A sidebar on the left lists various database engines: SQLite, MariaDB, PostgreSQL (selected), MS SQL, Oracle, Docker, and Syntax. Under PostgreSQL, a database instance 'TEST - v.4.2.180' is shown with tables 'demo', 'esami', and 'studenti'. The main editor area contains a PostgreSQL query: `1 INSERT INTO Esami VALUES (276545, 32, FALSE, 01);`. A red error message box is overlaid on the query, stating: 'PostgreSQL 23514 new row for relation "esami" violates check constraint "intervallovoto"'. Below the editor is a 'History' section with a table view and a 'Syntax' view. The bottom of the interface shows a file manager with several files: 'SQLite (1).csv', 'SQLite.csv', 'PostgreSQL (1).csv', and 'PostgreSQL.json'. A 'Mostra tutto' button is visible on the right.

Tipi di vincoli

- vincoli intrarelazionali
 - vincoli su valori (o di dominio)
 - vincoli di **ennupla**
 - vincoli di chiave
- vincoli interrelazionali

Vincolo di ennupla

- Esprime condizioni sui valori di ciascuna ennupla, indipendentemente dalle altre ennuple
- Il vincolo di dominio è un caso particolare

Sintassi ed esempi

- La sintassi: come il vincolo di dominio, ma in generale su più attributi
 - combinazione booleana
 - di condizioni semplici che confrontano un attributo con una costante o due attributi (o anche espressioni su di essi)

NOT ((Voto <> 30) AND (Lode = "sì"))

Nella sintassi reale (SQL)

```
ALTER TABLE Esami ADD CONSTRAINT LodeSoloConTrenta  
CHECK (NOT (Voto<>30 AND Lode=True));
```

- Verifichiamo anche questo sul sistema
 - prima con la violazione
 - poi senza

Prima verifica

```
DROP TABLE IF EXISTS Esami;  
CREATE TABLE Esami (  
  Studente INTEGER,  
  Voto INTEGER,  
  Lode BOOLEAN,  
  Corso INTEGER);  
INSERT INTO Esami VALUES (276545, 30, TRUE, 02);  
INSERT INTO Esami VALUES (787643, 27, TRUE, 03);  
INSERT INTO Esami VALUES (739430, 24, FALSE, 04);  
ALTER TABLE Esami ADD CONSTRAINT LodeSoloConTrenta  
  CHECK (NOT (Voto<>30 AND Lode=True));
```

Seconda verifica

```
DROP TABLE IF EXISTS Esami;
CREATE TABLE Esami (
  Studente INTEGER,
  Voto INTEGER,
  Lode BOOLEAN,
  Corso INTEGER);
ALTER TABLE Esami ADD CONSTRAINT LodeSoloConTrenta
  CHECK (NOT (Voto<>30 AND Lode=True));
INSERT INTO Esami VALUES (276545, 30, TRUE, 02);
--INSERT INTO Esami VALUES (787643, 27, TRUE, 03);
INSERT INTO Esami VALUES (739430, 24, FALSE, 04);

INSERT INTO Esami VALUES (787643, 27, TRUE, 03);
```

Vincoli di ennupla, altro esempio

Stipendi	Impiegato	Lordo	Ritenute	Netto
	Rossi	55.000	12.500	42.500
	Neri	45.000	10.000	35.000
	Bruni	47.000	11.000	36.000

Ritenute ≥ 0 AND (Lordo = (Ritenute + Netto))

Vincoli di ennupla, violazione

Stipendi	Impiegato	Lordo	Ritenute	Netto
	Rossi	55.000	12.500	42.500
	Neri	45.000	10.000	35.000
	Bruni	50.000	11.000	36.000

$$\text{Lordo} = (\text{Ritenute} + \text{Netto})$$

Tipi di vincoli

- vincoli intrarelazionali
 - vincoli su valori (o di dominio)
 - vincoli di enunzia
 - vincoli di **chiave**
- vincoli interrelazionali

Il tipo di vincolo più importante

- Matricola
- Codice fiscale
- Codice del corso
- ...

Identificazione delle ennuple

Matricola	Cognome	Nome	Corso	Nascita
27655	Rossi	Mario	Ing Inf	5/12/98
78763	Rossi	Mario	Ing Inf	3/11/96
65432	Neri	Piero	Ing Mecc	10/7/99
87654	Neri	Mario	Ing Inf	3/11/96
67653	Rossi	Piero	Ing Mecc	5/12/98

- non ci sono due ennuple con lo stesso valore sull'attributo Matricola
- Matricola è la **chiave** della relazione

Vincolo di chiave

- Se un insieme K è chiave,
 - imponiamo che non ci siano due ennuple uguali fra loro su K

Più chiavi

Matricola	Cognome	Nome	Corso	CF
27655	Rossi	Mario	Ing Inf	RSSMRI98...
78763	Rossi	Mario	Ing Inf	RSSMRI99...
65432	Neri	Piero	Ing Mecc	NRIPTR95...
87654	Neri	Mario	Ing Inf	NRIMRI99...
67653	Rossi	Piero	Ing Mecc	RSSPRI99...

studenti

Matricola	Cognome	Nome	Data di nascita
6554	Rossi	Mario	05/12/1998
8765	Neri	Paolo	03/11/1996
9283	Verdi	Luisa	12/11/1999
3456	Rossi	Maria	01/02/1998

esami

Studente	Voto	Corso
3456	30	04
3456	30	02
9283	26	01
6554	26	01

corsi

Codice	Titolo	Docente
01	Analisi	Mario
02	Chimica	Bruni
04	Chimica	Verdi

Chiave su più attributi

esami

Studente	Voto	Corso
3456	30	04
3456	30	02
9283	26	01
6554	26	01

"Minimalità" delle chiavi

- Una chiave di una relazione è un insieme di attributi che, insieme, permettono di identificare le ennuple e **che è minimale rispetto a questa proprietà**
 - Anche Matricola e Cognome identificano le ennuple della relazione studenti, ma non formano una chiave, perché l'insieme non è minimale:
 - Matricola identifica ed è sottoinsieme di Matricola, Cognome
 - Matricola, Cognome formano una **superchiave**
 - Una **chiave** è una **superchiave minimale**

Esistenza delle chiavi

- Ogni relazione è un insieme:
 - non può contenere due ennuple uguali
- Ogni relazione ha come superchiave l'insieme degli attributi su cui è definita
- e quindi ha (almeno) una chiave

Importanza delle chiavi

- L'esistenza delle chiavi garantisce l'accessibilità a ciascun dato della base di dati
- le chiavi permettono di correlare i dati in relazioni diverse:
 - il modello relazionale è basato su valori

Chiavi e valori nulli

- In presenza di valori nulli, i valori della chiave non permettono
 - di identificare le ennuple
 - di realizzare facilmente i riferimenti da altre relazioni

Matricola	Cognome	Nome	Corso	CF
<i>NULL</i>	Rossi	Mario	Ing Inf	RSSMRI93...
78763	Rossi	Mario	Ing Inf	<i>NULL</i>
65432	Neri	<i>NULL</i>	Ing Mecc	NRIPTR95...
<i>NULL</i>	Neri	Mario	Ing Inf	NRIMRI99...
67653	Rossi	Piero	Ing Mecc	RSSPRI98...

- La presenza di valori nulli, soprattutto nelle chiavi, deve essere limitata o almeno controllata

Chiave primaria

- Chiave su cui non sono ammessi nulli
- Notazione: sottolineatura

<u>Matricola</u>	Cognome	Nome	Corso	CF
27655	Rossi	Mario	Ing Inf	RSSMRI93...
78763	Rossi	Mario	Ing Inf	<i>NULL</i>
65432	Neri	<i>NULL</i>	Ing Mecc	NRIPTR95...
87654	Neri	Mario	Ing Inf	NRIMRI99...
67653	Rossi	Piero	Ing Mecc	RSSPRI98...

Notazione SQL

<u>Matricola</u>	Cognome	Nome	Corso	CF
27655	Rossi	Mario	Ing Inf	RSSMRI93...
78763	Rossi	Mario	Ing Inf	<i>NULL</i>
65432	Neri	<i>NULL</i>	Ing Mecc	NRIPTR95...
87654	Neri	Mario	Ing Inf	NRIMRI99...
67653	Rossi	Piero	Ing Mecc	RSSPRI98...

```
CREATE TABLE Studenti (  
  Matricola INTEGER PRIMARY KEY,  
  Cognome TEXT,  
  Nome TEXT,  
  Corso TEXT,  
  CF TEXT UNIQUE);
```

studenti

<u>Matricola</u>	Cognome	Nome	Data di nascita
6554	Rossi	Mario	05/12/1998
8765	Neri	Paolo	03/11/1996
9283	Verdi	Luisa	12/11/1999
3456	Rossi	Maria	01/02/1998

esami

<u>Studente</u>	Voto	<u>Corso</u>
3456	30	04
3456	30	02
9283	26	01
6554	26	01

corsi

<u>Codice</u>	Titolo	Docente
01	Analisi	Mario
02	Chimica	Bruni
04	Chimica	Verdi

esami

<u>Studente</u>	Voto	<u>Corso</u>
3456	30	04
3456	30	02
9283	26	01
6554	26	01

```
CREATE TABLE esami (  
  Studente INTEGER,  
  Voto INTEGER,  
  Corso INTEGER,  
  PRIMARY KEY (Studente, Corso));
```

Tipi di vincoli

- vincoli intrarelazionali
 - vincoli su valori (o di dominio)
 - vincoli di enunzia
 - vincoli di chiave
- vincoli interrelazionali
 - vincoli di **integrità referenziale**

Integrità referenziale

Infrazioni

<u>Codice</u>	Data	Vigile	Stato	Targa
34321	1/2/2018	3987	I	AA954KK
53524	4/3/2020	3295	F	BB395BB
64521	5/4/2020	3295	E	BB395BB
73321	5/2/2020	9345	E	BB395BB

Vigili

<u>Matricola</u>	Cognome	Nome
3987	Rossi	Luca
3295	Neri	Piero
9345	Neri	Mario
7543	Mori	Gino

Infrazioni

<u>Codice</u>	Data	Vigile	Stato	Targa
34321	1/2/2018	3987	I	AA954KK
53524	4/3/2020	3295	F	BB395BB
64521	5/4/2020	3295	E	BB395BB
73321	5/2/2020	9345	E	BB395BB

Auto

<u>Stato</u>	<u>Targa</u>	Cognome	Nome
I	AA954KK	Rossi	Mario
F	BB395BB	Rossi	Mario
E	BB395BB	Neri	Luca

Vincolo di integrità referenziale

- Vincolo di **integrità referenziale** (“**foreign key**”) fra X di R_1 e R_2
 - impone ai valori su X in R_1 di comparire come valori della chiave primaria di R_2

- vincoli di integrità referenziale fra:
 - l'attributo Vigile della relazione INFRAZIONI e la relazione VIGILI
 - gli attributi Stato e Numero di INFRAZIONI e la relazione AUTO

Violazione di vincolo di integrità referenziale

Infrazioni

<u>Codice</u>	Data	Vigile	Stato	Numero
34321	1/2/2018	3987	I	AA954KK
53524	4/3/2020	3295	F	EE395BB
64521	5/4/2020	3295	E	BB395BB
73321	5/2/2020	9345	E	BB395BB

Auto

<u>Stato</u>	<u>Numero</u>	Cognome	Nome
I	EE395BB	Rossi	Mario
F	FF342BB	Rossi	Mario
E	BB395BB	Neri	Luca

Definizioni in SQL

```
CREATE TABLE Auto (  
  Stato TEXT,  
  Numero INTEGER,  
  PRIMARY KEY (Stato,Numero),  
  Cognome TEXT,  
  Nome TEXT  
);  
CREATE TABLE Vigili (  
  Matricola INTEGER PRIMARY KEY,  
  Cognome TEXT,  
  Nome TEXT  
);  
CREATE TABLE Infrazioni (  
  Codice INTEGER PRIMARY KEY,  
  Data DATE,  
  Vigile INTEGER REFERENCES Vigili(Matricola),  
  Stato TEXT,  
  Numero INTEGER,  
  FOREIGN KEY (Stato, Numero) REFERENCES Auto (Stato, Numero)  
);
```

Integrità referenziale e valori nulli

Impiegati

<u>Matricola</u>	Cognome	Progetto
34321	Rossi	IDEA
53524	Neri	XYZ
64521	Verdi	<i>NULL</i>
73032	Bianchi	IDEA

Progetti

<u>Codice</u>	Inizio	Durata	Costo
IDEA	01/2019	36	200
XYZ	07/2020	24	120
BOH	09/2019	24	150

Vincoli multipli su più attributi

Incidenti

<u>Codice</u>	Data	StatoA	NumeroA	StatoB	NumeroB
34321	1/2/19	I	EE954AB	F	AA954AK
64521	5/4/20	E	BB395AB	I	EE395AB

Auto

<u>Stato</u>	<u>Numero</u>	Cognome	Nome
F	AA954AK	Rossi	Mario
I	EE395AB	Rossi	Mario
E	BB395AB	Neri	Luca

Vincoli multipli su più attributi, 2

- vincoli di integrità referenziale fra:
 - gli attributi StatoA e NumeroA di INCIDENTI e la relazione AUTO
 - gli attributi StatoB e NumeroB di INCIDENTI e la relazione AUTO
- L'ordine degli attributi è significativo

Ordine degli attributi: notazione posizionale

```
CREATE TABLE Auto (  
  Stato TEXT,  
  Numero INTEGER,  
  PRIMARY KEY (Stato,Numero),  
  Cognome TEXT,  
  Nome TEXT  
);  
CREATE TABLE Incidenti (  
  Codice INTEGER PRIMARY KEY,  
  Data DATE,  
  StatoA TEXT,  
  NumeroA INTEGER,  
  FOREIGN KEY (StatoA, NumeroA) REFERENCES Auto (Stato, Numero),  
  StatoB TEXT,  
  NumeroB INTEGER,  
  FOREIGN KEY (StatoB, NumeroB) REFERENCES Auto (Stato, Numero)  
);  
• OK
```

Ordine degli attributi: notazione posizionale

```
CREATE TABLE Auto (  
  Stato TEXT,  
  Numero INTEGER,  
  PRIMARY KEY (Stato,Numero),  
  Cognome TEXT,  
  Nome TEXT  
);  
CREATE TABLE Incidenti (  
  Codice INTEGER PRIMARY KEY,  
  Data DATE,  
  StatoA TEXT,  
  NumeroA INTEGER,  
  FOREIGN KEY (StatoA, NumeroA) REFERENCES Auto (Stato, Numero),  
  StatoB TEXT,  
  NumeroB INTEGER,  
  FOREIGN KEY (StatoB, NumeroB) REFERENCES Auto (Numero, Stato)  
);
```

- **????**