

Basi di dati

2

IL MODELLO RELAZIONALE

Organizzazione dei dati

- Come sono organizzati i dati in una base di dati?

Modello dei dati

- Insieme di costrutti secondo cui i dati di interesse sono organizzati e utilizzati
- Esempio:
 - il **modello relazionale** prevede il costrutto **relazione** (o **tabella**), che permette di definire insiemi di record (righe) omogenei

I modelli logici dei dati

- “Tradizionali”
 - ... relazionale
- Altri più recenti:
 - a oggetti, XML, "NoSQL"

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	24	02
9283	28	01
6554	26	01

corsi

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

Modello "relazionale"

- relazione matematica ... (alla lavagna)

$$D_1 = \{x, y\}$$

$$D_2 = \{1, 2, 3\}$$

DOMINI

PRODOTTO
CARTESIANO

$\{(x, 1),$
 $(x, 2),$
 $(x, 3),$
 $(y, 1),$
 $(y, 2),$
 $(y, 3)\}$

$D_1 \times D_2$

RELAZIONE

$$R \subseteq D_1 \times D_2$$

$\{(x, 1),$
 $(x, 3),$
 $(y, 1)\}$

x	1
x	3
y	1

$D_1 \ D_2 \ D_3$

DOMINI

$D_1 \times D_2 \times D_3$

PRODOTTO CARTESIANO

$R \subseteq D_1 \times D_2 \times D_3$

RELAZIONE

insieme di terne

(d_1, d_2, d_3)

$d_1 \in D_1$

$d_2 \in D_2$

$d_3 \in D_3$

Due osservazioni importanti

- Struttura non posizionale
- Modello basato su valori

Struttura "posizionale" o "non posizionale"

Partite* $\subseteq$ *string* $\times$ *string* $\times$ *int* $\times$ *int

Manchester	Arsenal	3	1
Arsenal	Liverpool	2	0
Manchester	Chelsea	0	2
Chelsea	Liverpool	0	1

Struttura non posizionale

- A ciascun dominio si associa un nome:
attributo (o **colonna**)

Casa	Fuori	RetiCasa	RetiFuori
Manchester	Arsenal	3	1
Arsenal	Liverpool	2	0
Manchester	Chelsea	0	2
Chelsea	Liverpool	0	1

Il modello è basato su valori

- I riferimenti fra dati in relazioni diverse sono rappresentati per mezzo di valori dei domini che compaiono nelle ennuple

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	24	02
9283	28	01
6554	26	01

corsi

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

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
	30	
	24	
	28	
	26	

corsi

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

Basi di dati: schema e istanza

Orario

Insegnamento	Docente	Aula	Ora
Analisi matem. I	Luigi Neri	N1	8:00
Basi di dati	Piero Rossi	N2	10:00
Chimica	Nicola Mori	N1	10:00
Fisica I	Mario Bruni	N1	12:00
Fisica II	Mario Bruni	N3	10:00
Sistemi inform.	Piero Rossi	N3	8:00

Basi di dati: schema e istanza

Lo **schema** della relazione

Orario

Insegnamento	Docente	Aula	Ora
--------------	---------	------	-----

Basi di dati: schema e istanza

Lo **schema** della relazione

Orario

Insegnamento	Docente	Aula	Ora
Analisi matem. I	Luigi Neri	N1	8:00
Basi di dati	Piero Rossi	N2	10:00
Chimica	Nicola Mori	N1	10:00
Fisica I	Mario Bruni	N1	12:00
Fisica II	Mario Bruni	N3	10:00
Sistemi inform.	Piero Rossi	N3	8:00

L'**istanza** della relazione

Definizioni e notazioni (alla lavagna)

- Schema di relazione
 - Ennupla su uno schema di relazione
 - Valore di una ennupla su un attributo
 - Relazione (“istanza di relazione”)
-
- Schema di base di dati
 - (Istanza di) base di dati

Scheme di relazione

R nome della relazione

$A_1, A_2, \dots, A_n$ insieme di attributi

$\downarrow \quad \downarrow \quad \downarrow$
 D_1, D_2, D_n domini

$R(A_1, A_2, \dots, A_n)$

STUDENTI (Matricola, Cognome, ...)

Scheme di base di dati

$\mathcal{R} = \{ R_1(X_1), \dots, R_n(X_n) \}$

UNIVERSITÀ = {
 studenti (Matricola, ...),
 esami (Studento, Voto, Corso),
 corsi (Codice, ...) }

ISTANZA

(istanza di) relazione su $R(A, B, C)$
insieme di esempio (tuple,
righe)
con
un valore per ciascun
attributo

(istanza di) base di dati
insieme di relazioni,
una per ciascuno schema
di relazione nello schema della BD

ennupla (o tupla) t

	A	B	C
$t \rightarrow$	a_1	b_2	c_1

$$t[A] = a_1$$

(oppure
 $t.A = a_1$)

VALORE DELLA
ENNUPLA t
SULL'ATTRIBUTO
 A

- Usiamo $t[A]$ per indicare il valore della ennupla t sull'attributo A

Altra osservazione

- La struttura è rigida e non sempre corrisponde ai dati:
 - "informazione incompleta"

Informazione incompleta

Nome	SecondoNome	Cognome
Franklin	Delano	Roosevelt
Winston		Churchill
Charles		De Gaulle
Josip		Stalin

Informazione incompleta

Nome	Ufficio	Età
Mario	Roma	34
Luca	Milano	
Piero	Firenze	42

Informazione incompleta: soluzioni?

- Potremmo pensare:
 - stringa nulla
 - "0"
 - "99"
- Si può fare?
 - Meglio di no!!

Informazione incompleta nel modello relazionale

- Tecnica semplice ma efficace:
 - **valore nullo** (un valore **diverso** dai valori del dominio, indicato qui con *NULL*)
 - $t[A]$, per ogni attributo A , è
 - un valore del dominio $\text{dom}(A)$
 - oppure il valore nullo *NULL*
- Commenti
 - tecnica semplice
 - attenzione ai troppi valori nulli

Troppi valori nulli

studenti

Matricola	Cognome	Nome	Data di nascita
6554	Rossi	Mario	05/12/1998
9283	Verdi	Luisa	12/11/1999
<i>NULL</i>	Rossi	Maria	01/02/1998

esami

Studente	Voto	Corso
<i>NULL</i>	30	<i>NULL</i>
<i>NULL</i>	24	02
9283	28	01

corsi

Codice	Titolo	Docente
01	Analisi	Mario
02	<i>NULL</i>	<i>NULL</i>
04	Chimica	Verdi

Un'anticipazione (parziale)

- Riassumendo, quanto visto finora:
 - La definizione dello schema di una relazione in SQL (rivedremo più avanti)

```
postgres on paolo@PostgreSQL 11
```

```
create table esami(  
    studente integer not null,  
    corso integer not null,  
    voto integer not null  
)
```

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 dovrete 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 enunzia
 - 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 postgresSQL 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) FROM
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

MariaDB

PostgreSQL

MS SQL

Oracle

Docker

Syntax

Business

```
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

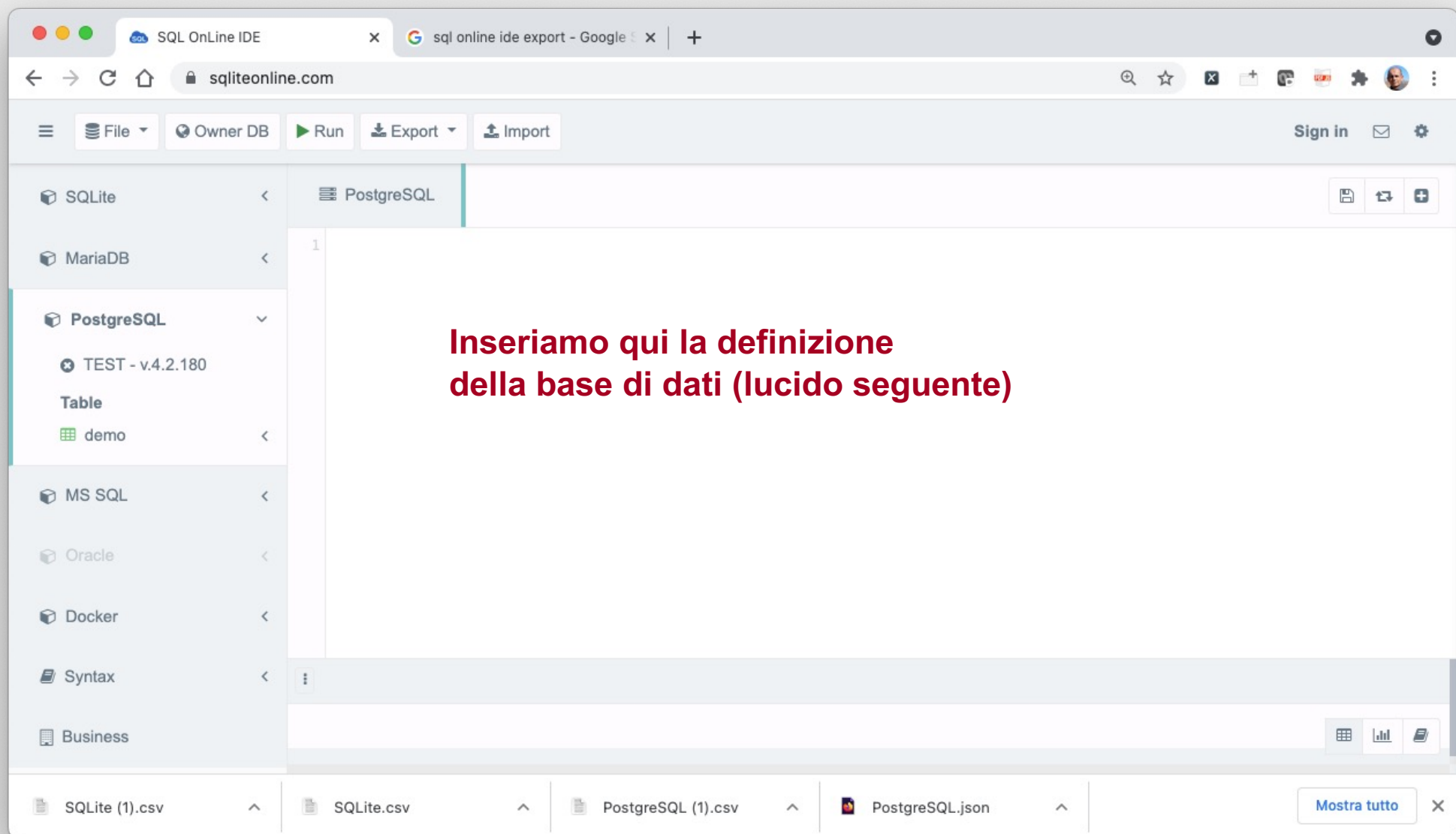
SQL OnLine IDE

sqliteonline.com

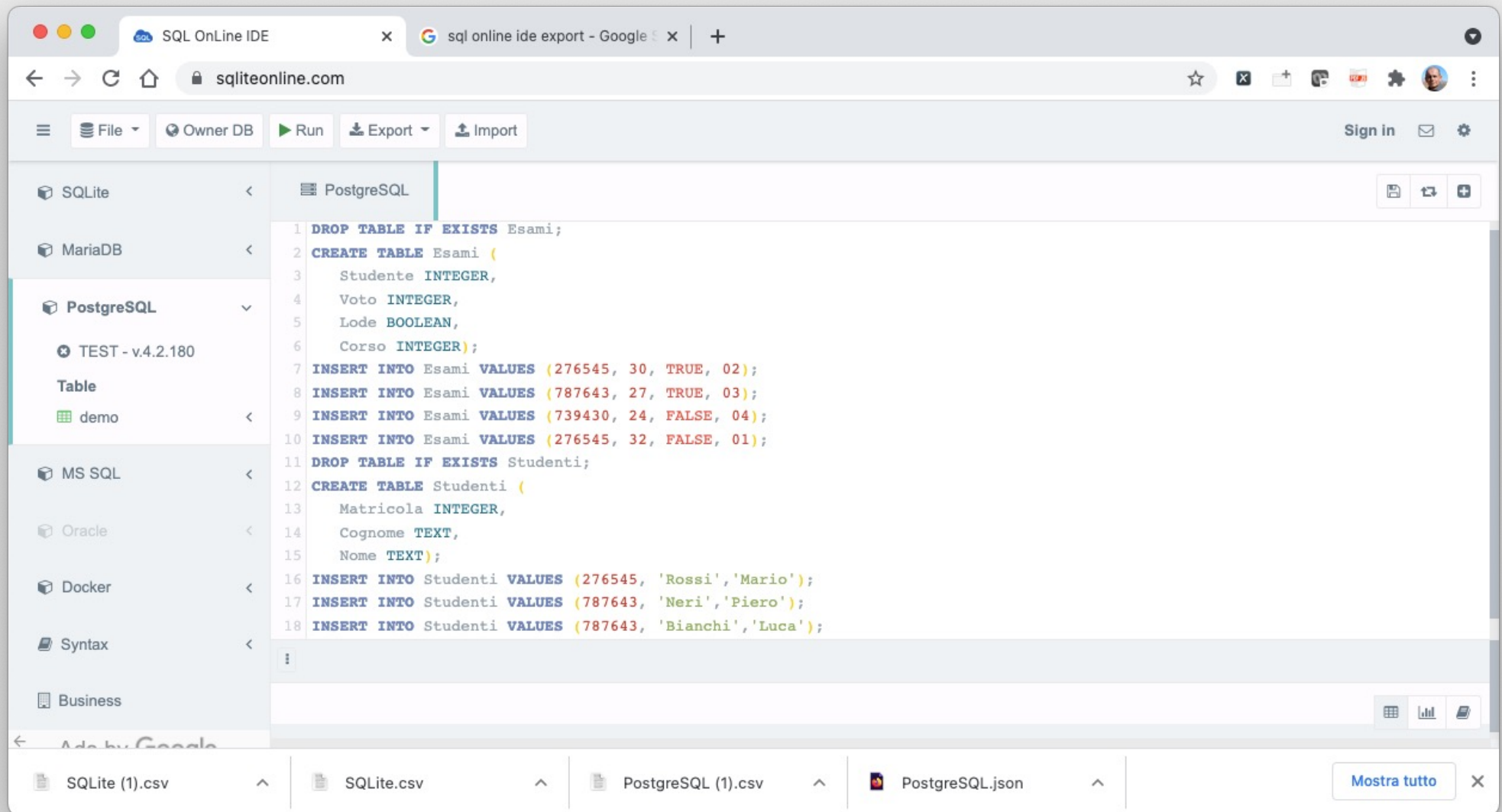
File Owner DB Run Export Import Sign in

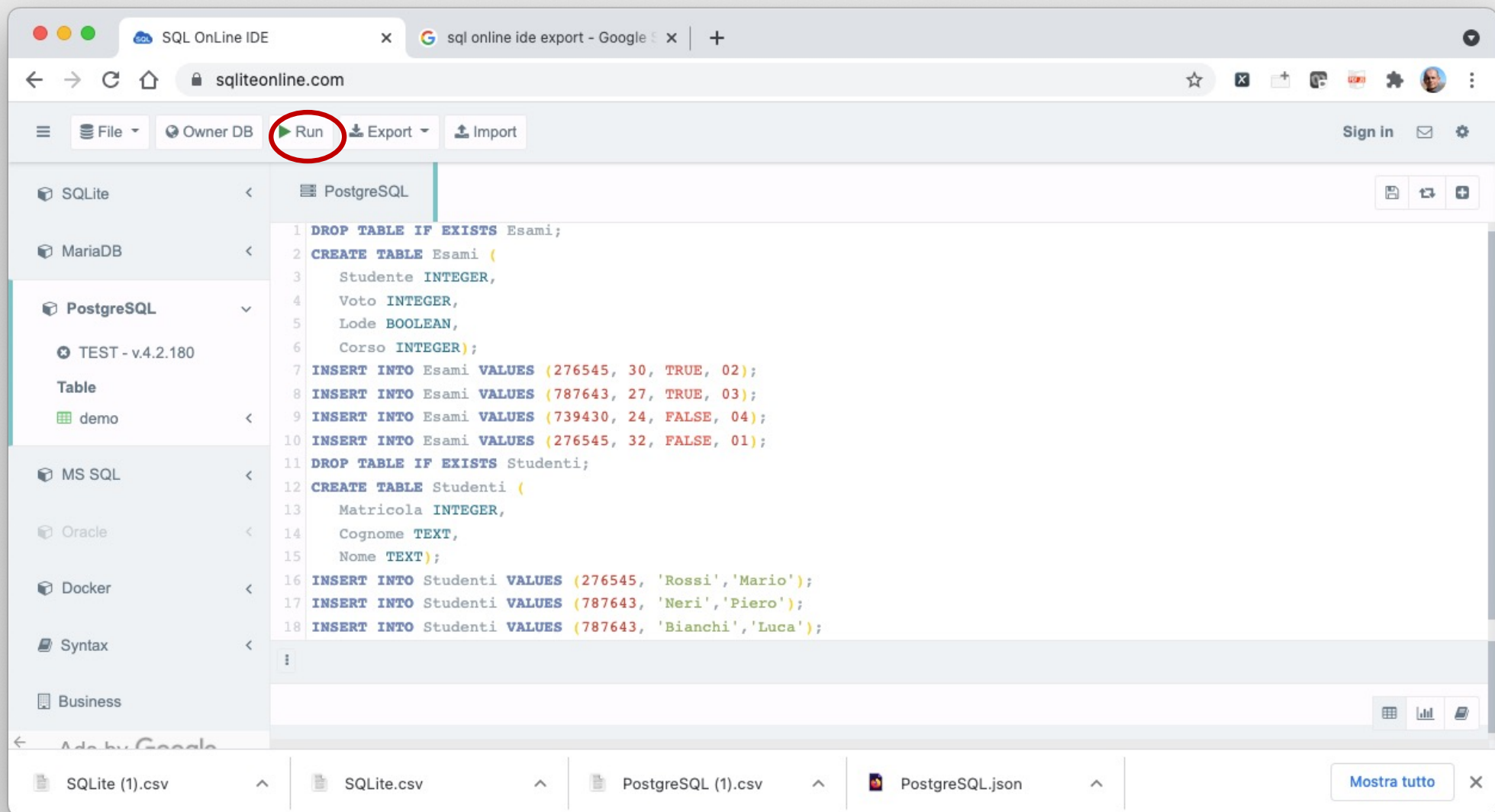
SQLite PostgreSQL MariaDB PostgreSQL Click to connect MS SQL Oracle Docker Syntax Business

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) FROM
5	Short CODE	s* tableName => SELECT * FROM table
6	SQLite 3.36.0	SQL OnLine on JavaScript
7	[RightClick] mouse	Opens many additional



```
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');
```





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

studenti

MS SQL

Oracle

Docker

Syntax

```
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

TEST - v.4.2.180

Table

demo esami

Column

studente integer voto integer lode boolean corso integer studenti

MS SQL Oracle Docker Syntax

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);

esami

SELECT (Show table)

INSERT (Add row)

RENAME

DELETE

DROP

Per vedere il contenuto: tasto destro sul nome della tabella e poi 'SELECT'

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 studenti

MS SQL Oracle Docker Syntax

```
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

03/10/2022

Atzeni e Merialdo - Basi di dati I
Modello relazionale

75

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

Browser tabs: <https://sqliteonline.com> | sql online ide export - Google

Address bar: sqliteonline.com

Navigation: File | Owner DB | Run | Export | Sign in

Database Selection:

- SQLite
- MariaDB
- PostgreSQL**
 - TEST - v.4.2.180
 - Table
 - demo
 - esami
 - studenti
- MS SQL
- Oracle
- Docker
- Syntax
- Business

SQL Editor:

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

PostgreSQL Error Message:

23514 check constraint "IntervalloVoto" of relation "esami" is violated by some row

History: Table | Syntax | History

Downloads:

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

Mostra tutto

Il vincolo funziona

- Se la base di dati non avesse avuto la ennuola 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 SQLiteonline.com interface. The URL bar displays `https://sqliteonline.com`. The page title is "sqliteonline.com".

The interface includes a sidebar on the left with a menu for database types: SQLite, MariaDB, PostgreSQL (selected), MS SQL, Oracle, Docker, Syntax, and Business. Under PostgreSQL, there is a section for "TEST - v.4.2.180" with a "Table" list containing "demo", "esami", and "studenti".

The main editor area shows SQL code for PostgreSQL:

```
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 bottom section of the interface is labeled "History" and contains a table with columns "Table", "Syntax", and "History". The "Table" column shows "PostgreSQL" and the "Syntax" column shows the SQL code for the `ALTER TABLE` statement.

At the bottom of the browser window, there are tabs for "SQLite (1).csv", "SQLite.csv", "PostgreSQL (1).csv", and "PostgreSQL.json". A "Mostra tutto" button is visible on the right.

Large red text "OK" is overlaid on the right side of the image.

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);
```

sqliteonline.com

File Owner DB Run Export Import Sign in

SQLite PostgreSQL

MariaDB

PostgreSQL

TEST - v.4.2.180

Table

demo

esami

studenti

MS SQL

Oracle

Docker

Syntax

```
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
```

History

Table | Syntax | History

PostgreSQL

```
DROP TABLE IF EXISTS Esami;  
CREATE TABLE Esami (  
    Studente INTEGER,
```

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

OK

https://sqliteonline.com x sql online ide export - Google x +

sqliteonline.com

File Owner DB Run Export Sign in

SQLite PostgreSQL

MariaDB

PostgreSQL

TEST - v.4.2.180

Table

demo esami studenti

MS SQL Oracle Docker Syntax

History

Table | Syntax | History

PostgreSQL

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

23514 new row for relation "esami" violates check constraint "intervallovoto"

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

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 ennuola, 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

$Ritenute \geq 0$ 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

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