

Ripasso su SQL

Mattia Mascarello

Introduzione

Sql è un linguaggio utilizzato per comunicare con un server-database e fornirgli istruzioni per la manipolazione e l'estrapolazione di dati del database stesso

Ecco una breve guida al linguaggio SQL

Condizioni

```
campo="valore"
```

```
campo=1
```

Operatori logici

```
campo > 1 AND campo <10
```

```
campo > 1 OR campo2 = "OK"
```

```
eta > 18 AND NOT precedentiPenali = TRUE
```

Like

il "%" in like indica un numero indefinito di caratteri prima della stringa che lo segue o dopo la stringa che lo precede

Ex.

campo 1 in questo caso è "riconoscimento facciale"

per ottenere un valore true nella condizione:

```
campo1 LIKE "%faccia%"  
campo1 LIKE "riconosci%"
```

Clausole

Create

```
CREATE TABLE "{Nome Tabella}" (  
    "{Campo Id}"    INTEGER /*INTERO*/ NOT NULL /* NON NULLO*/ PRIMARY KEY  
    /*CHIAVE PRIMARIA (Denota ogni riga in modo univoco)*/AUTOINCREMENT /*INCREMENTA  
AUTOMATICAMENTE DI 1 OGNI VOLTA*/ UNIQUE /*L'ID È UNICO, PERCHÈ DEVE IDENTIFICARE  
OGNI RIGA IN MODO UNIVOCO*/,  
    "{campo1}"    TEXT NOT NULL,  
    "{campo2}"    INTEGER NOT NULL,  
    "{campo3}"    DECIMAL,/* Questo campo può essere nullo */  
    "eta"    INTEGER NOT NULL CHECK("eta" > 0 AND "eta" < 100), /* La clausola  
CHECK({CONDIZIONE}) impone un VINCOLO DI INTEGRITÀ per impedire l'inserimento di  
valori che non rispettano la condizione */  
    "classId"    INTEGER NOT NULL,  
    FOREIGN KEY("classId") REFERENCES "Classes"("id") /* Vincolo di integrità:  
per inserire una riga il valore del campo "classId" deve essere presente nel  
campo "id" in almeno una riga di "Classes"*/  
);
```

Select

```
SELECT {elenco dei campi | *} FROM {Tabella} WHERE {CONDIZIONE}
```

```
SELECT {Tabella.campo, Tabella2.campo2 | *} FROM {Tabella} JOIN Tabella2 ON  
Tabella2.campo2 = Tabella.campo {, Tabella.campoA=Tabella2.campoB} WHERE  
{CONDIZIONE}
```

```
SELECT {Tabella.campo, Tabella2.campo2 | *} WHERE {CONDIZIONE} LIMIT {numero di  
righe da restituire} OFFSET {righe da saltare nel risultato}
```

Insert

```
INSERT INTO {Tabella}({campo1, campo3}) VALUES ({valore1, valore3})
```

```
INSERT INTO {Tabella} VALUES ({valore1, valore2, valore3})
```

Update

```
UPDATE {TABELLA} SET {campo}={valore}, {campo2}={valore2} WHERE {CONDIZIONE}
```



Delete

```
DELETE FROM {TABELLA} WHERE {CONDIZIONE}
```

Esercizio

Segui i passaggi inserendo le query nel pannello *Esegui SQL* e verifica il funzionamento delle query

```
CREATE TABLE "Classes" (  
  "classId" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,  
  "className" TEXT NOT NULL  
)
```

Classes	
 classId	INTEGER
 className	TEXT

```
CREATE TABLE "Students" (  
  "studentId" INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT UNIQUE,  
  "name" TEXT NOT NULL,  
  "surname" TEXT NOT NULL,  
  "notes" TEXT,  
  "eta" INTEGER NOT NULL CHECK("eta" > 0 AND "eta" < 100),  
  "classId" INTEGER NOT NULL,  
  FOREIGN KEY("classId") REFERENCES "Classes"("classId")  
);
```

Students	
 studentId	INTEGER
 name	TEXT
 surname	TEXT
 notes	TEXT
 eta	INTEGER
 classId	INTEGER

```
INSERT INTO Classes(className) VALUES ("4E");  
INSERT INTO Classes(className) VALUES ("4D");
```

		classId	className
	Filtro	Filtro	
1	1		4E
2	2		4D

```
INSERT INTO Students(name,surname,eta,classId) VALUES  
("Mattia","Mascarello",-1,1)
```

Risultato: CHECK constraint failed: Students

Alla riga 1:

```
INSERT INTO Students(name,surname,eta,classId) VALUES ("Mattia","Mascarello",-1,1)
```

```
INSERT INTO Students(name,surname,eta,classId) VALUES
("Mattia","Mascarello",18,1);
INSERT INTO Students(name,surname,notes,eta,classId) VALUES
("Matilde","Berzia","A friend Of Mine",17,2);
```

Risultato: query eseguita con successo. Impiegati 0ms, 2 righe modificate

Alla riga 1:

```
INSERT INTO Students(name,surname,eta,classId) VALUES ("Mattia","Mascarello",18,1)
```

Alla riga 2:

```
INSERT INTO Students(name,surname,notes,eta,classId) VALUES ("Matilde","Berzia","A friend
Of Mine",17,2);
```

	studentId	name	surname	notes	eta	classId
	Filtro	Filtro	Filtro	Filtro	Filtro	Filtro
1	1	Mattia	Mascarello	NULL	18	1
2	2	Matilde	Berzia	A friend Of ...	17	2

```
SELECT surname,eta FROM Students WHERE eta<18
```

	surname	eta
1	Berzia	17

```
SELECT Students.surname, Students.eta, Classes.className FROM Students JOIN
Classes ON Students.classId=Classes.classId
```

	surname	eta	className
1	Mascarello	18	4E
2	Berzia	17	4D

```
SELECT Students.surname, Students.eta, Classes.className FROM Students JOIN
Classes ON Students.classId=Classes.classId WHERE eta>17
```

	surname	eta	className
1	Mascarello	18	4E

```
SELECT Students.name, Students.notes, Classes.className FROM Students JOIN
Classes ON Students.classId=Classes.classId WHERE Classes.className="4D"
```

	name	notes	className
1	Matilde	A friend Of Mine	4D

```
SELECT * FROM Students
```

	studentId	name	surname	notes	eta	classId
1	1	Mattia	Mascarello	NULL	18	1
2	2	Matilde	Berzia	A friend Of Mine	17	2

```
SELECT * FROM Students LIMIT 1
```

	studentId	name	surname	notes	eta	classId
1	1	Mattia	Mascarello	NULL	18	1

```
SELECT * FROM Students LIMIT 1 OFFSET 1
```

	studentId	name	surname	notes	eta	classId
1	2	Matilde	Berzia	A friend Of Mine	17	2

```
UPDATE Students SET name="Giorgia" WHERE studentId=2
```

Risultato: query eseguita con successo. Impiegati 0ms, 1 righe modificate

Alla riga 1:

```
UPDATE Students SET name="Giorgia" WHERE studentId=2
```

	studentId	name	surname	notes	eta	classId
	Filtro	Filtro	Filtro	Filtro	Filtro	Filtro
1	1	Mattia	Mascarello	NULL	18	1
2	2	Giorgia	Berzia	A friend Of ...	17	2

```
INSERT INTO Students(name,surname,notes,eta,classId) VALUES ("Giovanni","Rana","YES",22,2);
```

	studentId	name	surname	notes	eta	classId
	Filtro	Filtro	Filtro	Filtro	Filtro	Filtro
1	1	Mattia	Mascarello	NULL	18	1
2	2	Giorgia	Berzia	A friend Of ...	17	2
3	3	Giovanni	Rana	Y E S	22	2

```
SELECT * FROM Students WHERE name LIKE "Gio%"
```

	studentId	name	surname	notes	eta	classId
1	2	Giorgia	Berzia	A friend Of Mine	17	2
2	3	Giovanni	Rana	Y E S	22	2

```
SELECT * FROM Students
```

	studentId	name	surname	notes	eta	classId
1	1	Mattia	Mascarello	NULL	18	1
2	2	Giorgia	Berzia	A friend Of Mine	17	2
3	3	Giovanni	Rana	Y E S	22	2

```
DELETE FROM Students WHERE studentId=3
```

Risultato: query eseguita con successo. Impiegati 0ms, 1 righe modificate

Alla riga 1:

```
DELETE FROM Students WHERE studentId=3
```

	studentId	name	surname	notes	eta	classId
1	1	Mattia	Mascarello	NULL	18	1
2	2	Giorgia	Berzia	A friend Of Mine	17	2