11 - Constraints Campi in SQL

I vincoli sui campi (`constraints`) sono un componente essenziale della progettazione di database in SQL. Questi vincoli definiscono regole e condizioni che i dati devono soddisfare per garantire l'integrità e la coerenza del database. In questa lezione, esploreremo in dettaglio i diversi tipi di vincoli sui campi, il motivo per cui sono utilizzati, come applicarli e forniremo una lista di vincoli comuni insieme a esempi pratici.

Importanza dei Vincoli sui Campi

I vincoli sui campi sono fondamentali per garantire che i dati immagazzinati nel database siano accurati e coerenti. Questi vincoli stabiliscono regole che limitano quali valori possono essere inseriti o modificati nelle colonne di una tabella. Ciò contribuisce a prevenire errori, anomalie e dati non validi nel database.

Tipi di Vincoli sui Campi

Ecco alcuni dei vincoli più comuni sui campi in SQL:

- **NOT NULL:** Impedisce l'inserimento di valori NULL nella colonna.
- **UNIQUE:** Garantisce che tutti i valori nella colonna siano univoci.
- **PRIMARY KEY:** Identifica univocamente ogni riga in una tabella.
- **FOREIGN KEY:** Crea una relazione tra tabelle e impone coerenza referenziale.
- **CHECK:** Definisce una condizione che i valori devono soddisfare.
- **DEFAULT:** Imposta un valore predefinito quando non ne viene specificato uno.

Vincolo NOT NULL

Il vincolo NOT NULL impedisce l'inserimento di valori NULL nella colonna. In altre parole, richiede che ogni valore nella colonna debba essere non nullo.

Questo vincolo è utile quando si desidera garantire che un campo obbligatorio non rimanga vuoto. Viene utilizzato per mantenere la coerenza e l'integrità dei dati.

Sintassi

```
CREATE TABLE studenti (
    id INT NOT NULL,
    nome VARCHAR(50) NOT NULL
);
```

Esempio

```
INSERT INTO studenti (id, nome) VALUES (1, 'Mario'); -- OK
INSERT INTO studenti (id, nome) VALUES (2, NULL); -- Errore
```

Vincolo UNIQUE

Il vincolo UNIQUE garantisce che tutti i valori nella colonna siano univoci, impedendo la duplicazione di valori.

Questo vincolo è utile quando si desidera evitare l'inserimento di dati duplicati in una colonna chiave o in una colonna che dovrebbe contenere solo valori univoci.

Sintassi

```
CREATE TABLE dipendenti (
    codice_fiscale VARCHAR(16) UNIQUE,
    nome VARCHAR(50)
);
```

Esempio

```
INSERT INTO dipendenti (codice_fiscale, nome) VALUES ('ABCD1234',
'Alice'); -- OK
INSERT INTO dipendenti (codice_fiscale, nome) VALUES ('ABCD1234', 'Bob');
-- Errore
```

Vincolo PRIMARY KEY

Il vincolo PRIMARY KEY identifica univocamente ogni riga in una tabella. Combina il vincolo NOT NULL e il vincolo UNIQUE.

Questo vincolo è utilizzato per identificare in modo univoco le righe in una tabella, spesso in colonne che fungono da chiavi primarie.

Sintassi

```
CREATE TABLE prodotti (
    codice_prodotto INT PRIMARY KEY,
    nome VARCHAR(100)
);
```

Esempio

```
INSERT INTO prodotti (codice_prodotto, nome) VALUES (1, 'Prodotto A'); --
OK
INSERT INTO prodotti (codice_prodotto, nome) VALUES (1, 'Prodotto B'); --
Errore
```

Vincolo FOREIGN KEY

Il vincolo FOREIGN KEY crea una relazione tra tabelle, stabilendo coerenza referenziale tra le colonne.

Questo vincolo è usato per creare relazioni tra tabelle e garantire che i valori nelle colonne correlate siano coerenti.

Sintassi

```
CREATE TABLE ordini (
    id INT PRIMARY KEY,
    codice_cliente INT,
    FOREIGN KEY (codice_cliente) REFERENCES clienti(id)
);
```

Esempio

```
INSERT INTO ordini (id, codice_cliente) VALUES (1, 101); -- OK
INSERT INTO ordini (id, codice_cliente) VALUES (2, 201); -- Errore (nessun
cliente con id 201)
```

Vincolo CHECK

Il vincolo CHECK definisce una condizione che i valori devono soddisfare.

Questo vincolo è usato per imporre regole personalizzate sui valori delle colonne.

Sintassi

```
CREATE TABLE dipendenti (
    eta INT CHECK (eta >= 18),
    nome VARCHAR(50)
);
```

Esempio

```
INSERT INTO dipendenti (eta, nome) VALUES (25, 'Alice'); -- OK
INSERT INTO dipendenti (eta, nome) VALUES (16, 'Bob'); -- Errore
```

Vincolo DEFAULT

Il vincolo DEFAULT imposta un valore predefinito quando non ne viene specificato uno.

Questo vincolo è utilizzato per fornire un valore di default quando un valore non è esplicitamente specificato durante l'inserimento dei

dati.

Sintassi

```
CREATE TABLE clienti (
    id INT PRIMARY KEY,
    nome VARCHAR(50),
    stato VARCHAR(2) DEFAULT 'US'
);
```

Esempio

```
INSERT INTO clienti (id, nome) VALUES (1, 'Alice'); -- OK (stato sarà 'US'
per default)
INSERT INTO clienti (id, nome, stato) VALUES (2, 'Bob', 'CA'); -- OK
(stato sarà 'CA')
```

Vincoli e Integrità dei Dati

I vincoli sui campi contribuiscono a mantenere l'integrità dei dati nel database. Impedendo l'inserimento di valori non validi o duplicati, si preservano l'affidabilità e la precisione dei dati.

Conclusioni

I vincoli sui campi in SQL sono strumenti potenti per garantire l’integrità e la coerenza dei dati nel database. La loro applicazione accurata aiuta a prevenire errori e a mantenere i dati in uno stato coerente. Ogni tipo di vincolo ha uno scopo specifico e può essere utilizzato per risolvere diverse esigenze di progettazione.