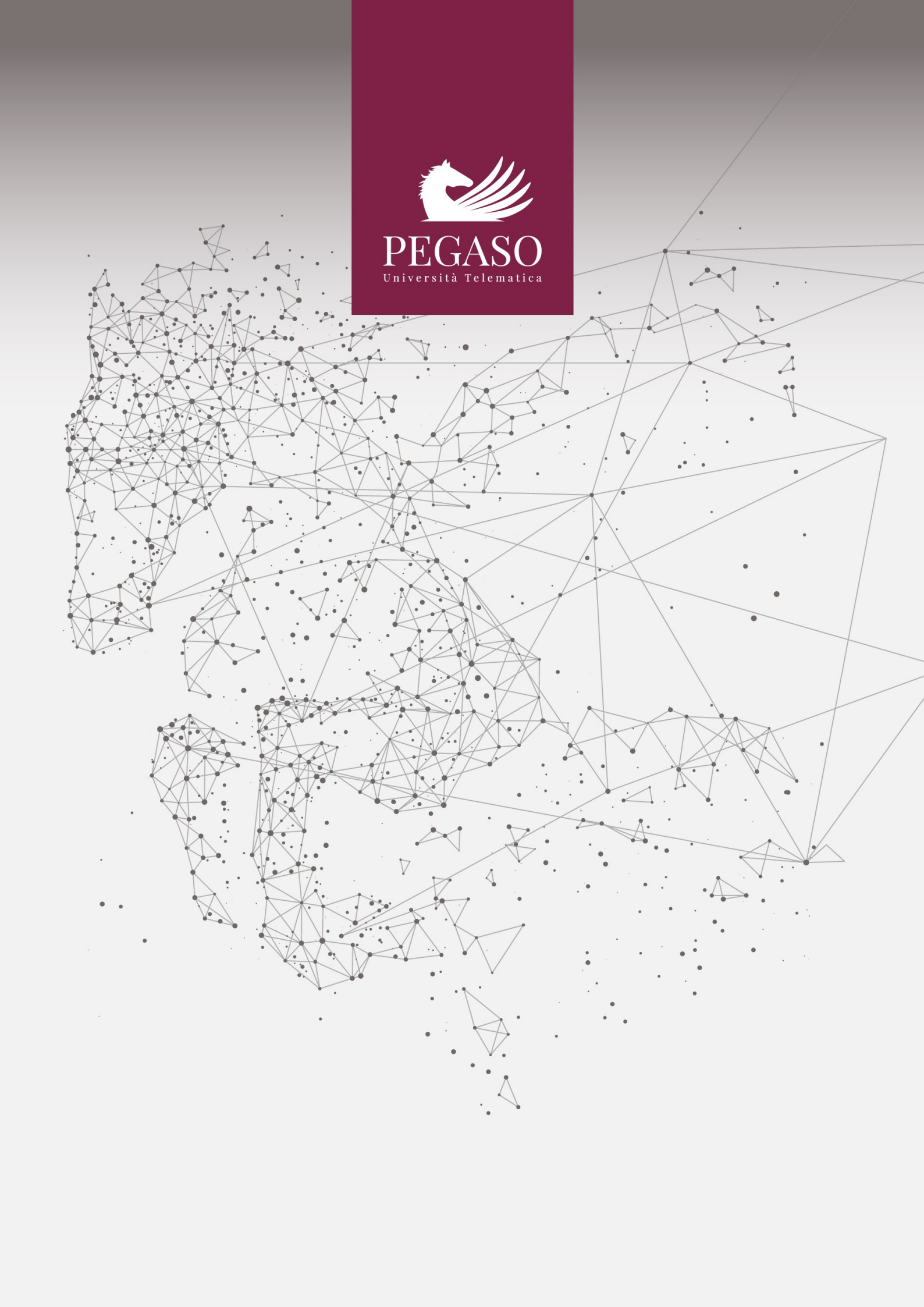

PEGASO
Università Telematica

Attenzione! Questo materiale didattico è per uso personale dello studente ed è coperto da copyright. Ne è severamente vietata la riproduzione o il riutilizzo anche parziale, ai sensi e per gli effetti della legge sul diritto d'autore (L. 22.04.1941/n. 633).

Indice

PEGASO
Università Telematica

Attenzione! Questo materiale didattico è per uso personale dello studente ed è coperto da copyright. Ne è severamente vietata la riproduzione o il riutilizzo anche parziale, ai sensi e per gli effetti della legge sul diritto d'autore (L. 22.04.1941/n. 633).

1. Introduzione

In questa unità didattica, si affronta principalmente lo studio della istruzione SQL base SELECT, ovvero di quella istruzione avente il compito importantissimo di reperire dati da un database per renderli disponibile all'utente finale. Pertanto vengono illustrati esempi applicativi e varianti base. Inoltre vengono approfonditi alcuni aspetti dei vincoli possibili nella creazione di una tabella e del modo con il quale aggiungere o togliere righe. Introduciamo quindi il concetto di query: "In informatica il termine **query** viene utilizzato per indicare l'interrogazione da parte di un utente di un database, strutturato tipicamente secondo il modello relazionale, per compiere determinate operazioni sui dati (selezione, inserimento, cancellazione dati, aggiornamento ecc.). Solitamente una *query* utilizza un linguaggio di interrogazione *interpretato* rappresentato dallo standard SQL (*Structured Query Language*) nei suoi sottolinguaggi Data Query Language e Data Manipulation Language, per renderla più comprensibile al DBMS. L'analisi del risultato della *query* è oggetto di studio dell'algebra relazionale[1]".

[1] https://it.wikipedia.org/wiki/Query

PEGASO
Università Telematica

Attenzione! Questo materiale didattico è per uso personale dello studente ed è coperto da copyright. Ne è severamente vietata la riproduzione o il riutilizzo anche parziale, ai sensi e per gli effetti della legge sul diritto d'autore (L. 22.04.1941/n. 633).

2. Vincoli interrelazionali

I vincoli interrelazionali sono vincoli tra tabelle. Di seguito ne vediamo i più importanti.

2.1 Vincolo CHECK

Il vincolo CHECK viene utilizzato per limitare l'intervallo di valori che può essere inserito in una colonna. Se si definisce un vincolo CHECK su una singola colonna, esso consente solo determinati valori per questa colonna. Se si definisce un vincolo CHECK su una tabella, è possibile limitare i valori a determinate colonne in base ai valori di altre colonne sulla riga.

Esempio:

Il seguente SQL crea un vincolo CHECK sulla colonna "Età" quando viene creata la tabella "Persone". Il vincolo CHECK garantisce che non è possibile avere una persona di età inferiore ai 18 anni. L'istruzione è rappresentata in Figura 1.

```
CREATE TABLE Persons (
    ID int NOT NULL,
    LastName varchar(255) NOT NULL,
    FirstName varchar(255),
    Age int,
    CHECK (Age>=18)
);
```

Figura 1: creazione di una tabella con il vincolo CHECK.

2.2 Vincoli REFERENCES e FOREIGN KEY

Sono due vincoli che permettono di definire vincoli di integrità referenziale. Si hanno di nuovo due sintassi:

- per singoli attributi;
- su più attributi.

PEGASO
Università Telematica

Attenzione! Questo materiale didattico è per uso personale dello studente ed è coperto da copyright. Ne è severamente vietata la riproduzione o il riutilizzo anche parziale, ai sensi e per gli effetti della legge sul diritto d'autore (L. 22.04.1941/n. 633).

È inoltre possibile definire politiche di reazione alla violazione di tali vincoli.

Esempio 1

Il codice SQL illustrato in Figura 2, crea una FOREIGN KEY sulla colonna "PersonID" della relazione Persons, quando viene creata la tabella "Orders".

```
CREATE TABLE Orders (
    OrderID int NOT NULL,
    OrderNumber int NOT NULL,
    PersonID int,
    PRIMARY KEY (OrderID),
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)
);
```

Figura 2: esempio di utilizzo dei vincoli Foreign Key e References.

Tabelle 1: Tabella Orders creata con l'istruzione CREATE TABLE e con una foreign key.

OrderID	**OrderNumber**	**PersonID**
A123	1234	56
B234	456	77
…	…	..

Tabelle 2 relazione Persons.

PersonID	**Age**	**Name**
56	34	Filippo
77	45	Giuseppe
……	…..	…..

PEGASO
Università Telematica

Attenzione! Questo materiale didattico è per uso personale dello studente ed è coperto da copyright. Ne è severamente vietata la riproduzione o il riutilizzo anche parziale, ai sensi e per gli effetti della legge sul diritto d'autore (L. 22.04.1941/n. 633).

Esempio 2

Dato il database composto dalle tabelle Infrazioni, Vigili e Auto, rappresentate in Figura 3, scrivere l'istruzione SQL che colleghi la tabella Infrazioni alle tabelle Vigili e Auto attraverso vincoli di integrità referenziale in modo tale che, per ogni infrazione, si possa sapere il vigile che l'abbia segnalata ed il proprietario della macchina.

Infrazioni

Codice	Data	Vigile	Prov	Numero
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	3295	PR	839548
73321	5/2/98	9345	PR	839548

Vigili

Matricola	Cognome	Nome
3987	Rossi	Luca
3295	Neri	Piero
9345	Neri	Mario
7543	Mori	Gino

Auto

Prov	Numero	Cognome	Nome
MI	39548K	Rossi	Mario
TO	E39548	Rossi	Mario
PR	839548	Neri	Luca

Figura 3: database di esempio.

Dalla Figura 4 si vede che vogliamo avere dei vincoli di integrità referenziale tra le varie tabelle di partenza: Infrazioni (Vigile) con Vigili (Matricola) e Infrazioni (Prov, Numero) con Auto (Prov, Numero). Ciò ci consente infatti di recuperare per ciascuna infrazione, tutte le informazioni necessarie.

Infrazioni

Codice	Data	Vigile	Prov	Numero
34321	1/2/95	3987	MI	39548K
53524	4/3/95	3295	TO	E39548
64521	5/4/96	3295	PR	839548
73321	5/2/98	9345	PR	839548

Vigili

Matricola	Cognome	Nome
3987	Rossi	Luca
3295	Neri	Piero
9345	Neri	Mario
7543	Mori	Gino

Auto

Prov	Numero	Cognome	Nome
MI	39548K	Rossi	Mario
TO	E39548	Rossi	Mario
PR	839548	Neri	Luca

Figura 4: lo schema di database con i vincoli di integrità referenziale segnalati con blocchi dello stesso colore.

PEGASO
Università Telematica

Attenzione! Questo materiale didattico è per uso personale dello studente ed è coperto da copyright. Ne è severamente vietata la riproduzione o il riutilizzo anche parziale, ai sensi e per gli effetti della legge sul diritto d'autore (L. 22.04.1941/n. 633).

L'istruzione per creare in modo corretto la tabella Infrazioni è illustrata in Figura 5.

```
CREATE TABLE Infrazioni(
        Codice CHAR(6) NOT NULL PRIMARY KEY,
        Data DATE NOT NULL,
        Vigile INTEGER NOT NULL
                REFERENCES Vigili(Matricola),
        Provincia CHAR(2),
        Numero CHAR(6) ,
        FOREIGN KEY(Provincia, Numero)
                REFERENCES Auto(Provincia, Numero)
)
```

Figura 5: l'istruzione che consente di inserire i vincoli di integrità.

Attenzione! Questo materiale didattico è per uso personale dello studente ed è coperto da copyright. Ne è severamente vietata la riproduzione o il riutilizzo anche parziale, ai sensi e per gli effetti della legge sul diritto d'autore (L. 22.04.1941/n. 633).

3. Operazioni sui dati

In questo paragrafo si illustrano le caratteristiche base dell'istruzione SELECT, istruzione che consente di filtrare dati da tabelle. In generale comunque, le istruzioni che operano sui dati sono suddivise, in prima istanza, nel modo seguente:

- Operazioni di interrogazione:
  - SELECT
- Operazioni di modifica:
  - INSERT, DELETE, UPDATE

3.1 Istruzione SELECT (versione base)

L'istruzione SELECT ha la sintassi illustratata in Figura 6.

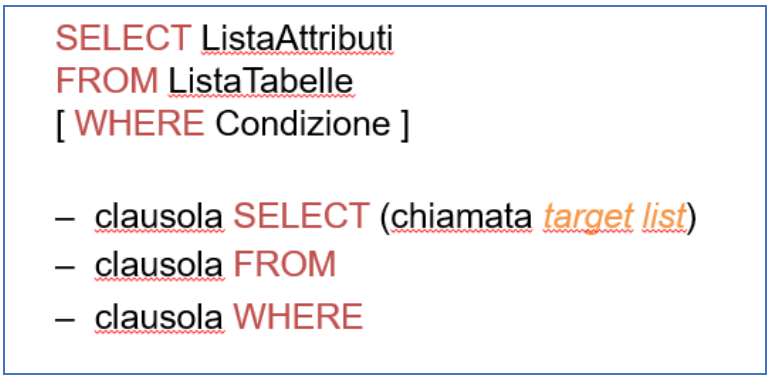


Figura 6: l'istruzione SELECT.

I dati recuperati sono memorizzati in una tabella chiamata **result-set.**

Esempio.

Data la tabella di Figura 7, Persone (Nome, Età, Reddito), contenente diverse ennuple (record), vogliamo avere il nome ed il reddito delle persone con meno di trenta anni. Utilizzare l'istruzione SELECT.

Attenzione! Questo materiale didattico è per uso personale dello studente ed è coperto da copyright. Ne è severamente vietata la riproduzione o il riutilizzo anche parziale, ai sensi e per gli effetti della legge sul diritto d'autore (L. 22.04.1941/n. 633).

Persone

Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

Figura 7: la tabella Persone.

$$\pi_{Nome,\ Reddito}(\sigma_{Eta<30}(Persone))$$

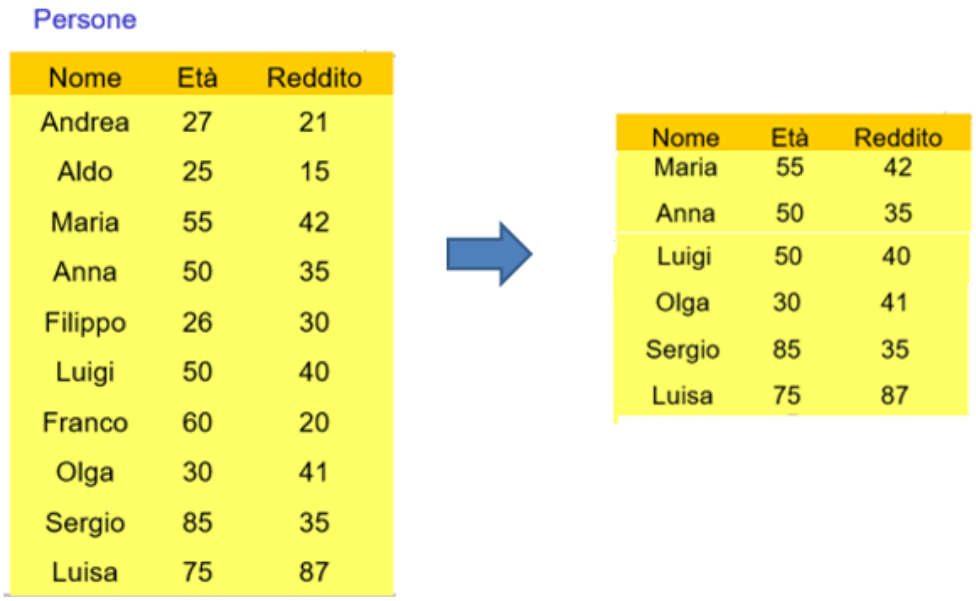
```
select Nome, Reddito
from Persone
where Eta < 30
```

Figura 8: utilizzo dell'istruzione SELECT.

L'istruzione raffigurata in Figura 8, produce un filtraggio dei dati che dà come result-set la tabella illustrata in Figura 9. Da notare la relazione che esiste tra l'operatore PROIEZIONE e l'operatore SELEZIONE dell'algebra relazionale. Dalla Figura 8, si deduce che il risultato viene prodotto attraverso una applicazione dei due operatori: prima SELEZIONE e successivamente PROIEZIONE.

PEGASO
Università Telematica

Attenzione! Questo materiale didattico è per uso personale dello studente ed è coperto da copyright. Ne è severamente vietata la riproduzione o il riutilizzo anche parziale, ai sensi e per gli effetti della legge sul diritto d'autore (L. 22.04.1941/n. 633).



Persone

Nome	Età	Reddito
Andrea	27	21
Aldo	25	15
Maria	55	42
Anna	50	35
Filippo	26	30
Luigi	50	40
Franco	60	20
Olga	30	41
Sergio	85	35
Luisa	75	87

Nome	Età	Reddito
Maria	55	42
Anna	50	35
Luigi	50	40
Olga	30	41
Sergio	85	35
Luisa	75	87

Figura 9: il result-set finale.

Questa istruzione sarà comunque oggetto di approfondimento nella prossima UDA.

Attenzione! Questo materiale didattico è per uso personale dello studente ed è coperto da copyright. Ne è severamente vietata la riproduzione o il riutilizzo anche parziale, ai sensi e per gli effetti della legge sul diritto d'autore (L. 22.04.1941/n. 633).

4. Modifiche degli schemi

SQL fornisce primitive per la manipolazione degli schemi delle basi di dati, che permettono di modificare le definizioni di tabelle precedentemente introdotte. I comandi che vengono utilizzati a questo scopo sono ALTER e DROP. Ad esempio:

- ALTER DOMAIN
- ALTER TABLE
- DROP DOMAIN
- DROP TABLE

Nei prossimi paragrafi studiamo queste due istruzioni.

4.1 ALTER

Il comando ALTER permette di modificare domini e schemi di tabelle. Il comando può assumere varie forme come ALTER domain e ALTER table.

- **ALTER TABLE – ADD COLUMN**
  - Elimina o aggiunge una colonna ad una tabella
  - ALTER TABLE *table_name*
    ADD *column_name datatype*;

Esempio:

  - ALTER TABLE Clienti
    ADD Email varchar (255);
  - Effetto: aggiunge una Colonna.

- **ALTER TABLE - DROP COLUMN**
  - Elimina una colonna da una tabella
  - ALTER TABLE *table_name*
    DROP COLUMN *column_name*;

Esempio:

  - ALTER TABLE Clienti
    DROP COLUMN Email;
  - Effetto: cancella la Colonna Email

PEGASO
Università Telematica

Attenzione! Questo materiale didattico è per uso personale dello studente ed è coperto da copyright. Ne è severamente vietata la riproduzione o il riutilizzo anche parziale, ai sensi e per gli effetti della legge sul diritto d'autore (L. 22.04.1941/n. 633).

Tramite ALTER DOMAIN e ALTER TABLE, è quindi possibile aggiungere e rimuovere vincoli e modificare i valori di default associati ai domini e agli attributi; è inoltre possibile aggiungere ed eliminare attributi e vincoli sullo schema di una tabella. Si noti che quando si definisce un nuovo vincolo, questo deve essere soddisfatto dai dati già presenti: se l'istanza contiene delle violazioni per il nuovo vincolo, l'inserimento viene rifiutato (Atzeni et al., 2018, pag. 105).

4.2 DROP

Il comando DROP permette di rimuovere dei componenti siano essi schemi, domini, tabelle, viste. La sintassi del comando è la seguente:

DROP <schema|domain|table|view|assertion|> *NomeElemento*[restrict|cascade]

L'opzione *restrict* specifica che il comando non deve essere eseguito in presenza di oggetti *non vuoti*: uno schema non è rimosso se contiene tabelle o altri oggetti; ad esempio una tabella non è rimossa se possiede delle righe.

L'opzione *cascade* specifica che tutti gli oggetti specificati devono essere rimossi. Quando si rimuove uno schema non vuoto, anche tutti gli oggetti che ne fanno parte vengono eliminati. Ad esempio l'istruzione:

DROP domain *StringaLunga* cascade

Tramite questo comando, tutti gli attributi definiti su quel dominio assumeranno direttamente il dominio varchar(100), se *StringaLunga* è definito come varchar(100).

Fare attenzione prima di eliminare una tabella. L'eliminazione di una tabella comporta la perdita di tutte le informazioni memorizzate nella tabella!

4.3 Definizione degli indici

Gli indici sono utilizzati per recuperare i dati dal database molto velocemente. Gli utenti non possono vedere gli indici, sono solo utilizzati per velocizzare le ricerche / query. L'aggiornamento di una tabella con indici richiede più tempo dell'aggiornamento di una tabella senza (poiché anche gli indici necessitano di un aggiornamento). Quindi, crea solo indici su colonne su cui si cerca frequentemente la ricerca. Quindi:

- è rilevante dal punto di vista delle prestazioni
- ma è a livello fisico e non logico

PEGASO
Università Telematica

Attenzione! Questo materiale didattico è per uso personale dello studente ed è coperto da copyright. Ne è severamente vietata la riproduzione o il riutilizzo anche parziale, ai sensi e per gli effetti della legge sul diritto d'autore (L. 22.04.1941/n. 633).

- in passato era importante perché in alcuni sistemi era l'unico mezzo per definire chiavi
- istruzione: CREATE INDEX

CREATE INDEX *index_name*

ON *table_name* (*column1*, *column2*, ...);

La sintassi per la creazione degli indici varia tra diversi database. Pertanto: controlla la sintassi per la creazione di indici nel tuo database. Vale la pena sottolineare la differenza tra chiave primaria e indice:

1) **Chiave primaria**. Mai NULL, possibilmente numerica, serve ad identificare in modo univoco un record della tabella. Quindi può essercene una sola per tabella e non può contenere dati duplicati.
2) **Indice**. È una lista ordinata secondo criteri definiti dall'utente dei record della tabella. Può essercene più di una uno. Accelera la ricerca, ma rallenta INSERT e DELETE perché' l'indice deve essere ricostruito ad ognuna di queste query.

PEGASO
Università Telematica

Attenzione! Questo materiale didattico è per uso personale dello studente ed è coperto da copyright. Ne è severamente vietata la riproduzione o il riutilizzo anche parziale, ai sensi e per gli effetti della legge sul diritto d'autore (L. 22.04.1941/n. 633).

Bibliografia

- Atzeni P., Ceri S., Fraternali P., Paraboschi S., Torlone R. (2018). Basi di Dati. McGraw-Hill Education.
- Batini C., Lenzerini M. (1988). Basi di Dati. In Cioffi G. and Falzone V. (Eds). Calderini. Seconda Edizione.

PEGASO
Università Telematica

Attenzione! Questo materiale didattico è per uso personale dello studente ed è coperto da copyright. Ne è severamente vietata la riproduzione o il riutilizzo anche parziale, ai sensi e per gli effetti della legge sul diritto d'autore (L. 22.04.1941/n. 633).

Sitografia

- https://www.w3schools.com/sql/sql_datatypes.asp
- https://forum.html.it/forum/showthread/t-896814.html
- https://www.youtube.com/watch?v=rtmeNwn4mEg

PEGASO
Università Telematica

Attenzione! Questo materiale didattico è per uso personale dello studente ed è coperto da copyright. Ne è severamente vietata la riproduzione o il riutilizzo anche parziale, ai sensi e per gli effetti della legge sul diritto d'autore (L. 22.04.1941/n. 633).