



I CURSORI

I **cursori** in MySQL sono variabili che permettono di **iterare su una serie di risultati** ottenuti da una query.

Sono particolarmente utili nelle **Stored Procedure** quando si vuole elaborare riga per riga un set di dati.

Esempio:

```
Create Procedure `Itera_su_tabella' ()
BEGIN

    declare done int default false;
    declare var_attribute varchar(45);
    declare cur cursor for      select nome from tabella;
    declare continue handler for not found set done = true;
    open cur;
    process_loop: loop
        fetch cur into var_attribute;
        if done then
            leave process_loop;
        end if;
    end loop;
    close cur;
```

END

Il cursore consente di eseguire cicli in cui, a partire dal risultato di una **query SELECT SQL** su una tabella, si legge una **riga** alla volta.

Posso effettuare una select ed ispezionare il **result set** della tabella una tupla alla volta.

Per iterare è necessario un ciclo e, in SQL, si utilizza il costrutto **LOOP ... END LOOP**.

Questo costrutto rappresenta un ciclo **infinito**, che continua ad essere eseguito fino a quando non si verifica

una **condizione di uscita**, altrimenti il ciclo si ripete indefinitamente.

Posso utilizzare una **label** per fornire un nome proprio a questo ciclo, in particolare **process loop**.

All'interno di questo ciclo utilizzo un cursore e per utilizzare un cursore devo dichiarare una variabile di tipo cursore:

```
declare cur cursor for select nome from tabella;
```

La variabile cur di tipo cursore è associata ad uno statement di selezione: **select** nome **from** tabella.

Per utilizzare il cursore lo devo aprire: **open cur**.

Nel momento in cui apro il cursore, il DBMS **esegue lo statement SQL associato al cursore stesso**.

A questo punto, è possibile estrarre **una riga alla volta** dalla tabella risultante dalla **query associata al cursore: FETCH**.

Ora, il contenuto di tutti gli attributi della riga estratta viene inserito nelle variabili locali dichiarate precedentemente.

Di conseguenza, ad ogni iterazione del cursore, la variabile **var_attribute** conterrà il valore dell'attributo **nome**, aggiornandosi man mano che il cursore scorre la tabella.

Ad un certo punto, il cursore raggiungerà la **EOT (End Of Table)**, ovvero la fine della tabella.

A questo punto, **non ci saranno più righe da leggere** e il comando **FETCH** genererà un'**eccezione**, poiché tenterà di estrarre dati inesistenti. Per gestire questo errore, è necessario **intercettarlo** utilizzando un **gestore di continuazione** (CONTINUE HANDLER) per l'eccezione **NOT FOUND**, che rileva quando non ci sono più dati da leggere e imposta una variabile di controllo per interrompere il ciclo:

```
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
```

Appena parte questo gestore, viene eseguito il suo statement SQL, ovvero: **SET done = TRUE**;

A questo punto nell'IF verifichiamo che la variabile è true, entro nell'IF, eseguo **LEAVE process loop.**
E questa LEAVE mi fa uscire dal ciclo.

Esempio di cursore che chiama una procedura:

```
declare done int default false;
declare var_attribute varchar(45);
declare cur cursor for call procedure();
declare continue handler for not found set done = true;
open cur;
process_loop: loop
    fetch cur into var_attribute;
    if done then
        leave process_loop;
    end if;
end loop;
close cur;
```