

Creazione tabella

```
CREATE TABLE NomeTabella (  
  NomeAttributo Dominio [ValoreDiDefault] [NOT NULL | UNIQUE | PRIMARY KEY |  
    REFERENCES NomeTabella(NomeAttributo)]  
  {NomeAttributo Dominio [ValoreDiDefault] [NOT NULL | UNIQUE | PRIMARY KEY |  
    REFERENCES NomeTabella(NomeAttributo)]}  
  [PRIMARY KEY(NomeAttributi)]  
  [FOREIGN KEY(NomeAttributi) REFERENCES NomeTabella(NomeAttributi)]  
    [ON DELETE CASCADE | SET NULL | SET DEFAULT | NO ACTION]  
    [ON UPDATE CASCADE | SET NULL | SET DEFAULT | NO ACTION]  
)
```

Vincoli intrarelazionali

- NOT NULL = indica che un valore nullo non è ammesso come valore dell'attributo.
- UNIQUE = si applica ad un attributo o ad un insieme di attributi di una tabella e impone che i valori dell'attributo/i siano una (super)chiave, cioè righe differenti della tabella non possano avere gli stessi valori.
- PRIMARY KEY = è definito su un singolo attributo oppure elencando più attributi che costituiscono l'identificatore. Gli attributi che fanno parte della chiave primaria non possono assumere valore nullo.

Vincoli interrelazionali

- REFERENCES = si utilizza se c'è un solo attributo coinvolto e si specifica la tabella esterna ed il relativo attributo al quale l'attributo in questione deve essere legato.
- FOREIGN KEY = si utilizza se c'è un insieme di attributi coinvolti. Il costrutto elenca gli attributi della tabella coinvolti nel legame, cui segue la definizione dei corrispondenti attributi della tabella esterna mediante il costrutto REFERENCES.

Modifica tabella

```
ALTER TABLE NomeTabella  
  [ALTER COLUMN NomeAttributo [SET DEFAULT ValoreDiDefault | DROP DEFAULT]] |  
  [ADD CONSTRAINT DefinizioneVincolo] |  
  [DROP CONSTRAINT NomeVincolo] |  
  [ADD COLUMN DefinizioneAttributo] |  
  [DROP COLUMN NomeAttributo]
```

Cancellazione tabella

```
DROP TABLE NomeTabella  
  [RESTRINCT | CASCADE]
```

Interrogazione

```
SELECT ListaAttributi [[AS] Alias] [COUNT({* | [DISTINCT | ALL] ListaAttributi})] |  
  [{Funzione}([DISTINCT | ALL] ListaAttributiOEspressioni {, ListaAttributiOEspressioni [[AS] Alias]  
  | COUNT({* | [DISTINCT | ALL] ListaAttributi})] | [{Funzione}([DISTINCT | ALL]  
  ListaAttributiOEspressioni}  
FROM ListaTabelle [[AS] Alias]
```

```
{[NATURAL | INNER | RIGHT | LEFT | FULL] JOIN NomeTabella [[AS] Alias] ON
  CondizioneDiJoin}
[WHERE CondizioniSemplici | Attributo LIKE Stringa | Attributo IS [NOT] NULL]
[GROUP BY ListaAttributiDiRaggruppamento ≥ ListaAttributiDellaSelect]
[HAVING CondizioniAggregate]
[ORDER BY ListaAttributiDiOrdinamento [ASC | DESC] {, ListaAttributiDiOrdinamento [ASC |
DESC]}]
```

Inserimento di ennuple

```
INSERT INTO NomeTabella [(ListaAttributi)] VALUES(ListaValori) | SELECT ...
```

Eliminazione di ennuple

```
DELETE FROM NomeTabella
[WHERE Condizione]
```

Modifica di ennuple

```
UPDATE NomeTabella
  SET Attributo = Espressione | SELECT ... | NULL | DEFAULT {,
  SET Attributo = Espressione | SELECT ... | NULL | DEFAULT}
[WHERE Condizione]
```

Creazione viste

```
CREATE [OR REPLACE] VIEW NomeVista [(ListaAttributi)] AS SELECT ...
[WITH [LOCAL | CASCADED ] CHECK OPTION]
```

Stringhe

```
FUNCTION(stringa [,opzione])
```

- || = Unisce o concatena due stringhe. Il simbolo | è detto barra verticale spezzata.
- CONCAT = Concatena due stringhe (come ||).
- INITCAP = Pone in maiuscolo la prima lettera di una parola o di una serie di parole.
- LENGTH = Indica la lunghezza di una stringa.
- LOWER = Converte ogni lettera di una stringa in minuscolo.
- UPPER = Converte ogni lettera di una stringa in maiuscolo.
- {} = racchiude le parole riservate come AND se fanno parte del termine di ricerca.
- % = carattere jolly che indica un numero qualsiasi di caratteri.
- _ = carattere jolly che indica un carattere singolo.
- \$ = esegue l'espansione della radice del termine di ricerca prima di eseguire la ricerca stessa.

Funzioni di gruppo

- AVG(*valore*) = Media di *valore*.
- COUNT(*valore*) = Conteggio di righe per colonna.
- MAX(*valore*) = Massimo di tutti i *valore* per gruppi di righe.
- MIN(*valore*) = Minimo di tutti i *valore* per gruppi di righe.

Data

```
SELECT SysDate from DUAL;
```

- GREATEST(data1, data2, data3,...) = prende la più recente fra le date elencate.
- LEAST(data1, data2, data3,...) = prende la meno recente fra le date elencate.
- LAST_DAY(data) = fornisce la data dell'ultimo giorno del mese a cui appartiene *data*.
- MONTHS_BETWEEN(data2, data1) = fornisce *data2-data1* in mesi (può essere un numero frazionario).
- NEXT_DAY(data, 'giorno') = fornisce la data del giorno successivo a *date*, dove 'giorno' è 'Monday', 'Tuesday' e così via.
- TO_CHAR(data, 'formato') = riformatta la data secondo il formato specificato.
- TO_DATE(stringa, 'formato') = converte la data da un particolare formato ('formato') in una data ORACLE. Accetta anche un numero invece di una stringa, con determinati limiti. 'formato' è ristretto.

Ora

select TO_CHAR(SysDate, iHH.MI.SS) Ora from DUAL;

Comandi vari

- DESCRIBE(*NomeTabella*) = dà la descrizione di una tabella.
- NVL(*valore, sostituto*) = sostituto di *valore* se *valore* è NULL.
- COLUMN = fornisce una varietà di istruzioni su titolo, formato e trattamento di una colonna.
- BREAK ON = indica dove inserire degli spazi tra le sezioni di un report, o dove interrompere per subtotali e totali.
- COMPUTE SUM = indica di calcolare subtotali.
- SET LINESIZE = imposta il massimo numero di caratteri consentiti su qualsiasi riga del report.
- SET PAGESIZE = imposta il massimo numero di righe per pagina.
- SET NEWPAGE = imposta il numero di righe vuote tra le pagine.
- COMMIT = conferma gli inserimenti, gli aggiornamenti e le cancellazioni.

N.B.: Le azioni che fanno in modo che si verifichi un commit, anche senza alcuna istruzione esplicita, sono quit, exit (equivalente a quit), create table o create view, drop table o drop view, grant o revoke, connect o disconnect, alter, audit o noaudit. Utilizzare uno di questi comandi significa eseguire automaticamente un commit.

- MINUS = restituisce un record se il punteggio della prima ricerca meno quello della seconda supera la soglia.

Creazione/modifica utenti

CREATE USER *Utente* IDENTIFIED {BY *password* | *externally*};
ALTER USER *Utente* IDENTIFIED BY *password*;

Grant

GRANT {privilegio sistema | ruolo}
[, {privilegio sistema | ruolo}, . . .]
TO {utente | ruolo} [, {utente | ruolo}], . .
[with admin option]

Operazioni sui cursori

Definizione del cursore

DECLARE *NomeCursore* [scroll] CURSOR FOR SELECT ...

Esecuzione dell'interrogazione

OPEN *NomeCursore*

Utilizzo dei risultati (una ennupla alla volta)

FETCH *NomeCursore* INTO *ListaVariabili*

Disabilitazione del cursore

CLOSE CURSOR *NomeCursore*

Accesso alla ennupla corrente (di un cursore su singola relazione a fini di aggiornamento)

CURRENT OF *NomeCursore*

nella clausola where