**Oracle 19c**

**-**

**SQL e PL/SQL**

Distribuito tramite il sito www.manualioracle.it

Copyright © 2023 Assi Loris

Qualsiasi abuso sarà perseguito e punito secondo i termini di legge.

Version: 1.1

Alcuni termini usati sono trademarks registrati dei rispettivi proprietari.

**Sommario**

[Introduzione al Manuale 4](#_Toc149574277)

[Cap. 1 - Introduzione 5](#_Toc149574278)

[1.1 – Introduzione ad Oracle 5](#_Toc149574279)

[1.2 – SQL\*Plus 5](#_Toc149574280)

[1.3 – Utilizzo di alcuni Comandi dentro SQL\*Plus 7](#_Toc149574281)

[1.4 – Introduzione all’SQL e al PL/SQL 8](#_Toc149574282)

[1.5 – Case sensitive in Oracle 9](#_Toc149574283)

[1.6 – Character encoding 10](#_Toc149574284)

[Cap. 2 - Le basi dell’SQL in Oracle 11](#_Toc149574285)

[2.1 – Elementi base dell’Oracle SQL 11](#_Toc149574286)

[2.2 – Creare una Tabella 12](#_Toc149574287)

[2.3 – Le tabelle EMP e DEPT 13](#_Toc149574288)

[2.4 – Popolare e Interrogare EMP e DEPT 14](#_Toc149574289)

[2.5 – Query e Subquery 16](#_Toc149574290)

[2.6 – Le Pseudocolumns 17](#_Toc149574291)

[2.7 – SQLERRM e SQLCODE 18](#_Toc149574292)

[Cap. 3 - Oracle SQL e Standard SQL 20](#_Toc149574293)

[3.1 – DECODE 20](#_Toc149574294)

[3.2 – CASE 20](#_Toc149574295)

[3.3 – Introduzione alla Inner Join 22](#_Toc149574296)

[3.4 – Tipi di Inner Join 23](#_Toc149574297)

[3.5 – Cross Join e Outer Join 25](#_Toc149574298)

[Cap. 4 - Comandi SQL Avanzati 27](#_Toc149574299)

[4.1 – Clausola WITH 27](#_Toc149574300)

[4.2 – MERGE SQL statement 28](#_Toc149574301)

[4.3 – Case-Insensivite and Accent-Insensitive Queries 32](#_Toc149574302)

[4.4 – Subquery Scalari 32](#_Toc149574303)

[4.5 – Introduzione alle Analytic Function 33](#_Toc149574304)

[4.6 – Altri SQL Statement 34](#_Toc149574305)

[Cap. 5 - Introduzione al linguaggio PL/SQL 37](#_Toc149574306)

[5.1 – Character Sets e NLS\_DATABASE\_PARAMETER 37](#_Toc149574307)

[5.2 – Lexical Units 37](#_Toc149574308)

[5.3 – Delimiters e Comments 38](#_Toc149574309)

[5.4 – Il PL/SQL Program Construct e i suoi Componenti 39](#_Toc149574310)

[5.5 – Variabili PL/SQL 40](#_Toc149574311)

[5.6 – Le Bind Variables 41](#_Toc149574312)

[5.7 – PL/SQL Control Statements 42](#_Toc149574313)

[5.8 – I Subprograms 43](#_Toc149574314)

[Cap. 6 - Le Procedure 45](#_Toc149574315)

[6.1 – Cosa è una stored procedure 45](#_Toc149574316)

[6.2 – Creare e droppare una procedura 45](#_Toc149574317)

[6.3 – Formal and Actual Parameters 47](#_Toc149574318)

[6.4 – Procedure con Parametri e i Tipi di parameter modes 48](#_Toc149574319)

[6.5 – Invocare una procedura con parametri 50](#_Toc149574320)

[6.6 – Il subprogram nella *Declarative Section* di una Procedura 50](#_Toc149574321)

[6.7 – Come vengono propagate le Exception 52](#_Toc149574322)

[Cap. 7 - Le Funzioni 54](#_Toc149574323)

[7.1 – Cosa è una stored function 54](#_Toc149574324)

[7.2 – Creare, Invocare e Droppare una Funzione 54](#_Toc149574325)

[7.3 – Vantaggi e restrizioni delle chiamate dagli SQL statement 56](#_Toc149574326)

[7.4 – Dove chiamare le Funzioni negli SQL statement 57](#_Toc149574327)

[7.5 – Differenze fra Procedure e Funzioni 58](#_Toc149574328)

[Cap. 8 - I Subprograms 59](#_Toc149574329)

[8.1 – Grants: System Privileges e Object Privileges 59](#_Toc149574330)

[8.2 – Differenze fra Invokers Rights e Definers Rights 61](#_Toc149574331)

[8.3 – Le viste di dizionario per gestire gli stored objects 61](#_Toc149574332)

[8.4 – Quando un subprogram cambia status (VALID / INVALID) 62](#_Toc149574333)

[Cap. 9 - I Package 65](#_Toc149574334)

[9.1 – Creare e Droppare Package specification e Package body 65](#_Toc149574335)

[9.2 – Usare la DESCRIBE per descrivere i package e i suoi componenti 66](#_Toc149574336)

[9.3 – Package Construct Public e Private 68](#_Toc149574337)

[9.4 – Invocare un package construct 68](#_Toc149574338)

[9.5 – Usare un bodiless package 68](#_Toc149574339)

[9.6 – I Package: benefici, errori e overloading 69](#_Toc149574340)

[9.7 – Usare la Forward Referencing 70](#_Toc149574341)

[9.8 – Inizializzare variabili con una *one-time-only* procedure 71](#_Toc149574342)

[9.9 – Usare le PL/SQL tables e i records nei Package 72](#_Toc149574343)

[Cap. 10 - Dynamic SQL e Oracle Supplied Packages 74](#_Toc149574344)

[10.1 – Native Dynamic SQL: l’Execute Immediate 74](#_Toc149574345)

[10.2 – Oracle supplied Packages: dal DBMS\_ADDM al DBMS\_LOCK 75](#_Toc149574346)

[10.3 – Oracle supplied Packages: dal DBMS\_METADATA all’UTL\_TCP 76](#_Toc149574347)

[Cap. 11 - I Trigger 79](#_Toc149574348)

[11.1 – Introduzione ai Trigger 79](#_Toc149574349)

[11.2 – Creare un DML trigger 80](#_Toc149574350)

[11.3 – I trigger *row level* e i trigger *statement level* 81](#_Toc149574351)

[11.4 – I conditional predicated in un trigger 82](#_Toc149574352)

[11.5 – Usare i qualifiers OLD e NEW in un DML Trigger 83](#_Toc149574353)

[11.6 – Gli INSTEAD OF trigger 85](#_Toc149574354)

[11.7 – Eventi che fanno scattare il DB trigger 86](#_Toc149574355)

[11.8 – Creare un trigger per un DDL statement 87](#_Toc149574356)

[11.9 – Creare un trigger per un system event 87](#_Toc149574357)

[11.10 – Cambiare un Trigger Status o Rimuovere un Trigger 88](#_Toc149574358)

[11.11 – Privilegi e Dictionary views associate ai trigger 89](#_Toc149574359)

[Cap. 12 - Oracle Data Redaction 91](#_Toc149574360)

[12.1 – Introduzione all’Oracle Data Redaction 91](#_Toc149574361)

[12.2 – Oracle Data Redaction policies 91](#_Toc149574362)

[12.3 – Character redaction shortcut 93](#_Toc149574363)

[Cap. 13 - Miscellaneous 94](#_Toc149574364)

[13.1 – Approccio DTAP nei database 94](#_Toc149574365)

[13.2 – Valid-Time Temporal : Definizioni 95](#_Toc149574366)

[13.3 – Settare ed Usare la Valid-Time Temporal 95](#_Toc149574367)

[13.4 – Row limiting clause 98](#_Toc149574368)

[13.5 – Introduzione al JSON 99](#_Toc149574369)

[13.6 – Vantaggi e svantaggi nell’uso dei JSON 100](#_Toc149574370)

Introduzione al Manuale

 **Contenuto**

In relazione al database Oracle 19c, il presente manuale tratta:

 - le basi dell’SQL

 - le basi del linguaggio di programmazione PL/SQL

 - cenni sulle nuove feature (SQL e PL/SQL) create nelle ultime versioni Oracle

 **Audience**

Il presente manuale è rivolto a chiunque voglia imparare le basi dell’SQL e del linguaggio PL/SQL.

 **Particolarità**

Il presente manuale è in italiano ma molti termini tecnici in esso contenuti sono in lingua inglese.

Abbiamo fatto tale scelta perché spesso a lavoro tali termini vengono detti in inglese e perché riteniamo più utile che i lettori del manuale conoscano tali termini nella lingua usata nella documentazione ufficiale Oracle.

Molti argomenti sono schematizzati al fine di fornire una comprensione ed una memorizzazione superiore.

A volte quando nel manuale si parla di una feature particolare, citiamo anche la versione Oracle in cui tale feature è stata creata.

Alcune abbreviazioni usate:

db : database

es.: esempio

init parameter: initialization parameter

CBO : Cost Based Optimizer (ottimizzatore Oracle basato sul “costo”)

ICT : Information and Communications Technology

MOS o M.O.S. : My Oracle Support

 **Principali Versioni**

- 26.10.2023: version 1.0

- 30.10.2023 : version 1.1 varie modifiche e rilascio al pubblico

**Disclaimer**

Non si fornisce alcuna garanzia relativamente al fatto che il presente documento sia privo di errori. Non si assume nessuna responsabilità sugli eventuali errori o danni derivanti dall’uso delle informazioni qui contenute.

1. Introduzione

### 1.1 – Introduzione ad Oracle

Riteniamo utile fare una premessa riguardante l’ambiente in cui si utilizza l’SQL e il PL/SQL e dunque qui forniamo una breve definizione dei concetti più importanti riguardanti il database Oracle.

**DBMS (Database Management System)**

Software progettato per creare e manipolare in modo efficiente un *database* (ovvero un insieme di dati strutturati) solitamente da parte di più utenti.

Esistono 2 differenti tipi di DBMS:

* RDBMS
* No-SQL database

**RDBMS (Relational Database Management System)**

Come si evince dal suo nome in inglese, è un sistema di gestione del database basato sul modello relazionale.

Il modello relazionale è un modello logico di rappresentazione dei dati.

Secondo tale modello, qualsiasi dato può essere rappresentato come relazione. E tale relazione può essere rappresentata graficamente tramite una tabella.

**Oracle Server**

E’ costituito da 2 principali componenti:

* Il database:

E’ una raccolta di dati correlati. Ha una struttura logica (tabelle, indici, ecc...) e una fisica (i file in cui sono immagazzinati i dati)

* L’istanza:

E’ costituita da una struttura di memoria e dai processi in background che gestiscono il db

..........................

..........................

..........................

1. Le basi dell’SQL in Oracle

### 2.1 – Elementi base dell’Oracle SQL

Gli elementi basilari da conoscere per capire gli *SQL statements* in Oracle sono tanti e sono elencati nella documentazione ufficiale della Oracle.

Qui trattiamo quelli più usati:

* I *data type*:

Ogni valore gestito da Oracle ha un *data type*.

Es.:

Quando crei una tabella, devi specificare un *data type* per ognuna delle sue colonne.

Quando crei una procedura o una funzione, devi specificare un *data type* per ognuno dei suoi argomenti (parametri di INPUT e di OUTPUT).

 I *data types* più usati in Oracle sono:

 - CHAR

- VARCHAR2

 - NUMBER

 - DATE

 - BLOB e CLOB

* *Concatenation Operator:*

Tramite il “doppio pipe” ( || ) puoi concatenare dati di tipo stringa e CLOB.

Es.:

 SELECT 'Name is ' || last\_name

 FROM emp

 ORDER BY last\_name;

..........................

..........................

..........................

1. Oracle SQL e Standard SQL

L’SQL è nato nei primi anni ’70 nei laboratori dell’IBM. Negli anni l’SQL si è evoluto e ne sono nate diverse versioni.

Per tentare di unificare tali versioni dell’SQL, sia l’*American National Standards Institute* (ANSI) che l’*International Organization for Standardization* (ISO) sin dagli anni ’80 hanno creato degli standard specifici per il linguaggio di interrogazione dei database.

E’ utile rispettare tali standard soprattutto perché così facendo potrai trasportare del codice SQL da un database non Oracle ad uno Oracle (e viceversa) senza dover fare modifiche.

Qui descriviamo alcuni comandi SQL: la maggioranza di essi rispetta la sintassi ANSI/ISO, alcuni invece non la rispettano ma li descriviamo comunque perchè può capitare di incontrarli.

### 3.1 – DECODE

Sintassi della DECODE:



La DECODE compara *expr* con *search*:

..........................

..........................

..........................

1. Comandi SQL Avanzati

### 4.1 – Clausola WITH

Questa clausola serve a dare un nome ad un *query block* contenuto dentro una SELECT che richiama più volte tale *query block*.

Alcune caratteristiche della *WITH clause*:

- ti permette di contenere multipli *query blocks*, separati da virgole

- a volte è utile per le performance e spesso è utile per una miglior leggibilità della query

- è internamente risolta come una *in-line view* o una *temporary table*; il CBO sceglie la risoluzione appropriata

**Es:**

Cerchiamo tutti i dipartimenti il cui totale dei salari è maggiore di 1/8 dei totali dei salari della ditta:

Senza la clausola WITH, la query verrebbe scritta così:

 SELECT d.dept\_name,

 Sum(e.salary) as dept\_total

 FROM emp e,

 dept d

 WHERE e.deptno=d.deptno

 GROUP BY d.dept\_name

 HAVING sum(e.salary) >

 (SELECT sum(e.salary) \* 1/8

 FROM emp e,

 dept d

 WHERE e.deptno=d.deptno)

 ORDER BY sum(e.salary) DESC;

..........................

..........................

..........................

1. Introduzione al linguaggio PL/SQL

### 5.1 – Character Sets e NLS\_DATABASE\_PARAMETER

I *character data* che devono essere processati dal PL/SQL o contenuti in un db devono essere rappresentati come una sequenza di *bytes*.

La *byte rapresentation* di un singolo carattere è chiamata *character code.*

Un insieme di *character code* è chiamato *character set.*

I database e il PL/SQL supportano un *database character set* e un *national character set*.

 **Database Character Set**

Il PL/SQL usa il *database character set* per rappresentare:

* *Stored source* *text* delle unità PL/SQL
* *Character values* dei *data types* CHAR, VARCHAR2, CLOB e LONG

Il *database character set* può essere o *single-byte* o *multi-byte*.

Es. di *database character set:*

..........................

..........................

..........................