**Oracle 12c**

**-**

**SQL e PL/SQL**

Distribuito tramite il sito www.manualioracle.it

Copyright © 2015 Assi Loris

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

Version: 3.7.2

Alcuni termini usati sono trademarks registrati dei rispettivi proprietari.

**Sommario**

[Introduzione al Manuale 4](#_Toc107418445)

[Cap. 1 - Introduzione 5](#_Toc107418446)

[1.1 – Introduzione ad Oracle 5](#_Toc107418447)

[1.2 – SQL\*Plus 6](#_Toc107418448)

[1.3 – Utilizzo di alcuni Comandi dentro SQL\*Plus 7](#_Toc107418449)

[1.4 – Introduzione all’SQL e al PL/SQL 8](#_Toc107418450)

[Cap. 2 - Le basi dell’SQL in Oracle 10](#_Toc107418451)

[2.1 – Elementi base dell’Oracle SQL 10](#_Toc107418452)

[2.2 – Creare una Tabella 11](#_Toc107418453)

[2.3 – Le tabelle EMP e DEPT 12](#_Toc107418454)

[2.4 – Popolare e Interrogare EMP e DEPT 13](#_Toc107418455)

[2.5 – Query e Subquery 15](#_Toc107418456)

[2.6 – Le Pseudocolumns 16](#_Toc107418457)

[2.7 – SQLERRM e SQLCODE 17](#_Toc107418458)

[Cap. 3 - Oracle SQL e Standard SQL 19](#_Toc107418459)

[3.1 – CASE 19](#_Toc107418460)

[3.2 – Introduzione alla Inner Join 20](#_Toc107418461)

[3.3 – Tipi di Inner Join 20](#_Toc107418462)

[3.4 – Cross Join e Outer Join 22](#_Toc107418463)

[Cap. 4 - Comandi SQL Avanzati 25](#_Toc107418464)

[4.1 – Clausola WITH 25](#_Toc107418465)

[4.2 – MERGE SQL statement 26](#_Toc107418466)

[4.3 – Case-Insensivite and Accent-Insensitive Queries 29](#_Toc107418467)

[4.4 – Altri SQL Statement 30](#_Toc107418468)

[Cap. 5 - Introduzione ai PL/SQL Programs 33](#_Toc107418469)

[5.1 – Character Sets 33](#_Toc107418470)

[5.2 – Lexical Units 33](#_Toc107418471)

[5.3 – Delimiters e Comments 34](#_Toc107418472)

[5.4 – Il PL/SQL Program Construct e i suoi Componenti 35](#_Toc107418473)

[5.5 – Variabili PL/SQL 36](#_Toc107418474)

[5.6 – Le Bind Variables 37](#_Toc107418475)

[5.7 – PL/SQL Control Statements 38](#_Toc107418476)

[5.8 – I Subprograms 39](#_Toc107418477)

[Cap. 6 - Le Procedure 41](#_Toc107418478)

[6.1 – Cosa è una stored procedure 41](#_Toc107418479)

[6.2 – Creare e droppare una procedura 41](#_Toc107418480)

[6.3 – Formal and Actual Parameters 43](#_Toc107418481)

[6.4 – Procedure con Parametri e i Tipi di parameter modes 44](#_Toc107418482)

[6.5 – Invocare una procedura con parametri 45](#_Toc107418483)

[6.6 – Il subprogram nella *Declarative Section* di una Procedura 46](#_Toc107418484)

[6.7 – Come vengono propagate le Exception 47](#_Toc107418485)

[Cap. 7 - Le Funzioni 49](#_Toc107418486)

[7.1 – Cosa è una stored function 49](#_Toc107418487)

[7.2 – Creare, Invocare e Droppare una Funzione 49](#_Toc107418488)

[7.3 – Vantaggi e restrizioni delle chiamate dagli SQL statement 51](#_Toc107418489)

[7.4 – Dove chiamare le Funzioni negli SQL statement 52](#_Toc107418490)

[7.5 – Differenze fra Procedure e Funzioni 52](#_Toc107418491)

[Cap. 8 - I Subprograms 54](#_Toc107418492)

[8.1 – Grants: System Privileges e Object Privileges 54](#_Toc107418493)

[8.2 – Differenze fra Invokers Rights e Definers Rights 55](#_Toc107418494)

[8.3 – Le viste di dizionario per gestire gli stored objects 56](#_Toc107418495)

[Cap. 9 - I Package 58](#_Toc107418496)

[9.1 – Creare e Droppare Package specification e Package body 58](#_Toc107418497)

[9.2 – Usare la DESCRIBE per descrivere i package e i suoi componenti 59](#_Toc107418498)

[9.3 – Package Construct Public e Private 61](#_Toc107418499)

[9.4 – Invocare un package construct 61](#_Toc107418500)

[9.5 – Usare un bodiless package 62](#_Toc107418501)

[9.6 – I Package: benefici, errori e overloading 62](#_Toc107418502)

[9.7 – Usare la Forward Referencing 63](#_Toc107418503)

[9.8 – Inizializzare variabili con una *one-time-only* procedure 64](#_Toc107418504)

[9.9 – Usare le PL/SQL tables e i records nei Package 65](#_Toc107418505)

[Cap. 10 - Dynamic SQL e Oracle Supplied Packages 67](#_Toc107418506)

[10.1 – Native Dynamic SQL: l’Execute Immediate 67](#_Toc107418507)

[10.2 – Oracle supplied Packages: dal DBMS\_ADDM al DBMS\_LOCK 68](#_Toc107418508)

[10.3 – Oracle supplied Packages: dal DBMS\_METADATA all’UTL\_TCP 69](#_Toc107418509)

[10.4 – Nuovi Oracle supplied Packages 71](#_Toc107418510)

[Cap. 11 - I Trigger 73](#_Toc107418511)

[11.1 – Introduzione ai Trigger 73](#_Toc107418512)

[11.2 – Creare un DML trigger 75](#_Toc107418513)

[11.3 – I trigger *row level* e i trigger *statement level* 76](#_Toc107418514)

[11.4 – I conditional predicated in un trigger 77](#_Toc107418515)

[11.5 – Usare i qualifiers OLD e NEW in un DML Trigger 78](#_Toc107418516)

[11.6 – Gli INSTEAD OF trigger 79](#_Toc107418517)

[11.7 – Eventi che fanno scattare il DB trigger 81](#_Toc107418518)

[11.8 – Creare un trigger per un DDL statement 81](#_Toc107418519)

[11.9 – Creare un trigger per un system event 82](#_Toc107418520)

[11.10 – La Mutating table 83](#_Toc107418521)

[11.11 – Cambiare un Trigger Status o Rimuovere un Trigger 84](#_Toc107418522)

[11.12 – Privilegi e Dictionary views associate ai trigger 84](#_Toc107418523)

[Cap. 12 - Miscellaneous 86](#_Toc107418524)

[12.1 – Oracle Application Express (APEX) 86](#_Toc107418525)

[12.2 – Approccio DTAP nei database 86](#_Toc107418526)

[Cap. 13 - New Features SQL e PL/SQL 88](#_Toc107418527)

[13.1 – Valid-Time Temporal : Definizioni 88](#_Toc107418528)

[13.2 – Settare ed Usare la Valid-Time Temporal 88](#_Toc107418529)

[13.3 – Introduzione all’Oracle Data Redaction 91](#_Toc107418530)

[13.4 – Oracle Data Redaction policies 91](#_Toc107418531)

[13.5 – Character redaction shortcut 93](#_Toc107418532)

[13.6 – Row limiting clause 93](#_Toc107418533)

Introduzione al Manuale

 **Contenuto**

In relazione al database Oracle 12c, il presente manuale tratta:

 - le basi dell’SQL

 - le basi del linguaggio di programmazione PL/SQL

 - un focus su alcune nuove feature create in Oracle 12c (descritte soprattutto nell'ultimo capitolo)

 **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

 **Principali Versioni**

- 02.10.2015: version 1.0

- 03.10.2015 : version 1.1 varie modifiche e rilascio al pubblico

- 02.02.2017 : version 2.1 varie modifiche

- 19.08.2017 : version 3.0 varie modifiche e aggiunti i paragrafi relativi all’SQL base

**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.

Fra i DBMS più diffusi c’è l’*Oracle* *database*, prodotto dalla società americana *Oracle Corporation.*

Altri DBMS diffusi sono:

* IBM DB2
* Microsoft Access
* Microsoft SQL Server
* MongoDB (questo è l’unico qui elencato che non è un RDBMS: viene definito un *document store*)
* Oracle MySQL
* PostgreSQL

**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.

…………..

…………..

…………..

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 types*:

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.

 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 perchè può capitare di incontrarli.

### 3.1 – CASE

L’ISO/ANSI ha 4 tipi di *CASE expressions*:

**Simple CASE:**

E’ simile alla DECODE.

 Es:

SELECT ename

 ,(CASE to\_char(hiredate, 'yyyy')

 WHEN '1981' THEN '35 anni di servizio'

 ELSE 'forse il prossimo anno!'

 END) as "Premi per il 2016"

 FROM emp;

**Searched CASE:**

…………..

…………..

…………..

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;

Ed in questo modo, Oracle avrebbe fatto 2 volte un *Full Table Scan* sulla tabella DEPT.

Con la clausola WITH, la stessa query la scriveresti così:

…………..

…………..

…………..

1. Introduzione ai PL/SQL Programs

### 5.1 – Character Sets

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 set 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:*

…………..

…………..

…………..