

Laboratorio Oracle

Prof. Alessandra Lumini
alessandra.lumini@unibo.it

1

Oracle SQL Developer

- ❑ Download:
 - <http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>
- ❑ Unzip, Run
- ❑ Piattaforme supportate
 - Microsoft Windows sia a 32 che a 64 bit, Mac OSX e Linux.

2

Oracle SQL Developer - funzionalità

- ❑ Per progettisti
 - integrazione di **SQL Developer Data Modeler**, per la progettazione e lo sviluppo di modelli dei dati.
 - opzioni di **esportazione e importazione dati**
- ❑ Per DBA
 - strumenti di **scheduling** delle operazioni
 - strumenti di gestione dei **parametri di configurazione** del database, dei **profili** di sicurezza e auditing e di tutti i **file** (*redo log, control file, archive log, data file*) che compongono il database.
 - **SQL Tuning Advisor**, per il **tuning degli statement SQL**
 - strumenti a supporto della **migrazione**
- ❑ Per sviluppatori
 - **Query Builder** visuale integrato con il corrispondente editor testuale di comandi SQL.

3

Caratteristiche di SQL Developer

- ❑ Ambiente per scrivere query SQL, PL/SQL e script
- ❑ Ambiente GUI per creare, cancellare, modificare oggetti Oracle (tabelle, viste, utenti, procedure...)
- ❑ Ambiente di debug per PL/SQL
- ❑ Connessioni multiple
- ❑ Explain Plain

4

Ambiente di lavoro

Foglio di Lavoro SQL:

- + per creare un foglio di lavoro associato ad una determinata connessione

Pannello Connessioni:

- + per creare una connessione (segue)

Pannello DBA:

- Visualizza->DBA
- Info per amministratori
- + per creare una connessione

Pannello Output DBMS:

- Visualizza->Output DBMS
- + per abilitare Output comandi: DBMS_OUTPUT.PUT_LINE()

5

Creare una connessione

- ❑ Nome host: **si-oracle-11.csr.unibo.it**
- ❑ SID: **SISINF**
- ❑ Utente: **<nome>** Password: **<pwd>**
- ❑ Porta **1521** (11521 per l'accesso da esterno)

Nuovo / Seleziona connessione al database

Nome connessione: **SISINF_G1**

Nome utente: **GRUPPO1**

Password: *********

☒ Salva password

Oracle Access

Tipo di connessione: **Base** Ruolo: **predefinito**

Nome host: **si-oracle-11.csr.unibo.it**

Porta: **1521**

☒ SID: **SISINF**

☐ Nome servizio

☐ Autenticazione sistema operativo ☐ Autenticazione Kerberos ☐ Connessione proxy

Stato: **2**

Salva Cancellata Test Cgnetti Annulla

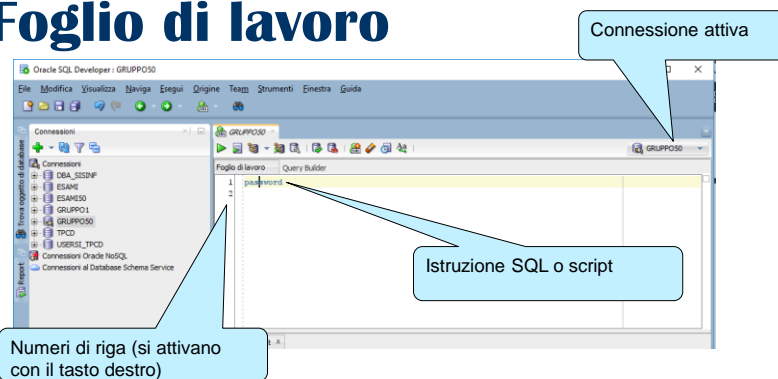
6

Importare una connessione

- ❑ Tasto destro su Connessioni->Importa
- ❑ Scegliere il file **stud_conn.xml**
- ❑ Chiave di cifratura **sisinfo**
- ❑ Modificare la connessione **Gruppo50**
- ❑ Nome Connessione: **<NomeAScelta >**
- ❑ Nome Utente: **<username>**
- ❑ Salvare
- ❑ Cambiare la password: digitare **password** nella finestra di comando

7

Foglio di lavoro



- Execute Statement (F9)
- Run Script (F5)
- Commit (F2)
- Rollback (F3)
- Cancel (Ctrl+Q)
- SQL History (F9)
- Execute Explain Plan (F6)
- Clear (Ctrl+D)

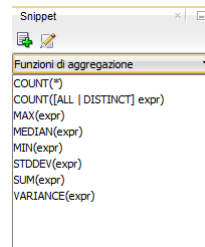
Esegue l'istruzione evidenziata o su cui è posizionato il cursore. Permette di richiedere variabili bind (precedute da ":").

Esegue più di una istruzione. Il risultato è mostrato nella finestra di script output

8

Snippet

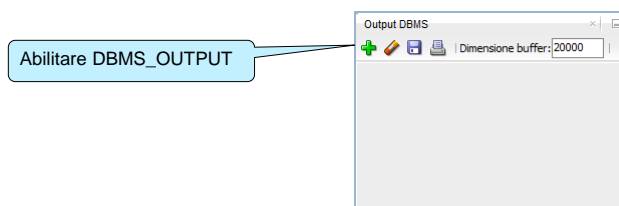
- ❑ SQL developer fornisce molti snippet per aiutare a scrivere codice
- ❑ Menu visualizza->snippets
- ❑ Scegliere la classe e trainare l'istruzione



9

DBMS_OUTPUT.PUT_LINE

- ❑ L'istruzione DBMS_OUTPUT.PUT_LINE può essere usata per scrivere in output valori di controllo o messaggi di errore utili in fase di debug
- ❑ L'output viene mostrato nella finestra DBMS_OUTPUT (Visualizza->Output DBMS)
- ❑ DBMS_OUTPUT deve essere abilitato per la connessione corrente prima di lanciare i test (tramite pulsante o con istruzione DBMS_OUTPUT.ENABLE)

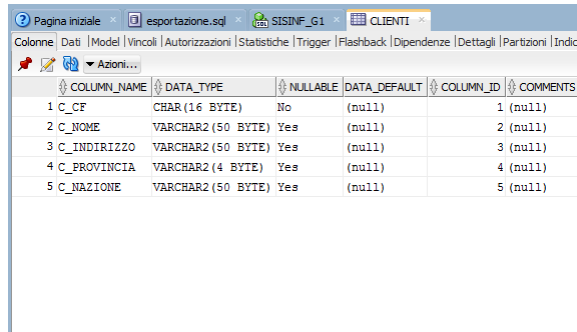


10

Visualizzare una Tabella

- Selezionando una tabella si apre un pannello multi-tab che permette di visualizzarne:

- Struttura
- Dati
- Modello
- Vicoli
- Statistiche
- Trigger
-
- Istruzioni SQL



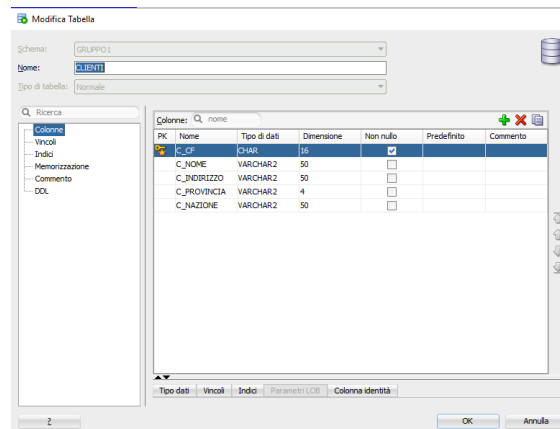
The screenshot shows a database management interface with a multi-tabbed window. The active tab is 'CLIENTI'. Below the tabs, there is a table showing the structure of the 'CLIENTI' table. The table has columns: COLUMN_NAME, DATA_TYPE, NULLABLE, DATA_DEFAULT, COLUMN_ID, and COMMENTS. The data is as follows:

COLUMN_NAME	DATA_TYPE	NULLABLE	DATA_DEFAULT	COLUMN_ID	COMMENTS
C_CF	CHAR (16 BYTE)	No	(null)	1	(null)
C_NOME	VARCHAR2 (50 BYTE)	Yes	(null)	2	(null)
C_INDIRIZZO	VARCHAR2 (50 BYTE)	Yes	(null)	3	(null)
C_PROVINCIA	VARCHAR2 (4 BYTE)	Yes	(null)	4	(null)
C_NAZIONE	VARCHAR2 (50 BYTE)	Yes	(null)	5	(null)

11

Modificare una tabella

- Con il menù contestuale Modifica si apre un pannello per editarne lo schema



The screenshot shows a 'Modifica Tabella' dialog box. It has a 'Schema' dropdown set to 'GRUPPO1' and a 'Nome' text box containing 'CLIENTI'. Below these, there is a 'Tipo di tabella' dropdown set to 'Normale'. On the left, there is a 'Ricerca' section with a list of options: 'Colonne', 'Vicoli', 'Indici', 'Memorizzazione', 'Commento', and 'DDL'. The 'Colonne' option is selected. The main area shows a table with columns: PK, Nome, Tipo di dati, Dimensione, Non nullo, Predefinito, and Commento. The data is as follows:

PK	Nome	Tipo di dati	Dimensione	Non nullo	Predefinito	Commento
<input checked="" type="checkbox"/>	C_CF	CHAR	16	<input checked="" type="checkbox"/>		
<input type="checkbox"/>	C_NOME	VARCHAR2	50	<input type="checkbox"/>		
<input type="checkbox"/>	C_INDIRIZZO	VARCHAR2	50	<input type="checkbox"/>		
<input type="checkbox"/>	C_PROVINCIA	VARCHAR2	4	<input type="checkbox"/>		
<input type="checkbox"/>	C_NAZIONE	VARCHAR2	50	<input type="checkbox"/>		

At the bottom, there are tabs for 'Tipo dati', 'Vicoli', 'Indici', 'Parametri LCB', and 'Colonna identità'. The 'Tipo dati' tab is active. There are 'OK' and 'Annulla' buttons at the bottom right.

12

Esercizi

A. Creare le seguenti tabelle:

LA_VENDITE(V_Num,Prodotti:V_Prodotto,V_Data,V_Qta)

LA_PRODOTTI(P_Cod,P_Nome,P_QtaDisp,P_PrezzoList)

B. Aggiungere la tupla alla tabella LA_PRODOTTI

P01,Vite,1000, 0.05

C. Creare un indice sull'attributo P_Nome della tabella LA_PRODOTTI memorizzandolo nel tablespace INDX.

13

Tips

- Creazione di tabelle (vedi 16.6 SQL Language Reference)

```
CREATE TABLE nometabella (  
  nomecol DATATYPE,  
  nomecol DATATYPE,  
  PRIMARY KEY (nomecol,...,nomecol)  
  [,FOREIGN KEY (nomecol,...,nomecol)  
  REFERENCES nometabella (nomecol,...,nomecol)];
```

dove DATATYPE è uno dei tipi di dati previsti da ORACLE
- Inserimento valori (vedi 18.54 SQL Language Reference)

```
INSERT INTO nometabella  
VALUES (valore,...,valore);
```
- Creazione di indici (vedi 14.61 SQL Language Reference)

```
CREATE INDEX [UNIQUE,BITMAP] nomeindice  
ON nometabella (nomecol,...,nomecol);
```

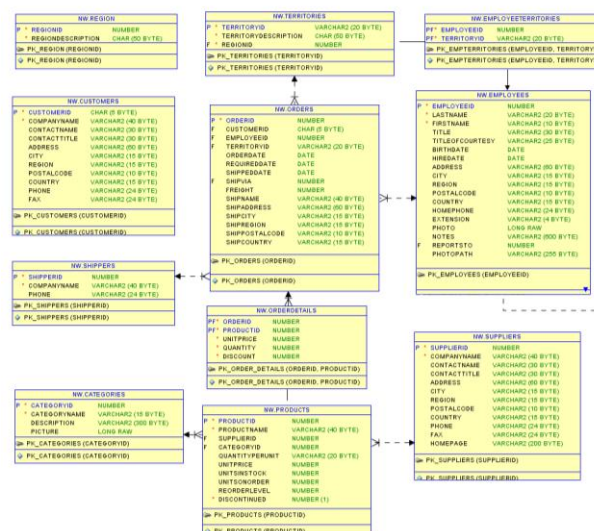
14

Il dialetto PL/SQL

- ❑ SQL su Oracle differisce in alcuni costrutti dallo standard
 - ❑ Gestione delle date
 - ❑ Table expression
 - ❑ Query Top K
 - ❑ Statement Case
 - ❑ ...
- ❑ Si vedano gli esercizi su 0_LabSQL per un ripasso del linguaggio SQL base

16

Schema DB



17

L'istruzione SELECT

- È l'istruzione che permette di eseguire interrogazioni (*query*) sul DB

```
SELECT  A1,A2,...,Am
FROM    R1,R2,...,Rn
[WHERE  <condizione>]
[GROUP BY <listaAttributi>]
[HAVING <condizione>]
[ORDER BY <listaAttributi>]
```

➤ ovvero:

- SELECT (o TARGET) list (cosa si vuole come risultato)
- clausola FROM (da dove si prende)
- clausola WHERE (che condizioni deve soddisfare)
- clausola GROUP BY (le colonne su cui raggruppare)
- clausola HAVING (condizione relative ai gruppi)
- clausola ORDER BY (ordinamento)

Il comando SELECT permette di realizzare le operazioni di selezione, proiezione, join, raggruppamento e ordinamento.

Esempio di query complessa

- Selezionare gli impiegati che hanno gestito più di 50 ordini

```
SELECT A.LASTNAME, A.FIRSTNAME, count(*) as NumeroOrdini
FROM NW.ORDERS B, NW.EMPLOYEES A
WHERE A.EMPLOYEEID = B.EMPLOYEEID
group by A.EMPLOYEEID, A.LASTNAME, A.FIRSTNAME
having count(*)> 50
order by 3 desc;
```

1. Selezionare i prodotti che sono stati venduti a più di 10 clienti diversi

Gestione delle date

Funzioni di conversione Stringa-Data

TO_DATE (stringa [, formato])	TO_DATE('2003/07/09', 'yyyy/mm/dd')	Result: date value of July 9, 2003
	TO_DATE('070903', 'MMDDYY')	Result: date value of July 9, 2003
	TO_DATE('20020315', 'yyyymmdd')	Result: date value of Mar 15, 2002

TO_CHAR (value [, format_mask])	TO_CHAR(sysdate, 'yyyy/mm/dd')	Result: '2003/07/09'
	TO_CHAR(sysdate, 'Month DD, YYYY')	Result: 'July 09, 2003'
	TO_CHAR(sysdate, 'YYYY')	Result: '2003'

Selezionare gli ordini dal 1/1/1998 al 1/2/1998

```
SELECT ORDERID, ORDERDATE
FROM NW.ORDERS
WHERE ORDERDATE
BETWEEN TO_DATE ('01/01/1998', 'dd/mm/yyyy')
AND TO_DATE ('1998/02/01', 'yyyy/mm/dd');
```

2. Selezionare gli impiegati che hanno più di 60 anni

20

Gestione delle date

Funzione di EXTRACT

EXTRACT ({ YEAR MONTH DAY } FROM date_value)	EXTRACT(YEAR FROM DATE '2003-08-22')	Result: 2003
	EXTRACT(MONTH FROM DATE '2003-08-22')	Result: 8
	EXTRACT(DAY FROM DATE '2003-08-22')	Result: 22

Ordinare gli impiegati per anno di nascita (dal più giovane)

```
SELECT EMPLOYEEID, LASTNAME, FIRSTNAME, EXTRACT( YEAR FROM BIRTHDATE)
FROM NW.EMPLOYEES
ORDER BY 4 DESC;
```

```
SELECT EMPLOYEEID, LASTNAME, FIRSTNAME, to_char(BIRTHDATE, 'YYYY')
FROM NW.EMPLOYEES
ORDER BY 4 DESC;
```

3. Selezionare gli ordini per cui l'anno di spedizione è diverso da quello di ordine

21

Table expressions

- Table expressions = viste create dinamicamente in memoria su cui si possono poi effettuare delle query.

```
WITH <alias_name> AS (subquery_sql_statement)
SELECT <column_name_list> FROM <alias>;
```

- Selezionare l'impiegato che ha gestito il maggior numero di ordini

```
WITH imp1 AS (SELECT LASTNAME, FIRSTNAME, count(ORDERID) as NumOrdini
FROM NW.EMPLOYEES E join NW.ORDERS o on (e.EMPLOYEEID=o.EMPLOYEEID)
group by e.EMPLOYEEID, e.LASTNAME, e.FIRSTNAME
order by 3 desc)
```

```
SELECT LASTNAME, FIRSTNAME, NumOrdini
FROM imp1
where NumOrdini>=ALL(SELECT NumOrdini FROM imp1);
```

- Se si vogliono usare più table expression non bisogna ripetere WITH ma separarle con virgole

22

Table expressions

Oppure si può formulare la query direttamente nella clausola FROM (ma è meno leggibile)

```
SELECT <column_name_list> FROM (<query>);
```

```
SELECT LASTNAME, FIRSTNAME, NumOrdini
FROM (SELECT LASTNAME, FIRSTNAME, count(ORDERID) as NumOrdini
FROM NW.EMPLOYEES E join NW.ORDERS o on (e.EMPLOYEEID=o.EMPLOYEEID)
group by e.EMPLOYEEID, e.LASTNAME, e.FIRSTNAME
order by 3 desc)
where NumOrdini>=ALL(SELECT NumOrdini FROM (SELECT LASTNAME, FIRSTNAME,
count(ORDERID) as NumOrdini
FROM NW.EMPLOYEES E join NW.ORDERS o on (e.EMPLOYEEID=o.EMPLOYEEID)
group by e.EMPLOYEEID, e.LASTNAME, e.FIRSTNAME
order by 3 desc));
```

4. Selezionare il cliente che ha effettuato il maggior numero di ordini
5. Selezionare il cliente che ha generato il più alto introito nel 1996

23

ROWNUM

- ❑ La funzione ROWNUM restituisce l'ordinale di una riga in una tabella (la prima riga ha ROWNUM 1, ecc.)
- ❑ Se la tabella è ordinata si può filtrare su ROWNUM per selezionare il primo valore
- ❑ Selezionare l'impiegato che ha gestito il maggior numero di ordini

```
SELECT EMPLOYEEID, LASTNAME, FIRSTNAME, NumOrdini
FROM (SELECT e.EMPLOYEEID, e.LASTNAME, e.FIRSTNAME, count(ORDERID) as NumOrdini
FROM NW.EMPLOYEES E join NW.ORDERS o on (e.EMPLOYEEID=o.EMPLOYEEID)
group by e.EMPLOYEEID, e.LASTNAME, e.FIRSTNAME
order by 4 desc)
where rownum<=1;
```

```
WITH imp1 AS (SELECT LASTNAME, FIRSTNAME, count(ORDERID) as NumOrdini
FROM NW.EMPLOYEES E join NW.ORDERS o on (e.EMPLOYEEID=o.EMPLOYEEID)
group by e.EMPLOYEEID, e.LASTNAME, e.FIRSTNAME
order by 4 desc)
SELECT LASTNAME, FIRSTNAME, NumOrdini
FROM imp1
where rownum<=1;
```

24

Select Top K in Oracle

- ❑ **TOP K** (MSSql) e **Limit** (MySQL) non esistono in Oracle

```
select a, b from tabella where rownum<K order by b;
--non funziona perchè rownum è valutato prima di ordinare
```

- ❑ 2 Soluzioni (con Table expression):

```
select a, b
from (select a, b from tabella order by b)
where rownum<K;
```

```
select a, b
from (select a, b, rank() over (order by b) r from tabella)
where r<K;
```

- ❑ La funzione **RANK** restituisce l'ordine in un gruppo di valori (vedi anche **DENSE_RANK**)

25

Select Top K

- Selezionare i 10 ordini con valore più alto

```
select ORDERID, Totale from
(select ORDERID, sum (UNITPRICE*QUANTITY*(1-DISCOUNT)) as Totale
from NW.ORDERDETAILS
group by ORDERID
order by 2 desc)
where rownum<=10;
```

6 Selezionare i 5 clienti che hanno effettuato il maggior numero di ordini

26

CASE Statement

- Nella SELECT LIST:

```
CASE [ expression ]
WHEN condition_1 THEN result_1
...
WHEN condition_n THEN result_n
ELSE result
END
```

- Raggruppare i prodotti in fasce in base alle quantità vendute: fascia alta (>1000), fascia bassa (<500), fascia media

```
with PView as (SELECT PRODUCTID, sum (o.quantity) as QtaVenduta
FROM NW.ORDERDETAILS O
group by PRODUCTID)
select PRODUCTID, case
when QtaVenduta>1000 then 'fascia alta'
when QtaVenduta<500 then 'fascia bassa'
else 'fascia media'
end fascia, QtaVenduta
from PView;
```

7 Modificare le soglie in $(s*2/3)$, $(s*1/3)$, dove s è la quantità venduta dal prodotto più venduto.

27