

#### **Data Management and Visualization**

Politecnico di Torino

Data warehousing in Oracle – Practice 1			

The practice purpose is to write some queries, in extended SQL, to retrieve data from the data warehouse described in Point 1. Furthermore, it will be necessary to create some materialized views in order to improve the performance of frequent queries.

# 1. Problem specifications

A telephone company is interested in analyzing its own data to improve customer services. At present, the company has a database with call logs. For each call, the caller and receiver phone numbers, the duration, the type of charge (e.g., peak, off-peak rates), the start time (date, hour, minute, second) are known. The managers want to obtain very fast the information about the telephone traffic on the company lines and the daily income based on the caller location, the day and the phone rate.

In particular, the managers want to analyze the following situations:

- Monthly net income and number of calls for each caller city.
- Monthly net income and number of calls for each receiver city.
- Monthly net income and number of calls for each caller province and region.
- Monthly net income and number of calls for each receiver province and region.
- Daily net income and number of calls for each caller province.
- Yearly net income and number of calls for each caller province and region.
- Monthly net income and number of calls for each phone rate (type of charge).
- Net income and number of calls for each day of the week and phone rate.
- Daily number of calls for each caller region.
- Daily number of calls for each receiver region.

The OLTP database of the telephone company is reported in Table 1.

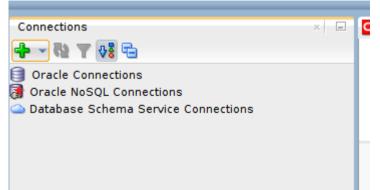
### 2. Querying the data warehouse

In Table 2 the proposed solution for the data warehouse is shown. The corresponding tables have already been created in Oracle and they contain some sample data. Use these tables as source for the following queries.

- 1. Select the yearly income for each phone rate, the total income for each phone rate, the total yearly income and the total income.
- 2. Select the monthly number of calls and the monthly income. Associate the RANK() to each month according to its income (1 for the month with the highest income, 2 for the second, etc., the last month is the one with the least income).
- 3. For each month in 2003, select the total number of calls. Associate the RANK() to each month according to its total number of calls (1 for the month with the highest number of calls, 2 for the second, etc., the last month is the one with the least number of calls).
- 4. For each day in July 2003, select the total income and the average income over the last 3 days.
- 5. Select the monthly income and the cumulative monthly income from the beginning of the year.
- Estimate the cardinality of the data warehouse tables (facts and dimensions) and decide whether and which materialized views are needed to improve performance of the previous queries.
- Create the materialized views you consider convenient and compare the execution plan cost of the queries using and without using the materialized views.

# 3. Connection to the database

1. Create a new connection at Oracle database



Use user SYSTEM and the password you specified (during the Oracle Database Express Edition installation or with the docker run command). Use a (connection) name you prefer.

New / Select Database Connection		×
Connection Name Connection Details	Name Connection 1 Database Type Oracle  User Info Proxy User	Color
	Authentication Type Default  Username SYSTEM Password	Ro <u>l</u> e default ▼ Sa <u>v</u> e Password
	Connection Type Basic   Details Advanced	
	Hostname localhost	
	Port 1521	
	SID xe     Service name	
Status :		
Help	Save Clear Test Conr	nect Cancel

2. Import database (CSV file)

Right click on Tables  $\rightarrow$  Import Data

<u>File Edit V</u> iew <u>N</u> avigate <u>R</u> un Tea <u>m</u> <u>T</u> ools <u>W</u> indow	w <u>H</u> elp
🔮 🗁 🗐 🗐 🔍 🕥 - 🕥 - 🤮 - 🐽	
Connections ×	O Welcome Page
🕂 - 🔞 🔨 🚱	▶ <b>■ 13 - 1</b> 3
Oracle Connections Conn Connections	Worksheet Que
● ₩ Vie ₩ New <u>T</u> able ● ₩ Vie ₩ Depen	
Import Data     Import Data     Import Using Oracle SQL Connector for H     Import Using Oracle SQL Connector for H	DFS
<ul> <li>⊕ - ⓐ Op ⓓ Refresh</li> <li>⊕ - ⓐ Qu</li> <li>♀ ▲ pply Filter</li> <li>♀ → ⓐ Qu</li> <li>Clear Filter</li> </ul>	Ctrl-R
⊕ 🚰 Tri ⊕- 📴 Typ ⊕ 🛐 Sequences	
Geographics     Geographi	
₽⊡ Synonyms ₽⊡ Public Synonyms ₽@ Database Links	
⊕-@ Public Database Links ⊕-@ Directories ⊕-@ Editions	,

Follow Wizard instructions to setup. Set Format with "csv" and insert csv file path.

🐻 Data Import Wizard - St	Step 1 of 5	×
Data Preview		
Data Preview     Import Method     Choose Columns     Column Definition     Finish	Source: Local File  File:  File Format File Format	Restore State
	Image: Construction of the second	
	Encoding: Cp 1252	
	Delimiter: , ▼ Line Terminator: standard: CR LF, CR or LF ▼	
	Left Endosyre: *  Right Endosure: *	
	File Contents	
	4	
Help	Eack Next > Einish	Cancel

Click on next and set table's name.

🔂 Data Import Wizard - S	tep 2 of 4		×
Import Method			
Data Preview Data Preview Column Definition Finish		r importing data. For External Table method, an external table will be created to read the data in the file. For Staging External Table method, an created as a staging table for importing the target table. For other methods, a new table is created and the data is imported. Insert Send Create Script to SQL Worksheet	
	Table Name:	Table 1	
	Import Row Limit:	100	
	File Contents		

Click next and import all attributes. Pay attention on KEY attributes, you have to set the type =

## INTEGER and must be NOT NULLABLE.

Source Data Columns	Target Table Columns
CONCORSO	Name ID
RISERVATO_A_STUDENTI_DI	
TIPO_CDL	Data Type INTEGER 🔹
SEDE_ATTIVITA'	Nullable? Default
STRUTTURA/CORSO_PRESSO_CUI_SI_SVOLGE	Comment
POSTI	Commerc
RIPARTIZIONE_DELLE_BORSE_	
IMPORTO_ORARIO_	
ATTIVITÀ_RICHIESTE_AL_BORSISTA	
CONDIZIONI,_REQUISITI_O_CONOSCENZE_PAR	
DATA_DI_INIZIO_DELLE_DOMANDE_	
Click to End.	

3. Execute the query and create the materialized views by means of the interface.

Tables			Description
DWABD.PHONERATES			
(			Different phone rates
phoneRateType	INT	NOT NULL,	
phoneRateName	VARCHAR (20)	NOT NULL,	7 rows
phoneRate_CostPerSecond	FLOAT	NOT NULL,	1 10000
PRIMARY KEY(phoneRateType	e)		
);			
DWABD.PLACES			
(			Places
Places ID	INT	NOT NULL,	
_ City	VARCHAR (20)	NOT NULL,	1500 rows
Province	VARCHAR (20)	NOT NULL,	
Region	VARCHAR (20)	NOT NULL,	
PRIMARY KEY(Places ID)			
);			
DWABD.CALLS			
(			Calls in 2003 and 2004
` CallerPhoneNumber	VARCHAR(20)	NOT NULL,	
ReceiverPhoneNumber	VARCHAR (20)	NOT NULL,	~ 1300000 rows
CallerLocation	INT	NOT NULL,	
ReceiverLocation	INT	NOT NULL,	
FullDate	DATE	NOT NULL,	
StartTimeHour	INT	NOT NULL,	
StartTimeMinute	INT	NOT NULL,	
StartTimeSecond	INT	NOT NULL,	
CallDuration	FLOAT	NOT NULL,	
phoneRateType	INT	NOT NULL,	
PRIMARY			
KEY(CallerPhoneNumber, Rece	iverPhoneNumber	,FullDate,StartTime	Hour
,StartTimeMinute,StartTimeS	Second),		
FOREIGN KEY(phoneRateType	e) REFERENCES		
PhoneRates (phoneRateType)	ON DELETE CASCA	DE,	
FOREIGN KEY(CallerLocatio	on)REFERENCES P	laces(Places_ID) ON	
DELETE CASCADE,			
FOREIGN KEY(ReceiverLocation) REFERENCES Places(Places_ID) ON			ON
DELETE CASCADE			
);			

Table 1 – Source data base with single call information

Tables	Description
DWABD.TIMEDIM	
(	Time dimension
ID time INT NOT NULL,	
DayOfWeek CHAR(15) NOT NULL,	10 rows
DateMonth INT NOT NULL,	
DateYear INT NOT NULL,	
PRIMARY KEY(ID_time)	
);	
DWABD.PHONERATE	
(	Phone rate
ID phoneRate INTEGER NOT NULL,	dimension
ID_phoneRate INTEGER NOT NULL, phoneRateType VARCHAR(20) NOT NULL,	
PRIMARY KEY(ID_phoneRate)	7 rows
);	
DWABD.LOCATION	
(	Place
ID_location INTEGER NOT NULL,	dimension
City VARCHAR(20) NOT NULL,	
Province CHAR(20) NOT NULL,	1500 rows
Region CHAR(20) NOT NULL,	
PRIMARY KEY(ID location)	
);	
DWABD.FACTS	Fact table
(	
ID_time INTEGER NOT NULL, ID_phoneRate INTEGER NOT NULL,	7809 rows
ID phoneRate INTEGER NOT NULL,	
ID_location_Caller INTEGER NOT NULL,	
ID_location_Receiver INTEGER NOT NULL,	
Price FLOAT NOT NULL,	
NumberOfCalls INTEGER NOT NULL,	
PRIMARY	
<pre>KEY(ID_time,ID_phoneRate,ID_location_Caller,ID_location_Receiver),</pre>	
FOREIGN KEY(ID_time) REFERENCES timeDim(ID_time),	
FOREIGN KEY(ID_phoneRate) REFERENCES phoneRate(ID_phoneRate),	
FOREIGN KEY(ID_location_Caller) REFERENCES location(ID_location),	
FOREIGN KEY(ID_location_Receiver) REFERENCES location(ID_location)	
);	

 Table 2 – Proposed solution - Data warehouse tables