Introduction to Oracle

Architecture

- Client-server system
- Server: SERVEDB,
  - Internal addess (from the lab) 192.168.0.252
  - External address (from home with OpenVPN) 10.17.2.91
- Client:
  - Web interface:
    - http://192.168.0.252:8080/apex
  - Command line: sqlplus

Architecture

- A server can contain one or more databases
- Each database can have one or more instances
  - An instance is a set of processes that access the data of the database
  - Different from SQL Server/DB2
- Objects are contained in databases: table, views, stored procedures
- Objects are divided into schemas
  - One schema per user with the name of the user
  - No other schemas (different from SQL Server/DB2)
- On server SERVEDB only one database, only one instance
  - Oracle Express allows only one database per server
  - Accessible at http://10.17.2.91:8080/apex

Help

- Local help:

http://192.168.0.252:8080/apex/wwv_flow_help.show_help?p_flow_id=4500&p_step_id=1000

- Web help:

http://www.oracle.com/pls/xe102/homepage

http://www.oracle.com/pls/db102/homepage

Connecting

- Launch browser
- Go to http://192.168.0.252:8080/apex
- Insert:
  - Username: hr
  - Password: Infonew1

Browsing Objects

- On the Database Home Page, click the Object Browser icon.
- The Object Browser page displays with two sections:
- The Object Selection and the Detail pane
- Click the EMPLOYEES table in the Tables object list to display information about the structure of that table.

Connecting with the command line

- Run SQL Command Line from the start menu or start a command window and type:

```
SQLPLUS /NOLOG
CONNECT username/password@[//]host[:port][/service_name]
```

- // is optional
- host is the host name or IP address of the computer that is running Oracle Database XE
- port (optional) is the TCP port number on which the Oracle Net listener is listening. If not specified, the default port number 1521 is assumed.
- service_name (optional) is the name of the database service to which to connect. For Oracle Database XE, the service name is XE. If service_name is omitted, Oracle Database XE Client appends a request for the default database service, which is configured during installation as XE.

Connecting with the command line

- Example:

```
CONNECT hr/Infonew1@192.168.0.252
```

- DISCONNECT to disconnect

Issuing SQL Commands

SELECT EMPLOYEE_ID,FIRST_NAME,LAST_NAME FROM EMPLOYEES;

```
EMPLOYEE_ID FIRST_NAME           LAST_NAME
----------- -------------------- -------------------------
        100 Steven               King
        101 Neena                Kochhar
        102 Lex                  De Haan
        103 Alexander            Hunold
        104 Bruce                Ernst
        105 David                Austin
        106 Valli                Pataballa
        107 Diana                Lorentz
        108 Nancy                Greenberg
```

Ending a SQL Command

- You can end a SQL command in one of three ways:
  - with a semicolon (;)
  - with a slash (/) on a line by itself
  - with a blank line
- A semicolon (;) tells SQL*Plus that you want to run the command. Type the semicolon at the end of the last line of the command and press Return. SQL*Plus processes the command and also stores the command in the SQL buffer.
- A slash (/) on a line by itself also tells SQL*Plus that you wish to run the command. SQL*Plus executes the command and stores it in the buffer.
- A blank line in a SQL statement or script tells SQL*Plus that you have finished entering the command, but do not want to run it yet.

Authentication

- Oracle uses user accounts separated from those of the operating system
- A user account is identified by a user name and defines the user's attributes, including the following:
  - Password for database authentication
  - Privileges and roles
  - Default tablespace for database objects
  - Default temporary tablespace for query processing work space
- When you create a user, you are also implicitly creating a schema for that user.

Privileges

- There are two main types of user privileges:
  - System privileges—A system privilege is the right to perform a particular action, or to perform an action on any schema objects of a particular type. For example, the privileges to create tables and to delete the rows of any table in a database are system privileges.
  - Object privileges—An object privilege is a right to perform a particular action on a specific schema object.

Roles

- Managing and controlling privileges is made easier by using roles, which are named groups of related privileges.
- You create roles, grant system and object privileges to the roles, and then grant roles to users.
- Unlike schema objects, roles are not contained in any schema.

Predefined Roles

- CONNECT: Enables a user to connect to the database.
- RESOURCE:
  - Enables a user to create certain types of schema objects in his own schema.
  - Grant this role only to developers and to other users that must create schema objects.
- DBA: Enables a user to perform most administrative functions, including creating users and granting privileges; creating and granting roles; creating and dropping schema objects in other users' schemas; and more.

Internal User Accounts

- Certain user accounts are created automatically for database administration.
- Examples are SYS and SYSTEM.
- These automatically created accounts are called internal user accounts, and their schemas are called internal schemas.

Internal User Accounts

- SYSTEM: This is the user account that you log in with to perform all administrative functions other than starting up and shutting down the database.
- SYS:
  - All base tables and views for the database data dictionary are stored in the SYS schema.

Authentication

- Oracle uses OS authentication for starting and stopping the database
- Starting and shutting down: to be done on the server computer logged as a member of a special grounp with a command from the menu or from the command line
- Windows: ORA_DBA user group. Tiplically Administrator is part of it
- Linux: dba user group
- OS authentication is needed because there must be a way to identify administrative users even if the database is shut down.

Names

- Every database object has a name. In a SQL statement, you represent the name of an object with a quoted identifier or a nonquoted identifier.
  - A quoted identifier begins and ends with double quotation marks ("). If you name a schema object using a quoted identifier, then you must use the double quotation marks whenever you refer to that object.
  - A nonquoted identifier is not surrounded by any punctuation.

Names

- The following list of rules applies to both quoted and nonquoted identifiers unless otherwise indicated:
- Names must be from 1 to 30 bytes long with this exception:
  - Names of databases are limited to 8 bytes.

Names

- Nonquoted identifiers cannot be Oracle reserved words.
- Nonquoted identifiers must begin with an alphabetic character from the database character set.
- Quoted identifiers can begin with any character.

Names

- Nonquoted identifiers can contain only alphanumeric characters from your database character set and the underscore (_), dollar sign ($), and pound sign (#).
- Quoted identifiers can contain any characters and punctuations marks as well as spaces.
- Neither quoted nor nonquoted identifiers can contain double quotation marks or the null character (\0).
- Nonquoted identifiers are not case sensitive. Oracle interprets them as uppercase.
- Quoted identifiers are case sensitive.

Names

- Within a namespace, no two objects can have the same name.
- Because tables and views are in the same namespace, a table and a view in the same schema cannot have the same name.
- However, tables and indexes are in different namespaces. Therefore, a table and an index in the same schema can have the same name.
- Each schema in the database has its own namespaces for the objects it contains. This means, for example, that two tables in different schemas are in different namespaces and can have the same name.

Names Examples

- The following examples are valid schema object names:
  - last_name_horse
  - "EVEN THIS & THAT!"
  - a_very_long_and_valid_name
- Invalid:
  - a_very_very_long_and_valid_name

Naming Guidelines

- Use the same names to describe the same things across tables.
- For example, the department number columns of the sample employees and departments tables are both named department_id.

Case Sensitiveness

- Case is insignificant in reserved words, keywords, identifiers and parameters.
- However, case is significant in text literals and quoted names.

Physical Database Structures

- Every database has one or more physical datafiles. The datafiles contain all the database data.
- The characteristics of datafiles are:
  - A datafile can be associated with only one database.
  - Datafiles can have certain characteristics set to let them automatically extend when the database runs out of space.

Physical Database Structures

- Every Oracle database has a control file.
- A control file contains entries that specify the physical structure of the database.
- For example, it contains the following information:
  - Database name
  - Names and locations of datafiles and redo log files
  - Time stamp of database creation

Physical Database Structures

- Every Oracle database has a set of two or more redo log files.
- The set of redo log files is collectively known as the redo log for the database.
- To protect against a failure involving the redo log itself, Oracle allows a multiplexed redo log so that two or more copies of the redo log can be maintained on different disks

Logical Database Structures

- A database is divided into logical storage units called tablespaces, which group related logical structures together.
- For example, tablespaces commonly group together all application objects to simplify some administrative operations.

Tablespaces

- Each database is logically divided into one or more tablespaces.
- One or more datafiles are explicitly created for each tablespace to physically store the data of all logical structures in a tablespace.
- The combined size of the datafiles in a tablespace is the total storage capacity of the tablespace.

Tablespaces

- A bigfile tablespace is a tablespace with a single, but very large (up to 4G blocks) datafile.
- Smallfile tablespaces, in contrast, can contain multiple datafiles, but the files cannot be as large.
- Bigfile tablespaces let Oracle utilize the ability of 64-bit systems to create and manage ultralarge files.
- In this way Oracle can scale up to 8 exabytes in size.

Logical Database Structures

- Every Oracle database contains a SYSTEM tablespace and a SYSAUX tablespace.
- Created automatically when the database is created.
- The system default is to create a smallfile tablespace
- The SYSTEM and SYSAUX tablespaces are created as smallfile tablespaces.

Database Creation Example

```
CREATE DATABASE mynewdb
USER SYS IDENTIFIED BY pz6r58
USER SYSTEM IDENTIFIED BY y1tz5p
--The password for user SYS is pz6r58 and the password for SYSTEM is y1tz5p.
LOGFILE GROUP 1 ('/u01/oracle/oradata/mynewdb/redo01.log') SIZE 100M,
GROUP 2 ('/u01/oracle/oradata/mynewdb/redo02.log') SIZE 100M,
GROUP 3 ('/u01/oracle/oradata/mynewdb/redo03.log') SIZE 100M
MAXLOGFILES 5 MAXLOGMEMBERS 5 MAXLOGHISTORY 1
--The new database has three redo log files
-- MAXLOGFILES, MAXLOGMEMBERS, and MAXLOGHISTORY define limits for the redo log.
```

Database Creation Example

MAXDATAFILES 100

-- MAXDATAFILES specifies the maximum number of datafiles that can be open in the database.

MAXINSTANCES 1

-- only one instance can have this database mounted and open.

CHARACTER SET US7ASCII

NATIONAL CHARACTER SET AL16UTF16

--The AL16UTF16 character set is used to store data in columns specifically defined as NCHAR, NCLOB, or NVARCHAR2.

Database Creation Example

DATAFILE '/u01/oracle/oradata/mynewdb/system01.dbf' SIZE 325M REUSE EXTENT

MANAGEMENT LOCAL

--The SYSTEM tablespace, consisting of the operating system file /u01/oracle/oradata/mynewdb/system01.dbf is created as specified by the DATAFILE clause. If a file with that name already exists, it is overwritten.

Database Creation Example

SYSAUX DATAFILE '/u01/oracle/oradata/mynewdb/sysaux01.dbf' SIZE 325M REUSE

-- A SYSAUX tablespace is created, consisting of the operating system file /u01/oracle/oradata/mynewdb/sysaux01.dbf

DEFAULT TABLESPACE tbs_1

--Clause creates and names a default permanent tablespace for this database.

Data Types

- Character datatypes
- Numeric datatypes
- Date and time (date-time) datatypes
- Large Object (LOB) datatypes

Character datatypes

- The VARCHAR2 datatype stores variable-length character literals.
- Length between 1 and 4000 bytes
- The CHAR datatype stores fixed-length character literals.
- Length: between 1 and 2000 bytes
- NCHAR and NVARCHAR2 datatypes store Unicode character data.

Numeric

- NUMBER: as DECIMAL in SQL Server and DB2
- BINARY_FLOAT, BINARY_DOUBLE: store floating-point data in the 32-bit IEEE 754 format and the double precision 64-bit IEEE 754 format respectively.
- Compared to the NUMBER datatype, arithmetic operations on floating-point data are usually faster for BINARY_FLOAT and BINARY_DOUBLE.
- High-precision values require less space when stored as BINARY_FLOAT and BINARY_DOUBLE datatypes. .

Date and Time

- DATE: century, year, month, day, hours, minutes, and seconds.
- TIMESTAMP: values that are precise to fractional seconds
- TIMESTAMP WITH TIME ZONE: datatype can also store time zone information

Large Objects (LOBs)

- A set of datatypes that are designed to hold large amounts of data.
- A LOB can hold up to a maximum size ranging from 8 terabytes to 128 terabytes
- The BLOB, CLOB, and NCLOB datatypes are internal LOB datatypes and are stored in the database.
- The BFILE datatype is the only external LOB datatype and is stored in an operating system file, outside the database.

Creating a Table with the Object Browser

- Log in with:
  - Username: Utente
  - Password: Infonew1
- On the Database Home Page, click the Object Browser icon.
- In the object list under Create, select Table.
- Table name: personal_info_NUMERO_DI_MATRICOLA

Creating a Table with the Object Browser

- Columns

employee_id	NUMBER(6,0)	NOT NULL
birth_date	DATE	NOT NULL
social_security_id	VARCHAR2(12)	NOT NULL
marital status	VARCHAR2(10)	
dependents_claimed	NUMBER(2,0)	
contact_name	VARCHAR2(45)	NOT NULL
contact_phone	VARCHAR2(20)	NOT NULL
contact_address	VARCHAR2(80)	NOT NULL

Creating a Table with the Object Browser

- After you have entered the column information, click Next.
- On the Primary Key page, do not create a key at this time. Click the Next button
- On the Foreign Key page, do not create a key at this time. Click the Next button.
- On the Constraints page, do not create a constraint at this time. Click the Finish button.
- On the Create Table page, click the SQL button to view the SQL statements that produce the table. This option shows the statement even if it is incomplete. You need to complete your input to see the complete SQL statement when using this option.
- Click the Create button to create the table.

Adding a Column To a Table

- You can use Object Browser to add columns to a table.
- On the Database Home Page, click the Object Browser icon.
- In the Object list, select Tables and then click the personal_info table that you previously created.
- Click Add Column.

Altering a Table

- Modifying a Column In a Table
- Dropping a Column From a Table
- Adding a Check Constraint
- Adding a Unique Constraint
- Adding a Primary Key Constraint
- Adding a Foreign Key Constraint
- Similar