

# I. The Oracle database management system

Revision 15

*Authors: Kardkovács Zsolt, Győr Ferenc, Marton József, Zsombor Paróczy*

<b>I.</b>	<b>THE ORACLE DATABASE MANAGEMENT SYSTEM.....</b>	<b>1</b>
1.	WHY ORACLE? .....	2
2.	THE HISTORY OF ORACLE .....	2
3.	ORACLE BUILDING BLOCKS.....	2
3.1.	<i>Logical structure.....</i>	<i>3</i>
3.2.	<i>Physical structure.....</i>	<i>7</i>
3.3.	<i>The connection between physical and logical structure.....</i>	<i>8</i>
3.4.	<i>How the system works.....</i>	<i>9</i>
3.5.	<i>Security in Oracle.....</i>	<i>11</i>
4.	ORACLE ADMINISTRATION .....	12
4.1.	<i>Starting SQL Developer.....</i>	<i>12</i>
4.2.	<i>DBA mode – Database Configuration.....</i>	<i>13</i>
4.3.	<i>Reports mode - Sessions.....</i>	<i>13</i>
4.4.	<i>Reports mode - Locks.....</i>	<i>13</i>
4.5.	<i>Connections mode.....</i>	<i>13</i>
4.6.	<i>DBA mode – Security.....</i>	<i>14</i>
4.7.	<i>DBA mode – Storage.....</i>	<i>14</i>
5.	OTHER ORACLE PRODUCTS .....	14
6.	APPENDIX.....	15
6.1.	<i>Comparing newer Oracle versions.....</i>	<i>15</i>
6.1.1.	<i>Oracle8.....</i>	<i>15</i>
6.1.2.	<i>Oracle8i.....</i>	<i>15</i>
6.1.3.	<i>Oracle9i.....</i>	<i>15</i>
6.1.4.	<i>Oracle10g.....</i>	<i>16</i>
6.1.5.	<i>Oracle11g.....</i>	<i>16</i>

*„I am Sir Oracle,  
And when I open my lips, let no dog bark!”  
(Shakespeare: The Merchant of Venice)*

## **1. Why Oracle?**

There is a broad variety of different technology-based products available at the database management systems' market. The present workshop is aiming to familiarize the students with some of the most important parts of up-to-date database management, and to present them with useful practical knowledge despite our limited resources. In order to achieve this goal, the workshop will feature the most widespread and dominant product on the Hungarian market: Oracle. Herein we include of list of other significant and supported database management systems (and their manufacturers): SQL Server (relational, Microsoft), DB/2, Informix (relational, IBM), Ingres II (relational, Computer Associates), Sybase Adaptive Server (relational, Sybase), *MySQL*<sup>1</sup> (relational, Oracle Corporation, GNU GPL), O2 (object-oriented, O2 Inc.), GemStone (object-oriented, GemStone Inc.), ObjectStore (object-oriented, Object Design). Enthusiastic students who would like to hear about the more interesting features of Oracle and other database management systems are encouraged to contact the Department of Telecommunications and Media Informatics.

During the workshops an Oracle 11gR2<sup>2</sup> version server and Windows XP client tools will be used.

## **2. The history of Oracle**

As relational database management systems emerged at the beginning of the 70's the CIA started a project with the ambitious goal of creating a knowledge base that provides “the answer for every question”. It run under the code names “revelation, prophecy” and was finally named “Oracle”. Unfortunately the project was stopped due to lack of resources, however three expert project team members: Larry Ellison (now the CEO), Bob Miner and Ed Oates founded the Software Development Laboratories company in 1977, which is called Oracle Corporation since 1983. Shortly afterwards its foundation the company presented their first Oracle database manager (actually, CIA was their first customer). Nowadays Oracle is the most widespread database management system, which can be accessed and installed on every major operating system and hardware architecture.

## **3. Oracle building blocks**

Oracle is a relational (in fact, object-relational) database management system which is maintained and operated by a standard language called SQL (SQL:2008 to be precise). The handling of Oracle is greatly simplified by the services of the Oracle SQL Developer, which will be introduced during the first workshop. The query part of SQL will be the subject of the second workshop. Since SQL will be more actively used later on, there will be a thorough SQL assessment at the beginning of the next workshop.

These are the most important features of the Oracle database management system:

- Basically it features client-server architecture.
- Depending on the OS Oracle enables a multitasking, multi-user behavior and parallel access to data.
- Can act as spatially distributed system.

---

<sup>1</sup> MySQL was the product of the company MySQL AB, which was first acquisition by Sun Microsystems (2008), than later the Oracle Corp. (2010).

<sup>2</sup> R2 means Release 2 in this context, it's the second release of the database management system

- Supports all the main network protocols and operating systems.
- Supports all phases of software and application development.
- Co-operates with major interpreters and development environments.
- Manages practically arbitrary amount of data (with varying efficiency, though).
- Guarantees 24/7 availability and secure operation.
- Provides high-level data security, integrity and consistency.
- Enables storing complex structures (objects, multimedia data, procedures), as well.
- Provides advanced system administration by using Oracle Management Server and related Agents, meaning that an arbitrarily sized database-pool administration / remote supervision can be conducted using the Enterprise Management Console.
- Oracle as a company provides reliable product support for its customers.

In the present document by server we always mean a database and a server instance. The database stores the user and system data, while the server instance consists of the processes (threads) required to run the service. One server computer can host more than one database. The highest-level storage unit identified by a name is a database, which is in fact more than just a massive amount of data. The instance defines the server behavior, its operations.

A database can be separated into a physical and a logical part, which can be individually maintained. In the followings these two parts, and the connections and dependencies between them will be described in more details.

### 3.1. Logical structure

Databases are divided into tablespaces, which serve as logical storage units. A tablespace is the biggest logical storage unit. Data with similar behavior and maintenance need is usually stored in a tablespace. Tablespaces provide means to manage the database data storage; to distribute data between tablespaces, to define quotas for users etc. In every case there is a tablespace called **system**. Of course more than one tablespace might be created; moreover, it is recommended to store user and system data separately. Therefore, a typical system contains the following tablespaces:

- **system**: tablespace containing system information (such as data dictionary),
- **sysaux**: extended tablespace, which was introduced in 10g. Some of the functionalities moved to this tablespace from other tablespace. These functionalities includes LogMiner (can be used to parse database logs) and Oracle Data Mining (data mining utility) and many others.
- **rbs**: in the versions before Oracle 9 this tablespace contains the logs and results of database operations (for more information please refer to the chapter describing the rollback segments). In the later versions this role is filled by the **undo** tablespace.
- **temp**: the tablespace for storing all temporary data (e.g. partial result sets of queries, sorting results),
- **tools**: the tablespace storing the patterns used by applications and general tools (e.g. the Form Builder),
- **users**: an example of a general user tablespace.

Tablespaces contain system or user objects, also known as units. These tablespace objects might be tables, views, sequences, synonyms, indexes, groups, clusters, database links etc. and complex structures created based on these units, such as data dictionary, (Oracle) schema, and (executable) stored procedures.

One object belongs to one tablespace only, but complex structures might spread across more tablespaces.

The most important user objects are the followings:

**Table:** the base unit of logical data storage, which consists of rows and columns. A table can be mapped into a relation. A table can be unambiguously identified by its name (e.g. SCOTT.EMP). Columns are characterized by their name, type and size. A table can store only one type of data record. The order of the rows is irrelevant. When creating a table, a table name, the column names, types and sizes must be defined. The figure below shows a table and the data stored in the table.

	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7329	SMITH	CLERK	7902	17-DEC-88	800.00	300.00	20
7499	ALLEN	SALESMAN	7698	20-FEB-88	1600.00	300.00	30
7521	WARD	SALESMAN	7698	22-FEB-88	1250.00	500.00	30
7566	JONES	MANAGER	7839	02-APR-88	2975.00		20

**1. Figure: Table**

**View:** a user object that enables displaying data gathered from one or more tables. It is similar to a stored query; it does not either contain or store data (physically), only derives data from tables on which it was defined. These tables are called the base tables or master tables of the view.

**Sequence:** continuous and unique number generator generally used to create unique identifiers. Please note, that the sequence value is not transaction-oriented, meaning that for example a rollback operation does not modify the sequence value.

**Synonym:** can be used to assign more than one names to a table, a view or a sequence. Basically, it can be used to shorten or to make transparent the storage place of an object. A synonym can be either public or private. A public synonym can be used by everyone, while a private one is accessible for only a certain group of users. A user needs special authorization to create and delete public synonyms.

**Index:** a great tool to increase data access speed - in Oracle it is essentially a B\* tree. It is recommended to create an index for table column(s) that are frequently used in queries. Indexes are automatically created for those columns that are marked as unique during table creation. Indexes are automatically refreshed by the system. An index can be complex, meaning that the index consists of more fields. In case of such complex indexes high-cardinality columns should be at the beginning of the attribute list. In certain cases (in critical, resource-demanding environments) index creation requires precise design work, since an inappropriate choice of indexes can slow down the system.

**Group:** a structure containing more servers in Oracle 9i. Groups are used to facilitate system administration.

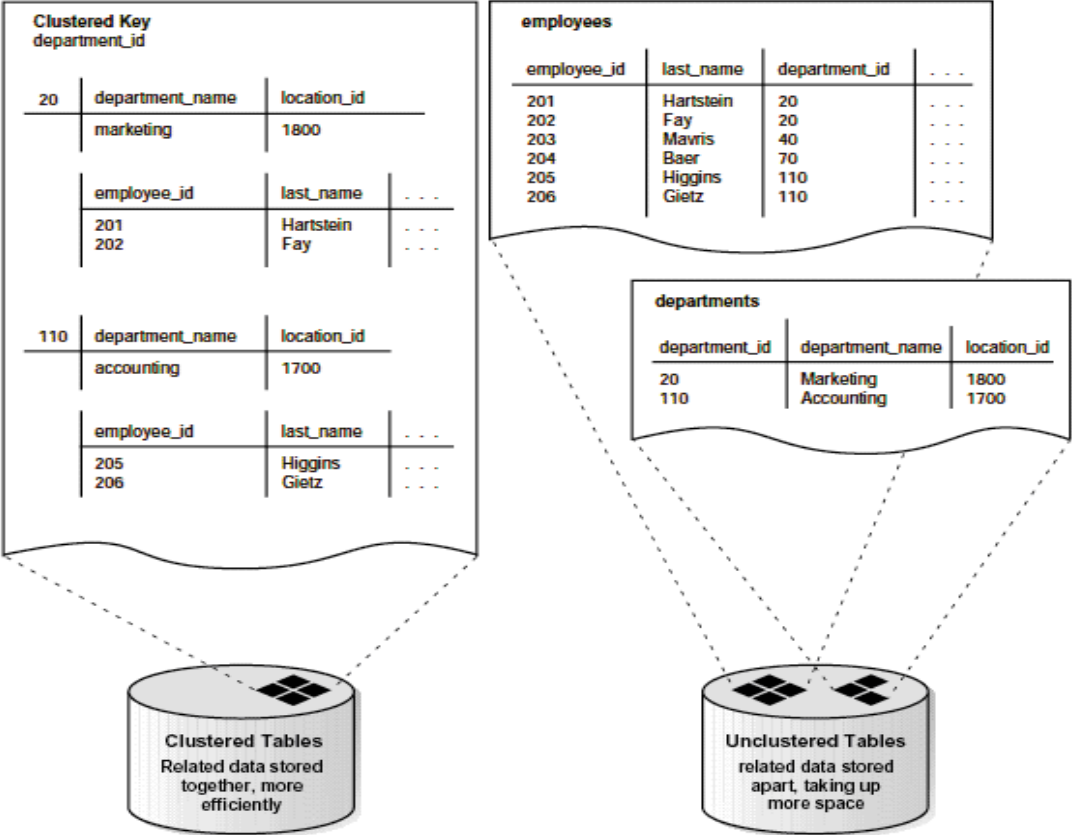
**Database Link:** a special synonym used for addressing databases. It has already been noted that a server computer might contain multiple databases, moreover there can be distributed databases stored on different computers which are data-wise connected. In these cases it might be required to define database link

**Data Dictionary:** a central set of read-only reference tables and views containing the actual state of the system. It contains all information on the users currently logged into the system, on the object accessible by these users; the constraints related to each field; column properties; how much storage space is occupied by the objects, how much space is free; action logs, change logs etc.

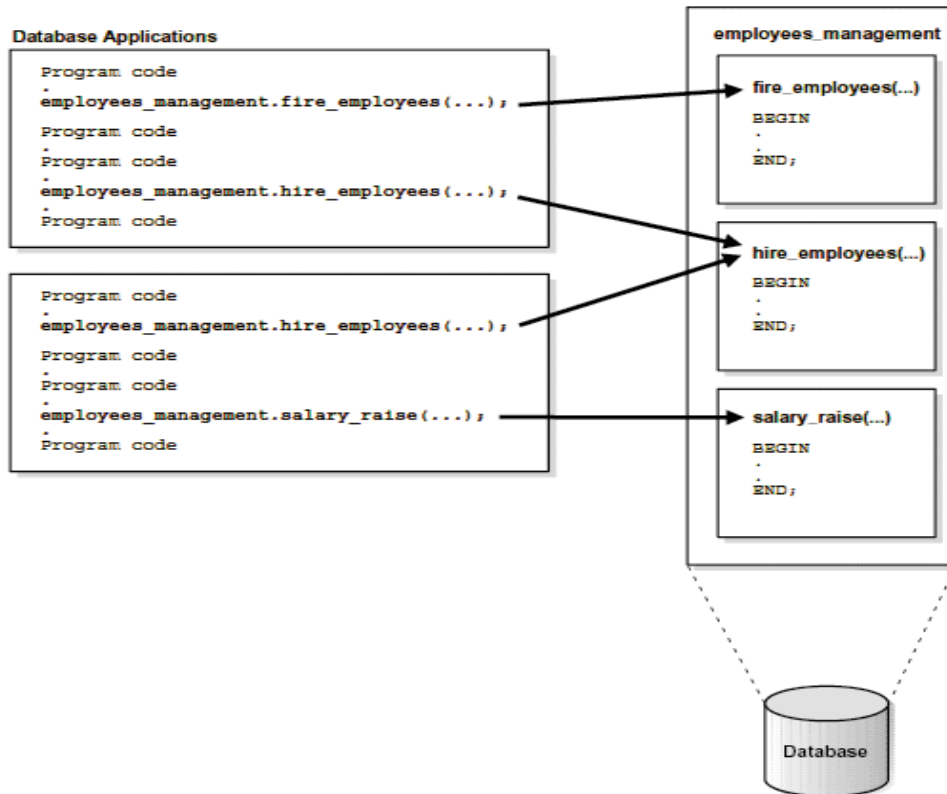
*Schema:* the set of objects accessible by a given user is called the user's schema. There is a one-to-one relation between a user and his or her schema. Object access rights might be constrained by authorization system settings, which can refer to a certain user or groups of users.

*Cluster:* it is recommended to store data requiring similar handling mechanisms or access methods in one single group, one physical location. Data access times can significantly decrease upon reading from disk if related data are stored physically "close". Therefore organizing data into groups increases efficiency. Figure 2 shows a typical example for this.

*Stored Procedures:* a set of executable objects stored in the database. Executable objects (e.g. PL/SQL, Java or in case of appropriate settings other source language programs / program segments: particularly those that are widely used) that run on the server can be stored in the database tablespaces. When setting up the Oracle system several "built in" stored procedures are installed that e.g. facilitate system administration, development. See Figure 3.



2. Figure: Clustered (left side) and un-clustered data management (right side)



**3. Figure: Stored procedures are stored in the database and are run on the server.**

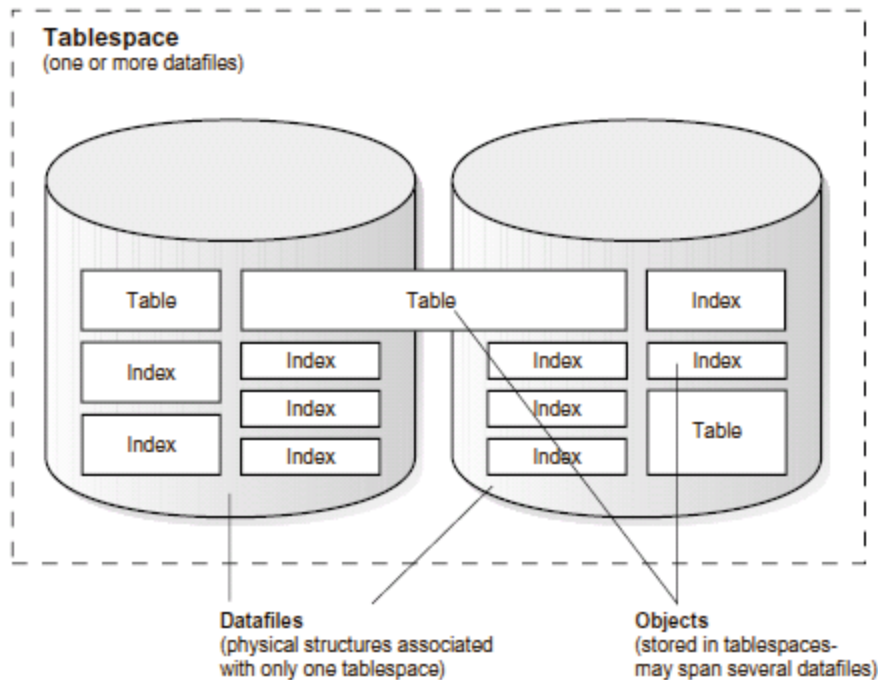
As mentioned before, individual objects consist of basic data types. Oracle features the following data types:

Data type	Short description
CHAR ( <i>n</i> )	Fixed-sized string type. Its size is fixed, meaning that the system always allocates the predefined length, regardless of the string taking up all the available space or not. If the inserted text does not fill up the space entirely, the database manager inserts trailing spaces automatically. (This has to be considered in the queries, as well!) From Oracle8 onwards the maximum size of the type is 2 kilobytes. It has to be noted that by default the maximum size of a given data type is always given in bytes and not number of characters, since there are character sets consisting of multiple byte long characters (e.g. UTF8). If the length of the data type must be defined in character numbers nonetheless, the number indicating the length must be followed by the word "char". If the length of a given type field is not specified otherwise, it is defaulted 1 by the system.
VARCHAR2 ( <i>n</i> )	Variable length string type. Unlike in the case of type CHAR, Oracle only allocates the space that is effectively used thus using VARCHAR is more reasonable and results in higher efficiency. From Oracle8 onwards the maximum size of the type is 4 kilobytes. The default size is 1.
NCHAR ( <i>n</i> ) NVARCHAR2 ( <i>n</i> )	The NCHAR and NVARCHAR2 are similar to CHAR and VARCHAR2 data types, but always store Unicode strings. They also have the same size limit. In this data type only Unicode strings can be stored, the maximum size must be specified in characters.
CLOB (LONG)	A data type intended to store a huge amount of text. In case the size of the character string to be stored is bigger than the maximum values mentioned above (no upper limit has to be defined) the type of the respective field should be CLOB (Character type Large Object). CLOB has a maximum size of 4 gibibytes, but that is big enough. Due to downward compatibility requirements with previous versions

	the LONG type is still there, but is implemented as a CLOB. However, these two data types are not the same. Some of the most significant differences: a schema object can contain only one LONG type, but an arbitrary number of CLOBs; the maximum size of type LONG is limited to 2 gibibytes; LONG supports sequential access, while CLOB can be random access, as well.
NUMBER ( <i>s, p</i> )	Data type to display an arbitrary number. Can be used for both fixed and floating point number representation. The range is between $\pm 1 \cdot 10^{-130}$ and $\pm 9,9 \dots 99 \cdot 10^{125}$ including 0 of course. In case of NUMBER type there are two size parameters ( <i>p, s</i> ) to be defined. <i>P</i> is the precision, or the maximum number of significant decimal digits, where the most significant digit is the left-most nonzero digit, and the least significant digit is the right-most known digit. <i>S</i> is the scale, or the number of digits from the decimal point to the least significant digit. For example when displaying a number with a maximum value of $\pm 999,99$ with two fractional digits precision the required data type is NUMBER (5, 2).
DATE	Data type for storing and displaying dates. Oracle is able to store all dates within the range of January 1st, 4713 BC and December 31st, 9999 AD. The date consists of seven fields: century, year, month, day, hour, minute, second. There are several derived units available, such as day of the week, week of the year etc. There are several other calendar systems (such as Persian, Hebrew, Chinese, Hindu, Julian) available in Oracle, moreover the dates can be converted from one calendar system to another. For more information on date handling please refer to the Appendices of the present document.
ROWID	Unique logical and physical identifier of the records. All tables have a pseudo column called ROWID. The ROWID values identify the table rows unambiguously. Generally it contains physical location and access information in hexadecimal form. This hexadecimal coded type is called ROWID.
UROWID	The UROWID type is a unique logical identifier for records, whose physical location depend on operation on other records or depend on parameters outside the Oracle's scope. In these tables, the ROWID pseudo column is of UROWID type.  In index-organized tables (IOT) the data records are stored in the leaves of the index tree, which can be moved (to other physical block) during an insert or delete operation. In IOT-s the logical record identifier of UROWID type is guaranteed to be unchanged as long as the value for the primary key is left unchanged.  ROWID in tables stored outside the Oracle database are also of UROWID type.

### 3.2. Physical structure

**Data file.** Oracle places tablespace data into one or more data files, but a data file can belong to only one tablespace – and, hence to only one database (see Figure 4). Data to be stored does not get into the database the moment the operation is executed, since batch data output is more efficient - as it has been mentioned in subjects discussing file system management (such as Operating Systems, Computer Architectures).



4. Figure: The connection between logical and physical data storage

**Redo-log file.** Generally there are at least two or more redo-log files (for more information please refer to the subject "Databases") in an Oracle database. This set of files is actually referred to as the redo-log file of the database. Its primary task is to store the database operations until the data output is successfully completed. At least two redo-log files are required, since while the contents of the redo log buffer is written to one of these files, the other is available for other database processes (eg. ARCn). Since the functionality of redo-log files bears critical importance in terms of data security Oracle supports the usage of multiplexed redo-logs.

**Control file.** The control file contains information on the physical structure of the database. Information such as the name of the database, the name and physical location of data files, the location of the redo-log files, the date of the database creation etc. (for more information please refer to the introduction of the SQL Developer DBA mode). One control file is enough, but for security reasons it might be copied and distributed between different storage areas.

### 3.3. The connection between physical and logical structure

The logical units of physical data storage consist of three elements. The smallest unit is the data block (logical block, Oracle block). A data block represents a fixed sized, continuous storage area with a size that is a multiple of the block size (typically 512-4096 bytes) of the operating system or the partition (on the disk). Data blocks are important in terms of data output and input, since the data block is the smallest unit handled by Oracle as a whole.

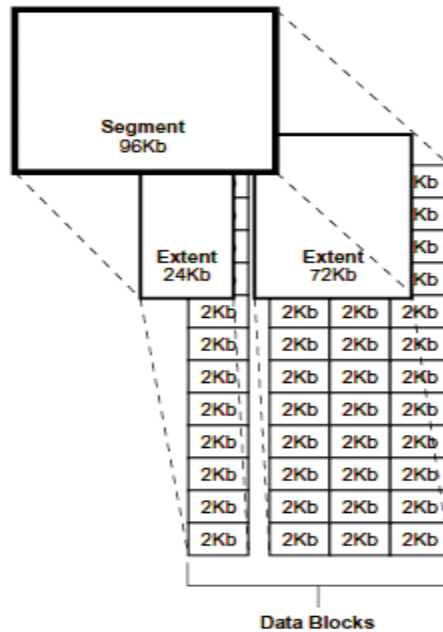
An extent is a connected set of data blocks. Extents are coming into focus when a segment (the next logical storage unit) is filled; this is when Oracle extends the usable disk space with an extent-sized space (if not specified otherwise). Therefore one extent belongs to one file.

Figure 5 illustrates how a segment is made up from several logically connected extents. There are four types of segment in Oracle:

- *Data segment:* every table data is stored in a data segments.
- *Index segment:* is used to efficiently store different indexes.



- *Temporary segment*: for every operation Oracle might use a temporary work area that is discarded after the operation is successfully completed.
- *Rollback segment*: all modified but not yet committed values and elements can be stored here. In the latest Oracle versions (9 and upwards) this segment does not exist.



5. Figure: Physical structure of Oracle

A segment can spread across more files. Segments are considered to be the physical representations of tablespaces.

### 3.4. How the system works

Upon initializing Oracle the system allocates memory area and starts several processes (threads). This is called an Oracle instance. For every Oracle database there is an Oracle instance responsible for the seamless operation of the database.

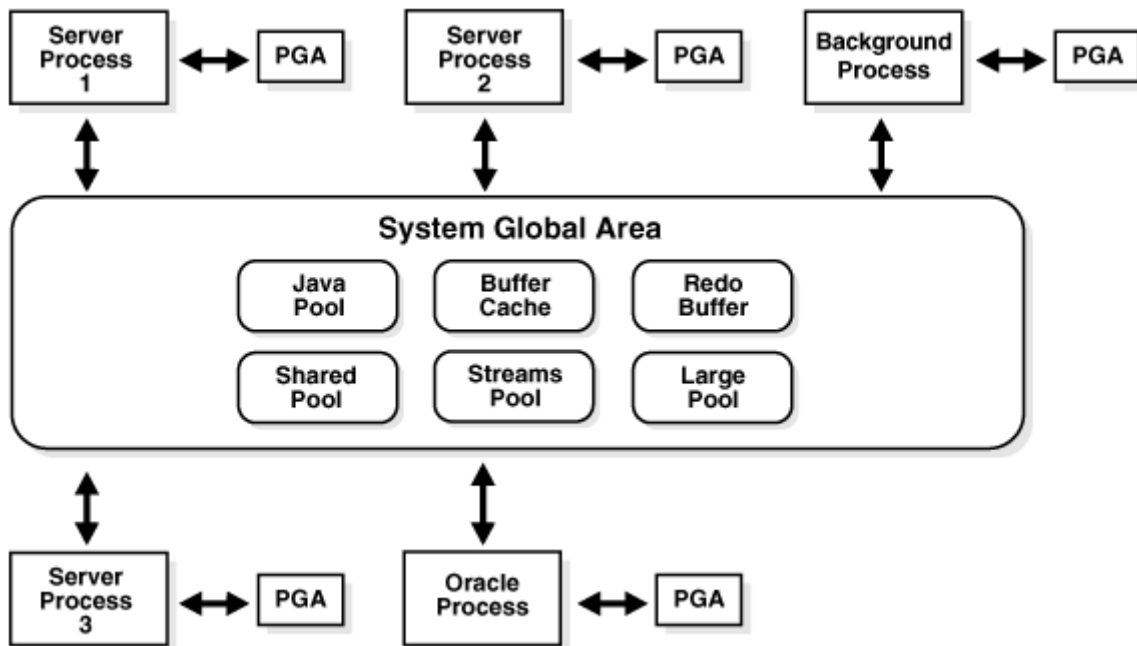
The *shared* memory area allocated at initialization is accessible for all processes and is called the SGA (System Global Area). The SGA contains all information required to control Oracle, and also functions as a cache: includes the most recently used blocks, a redo-log buffer, the data from most recently used directories and files, most recently executed statements and their results (database buffer cache), memory structures of Java processes used in Oracle, and parts of the data dictionary (see Figure 6).

The server processes illustrated on Figure 6 will be described later in more details.

For each server process a memory structure called PGA (Program Global Area) storing the state of the given server process is also allocated.

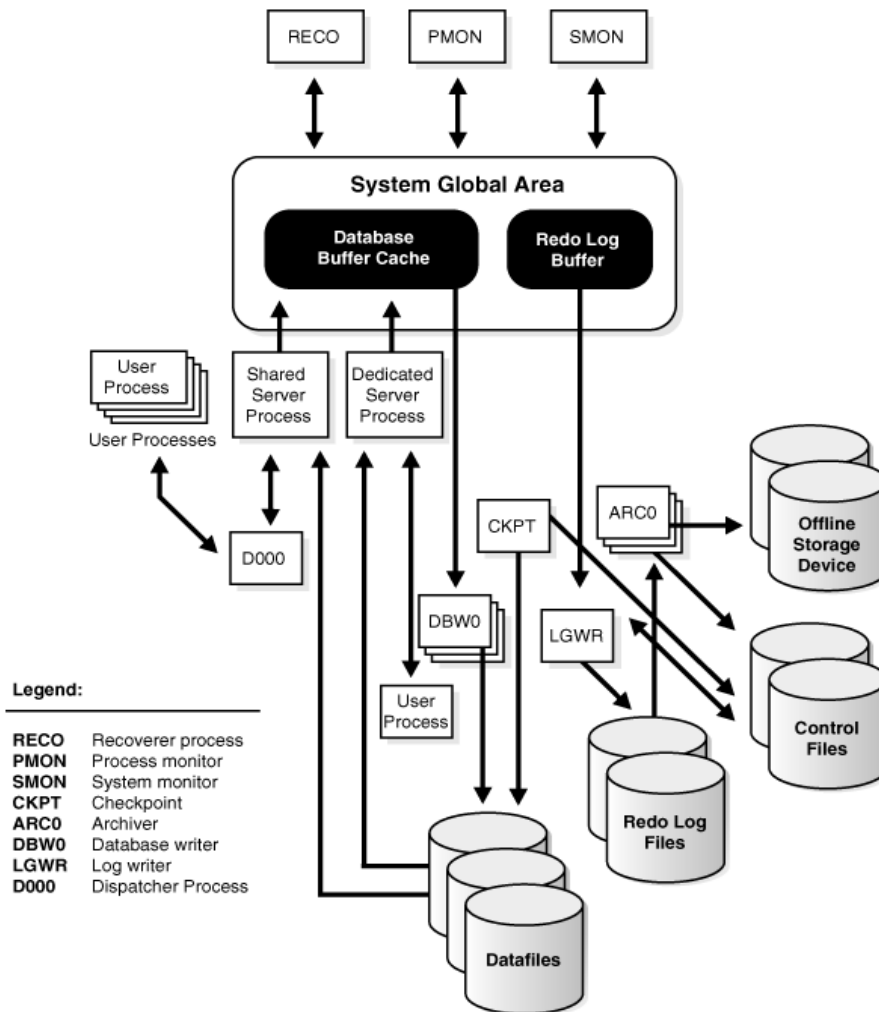
The processes started by Oracle are either server processes or background processes. Server processes are operations serving the user, such as SQL parsing and execution process, data output and input process between the backup storage and the SGA, and the process presenting the results to the user. The background processes - which significantly outnumber server processes in Oracle - are

responsible for different maintenance tasks, and for preserving system efficiency. The most important background processes are:



6. Figure: Data stored by the System Global Area (SGA).

- *System monitor (SMON)*: a process that carries out recovery operations after system failures. It is automatically started during every Oracle startup and shutdown. In other cases it is periodically "awakened" to see if there is anything to do, for example performing periodic cleanup of temporary segments that are no longer needed.
- *Process monitor (PMON)*: while SMON performs cleanup after system failures, PMON is in charge to perform process recovery when a user process fails. When a user process is not executed completely, PMON cleans up the cache and frees the transactions, locks and other resources that the process used.
- *Database writers (DBWn)*: write modified blocks from the SGA to the corresponding data files. Oracle allows a maximum of 10 database writer processes.
- *Log writes (LGWR)*: writes redo-log buffer entries from SGA into disk. LGWR writes the redo log entries sequentially into a redo log file.
- *Archiver (ARCn)*: the archiver process copies the redo log files to archival storage when they are full.
- *Lock manager server processes (LMS)*: in an Oracle Real Application Clusters environment, this process manages resources and provides inter-instance resource control (like Oracle parallel server).
- *Dispatcher (D000)*: dispatches the few server processes between the user processes (in case the shared server configuration is used). Thus more users can be served with the same number of server processes. The dispatcher provides a link between the Oracle server and the user processes (running on the client). The dispatcher can be accessed using the SQL\*Net (Net8) protocol.



7. Figure: Oracle background processes

The relationships between the background processes are illustrated by Figure 7.

To summarize the above: an Oracle instance consists of server and background processes, and the memory areas (e.g. SGA) allocated by the system.

Oracle and user processes (client programs) always communicate using the Net8 protocol. Net8 offers transparent connection, by providing a transparent layer between Oracle and the different networking software and programming environments (session and presentation-level protocol). Thus Net8 can be adapted to e.g. IPX, SPX, IPv4, IPv6, TCP, TCPS networks. The same protocol is responsible for language settings. Therefore it is only the client side (user) settings of the Net8 that define in what language the Oracle messages, the data field names are displayed, or which character set serves as a basis for sorting.

### 3.5 Security in Oracle

Only users defined in Oracle can access the Oracle system. These Oracle users and their privileges are (primarily) independent from the users and the security system of the underlying operating system.

In Oracle user privileges can be defined on two levels. Firstly, system privileges (e.g. a create privilege - CREATE TABLE, a session privilege - ALTER SESSION) which allow the user to perform system level activities; and secondly object privileges are granted.

A combination of certain system and object privileges defines a role. The system already contains a few pre-defined roles, such as DBA (DataBase Administrator), CONNECT<sup>3</sup> (enables connection to the database) and RESOURCE.

Since in most cases it would be extremely inconvenient to enumerate all the required privileges to perform a certain task, it is recommended to select a role that contains all these privileges. For example one needs over twenty different system privileges for database disk space maintenance, but instead of listing these all, we just say one needs a DBA role. However, for user data and certain privilege maintenance the CREATE/ALTER/DROP USER privileges are sufficient.

## 4. Oracle administration

By providing different tools, Oracle supports remote administration (as well). The key element of (remote) database maintenance is the Enterprise Manager, which contains several tools - such as Instance Manager, Schema Manager, Security Manager and Storage Manager - that can also be accessed as separate applications. In the following paragraphs the main features of the Oracle SQL Developer will be described in more details, The SQL Developer 3.0 is a Java 1.6 based administration system, which can run on various platforms.

### 4.1. Starting SQL Developer

There is no setup for SQL Developer, simply download the zip package that contains the program, unzip it and run sqldeveloper.exe (on Windows) or sqldeveloper.sh (on Linux).

If you start the program for the first time, it will ask for the java JDK install path. So it is important to have JDK (not JRE) to run SQL Developer.

SQL Developer has three modes, you can access each mode with the tabs on the left side of the screen. The first one is Connections (you can browse database connections here), Reports (detailed report information about the data in the database) and DBA (database administrator functions).

To connect to a database click on the green plus sign inside the Connections tab. Here you can specify the details of the connection:

- *Connection name:* custom user chosen name for the connection
- *Username / Password:* Optional field for the database connection username/password
- *Save Password:* If you check this box, SQL Developer will save the password on the local machine (don't use it at the workshop)

Under this box there is the Oracle tab, where you can define the connection parameters:

- *Connection type:* Defines the connection type to the database.
  - With the Basic mode you can define the server name (dns), port (1521 or 1526 is very common) and SID (System Identifier for the database instance) or Service name (if the database is a service)
  - With the TNS mode you can connect to the database through an OCI client.
  - With the Advanced mode you need a JDBC URL to connect the database (usually specifies dns, port, username and password, etc.)
- *Role:* You can specify the role of the user which you use for the connection. During the course you can only use default role, SYSDBA is for system administrators and SYSOPER is for system operators.

During the workshop we will use the Basic connection mode with the following parameters:

- *Hostname:* rapid.eik.bme.hu
- *Port:* 1521
- *SID:* szglab

---

<sup>3</sup> The CONNECT role before 10gR1 allow users to connect to the database, create and use some type of objects.

(You will get your username and password at the first workshop.)

If all given data is valid (can be tested with the Test button), then you can connect to the database. It is wise to save the database connection for further use. After connection/saving the connection a new connection will appear in the Connections tab. By clicking on the plus sign, you can connect to the database (a plug will show on the database icon, that the connection is alive). If you want to disconnect, left click on the connection name and from the popup menu choose the “Disconnect” menu item.

In the DBA tab (if not shown click the View menu DBA item) you can add a new connection with the plus sign. You can choose from the defined connections and after adding a new connection a tree will appear when you click on the plus sign next to the database connection’s name (more in the DBA mode – Database Configuration section).

The Reports mode works different from the others. Here you can generate reports on the database by clicking on leaves of the All reports tree. By clicking on a leaf (E.g. Data Dictionary Reports > About Database > Version Banner) will pop up a question, where you can define on which database (and with which parameters) you want the report to run.

In the next sections there will be more detailed information about the important parts of each tab and leaves on those trees.

#### 4.2. DBA mode – Database Configuration

In this part, you can check the basic metrics of the database, and check the settings of the database instance. Changing some of the static parameters require to restart the database instance, but dynamic parameters can be changed without restarting the database.

If we have SYSDBA role on the database, then we can change these parameters, but it’s not wise to do this using SQL Developer (there are other ways to change these parameters). SYSDBAs can restart the instance by clicking on the connection name and choose *Restart server* from the popup menu.

#### 4.3. Reports mode - Sessions

Inside the Reports tab there is a Sessions part (Data Dictionary Reports > Database Administration > Sessions), where you can check the current sessions and their parameters / settings and you can also perform various actions with them. (E.g. kill a “stuck” session). The most interesting part is the “Active SQL” tab on the left side of the screen, where you can check the currently active (running) SQL commands. Using this part of the software, developers can find the weak points (slow queries) of the applications using the database.

#### 4.4. Reports mode - Locks

Like with the sessions, there is a way (Data Dictionary Reports > Database Administration > Locks) to check the current locks on the database. Here you can also see the internal locks (made by processes inside Oracle). Using this information, developers can find the malfunction points of long functions and complicated queries. This view can also be accessed through the V\$LOCK view from any SQL interface.

#### 4.5. Connections mode

In the connections tab there are the user defined objects grouped by type. (This whole part is also called Schema Manager). You can also check objects created by other users under the “Other Users” leaf.

Here you can define your own objects (Tables, Views, etc.) using the built in wizards (left click on each type, then “new ...”), and also modify existing data and change the object settings (modify column type, add constraints, triggers, etc.).

Inside SQL Designer there are a lot of graphical interfaces, which can help you to perform common actions.

#### 4.6. DBA mode – Security

The Security part in the DBA tab contains information about the access control, namely to audit, profiles, roles and users. The access restriction usually is defined with the users, in Oracle a user is defined with the username and password, but Oracle also supports strong encryption and open key identification (PKI) in authentication. (See: Oracle Wallet and Enterprise Security Manager in OCI client) By default Oracle only encrypts the passwords, all the other data communications are pure text, but the database administrator can configure the system otherwise. (Recommended to have at least SSL based encryption – but it also requires higher bandwidth and makes the login sequence longer.)

Distributed databases have different login systems. (See: Enterprise Logon Assistant)

In the DBA tab administrators can manage the users (Security > Users), for example to add a new user to the system, left click on Users and click “Create new...” from the popup menu. In the Create user window there are several tabs, which are (some tabs may not present):

- *User*: Basic user information with username, password, default and temporary tablespaces
- *Granted Roles*: Users can have multiple roles, here you can configure which role the user will have (also can grant and admin a role)
- *System Privileges*: Users can have system wide privileges, in this tab those can be set.
- *Object Privileges*: Just like the system privileges, object (and type) privileges can be set here.
- *Quotas*: Just like in a UNIX system users can have limited physical space in each tablespace.
- *Proxy users*: “Average” users can have extra privileges using a proxy users option.

#### 4.7. DBA mode – Storage

In the DBA tab there is the Storage part, where metrics of physical storage can be seen. Here database administrators can create and manage tablespaces, rollback and data files. It also shows the disk usage and detailed information about each tablespace. There is one particularly important metric, the High Watermark, which shows the maximum space used by each tablespace. This is not the limit, but the actual highest usage. To change anything in Storage, the user must have DBA role.

### 5. Other Oracle products

- *Pro\*languages*: contains different linking points, pre-compilers for high-level programming languages, such as C, C++, Cobol, Fortran, Pascal, PL/I.
- *PL/SQL*: unique for the Oracle database manager, an SQL enhanced with procedural elements. Features: control statements, procedure / method definition, conditional execution (command IF), auxiliary variable definition etc. These can be stored in the Oracle database and run from applications on either the server or the client side.
- *Designer*: a fourth generation schema designer and automatic code generation tool. Target database might be - apart from Oracle - DB/2, Sybase Adaptive Server, Microsoft SQL Server or any ODBC compatible database-manager.
- *Internet File System*: provides services via the Internet, such as: remote administration, remote data access and SQL execution.
- *InterMedia*: a service package supporting the development and serving of Internet and wireless applications containing multimedia data.

- *JDeveloper*: e-business and Internet content developer tool. Generally includes the Internet File System and InterMedia packages. It is entirely written in Java and contains a UML modeler and several software development-related assistance programs.
- *Reports*: is used for quickly creating and formatting database queries. Even though Internet use was the main focus, it is accessible through XML, CORBA and EJB protocols.
- *Warehouse Builder*: design and development framework for supporting data warehouse development.
- *Data Mining Suite* (Oracle Darwin): a simple-to-use application that uncovers hidden facts and relationships.

## 6. Appendix

The appendix does not relate closely to the subject of the workshop, therefore will not be part of the assessment.

These special features of Oracle versions will not be used during the workshops. If applicable, these modifications will be highlighted by the lecturers. However, including this information here illustrates the differences between versions and gives a general idea of Oracle development trends.

### 6.1. Comparing newer Oracle versions

#### 6.1.1. Oracle8

Oracle8 is the first object-relational database manager of the Oracle series. In a nutshell this means Oracle can store object data and methods, the type concept is introduced and multimedia data can be handled more efficiently.

The most significant change about data types are that the more dynamic and efficient LOB types are used instead of LONG. The maximum sizes of CHAR and VARCHAR2 changed from 255 and 2000 to 2000 and 4000.

For application development purposes this version has been extended with JDBC, enabling data access in Java. External methods can be called from the database using HTTP or (CORBA standard) IIOP protocols.

#### 6.1.2. Oracle8i

Oracle is specifically adapted to support Internet applications. The installer and the clients are Java-based, moreover using the built-in Java virtual machine (VM) the server itself became capable of running Java applications. Replacing the earlier Webserver, the more efficient WebDB is introduced as a separate product. Webserver created HTML page contents using PL/SQL. However, in WebDB these two things were swapped: HTML pages are put together using wizards, while the PL/SQL source is generated automatically.

The server also includes the InterMedia package, which supports multimedia data Internet display, storage, play and transfer and also provides query and report results in HTML, PDF, Word or Excel format. All of these features support the EDI (Electronic Document Interchange).

The element set belonging to type DATE is extended, primarily in the field of conversions.

For example notions like mid-month, 10 days and mid-year are introduced. Dates can be freely overridden according to the expectations and demands of the business context. SQL command DROP COLUMN finally enables column deletion. Type ROWID is generalized into UROWID - a half-logical key, depending on the primary keys of the tables - to further increase data access speed.

#### 6.1.3. Oracle9i

The database-manager is almost completely refurbished, as for the fields of data mining and data warehouses. The most important feature is the SQL-99 compatibility (versions before Oracle 9i provide only SQL-92 compatibility).

Unlike the earlier Oracle products this version features dynamic - runtime - modification of SGA area and contents. The respective commands are part of the SQL command set. This version supports the

handling of different block sizes, moreover, provides an interface to access user objects in external databases.

#### 6.1.4. Oracle10g

The Oracle 10g was first released (R1) in 2003, than the second big release (R2) came in 2005. The “g” in the product’s name refers to grid computing. The grid in the name is not a buzzword, it actually refers to the distributed logical structure and resources that the program can handle and manage. 10g also has hundreds of improvements and performance tuning.

If advanced features such as load balancing and automatic failover are desired, there are optional sections of the listener.ora file that must be present. Automatic Storage Management (ASM) includes multiple disk operations and a non-ASM database migration utility.

Introduced RECYCLEBIN command for storing objects before they are dropped. Required new PURGE command for maintenance.

The AWR is also introduced in 10g, which collect performance statistics about the database instance. The Automatic Database Diagnostic Monitor (ADDM) analyzes data in the Automatic Workload Repository (AWR) to identify potential performance bottlenecks. For each of the identified issues it locates the root cause and provides recommendations for correcting the problem.

#### 6.1.5. Oracle11g

In 2007 Oracle introduced 11gR1, than 2 years later the Release 2 came. Oracle has invested heavily into self-tuning capabilities including automated storage and memory management and intelligent tuning advisors. Now in 11g, Oracle closes the loop and offers intelligent automation tools to create a self-healing database. You can tell 11g to automatically apply SQL profiles for statements where the suggested profile give 3-times better performance than the existing statement.