

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación



MIGRATION TO THE ORACLE CLOUD

TRABAJO FIN DE MÁSTER

Jaime Cesar Martínez Avendaño

2013

Universidad Politécnica de Madrid
Escuela Técnica Superior de Ingenieros de Telecomunicación

**Máster Universitario en
Ingeniería de Redes y Servicios Telemáticos**

TRABAJO FIN DE MÁSTER

MIGRATION TO THE ORACLE CLOUD

Autor

Jaime Cesar Martínez Avendaño

Directora

Dra. Carmen Costilla Rodríguez

Departamento de Ingeniería de Sistemas Telemáticos

2013

Abstract

This work is divided in two parts.

The first part follows a top-down approach, describing the necessary technological state of the art in order to carry out the migration of databases to the Oracle Cloud. For this, many tools and their corresponding functionalities are here briefly described. It includes the procedures for developing cloud applications using the components offered by Oracle Cloud. Below the cloud, the ODI (Oracle Data Integrator) is mentioned. Finally, at the lower level, Oracle11g and TOAD are described as very powerful tools for databases.

In contrast, the second part adopts a bottom-up approach, from how to create an information source (silo or island) in order to migrate it to the Oracle Cloud. For this, we develop an academic and demonstrative relational database (its design, creation and population), that finally will be migrated to the Oracle Cloud.

General Index

Abstract	5
General Index.....	6
Figures Index.....	8
Part I	10
1 Objectives of this work.....	11
2 Technological context used	11
2.1 Cloud Computing.....	11
2.2 ODI.....	11
2.3 SGBD Oracle	11
2.4 TOAD.....	12
3 Considerations for a Database Migration	12
3.1 Database Migration	12
3.1.1 Metadata Migration	12
3.1.2 Language Migration.....	13
3.2 Integration Architecture	13
3.2.1 Products, Utilities and Tools.....	13
3.3 Methodology and Design.....	13
3.4 Migration Options	13
3.5 Methodology and Design.....	14
3.5.1 Assessment.....	15
3.5.2 Analysis and Design	16
3.5.3 Migration.....	16
3.5.4 Testing	16
3.5.5 Optimization.....	16
3.5.6 Deployment.....	17
3.5.7 Post-Production Support.....	17
4 Cloud.....	17
4.1 What Is Cloud Computing?	17
4.2 Why Migrate?.....	17
4.2.1 Why Oracle?	19

5	ETL Integration	21
5.1	ODI Features.....	21
5.2	ODI Fundamentals.....	21
5.2.1	E-LT Focus.....	21
5.2.2	Declarative Design.....	22
5.2.3	Knowledge Modules.....	22
5.3	Architecture	23
5.3.1	Repositories.....	23
5.3.2	Consoles	24
5.3.3	Agents.....	24
5.4	Integration	25
6	TOAD	26
6.1	Schema Browser	26
6.2	SQL Editor.....	27
	Part II	28
7	Creating a Database Application over Oracle Cloud	29
7.1	Generation of a database with Oracle 11g and TOAD.....	29
7.1.1	Installation of the Oracle Developed Days Virtual Machine.....	29
7.1.2	Creating a Database with Oracle SQL Developer	30
7.1.2.1	Creating tables	32
7.1.2.2	Creating the referential integrity rules.....	33
7.1.2.3	Inserting Data	33
7.1.2.4	Creating a Backup SQL Script.....	34
7.2	Migration to Oracle Cloud	36
7.2.1	Accessing the Cloud.....	36
7.2.2	Loading the Database in the Cloud	37
7.2.3	Creating an application in the Cloud	38
7.2.3.1	How show tables	39
7.2.3.2	Creating links in the main page	40
7.2.3.3	Creating a form with privilege of Creation, Update and Delete.....	42
7.2.4	Execution of the application on the Oracle Cloud.....	44
8	Conclusions	45
9	Bibliography.....	46

Figures Index

Figure 1 - Software Development Methodology	15
Figure 2 - Typical Migration Project Life Cycle	15
Figure 3 - Approaches over E-T-L and E-TL	22
Figure 4 - ODI Architecture	23
Figure 5 - Integration to Oracle BI	25
Figure 6 - TOAD SQL Editor.....	27
Figure 7 - Importing a new appliance	29
Figure 8 - Running the Importation.....	30
Figure 9 - Oracle Developed Days running.....	30
Figure 10 - Oracle SQL Developer main page.....	31
Figure 11 - Creating a new connection.....	31
Figure 12 - Testing the new connection.....	31
Figure 13 - Creating a SQL Script with the export wizard	35
Figure 14 - Export SQL File	35
Figure 15 - Oracle Cloud Website	36
Figure 16 - Choosing Oracle Database in the Cloud	36
Figure 17 - My Account page in Oracle Cloud.....	37
Figure 18 - SQL Commands	37
Figure 19 - Object Browser	37
Figure 20 - Database Workspace	38
Figure 21 - Creating a new Application	38
Figure 22 - Creating a view	39
Figure 23 - Report Icon	39
Figure 24 - Classic Report Icon	39
Figure 25 - Page Attributes.....	39
Figure 26 - Tab Options	39
Figure 27 - Report Source	40
Figure 28 - Report Attributes	40
Figure 29 - Confirm	40
Figure 30 - Main Page	40
Figure 31 - Regions Section	40
Figure 32 - Creating a HTML Region	41
Figure 33 - Display Attributes	41
Figure 34 - Buttons Section.....	41
Figure 35 - Tasks	41

Figure 36 - Creating Multiples Buttons	41
Figure 37 - Display Region	42
Figure 38 - Action Button Region	42
Figure 39 - Form Icon	42
Figure 40 - Form with Report Icon	42
Figure 41 - Report Page	42
Figure 42 - Data Source	43
Figure 43 - Tabs	43
Figure 44 - Report Columns	43
Figure 45 - Action Button Region	43
Figure 46 - Running the application	44
Figure 48 - Main page of the application	44
Figure 50 - Making a Query	44

Part I

State of the Art

1 Objectives of this work

The main objectives of this work can be classified into the next points.

- Demonstrate the operation of Oracle Cloud and its benefits for the development of new applications in the cloud.
- Integrate databases of local servers to servers located in the cloud.
- Teach the basic steps for building an application in the cloud through the stages of creating databases, migration of database and application development using data migrated to the cloud.

2 Technological context used

2.1 Cloud Computing

Cloud computing is a colloquial expression used to describe a variety of different computing concepts that involve a large number of computers that are connected through a real-time communication network. Cloud computing is a jargon term without a commonly accepted non-ambiguous scientific or technical definition. In science, cloud computing is a synonym for distributed computing over a network and means the ability to run a program on many connected computers at the same time. The popularity of this term can be attributed to its use in marketing to sell hosted services in the sense of application service provisioning that run client server software on a remote location.

2.2 ODI

Oracle Data Integrator is a comprehensive data integration platform that covers relevant data integration requirements:

From high-volume, high-performance batch loads, to event-driven, trickle-feed integration processes, to SOA-enabled data services.

2.3 SGBD Oracle

The Oracle Database is an object-relational database management system produced and marketed by Oracle Corporation.

Oracle Database is considered as one of the systems most comprehensive databases, including:

- Transaction support

- Stability
- Scalability and
- Multiplatform support.

2.4 TOAD

Tool for Oracle Application Developers (TOAD) is a software application of development and a SQL database administration, considered a useful tool for Oracle DBAs. Currently available for the following databases: Oracle Database, Microsoft SQL Server, IBM DB2, and MySQL.

3 Considerations for a Database Migration

3.1 Database Migration

As it is well known, to carry out the migration process information in a database, the time required to execute it, depends on several factors, which play an important role in relation to time delay to the migration, among these factors usually involve the number and size of Stored Procedures, Triggers, Views, Index numbers, Tables, Users and Data Types, just to mention some of them.

3.1.1 Metadata Migration

The Oracle SQL Developer Migration Workbench can be used to migrate the data. This tool will also automatically migrate Data Definition Language (DDL) statements with close to 98 percent accuracy.

Data migration volumes and migration times are quickly moving targets as disk speeds increase and interconnects become faster. At the time of this writing, simple to very complex database migrations and their equivalent volumes are as follows:

- Simple: less than 100 GB of data
- Medium: less than 500 GB of data
- Complex: less than 2 TB of data
- Very complex: more than 2 TB of data

Complexity	Time Estimate
Simple	1 - 2 days
Medium	2 - 4 days
Complex	5 - 8 days
Very complex	8 - 12 days

Table 1 - Data Migration based upon Migration Complexity.

3.1.2 Language Migration

Migrating an application is typically three to 10 times more time consuming, and therefore more costly, than migrating a relational database, because you have to take into account changing business requirements, changing technical specifications, unit testing, bug fixing, compile time, build time, and other factors. This makes it extremely important in an application migration that as much of the legacy client/server code as possible is refactored and code migration tools are leveraged.

3.2 Integration Architecture

3.2.1 Products, Utilities and Tools

Many tools are available today for everyone who wants to migrate their systems, between them should be recognized the followings:

- Job schedulers
- Database and application server management and monitoring tools
- Testing tools and
- Even database and application deployment scripts

If these solutions are custom, they will take more time to migrate than vendor tools, because sometimes these solutions are not typically used in client/server environments.

3.3 Methodology and Design

Like any software development project, migration projects require careful planning and good methodology to ensure successful execution. When migrating from one database platform to another, a good portion of the database design can be carried forward, especially the schema design for relational database migrations. It may be necessary to make some changes to the existing design and architecture if you want to leverage new features in the Oracle database in favor of older mechanisms, such as using database table partitioning and Oracle Real Application Clusters (RAC) to support multiple workloads instead of maintaining different databases in sync with database replication technologies. Therefore, a certain amount of rationalization is possible when migrating to Oracle from other databases, resulting in consolidation of many databases and schemas from the source database into a few schemas in Oracle.

3.4 Migration Options

Database migration is the most common because it allows users to migrate to new database platforms, yet leave applications intact, with no changes to existing

functionality and business rules, including the languages in which the applications were originally developed.

Legacy applications deployed on IBM mainframes and other proprietary and legacy platforms are usually retooled to run on open and distributed platforms using software that can mimic IBM mainframe environments or that can effectively provide the same capabilities on their own.

Which migration option you choose will depend on your business requirements and constraints. The easiest migration option is to Web service-enable an existing application so that it can interact with other applications on the Web or with applications deployed in a cloud environment.

Migration options such as database platform migrations and replatforming applications developed on legacy platforms are very popular because they are easier to execute and realize the benefits of such migrations quickly. On the other hand, migration options involving a complete rewrite/rearchitecture of an application or development of a completely new piece of software are less likely to be chosen due to the cost and time involved in executing such a projects.

The goal of this approach is to significantly reduce application changes as a result of database migrations. However, organizations need to thoroughly test these technologies for performance and for accuracy of the converted SQL statements before deploying them in mission-critical environments.

These emerging technologies definitely have a role in reducing the overall migration effort in large migration projects by reducing changes to the application due to database migrations.

3.5 Methodology and Design

Regardless of the migration option chosen for implementation, a sound migration methodology and design needs to be in place before embarking on such a project.

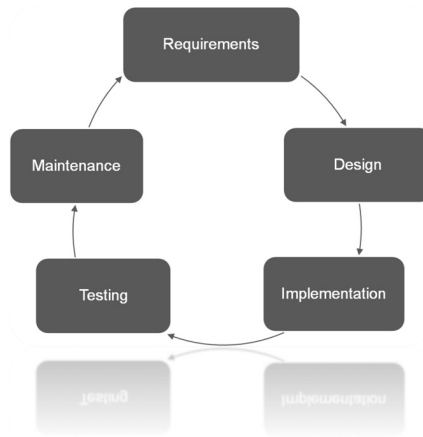


Figure 1 - Software Development Methodology

The Waterfall methodology can be applied to migration projects, but sometimes the requirements gathering phase is replaced with an assessment phase.

For instance, for migrations involving a full rewrite or significant transformation to existing applications, the requirements gathering phase is applicable. In projects in which only the database platforms are being considered for migration, there is no need to gather business requirements again, since existing applications already have all the business rules and requirements taken care of. So, instead of requirements gathering, an assessment phase is used to understand the existing environment and determine the best way to migrate it to a new environment.

Migration of one database platform to another requires some changes to existing applications due to the technical differences between various databases.

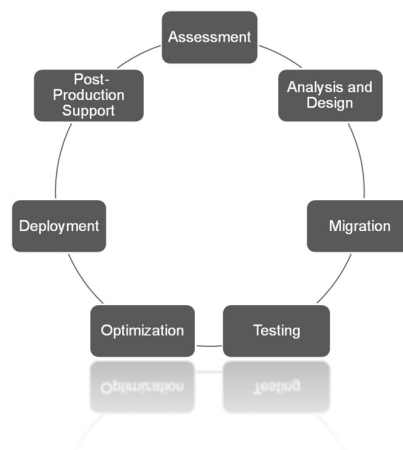


Figure 2 - Typical Migration Project Life Cycle

3.5.1 Assessment

The assessment phase is akin to the requirements gathering phase of the Waterfall method for software development. Instead of gathering business requirements from a software development perspective, however, you collect information pertaining to project management, the potential cost of the migration, migration approaches, tools to use, and so on.

3.5.2 Analysis and Design

The analysis and design phase usually consists of determining the implementation details on the target database. Because of the differences in implementation of data types, security roles and privileges, transaction management, and SQL code, it is important to develop a plan that leverages appropriate features and functionalities in the Oracle database.

3.5.3 Migration

The amount of time it takes to complete the actual migration of objects and data from one database is relatively less than the amount of time it takes to complete an overall migration from assessment to production rollout. Migrations of one relational database to another are comparatively easier than migrations of a non-relational database to a relational database, because the organization of objects in a relational database is quite similar compared to non-relational databases such as hierarchical and network databases.

Database migration tasks can be divided into the following categories:

- Database schema migration
- Data migration
- Database stored program migration
- Application migration
- Database administration script migration

3.5.4 Testing

Effort involved in testing the application and the database after migration usually is the largest contributor to the migration effort. Testing in a migration project usually comprises tasks such as data verification, testing of migrated business logic in stored procedures, functions, and triggers, testing of application interaction with the new database platforms, and testing of database maintenance scripts. Some of these tasks can be performed easily with the help of automated tools or relatively simple scripting. But some tasks, such as testing of database objects with business logic, can be cumbersome because of lack of automated testing tools.

3.5.5 Optimization

Migrating applications from one database to another database sometimes results in poor performance. This occurs because these applications are highly optimized for a particular database system over long periods of time. OEM can be used to help resolve any performance issues post-migration. It is very common for organizations to set aside at least two to three months for performance testing after migration of mission critical applications. Some performance issues can also be caught during the functional / integration testing phase. But some issues may crop up only under certain amounts

of load on the system. Therefore, it is essential to test the performance of the new platform thoroughly to identify potential bottlenecks in the system and address them before the new platform is rolled out into production.

3.5.6 Deployment

Many tasks to be executed in the deployment phase get their input from the assessment phase and from the analysis and design phase. During these phases, the target system architecture and all necessary software components to be used in the new system are evaluated. Based on the outcome of these phases, new software and hardware systems are acquired. Work on this phase may begin early in the migration project, as many organizations have to follow certain business practices regarding acquiring new hardware and software.

3.5.7 Post-Production Support

It is a common practice to support a newly deployed database environment with personnel who were involved in the migration process to troubleshoot any issues that may come up immediately after the new environment goes live.

4 Cloud

4.1 What Is Cloud Computing?

The National Institute of Standards and Technology (NIST) defines cloud computing as follows:

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction.

The basic premise of cloud computing is that users can get access to any IT resource, including storage, CPU resources, memory, and software, over the Internet whenever they want, and can pay for their actual use of the resource rather than incurring capital and operational expenditures in order to own and operate the IT infrastructure.

4.2 Why Migrate?

One of the main problems when we try to adopt a new technology, it is to close the doors to the advance, as in many organizations believe that if it works until it is old, why change.

These organizations miss out on many important advances in software and hardware technology, as well as the advantages offered by the latest standards and architectures.

Not adopting these new technologies can give you some of these disadvantages:

- **Less agility:** Significant effort is required to make changes to existing applications.
- **Longer time to market:** It takes longer to roll out new services and features to support business expansion and acquisitions.
- **Cost of maintenance:** Over the years, it becomes expensive to maintain staff for ongoing system maintenance and routine updates. Legacy software vendors also incur increased costs in supporting the very large number of customers using their products.
- **Difficult to integrate:** Integrating legacy applications with newer and modern standards-based applications, special tools, and services is necessary, but difficult.
- **Ease of access/globalization Legacy:** Applications, especially client/server applications, require client software to be installed on desktop computers so that users can access the applications. This, in turn, forces organizations to develop plans to push periodic upgrades to the desktop whenever changes are made to the software.

These would be some advantages if you choose to adopt these new technologies.

- **Faster, more efficient hardware:** Technology is improving at a faster rate, resulting in systems on the market today that are exponentially faster than their predecessors. Therefore, organizations can benefit from periodically updating their systems by having similar or smaller hardware configurations delivering the same level of performance.
- **Leveraging of the latest standards:** It becomes much easier to integrate and exploit the latest trends in software and hardware technologies, such as SOA and cloud computing. This can result in standardization of application architecture and leveraging of business rule engines, business process management methodologies, and Web services.
- **Reduced total cost of ownership:** Migrating to newer software and hardware platforms can result in consolidation, therefore reducing the number of disparate environments that must be maintained in the future. It may also result in higher utilization of IT resources such as CPUs and storage.
- **Increased agility:** It becomes easier to modify a business rule, alter a business process, or change a business service on the fly as doing so does not require unnecessary system downtime.
- **Governance/compliance:** In older technologies, it is very difficult to implement new requirements concerning governance and compliance.

Newer software products have out-of-the-box features to support these requirements, such as extensive auditing, logging, monitoring, and security features.

- **Longer shelf life:** Newer technologies have a longer shelf life, so organizations don't need to pay exorbitant fees for maintenance.

4.2.1 Why Oracle?

Many vendors today offer databases, application servers, development tools, and platforms. So, what sets Oracle apart from these other vendors? Key differentiators between Oracle and other software and hardware vendors include the following:

- Oracle is the only company that offers a full range of products comprising business applications, enterprise software, application servers, and hardware systems including servers, storage, and networking tools.
- Oracle is the first company to offer a highly optimized database platform and compute platforms by combining the latest in hardware and software technologies (Oracle Exadata, Exalogic). It also brought to the forefront the notion of balanced configurations. A balanced configuration approach ensures that when computing resources are required they can be added in a modular fashion to existing infra-structure. In other words, CPU resources, network and I/O bandwidth, and storage capacity are all added proportionately so that all resources are evenly configured.
- Oracle has a history of bringing important technologies to the realm of main-stream workloads, such as the use of encryption, compression by regular OLTP, and data warehousing applications. Typically, using these features involves a significant performance overhead. With Oracle Exadata, compression can be enabled even for high-performance OLTP applications in addition to data warehousing applications.
- Oracle has had a history of providing enterprise software that enables consolidation, scalability, and capacity on demand (e.g., Oracle Database Real Application Clusters or RAC) since the release of Oracle Database 10g and Oracle Application Server 10g in early 2000.
- Oracle offers out-of-the-box integration of various features and functionalities in its software offerings. Examples include Oracle Business Intelligence Enterprise Edition (OBIEE), which combines an application server, common metadata models, E-T-L (Extract-Transform-Load) tools, technologies and dashboards, and reporting tools; and Oracle Database, which offers full support for database-wide encryption and compression out of the box.
- Oracle software products offer unique features, such as:

- Oracle Database row-level locking, which avoids locking rows of data in tables when reading them. For all other relational databases, this is a nightmare, as they have to enable row locking even for a brief period, or enable “dirty reads” to avoid locking, which has its own side effects, such as reading uncommitted data.
- Oracle Database 10g Real Application clusters, which allow active-active clustering of multiple instances accessing a single Oracle database. Recently, other database vendors have begun to offer this type of solution as well.
- Organizations use an array of tools and technologies to manage and monitor different layers of IT systems. Oracle’s consolidated management software, Oracle Enterprise Manager (OEM) Grid Control, can monitor and manage all components in a data center. OEM also offers a simple Web-based user interface that can be accessed from any device, such as a PDA, laptop/desktop, or mobile phone.

Oracle	Other Vendors
Engineered systems (hardware and software)	Usually hardware- or software-only solution
Out-of-the-box integrated Oracle products and features	Varying degrees of integration required based on vendor products used
Superior performance, cost/ performance out of the box	Performance dependent on optimization of all layers
Supports all types of workloads (OLTP, DW/DSS)	Appliances exist for specific workloads, mostly concentrating on DW/BI analytics
Reduced total cost of ownership due to use of standard x86-based architecture, high-performance characteristics, and engineered system requiring less maintenance effort	Total cost of ownership varies depending on configuration used
Unified management infrastructure with applications to disk monitoring capabilities	Requires multiple products to manage and monitor the full stack

Table 1 - Highlights the primary differences between Oracle and other software and hardware vendors in the cloud infrastructure/platform space.

5 ETL Integration

5.1 ODI Features

There is a wide range of features that make ODI an excellent choice to integrate data. Its features translate into benefits for the developer, business user, and for other applications requiring integration previous layer. Thus, within the main features can indicate the following:

- The declarative approach to establish rules, which involves separating the definition and implementation of processes, in addition to reducing the learning curve for the developer, which shortens training time.
- Quality assurance of data through the use of rules that identify faulty data or that do not meet the conditions for integration.
- Best performance at runtime because of its particular focus the concept ETL (Extraction-Translation-Loading).
- Operating system independence, due to inherited characteristic of Java technology that allows the execution of a virtual machine.
- Support a wide range of data sources including RDBMSs, office files, XML, flat files, etc..
- Integration technologies within Oracle Fusion Middleware solutions.
- Prepare the data for further processing by BI (Business Intelligence).
- Integration into SOA (Services Oriented Architecture), because of its ease to call external services and even create and deploy their own services.
- Presentation of administration and management console, unified through a Java EE (Java Enterprise Edition).
- Interface design from JDeveloper.

5.2 ODI Fundamentals

The potential of ODI is based on three main aspects, namely: an E-LT architecture, declarative design, and use of knowledge modules.

5.2.1 E-LT Focus

Within materialization mechanisms, the use of mechanisms of E-T-L referred to those focused on the steps of: extracting information from any data source, processing the data, and loading filtered data and standardized in the target repository. Under this concept and order of stages, an intermediate server is used for the translation stage. However, it is not essential the order given to the stages. In fact it is possible to mention that reconfigurations of this mechanism, depending on the needs of integration.

ODI uses an E-LT approach that eliminates the use of intermediate server, shown in Figure 1, getting free for their dependency and avoiding potential bottlenecks. It uses the engine power of the RDBMS source or destination. Although, in theory, you could make the translation phase at the origin, it is best to do it at the destination and which thus can more efficiently perform integrity checks against the final repository.

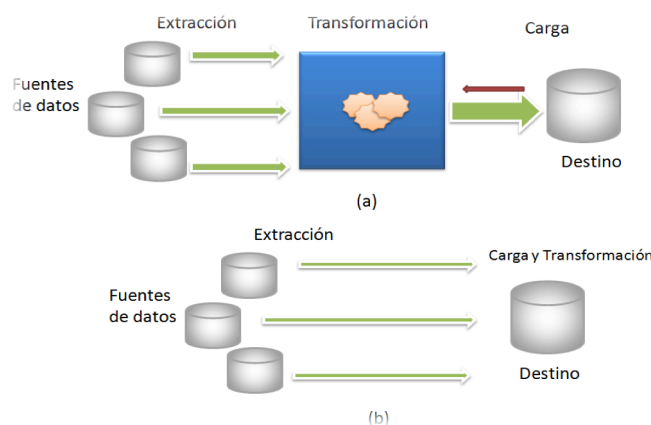


Figure 3 - Approaches over E-T-L and E-TL

5.2.2 Declarative Design

ODI is based on a declarative approach, it is an advantage to define the processes of translation and integration, as it manages to separate the logic "do's" implementation from the "how." Representing this an advantage over a traditional model in which both are intertwined, making it difficult to maintain. The rules are equivalent to the logic, thus being declarative rules described in natural language and expressions can be implemented with SQL (Structured Query Language).

There are four types of declarative rules:

- **Mapping**, used to transform multiples columns of a source data set to the destination.
- **Joints**, useful for linking two or more columns of data sets.
- **Filters** are applied to the source data sets to allow the passage of data that meet a given expression.
- **Restrictions** ensure the validity of the data before integration.

5.2.3 Knowledge Modules

They are intended to implement the "how to" in the software processes. Each is dedicated to a specific task within the process of integration. While independent of the rules, work together to generate code at runtime.

Although defaults are used as templates, given its focus on RDBMS technologies or specific tasks can also be edited to add new methods. A way of showing the following may be mentioned knowledge modules:

- **RKM** (Reverse-Engineering Knowledge Model), deals with the reverse engineering of heterogeneous systems.
- **LKM** (Loading Knowledge Model) module to load data from one system to another.
- **IKM** (Integration Knowledge Model) for integration into the target system.
- **CKM** (Check Knowledge Model) for checking the integrity of data flow.
- **SKM** (Service Knowledge Model), can expose data via web services.

5.3 Architecture

ODI architecture is completely modular, ensuring scalability and integration with other Oracle middleware components or even with other applications that work across services.

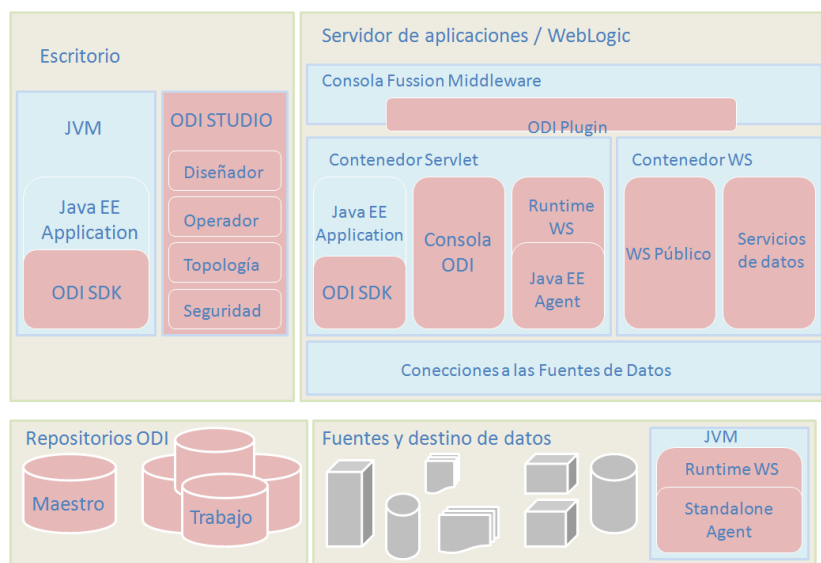


Figure 4 - ODI Architecture

5.3.1 Repositories

The importance of metadata in the integration process is vital, that metadata are stored in repositories ODI, and become the central part of the architecture. We have two types of repository: the master repository and work repository.

- **Master Repository:** It is the most important; there is usually only one. The master repository stores information security, the topology information, and source code of different versions of objects.
- **Work Repository:** Also important, It can be multiples and contain actual developed objects, their models, projects and the stage of execution.

The scheme working multiple repositories is recommended for any work environment.

5.3.2 Consoles

ODI has 3 options user interface for administration and to develop the processes. These are through the ODI Studio environment using ODI console and through the management console of Oracle Fusion Middleware using a plugin.

ODI Studio

Interface is the most complete and gives us direct access on the computer where you installed ODI. In his description, it is worth noting that there are four browsers that can be used by users depending on their roles.

These browsers are:

- **Designer** is the primary browser for developers and administrators because it is here where you will define and import the data sources through reverse engineering. Also, here are defined transformations and integration processes through declarative rules and even can generate automatic documentation.
- **Operator**, is useful once started and after the execution of the process, as it is oriented to the management and monitoring.
- **Topology**, here defined physical and logical architecture of the information system. It allows the registration of master and work repositories, and agents.
- **Security**, which allows management of users, their profiles and their corresponding permissions on the artifacts.

Although there are four different browsers oriented features, it should be noted that all of them work on repositories ODI.

5.3.3 Agents

Agents are Java components, whose function is to execute and orchestrate all the user-defined process. Initially connects to the master repository and it connect to any repository linked to it and retrieves scheduled and updated scenarios your own cleaning prior to a previous session. Depending on the action scheduled manages the processes that allow to obtain data from their various sources, then returns the status of implementation, which includes messages, update logs, etc.. Once the process starts, you may receive more requests due to its multi-threaded nature.

It should be noted that the agents do not perform the transformation process. This task is performed by the engine normally target database, leaving the main function of the agents orchestrate processes.

The agents can be deployed in two modes, as Java EE application deployed on a WebLogic server with all the benefits that this implies (clustering), or stand-alone for which is deployed on the Java Virtual Machine.

Finally, although defined in ODI, also can be called through a Java API or a web service, allowing integration into SOA.

5.4 Integration

In ODI should identify two integration approaches. The first refers to ODI, as a tool that allows the integration of different data sources, is what has been revised so far. The second approach means that, given its design and architecture can be integrated with other applications or technologies. Thus:

Integration facing BI applications within the process helps decision making by OLAP and data warehouses. In this particular case, ODI helps cover most stages from extraction to loading data and data quality audit, providing an architecture that can be trusted.

Integration through messages, ODI supports the standard JMS (Java Message Services). This ensures integration with applications that also support or are compatible with this standard

Integration through services, ODI can integrate into SOA and data services that can be deployed as Web services. Similarly, the transformation processes can be deployed.

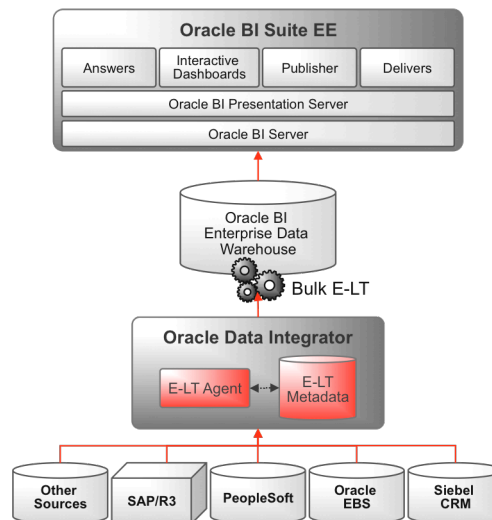


Figure 5 - Integration to Oracle BI

6 TOAD

TOAD is defined as a very powerful development tool for SQL and PL/SQL, with plenty of facilities for the programmer. TOAD design focuses on a friendly navigation interface, enabling management graphically database object creation, support for the creation of procedures, functions, jobs, etc.

Regarding editors, TOAD can work simultaneously with multiple files open, even different types such as SQL, PL/SQL, HTML, Java, etc.

Like Oracle Procedure Builder, TOAD has a parser that facilitates coding.

Among the most notable features of TOAD can be highlight the following:

- Simultaneous and automatic connection to different databases.
- Code Editor:
 - Code Templates.
 - Utility auto Replace.
 - Formatting code.
 - Management and navigation specification and package body.
 - Automatic parsing errors spigot.
 - Navigation Aid code (with highlighting in different colors).
- Building Tool Queries.
- Diagram Generation Tool E / R.
- Project Manager.
- Comparison of schemas and databases.
- Network Integration tools / OS (Ping, FTP, Telnet ...).
- Utility AUTOTRACE in the editor.
- Filtering of the data shown in the graphical tool.
- Windows helps DDL code generation, including: DBLinks, PIPES, Triggers, Jobs, Indexes, Tables, Sequences, etc..
- View and edit all Oracle types, including: Jobs, PIPES, DBLinks, cursors, etc..
- Outline Script Generation and database.
- Compilation and debuggin the generated code.
- Analysis of the results of the execution of judgments DBMS.
- Preparation of Plan Implementation and results.
- Storage of the last executions.

6.1 Schema Browser

This utility allows analysis from a single window much of the information in the database. Using the Schema Browser window, we can graphically navigate through the

different elements of the database and, using different windows help, we can: create, modify or delete those items.

6.2 SQL Editor

This utility allows you to execute SQL queries with building in the lower sections showing execution information, such as: the result of execution, the output of log messages, execution plans, etc.

In addition, this utility has a multitude of aids to the programmer as a highlight of the keywords or the objects used, analysis of the errors found, helps query writing showing the features available in packages or selected objects, table columns, etc.

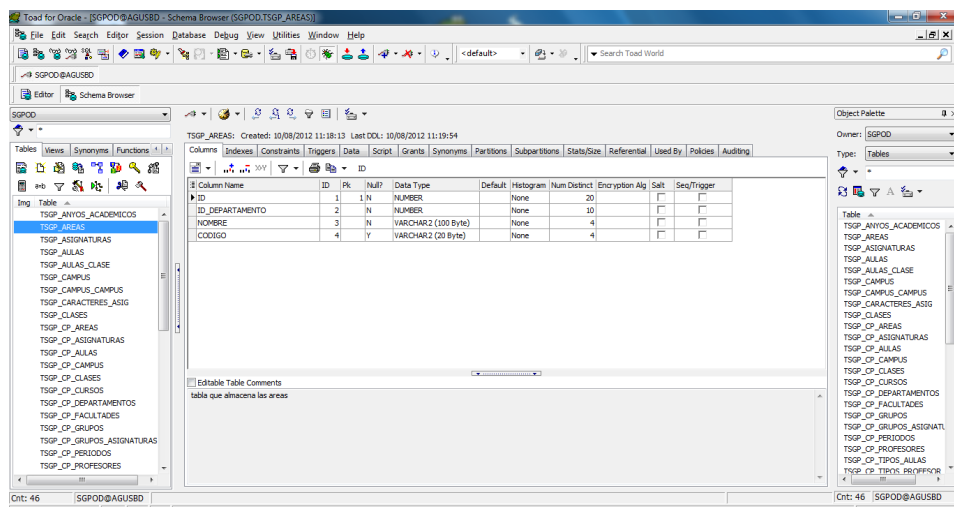


Figure 6 - TOAD SQL Editor

Part II
Practical Case

7 Creating a Database Application over Oracle Cloud

The second part of this work, intends demonstrate some practical aspects based on the technological context we have described in the first part.

A bottom-up approach will be adopted here, in order to describe the followed process of the information migration.

We start designing and creating a database on Oracle11g and TOAD as a silo, or island, of information. Then, through the ODI integration information tool, this silo can be integrated with other ones belonging to the same company, for example. Finally, the migration of all this information could be uploading to the Oracle Cloud.

7.1 Generation of a database with Oracle 11g and TOAD

7.1.1 Installation of the Oracle Developed Days Virtual Machine

First one you must have Oracle Virtual box. If you don't have it, you can download from:

<https://www.virtualbox.org/>

Also you must download the latest version of Oracle Developed Days from the Oracle web page.

<http://www.oracle.com/technetwork/database/enterprise-edition/databaseappdev-vm-161299.html>

Open Oracle VirtualBox, choose **file** and then **Import Appliance**.

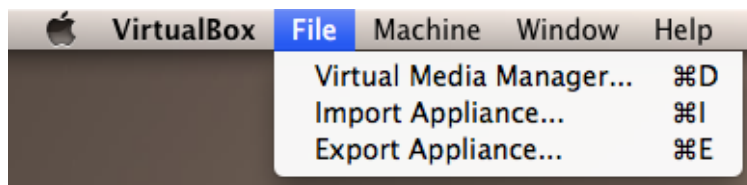


Figure 7 - Importing a new appliance

Now clicking on **Open appliance** and locating your Oracle developed days Virtual Machine file its an .ova extension.

Then click on **Import** and finally be **agree** with Software License Agreement and the importing will be running.

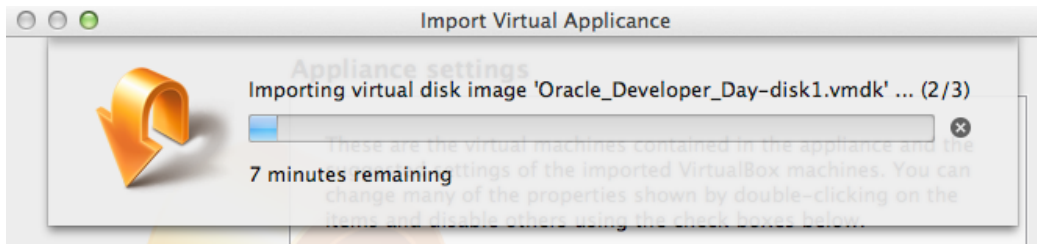


Figure 8 - Running the Importation

When the importation is finished just press start and the virtual machine will be working and now you can work over it, as we will describe in the next point.

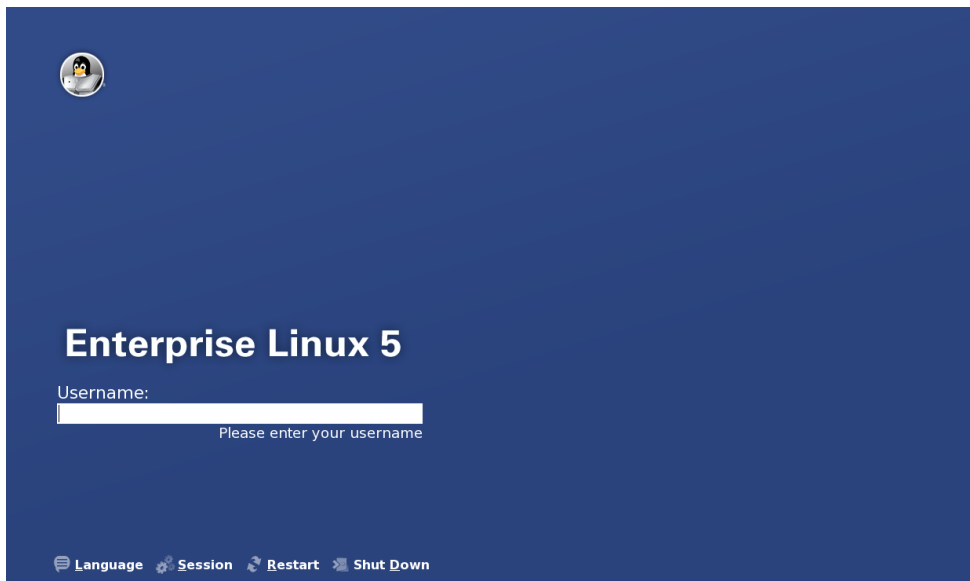


Figure 9 - Oracle Developed Days running

7.1.2 Creating a Database with Oracle SQL Developer

First one you need start the Oracle listener with the following command:

```
emctl start dbconsole
```

Now its time to create a new user for the database with the followings commands:

```
su - oracle
sqlplus / as sysdba

create user [username] identified by [password]
grant connect, resource, create view to [username]
```

Now logout and start a new login with your new user:

```
sqlplus [username]/[password]@ORCL
```

Its time to start the Oracle SQL Developer

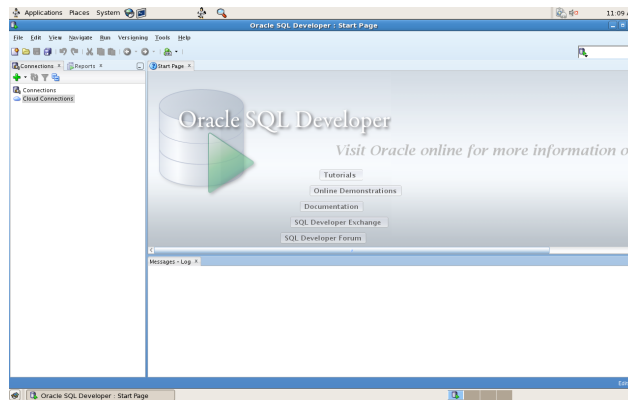


Figure 10 - Oracle SQL Developer main page

In the connections zone, you must choose a new connection:

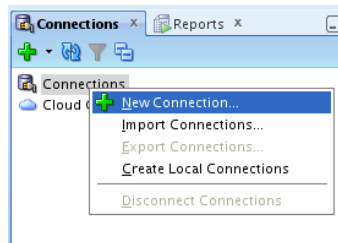


Figure 11 - Creating a new connection

Then you need define a new connection.

- Connection Name: It's a name to identify your connection.
- Username: It's the name that's you create in the last point.
- Password: It's the password that you introduce in the installation point.
- Connection type: Basic for default.
- Role: Default.
- Hostname: If you are working in your own server user, localhost for default.
- Port: port 1521 for default.
- SID: the default is 'xe', but in this case you need change it for 'orcl'

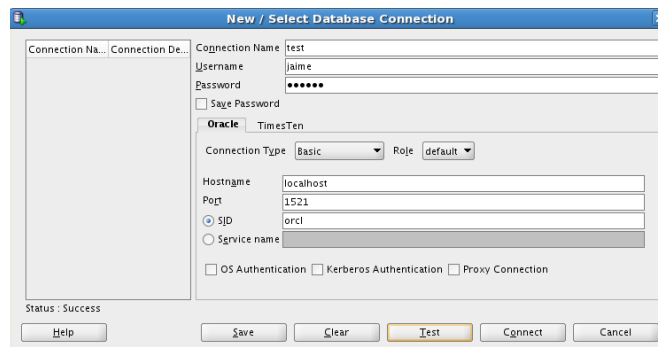


Figure 12 - Testing the new connection

Once the connection is ready, its time of creating the database and inserting the corresponding tables and tuple values.

For practical purposes, we will use the data provided in a practice carried out in the database course "[Sistemas de Información y Bases de Datos Web. Tecnologías de la Información.](#)"

7.1.2.1 Creating tables

```
CREATE TABLE DEPARTAMENTO
(
    ID NUMBER(10) NOT NULL,
    NOMBRE VARCHAR2(50),
    LOCALIDAD VARCHAR2(50),
    PRESUPUESTO NUMBER(7,2),
    DIRECTOR_ID NUMBER(10),
    CONSTRAINT DEPARTAMENTO_PK PRIMARY KEY (ID) USING INDEX
);
```

```
CREATE TABLE EMPLEADO
(
    ID NUMBER(10) NOT NULL,
    NOMBRE VARCHAR2(30),
    APELLIDOS VARCHAR2(60),
    DEPARTAMENTO_ID NUMBER(10),
    CARGO VARCHAR2(20),
    GRADO NUMBER(2) NOT NULL,
    JEFE_ID NUMBER(10),
    FECHA_ALTA DATE,
    SUELDO NUMBER(9,2),
    COMISION NUMBER(9,2),
    CONSTRAINT EMPLEADO_PK PRIMARY KEY (ID) USING INDEX
);
```

```
CREATE TABLE JAIME.RANGO_SUELDO
(
    GRADO    NUMBER(2) NOT NULL,
    SUELDO_MIN  NUMBER(9,2),
    SUELDO_MAX  NUMBER(9,2),
    CONSTRAINT RANGO_SUELDO_PK PRIMARY KEY (GRADO)
);
```


7.1.2.2 Creating the referential integrity rules

```
ALTER TABLE JAIME.DEPARTAMENTO
ADD CONSTRAINT DEPARTAMENTO_DIRECTOR_ID
FOREIGN KEY (DIRECTOR_ID)
REFERENCES JAIME.EMPLEADO (ID);

ALTER TABLE JAIME.EMPLEADO
ADD CONSTRAINT EMPLEADO_GRADO
FOREIGN KEY (GRADO)

ALTER TABLE JAIME.EMPLEADO
ADD CONSTRAINT EMPLEADO_DEPARTAMENTO_ID
FOREIGN KEY (DEPARTAMENTO_ID) REFERENCES JAIME.DEPARTAMENTO(ID);
REFERENCES JAIME.RANGO_SUELDO (GRADO);

ALTER TABLE JAIME.EMPLEADO
ADD CONSTRAINT EMPLEADO_JEFE_ID
FOREIGN KEY (JEFE_ID) REFERENCES JAIME.EMPLEADO(ID);
```

7.1.2.3 Inserting Data

```
INSERT INTO RANGO_SUELDO (GRADO, SUELDO_MIN, SUELDO_MAX)
VALUES (1, 700, 1200);
INSERT INTO RANGO_SUELDO (GRADO, SUELDO_MIN, SUELDO_MAX)
VALUES (2, 1201, 1400);
INSERT INTO RANGO_SUELDO (GRADO, SUELDO_MIN, SUELDO_MAX)
VALUES (3, 1401, 2000);
INSERT INTO RANGO_SUELDO (GRADO, SUELDO_MIN, SUELDO_MAX)
VALUES (4, 2001, 2500);
INSERT INTO RANGO_SUELDO (GRADO, SUELDO_MIN, SUELDO_MAX)
VALUES (5, 2501, 3999);

INSERT INTO DEPARTAMENTO (ID, NOMBRE, LOCALIDAD, PRESUPUESTO, DIRECTOR_ID)
VALUES (1, 'TECNOLOGÍAS DE LA INFORMACIÓN', 'MADRID', '99000,55', null);
INSERT INTO DEPARTAMENTO (ID, NOMBRE, LOCALIDAD, PRESUPUESTO, DIRECTOR_ID)
VALUES (2, 'INVESTIGACION', 'BILBAO', '79000,55', null);
INSERT INTO DEPARTAMENTO (ID, NOMBRE, LOCALIDAD, PRESUPUESTO, DIRECTOR_ID)
VALUES (3, 'VENTAS', 'BARCELONA', '89000,55', null);
INSERT INTO DEPARTAMENTO (ID, NOMBRE, LOCALIDAD, PRESUPUESTO, DIRECTOR_ID)
```

```

VALUES (4, 'PRODUCCION', 'VALENCIA', '90000,55', null);

INSERT INTO EMPLEADO (ID, NOMBRE, APELLIDOS, DEPARTAMENTO_ID, CARGO, Grado,
JEFE_ID, FECHA_ALTA, SUELDO,COMISION)
VALUES (6, 'INES', 'FERNANDEZ SANCHEZ',1, 'PRESIDENTA', 5, 6, TO_Date('17/11/1991',
'DD/MM/YYYY'), 3500, 25);
INSERT INTO EMPLEADO (ID, NOMBRE, APELLIDOS, DEPARTAMENTO_ID, CARGO, Grado,
JEFE_ID, FECHA_ALTA, SUELDO,COMISION)
VALUES (4, 'ESTHER', 'PEÑA MATA', 1,'DIRECTORA', 5, 6, TO_Date('01/05/1991', 'DD/MM/YYYY'),
2850, 5);
INSERT INTO EMPLEADO (ID, NOMBRE, APELLIDOS, DEPARTAMENTO_ID, CARGO, Grado,
JEFE_ID, FECHA_ALTA, SUELDO,COMISION)
VALUES (5,'DANIEL', 'LOPEZ LIRON', 2, 'DIRECTOR', 4, 4, TO_Date('29/06/1992', 'DD/MM/YYYY'),
2500, 25);
INSERT INTO EMPLEADO (ID, NOMBRE, APELLIDOS, DEPARTAMENTO_ID, CARGO, Grado,
JEFE_ID, FECHA_ALTA, SUELDO,COMISION)
VALUES (1, 'CARLOS', 'ALONSO GARCIA', 2,'COMERCIAL', 1, 5,
TO_Date('20/02/2010','DD/MM/YYYY'), 1100, 15);
INSERT INTO EMPLEADO (ID, NOMBRE, APELLIDOS, DEPARTAMENTO_ID, CARGO, Grado,
JEFE_ID, FECHA_ALTA, SUELDO,COMISION)
VALUES (2, 'ANDRES', 'CREMADES MOLINERO', 1,'DIRECTOR',5, 6,
TO_Date('22/02/1998','DD/MM/YYYY'), 3990, 10);
INSERT INTO EMPLEADO (ID, NOMBRE, APELLIDOS, DEPARTAMENTO_ID, CARGO, Grado,
JEFE_ID, FECHA_ALTA, SUELDO,COMISION)
VALUES (3, 'RAUL', 'FERNANDEZ HERRANZ', 1,'DIRECTOR',5, 6, TO_Date('22/02/1998',
'DD/MM/YYYY'), 3990, 20);
INSERT INTO EMPLEADO (ID, NOMBRE, APELLIDOS, DEPARTAMENTO_ID, CARGO, Grado,
JEFE_ID, FECHA_ALTA, SUELDO,COMISION)
VALUES (7, 'JOAQUIN', 'CALLEJA ARCAZ', 1,'DIRECTOR',5, 6, TO_Date('22/02/1998',
'DD/MM/YYYY'), 3990, 25);

```

7.1.2.4 Creating a Backup SQL Script

In this topic you will see how to make a full backup of your database.

Click on **Tools** then in **Database Export**

In the export wizard window choose your connection and keep the rest of the configurations as default. Click **Next**.

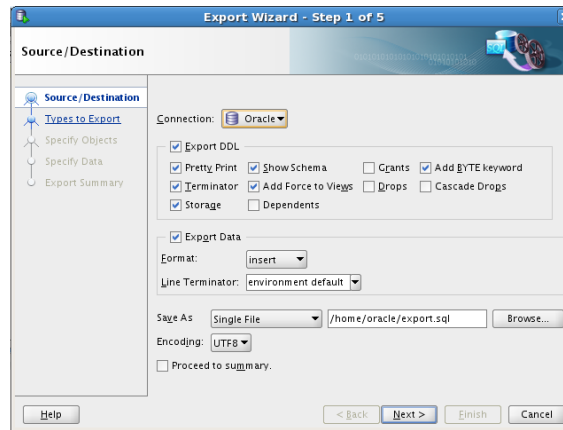


Figure 13 - Creating a SQL Script with the export wizard

In the step two, retain the default values of all object types selected. Click **Next**.

For the steps 3 - 5 you only need to press **Next**. Then the only thing that remains is to wait to finish the script generation. Observe that after exporting, the export.sql file automatically opens in SQL Worksheet.

You can scroll down to view the entire data.

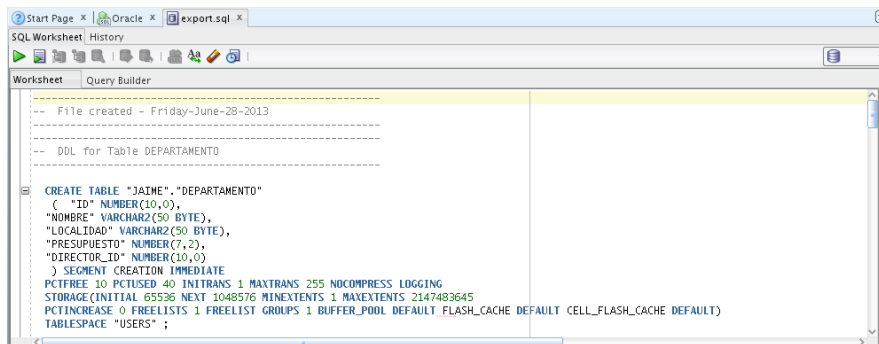


Figure 14 - Export SQL File

7.2 Migration to Oracle Cloud

7.2.1 Accessing the Cloud

Before to start, we need to create an account to use the service of Oracle Cloud as a Trial version, and we need to go, to this web site.

<https://cloud.oracle.com/mycloud/f?p=service:home:0>



Figure 15 – Oracle Cloud Website

Now, in the page of Oracle Cloud PaaS and SaaS Trials, we choose Database in the PaaS section.

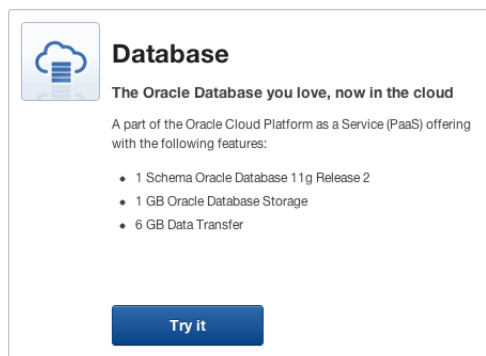


Figure 16 – Choosing Oracle Database in the Cloud

Now you need to sign in, in Oracle with your own credentials, after that you must need to complete the form and you will receive an email with your Account Administrator key and Identity Domain Administrator key for the Oracle Cloud service. (This process takes around 24hrs for the checkout)

Once you have it you must go to your account page

<https://cloud.oracle.com/mycloud/f?p=5300>

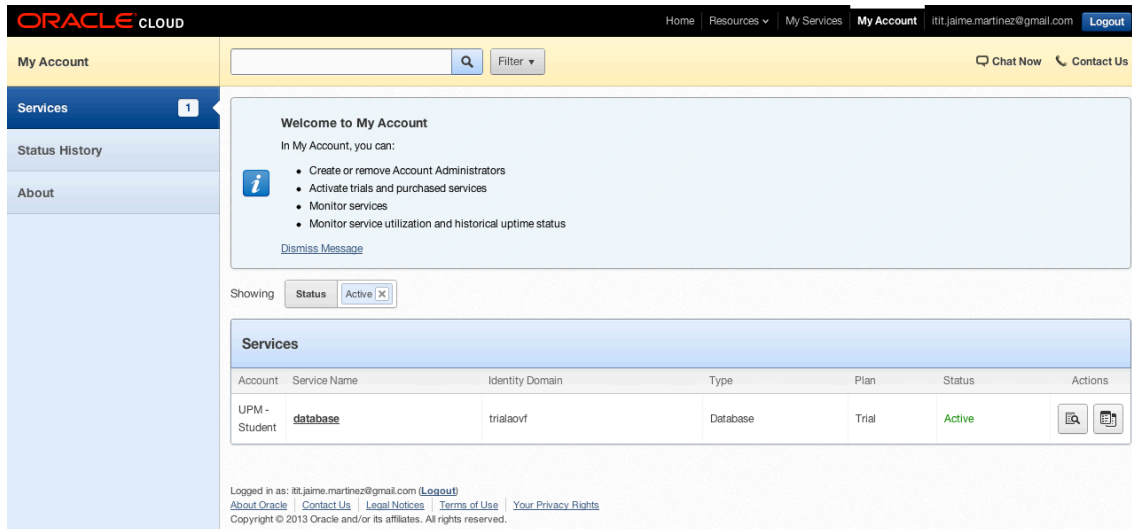


Figure 17 - My Account page in Oracle Cloud

7.2.2 Loading the Database in the Cloud

To upload your database you must access to **My Services** category in Oracle Cloud and then in the **Actions** zone, click on **Launch Service**.

Then choose **SQL Workshop >> SQL Commands**, and in the text box copy and paste the code that you create in the point 7.1.2.4 of this work and make a click on **Run** to see the results.



Figure 18 - SQL Commands

After that returns to the **SQL Workshop** main page and go to **Object Browser** to see the tables and values has you been uploaded.

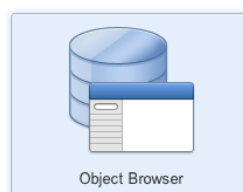


Figure 19 - Object Browser

7.2.3 Creating an application in the Cloud

In order to create a new application you must access to **My Services** category in Oracle Cloud and then in the **Actions** zone, clicking on **Launch Service**.

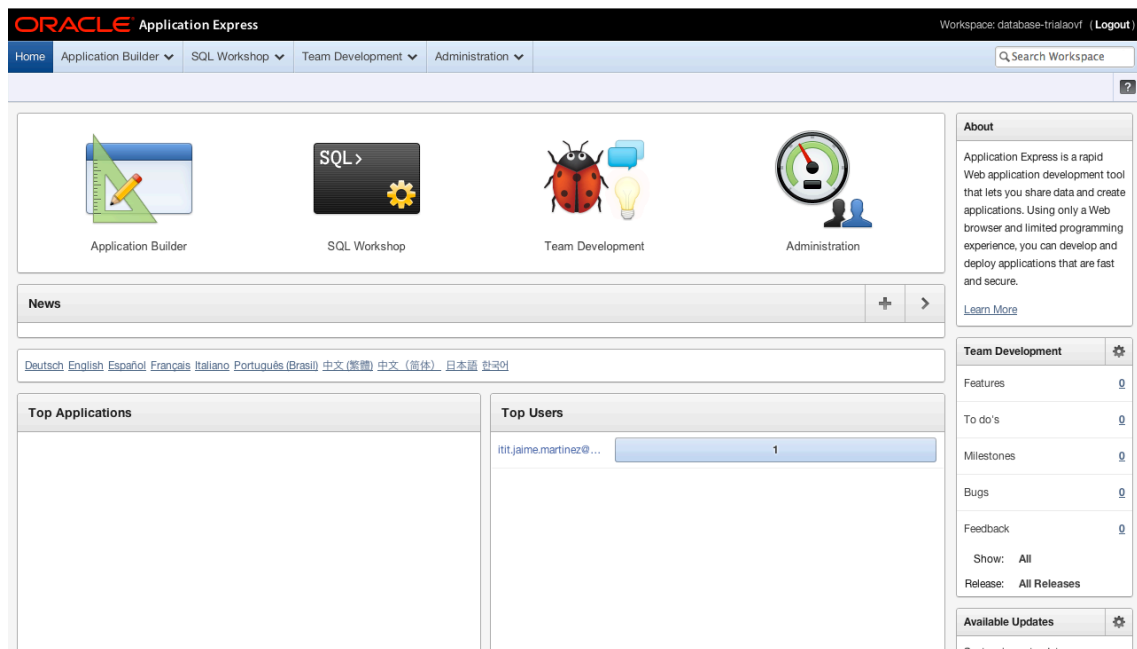


Figure 20 – Database Workspace

Select **Application Builder**, once being in this screen you click on the button to create, select **Database** and next.

Left the **ID** of the application by default, and just change the name by which you prefers, the terms of this **scheme** is preserved since we use a trial version, at the point of creating options you select **Include Home Page**, and finally the user interface will be given desktop.

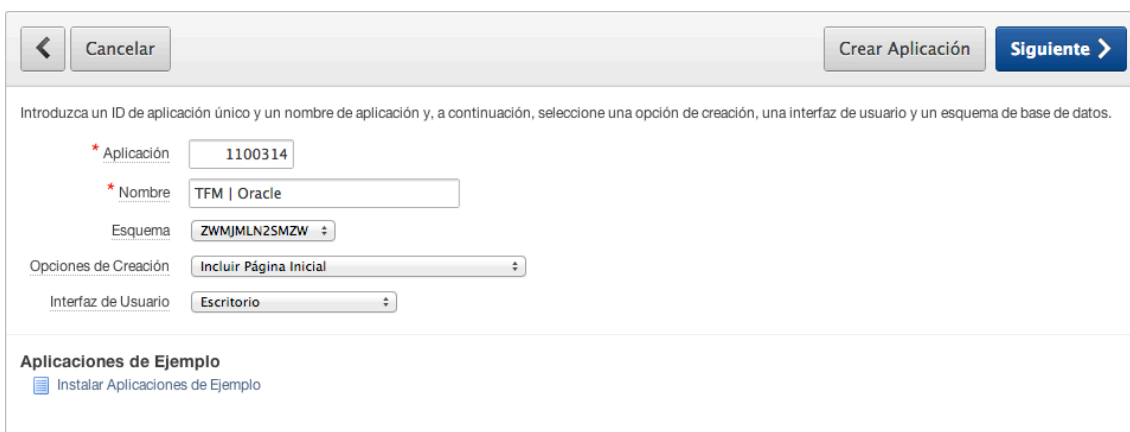


Figure 21 – Creating a new Application

7.2.3.1 How show tables

Once created our base cloud application, we will enter it and create a new page by clicking on the button.

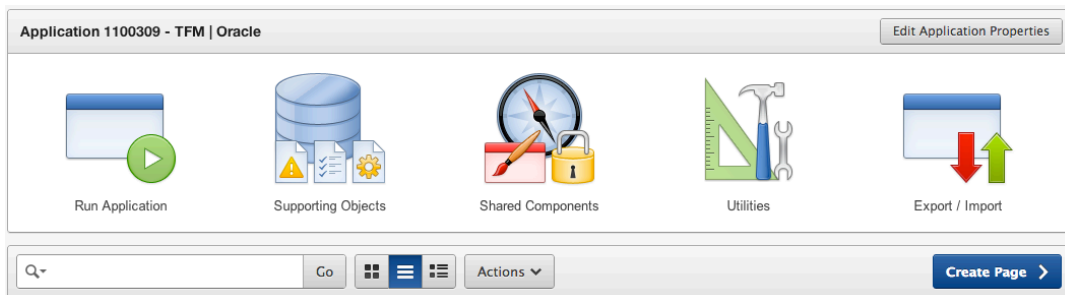


Figure 22 - Creating a view

It will select **Report** (Figure 23), and then **Classic Report** (Figure 24), be preserved most fields with default values and you only need to modify the names of the product and the region (Figure 25). Next

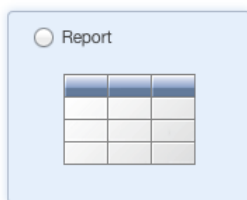


Figure 23 - Report Icon

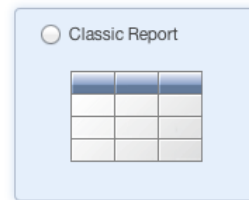


Figure 24 - Classic Report Icon

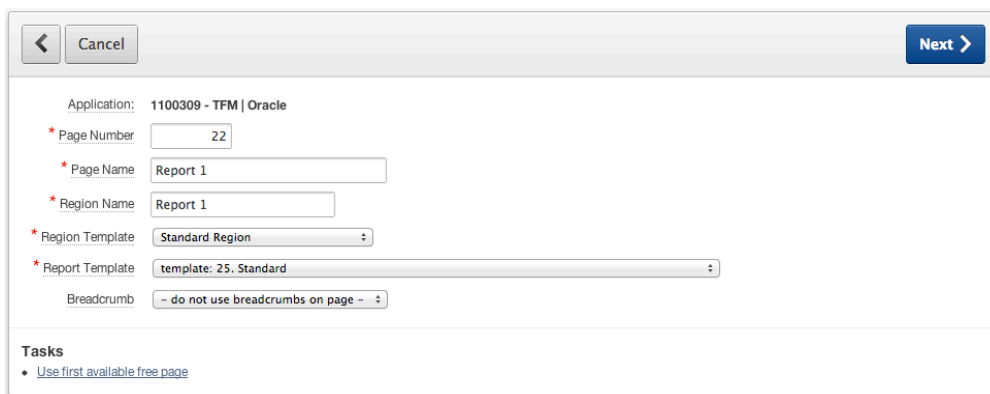


Figure 25 - Page Attributes

In tab options will choose **Do not use tabs**. Next



Figure 26 - Tab Options

Now will choose the table we want to show. Next

Figure 27 - Report Source

In the next point will keep the default values. Next

Figure 28 - Report Attributes

And now just press on Create, we will follow the same steps if we want to show more tables.

Figure 29 - Confirm

7.2.3.2 Creating links in the main page

To create the links, we need to place on the main page of our application.

Page	Name	Updated	Updated By	Page Type	User Interface	Group	Lock	Run
1	Inicio	40 hours ago	itit.jaime.martinez@gmail.com	Home	Desktop	Unassigned		

Figure 30 - Main Page

After entering the edit mode we select the **Page Rendering** box and then we select the section of regions.

Figure 31 - Regions Section

We click on the icon **create** in that section, then select **create HTML** (Figure 32), and in this stage you only need to create an object name and then create a region (Figure 33).

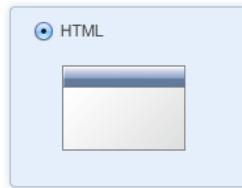


Figure 32 - Creating a HTML Region

Figure 33 - Display Attributes

Now, in the buttons section of the Page Rendering Box, you need to create a new button (Figure 34). Then, into the task part, make a click on **Create Multiple Buttons** (Figure 35), only fill the **Name** and **Label** boxes and click on **Create Buttons**.

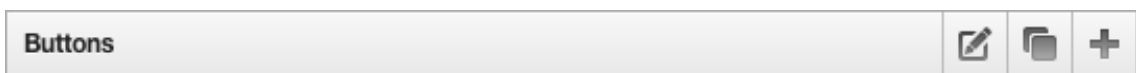


Figure 34 - Buttons Section

- Tasks**
- [Create Multiple Buttons](#)

Figure 35 - Tasks

Sequence	Name	Label	Position	Attributes
10			Bottom of Region	

Figure 36 - Creating Multiples Buttons

Edit the button you have been created and in the display region choose the region you have been created in the last point.

The 'Displayed' configuration panel contains the following settings:

- Sequence: 20
- Display in Region: Data Query (20)
- Button Position: Region Template #DEFAULT# Position
- Button Alignment: Left

Figure 37 - Display Region

In the action zone when the button is clicked choose Redirect to Page in this Application and choose the page that you want to link and finally apply the changes.

The 'Action When Button Clicked' configuration panel contains the following settings:

- Action: Redirect to Page in this Application
- Page: 7
- reset pagination for this page:

Figure 38 - Action Button Region

7.2.3.3 Creating a form with privilege of Creation, Update and Delete

To create this form, we need to be on the main page of our application and create a new **form** in the **create page** section (Figure 39), and then choose **form on a table with report** (Figure 40).

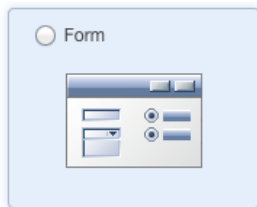


Figure 39 - Form Icon

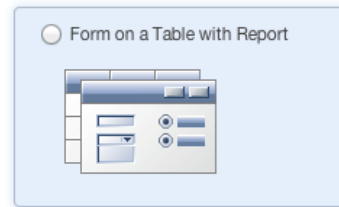


Figure 40 - Form with Report Icon

Here only change the page and region name (Figure 41), then choose the table source (Figure 42). In the tab option select **do not use tabs** (Figure 43), in the next page select only the columns you want to use (Figure 44).

The 'Report Page' configuration dialog contains the following settings:

- Implementation: Interactive
- Page Number: 24
- Page Name: Report 1
- Region Title: Report 1
- Region Template: Interactive Report Region
- Report Template: template: 25: Standard
- Pagination Size: 15
- Breadcrumb: do not add breadcrumb region to page

Figure 41 - Report Page

This wizard builds a two page Report and form combination on a single table or view. The first page enables users to specify the row to be updated. The second page provides users with ability to update the selected table or view. Select the schema owner who owns the database table or view.

* Table / View Owner: ZWMJMLN2SMZW

* Table / View Name: - Select Value -

Figure 42 - Data Source

Both report and form pages will use the same tab definition

Page: 24

Tab Options:

- Do not use tabs
- Use an existing tab set and create a new tab within the existing tab set.
- Use an existing tab set and reuse an existing tab within that tab set.

> Tabs

Figure 43 - Tabs

Select the columns to include in the Report page. Define an optional WHERE clause to restrict the result set.

Page: 24

Owner: ZWMJMLN2SMZW

Table Name: EMPLEADO

Use User Interface Defaults: No Yes

* Select Column(s)

- 1.ID (Number)
- 2.NOMBRE (Varchar2)
- 3.APELLIDOS (Varchar2)
- 4.DEPARTAMENTO_ID (Num)
- 5.CARGO (Varchar2)
- 6.GRADO (Number)
- 7.JEFE_ID (Number)
- 8.FECHA_ALTA (Date)
- 9.SUELDO (Number)
- 10.COMISION (Number)

Figure 44 - Report Columns

Then click next, next and next; until it ask you again what columns you want to use for the form page. Next.

Now the system asks you what privileges you want to bring for this form, when you choose your privileges making a click on Next, and then in Create, and it is done.

Identify the process options for this form page. For example, to prevent users from deleting from this form page, select No for the delete option.

Form Page: 25

Owner: ZWMJMLN2SMZW

Table Name: EMPLEADO

Insert: Yes

Update: Yes

Delete: Yes

Figure 45 - Action Button Region

7.2.4 Execution of the application on the Oracle Cloud

In the application Builder page, you only need to make a click on Run of your application, and then you can see how does it works in real time.

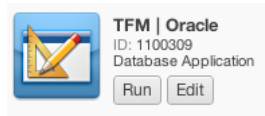


Figure 46 - Running the application

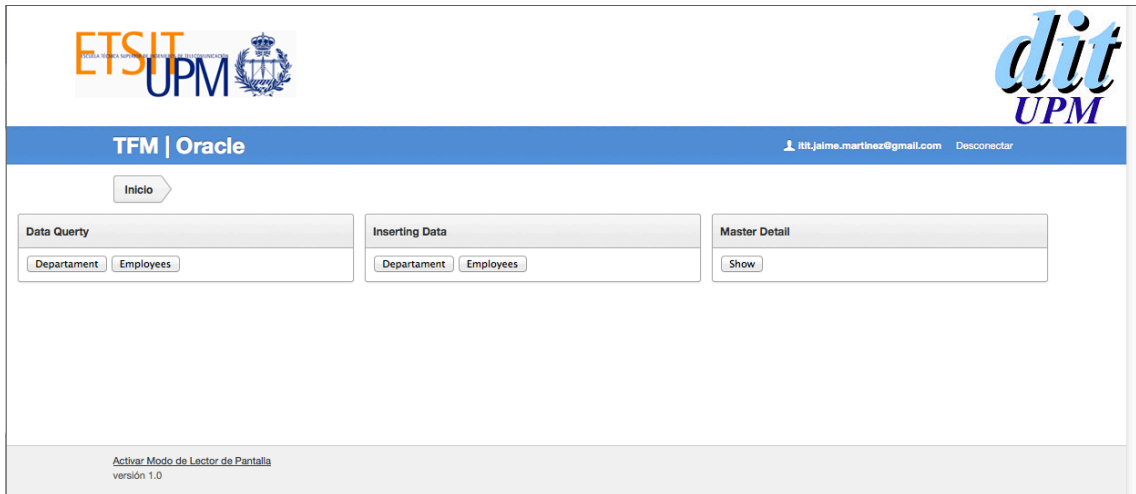


Figure 47 - Main page of the application

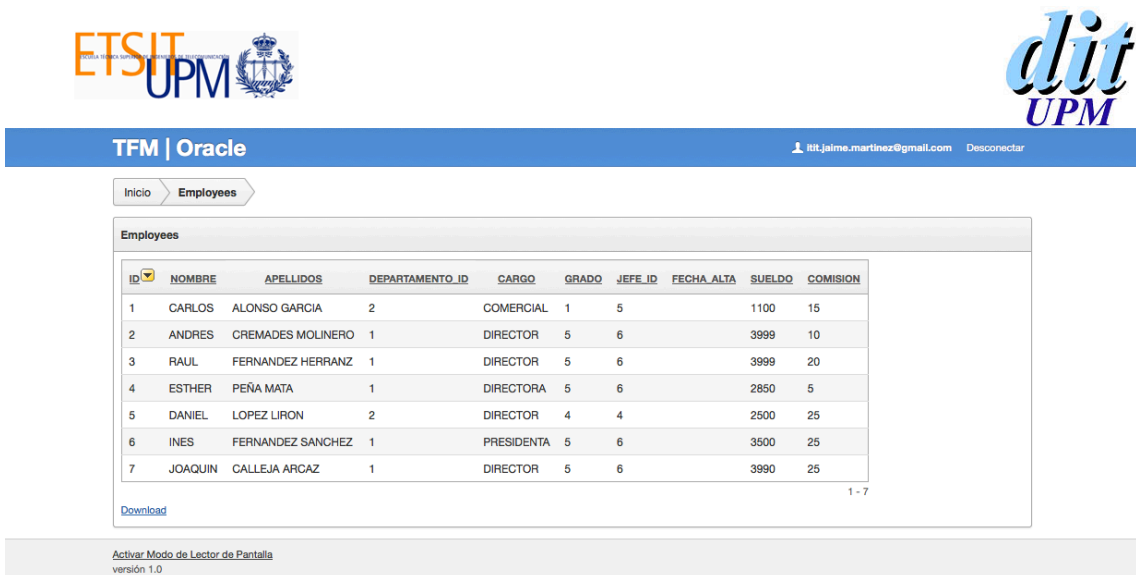


Figure 48 - Making a Query

The figure 50 finally shows how we can see many tuples on our migrated database.

8 Conclusions

On the one hand, the first part of this work highlights the necessary technologies to carry out the migration, in order to store the information coming from a local database server into another server located in the cloud.

On the other hand, the second part of this work was carried out an academical demonstration of the mentioned tools in the first part, in order to create an application using our data migrated from the local server.

With this work, we could demonstrate the ease way of creating many high-performance database applications and their capabilities. In such a way, our company or organization, can offer its operations and functionalities through the proper services it wants to offer in the Cloud Computing.

Generally speaking, the main advantages of using such a cloud services can be emphasized as follows:

- The ease of migration
- Easy way to develop applications
- The use of standards that will allow in some future the scalability for it management
- The support provided by Oracle
- The reliability and security of the information.

9 Bibliography

- Laszewski, T., & Nauduri, P. (2011). *Migrating to the Cloud: Oracle Legacy Client/Server Applications*. Syngress.
- Co., O. (s.f.). *Oracle Cloud*. Obtained from Oracle Cloud: <https://cloud.oracle.com/mycloud/f?p=service:home:0>
- Co., O. (s.f.). *Oracle Data Integrator*. Obtained from Oracle: <http://www.oracle.com/technetwork/middleware/data-integrator/overview/index.html>
- Scalzo, B., & Hotka, D. (2003). *TOAD® Handbook*. Sams.
- Co., O. (s.f.). *Developer Days - Hands-on Database Application Development*. Obtained from Oracle: <http://www.oracle.com/technetwork/database/enterprise-edition/databaseappdev-vm-161299.html>