|  |
| --- |
| , RD Dep.  MTN.\*NIX.07 Oracle DB. Introduction to DWH |
| MTN.\*NIX.07 Labs - Data Warehouse Architecture |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1.0 | Initial status of document | [**Kiryl Bucha**](mailto:Kiryl_bucha@epam.com) | 16-JAN-2012 |  |  |
|  |  |  |  |  |  |

*Contents*

[1. Prerequisites Task Information 3](#_Toc320624559)

[1.1. Passwords Index 3](#_Toc320624560)

[1.2. Folder Paths Index 3](#_Toc320624561)

[2. Data Warehouse Architecture 3](#_Toc320624562)

[2.1. Task 01: CREATE Schema of simple Data Warehouse Architecture 3](#_Toc320624563)

[2.2. Task 02: CREATE Storage Level 5](#_Toc320624564)

[2.3. Task 03: CREATE Data warehouse Cleansing Level 5](#_Toc320624565)

[2.4. Task 04: CREATE Data warehouse Start Cleansing Level 6](#_Toc320624566)

[2.5. Task 05: CREATE Start Cleansing Level 6](#_Toc320624567)

[2.6. Task 06: CREATE Data warehouse Start Level and Data Marts 6](#_Toc320624568)

[3. Star – Business analyses task 6](#_Toc320624569)

[3.1. Task 07: CREATE Prepared Start Objects 6](#_Toc320624570)

[3.2. Task 09: DataFlow Diagram 7](#_Toc320624571)

[3.3. Task 09: Grants Object Privileges 7](#_Toc320624572)

# Prerequisites Task Information

## Passwords Index

|  |  |  |
| --- | --- | --- |
| Password Group | Login Name | Password |
| Operation System | root | “rootadmin” |
|  | oracle | “oracleadmin” |
|  |  |  |
| Oracle System | sys | “sysadmin” |
|  | system | “sysadmin” |
|  |  |  |
| Oracle Users | All DB users | “%PWD%” |
|  |  |  |
|  |  |  |

## Folder Paths Index

|  |  |  |
| --- | --- | --- |
| Path Group | Path Description | Path |
| Operation System | Oracle RDBMS – BIN | /oracle/app/oracle |
|  | Oracle Inventory | /oracle/app/oraInventory |
|  | Oracle Database Storage | /oracle/oradata |
|  | Oracle Install Directory | /oracle/install |
| Oracle | ORACLE\_BASE | /oracle/app/oracle |
|  | ORACLE\_HOME | $ORACLE\_BASE/product/11.2 |
|  |  |  |
| FTP | ftp Incoming Folder | **/ftp/incoming** |
|  |  |  |
|  |  |  |

# Data Warehouse Architecture

## Task 01: CREATE Schema of simple Data Warehouse Architecture

**The Main Task** is to create Diagram to describe yours simple Data Warehouse.

According yours Business Task (Solution Proposal) prepare naming conversation table for all levels of Data Warehouse.

**Task Results:**

Create document or Add chapter to Solution Concept, which contained next:

* + Data Warehouse Architecture diagram



* + Name Conversation table

|  |  |  |  |
| --- | --- | --- | --- |
| Level Type | Object Name | Tablespace | Desctiption |
| Storage level  SA\_\* | U\_SA\_CUSTOMERS | ts\_sa\_customers\_data\_01  (AUTOALLOCATE,  SEGMENT SPACE MANAGEMENT AUTO,  LOGGING,  Size 150M,  Autoextend clause ON next 50M) | Loading from db storage system. Contains Customer, products, channels information. |
| U\_SA\_SALES | ts\_sa\_SALES\_data\_01  (AUTOALLOCATE,  SEGMENT SPACE MANAGEMENT AUTO,  LOGGING,  Size 150M,  Autoextend clause ON next 50M) | Loading from flatfile storage system. Contains information about sales. |
| DW - Cleansing Level | U\_DW\_CL | ts\_DW\_CL  (AUTOALLOCATE,  SEGMENT SPACE MANAGEMENT AUTO,  NOLOGGING,  Size 100M,  Autoextend clause ON next 50M) | LOADING from stage level system. Contains all information and prepare it for further usage (cleaning it). |
| DW – Level | U\_DW\_DATA | ts\_DW\_DATA\_01  (AUTOALLOCATE,  SEGMENT SPACE MANAGEMENT AUTO,  LOGGING,  Size 150M,  Autoextend clause ON next 50M) | LOADING data from cleansing system. Contains clean information tending to the 3rd normal form ready for preparing star schema. |
| DW– Prepare Star Cleansing Level | U\_DW \_STR\_CLS | ts\_ DW \_STR\_CLS  (AUTOALLOCATE,  SEGMENT SPACE MANAGEMENT AUTO,  NOLOGGING,  Size 150M,  Autoextend clause ON next 50M) | LOADING data from DW system. Contains views merging objects from DW level. |
| STAR - Cleansing | U\_SAL\_CL | ts\_SAL\_CL  (AUTOALLOCATE,  SEGMENT SPACE MANAGEMENT AUTO,  NOLOGGING,  Size 150M,  Autoextend clause ON next 50M) | LOADING data from DW\_CL system. Contains views from previous level but clean any redundancy. |
| STAR – Level | U\_STR\_DATA | ts\_ STR\_DATA  (AUTOALLOCATE,  SEGMENT SPACE MANAGEMENT AUTO,  LOGGING,  Size 200M,  Autoextend clause ON next 50M) | LOADING data from star cleansing system. Contains information about facts. |
| U\_STR\_REFERENCES | ts\_ STR\_REFERENCES  (AUTOALLOCATE,  SEGMENT SPACE MANAGEMENT AUTO,  LOGGING,  Size 150M,  Autoextend clause ON next 50M) | LOADING data from star cleansing system. Contains information about dimensions. |

## Task 02: CREATE Storage Level

**The Main Task** is to create Physical Objects According Diagram from task 01

**Task Results:**

Create scripts and edit scheme, which contained next:

* + Data Warehouse Architecture – Storage level Objects
  + Scripts put on Git

## Task 03: CREATE Data warehouse Cleansing Level

**The Main Task** is to create Physical Objects According Diagram from task 01

**Task Results:**

Create scripts and edit scheme, which contained next:

* + Data Warehouse Architecture – Data warehouse cleansing level Objects
  + Scripts put on Git

## Task 04: CREATE Data warehouse Start Cleansing Level

**The Main Task** is to create Physical Objects According Diagram from task 01

**Task Results:**

Create scripts and edit scheme, which contained next:

* + Data Warehouse Architecture – Data warehouse Star Cleansing level Objects
  + Scripts put on Git

## Task 05: CREATE Start Cleansing Level

**The Main Task** is to create Physical Objects According Diagram from task 01

**Task Results:**

Create scripts and edit scheme, which contained next:

* + Data Warehouse Architecture –Star Cleansing level Objects
  + Scripts put on Git

## Task 06: CREATE Data warehouse Start Level and Data Marts

**The Main Task** is to create Physical Objects According Diagram from task 01

**Task Results:**

Create scripts and edit scheme, which contained next:

* + Data Warehouse Architecture – Data warehouse Star and Data Marts level Objects
  + Scripts put on Git

# Star – Business analyses task

## Task 07: CREATE Prepared Start Objects

**The Main Task** is to create Physical Objects according your Business Star, that was developed on Labwork 6, labwork 7.

**Task Results:**

Create scripts and edit scheme, which contained next:

* + Data warehouse Star and Data Marts level Objects
  + Scripts put on Git

## Task 09: DataFlow Diagram

**The Main Task** is to create DataFlow Diagram to describe refresh process of yours Business STAR

**Task Results:**

Create DataFlow Diagram:

* + Add chapter to Solution Concept with DataFlow Diagrams

## Task 09: Grants Object Privileges

**The Main Task** is to grant all required Object Privileges to cleansing layers.

**Task Results:**

* + Scripts put on Git
  + Create screenshot with all granted privileges to DW\_CL, SAL\_CL\_DW, SAL\_CL layers.

|  |  |  |  |
| --- | --- | --- | --- |
| Level Type | Object Name | Tablespace | Granted privileges |
| Storage level  SA\_\* | U\_SA\_CUSTOMERS | ts\_sa\_customers\_data\_01 | Loading data from db storage system customer’s, product’s, channel’s information (Read only). Hold it in its tablespace (create tables, create views on this dirty tables and push data upstairs (update upper user(ts\_DW\_CL)). |
| U\_SA\_SALES | ts\_sa\_SALES\_data\_01 | Loading data from flatfile (Read only) storage system the most secure information about sales. Hold it in its tablespace (create tables and views push data to the upper level (update objects ts\_DW\_CL) user). |
| DW - Cleansing Level | U\_DW\_CL | ts\_DW\_CL | LOADING data from stage level system (Read objects ts\_sa\_SALES\_data\_01 and ts\_sa\_customers\_data\_01). Hold it in its tablespace (create tables and views and push it upstairs (update objects ts\_DW\_DATA\_01) ). |
| DW – Level | U\_DW\_DATA | ts\_DW\_DATA\_01 | LOADING data from DW - Cleansing Level (Read objects ts\_DW\_CL). Hold it in its tablespace (create tables and views and push it upstairs (update objects ts\_ DW \_STR\_CLS) ). |
| DW– Prepare Star Cleansing Level | U\_DW \_STR\_CLS | ts\_ DW \_STR\_CLS | LOADING data from DW – Level (Read objects ts\_DW\_DATA\_01). Hold it in its tablespace (create tables and views and push it upstairs (update objects ts\_SAL\_CL) ). |
| STAR - Cleansing | U\_SAL\_CL | ts\_SAL\_CL | LOADING data from DW– Prepare Star Cleansing Level (Read objects ts\_ DW \_STR\_CLS). Hold it in its tablespace (create tables and views and push it upstairs (update objects ts\_ STR\_DATA) ). |
| STAR – Level | U\_STR\_DATA | ts\_ STR\_DATA | LOADING data from STAR - Cleansing Level (Read objects ts\_SAL\_CL). Hold it in its tablespace (create tables and views ). |
| U\_STR\_REFERENCES | ts\_ STR\_REFERENCES | LOADING data from STAR - Cleansing Level (Read objects ts\_SAL\_CL). Hold it in its tablespace (create tables and views ). |

Locally managed tablespaces track all extent information in the tablespace itself, using bitmaps, resulting in the following benefits:

* Improved concurrency and speed of space operations, because space allocations and deallocations predominantly modify locally managed resources (bitmaps stored in header files) rather than requiring centrally managed resources such as enqueues;

* Improved performance, because recursive operations that are sometimes required during dictionary-managed space allocation are eliminated;

* Readable standby databases are allowed, because locally managed temporary tablespaces (used, for example, for sorts) are locally managed and thus do not generate any undo or redo.

* Simplified space allocation--when the AUTOALLOCATE clause is specified, appropriate extent size is automatically selected

* Reduced user reliance on the data dictionary because necessary information is stored in file headers and bitmap blocks

When you create a locally managed tablespace SEGMENT SPACE MANAGEMENT clause allows you to specify how free and used space within a segment is to be managed. Your choice is:

* AUTO

This keyword tells Oracle that you want to use bitmaps to manage the free space within segments. A bitmap, in this case, is a map that describes the status of each data block within a segment with respect to the amount of space in the block available for inserting rows. As more or less space becomes available in a data block, its new state is reflected in the bitmap. Bitmaps allow Oracle to manage free space more automatically, and thus, this form of space management is called automatic segment-space management.

Automatic segment-space management is a simpler and more efficient way of managing space within a segment. It completely eliminates any need to specify and tune the PCTUSED, FREELISTS, and FREELISTS GROUPS storage parameters for schema objects created in the tablespace. If such attributes should be specified, they are ignored.

Automatic segment-space management delivers better space utilization than manual segment-space management, and it is self tuning in that it scales with increasing the number of users, as well as instances. For many standard workloads, application performance when using automatic segment space management is better than the performance of a well tuned application using manual segment-space management.

Coalescing free space is not necessary for locally managed tablespaces because bitmaps automatically track adjacent free space.

Controlling the Writing of Redo Records

We specify NOLOGGING mode for CLS level, improve performance (decrease resource usage) and reduce space usage.

We specify LOGGING mode to improve safety of our data on the critical areas.