UNIVERSITÉ MOHAMMED V DE RABAT

Faculté des Sciences

Département d'Informatique

Master Informatique Appliquée Offshoring (IAO)

project

Intitulé :

Business Intelligence : Oracle SQL BI

Présenté par :

MONCEF BAHJA

Supervised by :

By Mr. BENKHALIFA Mohammed

Année Universitaire 2024-2025

Abstract

This study examines the fundamentals of Business Intelligence (BI) technologies and their integration through Oracle Data Integrator (ODI). After defining BI and explaining its meaning, we will summarize the common use cases in data analysis. Key concepts such as the Data Warehouse (DWH), Data Mart (DM) and Data Integration are discussed in detail, as well as popular BI tools. The document focuses on the ETL (Extract, Transform, Load) process, providing a comparison between ETL and ELT, and presenting the architecture and functionality of ODI. In addition, ODI Studio is being studied, with a focus on modules such as Topology, Designer, Operator and Security.Concrete demonstration of the integration of ODI is proposed, which deals with database structure and data stocks, as well as Power BI connection and visualization methods. This document is both an introductory guide to BI and a detailed reference for using the ODI for data integration and visualization.

List of Abbreviations

0.1 ODI : Oracle data integrator

0.2 BI :Business Intelligence

0.3 ETL :Extraction, Transformation, Load

0.4 ELT :Extraction, Load, Transformation

0.5 DWH : Data Warehouse

0.6 EDW : enterprise data warehouse

0.7 DM :Data Mart

0.8 KM :Knowledge Module

0.9 SQL :Structured Query Language

Table des matières

Introduction

Most companies track and record thousands of transactions daily. Not just customer purchases which might include information such as the customer, the products/items sold, the store from which the purchase was made, and the date and time of the purchase – but also transactions such as warehouse activity, inventory purchases, employee hours and time off, and daily operating costs.

In fact, most companies are virtually swimming in data. If only there was a way to collect this data in one place and make sense of it all with a simple report (or set of reports). Companies who can successfully extract pertinent data from their mountain of information are gaining unique perspectives of their business, enabling them to become leaner and more competitive.

The process of collecting, organizing, and analyzing business data and turning it into useful and actionable information is commonly referred to as Business Intelligence. With business intelligence, companies have greater insight into their organization, yielding new opportunities, corrections to existing procedures or processes, competitive advantages, and more, including the ability to :

- Identify top-selling products by region, store, or sales person.
- Identify trends, both good and bad, early on.

- Generate ad-hoc financial reports.
- Track competitors in their area.
- Compare information about customers, products, prices, and costs over time

Chapitre 1

BI technologies

1.1 Business Intelligence

Business intelligence (BI) is a technology-driven process for analyzing data and presenting actionable information to help executives, managers and other corporate end users make informed business decisions.

BI encompasses a wide variety of tools, applications and methodologies that enable organizations to collect data from internal systems and external sources ; prepare it for analysis ; develop and run queries against that data ; and create reports, dashboards and data visualizations to make the analytical results available to corporate decision-makers, as well as operational workers.

1.2 Why do we need a Business Intelligence Solution ?

With the amount of data stored by companies growing exponentially, it is no surprise that finding the right data management solution continues to show up on the priority list of Chief Information Officers (CIO). Data has to be secure and distributed efficiently for important up-to-date business decisions.

Companies need to translate data into information to plan for future business strategies. For most companies, valuable data is stored in massive spreadsheets or servers. Ideally, this data should provide you with information on sales trends, consumer behavior and resources allocation. Company data can indicate the viability of your product and help in the planning of your future growth. Hence data can help maximize revenues and reduce costs.

A Business Intelligence (BI) solution helps in producing accurate reports by extracting data directly from your data source. With Business Intelligence solutions today eliminate the time consuming task of consolidating data manually. Since BI tools can produce recent data, it allows managers to monitor businesses in real-time. A BI solution provides realtime reports directly to managers on-demand from any location. This helps to reduce the scope of error by providing managers with accurate data to make better decisions on what is happening now and to forecast for the future. BI solutions also focus on providing data security by using existing established security infrastructures to keep data private

Therefore if you are looking for a way to gain actionable insights from your huge piles of data, and to speed up your data extraction process, a BI solution such as Power BI is your best choice

1.3 Data Warehouse

In computing, a data warehouse (DWH), also known as an enterprise data warehouse (EDW), is a system used for reporting and data analysis, and is considered a core component of business intelligence. DWs are central repositories of integrated data from one or more disparate sources. They store current and historical data in one single place that are used for creating analytical reports for workers throughout the enterprise.

The data stored in the warehouse is uploaded from the operational systems (such as marketing or sales). The data may pass through an operational data store and may require data cleansing for additional operations to ensure data quality before it is used in the DW for reporting

1.4 Data Mart.

A data mart is a structure / access pattern specific to data warehouse environments, used to retrieve client-facing data. The data mart is a subset of the data warehouse and is usually oriented to a specific business line or team. Whereas data warehouses have an enterprise-wide depth, the information in data marts pertains to a single department. In some deployments, each department or business unit is considered the owner of its data mart including all the hardware, software and data. This enables each department to isolate the use, manipulation and development of their data. In other deployments where conformed dimensions are used, this business unit ownership will not hold true for shared dimensions like customer,product, etc...

1.5 Data Integration

Integrating data and applications throughout the enterprise, and presenting them in a unified view is a complex proposition. Not only are there broad disparities in technologies, data structures, and application functionality, but there are also fundamental differences in integration architectures. Some integration needs are Data Oriented, especially those involving large data volumes. Other integration projects lend themselves to an Event Driven Architecture (EDA) or a Service Oriented Architecture (SOA), for asynchronous or synchronous integration.

Data Integration ensures that information is timely, accurate, and consistent across complex systems. Although it is still frequently referred as Extract-Load-Transform (ETL) - Data Integration was initially considered as the architecture used for loading Enterprise Data Warehouse systems - data integration now includes data movement, data synchronization, data quality, data management, and data services.

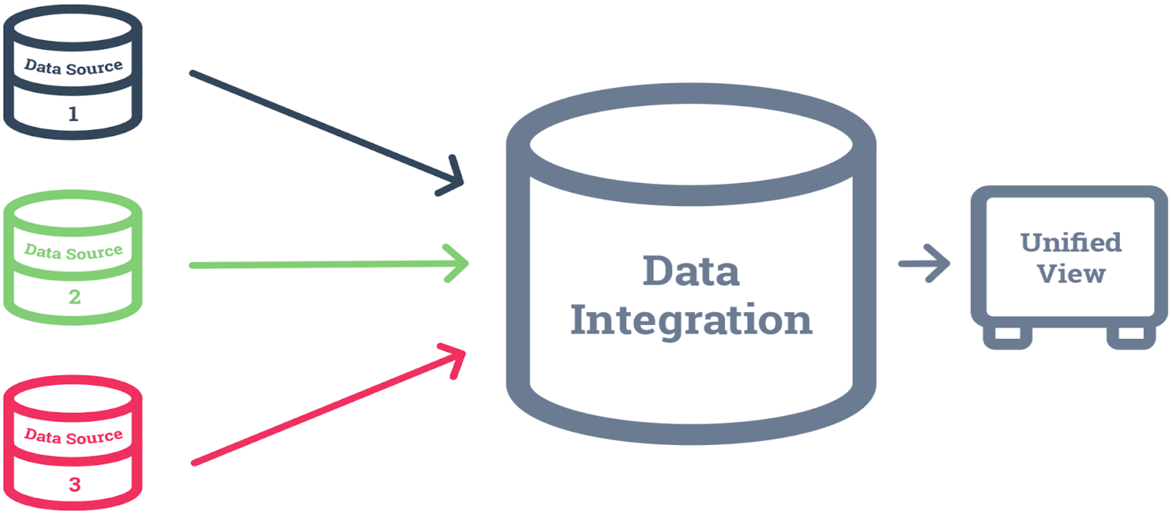


FIGURE 1.5.1 – Data Integration

1.6 Popular BI Tools

BI Tool	**Description**
Power BI	Développé par Microsoft, cet outil offre des tableaux de bord interactifs, des visualisations de données, et une intégration avec diverses sources de données. Il est connu pour sa facilité d'utilisation et ses fonctionnalités avancées d'analyse.
OBIEE	Une suite BI complète d'Oracle qui permet la création de rapports, de tableaux de bord et d'analyses ad hoc. Il est particulièrement adapté aux entreprises utilisant l'écosystème Oracle.
Tableau	Cet outil se distingue par ses puissantes fonctionnalités de visualisation de données. Il permet de créer des tableaux de bord interactifs et d'explorer les données de manière intuitive.

Chapitre 2

ETL (Extraction, Transformation, Load)

2.1 Data integration vs ETL VS ELT

Data integration is a broader concept that includes both ETL (Extract, Transform, Load) and ELT (Extract, Load, Transform) as subsets. It involves additional tasks like data quality management, defining reference data, and integrating data from different systems to create a unified view.

ETL focuses specifically on extracting data from various sources, transforming it into a format suitable for analysis, and then loading it into a target system (such as a data warehouse).

ELT, on the other hand, first extracts the data from various sources and loads it directly into the target system, after which the transformation takes place within the target system itself. ELT is more suited for environments where the target system has the computational power to handle transformations efficiently, particularly in big data or cloud-based settings.

2.1.1 ETL Process

Extract : Raw data is extracted from various sources.

Transform : A secondary processing server is used to transform the data.

Load : The transformed data is loaded into a target database.

The transformation step ensures that the data complies with the structural requirements of the target database. Data is only moved once it has been transformed and is ready for use.

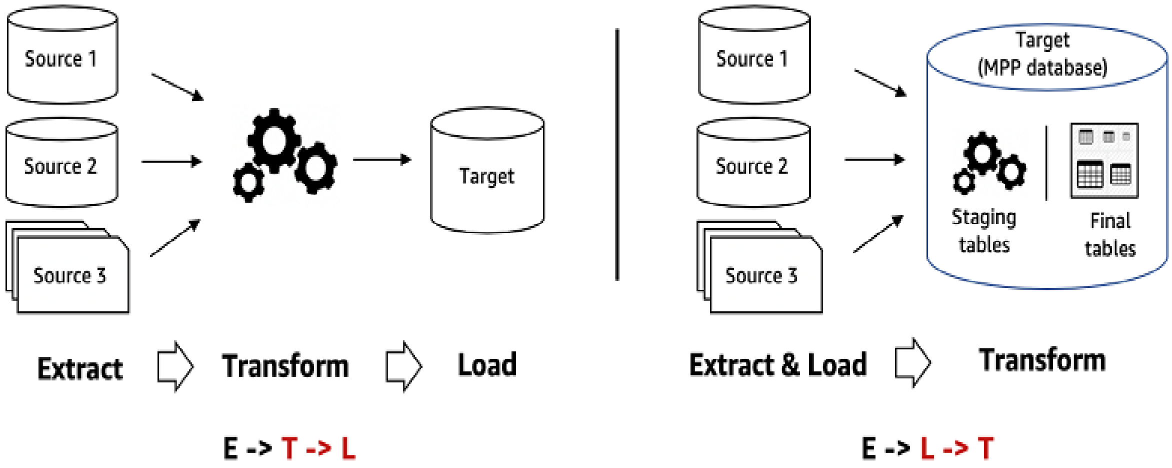


FIGURE 2.1.1 – ETL vs ELT

2.1.2 ELT Process

Extract : Raw data is extracted from various sources.

Load : The data is loaded into a data warehouse or data lake in its raw form.

Transform : The data is transformed as needed within the target system.

With ELT, all cleaning, transformation, and enrichment of the data take place within the data warehouse. You can interact with the raw data and transform it as many times as necessary.

2.2 ODI architecture components

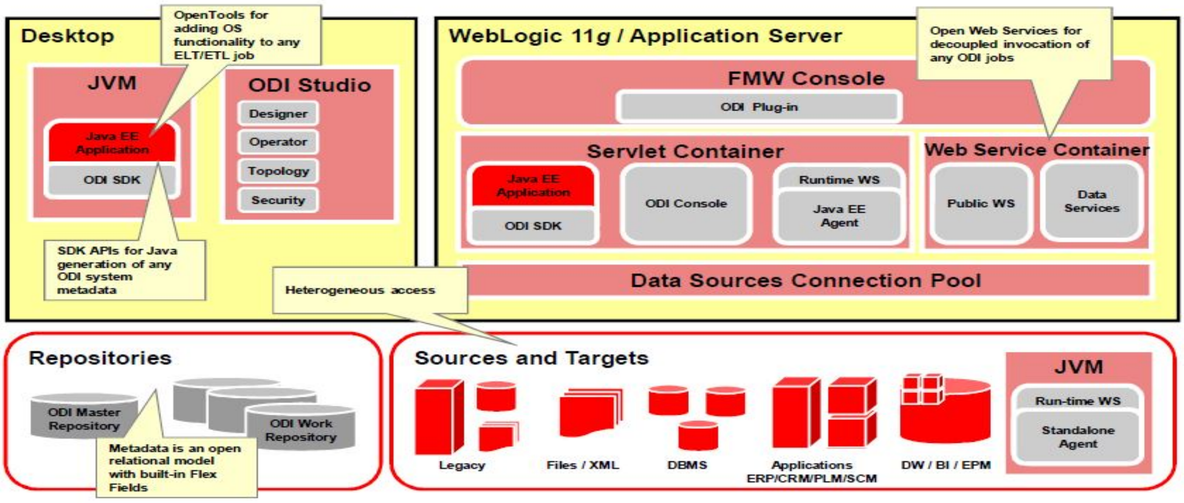


FIGURE 2.2.1 – ODI architecture components

2.2.1 Repositories

Repository is a centralized component which stores metadata information about all four navigators.This stores configuration information about the IT infrastructure ; the metadata for all applications, projects, scenarios,and execution logs. Repositories can be installed in an online transaction processing (OLTP) relational database. The repository also contains information about the ODI infrastructure, defined by the administrators. We have two types of repositories namely Master Repository and Work Repository.

Master Repository which stores all the information related to Topology and Security Navigators in ODI studio.It is a place to where set up all the connections and setting up appropriate privileges to the users based on their roles.

Work Repository which stores all the information related to Designer and Operator Navigators in ODI studio.It is a place where all the development work of project will takes place.

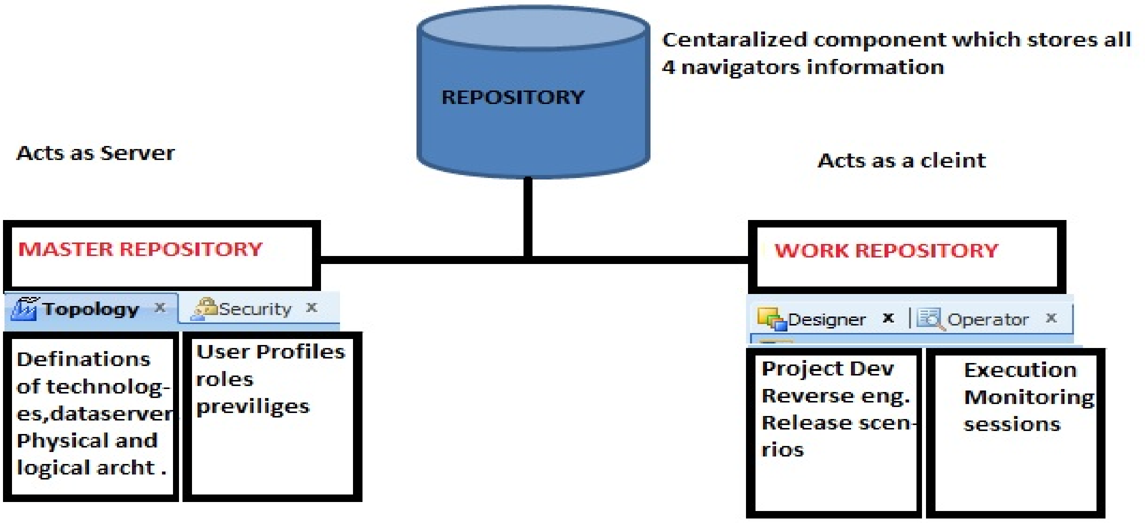

REPOSITORY
Centaralized component which stores all 4 navigators information
Acts as Server
Acts as a cleint
MASTER REPOSITORY
WORK REPOSITORY
Topology
Security
Designer
Operator
Definations of technolog-es,dataserver Physical and logical archt .
User Profiles roles previliges
Project Dev Reverse eng. Release scen-rios
Execution Monitoring sessions

FIGURE 2.2.2 – Repository

2.2.2 Agents

ODI can't execute anything by it self,it makes use of one component is called "Agent".An agent is a runtime component of ODI that orchestrates the integration process.It is a lightweight Java program that retrieves code from the repository at run time.

We have two types of agents namely
1) Stand alone Agent
2) Java EE Agent

2.2.3 ODI Studio

It is a user interface in ODI, if any user wants to interact with ODI, they have to make use of ODI studio.It is used to develop ETL mappings as well as to administer and monitor the ODI. ODI Studio provides four navigators for managing the different aspects and steps of an ODI integration project namely

1) Designer
2) Operator
3) Topology
4) security

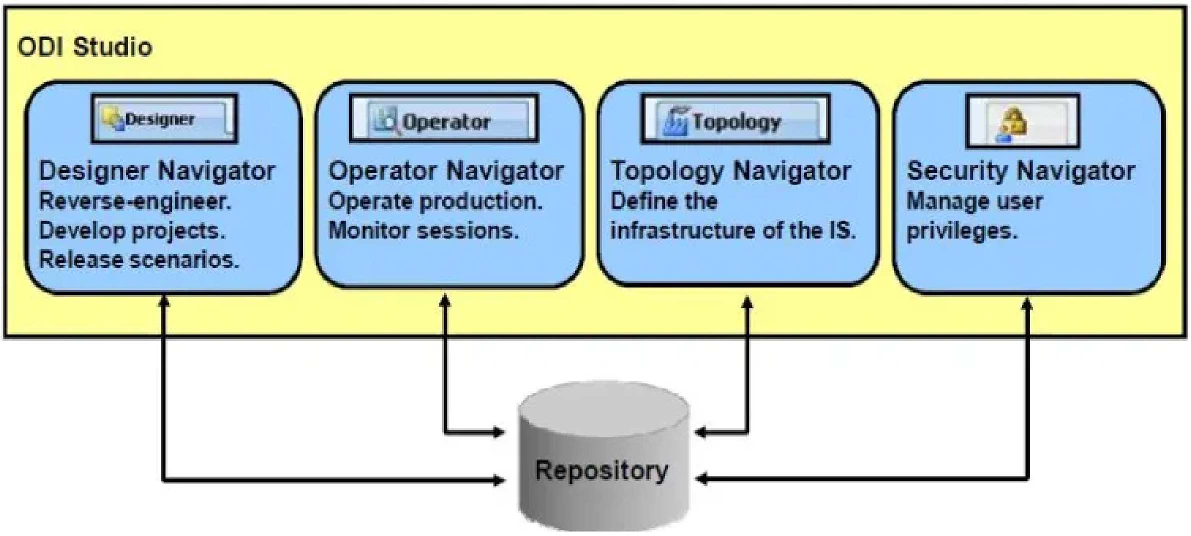


FIGURE 2.2.3 – ODI Studio

2.2.4 JEE components

The following JEE components are deployed in any application server

– Enterprise manager fusion middleware control SDK plugin for ODI : It is used to monitor ODI from a web browser

–Public web services : They are used to start and monitor scenarios through the JEE or a standalone agent

– ODI console : It is used to execute and monitor ODI jobs as well as browse ODI metadata from a web browser.In 10g it was called as “metadata navigator”.

– JEE agent : JEE enabled ODI agent that allows the ODI agent to inherently take advantage of the application server’s enterprise features, such as high availability, connection pooling, and so on.

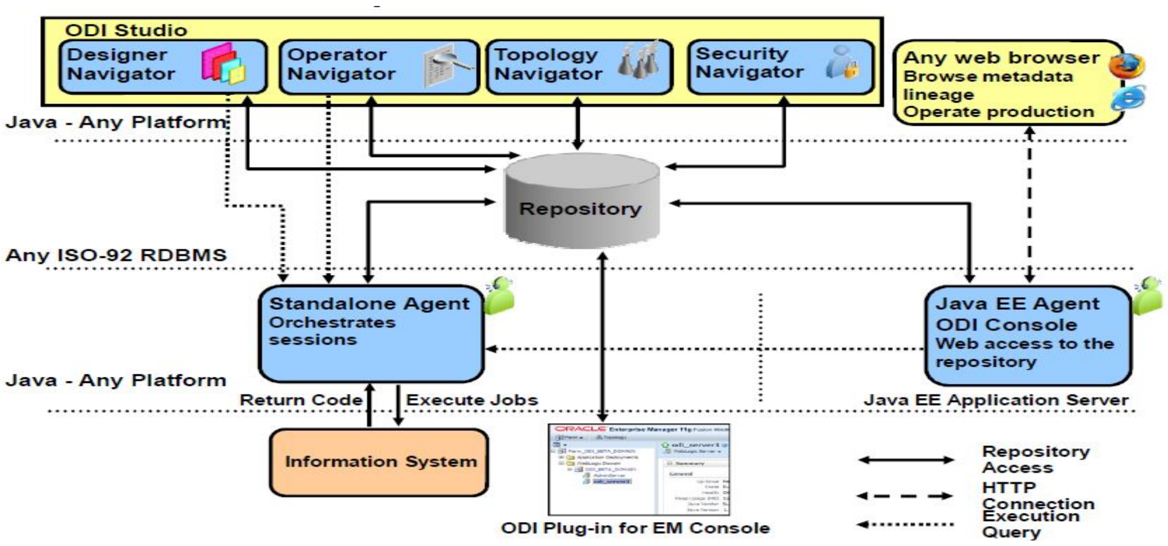


FIGURE 2.2.4 – JEE agent

2.3 Odi functionalities

2.3.1 Mapping

Mapping in ODI refers to the definition of data transformations between sources and targets. It allows mapping fields from a source (database, file, etc.) to those of a destination, while applying data transformations such as filters, type conversions, aggregations, etc.

2.3.2 Procedure

Procedures in ODI are scripts or sets of actions that can be executed during the data integration process. They are used to manage specific tasks such as executing SQL commands, calling external functions, or orchestrating complex operations.

2.3.3 package

A package in ODI is a unit of work composed of a set of tasks, such as mappings, procedures, and other operations related to a specific data flow. Packages can be used to organize and automate complex integration processes. They allow the execution of multiple mappings or processes in a sequential or conditional manner.

2.3.4 Knowledge Modules (KMs)

Knowledge Modules (KMs) are preconfigured components in ODI that define how to access and manipulate data optimally. There are several types of KMs : LKM (Loading Knowledge Module) : Manages loading data from a source into a staging area. IKM

(Integration Knowledge Module) : Used to perform transformations on data during integration. RKM (Reverse Knowledge Module) : Allows reverse-engineering of metadata from the source. CKM (Check Knowledge Module) : Performs data validation (e.g., checks data quality before loading).

Chapitre 3

ODI Studio

ODI 12c interface is straightforward and easy to use. It has 4 main module (tabs) : Designer, Operator, Topology and Security Navigators, all connected to a Repository, as shown below :

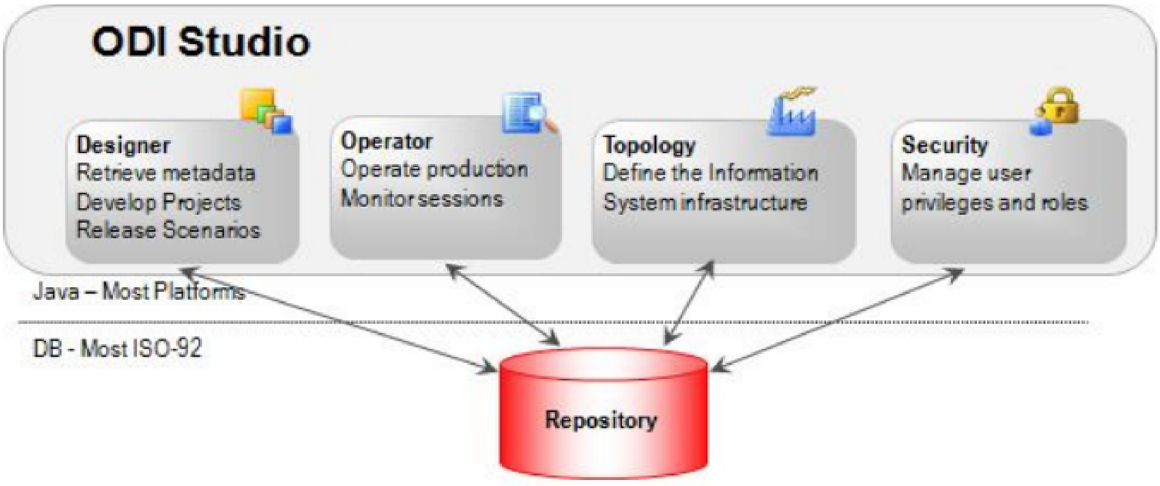


FIGURE 3.0.1 – 4 modules ODI Studio

3.1 Operator module

Operator handles the execution and monitoring part, by helping developers perform debugs and code checks. Its components are : Session List (that keeps all the sessions split by data, agents etc), Hierarchical Sessions, Load Plans Executions, Scheduler, Load Plan and Scenarios and Solutions.

3.2 Security module

Security encapsulates the user's rights, roles and rights on objects.

3.3 Designer module

Designer is used to create and modify the metadata, ensure data integrity and transformation. The main components used here are the data models (that keep all the details about the data, for example the tables, columns) and the projects (that refer to the rules for loading and transforming the data, such as mappings, procedures, variables etc.). The Designer tab components are : Projects, Models, Load Plans and Scenarios, Global Objects and Solutions.

3.4 Topology module

Topology is used to define the logical and physical architecture of the infrastructure.

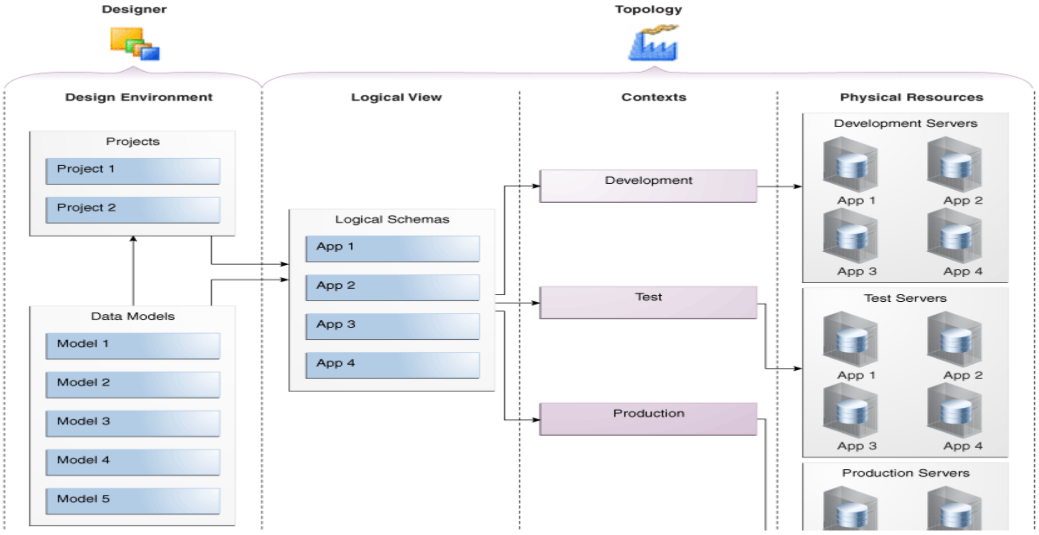


FIGURE 3.4.1 – Topology module

The logical view describes logical schemas that represent the physical schemas of the existing applications independently of their physical implementation. These logical schemas are then linked to the physical resources through contexts.

Designers always refer to the logical view defined in the Topology. All development done therefore becomes independent of the physical location of the resources they address. At runtime, the logical information is mapped to the physical resources, given the appropriate contexts. The same scenario can be executed on different physical servers and applications simply by specifying different contexts. This brings a very flexible architecture where developers don't have to worry about the underlying physical implementation of the servers they rely on.

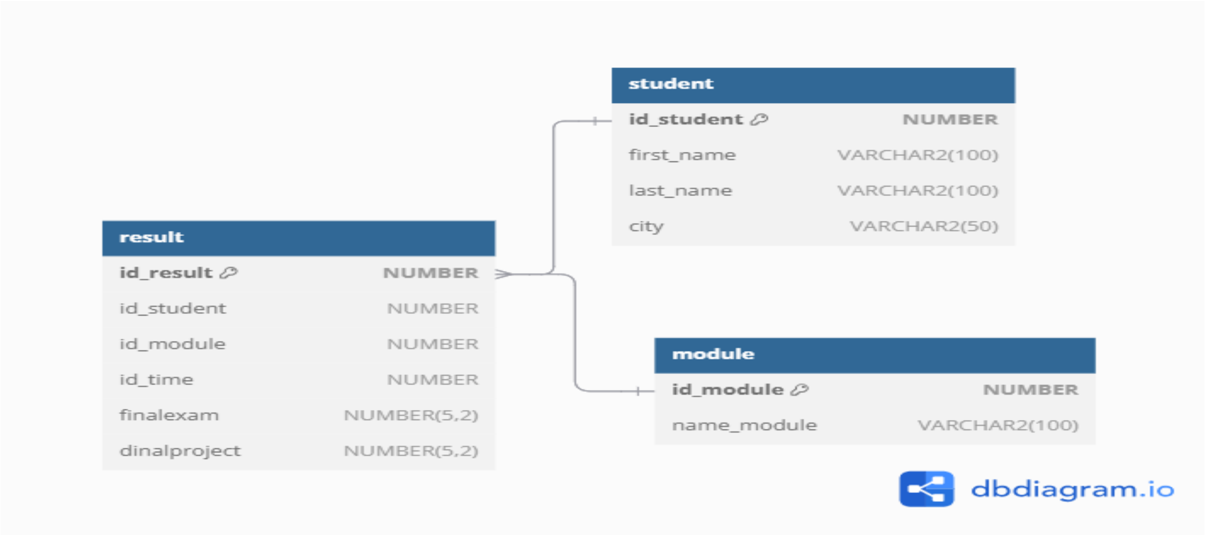
Chapitre 4

Demonstration with Oracle Data Integrator

4.1 Data base and Data warehouse Structure

4.1.1 oracle Data Base source

For my oracle src-University database i have 3 basic tables (Result, Student, Module,), and each table has some random data about Students and Modules and their Results. Here in the image below you can find the structure of my Data Base.



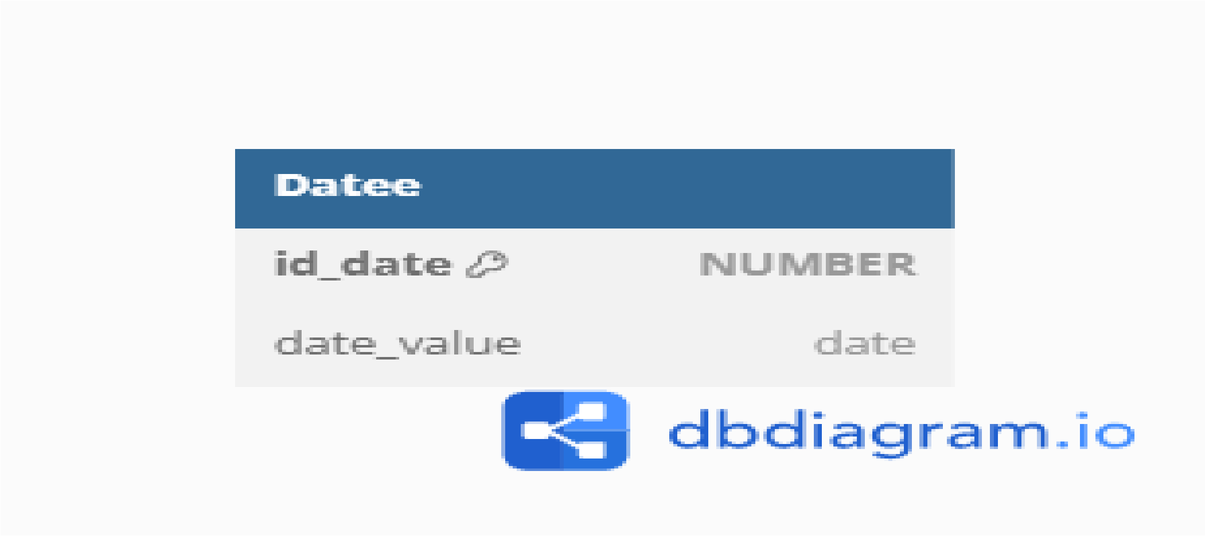
student	
id_student	NUMBER
first_name	VARCHAR2(100)
last_name	VARCHAR2(100)
city	VARCHAR2(50)

result	
id_result	NUMBER
id_student	NUMBER
id_module	NUMBER
id_time	NUMBER
finalexam	NUMBER(5,2)
dinalproject	NUMBER(5,2)

module	
id_module	NUMBER
name_module	VARCHAR2(100)

FIGURE 4.1.1 – The structure of my data base.

4.1.2 flat file source

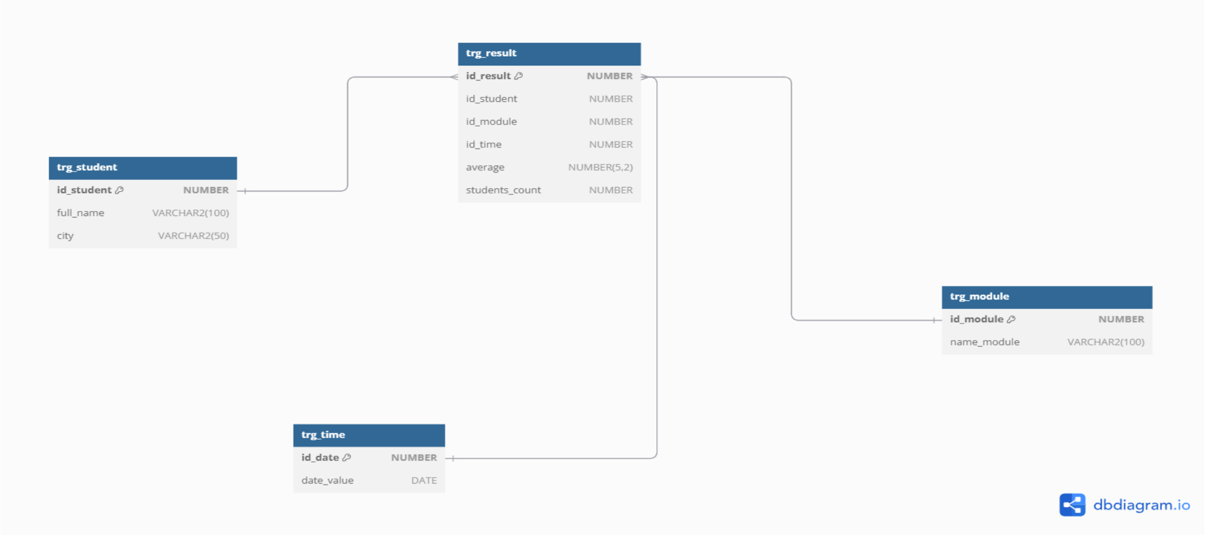


Datee	
id_date	NUMBER
date_value	date

FIGURE 4.1.2 – The structure of flat file.

4.1.3 Data warehouse

For my data warehouse structure i have as a Fact Table "Result" and for now 3 dimensions (Time ,Module ,Student) and finally 2 Measures (Average, Students-count) Here is a proposed schema of my data warehouse.



trg_result	
id_result	NUMBER
id_student	NUMBER
id_module	NUMBER
id_time	NUMBER
average	NUMBER(5,2)
students_count	NUMBER

trg_student	
id_student	NUMBER
full_name	VARCHAR2(100)
city	VARCHAR2(50)

trg_module	
id_module	NUMBER
name_module	VARCHAR2(100)

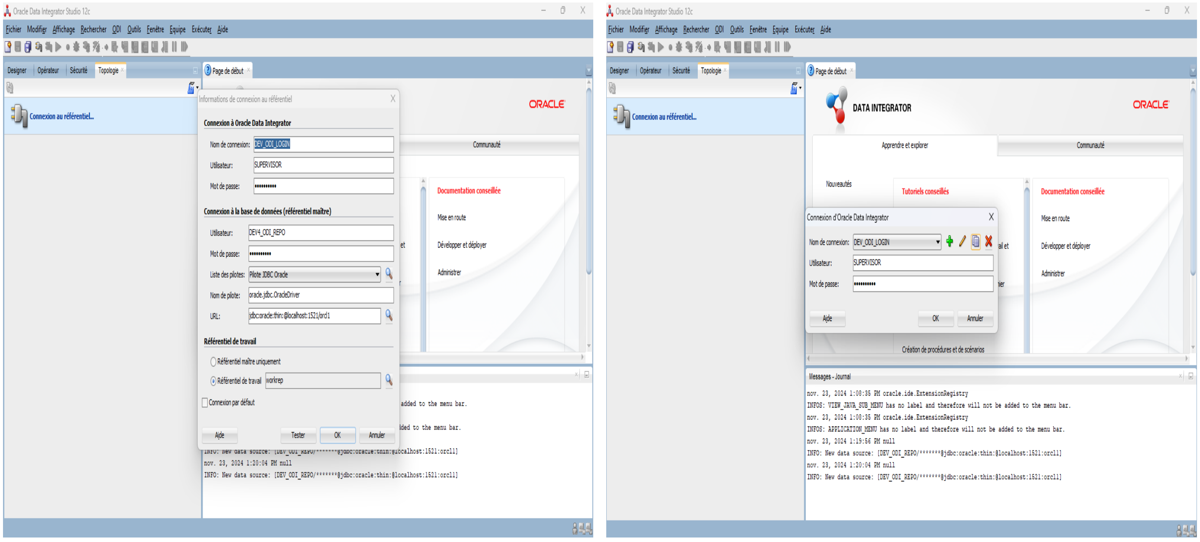
trg_time	
id_date	NUMBER
date_value	DATE

FIGURE 4.1.3 – The data warehouse schema.

4.2 Intégration avec ODI

4.2.1 Connecting to ODI

To establish a connection in Oracle Data Integrator (ODI), I configured the necessary parameters in three main sections. In the "Connection to Oracle Data Integrator" section, I provided the connection name, user name, and password to authenticate with ODI. In the "Connection to the Database" section, I entered the database credentials, including the user name, password, driver name, and the database connection URL (containing the host, port, and database service name). Lastly, in the "Work Repository" section, I specified the unique master repository reference and the work repository reference, which is named workrep, to link the configuration to the correct repositories. Once all the details were filled in, I saved the configuration and successfully tested the connection.



Informations de connexion au référentiel

Connexion à Oracle Data Integrator

Nom de connexion: DEV_ODI_LOGIN
Utilisateur: SUPERVISOR
Mot de passe: ••••••••••

Connexion à la base de données (référentiel maître)

Utilisateur: DEV4_ODI_REPO
Mot de passe: ••••••••••
Liste des pilotes: Pilote JDBC Oracle
Nom de pilote: oracle.jdbc.OracleDriver
URL: jdbc:oracle:thin:@localhost:1521/orcl1

Référentiel de travail

○ Référentiel maître uniquement
● Référentiel de travail: workrep

□ Connexion par défaut

Connexion d'Oracle Data Integrator

Nom de connexion: DEV_ODI_LOGIN
Utilisateur: SUPERVISOR
Mot de passe: ••••••••••

Messages - Journal

```
nov. 23, 2024 1:08:35 PM oracle.ide.ExtensionRegistry
INFOS: VIEW_JAVA_SUB_MENU has no label and therefore will not be added to the menu bar.
nov. 23, 2024 1:08:35 PM oracle.ide.ExtensionRegistry
INFOS: APPLICATION_MENU has no label and therefore will not be added to the menu bar.
nov. 23, 2024 1:19:56 PM null
INFO: New data source: [DEV_ODI_REPO/*******@jdbc:oracle:thin:@localhost:1521:orcl1]
nov. 23, 2024 1:20:04 PM null
INFO: New data source: [DEV_ODI_REPO/*******@jdbc:oracle:thin:@localhost:1521:orcl1]
```

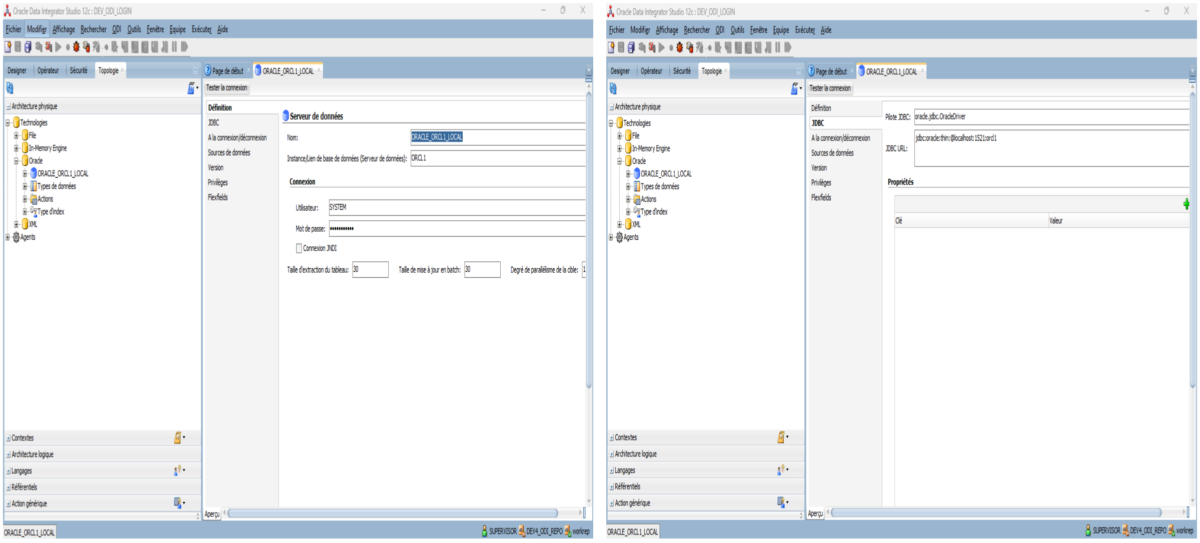
FIGURE 4.2.1 – Connecting to ODI

4.2.2 Topologie odi

Creation of a Data Server and Physical Schema

Data Server

In the ODI topology model, I created a Data Server. I entered the server name and the database link (orcl1). Then, I configured the connection by entering the username, password, and JDBC parameters. For this, I specified the JDBC driver (oracle.jdbc.OracleDriver) and the JDBC URL.



Serveur de données

Nom: ORACLE_ORCL1_LOCAL

Instance/Lien de base de données (Serveur de données): ORCL1

Connexion

Utilisateur: SYSTEM

Mot de passe: ••••••••••••

□ Connexion JNDI

Taille d'extraction du tableau: 30 Taille de mise à jour en batch: 30 Degré de parallélisme de la cible: 1

Pilote JDBC: oracle.jdbc.OracleDriver

JDBC URL: jdbc:oracle:thin:@localhost:1521:orcl1

Propriétés

Clé	Valeur

FIGURE 4.2.2 – Data Server

Physical Schema To create the physical schema in Oracle Data Integrator (ODI), I navigated to the Physical Architecture tab and initiated the creation process. I specified the source schema, naming it src-university1, and assigned the work schema as odi-tmpee1. After defining these schemas, I configured the necessary connections to ensure proper access to the source schema. Finally, I validated the setup by testing the connection, ensuring that the physical schema was correctly configured and ready for use

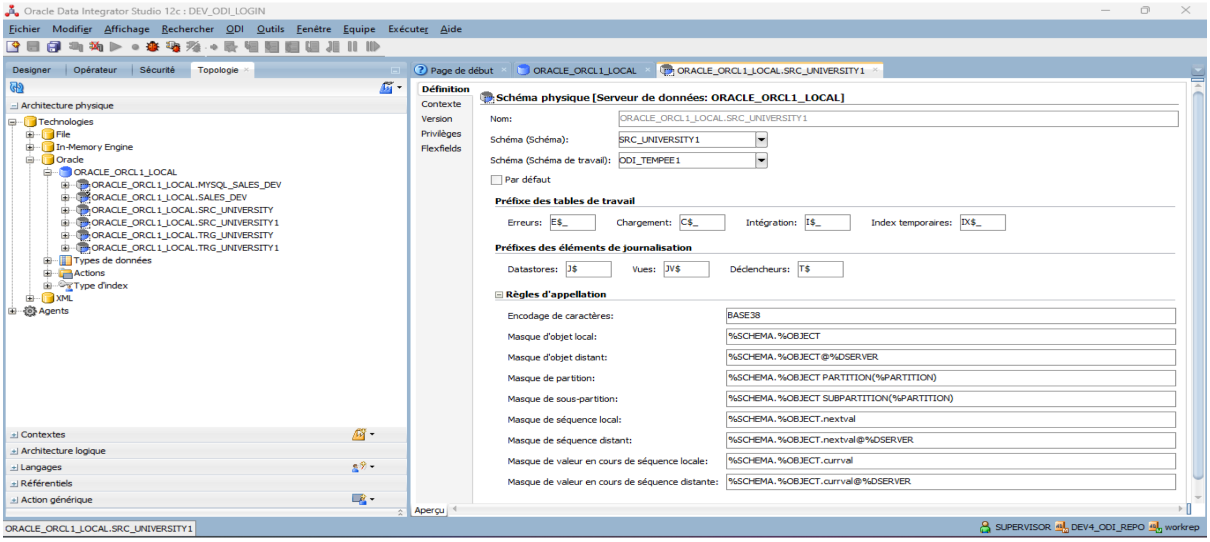


Schéma physique [Serveur de données: ORACLE_ORCL1_LOCAL]

Nom: ORACLE_ORCL1_LOCAL.SRC_UNIVERSITY1

Schéma (Schéma): SRC_UNIVERSITY1

Schéma (Schéma de travail): ODI_TEMPEE1

□ Par défaut

Préfixe des tables de travail

Erreurs: E$_ Chargement: C$_ Intégration: I$_ Index temporaires: IX$_

Préfixes des éléments de journalisation

Datastores: J$ Vues: JV$ Déclencheurs: T$

Règles d'appellation

Encodage de caractères: BASE38

Masque d'objet local: %SCHEMA.%OBJECT

Masque d'objet distant: %SCHEMA.%OBJECT@%DSERVER

Masque de partition: %SCHEMA.%OBJECT PARTITION(%PARTITION)

Masque de sous-partition: %SCHEMA.%OBJECT SUBPARTITION(%PARTITION)

Masque de séquence local: %SCHEMA.%OBJECT.nextval

Masque de séquence distant: %SCHEMA.%OBJECT.nextval@%DSERVER

Masque de valeur en cours de séquence locale: %SCHEMA.%OBJECT.currval

Masque de valeur en cours de séquence distante: %SCHEMA.%OBJECT.currval@%DSERVER

Oracle Data Integrator Studio 12c : DEV_ODI_LOGIN
Fichier Modifier Affichage Rechercher ODI Outils Fenêtre Equipe Exécuter Aide
Designer Opérateur Sécurité Topologie
Page de début
ORACLE_ORCL1_LOCAL
ORACLE_ORCL1_LOCAL.SRC_UNIVERSITY1
Définition
Contexte
Version
Privilèges
Flexfields
Architecture physique
Technologies
File
In-Memory Engine
Oracle
ORACLE_ORCL1_LOCAL
ORACLE_ORCL1_LOCAL.MYSQL_SALES_DEV
ORACLE_ORCL1_LOCAL.SALES_DEV
ORACLE_ORCL1_LOCAL.SRC_UNIVERSITY
ORACLE_ORCL1_LOCAL.SRC_UNIVERSITY1
ORACLE_ORCL1_LOCAL.TRG_UNIVERSITY
ORACLE_ORCL1_LOCAL.TRG_UNIVERSITY1
Types de données
Actions
Type d'index
XML
Agents
Contextes
Architecture logique
Langages
Référentiels
Action générique
Aperçu
ORACLE_ORCL1_LOCAL.SRC_UNIVERSITY1
SUPERVISOR DEV4_ODI_REPO workrep

FIGURE 4.2.3 – Physical Schema

logical Schema In the logical architecture, I created a logical schema named src-university1. This logical schema serves as an abstraction layer, mapping to the physical schema defined earlier. By associating it with the corresponding physical schema, I ensured that the logical schema could seamlessly interact with the source data. This setup simplifies the design process by allowing ODI to reference the data source through the logical schema, making it easier to manage and reuse in different contexts.

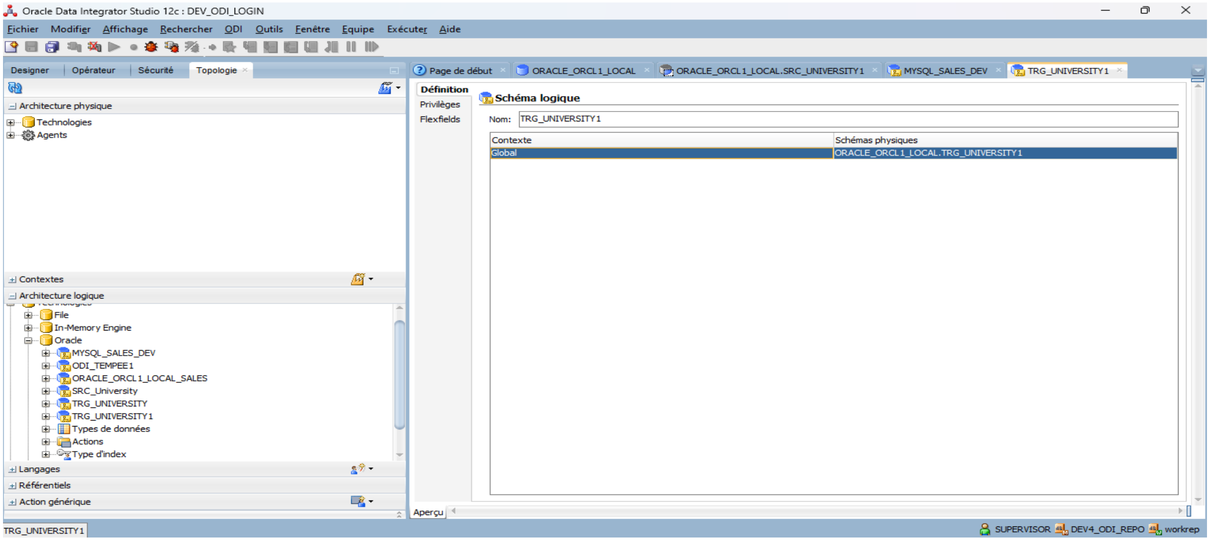


Schéma logique

Nom: TRG_UNIVERSITY1

Contexte	Schémas physiques
Global	ORACLE_ORCL1_LOCAL.TRG_UNIVERSITY1

FIGURE 4.2.4 – logical Schema

contexe To establish the connection between the physical schema and the logical schema, I configured the context in Oracle Data Integrator (ODI). This step involved associating the logical schema src-university1 with the corresponding physical schema for a specific execution context. By doing so, I ensured that ODI could correctly resolve and use the physical schema during data integration processes, depending on the selected context. This configuration provides flexibility and enables seamless environment transitions, such as moving between development, testing, and production stages.

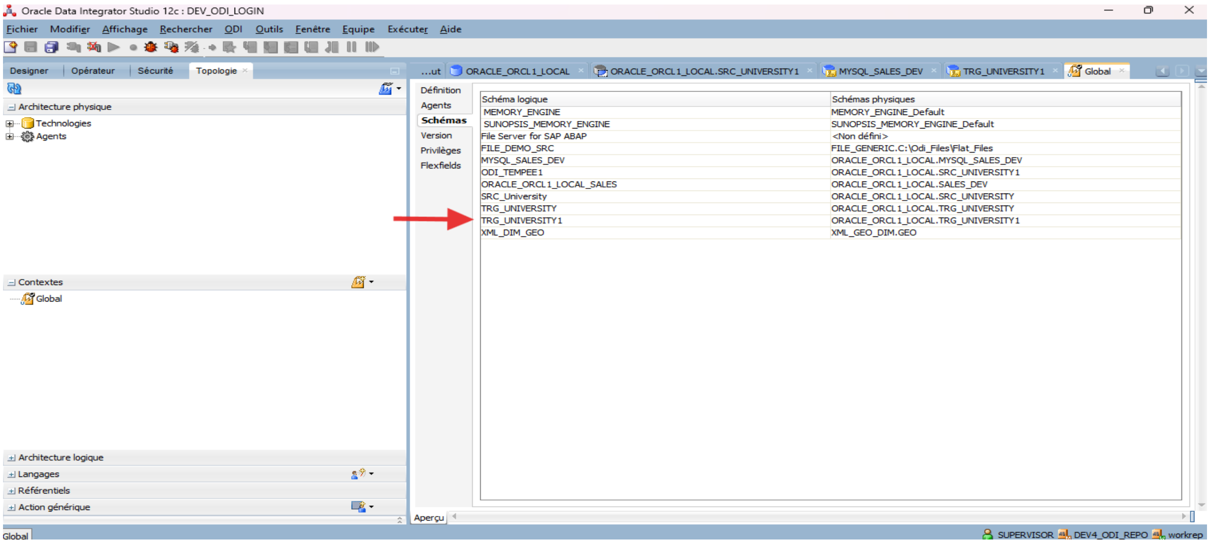
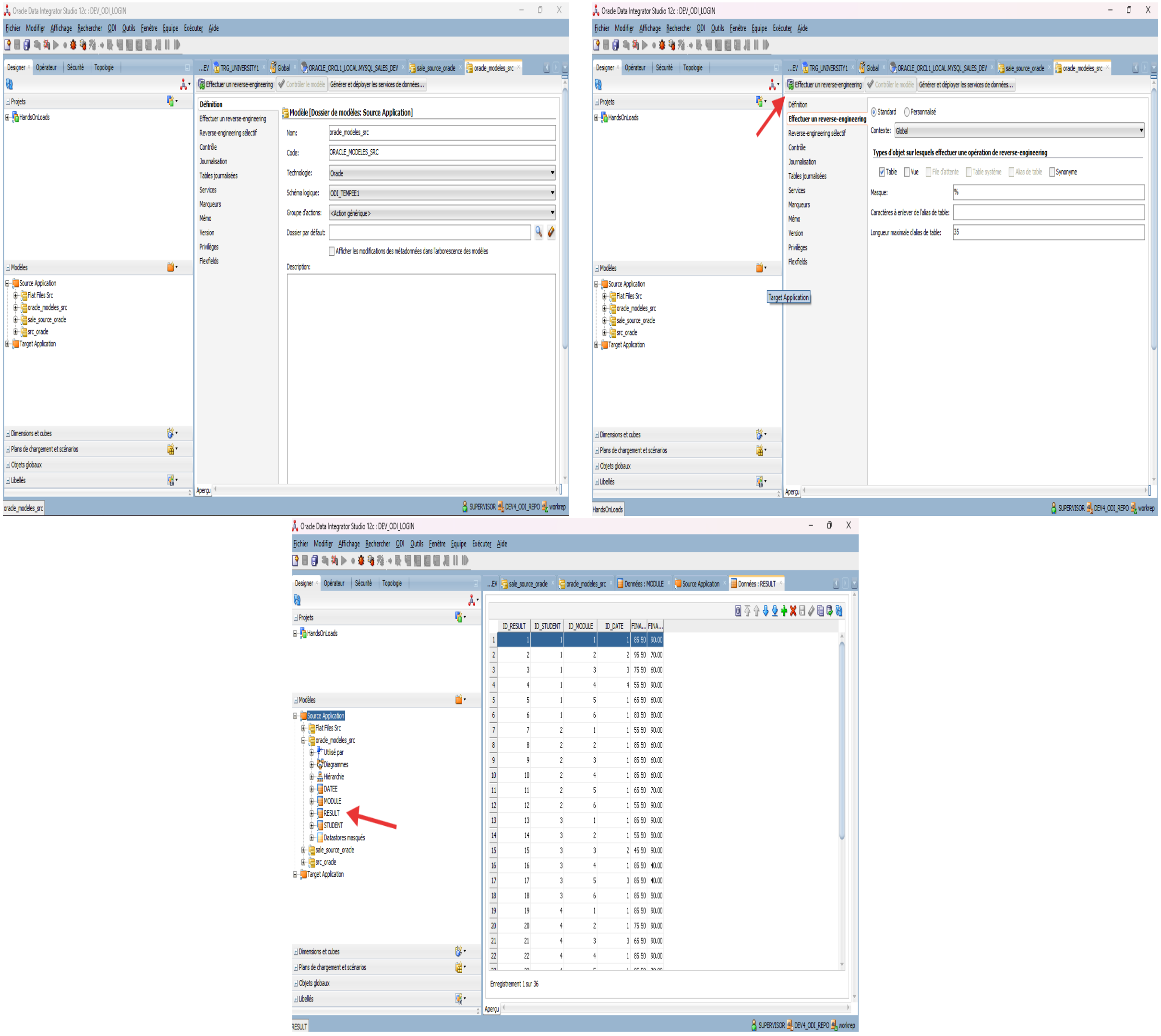


Schéma logique	Schémas physiques
MEMORY_ENGINE	MEMORY_ENGINE_Default
SUNOPSIS_MEMORY_ENGINE	SUNOPSIS_MEMORY_ENGINE_Default
File Server for SAP ABAP	<Non défini>
FILE_DEMO_SRC	FILE_GENERIC.C:\Odi_Files\Flat_Files
MYSQL_SALES_DEV	ORACLE_ORCL1_LOCAL.MYSQL_SALES_DEV
ODI_TEMPEE1	ORACLE_ORCL1_LOCAL.SRC_UNIVERSITY1
ORACLE_ORCL1_LOCAL_SALES	ORACLE_ORCL1_LOCAL.SALES_DEV
SRC_University	ORACLE_ORCL1_LOCAL.SRC_UNIVERSITY
TRG_UNIVERSITY	ORACLE_ORCL1_LOCAL.TRG_UNIVERSITY
TRG_UNIVERSITY1	ORACLE_ORCL1_LOCAL.TRG_UNIVERSITY1
XML_DIM_GEO	XML_GEO_DIM.GEO

FIGURE 4.2.5 – contexe

4.2.3 ODI Designer

model In ODI Designer, I created a model by specifying the model name, code, technology, and associating it with the previously defined logical schema src-university1. After completing these settings, I initiated the reverse engineering process by clicking on Execute a reverse engineering. This action triggered ODI to extract all relevant metadata and data from the source system, allowing the model to reflect the structure and content of the source data. The reverse engineering process automatically populated the model with the necessary objects, such as tables and columns, which are now available for further data integration tasks.


Oracle Data Integrator Studio 12c : DEV_ODI_LOGIN
Fichier Modifier Affichage Rechercher ODI Outils Fenêtre Equipe Exécuter Aide
Designer Opérateur Sécurité Topologie
...EV TRG_UNIVERSITY1 Global ORACLE_ORCL1_LOCAL.MYSQL_SALES_DEV sale_source_oracle oracle_modeles_src
Effectuer un reverse-engineering
Contrôler le modèle
Générer et déployer les services de données...
Projets
HandsOnLoads
Modèles
Source Application
Flat Files Src
oracle_modeles_src
sale_source_oracle
src_oracle
Target Application
Utilisé par
Diagrammes
Hiérarchie
DATEE
MODULE
RESULT
STUDENT
Datastores masqués
Dimensions et cubes
Plans de chargement et scénarios
Objets globaux
Libellés
Définition
Effectuer un reverse-engineering
Reverse-engineering sélectif
Contrôle
Journalisation
Tables journalisées
Services
Marqueurs
Mémo
Version
Privilèges
Flexfields
Aperçu
SUPERVISOR DEV4_ODI_REPO workrep
Données : MODULE
Données : RESULT

Modèle [Dossier de modèles: Source Application]

Nom: oracle_modeles_src

Code: ORACLE_MODELES_SRC

Technologie: Oracle

Schéma logique: ODI_TEMPEE1

Groupe d'actions: <Action générique>

Dossier par défaut:

□ Afficher les modifications des métadonnées dans l'arborescence des modèles

Description:

● Standard ○ Personnalisé

Contexte: Global

Types d'objet sur lesquels effectuer une opération de reverse-engineering

☑ Table □ Vue □ File d'attente □ Table système □ Alias de table □ Synonyme

Masque: %

Caractères à enlever de l'alias de table:

Longueur maximale d'alias de table: 35

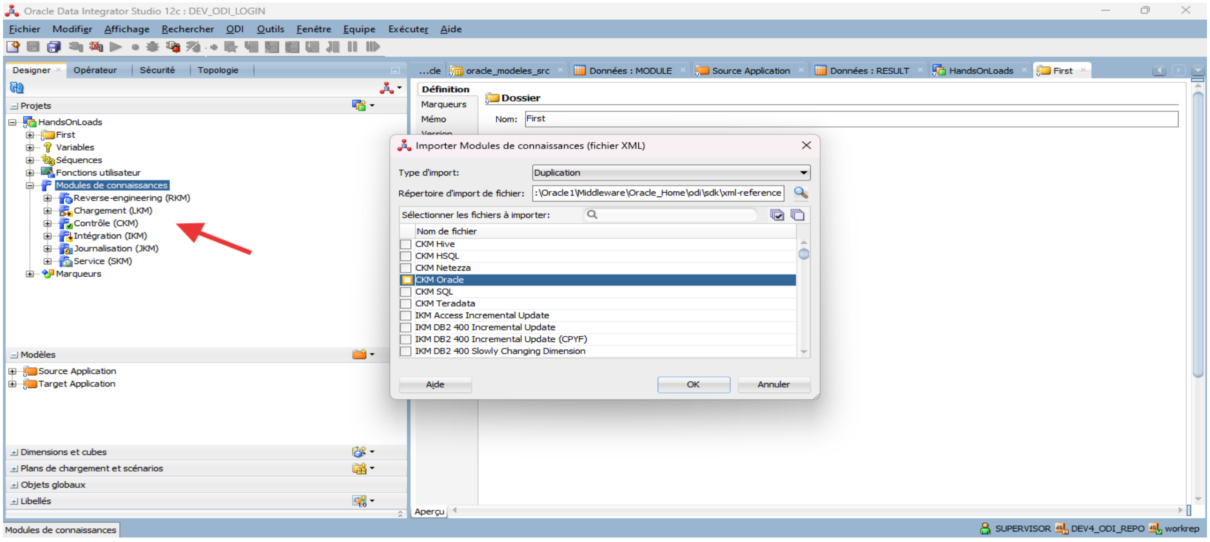
	ID_RESULT	ID_STUDENT	ID_MODULE	ID_DATE	FINA...	FINA...
1	1	1	1	1	85.50	90.00
2	2	1	2	2	95.50	70.00
3	3	1	3	3	75.50	60.00
4	4	1	4	4	55.50	90.00
5	5	1	5	1	65.50	60.00
6	6	1	6	1	83.50	80.00
7	7	2	1	1	55.50	90.00
8	8	2	2	1	85.50	60.00
9	9	2	3	1	85.50	60.00
10	10	2	4	1	85.50	60.00
11	11	2	5	1	65.50	70.00
12	12	2	6	1	55.50	90.00
13	13	3	1	1	85.50	90.00
14	14	3	2	1	55.50	50.00
15	15	3	3	2	45.50	90.00
16	16	3	4	1	85.50	40.00
17	17	3	5	3	85.50	40.00
18	18	3	6	1	85.50	50.00
19	19	4	1	1	85.50	90.00
20	20	4	2	1	75.50	90.00
21	21	4	3	3	65.50	90.00
22	22	4	4	1	85.50	90.00

Enregistrement 1 sur 36

FIGURE 4.2.6 – model

Knowledge Module

Here, I imported the appropriate Knowledge Module (KM) into the model. The Knowledge Module (KM) is a set of pre-built scripts and processes that define how data is extracted, transformed, and loaded (ETL). By importing the KM, I ensured that the model would have the necessary logic to handle the data operations in alignment with the technology and requirements of the source and target systems. This step streamlined the ETL process and enabled the efficient execution of data integration tasks by leveraging the predefined functionality within the Knowledge Module



Nom: First

Importer Modules de connaissances (fichier XML)

Type d'import: Duplication

Répertoire d'import de fichier: :\Oracle1\Middleware\Oracle_Home\odi\sdk\xml-reference

Sélectionner les fichiers à importer:

	Nom de fichier
□	CKM Hive
□	CKM HSQL
□	CKM Netezza
□	CKM Oracle
□	CKM SQL
□	CKM Teradata
□	IKM Access Incremental Update
□	IKM DB2 400 Incremental Update
□	IKM DB2 400 Incremental Update (CPYF)
□	IKM DB2 400 Slowly Changing Dimension

FIGURE 4.2.7 – Knowledge Module

mapping with Distinct component Here, I performed a mapping of the student table using the Distinct component to remove duplicates. This step ensures that only unique records are processed and transferred during the ETL process. By incorporating the Distinct component in the mapping, I filtered out any redundant data, ensuring data integrity and consistency in the target system. This approach helps optimize the quality of the data by eliminating unnecessary duplicates before further processing or loading into the target database

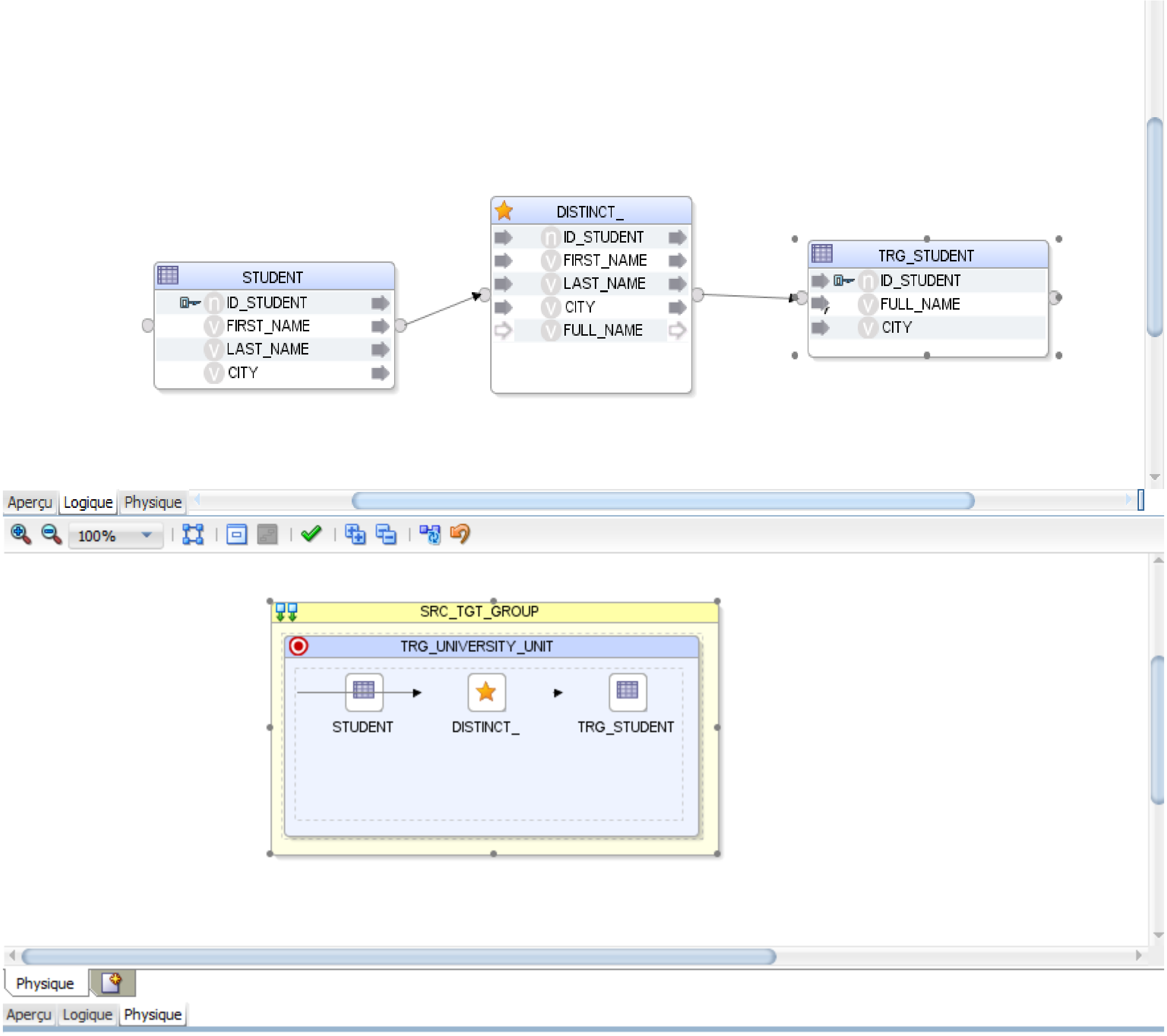
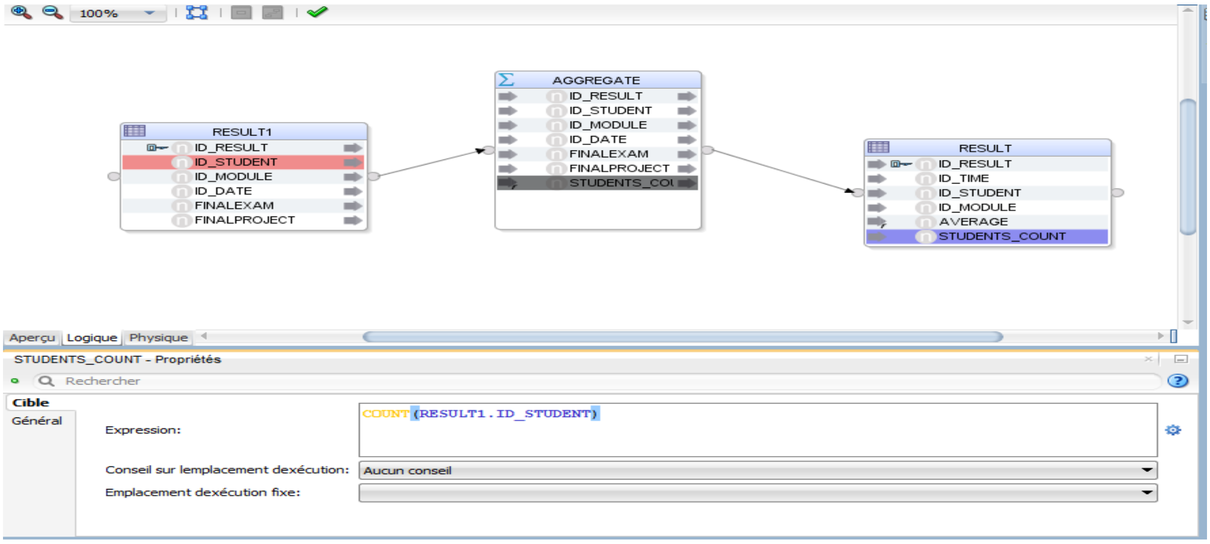


FIGURE 4.2.8 – mapping with Distinct component

mapping with Distinct Aggregate Here, I mapped the result table using the Aggregate component to perform aggregations on specific attributes. I applied a COUNT function to the student-account attribute to determine the number of records for each account. Additionally, I calculated the average for the finalexam and finalproject attributes, providing valuable insights into students' performance. The use of the Aggregate component allowed me to efficiently summarize the data and generate the required metrics, which will be useful for reporting or further analysis in the target system



STUDENTS_COUNT - Propriétés

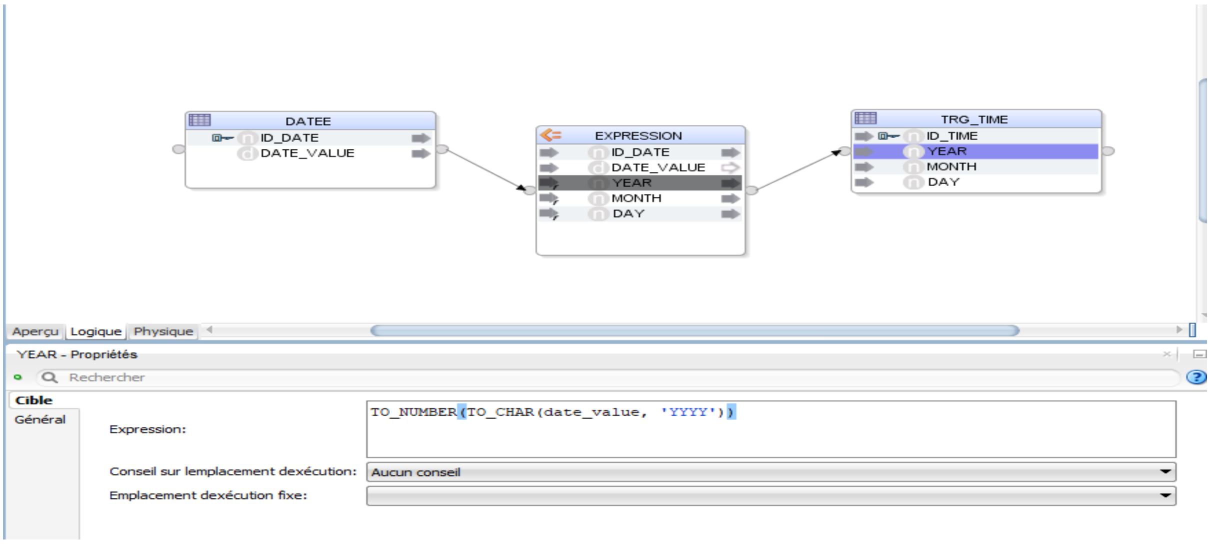
Expression: COUNT(RESULT1.ID_STUDENT)

Conseil sur lemplacement dexécution: Aucun conseil

Emplacement dexécution fixe:

FIGURE 4.2.9 – mapping with Distinct Aggregate

mapping with Distinct Expression Here, I mapped the date table using the Expression component to transform the date into separate day, month, and year components. This transformation allowed me to break down the date into its individual elements, which can be more useful for reporting, analysis, or filtering purposes. By applying this logic within the Expression component, I ensured that the data would be formatted correctly, enabling easier manipulation and comparison of dates in downstream processes or target systems.



YEAR - Propriétés

Expression: TO_NUMBER(TO_CHAR(date_value, 'YYYY'))

Conseil sur lemplacement dexécution: Aucun conseil

Emplacement dexécution fixe:

FIGURE 4.2.10 – mapping with Distinct Expression

4.3 visualization with Power BI

Here, I created a visualization using Power BI to display the average student results by module and the overall average student results. To achieve this, I imported the processed data from ODI, which includes the calculated averages for the final exam and final project. Using Power BI's visualization tools, I created charts and tables that clearly present the average performance for each module and the overall student results. This visualization provides a comprehensive overview of student performance, helping to identify trends and insights across different modules and overall academic achievement

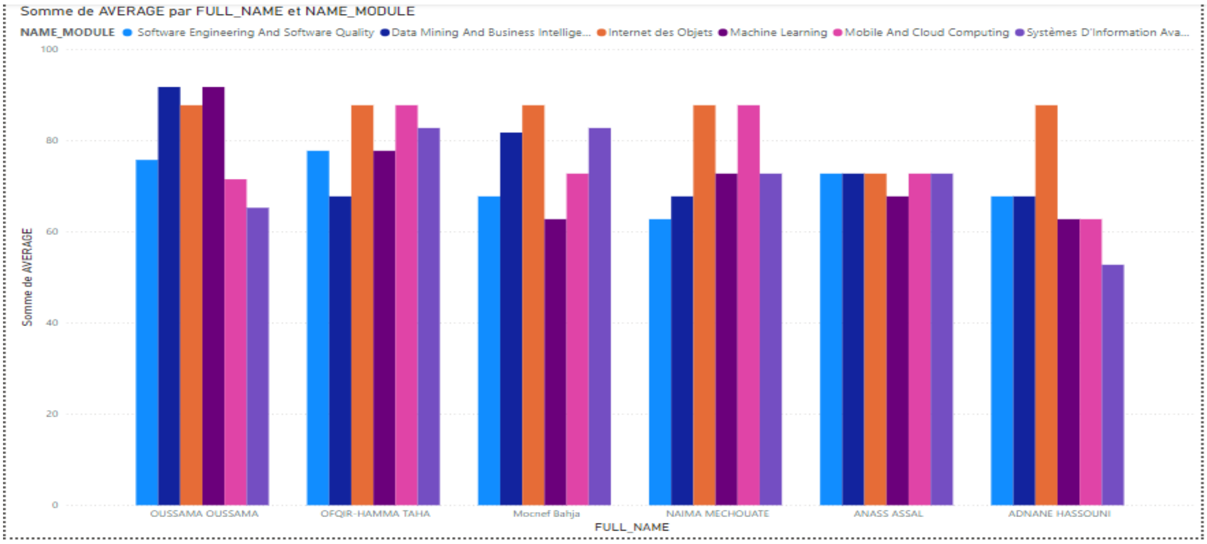


FIGURE 4.3.1 – visualization with Power BI

viii

Here, I created a visualization using Power BI to display the average student results. I imported the relevant data, which includes the calculated averages for the final exam and final project. Using Power BI's visualization tools, I created a chart or table that presents the overall average student performance. This visualization helps to easily track and analyze student outcomes, offering a clear view of their academic achievements and enabling data-driven decisions for improving performance.

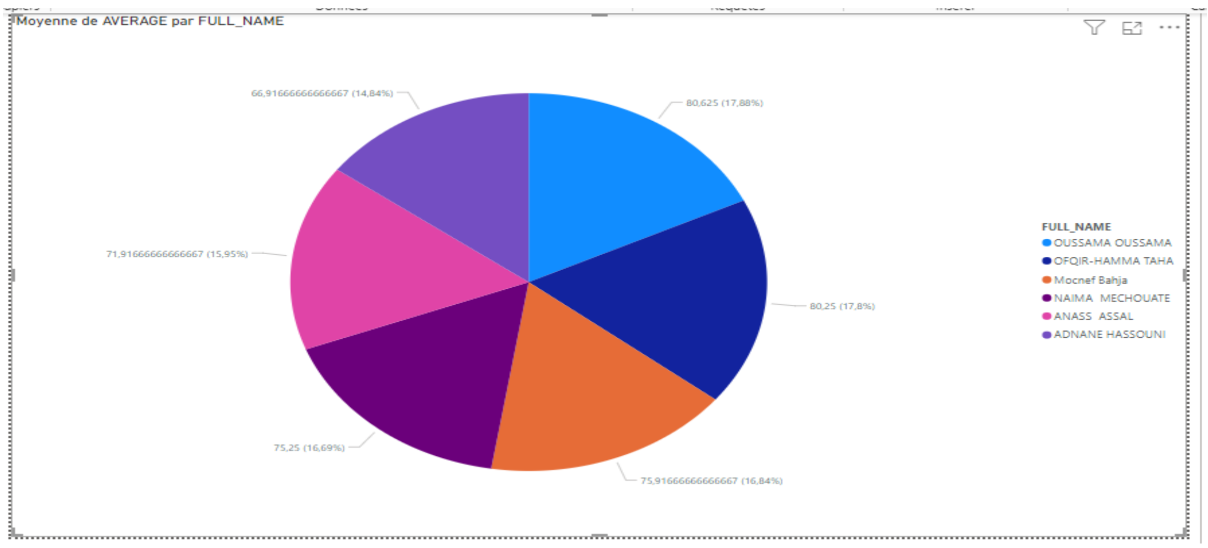


FIGURE 4.3.2 – visualization with Power BI

Conclusion

Conclusion In an organization, managers have a key role in the success of operations. The managers have to take decisions that directly affect the organization. A bad decision by a poorly informed source will lead to disastrous outcomes. The decisions are taken at individual level right to the organizational level The implementation of BI will improve the decisions taken by the mangers. The BI systems will in fact empower all the employees with enough information and make them capable of decision making.

In an organization a lot of data is present. Data can be related to customers, suppliers, invoices, purchase orders, pay slips, employee information, sales data, financial data, training

data, product data, client data, etc. It becomes cumbersome to manage such huge amounts of ever increasing data. BI systems can make the management of these huge amounts of data easy by using technologies like ODI and Data Warehousing

Références

1. TechTarget, *Business Intelligence (BI) : Definition and Examples*. URL : https://www.techtarget.com/searchbusinessanalytics/definition/business-intelligence-BI
2. AWS, *The Difference Between ETL and ELT*. URL : https://aws.amazon.com/fr/compare/the-difference-between-etl-and-elt/
3. Talend, *What is a Data Mart ?*. URL : https://www.talend.com/fr/resources/what-is-data-mart/
4. Qlik, *Data Integration Overview*. URL : https://www.qlik.com/us/data-integration
5. ODI Blog, *ODI Architecture and Components*. URL : https://odielt.wordpress.com/2016/04/16/odi-architecture-and-components/
6. Oracle Documentation, *ODI Studio and Knowledge Modules*. URL : https://docs.oracle.com/en/middleware/fusion-middleware/data-integrator/12.2.1.4/quick-ref/knowledge-modules.html#GUID-FFC6D4D3-13E8-4459-9B50-B2F699D07F7A
7. Oracle Documentation, *Introduction to Oracle Data Integrator*. URL : https://docs.oracle.com/cd/E14571_01/integrate.1111/e12643/intro.htm
8. nextdecision, *Fonctionnalités principales d'ODI*. URL : https://www.next-decision.fr/editeurs-bi/etl/oracle-odi