UNIVERSITE ABDELMALEK ESSAÂDI
ECOLE NATIONALE DES SCIENCES APLIQUER
MASTER BIO-INFORMATIQUE AND DATA SCIENCES
TANGER

RAPPORT DE PROJET D'EVALUATION DE
MODULE DE BASE DE DONNEES COMPLEXE
BISD1

Sujet :

Etude de la base de données « Sale History » (SH) Schéma dans Oracle & MongoDB

Préparés par :

ANNASSIRI Fatima Zahra

EL MAHDALI Oussama

Encadré par

Mr Hassan BADIR

Année Universitaire 2019 / 2020

Résumé

Dans le cadre d'évaluation de module de base de données complexe, notre sujet consiste à mettre en pratique un cas d'usage de schéma d'utilisateurs SH - Sale History - d'Oracle et la migration de cette base de données vers un SGBD NoSQL orienté document : MongoDB. Notre travaille se résume autours les étapes suivants :

- Installation et configuration du schéma SH sur Windows 10 utilisant oracle 12c .
- L'exécution des requêtes simple et complexe sur le schéma à l'aide de SQL Développer.
- La modélisation de schéma SH selon les règles de mongoDB et l'obtention d'un schéma SH sous format JSON proposé.
- La migration des données de schéma SH oracle vers mongoDB à l'aide du Studio 3T
- L'exécution des requêtes NOSQL sur le nouveaux schéma à l'aide du Shell
- une petite comparaison selon l'étude faite entre les deux SGBD.

.

Introduction

Les bases de données relationnelle ont été le fondement de gestion des données d'entreprise depuis plus de trente ans. Mais la façon dont nous construisons et exécutons des applications aujourd'hui, associée à une croissance incessante des nouvelles sources de données et des charges d'utilisateurs pousser les bases de données relationnelles au-delà de leurs limites.

Cela peut inhibent l'agilité commerciale, limitent l'évolutivité et sollicitent les budgets, obligeant de plus en plus d'organisations à migrer vers alternatives. Environ 30% de tous les projets MongoDB sont désormais des migrations à partir de bases de données relationnelles. Cela du a plusieurs critères et exigence répondu par MongoDB.

Dans ce rapport on va parler sur la migration d'une base de données relationnelle SGBDR vers un système de gestion de la base de données NoSQL "orienté document".

Notre objectif est de mettre en pratique un cas d'usage, divisé en deux étapes, la première consiste à installer et configurer un schéma d'utilisateur SH comme étant une base de données source (Relationnelle) en utilisant Oracle 12C, et la deuxième étape consiste à créer une base de données cible (NoSQL) en utilisant MongoDB.

Dans la première partie de ce rapport, on va exposer notre SGBD relationnelle (oracle) et le schéma source SH (installation et configuration) avec lequel on va travailler. Or Oracle propose également de nombreux outils de développement permettant d'automatiser la création d'applications s interfaçant avec la base de données. Ces outils de développement sont SQL Développer qu'il s'agit d'une suite de produits destinés à la conception et à la création d'applications client-serveur. Et SQL*Plus qui est une interface interactive permettant d'envoyer des requêtes SQL et PL/SQL à la base de données.

En deuxième partie, on va exposé d'abord notre SGBD source MongoDB, ensuite nous allons étalé la modélisation des données du schéma SH qu'on a adopté selon les règles du MongoDB. Or MongoDB, est un SGBD NOSQL qui fait partie des bases de type "Documentaire" présentant les données au format JSON (Javascript Object Notation). Il s'agit d'un système qui est le plus populaire pour les applications modernes. MongoDB est particulièrement apprécié pour sa capacité à passer en mode distribué pour répartir le stockage et les traitements.

Table des matières

Contents

Résumé.....3

Introduction.....5

Table des matières 7

Présentation et l'Étude de cas 1 (SGBD Source). 10

1.1 SGBD Source (ORACLE) data base:.....10

1.2 Schéma SH :.....11

1.3 Description des données du Schéma SH :.....12

1.3.1 Le tableau des coûts (COSTS) a les attributs suivants : 12

1.3.2 La table des produits (PRODUCTS) a les attributs suivants :.....12

1.3.3 Le tableau des promotions (PROMOTIONS) a les attributs suivants:.....13

1.3.4 La table des ventes (SALES) a les attributs suivants.....13

1.3.5 La table des canaux (CHANNELS) a les attributs suivants: 13

1.3.6 La table des clients (CUSTOMERS) a les attributs suivants:.....14

1.3.7 La table des temps (TIMES) a les attributs suivants: 14

1.3.8 Le tableau des pays (COUNTRIES) a les attributs suivants: 14

Installation et configuration du schéma SH sur Windows : 16

1.1 Environnement d'installation :.....16

1.1.1 Télécharger le schéma SH : 16

1.1.2 Extraction et configuration du schéma : 16

1.1.3 Modification du script :.....17

1.1.4 Exécution du main script et les valeurs possible pour ses paramètres :.....18

1.1.5 Installez le schéma SH :.....19

Requête sur la base de données source (SH schéma) :.....22

1.1 1ère requête : 22
1.2 1ère requête : 22
1.3 3éme requête : 23
1.4 4éme requête : 24
1.5 5éme requête : 24
1.6 6éme requête : 25
1.7 7éme requête : 26
1.8 8éme requête : 27

Présentation et l'Étude de cas 2 (SGBD Cible) 29

1.1 SGBD Cible - Qu'est-ce que MongoDB ? 29
1.2 Migration de Oracle SH schéma vers MongoDB: 30
1.2.1 Règles à suivre dans la modélisation des données MongoDB: 31
1.2.2 Schéma proposé : 31
1.2.3 Outils utilisé pour la migration des données : 34
1.2.4 Migration des données à l'aide de 3T Studio 35

Requête sur la base de données Cible (SH schéma) : 42

1.1 1er requête : 42
1.2 2ème requête : 42
1.3 3ème requête : 43
1.4 4ème requête : 43
1.5 5ème requête : 44
1.6 6ème requête : 45
1.7 7ème requête : 45
1.8 8ème requête : 45

Comparaison et Synthèse : 47

1.1 Terminologie et concepts 47
1.2 Comparaison des fonctionnalités 48
1.3 Langage de requête 48

1.4 Au niveau de flexibilité : 49

Conclusion Générale 50

Présentation et l'Étude de cas 1 (SGBD Source).

1.1 SGBD Source (ORACLE) data base:

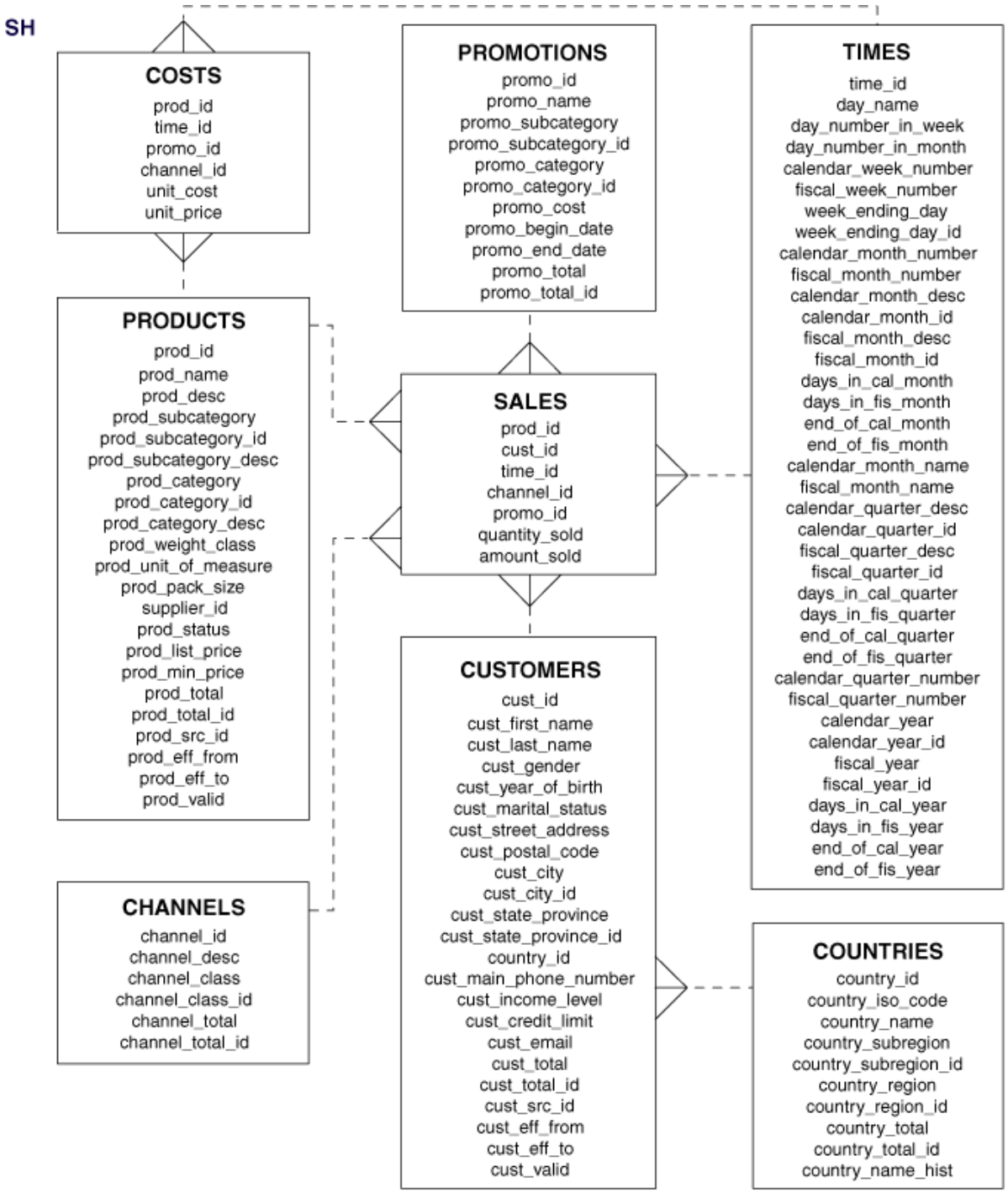
Un système de gestion de base de données (SGBD) est le logiciel qui permet à un ordinateur de stocker, récupérer, ajouter, supprimer et modifier des données. Un SGBD gère tous les aspects primaires d'une base de données, y compris la gestion de la manipulation des données, comme l'authentification des utilisateurs, ainsi que l'insertion ou l'extraction des données.

Un SGBD définit ce qu'on appelle le schéma de données ou la structure dans laquelle les données sont stockées. Les outils que nous utilisons tous au quotidien nécessitent des SGBD en coulisse. Cela comprend les guichets automatiques bancaires, les systèmes de réservation de vols, les systèmes d'inventaire au détail et les catalogues de bibliothèques, par exemple.

Oracle Database est un système de gestion de base de données relationnelle qui depuis l'introduction du support du modèle objet dans sa version 8 peut être aussi qualifié de système de gestion de base de données relationnel-objet .

Oracle est une société technologique mondiale spécialisée dans les systèmes de gestion de bases de données. L'offre de base de données principale est Oracle Database 12c Enterprise Edition (et la prochaine version 18c), qui est vendue via un modèle de licence par traitement avec des modules complémentaires sous licence requis pour des fonctionnalités spécifiques.

1.2 Schéma SH :



Le schéma SH ci-dessus représente Sale History d'une entreprise.

1.3 Description des données du Schéma SH :

La société de l'échantillon réalise un volume d'affaires élevé, de sorte qu'elle exécute des rapports de statistiques commerciales pour aider à la prise de décision. Beaucoup de ces rapports sont basés sur le temps et non volatils. Autrement dit, ils analysent les tendances des données passées.

L'entreprise charge régulièrement des données dans son entrepôt de données pour collecter des statistiques pour ces rapports. Ces rapports incluent les chiffres de ventes annuels, trimestriels, mensuels et hebdomadaires par produit. Ces rapports sont stockés à l'aide du schéma Sales History (SH).

L'entreprise publie également des rapports sur les canaux de distribution par lesquels ses ventes sont livrées. Lorsque l'entreprise organise des promotions spéciales sur ses produits, elle analyse l'impact des promotions sur les ventes. Il analyse également les ventes par zone géographique.

1.3.1 Le tableau des coûts (COSTS) a les attributs suivants :

- Colonnes: prod_id, time_id, promo_id, channel_id, unit_cost, unit_price
- Des relations :
  - La colonne prod_id relie une ou plusieurs lignes des coûts de la table à une ligne des produits de la table avec la valeur correspondante de prod_id.
  - La colonne time_id relie une ou plusieurs lignes des coûts de table à une ligne de la table times avec la valeur correspondante de time_id.

1.3.2 La table des produits (PRODUCTS) a les attributs suivants :

- Colonnes: prod_id (clé primaire), prod_name, prod_desc, prod_subcategory, prod_subcategory_id, prod_subcategory_desc, prod_category, prod_category_id, prod_category_desc, prod_weight_class, prod_unit_of_measure, prod_pack_size, supplier_id, prod_status, prod_list_price, prod_min_price, prod_total, prod_total_id, prod_src_id, prod_eff_from, prod_eff_to et prod_valid.

- Des relations:
    - La colonne prod_id relie les lignes des produits de la table à une ou plusieurs lignes de la table sales avec la valeur correspondante de prod_id.
    - La colonne prod_id relie les lignes des produits de la table à une ou plusieurs lignes des coûts de la table avec la valeur correspondante de prod_id.

1.3.3 Le tableau des promotions (PROMOTIONS) a les attributs suivants:

- Colonnes: promo_id (clé primaire), promo_name, promo_subcategory, promo_subcategory_id, promo_category, promo_category_id, promo_cost, promo_begin_date, promo_end_date, promo_total et promo_total_id
- Des relations: La colonne promo_id relie les lignes de la table des promotions à une ou plusieurs lignes de la table sales avec la valeur correspondante de promo_id.

1.3.4 La table des ventes (SALES) a les attributs suivants

- Colonnes: prod_id, cust_id, time_id, channel_id, promo_id, quantity_sold et amount_sold
- Des relations:
    - La colonne prod_id relie une ou plusieurs lignes de la table sales à une ligne de la table products avec la valeur correspondante de prod_id.
    - La colonne promo_id relie une ou plusieurs lignes de la table des ventes à une ligne de la table des promotions avec la valeur correspondante de promo_id.
    - La colonne channel_id relie une ou plusieurs lignes de la table sales à une ligne de la table channels avec la valeur correspondante de channel_id.
    - La colonne cust_id relie une ou plusieurs lignes de la table sales à une ligne de la table customers avec la valeur correspondante cust_id.
    - La colonne times_id relie une ou plusieurs lignes de la table sales à une ligne de la table times avec la valeur correspondante de times_id.

1.3.5 La table des canaux (CHANNELS) a les attributs suivants:

- Colonnes: channel_id (clé primaire), channel_desc, channel_class, channel_class_id, channel_total et channel_total_id
- Des relations: La colonne channel_id relie les lignes des canaux de la table à une ou plusieurs lignes de la table sales avec la valeur correspondante de channel_id.

1.3.6 La table des clients (CUSTOMERS) a les attributs suivants:

- Colonnes: cust_id (clé primaire), cust_first_name, cust_last_name, cust_gender, cust_year_of_birth, cust_marital_status, cust_street_address, cust_postal_code, cust_city, cust_city_id, cust_state_province, cust_state_provice_id, country_id, cust_main_phone_number,cust_income_level,cust_credit_limit,cust_email,cust_total, cust_total_id, cust_src_id, cust_eff_from, cust_eff_to, et cust_valid.

- Des relations:

  - La colonne cust_id relie les lignes de la table clients à une ou plusieurs lignes de la table sales avec la valeur correspondante de customer_id.

  - La colonne country_id relie une ou plusieurs lignes de la table customers à une ligne de la table countries avec la valeur correspondante de country_id

1.3.7 La table des temps (TIMES) a les attributs suivants:

- Colonnes: TIME_ID (clé primaire), day_name, day_number_in_week, day_number_in_month,calendar_week_number,fiscal_week_number,week_ending_day, week_ending_day_id,calendar_month_number,fiscal_month_numbercalendar_month_desc, calendar_month_id, fiscal_month_desc, fiscal_month_id, days_in_cal_month, days_in_fis_month, end_of_cal_month, end_of_fis_month, calendar_month_name,fiscal_month_name,calendar_quarter_desc,calendar_quarter_id, fiscal_quarter_desc, fiscal_quarter_id, days_in_cal_quarter, days_in_fis_quarter, end_of_cal_quarter,end_of_fis_quarter,calendar_quarter_number,fiscal_quarter_number, calendar_year,calendar_year_id,fiscal_ycal, fiscal_year_idin_id, days_in_calaire, fiscal_year_idin_de_facile, jours_calaire_de_fiscal, année_de_calaire .

- Des relations: La colonne time_id relie les lignes de la table times à une ou plusieurs lignes de la table sales avec la valeur correspondante de time_id.

1.3.8 Le tableau des pays (COUNTRIES) a les attributs suivants:

- Colonnes:country_id (clé primaire), country_iso_code, country_name, country_subregion, country_subregion_id, country_region, country_region_id, country_total, country_total_id, country_name_hist.

- Des relations: La colonne country_id relie les lignes de la table country à une ou plusieurs lignes de la table customers avec la valeur correspondante de country_id.

.

Installation et configuration du schéma SH sur Windows :

1.1 Environnement d'installation :

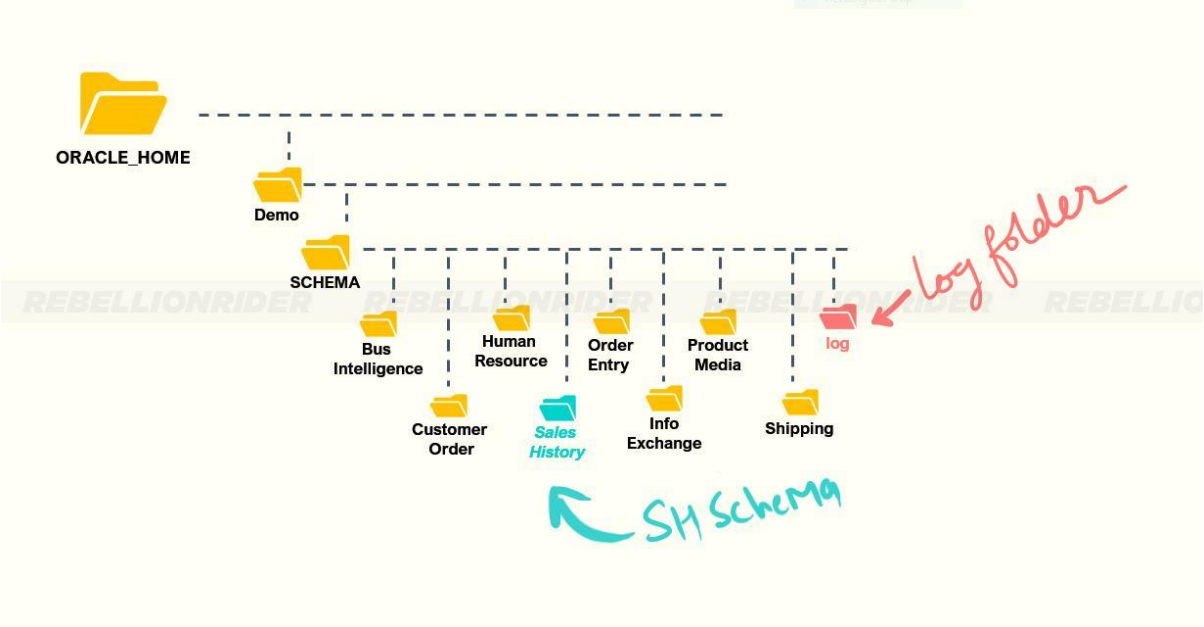
1.1.1 Télécharger le schéma SH :

Oracle Database 12c Release 2, ne fournit pas d'exemples de schémas avec l'ensemble par défaut, sauf le schéma de l'utilisateur HR. Par conséquent, nous devons les télécharger depuis GitHub. Sur le lien suivant : https://github.com/oracle/db-sample-schemas/releases

1.1.2 Extraction et configuration du schéma :

Une fois le téléchargement terminé, il est de préférence d'extraire le contenu du fichier zip dans le répertoire qu'Oracle Universal Installer a créé pour tous les exemples de schémas. L'emplacement de ce répertoire est : % oracle_home% \ demo \ schema .

Dans ce répertoire, l'existence de dossier ''log'' est important dans la procédure d'installation d'exemples de schéma. Par conséquent, assurez-vous que vous disposez de ce dossier ou vous pouvez le crier et le renommer manuellement. Une fois cette étape est terminée, la structure de répertoires ressemblera à ceci :



Dans le dossier '' sales history'' Il existe au total 4 scripts qui sont responsable de la création et la configuration du schéma SH dans la base de données Oracle. Ces 4 scripts sont :

- sh_main.sql - Ceci est le script principal (main script).
- csh_v3.sql - Ce script créera les objets
- lsh_v3.sql - Ce script chargera les tables
- psh_v3.sql - Ce script effectuera les opérations de post-chargement.

L'exécution du script «sh_main.sq » lance en son tour l'exécution des trois autres scripts.

1.1.3 Modification du script :

D'une voie générale la plupart de ces scripts sont optimisés pour les systèmes d'exploitation Linux. Par conséquent, c'est normal de rencontrer des erreurs l'ors de l'installation du schéma SH sur une machine Windows. Afin d'éviter ceux-ci, il faut faire quelques légères modifications seulement sur ces deux scripts : `sh_main.sql` et `lsh_v3.sql` .

Commençons par le "`sh_main.sql`", dans ce script, il faut remplacer la commande __SUB__CWD__ par l'emplacement du répertoire des scripts. Cette emplacement s'changera d'une machine à l'autre :

Before

__SUB__CWD__/sales_history/csh_&ver

__SUB__CWD__/sales_history/lsh_&ver

__SUB__CWD__/sales_history/psh_&ver

After

C:/App/db_home/demo/schema/sales_history/csh_&ver

C:/App/db_home/demo/schema/sales_history/lsh_&ver

C:/App/db_home/demo/schema/sales_history/psh_&ver

Dans le deuxième script `lsh_v3.sql`, il faut simplement remplacer le mot-clé NEWLINE par le caractère d'échappement «\ n».

1.1.4 Exécution du main script et les valeurs possible pour ses paramètres :

Pour exécuter de sh_main.sql script, nous aurons besoin d'un total de 9 éléments. Parmi ces 9 premiers se trouvera l'emplacement du script sh_main.sql. Rest 8 sera les paramètres pour l'installation du schéma SH dans Oracle Database. Ces 8 paramètres sont :

1 - Mot de passe que vous souhaitez attribuer à votre utilisateur SH.

2 - Nom du tablespace par défaut

3 - Nom du tablespace temporaire

4 - Mot de passe de votre utilisateur SYS

5 - Chemin du répertoire de vos fichiers de données. Ce sera le chemin où le compilateur peut trouver tous les fichiers liés au schéma SH. Comme lsh_v3.sql, csh_v3.sql, psh_v3.sql et autres.

6 - Chemin de l'emplacement de votre dossier de journal

7 - Version du schéma que vous installez.

8 - chaîne de connexion

Pour installer le schéma SH dans Oracle Database, nous devons spécifier les valeurs de chacun de ces paramètres et dans le même ordre.

1.1.5 Installez le schéma SH :

Maintenant pour l'installation du schéma SH, il faut combiner tous ces paramètres et les mettre en une seule chaîne. Pour faciliter les étapes comme ce qui est schématisé ici et suivant ses étapes :

Lancez l'invite de commande avec les privilèges administratifs, puis établi la connexion à la base de données en utilisant l'utilisateur SYS a l'aide de la commande suivant : `SQLPLUS / as sysdba`

```
Command Prompt - sqlplus
Microsoft Windows [Version 10.0.18363.1049]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\OussamaMahdali>sqlplus

SQL*Plus: Release 12.2.0.1.0 Production on Sat Aug 29 21:46:51 2020

Copyright (c) 1982, 2016, Oracle.  All rights reserved.

Enter user-name: sys as sysdba
Enter password:

Connected to:
Oracle Database 12c Enterprise Edition Release 12.2.0.1.0 - 64bit Production
```

Ainsi il faut être sur que la base de données enfichable (pluggable Database) dans laquelle vous installez le schéma SH est ouverte pour une opération de lecture / écriture :

```
COLUMN name FORMAT a10;
SELECT name, open_mode FROM v$pdbs;
```

```
SQL> COLUMN name FORMAT a10;
SQL> SELECT name, open_mode FROM v$pdbs;

NAME       OPEN_MODE
---------- ----------
PDB$SEED   READ ONLY
ORCLPDB    READ WRITE
```

Si une erreur est produite il faut être sur de changer la session. :

```
ALTER SESSION SET CONTAINER = orclpdb;
```

```
SQL> ALTER SESSION SET CONTAINER = orclpdb;

Session altered.

SQL>
```

Ensuite, l'appelle du script, ce dernier peut être réaliser selon deux façons :

- L'appelle simple du script en utilisant @? Suivi de l'emplacement du script sh_main.sql comme ceci : `@?\demo\schema\sales_history\sh_main.sql` et remplir les valeurs des paramètres par clavier chaque fois sont demandé .

- Ou Suivi de l'emplacement du script écrivez tous les paramètres dont nous avons discuté ci-dessus dans le même ordre, a la façons suivant : `@?/demo/schema/sales_history/sh_main.sql sh USERS TEMP oracle C:/App/db_home/demo/schema/sales_history/ %ORACLE_HOME%/demo/schema/log/ v3 localhost:1521/orclpdb`

Dans notre cas on a choisi la deuxième méthode Comme illustré dans la figure suivante :

```
Command Prompt - sqlplus
SQL> @?/demo/schema/sales_history/sh_main.sql sh_pass USERS TEMP oracle D:/app/OussamaMahdali/virtual/product/12.2.0/dbh
ome_1/demo/schema/sales_history/ D:/app/OussamaMahdali/virtual/product/12.2.0/dbhome_1/demo/schema/log/ v3 localhost:152
1/orclpdb
```

L'installation donc est faite avec succès :

```
Comment created.


Comment created.


Comment created.


gathering statistics ...

PL/SQL procedure successfully completed.


PL/SQL procedure successfully completed.

SQL> show user;
USER is "SH"
SQL>
```

Voilà les noms des tables contenu dans le schéma SH :

```
SQL> SELECT table_name FROM user_tables;

TABLE_NAME
--------------------------------------------------------------------
CAL_MONTH_SALES_MV
FWEEK_PSCAT_SALES_MV
DR$SUP_TEXT_IDX$I
DR$SUP_TEXT_IDX$K
DR$SUP_TEXT_IDX$R
DR$SUP_TEXT_IDX$N
DR$SUP_TEXT_IDX$U
SALES
COSTS
TIMES
PRODUCTS
CHANNELS
PROMOTIONS
CUSTOMERS
COUNTRIES
SUPPLEMENTARY_DEMOGRAPHICS
SALES_TRANSACTIONS_EXT

17 rows selected.
```

Requête sur la base de données source (SH schéma) :

Cette partie est réalisé a l'aide de sql développer, ce de suite les requête et leurs résultat sous forme des capture d'écran :

1.1 1ère requête :

```
SELECT * FROM TIMES WHERE CALENDAR_MONTH_NAME = 'May' AND
CALENDAR_YEAR = '2001';
```

```
SELECT * FROM TIMES WHERE CALENDAR_MONTH_NAME = 'May' AND CALENDAR_YEAR = '2001';
```

Script Output

Task completed in 0.085 seconds

14-MAY-01	Monday	1	14	20	20	20-MAY-01	1598	
15-MAY-01	Tuesday	2	15	20	20	20-MAY-01	1598	
16-MAY-01	Wednesday	3	16	20	20	20-MAY-01	1598	
17-MAY-01	Thursday	4	17	20	20	20-MAY-01	1598	
18-MAY-01	Friday	5	18	20	20	20-MAY-01	1598	
19-MAY-01	Saturday	6	19	20	20	20-MAY-01	1598	
20-MAY-01	Sunday	7	20	20	20	20-MAY-01	1598	
21-MAY-01	Monday	1	21	21	21	27-MAY-01	1646	
22-MAY-01	Tuesday	2	22	21	21	27-MAY-01	1646	

TIME_ID	DAY_NAME	DAY_NUMBER_IN_WEEK	DAY_NUMBER_IN_MONTH	CALENDAR_WEEK_NUMBER	FISCAL_WEEK_NUMBER	WEEK_ENDING_DAY	WEEK_ENDING_DAY_ID	CALENI
23-MAY-01	Wednesday	3	23	21	21	27-MAY-01	1646	
24-MAY-01	Thursday	4	24	21	21	27-MAY-01	1646	
25-MAY-01	Friday	5	25	21	21	27-MAY-01	1646	
26-MAY-01	Saturday	6	26	21	21	27-MAY-01	1646	
27-MAY-01	Sunday	7	27	21	21	27-MAY-01	1646	
28-MAY-01	Monday	1	28	22	22	03-JUN-01	1478	
29-MAY-01	Tuesday	2	29	22	22	03-JUN-01	1478	
30-MAY-01	Wednesday	3	30	22	22	03-JUN-01	1478	
31-MAY-01	Thursday	4	31	22	22	03-JUN-01	1478	

31 rows selected

1.2 1ère requête :

```
SELECT  * FROM  CUSTOMERS WHERE CUST_MARITAL_STATUS IS NOT NULL AND
rownum <= 100 ;
```

```
SELECT  * FROM  CUSTOMERS WHERE CUST_MARITAL_STATUS IS NOT NULL AND rownum <= 100 ;
```

Script Output

Task completed in 0.13 seconds

```
    30781 Abel                Vance                                  M              1977 married              87 South Tioga Avenue

   CUST_ID CUST_FIRST_NAME      CUST_LAST_NAME                         C CUST_YEAR_OF_BIRTH CUST_MARITAL_STATUS  CUST_STREET_ADDRESS
---------- -------------------- ---------------------------------------- - ------------------ -------------------- ----------------------------------------
    38782 Abel                Walsh                                  M              1924 married              107 West Tioga Avenue
    16559 Abel                Walsh                                  M              1925 single               17 East Pacific Court
    46115 Abel                Walsh                                  M              1974 married              27 Corson Boulevard
     3227 Abel                Walsh                                  M              1930 single               37 West Genesee Street
     6782 Abel                Walsh                                  M              1951 married              47 Tioga Road
    13893 Abel                Walsh                                  M              1941 single               57 West Zapata Drive
    17448 Abel                Walsh                                  M              1943 married              67 East McHenry Avenue
    24560 Abel                Walsh                                  M              1928 single               77 North Pacific Avenue
    28115 Abel                Walsh                                  M              1927 married              87 West Corson Avenue
    35226 Abel                Walsh                                  M              1976 single               97 South Genesee Avenue
    41449 Abigail             Grimes                                 M              1967 single               117 Mcintosh Avenue

   CUST_ID CUST_FIRST_NAME      CUST_LAST_NAME                         C CUST_YEAR_OF_BIRTH CUST_MARITAL_STATUS  CUST_STREET_ADDRESS
---------- -------------------- ---------------------------------------- - ------------------ -------------------- ----------------------------------------
    45004 Abigail             Grimes                                 M              1951 married              117 South Bradford Avenue

100 rows selected
```

1.3 3éme requête :

```
SELECT PROD_ID, PROD_NAME, PROD_SUBCATEGORY, PROD_CATEGORY FROM PRODUCTS ORDER BY PROD_CATEGORY, PROD_SUBCATEGORY, PROD_NAME;
```

```
SELECT PROD_ID, PROD_NAME, PROD_SUBCATEGORY, PROD_CATEGORY FROM PRODUCTS
ORDER BY PROD_CATEGORY, PROD_SUBCATEGORY, PROD_NAME;
```

Script Output

Task completed in 0.086 seconds

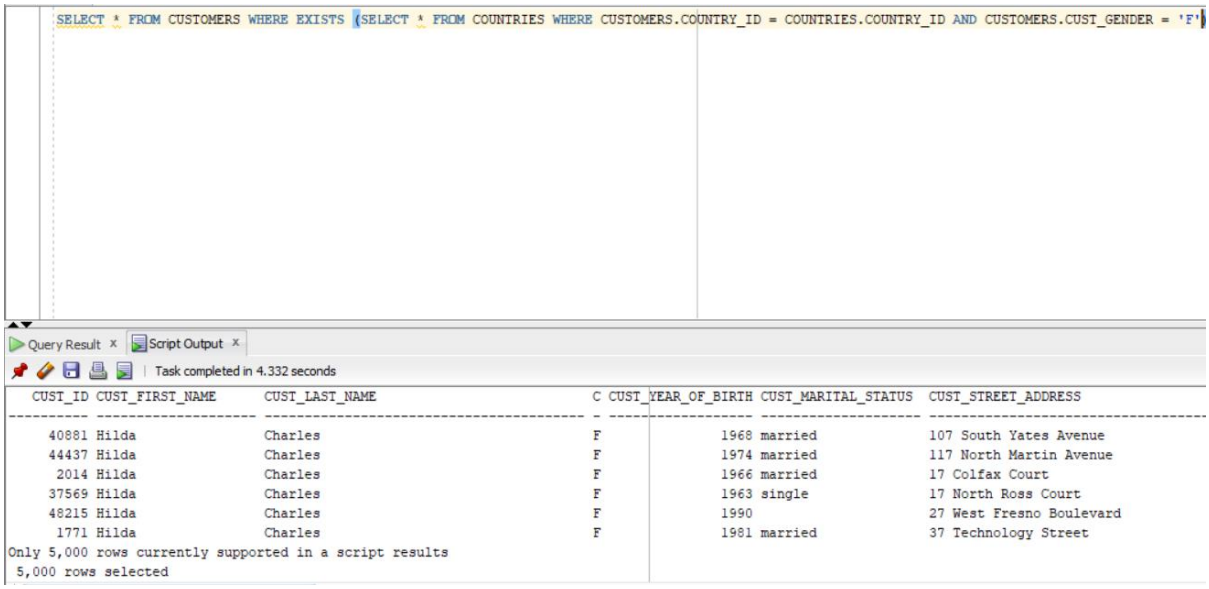
```
        41 O/S Documentation Set - German                    Documentation                                    Software/Other
        44 O/S Documentation Set - Italian                   Documentation                                    Software/Other
        45 O/S Documentation Set - Kanji                     Documentation                                    Software/Other
        43 O/S Documentation Set - Spanish                   Documentation                                    Software/Other
        28 Unix/Windows 1-user pack                          Operating Systems                                Software/Other
       113 CD-R Mini Discs                                   Recordable CDs                                   Software/Other
       119 CD-R with Jewel Cases, pACK OF 12                 Recordable CDs                                   Software/Other
       117 CD-R, Professional Grade, Pack of 10              Recordable CDs                                   Software/Other
       116 CD-RW, High Speed Pack of 5                       Recordable CDs                                   Software/Other
       115 CD-RW, High Speed, Pack of 10                     Recordable CDs                                   Software/Other
       114 Music CD-R                                        Recordable CDs                                   Software/Other

   PROD_ID PROD_NAME                                         PROD_SUBCATEGORY                                 PROD_CATEGORY
---------- ------------------------------------------------- ------------------------------------------------ -----------------
       118 OraMusic CD-R, Pack of 10                         Recordable CDs                                   Software/Other
       120 DVD-R Disc with Jewel Case, 4.7 GB                Recordable DVD Discs                             Software/Other
       123 DVD-R Discs, 4.7GB, Pack of 5                     Recordable DVD Discs                             Software/Other
       122 DVD-R Discs, 4.7GB, Pack of 5                     Recordable DVD Discs                             Software/Other
       121 DVD-RAM Jewel Case, Double-Sided, 9.4G            Recordable DVD Discs                             Software/Other
       124 DVD-RW Discs, 4.7GB, Pack of 3                    Recordable DVD Discs                             Software/Other

72 rows selected
```

1.4 4éme requête :

```
SELECT * FROM CUSTOMERS WHERE EXISTS (SELECT * FROM COUNTRIES WHERE
CUSTOMERS.COUNTRY_ID = COUNTRIES.COUNTRY_ID AND CUSTOMERS.CUST_GENDER
= 'F');
```



1.5 5éme requête :

```
SELECT
  a.cust_id,
  a.cust_last_name || ', ' || a.cust_first_name as customer_name,
  a.cust_city || ', ' || a.cust_state_province || ', ' ||
a.country_id as city_id,
  b.country_name,
  b.country_region as region
FROM CUSTOMERS a, COUNTRIES b
WHERE a.country_id = b.country_id AND CUST_VALID = 'I' AND
CUST_MARITAL_STATUS = 'married' AND rownum <= 100
ORDER BY region, country_name, city_id;
```

```
SELECT
  a.cust_id,
  a.cust_last_name || ', ' || a.cust_first_name as customer_name,
  a.cust_city || ', ' || a.cust_state_province || ', ' || a.country_id as city_id,
  b.country_name,
  b.country_region as region
FROM CUSTOMERS a, COUNTRIES b
WHERE a.country_id = b.country_id AND CUST_VALID = 'I' AND CUST_MARITAL_STATUS = 'married' AND country_region = 'Oceania'
ORDER BY region, country_name, city_id;
```

Script Output

Task completed in 0.239 seconds

```
    27980 Gradden, Grichrist                                   Wellington, Wellington, 52785
    33160 Gilboy, Randy                                        Wellington, Wellington, 52785
    12395 Cain, Guthrey                                        Wellington, Wellington, 52785
     6161 Bakker, Grady                                        Wellington, Wellington, 52785
    20700 Kuhler, Rae                                          Wellington, Wellington, 52785
    29700 Banas, Salisbury                                     Wellington, Wellington, 52785
     3395 Tavener, Diamond                                     Wellington, Wellington, 52785
    42161 Rossellett, Terry                                    Wellington, Wellington, 52785
    45782 Figgens, Paulette                                    Wellington, Wellington, 52785
    47161 Haarper, Della                                       Wellington, Wellington, 52785

   CUST_ID CUSTOMER_NAME                                       CITY_ID
---------- ------------------------------------------------------------ ------------------------------------------------------
     4782 Lengel, Roscoe                                       Wellington, Wellington, 52785
    38005 Weatherford, Byron                                   Wellington, Wellington, 52785
    36618 Paige, Mitchel                                       Wellington, Wellington, 52785
    42847 Callihan, Morton                                     Wellington, Wellington, 52785
    34914 Park, Bertilde                                       Wellington, Wellington, 52785
    11700 Lassiter, Marshal                                    Wellington, Wellington, 52785
      191 Whitehead, Stacia                                    Wellington, Wellington, 52785
    20005 Rowe, Troy                                           Wellington, Wellington, 52785

206 rows selected
```

1.6 6éme requête :

```
SELECT channel_desc, TO_CHAR(SUM(amount_sold),'9,999,999') SALES
FROM sh.sales, sh.products, sh.customers, sh.times, sh.channels,
sh.countries
WHERE sales.prod_id=products.prod_id AND
sales.cust_id=customers.cust_id
  AND customers.country_id = countries.country_id AND
sales.time_id=times.time_id
  AND sales.channel_id=channels.channel_id
  AND times.calendar_month_desc IN ('2000-09', '2000-10')
  AND country_iso_code='US'
GROUP BY channel_desc
ORDER BY SALES;
```

```
SELECT channel_desc, TO_CHAR(SUM(amount_sold),'9,999,999') SALES
FROM sh.sales, sh.products, sh.customers, sh.times, sh.channels, sh.countries
WHERE sales.prod_id=products.prod_id AND sales.cust_id=customers.cust_id
  AND customers.country_id = countries.country_id AND sales.time_id=times.time_id
  AND sales.channel_id=channels.channel_id
  AND times.calendar_month_desc IN ('2000-09', '2000-10')
  AND country_iso_code='US'
GROUP BY channel_desc
ORDER BY SALES;
```

Script Output

Task completed in 0.133 seconds

```
CHANNEL_DESC         SALES
-------------------- ----------
Internet                261,278
Partners                800,871
Direct Sales          1,320,497
```

1.7 7éme requête :

```
SELECT    p.prod_name, c.cust_gender, t.day_name, SUM(s.amount_sold)
AS TotalSold
FROM      SALES s, PRODUCTS p, CUSTOMERS c, TIMES t
WHERE     p.prod_id = s.prod_id
AND     c.cust_id = s.cust_id
AND     t.time_id = s.time_id
AND     p.prod_name = 'Deluxe Mouse'
AND     c.cust_gender = 'F'
AND     t.day_number_in_week = 1
GROUP BY p.prod_name, c.cust_gender, t.day_name ;
```

```
SELECT   p.prod_name, c.cust_gender, t.day_name, SUM(s.amount_sold) AS TotalSold
     FROM     SALES s, PRODUCTS p, CUSTOMERS c, TIMES t
     WHERE    p.prod_id = s.prod_id
       AND    c.cust_id = s.cust_id
       AND    t.time_id = s.time_id
       AND    p.prod_name = 'Deluxe Mouse'
       AND    c.cust_gender = 'F'
       AND    t.day_number_in_week = 1
     GROUP BY p.prod_name, c.cust_gender, t.day_name ;
```

Script Output

Task completed in 0.155 seconds

```
PROD_NAME                                          C DAY_NAME   TOTALSOLD
-------------------------------------------------- - --------- ----------
Deluxe Mouse                                       F Monday      19053.35
```

1.8 8éme requête :

```
SELECT PROD_ID, PROD_NAME FROM PRODUCTS WHERE PROD_LIST_PRICE < 100
```

Worksheet | Query Builder

```
SELECT PROD_ID, PROD_NAME FROM PRODUCTS WHERE PROD_LIST_PRICE < 100;
```

Script Output

Task completed in 0.024 seconds

```
       126 3 1/2" Bulk diskettes, Box of 100
       127 Model CD13272 Tricolor Ink Cartridge
       128 Model SM26273 Black Ink Cartridge
       130 Model A3827H Black Image Cartridge
       131 Model K3822L Cordless Phone Battery
       132 Model C9827B Cordless Phone Battery
       133 Model K8822S Cordless Phone Battery

   PROD_ID PROD_NAME
---------- --------------------------------------------------
       134 Model C93822D Wireless Phone Battery
       135 S27273M Extended Use w/l Phone Batt.
       136 64MB Memory Card
       137 128MB Memory Card
       138 256MB Memory Card
       139 Bounce
       140 Endurance Racing
       141 Smash up Boxing
       142 Martial Arts Champions
       143 Comic Book Heroes
       144 Fly Fishing

   PROD_ID PROD_NAME
---------- --------------------------------------------------
       145 Finding Fido
       146 Adventures with Numbers
       147 Extension Cable
       148 Xtend Memory

 59 rows selected
```

Présentation et l'Étude de cas 2 (SGBD Cible).

Ce chapitre se concentre sur MongoDB, vu comme une base distribuée pour le stockage de documents JSON. MongoDB cherche à répondre aux besoins de performance, à garantir la scalabilité horizontale (réplication et sharding) en offrant de nombreuses fonctionnalités qu'on trouve dans le monde relationnel. Il intègre aussi le support de recherche full-text et les traitements de type MapReduce et aussi la recherche géospatiale.

1.1 SGBD Cible - Qu'est-ce que MongoDB ?

MongoDB est une base de données non relationnelle développée par MongoDB, Inc. MongoDB stocke les données sous forme de documents dans une représentation binaire appelée BSON (Binary JSON).

Les informations associées sont stockées ensemble pour un accès rapide aux requêtes via le langage de requête MongoDB. Les champs peuvent varier d'un document à l'autre; il n'est pas nécessaire de déclarer la structure des documents au système - les documents sont auto-descriptifs.

Si un nouveau champ doit être ajouté à un document, le champ peut être créé sans affecter tous les autres documents de la collection, sans mettre à jour un catalogue système central et sans mettre le système hors ligne. En option, la validation de schéma peut être utilisée pour appliquer des contrôles de gouvernance des données sur chaque collection.

Le modèle de données de document de MongoDB correspond naturellement aux objets du code d'application, ce qui facilite l'apprentissage et l'utilisation par les développeurs. Les documents vous permettent de représenter des relations hiérarchiques pour stocker facilement des tableaux et d'autres structures plus complexes.

Des pilotes natifs et idiomatiques sont fournis pour plus de 10 langues - et la communauté en a construit des dizaines d'autres - permettant des requêtes ad hoc, une agrégation en temps réel et une indexation

riche pour fournir des moyens programmatiques puissants pour accéder et analyser les données de toute structure.

Étant donné que les documents peuvent rassembler des données associées qui seraient autrement modélisées sur des tables parent-enfant séparées dans un schéma relationnel, les opérations atomiques à document unique de MongoDB fournissent déjà une sémantique de transaction qui répond aux besoins d'intégrité des données de la majorité des applications. Un ou plusieurs champs peuvent être écrits en une seule opération, y compris des mises à jour de plusieurs sous-documents et éléments d'un tableau. Les garanties fournies par MongoDB assurent une isolation complète lors de la mise à jour d'un document; toute erreur entraîne la restauration de l'opération afin que les clients reçoivent une vue cohérente du document.

Contrairement à Oracle et à d'autres bases de données relationnelles, MongoDB est construit sur une architecture de systèmes distribués, plutôt que sur une conception monolithique à nœud unique. En conséquence, MongoDB propose une évolutivité et une localisation de données prêtes à l'emploi avec partitionnement automatique et des ensembles de réplicas pour maintenir une disponibilité permanente.

1.2 Migration de Oracle SH schéma vers MongoDB:

La migration des bases de données relationnelle vers MongoDB n'est pas une mince affaire. Il faut remodeler le schéma de données pour l'adapter à MongoDB.

Dans ce paragraphe en fait la démonstration de migrer le SH schéma de base de données Oracle vers MongoDB.

1.2.1 Règles à suivre dans la modélisation des données MongoDB:

MongoDB a offert un chapitre entier de documentation qui est dédié Certaines des limitations ou règles dont vous devez être conscient:

- Il y a une taille maximale de 16 Mo par document
- La taille totale de l'entrée d'index ne peut pas dépasser 1 Ko
- Vous ne pouvez pas incorporer plus de 100 documents sous / imbriqués.

1.2.2 Schéma proposé :

Basent sur la documentation officielle de MongoDB sur la modélisation des données, en aura le schéma suivant :

```
{
 "_id": {
 "PROD": {},
  "PROD_ID": {},
  "PROD_NAME": {},
  "PROD_DESC": {},
  "PROD_SUBCATEGORY": {},
  "PROD_SUBCATEGORY_ID": {},
  "PROD_SUBCATEGORY_DESC": {},
  "PROD_CATEGORY": {},
  "PROD_CATEGORY_ID": {},
  "PROD_CATEGORY_DESC": {},
  "PROD_WEIGHT_CLASS": {},
  "PROD_UNIT_OF_MEASURE": {},
  "PROD_PACK_SIZE": {},
  "PROD_SUPPLIER_ID": {},
  "PROD_STATUS": {},
  "PROD_LIST_PRICE": {},
  "PROD_MIN_PRICE": {},
  "PROD_TOTAL": {},
```

```
 "PROD_TOTAL_ID": {},
 "PROD_SRC_ID": {},
 "PROD_EFF_FROM": {},
 "PROD_EFF_TO": {},
 "PROD_VALID": {},
 "UNIT_COST": {},
 "UNIT_PRICE": {},
},

"CUST": {
 "CUST_ID": {},
 "CUST_FIRST_NAME": {},
 "CUST_LAST_NAME": {},
 "CUST_GENDER": {},
 "CUST_YEAR_OF_BIRTH": {},
 "CUST_MARITAL_STATUS": {},
 "CUST_STREET_ADDRESS": {},
 "CUST_POSTAL_CODE": {},
 "CUST_CITY": {},
 "CUST_CITY_ID": {},
 "CUST_STATE_PROVINCE": {},
 "CUST_STATE_PROVINCE_ID": {},
 "COUNTRY": {
 "COUNTRY_ID": {},
  "COUNTRY_ISO_CODE": {},
  "COUNTRY_NAME": {},
  "COUNTRY_SUBREGION": {},
  "COUNTRY_SUBREGION_ID": {},
  "COUNTRY_REGION": {},
  "COUNTRY_REGION_ID": {},
  "COUNTRY_TOTAL": {},
  "COUNTRY_TOTAL_ID": {},
  "COUNTRY_NAME_HIST": {}
},
 "CUST_MAIN_PHONE_NUMBER": {},
 "CUST_INCOME_LEVEL": {},
 "CUST_CREDIT_LIMIT": {},
 "CUST_EMAIL": {},
 "CUST_TOTAL": {},
 "CUST_TOTAL_ID": {},
 "CUST_SRC_ID": {},
 "CUST_EFF_FROM": {},
 "CUST_EFF_TO": {},
 "CUST_VALID": {}
},
"TIME": {
 "TIME_ID": {},
 "DAY_NAME": {},
 "DAY_NUMBER_IN_WEEK": {},
```

```
 "DAY_NUMBER_IN_MONTH": {},
 "CALENDAR_WEEK_NUMBER": {},
 "FISCAL_WEEK_NUMBER": {},
 "WEEK_ENDING_DAY": {},
 "WEEK_ENDING_DAY_ID": {},
 "CALENDAR_MONTH_NUMBER": {},
 "FISCAL_MONTH_NUMBER": {},
 "CALENDAR_MONTH_DESC": {},
 "CALENDAR_MONTH_ID": {},
 "FISCAL_MONTH_DESC": {},
 "FISCAL_MONTH_ID": {},
 "DAYS_IN_CAL_MONTH": {},
 "DAYS_IN_FIS_MONTH": {},
 "END_OF_CAL_MONTH": {},
 "END_OF_FIS_MONTH": {},
 "CALENDAR_MONTH_NAME": {},
 "FISCAL_MONTH_NAME": {},
 "CALENDAR_QUARTER_DESC": {},
 "CALENBDAR_QUARTER_ID": {},
 "FISCAL_QUARTER_DESC": {},
 "FISCAL_QUARTER_ID": {},
 "DAYS_IN_CAL_QUARTER": {},
 "DAYS_IN_FIS_QUARTER": {},
 "END_OF_CAL_QUARTER": {},
 "END_OF_FIS_QUARTER": {},
 "CALENDAR_QUARTER_NUMBER": {},
 "FISCAL_QUARTER_NUMBER": {},
 "CALENDAR_YEAR": {},
 "CALENDAR_YEAR_ID": {},
 "FISCAL_YEAR": {},
 "FISCAL_YEAR_ID": {},
 "DAYS_IN_CAL_YEAR": {},
 "DAYS_IN_FIS_YEAR": {},
 "END_OF_CAL_YEAR": {},
 "END_OF_FIS_YEAR": {}
},
"CHANNEL": {
 "CHANNEL_ID": {},
 "CHANNEL_DESC": {},
 "CHANNEL_CLASS": {},
 "CHANNEL_CLASS_ID": {},
 "CHANNEL_TOTAL": {},
 "CHANNEL_TOTAL_ID": {}
},
"PROMO": {
 "PROMO_ID": {},
 "PROMO_NAME": {},
 "PROMO_SUBCATEGORY": {},
 "PROMO_SUBCATEGORY_ID": {},
```

```
  "PROMO_CATEGORY": {},
  "PROMO_CATEGORY_ID": {},
  "PROMO_COST": {},
  "PROMO_BEGIN_DATE": {},
  "PROMO_END_DATE": {},
  "PROMO_TOTAL": {},
  "PROMO_TOTAL_ID": {}
 },
 "QUANTITY_SOLD": {},
 "AMOUNT_SOLD": {}
}
```

1.2.3 Outils utilisé pour la migration des données :

1.2.3.1 Studio 3T :

3T Studio offert une interface graphique client pour MongoDb.il est disponible pour Windows, macOS et Linux.

Il permet de créer des requêtes rapidement, générez du code instantané, importez / exportez dans plusieurs formats et bien plus encore il avait l'option de SQL to MongoDB Migration, qui consiste à migrer des données d'un serveur de base de données relationnelle vers un serveur MongoDB.

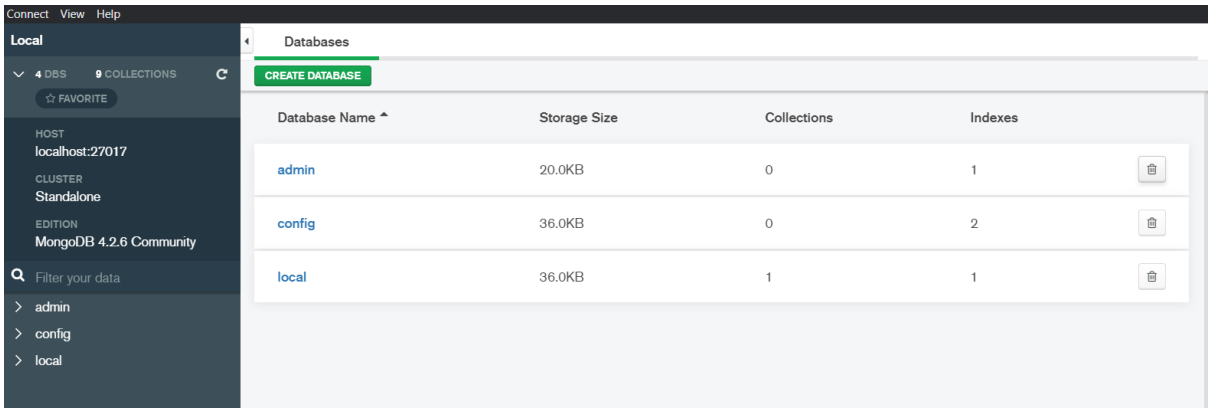
Il permet donc d'importer plusieurs tables SQL ou ensembles de données personnalisés dans une seule collection MongoDB et de mapper les relations de table (un à un, un à plusieurs) vers JSON selon les besoins - des fonctionnalités jamais disponibles auparavant dans l'espace d'outils MongoDB.

1.2.3.2 MongoDB Compass :

En tant qu'interface graphique de MongoDB, MongoDB Compass vous permet de prendre des décisions plus intelligentes concernant la structure du document, l'interrogation, l'indexation, la validation de document, etc.

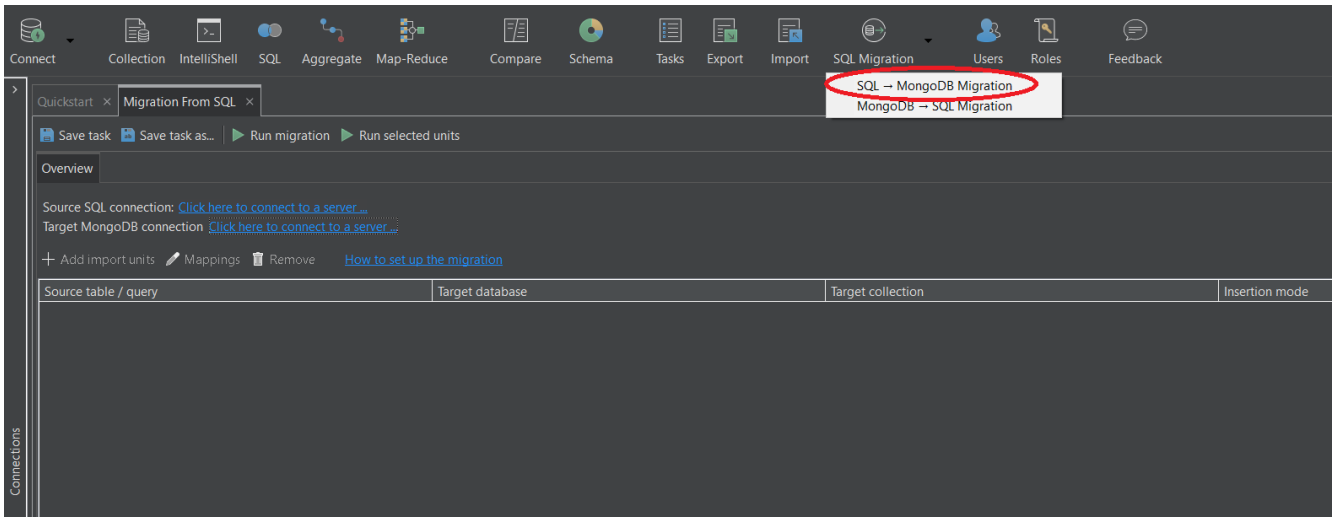
1.2.4 Migration des données à l'aide de 3T Studio

1.2.4.1 Connecter MongoDB Compass Community :

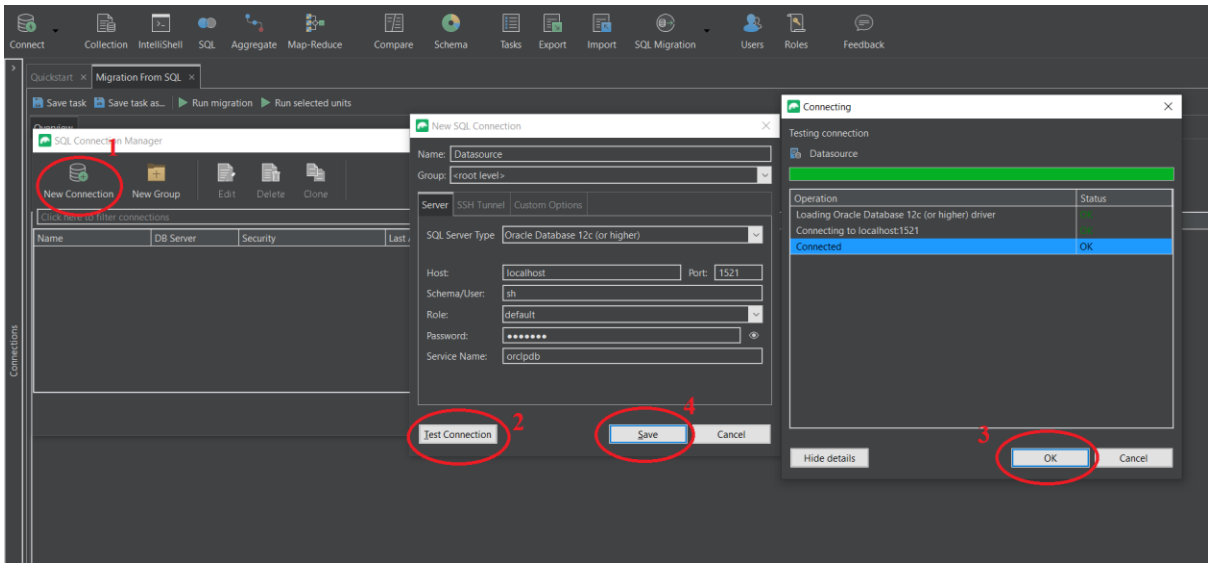


1.2.4.2 SQL to MongoDB Migration :

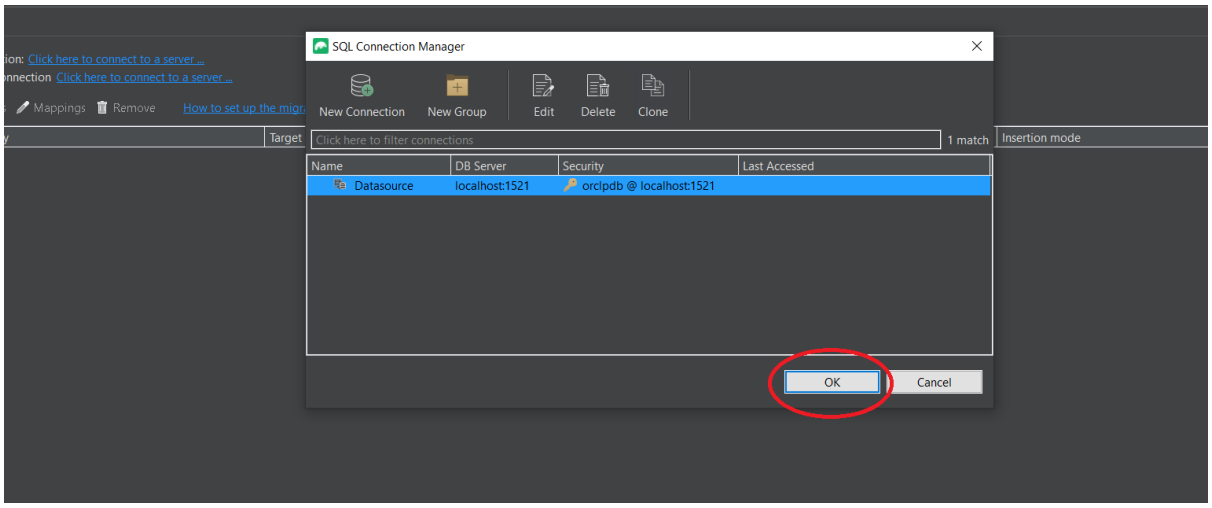
Cliquez sur le bouton Migration SQL dans la barre d'outils, Sélectionnez ensuite SQL → MongoDB Migration. Cela ouvrira une nouvel onglet dans lequel vous pourrez configurer et exécuter l'importation



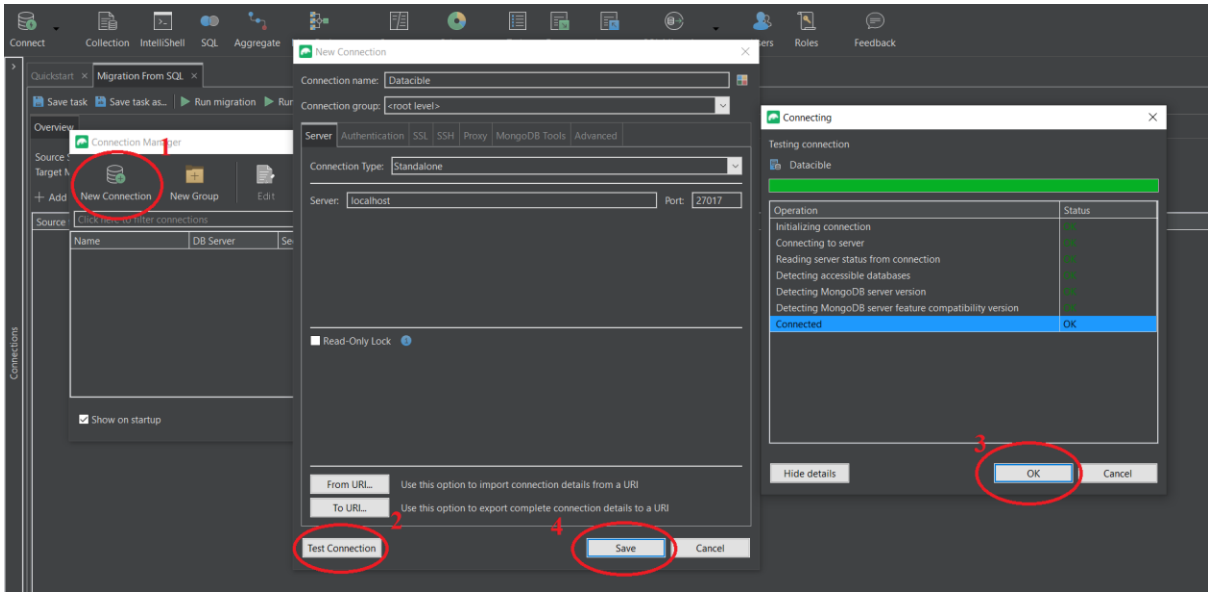
Ensuite la configuration du SQL source connexion, cela ce fait on cliquant sur New connection et en remplissant les données de connexion avec oracle puis on tester la connexion et enregistrer.



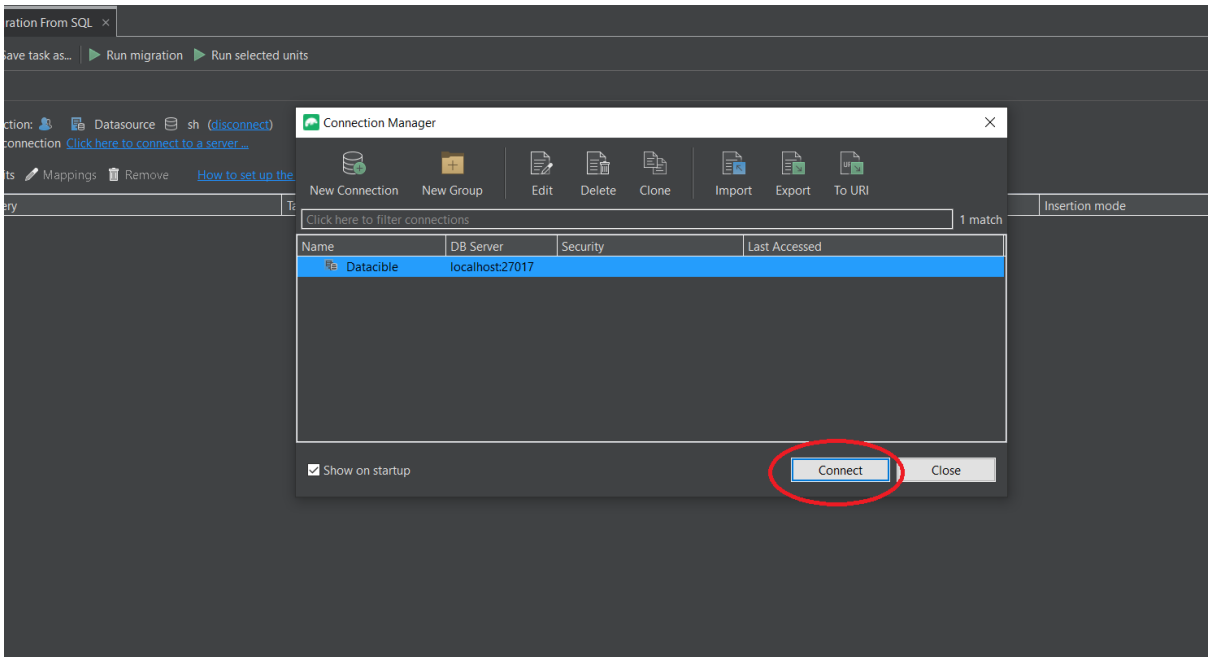
Cliquer sur OK pour connecter avec les données SH d’oracle.



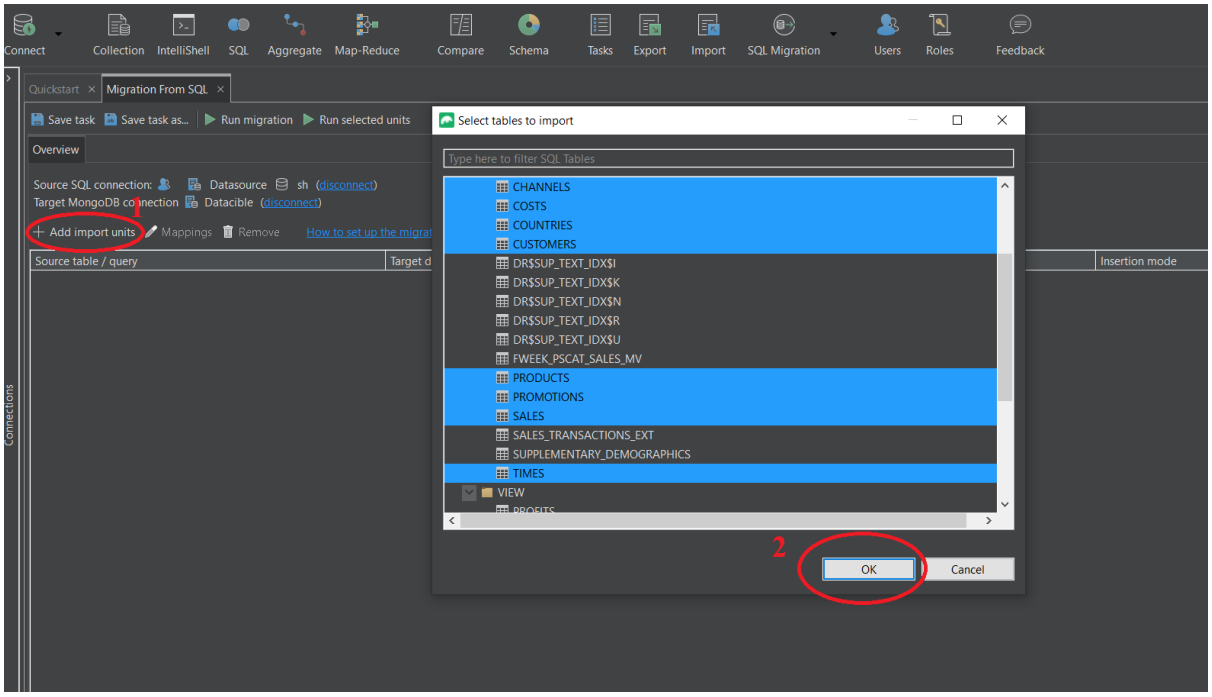
Maintenant de même manière on va se connecter avec MongoDB on cliquant sur New connection et remplir les données de connexion avec MongoDB puis on tester la connexion et enregistrer.



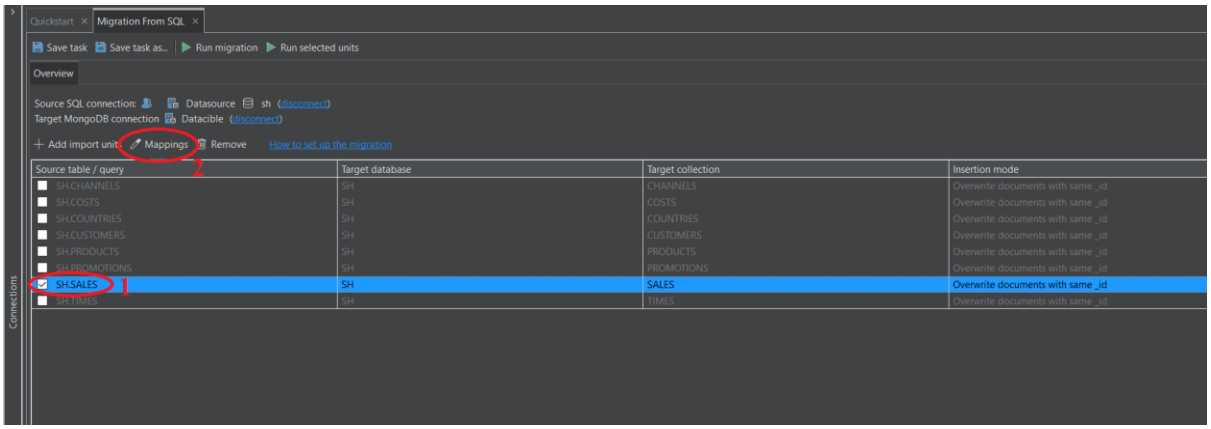
Cliquer sur Connecter pour connecter avec MongoDB .

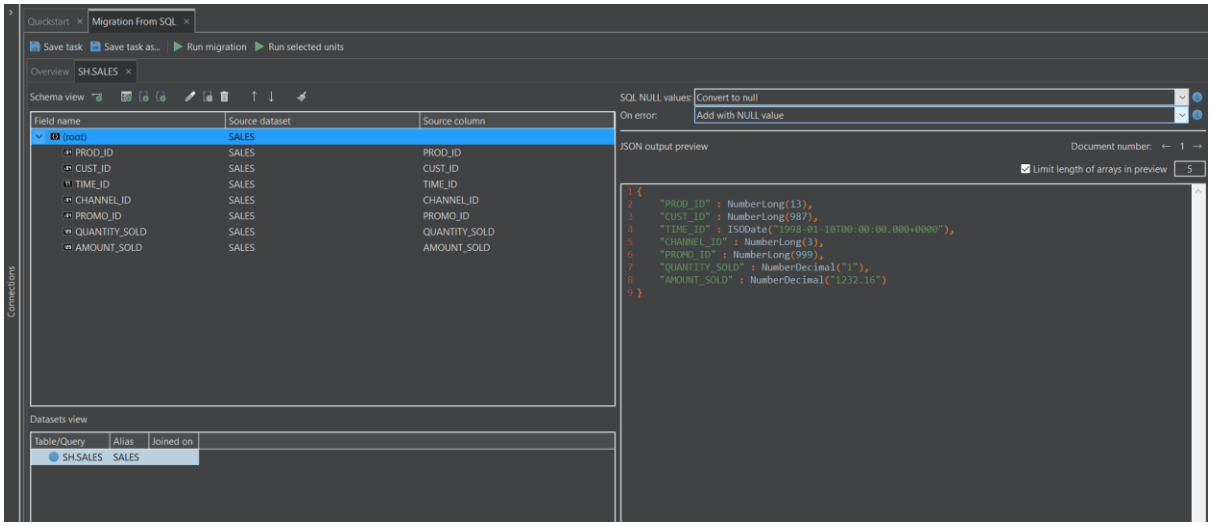


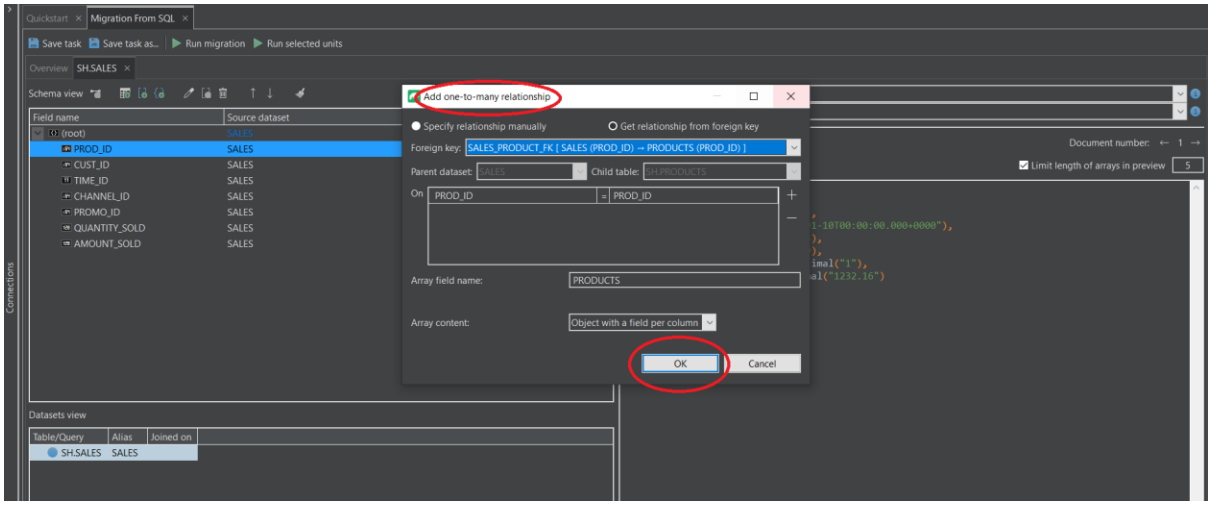
Cliquer sur (+add import units) et sélecter les tables qu'on va utiliser dans l'étude



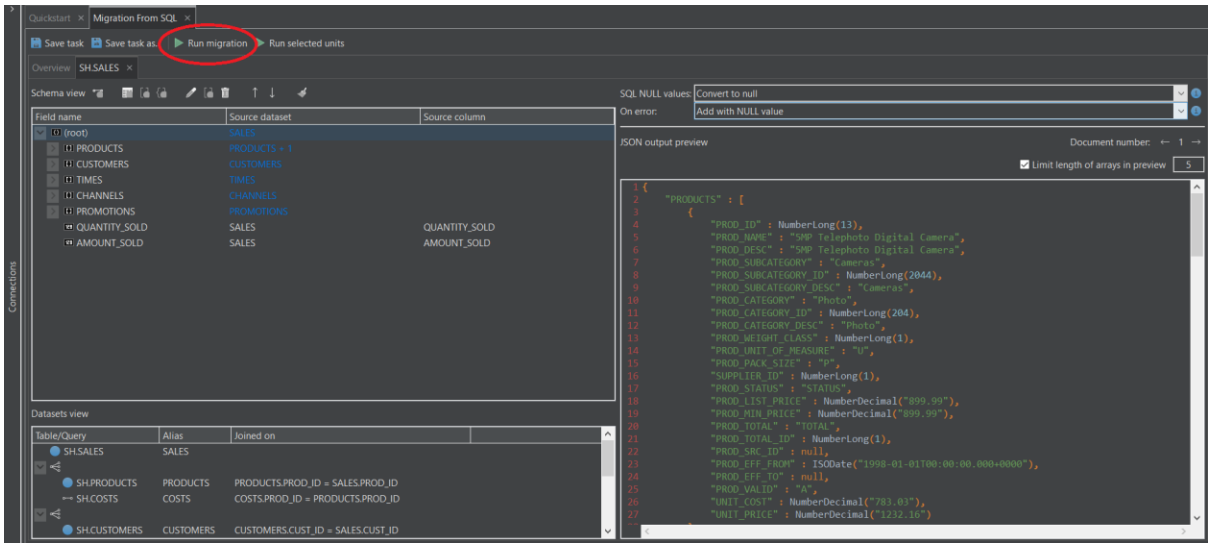
Cocher seulement sur la table sale pour faire des modifications sur la table on cliquant sur Mappings.



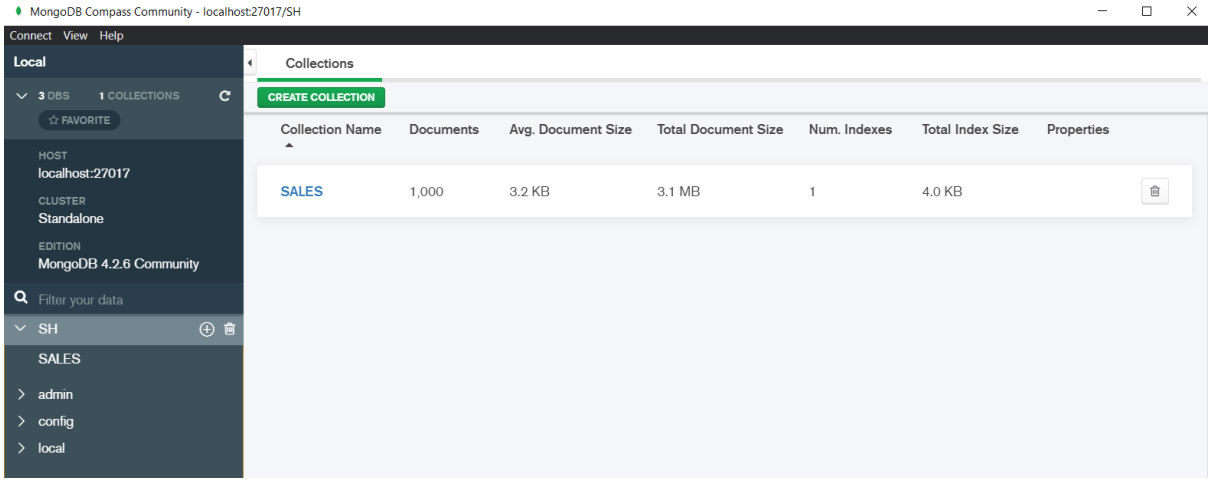




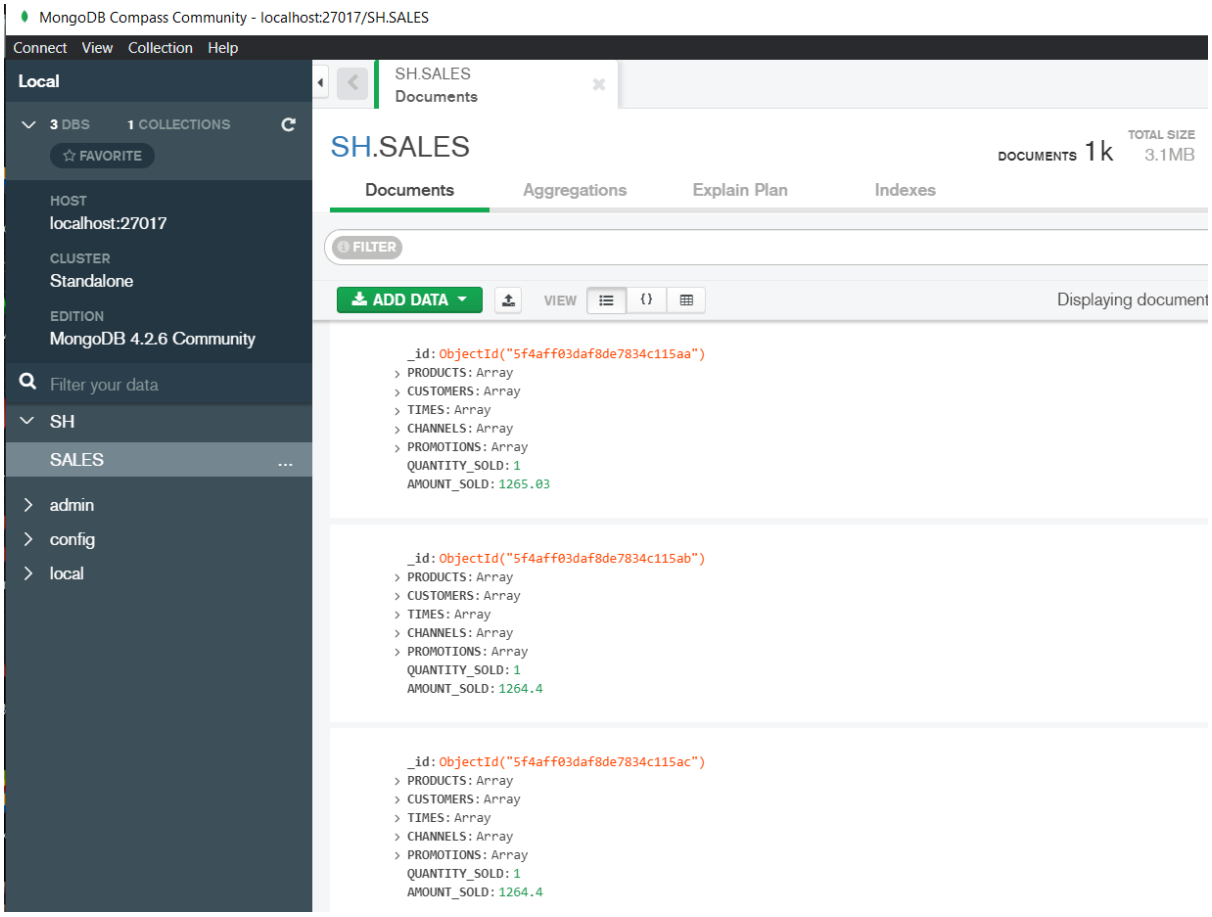
Après les modifications on va cliquer maintenant sur Run migration pour transférer les données de oracle à MongoDB (Les données de la migration est limitées à 1000 données à cause de la version d'essai Studio 3T).



à cette étape il faut actualisé MongoDB compact Community pour consulter le schéma SH qui sera créé après la validation de migration



Cette figure représente l'architectures des données dans un document de schéma SALES :


MongoDB Compass Community - localhost:27017/SH.SALES
Connect View Collection Help
Local
3 DBS 1 COLLECTIONS
FAVORITE
HOST
localhost:27017
CLUSTER
Standalone
EDITION
MongoDB 4.2.6 Community
Filter your data
SH
SALES
admin
config
local
SH.SALES
Documents
SH.SALES
DOCUMENTS 1k
TOTAL SIZE 3.1MB
Documents
Aggregations
Explain Plan
Indexes
FILTER
ADD DATA
VIEW
Displaying document
_id: ObjectId("5f4aff03daf8de7834c115aa")
PRODUCTS: Array
CUSTOMERS: Array
TIMES: Array
CHANNELS: Array
PROMOTIONS: Array
QUANTITY_SOLD: 1
AMOUNT_SOLD: 1265.03
_id: ObjectId("5f4aff03daf8de7834c115ab")
PRODUCTS: Array
CUSTOMERS: Array
TIMES: Array
CHANNELS: Array
PROMOTIONS: Array
QUANTITY_SOLD: 1
AMOUNT_SOLD: 1264.4
_id: ObjectId("5f4aff03daf8de7834c115ac")
PRODUCTS: Array
CUSTOMERS: Array
TIMES: Array
CHANNELS: Array
PROMOTIONS: Array
QUANTITY_SOLD: 1
AMOUNT_SOLD: 1264.4

Requête sur la base de données Cible (SH schéma) :

Cette partie est réalisé à l'aide de Shell de commande. pour connecter a mongo db il faut déplacer à l'emplacement de dossier bin et lancer la commande :

```
>mongod
>use SH
>db
```

```
Command Prompt - MONGO
> use SH
switched to db SH
> db
SH
```

1.1 1er requête :

```
db.SALES.find({ $and: [{"TIMES.CALENDAR_MONTH_NAME":"January"},{"TIMES.CALENDAR_YEAR": 1998}]})
```

```
> db.SALES.find({ $and: [{"TIMES.CALENDAR_MONTH_NAME":"January"},{"TIMES.CALENDAR_YEAR": 1998}]})
{ "_id" : ObjectId("5f4aff00daf8de7834c114ba"), "PRODUCTS" : [ { "PROD_ID" : NumberLong(13), "PROD_NAME" : "5MP Telephot
o Digital Camera", "PROD_DESC" : "5MP Telephoto Digital Camera", "PROD_SUBCATEGORY" : "Cameras", "PROD_SUBCATEGORY_ID" :
 NumberLong(2044), "PROD_SUBCATEGORY_DESC" : "Cameras", "PROD_CATEGORY" : "Photo", "PROD_CATEGORY_ID" : NumberLong(204),
 "PROD_CATEGORY_DESC" : "Photo", "PROD_WEIGHT_CLASS" : NumberLong(1), "PROD_UNIT_OF_MEASURE" : "U", "PROD_PACK_SIZE" : "
P", "SUPPLIER_ID" : NumberLong(1), "PROD_STATUS" : "STATUS", "PROD_LIST_PRICE" : NumberDecimal("899.99"), "PROD_MIN_PRIC
E" : NumberDecimal("899.99"), "PROD_TOTAL" : "TOTAL", "PROD_TOTAL_ID" : NumberLong(1), "PROD_SRC_ID" : null, "PROD_EFF_F
ROM" : ISODate("1998-01-01T00:00:00Z"), "PROD_EFF_TO" : null, "PROD_VALID" : "A", "UNIT_COST" : NumberDecimal("783.03"),
 "UNIT_PRICE" : NumberDecimal("1232.16") } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(987), "CUST_FIRST_NAME" : "York",
 "CUST_LAST_NAME" : "Raimey", "CUST_GENDER" : "M", "CUST_YEAR_OF_BIRTH" : NumberLong(1973), "CUST_MARITAL_STATUS" : "sin
gle", "CUST_STREET_ADDRESS" : "897 Freshened Street", "CUST_POSTAL_CODE" : "46274", "CUST_CITY" : "Adelaide", "CUST_CITY
_ID" : NumberLong(51045), "CUST_STATE_PROVINCE" : "South Australia", "CUST_STATE_PROVINCE_ID" : NumberLong(52741), "COUN
TRIES" : [ { "COUNTRY_ID" : NumberLong(52774), "COUNTRY_ISO_CODE" : "AU", "COUNTRY_NAME" : "Australia", "COUNTRY_SUBREGI
ON" : "Australia", "COUNTRY_SUBREGION_ID" : NumberLong(52794), "COUNTRY_REGION" : "Oceania", "COUNTRY_REGION_ID" : Numbe
rLong(52805), "COUNTRY_TOTAL" : "World total", "COUNTRY_TOTAL_ID" : NumberLong(52806), "COUNTRY_NAME_HIST" : null } ], "
CUST_MAIN_PHONE_NUMBER" : "(341) 841-7590", "CUST_INCOME_LEVEL" : "K: 250,000 - 299,999", "CUST_CREDIT_LIMIT" : NumberLo
ng(11000), "CUST_EMAIL" : "york.raimey@company2.example.com", "CUST_TOTAL" : "Customer total", "CUST_TOTAL_ID" : NumberL
ong(52772), "CUST_SRC_ID" : null, "CUST_EFF_FROM" : ISODate("1998-01-01T00:00:00Z"), "CUST_EFF_TO" : null, "CUST_VALID"
: "A" } ], "TIMES" : [ { "TIME_ID" : ISODate("1998-01-10T00:00:00Z"), "DAY_NAME" : "Saturday", "DAY_NUMBER_IN_WEEK" : Nu
mberLong(6), "DAY_NUMBER_IN_MONTH" : NumberLong(10), "CALENDAR_WEEK_NUMBER" : NumberLong(2), "FISCAL_WEEK_NUMBER" : Numb
```

1.2 2ème requête :

```
db.SALES.find( { "CUSTOMERS.CUST_MARITAL_STATUS": { $exists: true, $ne: null } }, { "CUSTOMERS.CUST_ID": 1, _id: 0 } )
```

```
Command Prompt - MONGO
> db.SALES.find( { "CUSTOMERS.CUST_MARITAL_STATUS": { $exists: true, $ne: null } }, { "CUSTOMERS.CUST_ID": 1, _id: 0 } )

{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(987) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(1660) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(1762) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(1948) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(2273) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(2380) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(2865) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(5203) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(5321) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(6277) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(8540) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(9076) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(12099) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(35834) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(524) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(188) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(531) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(848) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(949) } ] }
{ "CUSTOMERS" : [ { "CUST_ID" : NumberLong(1242) } ] }
Type "it" for more
>
```

1.3 3ème requête :

```
db.SALES.find( {}, { "PRODUCTS.PROD_ID": 1, "PRODUCTS.PROD_NAME": 1,
"PRODUCTS.PROD_SUBCATEGORY": 1, "PRODUCTS.PROD_CATEGORY": 1, _id: 0 }
).sort( { "PRODUCTS.PROD_CATEGORY": -1 })
```

```
Command Prompt - MONGO
> db.SALES.find( {}, { "PRODUCTS.PROD_ID": 1, "PRODUCTS.PROD_NAME": 1, "PRODUCTS.PROD_SUBCATEGORY": 1, "PRODUCTS.PROD_CA
TEGORY": 1, _id: 0 } ).sort( { "PRODUCTS.PROD_CATEGORY": -1 })
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13), "PROD_NAME" : "5MP Telephoto Digital Camera", "PROD_SUBCATEGORY" : "Camer
as", "PROD_CATEGORY" : "Photo" } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13), "PROD_NAME" : "5MP Telephoto Digital Camera", "PROD_SUBCATEGORY" : "Camer
as", "PROD_CATEGORY" : "Photo" } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13), "PROD_NAME" : "5MP Telephoto Digital Camera", "PROD_SUBCATEGORY" : "Camer
as", "PROD_CATEGORY" : "Photo" } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13), "PROD_NAME" : "5MP Telephoto Digital Camera", "PROD_SUBCATEGORY" : "Camer
as", "PROD_CATEGORY" : "Photo" } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13), "PROD_NAME" : "5MP Telephoto Digital Camera", "PROD_SUBCATEGORY" : "Camer
as", "PROD_CATEGORY" : "Photo" } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13), "PROD_NAME" : "5MP Telephoto Digital Camera", "PROD_SUBCATEGORY" : "Camer
as", "PROD_CATEGORY" : "Photo" } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13), "PROD_NAME" : "5MP Telephoto Digital Camera", "PROD_SUBCATEGORY" : "Camer
as", "PROD_CATEGORY" : "Photo" } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13), "PROD_NAME" : "5MP Telephoto Digital Camera", "PROD_SUBCATEGORY" : "Camer
as", "PROD_CATEGORY" : "Photo" } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13), "PROD_NAME" : "5MP Telephoto Digital Camera", "PROD_SUBCATEGORY" : "Camer
as", "PROD_CATEGORY" : "Photo" } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13), "PROD_NAME" : "5MP Telephoto Digital Camera", "PROD_SUBCATEGORY" : "Camer
as", "PROD_CATEGORY" : "Photo" } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13), "PROD_NAME" : "5MP Telephoto Digital Camera", "PROD_SUBCATEGORY" : "Camer
```

1.4 4ème requête :

```
db.SALES.find( { "AMOUNT_SOLD":{$gte:1230}}, { "PRODUCTS.PROD_ID": 1,
"CUSTOMERS.CUST_ID": 1, _id: 0 } )
```

```
Command Prompt - MONGO
> db.SALES.find( { "AMOUNT_SOLD":{$gte:1230}}, { "PRODUCTS.PROD_ID": 1, "CUSTOMERS.CUST_ID": 1, _id: 0 } )
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(987) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(1660) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(1762) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(1843) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(1948) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(2273) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(2380) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(2683) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(2865) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(4663) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(5203) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(5321) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(5590) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(6277) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(6859) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(8540) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(9076) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(12099) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(35834) } ] }
{ "PRODUCTS" : [ { "PROD_ID" : NumberLong(13) } ], "CUSTOMERS" : [ { "CUST_ID" : NumberLong(188) } ] }
Type "it" for more
>
```

1.5 5ème requête :

```
db.SALES.find({ $and: [{"PRODUCTS.PROD_SUBCATEGORY":"Monitors"},
{"CUSTOMERS.COUNTRIES.COUNTRY_REGION":"Americas"}]
},{"CUSTOMERS.CUST_FIRST_NAME":1,"CUSTOMERS.CUST_LAST_NAME":1, _id:
0})
```

```
Command Prompt - MONGO
> db.SALES.find({ $and: [{"PRODUCTS.PROD_SUBCATEGORY":"Monitors"}, {"CUSTOMERS.COUNTRIES.COUNTRY_REGION":"Americas"}] },
{"CUSTOMERS.CUST_FIRST_NAME":1,"CUSTOMERS.CUST_LAST_NAME":1, _id: 0})
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Robin", "CUST_LAST_NAME" : "Howel" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Helga", "CUST_LAST_NAME" : "Nickols" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Robin", "CUST_LAST_NAME" : "Howel" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Meriel", "CUST_LAST_NAME" : "Fairman" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Roderica", "CUST_LAST_NAME" : "Lavin" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Marvel", "CUST_LAST_NAME" : "Bakerman" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Meriel", "CUST_LAST_NAME" : "Fairman" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Roderica", "CUST_LAST_NAME" : "Lavin" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Letty", "CUST_LAST_NAME" : "Parkburg" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Letty", "CUST_LAST_NAME" : "Parkburg" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Marvel", "CUST_LAST_NAME" : "Bakerman" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Sargent", "CUST_LAST_NAME" : "Fairfax" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Heather", "CUST_LAST_NAME" : "Baltimore" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Hill", "CUST_LAST_NAME" : "Everley" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Lise", "CUST_LAST_NAME" : "Hamilton" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Holmes", "CUST_LAST_NAME" : "Glassman" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Blossom", "CUST_LAST_NAME" : "Reilly" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Dixie", "CUST_LAST_NAME" : "Cattlett" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Ethan", "CUST_LAST_NAME" : "Jeffreys" } ] }
{ "CUSTOMERS" : [ { "CUST_FIRST_NAME" : "Candida", "CUST_LAST_NAME" : "Wade" } ] }
Type "it" for more
>
```

1.6 6ème requête :

```
db.SALES.find( { "PRODUCTS.PROD_NAME": /HDTV/ }, {
"PRODUCTS.PROD_NAME": 1, _id: 0 } )
```

```
Command Prompt - mongo
> db.SALES.find( { "PRODUCTS.PROD_NAME": /HDTV/ }, { "PRODUCTS.PROD_NAME": 1, _id: 0 } )
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
{ "PRODUCTS" : [ { "PROD_NAME" : "17\" LCD w/built-in HDTV Tuner" } ] }
Type "it" for more
>
```

1.7 7ème requête :

```
db.SALES.aggregate([ { $match: {
"PRODUCTS.PROD_SUBCATEGORY":"Monitors",
"CUSTOMERS.COUNTRIES.COUNTRY_REGION":"Americas" } }, { $count:
"nombre" } ])
```

```
Command Prompt - mongo
> db.SALES.aggregate([ { $match: { "PRODUCTS.PROD_SUBCATEGORY":"Monitors", "CUSTOMERS.COUNTRIES.COUNTRY_REGION":"America
s" } }, { $count: "nombre" } ])
{ "nombre" : 217 }
>
```

1.8 8ème requête :

```
db.SALES.aggregate([ { $match:
{"PRODUCTS.PROD_SUBCATEGORY":"Monitors",
"CUSTOMERS.COUNTRIES.COUNTRY_REGION":"Americas",
"CUSTOMERS.CUST_MARITAL_STATUS": { $exists: true, $ne: null } } }, {
$project: { "CUSTOMERS.CUST_ID": 1, "CUSTOMERS.CUST_FIRST_NAME": 1,
"PRODUCTS.PROD_ID": 1 } },{ $sort: { "CUSTOMERS.CUST_FIRST_NAME": 1
} } ])
```

Command Prompt - MONGO

```
>   db.SALES.aggregate([ { $match: {"PRODUCTS.PROD_SUBCATEGORY":"Monitors", "CUSTOMERS.COUNTRIES.COUNTRY_REGION":"Americ
as", "CUSTOMERS.CUST_MARITAL_STATUS": { $exists: true, $ne: null } } }, { $project: { "CUSTOMERS.CUST_ID": 1, "CUSTOMERS
.CUST_FIRST_NAME": 1, "PRODUCTS.PROD_ID": 1 } },{ $sort: { "CUSTOMERS.CUST_FIRST_NAME": 1 } } ])
{ "_id" : ObjectId("5f4aff05daf8de7834c1165b"), "PRODUCTS" : [ { "PROD_ID" : NumberLong(14) } ], "CUSTOMERS" : [ { "CUST
_ID" : NumberLong(12123), "CUST_FIRST_NAME" : "Angie" } ] }
{ "_id" : ObjectId("5f4aff04daf8de7834c115cb"), "PRODUCTS" : [ { "PROD_ID" : NumberLong(14) } ], "CUSTOMERS" : [ { "CUST
_ID" : NumberLong(5028), "CUST_FIRST_NAME" : "August" } ] }
{ "_id" : ObjectId("5f4aff04daf8de7834c115cc"), "PRODUCTS" : [ { "PROD_ID" : NumberLong(14) } ], "CUSTOMERS" : [ { "CUST
_ID" : NumberLong(5028), "CUST_FIRST_NAME" : "August" } ] }
{ "_id" : ObjectId("5f4aff06daf8de7834c116b1"), "PRODUCTS" : [ { "PROD_ID" : NumberLong(14) } ], "CUSTOMERS" : [ { "CUST
_ID" : NumberLong(1475), "CUST_FIRST_NAME" : "Austin" } ] }
{ "_id" : ObjectId("5f4aff07daf8de7834c1171b"), "PRODUCTS" : [ { "PROD_ID" : NumberLong(14) } ], "CUSTOMERS" : [ { "CUST
_ID" : NumberLong(1478), "CUST_FIRST_NAME" : "Babetta" } ] }
{ "_id" : ObjectId("5f4aff05daf8de7834c1161e"), "PRODUCTS" : [ { "PROD_ID" : NumberLong(14) } ], "CUSTOMERS" : [ { "CUST
_ID" : NumberLong(5035), "CUST_FIRST_NAME" : "Bailey" } ] }
{ "_id" : ObjectId("5f4aff04daf8de7834c115e9"), "PRODUCTS" : [ { "PROD_ID" : NumberLong(14) } ], "CUSTOMERS" : [ { "CUST
_ID" : NumberLong(3268), "CUST_FIRST_NAME" : "Basil" } ] }
{ "_id" : ObjectId("5f4aff06daf8de7834c1170c"), "PRODUCTS" : [ { "PROD_ID" : NumberLong(14) } ], "CUSTOMERS" : [ { "CUST
_ID" : NumberLong(3268), "CUST_FIRST_NAME" : "Basil" } ] }
{ "_id" : ObjectId("5f4aff06daf8de7834c116ae"), "PRODUCTS" : [ { "PROD_ID" : NumberLong(14) } ], "CUSTOMERS" : [ { "CUST
_ID" : NumberLong(7715), "CUST_FIRST_NAME" : "Baylen" } ] }
{ "_id" : ObjectId("5f4aff06daf8de7834c116b0"), "PRODUCTS" : [ { "PROD_ID" : NumberLong(14) } ], "CUSTOMERS" : [ { "CUST
_ID" : NumberLong(7715), "CUST_FIRST_NAME" : "Baylen" } ] }
{ "_id" : ObjectId("5f4aff06daf8de7834c116eb"), "PRODUCTS" : [ { "PROD_ID" : NumberLong(14) } ], "CUSTOMERS" : [ { "CUST
_ID" : NumberLong(1518), "CUST_FIRST_NAME" : "Biddy" } ] }
{ "_id" : ObjectId("5f4aff06daf8de7834c116ef"), "PRODUCTS" : [ { "PROD_ID" : NumberLong(14) } ], "CUSTOMERS" : [ { "CUST
_ID" : NumberLong(1518), "CUST_FIRST_NAME" : "Biddy" } ] }
```

Comparaison et Synthèse :

le SGBD parfait c'est celui-ci qui résoudre ton problème et qui marche bien avec ton besoin, ça sera un peu difficile de valoriser un des SGBD par rapport à l'autre, en fait dans ce paragraphe en va essayer de faire des comparaison selon des fonctionnalité et caractéristique des deux SGBD .

1.1 Terminologie et concepts

De nombreux concepts dans Oracle ont des analogues proches dans MongoDB. Le tableau ci-dessous présente les concepts communs d'Oracle et de MongoDB.

Oracle	MongoDB
ACID Transactions	ACID Transactions
Table	Collection
Row	Document
Column	Field
Secondary Index	Secondary Index
JOINs	Embedded documents, $lookup & $graphLookup

1.2 Comparaison des fonctionnalités

Comme Oracle, MongoDB offre un riche ensemble de fonctionnalités et de fonctionnalités bien au-delà de celles offertes par les simples magasins de données NoSQL. MongoDB possède un langage de requête riche, des index secondaires hautement fonctionnels (y compris la recherche de texte et la géospatiale), un puissant cadre d'agrégation pour l'analyse des données, la recherche à facettes, le traitement de graphiques et plus encore. Avec MongoDB, vous pouvez également utiliser ces fonctionnalités sur des types de données plus diversifiés qu'une base de données relationnelle, et vous pouvez le faire à grande échelle.

1.3 Langage de requête

Oracle et MongoDB ont un langage de requête riche. Vous trouverez ci-dessous quelques exemples d'instructions SQL et leur correspondance avec MongoDB.

Oracle	MongoDB
INSERT INTO users (user_id, age, status) VALUES ('bcd001', 45, 'A')	db.users.insert({ user_id: 'bcd001', age: 45, status: 'A' })
SELECT * FROM users	db.users.find()
UPDATE users SET status = 'C' WHERE age > 25	db.users.update({ age: { $gt: 25 } }, { $set: { status: 'C' } }, { multi: true })
db.start_transaction() cursor.execute(orderInsert, orderData)	s.start*transaction() orders.insert*one(order, session=s) stock.update*one(item,*

cursor.execute(stockUpdate, stockData) db.commit()	*stockUpdate, session=s)* *s.commit*transaction()

1.4 Au niveau de flexibilité :

Dans Oracle, vous pré définissez le schéma de votre base de données en fonction de vos besoins et configurez des règles pour réagir les relations entre les champs de vos tables. Les informations associées peuvent être stockées dans des tables séparées, mais associées via l'utilisation de clés étrangères et de jointures.

Toute modification du schéma nécessite une procédure de migration qui peut mettre la base de données hors ligne ou réduire considérablement les performances de l'application. à l'opposé de MongoDB, Les champs peuvent varier d'un document à l'autre; il n'est pas nécessaire de déclarer la structure des documents au système - les documents sont autodescriptifs.

Si un nouveau champ doit être ajouté à un document, le champ peut être créé sans affecter tous les autres documents de la collection, sans mettre à jour un catalogue système central et sans mettre le système hors ligne. En option, la validation de schéma peut être utilisée pour appliquer des contrôles de gouvernance des données sur chaque collection.

Conclusion Générale

Actuellement, le NoSQL est une technologie qui émerge en puissance, il est mis œuvre dans des environnements manipulant de grandes masses de données tels que Google, Yahoo, Twitter, Facebook, etc. Les moteurs de recherche sont les premiers utilisateurs de ces technologies puisqu'ils ont besoin d'une grande puissance de stockage et de traitement de ces volumes de données, de même pour les réseaux sociaux qui gère une très grosse montée en charge dû au grand nombre d'utilisateurs et de requêtes simultanées.

Le NoSQL a efficacement contribué à résoudre la problématique d'élasticité et de flexibilité des bases de données dans des contextes distribués. Les systèmes de bases de données relationnelles répondant au besoin transactionnel grâce aux propriétés ACID (Atomicity, Consistency, Isolation, et Durability) sont en train d'atteindre leurs limites pour la gestion de grandes masses d'informations estimées en Tera et Peta Octets, en conséquence le passage vers les systèmes distribuées NoSQL s'impose comme la meilleure alternative pour pallier aux rudes contraintes liées à la performance.

L'objectif de ce projet étant la migration d'une base de données relationnelle ORACLE vers une base de données NoSQL orientée document MongoDB. L'exportation des données relationnelles vers un format reconnu par les deux systèmes était la clé de réussite de notre solution. L'importation des données à partir des fichiers JSON vers MongoDB était la deuxième phase du processus de migration. En fin, on peut estimer que l'objectif tracé au préalable a été atteint, c'était vraiment une expérience plein d'information et de données.