NF26 - Groupe 4

Mise en place d'une solution décisionnelle

SAVARY Tobias, SZENDROVICS Sacha, SAIDI Nassim, CREUZE Martin, LABOURÉ Alexandre

Sommaire

Introduction

Objectifs

Exploitation de données fictives d'un établissement hospitalier avec mise en place d’une solution décisionnelle pour :

- Prise de Décision Informée
- Automatisation du Reporting
- Fiabilité et Cohérence des Données
- Facilité d’Interprétation avec Tableaux de Bord Intuitifs

Secteur de la Santé

- Importance croissante de l'analyse de données pour améliorer les soins et l'efficacité des services hospitaliers.
- Besoin d'intégrer et de centraliser les données de différentes sources (patients, traitements, ressources).

Technologies Utilisées

- **Teradata :** SGBD robuste et scalable pour gérer de grands volumes de données.
- **Power BI** : Outil de reporting puissant pour créer des tableaux de bord interactifs et visuels.

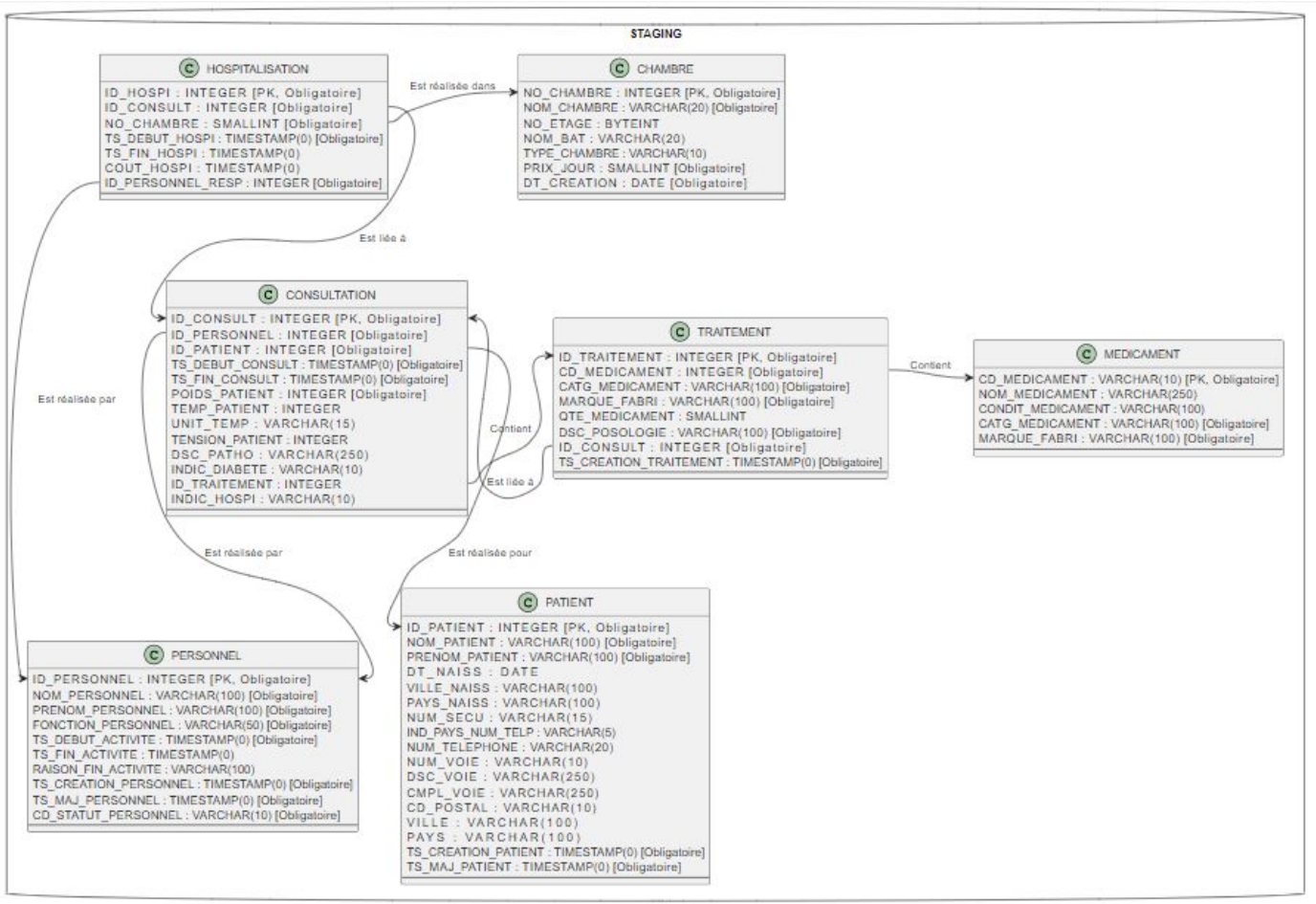
Lot 1 - Installation de l’environnement de travail et conception de la solution

Découverte des données (fichiers excels) et mise en place de l’environnement technique

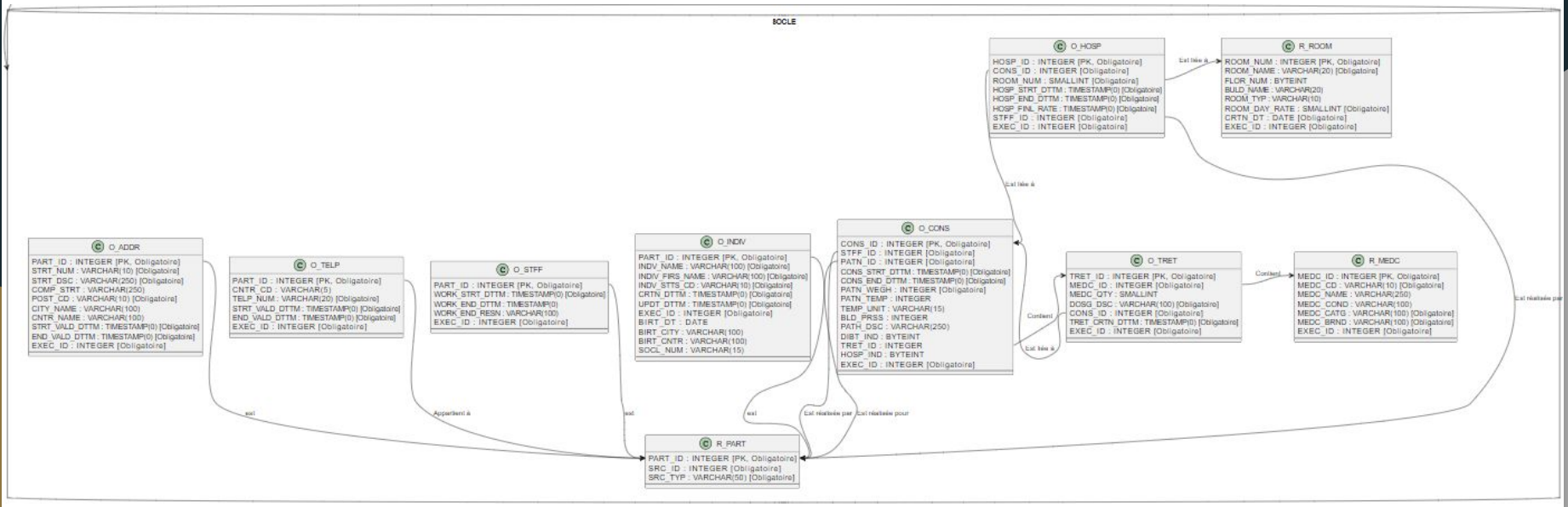
- Teradata en tant que SGBD (installation de la VM)
- Gitlab pour le versionning
- Teams pour la communication

Conception des différentes bases et tables (voir schémas UML ci-après)

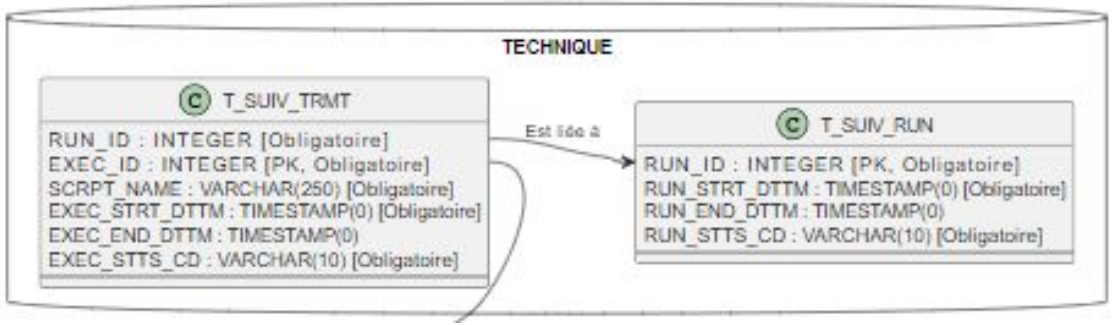
Base de Staging



Base de Work / Socle



Base Technique



Lot 2.1 - Installation du SID

- Scripts de création des bases de données et tables
- Script ‘install_SID.sh’
- Utilisation de BTEQ pour l’exécution du SQL

create_db.sql 878 B

```
SELECT * FROM dbc.Databases WHERE DatabaseName = 'STG';
.IF ACTIVITYCOUNT <> 0 THEN .GOTO SKIP_STG;
CREATE DATABASE STG from DBC as perm=1000000000;
GRANT ALL PRIVILEGES ON STG TO DBC WITH GRANT OPTION;
.LABEL SKIP_STG
```

create_table_soc.sql 4.69 KiB

```
-- create_tables_SOC.sql
-- Specifity: Only create if not already exists
-- Table R_PART
SELECT * FROM DBC.TablesV WHERE TableName = 'R_PART' AND DatabaseName = 'SOC';
.IF ACTIVITYCOUNT <> 0 THEN .GOTO OK_R_PART;

CREATE TABLE SOC.R_PART (
    PART_ID INTEGER NOT NULL,
    SRC_ID INTEGER NOT NULL,
    SRC_TYP VARCHAR(50) NOT NULL)
    UNIQUE PRIMARY INDEX (PART_ID);

.LABEL OK_R_PART
```

install_SID.sh 1.50 KiB

```
#!/bin/bash

LOGFILE="install_SID.log"
HOST="localhost"
DB_USER="dbc"
DB_PASS="dbc"

# Fonction pour ajouter un message avec horodatage au fichier de log
log() {
    echo "$(date '+%Y-%m-%d %H:%M:%S') - $1" >> $LOGFILE
}

# Fonction pour exécuter un script SQL via BTEQ
run_sql_script() {
    local script=$1
    bteq <<EOF >> $LOGFILE
.LOGON $HOST/$DB_USER,$DB_PASS;
.RUN FILE=$script;
.LOGOFF;
.QUIT;
EOF
}

log "Début de l'installation du SID"

# Création des bases de données
log "Création des bases de données..."
run_sql_script "./create_db.sql"
```

launch_load_SID.sh

Lot 2.2 - Ingestion des données

- Scripts TPT pour l'alimentation des tables STG
- Script 'LAUNCH_LOAD_SID.sh'

load_chambre.tpt

```
STEP Setup_Tables
(
    APPLY
    ('DELETE FROM STG.CHAMBRE;')
    TO OPERATOR (od_Chambre);
);

STEP Load_Chambre
(
    APPLY
    (
        'INSERT INTO STG.CHAMBRE (NO_CHAMBRE, NOM_CHAMBRE, NO_ETAGE, NOM_BAT, TYPE_CHAMBRE, PRIX_JOUR,
         VALUES (CAST(:NO_CHAMBRE AS INTEGER), :NOM_CHAMBRE, CAST(:NO_ETAGE AS BYTEINT), :NOM_BAT, :TY
    )
    TO OPERATOR (ol_Chambre)
    SELECT * FROM OPERATOR (op_Chambre);
);
```

```
find "$BASE_DIR" -type d -name 'BDD_HOSPITAL*' | while read -r dir; do
  if [ "$counter" -ge "$MAX_FOLDERS" ]; then
    # Maximum number of folders reached
    break
  fi

  log "Processing directory: $dir"

  if [ -f "$CHECKPOINT_FILE" ]; then
    rm -f "$CHECKPOINT_FILE"
    log "Removed checkpoint file: $CHECKPOINT_FILE"
  else
    log "Checkpoint file not found: $CHECKPOINT_FILE"
  fi

  # TXT -> STG : TPT
  run_sql_script "./create_exec_run_id.sql"
  ./run_tpt.sh "$dir"

  if [ $? -ne 0 ]; then
    log "run_tpt.sh failed for directory: $dir"
    run_sql_script "./update_exec_id_ko.sql"
    continue
  else
    run_sql_script "./update_exec_id_ok.sql"
  fi

  # STG -> WRK : SQL
  run_sql_script "./insert_to_wrk_from_stg.sql"
  # WRK -> SOC : SQL
  run_sql_script "./insert_to_soc_from_wrk.sql"

  if [ $? -ne 0 ]; then
    log "run_tpt.sh failed for directory: $dir"
    run_sql_script "./update_exec_id_ko.sql"
    continue
  else
    run_sql_script "./update_exec_id_ok.sql"
  fi

  # Check if any errors occurred during the process else update the RUN_ID
  if [ $? -ne 0 ]; then
    log "Error occurred during the process for directory: $dir"
    run_sql_script "./update_run_id_ko.sql"
    continue
  else
    run_sql_script "./update_run_id_ok.sql"
  fi
```

Lot 2.2 - Ingestion des données

run_tpt.sh

```
# Define file paths dynamically
CONSULTATION_FILE=$(find "$DIR" -type f -name 'CONSULTATION*.txt' | sort | tail -n 1)
HOSPITALISATION_FILE=$(find "$DIR" -type f -name 'HOSPITALISATION*.txt' | sort | tail -n 1)
TRAITEMENT_FILE=$(find "$DIR" -type f -name 'TRAITEMENT*.txt' | sort | tail -n 1)
CHAMBRE_FILE=$(find "$DIR" -type f -name 'CHAMBRE*.txt' | sort | tail -n 1)
MEDICAMENT_FILE=$(find "$DIR" -type f -name 'MEDICAMENT*.txt' | sort | tail -n 1)
PATIENT_FILE=$(find "$DIR" -type f -name 'PATIENT*.txt' | sort | tail -n 1)
PERSONNEL_FILE=$(find "$DIR" -type f -name 'PERSONNEL*.txt' | sort | tail -n 1)

# Check if all files are found
if [ -z "$CONSULTATION_FILE" ] || [ -z "$HOSPITALISATION_FILE" ] || [ -z "$TRAITEMENT_FILE" ] ||
  echo "Error: One or more files not found in $DIR."
  exit 1
fi

# Load chambre
tbuild -f load_chambre.tpt -u "FileName='$CHAMBRE_FILE'" | tee -a $RUN_TPT_LOGFILE
```

Lot 3 - Alimentation du Datawarehouse

- Alimentation des tables WRK (CORRECT)
- Bascule dans les tables SOC (UPDATE)
- Fonction de lancement des scripts .sql pour le suivi des traitements
- Script ‘LAUNCH_LOAD_SID.sh’ enrichi

insert_to_wrk_from_stg.sql

```
59  -- Delete and insert into WRK.R_MEDC
60  DELETE FROM WRK.R_MEDC;
61
62  INSERT INTO WRK.R_MEDC (MEDC_ID, MEDC_CD, MEDC_NAME, MEDC_COND, MEDC_CATG, MEDC_BRND, EXEC_ID)
63  SELECT
64      s.MEDC_ID,
65      s.MEDC_CD,
66      s.MEDC_NAME,
67      s.MEDC_COND,
68      s.MEDC_CATG,
69      s.MEDC_BRND,
70      s.EXEC_ID
71  FROM (
72      SELECT
73          (SELECT med_id FROM ID_TABLE) + ROW_NUMBER() OVER (ORDER BY CD_MEDICAMENT, CATG_MEDICAMENT, MARQUE_FABRI) AS MEDC_ID,
74          CD_MEDICAMENT AS MEDC_CD,
75          NOM_MEDICAMENT AS MEDC_NAME,
76          CONDIT_MEDICAMENT AS MEDC_COND,
77          CATG_MEDICAMENT AS MEDC_CATG,
78          MARQUE_FABRI AS MEDC_BRND,
79          (SELECT execid FROM VarTable) AS EXEC_ID
80      FROM
81          STG.MEDICAMENT
82  ) AS s;
```

Lot 3 - Alimentation du Datawarehouse

launch_load_SID.sh

```
find "$BASE_DIR" -type d -name 'BDD_HOSPITAL*' | while read -r dir; do
  if [ "$counter" -ge "$MAX_FOLDERS" ]; then
    # Maximum number of folders reached
    break
  fi

  log "Processing directory: $dir"

  if [ -f "$CHECKPOINT_FILE" ]; then
    rm -f "$CHECKPOINT_FILE"
    log "Removed checkpoint file: $CHECKPOINT_FILE"
  else
    log "Checkpoint file not found: $CHECKPOINT_FILE"
  fi

  # TXT -> STG : TPT
  run_sql_script "./create_exec_run_id.sql"
  ./run_tpt.sh "$dir"

  if [ $? -ne 0 ]; then
    log "run_tpt.sh failed for directory: $dir"
    run_sql_script "./update_exec_id_ko.sql"
    continue
  else
    run_sql_script "./update_exec_id_ok.sql"
  fi

  # STG -> WRK : SQL
  run_sql_script "./insert_to_wrk_from_stg.sql"
  # WRK -> SOC : SQL
  run_sql_script "./insert_to_soc_from_wrk.sql"

  if [ $? -ne 0 ]; then
    log "run_tpt.sh failed for directory: $dir"
    run_sql_script "./update_exec_id_ko.sql"
    continue
  else
    run_sql_script "./update_exec_id_ok.sql"
  fi

  # Check if any errors occurred during the process else update the RUN_ID
  if [ $? -ne 0 ]; then
    log "Error occurred during the process for directory: $dir"
    run_sql_script "./update_run_id_ko.sql"
    continue
  else
    run_sql_script "./update_run_id_ok.sql"
  fi
```

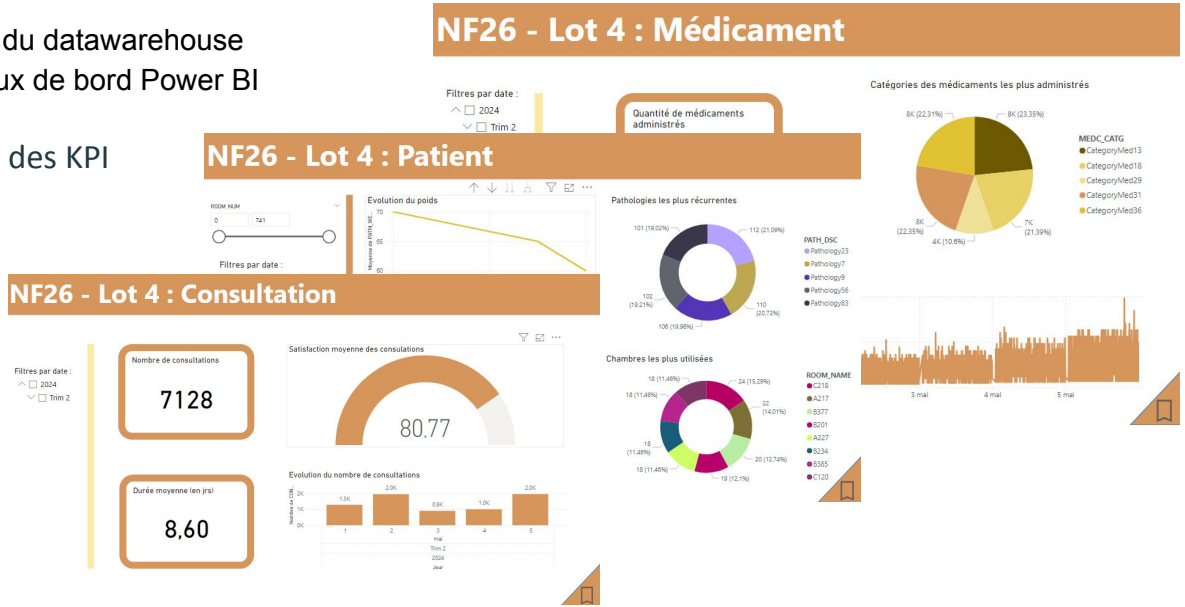
insert_to_soc_from_work.sql

```
MERGE INTO SOC.R_MEDC
USING (
    SELECT      Nassim SAIDI, il y a 5 jours • c est good lot3
        MEDC_ID,
        MEDC_CD,
        MEDC_NAME,
        MEDC_COND,
        MEDC_CATG,
        MEDC_BRND,
        (SELECT execid FROM VarTable) AS EXEC_ID
    FROM
        WRK.R_MEDC
) AS s(MEDC_ID, MEDC_CD, MEDC_NAME, MEDC_COND, MEDC_CATG, MEDC_BRND, EXEC_ID)
ON SOC.R_MEDC.MEDC_ID = s.MEDC_ID
WHEN MATCHED THEN UPDATE
    SET MEDC_CD = s.MEDC_CD,
        MEDC_NAME = s.MEDC_NAME,
        MEDC_COND = s.MEDC_COND,
        MEDC_CATG = s.MEDC_CATG,
        MEDC_BRND = s.MEDC_BRND,
        EXEC_ID = s.EXEC_ID
WHEN NOT MATCHED THEN INSERT (MEDC_ID, MEDC_CD, MEDC_NAME, MEDC_COND, MEDC_CATG, MEDC_BRND, EXEC_ID)
    VALUES (s.MEDC_ID, s.MEDC_CD, s.MEDC_NAME, s.MEDC_COND, s.MEDC_CATG, s.MEDC_BRND, s.EXEC_ID);
```

Lot 4 - Calcul des KPI et Développement des tableaux de bord Power BI

- Vues basées sur les tables du datawarehouse
- Développement des tableaux de bord Power BI

Place à une démo et description des KPI



Apprentissages

Techniques

- Compréhension du fonctionnement d'un ETL
- Gestion de cycle de vie des données (de la récupération à la visualisation en passant par le traitement)
- Développement SQL avec Teradata
- Développement Bash pour les scripts
- Utilisation de Power BI pour le reporting

Humains

- Travail en équipe importante
- Travail sur des temps restreints avec rendus hebdomadaires

Difficultés rencontrées

- Compréhension initiale du sujet
  - A quoi correspondent les tables
  - A quoi correspondent les scripts
- Contraintes de clés étrangères
- Documentation Teradata
- Types de variables (Integer dans Excels => BigInt dans .txt)
- “##” dans les DateTime

Axes d'améliorations

Techniques :

- Améliorer la documentation technique
- Optimisation des scripts pour la performance

Organisationnels :

- Optimisation de la répartition des tâches

Conclusion

Etapes importantes

- **Lot 1**: Mise en place de l'environnement de travail et conception du modèle de données.
- **Lot 2**: Installation du SID et ingestion des données pour initialiser le datawarehouse (.txt => STG).
- **Lot 3**: Alimentation et traitement des données dans le datawarehouse (STG => WRK => SOC).
- **Lot 4**: Calcul des KPI et développement de tableaux de bord avec Power BI.

Bénéfices

- Amélioration de la prise de décision grâce à des données fiables et en temps réel.
- Automatisation des processus de reporting, augmentant l'efficacité opérationnelle.
- Développement de compétences techniques et organisationnelles.

Axes d'Amélioration

- Optimisation des scripts et processus pour améliorer la performance.