

# Défi Python codoc

## Exercice 1 :

L'hôpital Truc nous demande d'alimenter Dr Warehouse.

La première étape est de récupérer l'historique du fichier administratif des patients.

Le chargement au fil de l'eau sera fait ultérieurement (non prévu dans cet exercice).

**Mission : écrire un script python qui charge le fichier export\_patient.xlsx dans les tables DWH\_PATIENT et DWH\_PATIENT\_IPPHIST**

Le GitHub de Dr Warehouse explique le contenu des tables

<https://github.com/imagine-bdd/DRWH/wiki/Tables-description-for-ETL>

*Pour info, l'id du patient dans l'hôpital s'appelle généralement IPP (en France). Il s'appelle HOSPITAL\_PATIENT\_ID dans dr Warehouse. Un patient a généralement un seul IPP, mais il se peut qu'il en ait plusieurs suite à des changements de système, ou suite à des erreurs administrative lors de son arrivée à l'hôpital (changement de nom etc.).*

Les tables ont été créées dans la BDD sqlite "drwh.db" comme suit :

doc sqlite / python : [https://www.python-course.eu/sql\\_python.php](https://www.python-course.eu/sql_python.php)

```
CREATE TABLE DWH_PATIENT
(
  PATIENT_NUM      INTEGER,
  LASTNAME         VARCHAR2(100),
  FIRSTNAME        VARCHAR2(40),
  BIRTH_DATE       DATE,
  SEX              VARCHAR2(2),
  MAIDEN_NAME      VARCHAR2(81),
  RESIDENCE_ADDRESS VARCHAR2(1000),
  PHONE_NUMBER     VARCHAR2(1000),
  ZIP_CODE         VARCHAR2(30),
  RESIDENCE_CITY   VARCHAR2(200),
  DEATH_DATE       DATE,
  RESIDENCE_COUNTRY VARCHAR2(100),
  RESIDENCE_LATITUDE VARCHAR2(300),
  RESIDENCE_LONGITUDE VARCHAR2(300),
  DEATH_CODE       VARCHAR2(2),
  UPDATE_DATE      DATE,
  BIRTH_COUNTRY    VARCHAR2(100),
```

```

    BIRTH_CITY      VARCHAR2(100),
    BIRTH_ZIP_CODE   VARCHAR2(10),
    BIRTH_LATITUDE   FLOAT(126),
    BIRTH_LONGITUDE  FLOAT(126),
    UPLOAD_ID        INTEGER,
    PRIMARY KEY (PATIENT_NUM)
);

CREATE TABLE DWH_PATIENT_IPPHIST
(
    PATIENT_NUM      INTEGER,
    HOSPITAL_PATIENT_ID VARCHAR2(100),
    ORIGIN_PATIENT_ID VARCHAR2(40),
    MASTER_PATIENT_ID INTEGER,
    UPLOAD_ID        INTEGER
);

```

## Exercice 2 :

Il s'agit maintenant de charger les comptes rendus des patients dans la table DWH\_DOCUMENT.

Une liste de Comptes rendus nous a été transmise en nous précisant que le nom des fichiers sont structurés de cette manière :

IPP\_IDDOCUMENT.pdf  
IPP\_IDDOCUMENT.docx

IPP : Id du patient dans l'hôpital

IDDOCUMENT : Id du document dans le logiciel source.

Nous n'avons pas l'unité hospitalière qui a produit le document.

Les fichiers PDF proviennent de la source : DOSSIER\_PATIENT

les fichiers DOCX proviennent de la source : RADIOLOGIE\_SOFTWARE

Les fichiers doivent être convertis au format ascii et chargés dans la colonne DISPLAYED\_TEXT de la table DWH\_DOCUMENT

**Mission : Écrire un script python qui charge les documents dans la table DWH\_DOCUMENT en lien avec la table DWH\_PATIENT.**

**Mission bonus : Récupérer le plus souvent possible les dates de comptes rendus et l'auteur à**

## **l'intérieur du document.**

La table a été créée dans la BDD sqlite ([https://www.python-course.eu/sql\\_python.php](https://www.python-course.eu/sql_python.php)) *drwh.db* comme suit :

```
CREATE TABLE DWH_DOCUMENT
(
  DOCUMENT_NUM      INTEGER      NOT NULL,
  PATIENT_NUM       INTEGER,
  ENCOUNTER_NUM     VARCHAR2(30),
  TITLE             VARCHAR2(400),
  DOCUMENT_ORIGIN_CODE VARCHAR2(40),
  DOCUMENT_DATE     DATE,
  ID_DOC_SOURCE     VARCHAR2(300),
  DOCUMENT_TYPE     VARCHAR2(40),
  DISPLAYED_TEXT    CLOB,
  AUTHOR            VARCHAR2(200),
  UNIT_CODE         VARCHAR2(30),
  UNIT_NUM          INTEGER,
  DEPARTMENT_NUM    INTEGER,
  EXTRACTCONTEXT_DONE_FLAG INTEGER,
  EXTRACTCONCEPT_DONE_FLAG INTEGER,
  ENRGENE_DONE_FLAG  INTEGER,
  ENRICHTEXT_DONE_FLAG INTEGER,
  UPDATE_DATE       DATE,
  UPLOAD_ID         INTEGER,
  PRIMARY KEY (DOCUMENT_NUM)
);
```