UPEC
UNIVERSITÉ PARIS-EST CRÉTEIL

iut
CRÉTEIL-VITRY

DÉPARTEMENT INFORMATIQUE

SAÉ 1.04

Création d'une base de données – Compte rendu

BERRICHE Djibril
CHERIFI Ismaël
DHALI Afnanmasud
EDDAHCHOURI Lokman

I. INTRODUCTION

Dans le cadre de cette SAÉ 1.04, Création d'une base de données, notre objectif dans ce projet était de concevoir une base de données à partir d'un fichier CSV brut, contenant des données démographiques mondiales de L'ONU, issues du Département des Affaires Économiques et Sociales (Division Population).

Pour ce qui est des outils et logiciels utilisés, nous avons utilisé le logiciel DBeaver, un logiciel permettant l'administration et le requêtage de base de données avec une interface graphique simple d'utilisation, le langage SQL permettant de traiter et de stocker des informations dans une base de données et SQLite comme moteur de base de données.
Le projet utilise SQLite car c'est un SGBD relationnel très rapide, simple à manipuler et ne nécessitant aucune installation de serveur. Toute la base est contenue dans un seul fichier, facilitant ainsi le partage et l'utilisation commune.

La base de données a été conçue avec un modèle hiérarchique tel que : Continent → Région → Sous – région →Pays. Des scripts ont été écrits afin de peupler, de nettoyer et de structurer la base de données puis à l'aide des requêtes codées, cela nous permet de tester et de valider l'intégralité des données importées mais aussi notre base.

II. PROBLÉMATIQUE ET DONNÉES TRAITÉES

L'objectif principal de ce projet était de transformer un fichier CSV brut contenant des données démographiques mondiales (ONU, 1950–2023) en une base de données relationnelle cohérente, nous permettant par la suite d'effectuer des analyses sur un ou plusieurs critères spécifiques, faire des comparaisons démographiques ou géographiques avec différents pays/région sur une période donnée ou sur une année et de rendre ces données exploitables pour des applications d'aide à la décision.

Pour réaliser cela, une bonne structuration et hiérarchie de la base de données est donc nécessaire afin de faciliter la lecture, l'exploitation et l'analyse de ces données.

C'est pour ceci que nous avons modélisé de façon hiérarchique la base de données sur 4 échelles géographiques telles que :

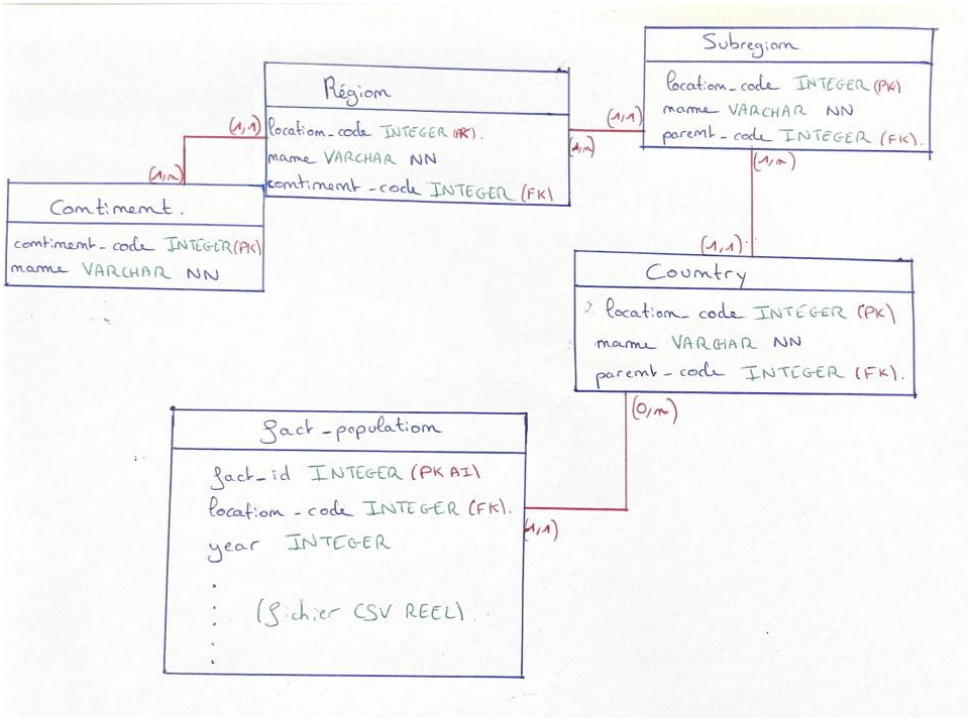
Continents → Région → Sous- région → Pays

Ces 4 échelles géographiques sont stockées dans une table appelée la « table de dimensions ». Ces données possèdent une « table de faits » qui relève les mesures démographiques sur la période donnée. Cela nous permet ainsi une compréhension et une lecture plus simplifiée des données, nous permettant donc l'écriture de requête ou de vues assez conséquentes, ce qui est très bénéfique pour l'exploitation des données.

Les données démographiques que nous avons exploitées sont issues du Département des Affaires Économiques et Sociales (Division Population) de l'ONU et constituent une référence mondiale pour l'étude des dynamiques démographiques. Elles offrent des estimations et projections détaillées de la population par pays, âge, sexe et indicateurs connexes (fécondité, mortalité, migration, espérance de vie) et sur une période donnée : entre 1950 et 2023.

Parmi les données traitées, on retrouve par exemple : la population totale (au 1er Janvier et au 1er Juillet sur un période), le taux de natalité et de mortalité, l'espérance de vie, la croissance démographique, la densité par population, la migration nette, la ratio Homme/Femme, et d'autre projections et indications liées à l'âge ou au sexe. Ces données étant contenues dans un fichier CSV, il a fallu que nous les importions sur DBeaver en modifiant quelques paramètres primordiaux comme par exemple le séparateur de virgule, qui permet la bonne délimitation des colonnes.

III. SCHÉMA RELATIONNEL CLAIR



IV. REQUÊTES PRINCIPALES UTILISÉES

Parmi les principales requêtes utilisées, on retrouve celles qui permettent de vérifier la bonne importation du fichier CSV, de créer et de peupler les tables de dimensions et de faits comme par exemple la table region, subregion et country. :

Pour vérifier si l'importation a été correcte :

```
SELECT *
FROM pop
LIMIT 50;
```

Et on obtient ceci :

1	Estimates	World		900			
2	Estimates	World		900			
3	Estimates	World		900			
4	Estimates	World		900			

Pour la table region :

```sql
CREATE TABLE region (
 location_code INTEGER PRIMARY KEY,
 name TEXT NOT NULL
);

INSERT INTO region (location_code, name)
SELECT DISTINCT
 "Location code",
 "Region, subregion, country or area"
FROM pop
WHERE "Type" = 'Region';
```

Pour la table Sous-region :

```sql
CREATE TABLE subregion (
 location_code INTEGER PRIMARY KEY,
 name TEXT NOT NULL,
 parent_code INTEGER,
 FOREIGN KEY (parent_code) REFERENCES region(location_code)
);

INSERT INTO subregion
SELECT DISTINCT
 "Location code",
 "Region, subregion, country or area",
 "Parent code"
FROM pop
WHERE "Type" = 'Subregion';
```

Pour la table country :

```sql
CREATE TABLE country (
 location_code INTEGER PRIMARY KEY,
 name TEXT NOT NULL,
 parent_code INTEGER,
 FOREIGN KEY (parent_code) REFERENCES subregion(location_code)
);

INSERT INTO country
SELECT DISTINCT
 "Location code",
 "Region, subregion, country or area",
 "Parent code"
FROM pop
WHERE "Type" = 'Country/Area';
```

Pour l'ajour de la table contry :

```sql
CREATE TABLE continent (
    continent_code INTEGER PRIMARY KEY,
    name           TEXT NOT NULL
);

-- 2) Insertion des 3 continents du sujet
INSERT INTO continent (continent_code, name) VALUES
    (1, 'Afro-Eurasien'),        -- Afrique + Europe + Asie
    (2, 'Americain'),           -- Northern America + Latin America and the Caribbean
    (3, 'Oceanien');            -- Oceania
```

Ensuite pour la table « fact_population », on la crée avec toutes les données démographiques. On fait de même pour les **INSERT** et les **SELECTS**. De plus, nous avons fait le choix de ne pas diviser cette table pour une question de praticité.

Enfin pour les requêtes principales permettant d'analyser les données démographiques, nous avons fait une petite sélection sur celles que nous avons jugées pertinentes. On retrouve par exemple celle qui permet d'afficher les 10 pays ayant la plus forte population en 2020 :

Country	Population_2020_Thousands
China	1 426 106,093
India	1 402 617,695
United States of America	339 436,159
Indonesia	274 814,866
Pakistan	235 001,746
Nigeria	213 996,181
Brazil	208 660,842
Bangladesh	166 298,024
Russian Federation	146 371,299
Mexico	126 799,054

Pour afficher le taux de natalité et le taux de mortalité moyens par région pour 2000.

Region	TauxNataliteMoyen	TauxMortaliteMoyen
Africa	36,95	12,2
Asia	22,65	6,47
Europe	11,26	10,08
Latin America and the Caribbean	20,58	6,76
Northern America	13,37	8,07
Oceania	25,67	6,55

Pour identifier les sous-régions dont la population représente plus de 40% de la population totale de la région en 2023, en exécutant la requête on trouve :

Region	SousRegion	PopSousRegion	PopRegion	Part_Region_Pourcentage
Oceania	Australia/New Zealand	31 623,96	45 562,783	69,41
Latin America and the Caribbean	South America	433 024,175	658 891,518	65,72
Asia	Southern Asia	2 043 083,159	4 778 004,49	42,76

En ce qui concerne la comparaison de l'espérance de vie homme/femme dans chaque région, on trouve ceci :

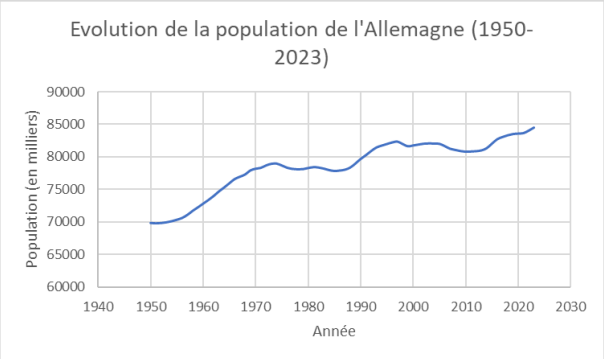
Region	Esperance_Hommes	Esperance_Femmes	Ecart_Femmes_Moins_Hommes
Latin America and the Caribbean	71,37244	77,99804	6,63
Europe	76,26786	82,03876	5,77
Asia	71,6375490196	77,1626666667	5,53
Oceania	69,6739565217	74,8353043478	5,16
Africa	62,354862069	67,1529482759	4,8

Enfin, dans ce projet nous avons appris à faire des vues ou autrement dit des requêtes nommées. Voici un exemple de résultat d'une vue qui permet d'afficher la population moyenne, minimale et maximale par région sur toute la période.

Apres exécution du **SELECT,** on obtient ceci :

region_name	pop_min	pop_max	pop_moyenne
Africa	227 776	1 480 771	672 221
Asia	1 368 075	4 778 004	3 028 060
Europe	548 867	749 524	687 737
Latin America and the Caribbean	167 782	658 892	413 721
Northern America	168 009	382 903	275 246
Oceania	12 582	45 563	26 790

V. UNE VISUALISATION



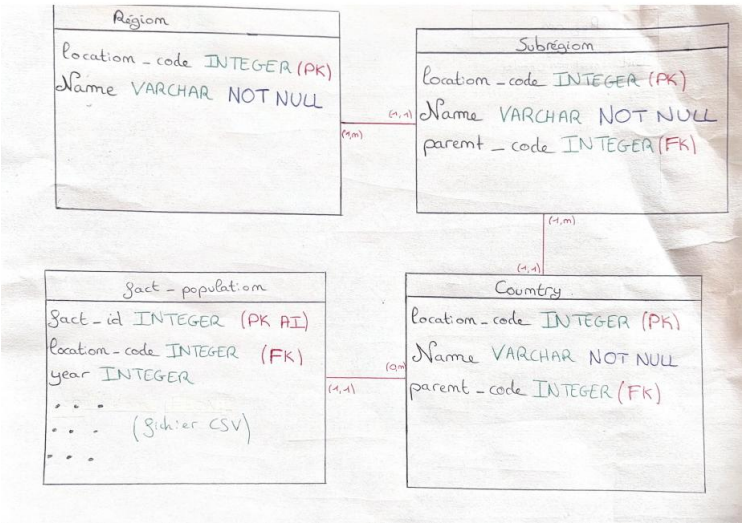
VI. RÉPONSES AUX QUESTIONS

Question de réflexion 1 :
Les données géographiques sont des informations textuelles et hiérarchiques qui changent rarement (noms des pays, régions, codes ISO). Elles sont stockées dans des tables de dimensions (ex : *country*, *region*).
Les données démographiques sont des valeurs numériques qui évoluent chaque année (population, natalité, etc.). Elles sont regroupées dans une table de faits (*fact_population*) reliant les dimensions via des clés étrangères (*location_code*).

Question de réflexion 2 :
Le projet utilise SQLite car c'est un SGBD relationnel très rapide, simple à manipuler et ne nécessitant aucune installation de serveur. Toute la base est contenue dans un seul fichier, ce qui facilite le partage et l'utilisation commune.
Cependant, SQLite a des limites : baisse de performances sur de très grosses bases et gestion d'une seule connexion à la fois. Pour des bases volumineuses, on utilisera plutôt PostgreSQL ou MySQL.

Question de vérification 3 :
En observant les premières lignes correspondant à "World", on remarque une répétition inutile des informations. Le code « 900 » (LOCATION CODE), le nom « World » (REGION, SUBREGION, TYPE) ou encore le code « 0 » (PARENT CODE) sont répétés pour chaque année (1950, 1951, 1952, etc.), ce qui crée une forte redondance.

Question de réflexion 4 :
On sépare régions, sous-régions et pays pour éviter cette redondance : les informations fixes ne sont stockées qu'une seule fois dans des tables dédiées. Cela renforce la cohérence, garantit l'intégrité des données et facilite les calculs statistiques grâce aux jointures et à l'usage d'identifiants numériques.

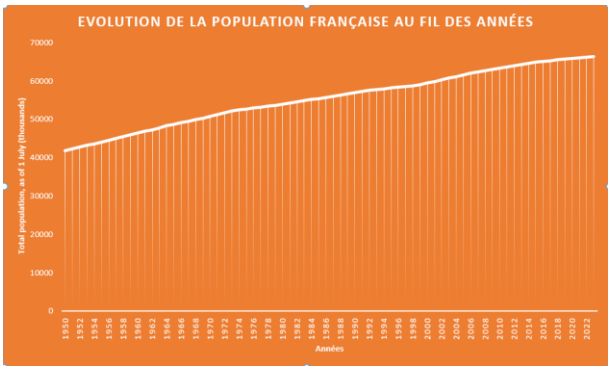


Question de réflexion 5 :
Une clé primaire identifie de manière unique chaque ligne d'une table (ex : *region.location_code*, *subregion.location_code*, *country.location_code*, *continent.location_code*, *fact_population.fact_id*). Elle empêche les doublons et permet aux autres tables de s'y référer.
Une clé étrangère est un champ qui référence la clé primaire d'une autre table pour établir une relation logique.
Les Primary Key et Foreign Key assurent la cohérence grâce à la cohérence référentielle : impossible, par exemple, de supprimer une région si des sous-régions y sont rattachées, ou de supprimer une sous-région si des pays en dépendent. Cela évite d'avoir des entités « qui sortent de nulle part ».

Question de réflexion 6 :
On utilise « fact_id » car c'est une clé artificielle. Elle est plus simple à gérer pour le moteur de base de données (un simple nombre entier) et reste stable même si les données métier changent.
« location_code » et « year » peuvent servir de clés, car leur combinaison est unique (il n'y a qu'une seule ligne par pays pour une année donnée).
La clé primaire logique est la clé composite (location_code, year), car c'est elle qui garantit l'unicité fonctionnelle de l'information.

Question #7a :
L'Asie est le continent le plus peuplé avec environ 3,7 milliards d'habitants, loin devant l'Afrique et l'Europe. Cette évolution depuis 1950 peut s'expliquer par la transition démographique, caractérisée par une baisse rapide de la mortalité alors que la natalité est restée forte durablement. Elle est aussi due à l'explosion démographique de pays majeurs comme l'Inde, la Chine ou le Japon, augmentant ainsi la croissance globale du continent.

Question #7b :
Voici le graphique qui correspond à l'évolution de la population Française depuis 1950



La tendance observée est une hausse croissante de la population française depuis 1950.

Question #7c :
On peut observer cette relation dans les résultats : Algeria → Northern Africa → Africa. Cela est donc cohérent avec nos connaissances étant donné que l'on sait que l'Algérie se trouve en Afrique et plus précisément en Afrique du nord.

Question #8a :
Pour construire cette vue, la clause « **GROUP BY** » est primordiale car la requête ici utilise la fonction d'agrégation (**SUM**). Cela nous permet donc de regrouper toutes les lignes qui partagent la même Région et la même année pour en calculer le total.

La clause « **ORDER BY** » à la fin de la vue sert à trier par défaut les résultats. Elle permet de présenter les données de manière structurée et lisible. Cela évite d'avoir un résultat en désordre lorsqu'on fait un simple **SELECT**

Question #8b :
On utilise **SUM ()** pour calculer le dénominateur, qui correspond à la population totale de la région. Sans cette somme globale, il est impossible de calculer la part que représente un seul pays.

Il faudrait effectuer trois changements : filtrer les années avec une clause **WHERE** (ex : **WHERE** year >= 2013), remplacer le calcul direct par la fonction **AVG ()** pour obtenir la moyenne des pourcentages et simplifier le regroupement en retirant l'année du **GROUP BY** pour n'avoir qu'une seule ligne par pays.

VII. CONCLUSION

En conclusion, ce projet qui s'inscrit dans le cadre de cette SAÉ 1.04, nous a permis de mettre en œuvre nos capacités et nos compétences à concevoir, importer, nettoyer, interroger et analyser une base de données à partir d'un fichier CSV brut, qui contenait des données démographiques officielles de L'ONU, du Département des Affaires Économiques et Sociales (Division Population), entre 1950 et 2023.

Toutes les étapes de ce projet ont été importantes et primordiales :

- La création de la base avec l'importation du fichier CSV sur DBeaver, en mettant le point-virgule comme séparateur de colonne, ce qui permet ainsi la bonne délimitation des colonnes et des données.

- La conception du modèle relationnel qui nous a permis de comprendre comment sont structurées les données. Nous avons remarqué qu'il y a deux catégories de données majeures : les données géographiques telles que « region », « subregion », « country », stockées dans des tables de « dimensions » et les données démographiques, « fact_populations », stockées dans une table de « faits ».

- L'exécution des requêtes ou la création des vues telles que la vérification de l'importation des données issues du CSV, le peuplement des tables de faits et de dimensions, l'évolution de la population d'un pays sur une période donnée, le taux de natalité/mortalité, la migration nette ou encore l'espérance de vie, permettent ainsi de valider et de vérifier la base de données tout en procédant à son analyse.

Cette SAÉ nous a permis de prendre conscience sur la réflexion à la modélisation et à la qualité des données. Toutes les données issues d'un fichier CSV doivent être importées, manipulées et traitées minutieusement avec beaucoup de sérieux.

A la suite de ce projet, la base de données créée sera utilisée comme point de départ dans le cadre de la SAÉ 1.01 pour le développement d'une application WEB interactive pour un client.
Ces deux projets sont complémentaires car ils permettent de voir très clairement le lien entre la gestion des données le *backend* et l'interface du client *frontend*.
Le *frontend* sera programmé en 2 langages de programmation tels que Python avec la technologie Flask et JavaScript, qu'on abordera dans la SAÉ 1.01.