

DOCUMENTATION PROJET IMMO - projet 3 Data Analyst OpenClassrooms

Préparation des données dans une BDD de brouillon appelée : brouillon_projet_3

- Récupération des données sources d'OC
- Prise de connaissance des données sources
- Vérification des compatibilités pour concaténation de clé codeDep_codeCommune
- Inclusion colonne avec concaténation des code dep et code commune dans les données sources
- Préparation de la structure des tables avec nom des colonnes et type dans phpMyAdmin
- Importation des données sous forme CSV dans phpMyAdmin

CRÉATION DE LA TABLE COMMUNES

```
### Création de la table COMMUNES dans la BDD de destination
```

```
CREATE TABLE COMMUNES ( id_codedep_codecommune VARCHAR (60), code_departement VARCHAR (60), code_commune VARCHAR (60), nom_commune VARCHAR (60), population VARCHAR (60), code_region VARCHAR (60) );
```

```
### Insertion des données dans table COMMUNES en provenance de la table DC
```

```
INSERT INTO COMMUNES (id_codedep_codecommune, code_departement, code_commune, nom, population, code_region) SELECT id_codedep_codecommune, CODDEP, CODCOM, COM, PTOT, CODREG FROM brouillon_projet_3.DONNEES_COMMUNES;
```

```
## CRÉATION DE LA TABLE RÉGIONS
```

```
CREATE TABLE REGIONS ( reg_code VARCHAR(60) PRIMARY KEY, reg_nom VARCHAR(60) ) AS SELECT DISTINCT reg_code, reg_nom FROM brouillon_projet_3.REFERENCIEL_GEO;
```

```
#### Création d'une clé étrangère qui permettra de lier la table COMMUNES à la table REGIONS
```

```
ALTER TABLE COMMUNES ADD CONSTRAINT fk_code_region FOREIGN KEY (code_region) REFERENCES  
REGIONS(reg_code);
```

```
## CRÉATION DE LA TABLE BIENS
```

```
### Vérification des données à récupérer
```

```
SELECT DISTINCT Code_departement, Code_commune, No_voie, Voie, BTQ, Type_voie, Nombre_pieces,  
Surface_reelle, Surface_Carrez, Type_local, Code_postal FROM brouillon_projet_3.VALEURS_FONCIERES
```

```
### Création de la table BIENS
```

```
CREATE TABLE BIENS ( id_biens INT, codedep_codecommune_id VARCHAR (60), code_departement  
VARCHAR (60), code_commune VARCHAR (60), No_voie VARCHAR (60), BTQ VARCHAR (60), Type_voie  
VARCHAR (60), Voie VARCHAR (60), Nombre_pieces VARCHAR (60), Surface_reelle VARCHAR (60),  
Surface_Carrez VARCHAR (60), Type_local VARCHAR (60), Code_postal VARCHAR (60)  
);
```

```
### Création d'une clé primaire dans COMMUNES qui permettra de lier la  
table COMMUNES à la table BIENS
```

```
ALTER TABLE COMMUNES ADD PRIMARY KEY (id_biens);
```

```
### Création d'une FK qui matche avec la clé primaire de la table COMMUNES  
(création de la relation)
```

```
#### (vérifier que les types de données sont identiques)
```

```
ALTER TABLE BIENS ADD FOREIGN KEY (codedep_codecommune_id) REFERENCES COMMUNES  
(id_codedep_codecommune);
```

```
## Création d'une colonne id_VF dans VALEURS_FONCIERES auto-incrémentée,
```

afin de rendre chaque vente unique
Un biens peut être vendus plusieurs fois, se situe donc au même endroit
mais génère plusieurs ventes
Cela permettra de rendre unique chaque vente dans l'implémentation de la
table BIENS et au moment de la jointure avec la table VENTES

```
ALTER TABLE brouillon_projet_3.VALEURS_FONCIERES ADD COLUMN id_VF INT AUTO_INCREMENT;
```

Implémenter cette table avec les données

```
INSERT INTO BIENS (id_biens, codedep_codecommune_id, code_departement,  
code_commune, No_voie, BTQ, Type_voie, Voie, Nombre_pieces,  
Surface_reelle, Surface_Carrez, Type_local, Code_postal)  
SELECT DISTINCT  
id_VF,  
id_codedep_codecommune,  
Code_departement,  
Code_commune,  
No_voie,  
BTQ,  
Type_voie,  
Voie,  
Nombre_pieces,  
Surface_reelle,  
Surface_Carrez,  
Type_local,  
Code_postal  
  
FROM brouillon_projet_3.VALEURS_FONCIERES;  
  
## CRÉATION TABLE VENTES
```

```
CREATE TABLE VENTES ( id_vente INT AUTO_INCREMENT, biens_id INT, date_vente DATE, prix  
VARCHAR(60), FOREIGN KEY (biens_id) REFERENCES BIENS(id_biens) );
```

Vérification des données dans table VALEURS_FONCIERES

```
SELECT * from brouillon_projet_3.VALEURS_FONCIERES WHERE voie = 'DU PARC 3';
```

**// Cela retourne des ventes sans valeur fonciere --> il faut en tenir

```
compte dans la requête SQL suivante/**
```

```
INSERT INTO VENTES (biens_id, date_vente, prix) SELECT b.id_biens, v.Date_mutation, v.Valeur_fonciere
FROM BIENS AS b JOIN brouillon_projet_3.VALEURS_FONCIERES AS v ON b.id_biens = v.id_VF, AND
b.code_departement = v.Code_departement AND b.code_commune = v.Code_commune AND b.No_voie =
v.No_voie AND b.BTQ = v.BTQ AND b.Type_voie = v.Type_voie AND b.Voie = v.Voie AND b.Nombre_pieces =
v.Nombre_pieces AND b.Surface_carrez = v.Surface_Carrez AND b.Type_local = v.Type_local AND
b.Code_postal = v.Code_postal WHERE v.Valeur_fonciere != ' ';
```

```
### Warning sur les champs dates que l'on corrige comme ceci :
```

```
UPDATE projet_data_immo.VENTES SET date_vente = REPLACE(date_vente, '/', '-') WHERE date_vente
LIKE '%/%';
```

```
## STRUCTURE DE LA BDD

- TABLE BIENS = 34 169 lignes
- TABLE COMMUNES = 34991 lignes
- TABLE REGIONS = 19 lignes
- TABLE VENTES = 34 169 lignes
```

VERIFICATION NOMBRES DE VENTES PAR BIENS

```
SELECT b.codedep_codecommune_id, b.No_voie, b.Voie, b.BTQ, COUNT(*) AS
nombre_ventes
FROM BIENS b
LEFT JOIN VENTES v ON b.id_biens = v.biens_id
GROUP BY b.codedep_codecommune_id, b.No_voie, b.Voie, b.BTQ
ORDER BY nombre_ventes DESC;
```

codedep_codecommune_id	No_voie	Voie	BTQ	nombre_ventes
33522	690	DE LA LIBERATION		13
13209	83	DU REDON		11
77014	9001	AVE DE GAULLE 27 FOUGERES		9
91161	36	MAZARIN		8
77285	120	DE PLEIN CIEL		8

codedep_codecommune_id	No_voie	Voie	BTQ	nombre_ventes
91226	4	DE LA ROCHE BENOTTE		8
14754	9006	RES SUPER DEAUVILLE		8
45006	200	DES 4 VENTS		7
76540	1	DE HANOVRE		7
91521	9001	DU TEMPLE		7
91477	9001	D ARDENAY		7
2A130	5400	DE TERRA BELLA		7
14754	9005	RES LE FRANCE		7
33550	9001	SARCIGNAN		7
33214	9002	BLEU MARINE		7
7324	113	DU HUIT MAI 1945		7
44109	104	DU CORPS DE GARDE		6
34301	9077	DU CHATEAU VERT		6
95428	77	DES CHESNEAUX	B	6
83144	9001	LA COUPIANE		6
77449	42	DU DANUBE		6
42005	9004	MOLIERE		6
94017	98	DU GENERAL DE GAULLE		6
69381	13	STE CATHERINE		6
78172	75	ARISTIDE BRIAND		6