

Projet 3 – Business Intelligence Analyst

Requêtez une base de données
avec SQL

Méthodologie du Projet 3

- 1. La création du dictionnaire de données
- 2. La création du schéma relationnel
- 3. La création de la Base sur Postgre SQL
- 4. L'exploitation de la Base

A blackboard filled with mathematical calculations and formulas. Some parts of the board are circled in white, highlighting specific equations or symbols.

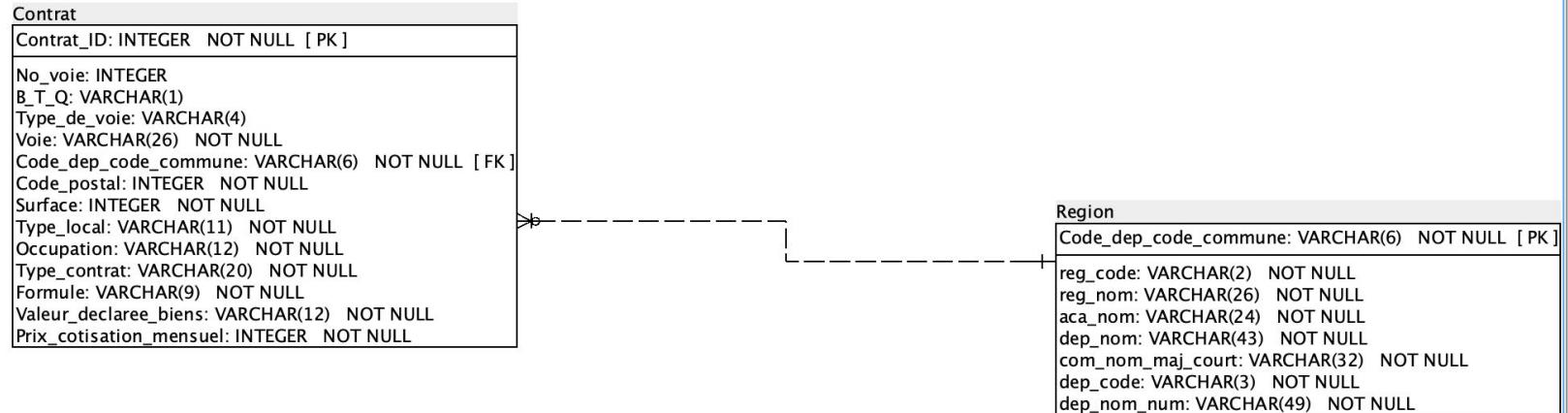
- A circled equation: $D(x) = a + b + 4.31447$
- A circled integral symbol: \int
- A circled equation: $\sqrt{a^2 + b^2} = x^2$
- A circled symbol: $c(x, y)$
- A circled equation: $x^2 + y^2 = ab + bc$
- A circled fraction: $\frac{2x}{c} + \frac{a^2 + b^2}{c} + x^2$
- A circled equation: $\text{men} = 584. + n^{0.07}$
- A circled value: $x = 9.23$
- A circled sum: $\sum N_{30} \cdot x - \frac{1}{2} [964 + xg + \dots]$
- A circled matrix:

010112
010002
011002
- A circled value: $t = 4$
- A circled diagram: A triangle with vertices labeled A, B, and C.
- A circled equation: $\beta = 9 + x^2 + y^2$

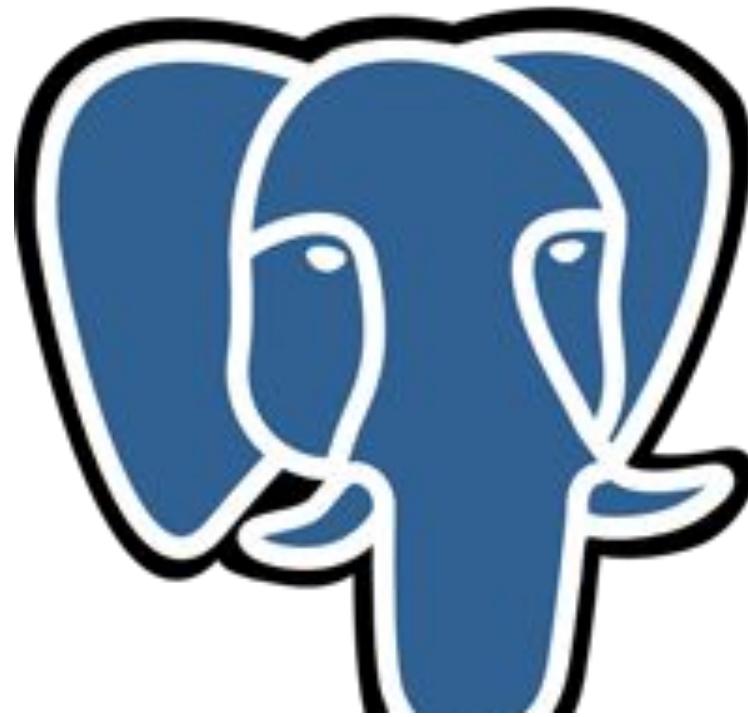
1. La création du dictionnaire de données

	Nom des colonnes	Type de données	Taille	Accepte les NULS	Clé	Description
CONTRAT.CSV	Contrat_ID	INT	6	NON	Clé primaire artificielle Id unique pour les contrats Numéro dans la voie pour l'adresse du logement assuré Indicateur éventuel de répétition pour l'adresse du logement assuré sur un caractère Type de voie pour l'adresse du logement assuré: rue, av (Avenue), rte (Route), ... Libellé de la voie pour l'adresse du logement assuré Concaténation du code département et code commune pour avoir une clé unique Code postal pour l'adresse du logement assuré Superficie assurée Type de logement: appartement, maison,... Statut de l'assuré vis-à-vis du bien : propriétaire, locataire,... Statut du logement pour l'assuré : résidence principale, secondaire,... le type de protection choisi le montant total des biens assurés selon une fourchette le prix versé mensuellement par l'assuré	Clé primaire artificielle Id unique pour les contrats Numéro dans la voie pour l'adresse du logement assuré Indicateur éventuel de répétition pour l'adresse du logement assuré sur un caractère Type de voie pour l'adresse du logement assuré: rue, av (Avenue), rte (Route), ... Libellé de la voie pour l'adresse du logement assuré Concaténation du code département et code commune pour avoir une clé unique Code postal pour l'adresse du logement assuré Superficie assurée Type de logement: appartement, maison,... Statut de l'assuré vis-à-vis du bien : propriétaire, locataire,... Statut du logement pour l'assuré : résidence principale, secondaire,... le type de protection choisi le montant total des biens assurés selon une fourchette le prix versé mensuellement par l'assuré
	No_voie	INT	4	OUI		
	B_T_Q	VARCHAR	1	OUI		
	Type_de_voie	VARCHAR	4	NON		
	Voie	VARCHAR	26	NON		
	Code_dep_code_commun	VARCHAR	6	NON		
	Code_postal	INT	5	NON		
	Surface	INT	3	NON		
	Type_local	VARCHAR	11	NON		
	Occupation	VARCHAR	12	NON		
	Type_contrat	VARCHAR	20	NON		
	Formule	VARCHAR	9	NON		
	Valeur_declaree_biens	VARCHAR	12	NON		
	Prix_cotisation_mensuel	INT	3	NON		
REGION.CSV	Code_dep_code_commun	VARCHAR	6	NON	Clé primaire artificielle Concaténation du code département et code commune pour avoir une clé unique Code de la Région Nom de la Région Nom de la capitale régionale nom du département Nom de la ville Code départemental Numéro + nom du département	Concaténation du code département et code commune pour avoir une clé unique Code de la Région Nom de la Région Nom de la capitale régionale nom du département Nom de la ville Code départemental Numéro + nom du département
	reg_code	VARCHAR	2	NON		
	reg_nom	VARCHAR	26	NON		
	aca_nom	VARCHAR	24	NON		
	dep_nom	VARCHAR	43	NON		
	com_nom_maj_court	VARCHAR	32	NON		
	dep_code	VARCHAR	3	NON		
	dep_nom_num	VARCHAR	49	NON		

2. La création du schéma relationnel



3. La création de la Base sur Postgre SQL



PostgreSQL

Cette photo par Auteur inconnu est soumise à la licence CC BY-SA

3.1.1. La création des tables

```
CREATE TABLE Schéma - Etudiant.Region ( Code_dep_code_commune VARCHAR(6) NOT
NULL, reg_code VARCHAR(2) NOT NULL,
reg_nom VARCHAR(26) NOT NULL,
aca_nom VARCHAR(24) NOT NULL,
dep_nom VARCHAR(43) NOT NULL, com_nom_maj_court VARCHAR(32) NOT NULL,
dep_code VARCHAR(3) NOT NULL,
dep_nom_num VARCHAR(49) NOT NULL,
CONSTRAINT code_dep_code_commune PRIMARY KEY
(Code_dep_code_commune);

CREATE TABLE Schéma - Etudiant.Cotrat ( Contrat_ID INTEGER NOT NULL,
No_voie INTEGER,
B_T_Q VARCHAR(1),
Type_de_voie VARCHAR(4),
Voie VARCHAR(26) NOT NULL, Code_dep_code_commune VARCHAR(6) NOT NULL, Code_postal INTEGER NOT NULL,
Surface INTEGER NOT NULL,
Type_local VARCHAR(11) NOT NULL,
Occupation VARCHAR(12) NOT NULL,
Type_contrat VARCHAR(20) NOT NULL,
Formule VARCHAR(9) NOT NULL, Valeur_declaree_biens VARCHAR(12) NOT NULL, Prix_coyer_mensuel INTEGER NOT
NULL, CONSTRAINT contrat_id PRIMARY KEY (Contrat_ID)
);

ALTER TABLE Schéma - Etudiant.Cotrat ADD CONSTRAINT region_contrat_ FOREIGN KEY (Code_dep_code_commune)
REFERENCES Schéma - Etudiant.Region (Code_dep_code_commune)
ON DELETE NO ACTION
ON UPDATE NO ACTION NOT DEFERRABLE;
```

3.1.2 La création des tables

The screenshot shows the pgAdmin 4 interface. The left pane is the Object Explorer, which lists various database objects like Foreign Data Wrappers, Languages, Publications, Schemas (1), and Tables (2). Under 'Tables (2)', the 'contrat' table is selected. The right pane contains a query editor with a 'Query' tab and a 'Scratch Pad' tab. Below the tabs is a data grid titled 'contrat'. The data grid shows columns: contrat_id [PK] integer, no_voie integer, b_tq character varying (1), type_de_voie character varying (4), and voie character varying (26). The data rows are as follows:

contrat_id	no_voie	b_tq	type_de_voie	voie
100601	190	A	RUE	CENTRALE
100602	347	[null]	RUE	DU CHATEAU
100603	58	[null]	AV	DU MONT BLANC
100604	140	[null]	RUE	DE LABBE JOLIVET
100605	39	[null]	RUE	BUFFON
100606	8	[null]	RUE	DE GENEVE
100607	2	[null]	RUE	DU RECULET

Total rows: 30326 Query complete 00:00:00.351 LF Ln 1, Col 1

The screenshot shows the pgAdmin 4 interface. The left pane is the Object Explorer, similar to the first screenshot but with different object types listed under Schemas (1). The 'region' table is selected. The right pane contains a query editor with a 'Query' tab and a 'Scratch Pad' tab. Below the tabs is a data grid titled 'region'. The data grid shows columns: code_dep_code_communne [PK] character varying (6), reg_code character varying (2), reg_nom character varying (26), eca_nom character varying (24), dep_nom character varying (43), and com_nom character v. The data rows are as follows:

code_dep_code_communne	reg_code	reg_nom	eca_nom	dep_nom	com_nom
1	10001	44	Grand Est	Aube	L' ABBAYE :
2	10002	44	Grand Est	Reims	AUBEVILLE
3	10003	44	Grand Est	Reims	AIX EN OTI
4	10004	44	Grand Est	Reims	ALLIBAUDI
5	10005	44	Grand Est	Reims	AMANCE
6	10006	44	Grand Est	Reims	ARCIS SUR
7	10007	44	Grand Est	Reims	ARCONVIL
8	10008	44	Grand Est	Reims	ARGANCOF

Total rows: 38916 Query complete 00:00:00.539 LF Ln 1, Col 1

4. L'exploitation de la Base

Requête 1

The screenshot shows the pgAdmin 4 interface with the Object Explorer on the left and a Query Editor window on the right.

Object Explorer: Shows the database structure under PostgreSQL 17, including Databases (2), Schemas (1), and Tables (2) named `contrat`.

Query Editor: Displays the following SQL query:

```
1  SELECT
2      c.Contrat_ID,
3      c.Surface
4  FROM
5      Contrat c
6  JOIN
7      Region r
8  ON
9      c.Code_dep_code_commune = r.Code_dep_code_commune
10     r.com_nom_maj_court = 'CAEN';
11
12
13
14
15
```

Data Output: Shows the results of the query:

	contrat_id	surface
1	103791	35
2	103792	99
3	103793	40
4	103794	20

Requête 2

The screenshot shows the pgAdmin 4 interface with the Object Explorer on the left and a Query Editor window on the right.

Object Explorer: Shows the database structure under PostgreSQL 17, including Databases (2), Schemas (1), and Tables (2) named `contrat`.

Query Editor: Displays the following SQL query:

```
1  SELECT
2      c.Contrat_ID,
3      c.Type_Contrat,
4      c.Formule
4  FROM
5      Contrat c
6  JOIN
7      Region r
8  ON
9      c.Code_dep_code_commune = r.Code_dep_code_commune
10     WHERE
11         c.Type_local = 'Maison'
12         AND r.dep_code = '71';
13
14
15
```

Data Output: Shows the results of the query:

	contrat_id	type_contrat	formule
1	114768	Residence principale	Integral
2	114779	Residence principale	Classique
3	114782	Residence principale	Classique
4	114812	Residence principale	Integral

Requête 3

```
pgAdmin 4
Object Explorer
PROJECT 3 /postgres@PostgreSQL 17
Query History
SELECT DISTINCT
    r.reg_nom
FROM
    Region r
ORDER BY
    r.reg_nom ASC;
```

The screenshot shows the pgAdmin 4 interface with the Object Explorer on the left and a query editor on the right. The query editor displays the following SQL code:

```
1 SELECT DISTINCT
2     r.reg_nom
3 FROM
4     Region r
5 ORDER BY
6     r.reg_nom ASC;
```

The results pane shows a list of regions from the 'region' table:

reg_nom
Ile-de-France
La Réunion
Martinique
Mayotte
Normandie
Nouvelle-Aquitaine
Occitanie
Pays de la Loire
Provence-Alpes-Côte d'Azur

Total rows: 19 Query complete 00:00:00.103

Requête 4

```
pgAdmin 4
Object Explorer
PROJECT 3 /postgres@PostgreSQL 17
Query History
SELECT
    Contrat_ID,
    Surface
FROM
    Contrat
ORDER BY
    Surface DESC
LIMIT 5;
```

The screenshot shows the pgAdmin 4 interface with the Object Explorer on the left and a query editor on the right. The query editor displays the following SQL code:

```
1 SELECT
2     Contrat_ID,
3     Surface
4 FROM
5     Contrat
6 ORDER BY
7     Surface DESC
8 LIMIT 5;
```

The results pane shows a list of contracts from the 'contrat' table:

Contrat_ID	Surface
104211	815
105463	742
1300878	595
100822	570
109872	559

Total rows: 5 Query complete 00:00:00.175

Requête 5

```
pgAdmin 4
Object Explorer
PROJECT 3 /postgres@PostgreSQL 17
Query History
SELECT
    AVG(Prix_cotisation_mensuel) AS Prix_moyen_cotisation
FROM
    Contrat;
```

The screenshot shows the pgAdmin 4 interface with the Object Explorer on the left and a query editor on the right. The query editor displays the following SQL code:

```
1 SELECT
2     AVG(Prix_cotisation_mensuel) AS Prix_moyen_cotisation
3 FROM
4     Contrat;
```

The results pane shows a single row of data:

Prix_moyen_cotisation
19.330468245070237

Total rows: 1 Query complete 00:00:00.127

Requête 6

```
pgAdmin 4
Object Explorer
PROJECT 3 /postgres@PostgreSQL 17
Query History
SELECT
    COUNT(*) AS Nombre_de_contrats
FROM
    Contrat
GROUP BY
    Valeur_declaree_biens
ORDER BY
    Nombre_de_contrats DESC;
```

The screenshot shows the pgAdmin 4 interface with the Object Explorer on the left and a query editor on the right. The query editor displays the following SQL code:

```
1 SELECT
2     COUNT(*) AS Nombre_de_contrats
3 FROM
4     Contrat
5 GROUP BY
6     Valeur_declaree_biens
7 ORDER BY
8     Nombre_de_contrats DESC;
```

The results pane shows a list of contract counts grouped by declared value:

Valeur_declaree_biens	Nombre_de_contrats
0-2500	22712
2500-5000	6814
5000-10000	696
10000+	104

Total rows: 4 Query complete 00:00:00.143

Requête 7

```
pgAdmin 4
Object Explorer
PROJECT 3 /postgres@PostgreSQL 17
Query History
SELECT COUNT(*) AS Nombre_formules_integral
FROM Contrat c
JOIN Region r
ON c.Code_dep_code_commune = r.Code_dep_code_commune
WHERE c.Formule = 'Integral'
AND r.reg_nom = 'Pays de la Loire';
Data Output
Showing rows: 1 to 1 | Page No: 1 of 1 | 14 <> >
1 589
Total rows: 1 Query complete 00:00:00.116 LF Ln 12, Col 1
```

Requête 8

```
pgAdmin 4
Object Explorer
PROJECT 3 /postgres@PostgreSQL 17
Query History
SELECT Contract_ID, Type_contrat, Formule
FROM Contrat c
JOIN Region r
ON c.Code_dep_code_commune = r.Code_dep_code_commune
WHERE c.Type_local = 'Maison'
AND r.dep_code = '71';
Data Output
Showing rows: 1 to 4 | Page No: 1 of 1 | 14 <> >
control_id [PK] integer type_contrat character varying(20) formule character varying(9)
1 114768 Residence principale Integral
2 114779 Residence principale Classique
3 114782 Residence principale Classique
4 114812 Residence principale Integral
Total rows: 4 Query complete 00:00:00.116 LF Ln 12, Col 1
```

Requête 9

```
pgAdmin 4
Object Explorer
PROJECT 3 /postgres@PostgreSQL 17
Query History
SELECT AVG(c.Surface) AS Surface_moyenne
FROM Contrat c
JOIN Region r
ON c.Code_dep_code_commune = r.Code_dep_code_commune
WHERE r.aca_nom ILIKE '%Paris%';
Data Output
Showing rows: 1 to 1 | Page No: 1 of 1 | 14 <> >
1 57.69549859157851
Total rows: 1 Query complete 00:00:00.123 LF Ln 11, Col 1
```

Requête 10

```
pgAdmin 4
Object Explorer
PROJECT 3 /postgres@PostgreSQL 17
Query History
SELECT r.dep_nom, AVG(c.Prix_cotisation_mensuel) AS Prix_moyen_cotisation
FROM Contrat c
JOIN Region r
ON c.Code_dep_code_commune = r.Code_dep_code_commune
GROUP BY r.dep_nom
ORDER BY Prix_moyen_cotisation DESC
LIMIT 10;
Data Output
Showing rows: 1 to 10 | Page No: 1 of 1 | 14 <> >
dep_nom character varying(45) prix_moyen_cotisation numeric
1 Paris 36.390087326208546
2 Hauts-de-Seine 26.293159407096460
3 Val-de-Marne 19.824163967950378
4 Yvelines 18.88069750598338
5 Rhône 18.465518272629298
6 Ain 18.238688888888889
7 Alpes-Maritimes 18.13762287592583
8 Charente-Maritime 17.317397623076923
9 Haute-Savoie 17.148734172151899
Total rows: 10 Query complete 00:00:00.119 LF Ln 15, Col 1
```

Requête 11

pgAdmin 4

Object Explorer

Query History

```
1 v SELECT
2   r.com_nom_maj_court AS Commune,
3   COUNT(*) AS Nombre_de_contrats
4   FROM
5     Contrat c
6   JOIN
7     Region r
8   ON
9     c.Code_dep_code_communne = r.Code_dep_code_communne
10  GROUP BY
11    r.com_nom_maj_court
12  HAVING
13    COUNT(*) >= 150
14  ORDER BY
15    Nombre_de_contrats DESC;
```

Data Output

commune	nombre_de_contrats
PARIS 18	515
PARIS 17	468
PARIS 15	407
PARIS 16	394
NICE	387
PARIS 11	381
BORDEAUX	302
PARIS 20	302
NANTES	291

Total rows: 20 Query complete 00:00:00.138 LF Ln 1, Col 1

Requête 12

pgAdmin 4

Object Explorer

Query History

```
1 v SELECT
2   r.reg_nom AS Region,
3   COUNT(*) AS Nombre_de_contrats
4   FROM
5     Contrat c
6   JOIN
7     Region r
8   ON
9     c.Code_dep_code_communne = r.Code_dep_code_communne
10  GROUP BY
11    r.reg_nom
12  ORDER BY
13    Nombre_de_contrats DESC;
```

Data Output

region	nombre_de_contrats
Ile-de-France	14177
Provence-Alpes-Côte d'Azur	3279
Auvergne-Rhône-Alpes	3042
Nouvelle-Aquitaine	2038
Occitanie	1609
Pays de la Loire	1196
Hauts-de-France	1189
Bretagne	947
Normandie	824

Total rows: 16 Query complete 00:00:00.129 LF Ln 14, Col 1