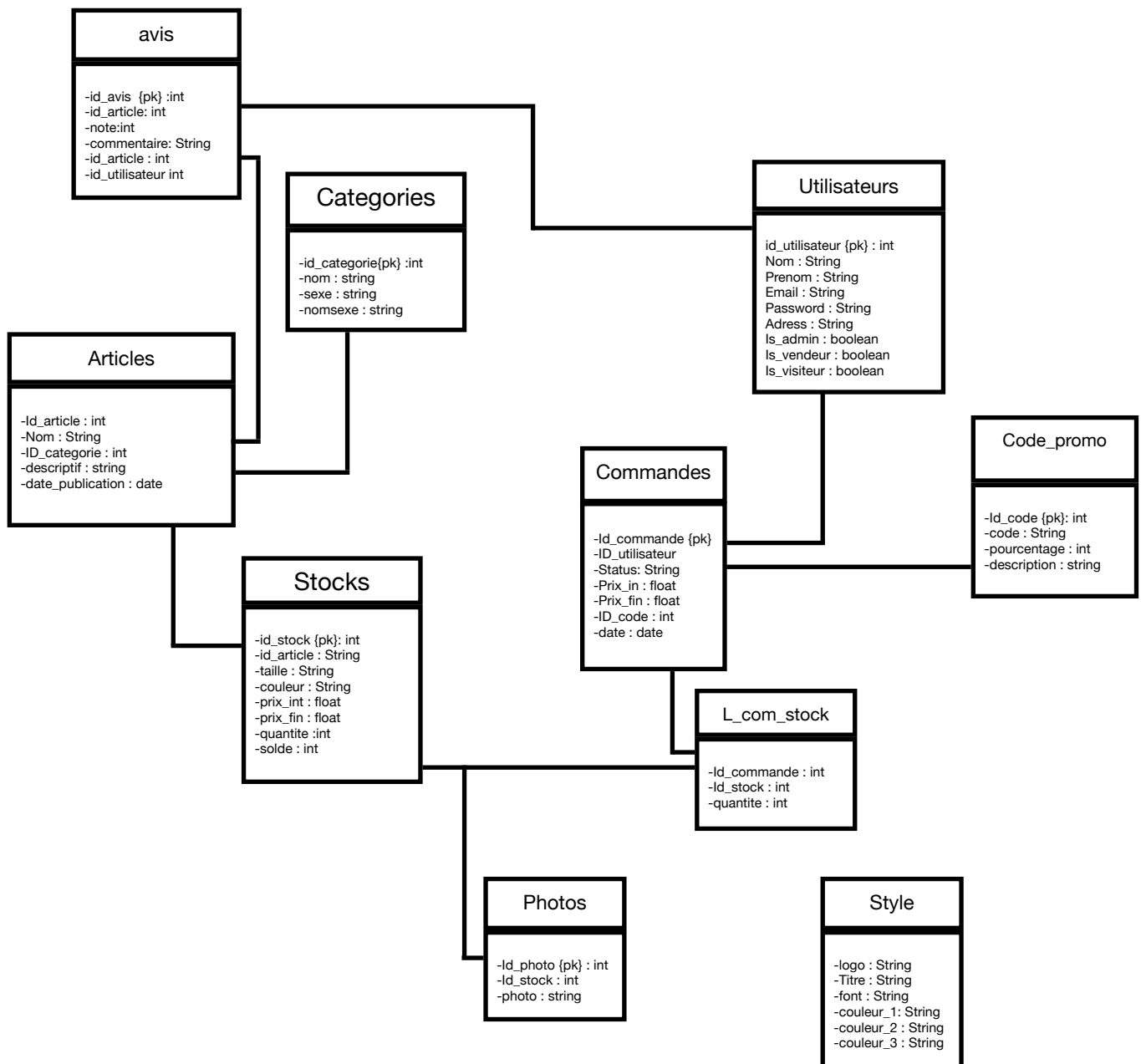


# Diagramme de la BDD



# Présentation de la BDD

## Utilisateur :

Cette table servira à stocker tous les utilisateurs du site (vendeur, inscrit ou invité).

- Id : clé primaire de la table
- Nom : nom de l'utilisateur
- Prenom : prénom de l'utilisateur
- Email : email de l'utilisateur
- Password : Mot-de-passe
- is\_admin : booléen pour dire si l'utilisateur est un admin ou pas
- is\_invite : booléen pour dire si l'utilisateur est un invité ou pas
- is\_vendeur : booléen pour dire si l'utilisateur est un vendeur ou pas

## Article :

Cette table servira à stocker tous les articles d'une boutique sans distinction de la couleur ou de la taille.

- Id
- Nom : Nom de l'article
- Id\_catégorie (Sexe&type) : clé étrangère pour les catégories (table Catégorie)
- Date\_publication La date de publication de l'article dans le site

## Commande :

Cette table servira à stocker les commandes faites sur le site ainsi que leurs statuts et s'ils sont dans le panier ou pas.

- Id
- id\_utilisateur : id de l'utilisateur ayant effectué la commande
- statut : statut de la commande qui changera en fonction du vendeur (livrée, en cours, etc)
- panier : booléen pour dire la commande est dans le panier ou si elle a été effectuée
- prix\_initial : prix avant la réduction
- prix\_final : prix après la réduction
- is\_code : int du code promotionnel

## Solde :

Cette table servira à stocker les différentes soldes générales. Ces soldes seront différentes de celles dans la table stock.

- Id
- Debut : date de début des soldes (exemple été, hiver, etc)
- Fin : date de fin des soldes
- Pourcentage : pourcentage des soldes
- id\_catégorie : catégorie concernée
- nom : nom de la solde

#### Style :

Table ne contenant qu'une seule ligne, permettant à l'administrateur de changer le nom du site, le logo du site, la couleur...

#### Stock :

Cette table servira à stocker les articles avec tous les détails. Ainsi, ici, nous ferons la distinction des articles en fonction de la couleur et de la taille.

- Id
- Taille : taille de l'article
- Couleur : couleur de l'article
- Quantité : quantité restante dans la boutique de l'article
- Prix\_initial: prix unitaire de base
- Solde : solde appliqué sur l'article
- Prix\_final : prix après la réduction de la solde
- id\_photo : clé étrangère pour la table photo
- id\_article : clé étrangère pour la table article

#### Catégorie :

Cette table permettra de stocker toutes les combinaisons de catégories. Par exemples : Homme & chemise ou Femme & robe.

- Id
- Nom : nom de la catégorie (exemple : homme, femme, col V, col rond, chemise, robe, etc)
- cat\_prec : catégorie spécifiée avant/ Catégorie précédente

#### Photo :

Cette table servira à stocker les liens vers les photos des articles qui ont été chargées et renseignées.

- Id
- id\_stock : clé étrangère sur la table stock
- photo : chemin vers la photo

#### CommandeStock :

Une table de liaison pour l'ensemble des articles commandés, justement par rapport à leur commande.

- id\_stock : clé primaire / clé étrangère sur la table stock
- id\_commande : clé primaire / clé étrangère sur la table commande
- quantité : quantité commandé

#### Code promo :

Cette table stockera toutes les promotions de la boutique sous forme de code.

- Id
- Code : Code de promotion
- Pourcentage : Pourcentage de la promotion
- description : description du code

# Script de la base de données

Créer une base de données appelé E-mirabeau.

```
-- Table: public.articles

-- DROP TABLE public.articles;

CREATE TABLE public.articles
(
    id_article integer NOT NULL,
    nom text COLLATE pg_catalog."default",
    id_categorie integer,
    CONSTRAINT articles_pkey PRIMARY KEY (id_article),
    CONSTRAINT articles_id_categorie_fkey FOREIGN KEY (id_categorie)
        REFERENCES public.categories (id_categorie) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE public.articles
    OWNER to postgres;


-- Table: public.categories

-- DROP TABLE public.categories;

CREATE TABLE public.categories
(
    id_categorie integer NOT NULL,
    nom text COLLATE pg_catalog."default",
    cat_prec integer,
    CONSTRAINT categories_pkey PRIMARY KEY (id_categorie),
    CONSTRAINT categories_cat_prec_fkey FOREIGN KEY (cat_prec)
        REFERENCES public.categories (id_categorie) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID
)

TABLESPACE pg_default;

ALTER TABLE public.categories
    OWNER to postgres;
```

```
-- Table: public.code_promos
```

```
-- DROP TABLE public.code_promos;
```

```
CREATE TABLE public.code_promos
(
    id_code integer NOT NULL,
    code text COLLATE pg_catalog."default",
    pourcentage integer,
    description text COLLATE pg_catalog."default",
    CONSTRAINT code_promos_pkey PRIMARY KEY (id_code)
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE public.code_promos
    OWNER to postgres;
```

```
-- Table: public.com_stock
```

```
-- DROP TABLE public.com_stock;
```

```
CREATE TABLE public.com_stock
(
    id_commande integer NOT NULL,
    id_stock integer NOT NULL,
    CONSTRAINT com_stock_id_fkey FOREIGN KEY (id_stock)
        REFERENCES public.stocks (id_stock) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
        NOT VALID,
    CONSTRAINT "none" FOREIGN KEY (id_commande)
        REFERENCES public.commandes (id_commande) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)
```

```
TABLESPACE pg_default;
```

```
ALTER TABLE public.com_stock
    OWNER to postgres;
```

```
-- Table: public.commandes
```

```
-- DROP TABLE public.commandes;
```

```
CREATE TABLE public.commandes
(
    id_commande integer NOT NULL,
```

```

id_utilisateur integer,

status text COLLATE pg_catalog."default",

prix_in double precision,

prix_fin double precision,

id_code integer,

CONSTRAINT commandes_pkey PRIMARY KEY (id_commande),

CONSTRAINT ddv FOREIGN KEY (id_code)

    REFERENCES public.code_promos (id_code) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

    NOT VALID,

CONSTRAINT id_utilisateur FOREIGN KEY (id_utilisateur)

    REFERENCES public.utilisateurs (id_utilisateur) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION
)

```

```

TABLESPACE pg_default;

```

```

ALTER TABLE public.commandes

    OWNER to postgres;

```

```

-- Table: public.photos

```

```

-- DROP TABLE public.photos;

```

```

CREATE TABLE public.photos

(
    id_photo integer NOT NULL,

    id_stock integer,

    photo text COLLATE pg_catalog."default",

    CONSTRAINT photos_pkey PRIMARY KEY (id_photo),

    CONSTRAINT photos_id_stock_fkey FOREIGN KEY (id_stock)

        REFERENCES public.stocks (id_stock) MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION
)

```

```

TABLESPACE pg_default;

```

```

ALTER TABLE public.photos

    OWNER to postgres;

```

```

-- Table: public.soldes

```

```

-- DROP TABLE public.soldes;

```

```

CREATE TABLE public.soldes

(
    id_solde integer NOT NULL,

```

```

debut date,

fin date,

nom text COLLATE pg_catalog."default",

id_categorie integer,

pourcentage integer,

CONSTRAINT soldes_pkey PRIMARY KEY (id_solde),

CONSTRAINT soldes_id_categorie_fkey FOREIGN KEY (id_categorie)

    REFERENCES public.categories (id_categorie) MATCH SIMPLE

    ON UPDATE NO ACTION

    ON DELETE NO ACTION

)

TABLESPACE pg_default;

ALTER TABLE public.soldes

    OWNER to postgres;

```

```

-- Table: public.stocks

-- DROP TABLE public.stocks;

CREATE TABLE public.stocks

(

    id_stock integer NOT NULL,

    id_article integer,

    taille text COLLATE pg_catalog."default",

    couleur text COLLATE pg_catalog."default",

    prix_in double precision,

    prix_fin double precision,

    solde integer,

    quantite bigint,

    CONSTRAINT stocks_pkey PRIMARY KEY (id_stock),

    CONSTRAINT stocks_id_article_fkey FOREIGN KEY (id_article)

        REFERENCES public.articles (id_article) MATCH SIMPLE

        ON UPDATE NO ACTION

        ON DELETE NO ACTION

)

TABLESPACE pg_default;

ALTER TABLE public.stocks

    OWNER to postgres;

```

```
-- Table: public.style

-- DROP TABLE public.style;

CREATE TABLE public.style
(
    logo text COLLATE pg_catalog."default",
    font text COLLATE pg_catalog."default",
    couleur1 text COLLATE pg_catalog."default",
    couleur2 text COLLATE pg_catalog."default",
    couleur3 text COLLATE pg_catalog."default"
)

TABLESPACE pg_default;

ALTER TABLE public.style
    OWNER to postgres;


-- Table: public.utilisateurs

-- DROP TABLE public.utilisateurs;

CREATE TABLE public.utilisateurs
(
    id_utilisateur integer NOT NULL,
    nom text COLLATE pg_catalog."default",
    prenom text COLLATE pg_catalog."default",
    email text COLLATE pg_catalog."default",
    password text COLLATE pg_catalog."default",
    is_invite boolean,
    is_admin boolean,
    is_vendeur boolean,
    CONSTRAINT utilisateurs_pkey PRIMARY KEY (id_utilisateur)
)

TABLESPACE pg_default;

ALTER TABLE public.utilisateurs
    OWNER to postgres;
```