

**21 Janvier 2025**

|  |
| --- |
| Université Savoie Mont Blanc |
| Rendu final |
|  |

|  |
| --- |
|  |

**SOMMAIRE**

**Année Universitaire 2024 – 2025**

**Réalisé Par :**

* AMARAL Nathan
* BEKHOUCHE Mohamed Amir
* CETINKAYA Melih
* MASHOVETS Nazar
* TINASTEPE Feyza

**Encadré Par :**

* M° V. COUTURIER



Introduction

[1 - MPD 3](#_Toc188293404)

[Initial 3](#_Toc188293405)

[Final 4](#_Toc188293406)

[2 - Indexation 6](#_Toc188293407)

[Indexe proposés : 6](#_Toc188293408)

[Clés étrangères : 6](#_Toc188293409)

[Clés alternatives : 8](#_Toc188293410)

[Volume des tables et des index : 19](#_Toc188293411)

[3 - Procédures Stockées 20](#_Toc188293412)

[4 - Triggers 20](#_Toc188293413)

[*-- 1. Empêcher les commandes de plus de 500€* 20](#_Toc188293414)

[5 - Script SQL COMPLET 24](#_Toc188293415)

[6 - Optimisation pour Power BI 25](#_Toc188293416)

[Fichier Power BI 25](#_Toc188293417)

[Création de vues SQL 26](#_Toc188293418)

*D*

*ans le cadre du développement de notre projet au sein du module* ***Conception et Développement d'Application****, notre équipe de cinq membres a été chargée de réaliser une solution répondant aux besoins identifiés pour le sujet* ***Uber.*** *Ce livrable a pour objectif*

# MPD

## Initial

**Une image contenant texte, diagramme, Plan, ligne

Description générée automatiquement**

## Final

Une image contenant texte, capture d’écran, diagramme, ligne

Description générée automatiquement

Modification effectuer :

# Indexation

*Aucun index explicite n'a été trouvé dans le script SQL. Nous avons donc ajouté des index sur des champs stratégiques utilisés dans les recherches.*

## Indexe proposés :

*Nous avons fait des index pour toutes les clés étrangères de chaque table dans la base.*

### Clés étrangères :

CREATE INDEX IDX\_ADRESSE\_IDVILLE ON ADRESSE (IDVILLE);

CREATE INDEX IDX\_APPARTIENT\_2\_IDCLIENT ON APPARTIENT\_2 (IDCLIENT);

CREATE INDEX IDX\_A\_3\_IDCATEGORIE ON A\_3 (IDCATEGORIE);

CREATE INDEX IDX\_A\_COMME\_TYPE\_IDPRESTATION ON A\_COMME\_TYPE (IDPRESTATION);

CREATE INDEX IDX\_CLIENT\_IDENTREPRISE ON CLIENT (IDENTREPRISE);

CREATE INDEX IDX\_CLIENT\_IDADRESSE ON CLIENT (IDADRESSE);

CREATE INDEX IDX\_OTP\_IDCLIENT ON OTP (IDCLIENT);

CREATE INDEX IDX\_CODE\_POSTAL\_IDPAYS ON CODE\_POSTAL (IDPAYS);

CREATE INDEX IDX\_COMMANDE\_IDPANIER ON COMMANDE (IDPANIER);

CREATE INDEX IDX\_COMMANDE\_IDLIVREUR ON COMMANDE (IDLIVREUR);

CREATE INDEX IDX\_COMMANDE\_IDCB ON COMMANDE (IDCB);

CREATE INDEX IDX\_COMMANDE\_IDADRESSE ON COMMANDE (IDADRESSE);

CREATE INDEX IDX\_CONTIENT\_2\_IDPRODUIT ON CONTIENT\_2 (IDPRODUIT);

CREATE INDEX IDX\_CONTIENT\_2\_IDETABLISSEMENT ON CONTIENT\_2 (IDETABLISSEMENT);

CREATE INDEX IDX\_COURSE\_IDCOURSIER ON COURSE (IDCOURSIER);

CREATE INDEX IDX\_COURSE\_IDCB ON COURSE (IDCB);

CREATE INDEX IDX\_COURSE\_IDADRESSE ON COURSE (IDADRESSE);

CREATE INDEX IDX\_COURSE\_IDRESERVATION ON COURSE (IDRESERVATION);

CREATE INDEX IDX\_COURSE\_IDPRESTATION ON COURSE (IDPRESTATION);

CREATE INDEX IDX\_COURSE\_ADR\_IDADRESSE ON COURSE (ADR\_IDADRESSE);

CREATE INDEX IDX\_COURSIER\_IDENTREPRISE ON COURSIER (IDENTREPRISE);

CREATE INDEX IDX\_COURSIER\_IDADRESSE ON COURSIER (IDADRESSE);

CREATE INDEX IDX\_DEPARTEMENT\_IDPAYS ON DEPARTEMENT (IDPAYS);

CREATE INDEX IDX\_ENTREPRISE\_IDADRESSE ON ENTREPRISE (IDADRESSE);

CREATE INDEX IDX\_ENTRETIEN\_IDCOURSIER ON ENTRETIEN (IDCOURSIER);

CREATE INDEX IDX\_EST\_SITUE\_A\_2\_IDETABLISSEMENT ON EST\_SITUE\_A\_2 (IDETABLISSEMENT);

CREATE INDEX IDX\_ETABLISSEMENT\_IDADRESSE ON ETABLISSEMENT (IDADRESSE);

CREATE INDEX IDX\_ETABLISSEMENT\_IDRESTAURATEUR ON ETABLISSEMENT (IDRESTAURATEUR);

CREATE INDEX IDX\_HORAIRES\_IDETABLISSEMENT ON HORAIRES (IDETABLISSEMENT);

CREATE INDEX IDX\_HORAIRES\_COURSIER\_IDCOURSIER ON HORAIRES\_COURSIER (IDCOURSIER);

CREATE INDEX IDX\_HORAIRES\_LIVREUR\_IDLIVREUR ON HORAIRES\_LIVREUR (IDLIVREUR);

CREATE INDEX IDX\_FACTURE\_IDRESERVATION ON FACTURE (IDRESERVATION);

CREATE INDEX IDX\_FACTURE\_IDCOMMANDE ON FACTURE (IDCOMMANDE);

CREATE INDEX IDX\_FACTURE\_IDPAYS ON FACTURE (IDPAYS);

CREATE INDEX IDX\_FACTURE\_IDCLIENT ON FACTURE (IDCLIENT);

CREATE INDEX IDX\_LIEU\_FAVORI\_IDCLIENT ON LIEU\_FAVORI (IDCLIENT);

CREATE INDEX IDX\_LIEU\_FAVORI\_IDADRESSE ON LIEU\_FAVORI (IDADRESSE);

CREATE INDEX IDX\_LIVREUR\_IDENTREPRISE ON LIVREUR (IDENTREPRISE);

CREATE INDEX IDX\_LIVREUR\_IDADRESSE ON LIVREUR (IDADRESSE);

CREATE INDEX IDX\_PANIER\_IDCLIENT ON PANIER (IDPANIER);

CREATE INDEX IDX\_PLANNING\_RESERVATION\_IDCLIENT ON PLANNING\_RESERVATION (IDCLIENT);

CREATE INDEX IDX\_REGLEMENT\_SALAIRE\_IDCOURSIER ON REGLEMENT\_SALAIRE (IDCOURSIER);

CREATE INDEX IDX\_RESERVATION\_IDCLIENT ON RESERVATION (IDCLIENT);

CREATE INDEX IDX\_RESERVATION\_IDPLANNING ON RESERVATION (IDPLANNING);

CREATE INDEX IDX\_RESERVATION\_IDVELO ON RESERVATION (IDVELO);

CREATE INDEX IDX\_GESTION\_ETABLISSEMENT\_IDETABLISSEMENT ON GESTION\_ETABLISSEMENT (IDETABLISSEMENT);

CREATE INDEX IDX\_GESTION\_ETABLISSEMENT\_IDRESPONSABLE ON GESTION\_ETABLISSEMENT (IDRESPONSABLE);

CREATE INDEX IDX\_A\_COMME\_CATEGORIE\_IDETABLISSEMENT ON A\_COMME\_CATEGORIE (IDETABLISSEMENT);

CREATE INDEX IDX\_VEHICULE\_IDCOURSIER ON VEHICULE (IDCOURSIER);

CREATE INDEX IDX\_VELO\_IDADRESSE ON VELO (IDADRESSE);

CREATE INDEX IDX\_VELO\_RESERVATION\_IDVELO ON VELO\_RESERVATION (IDVELO);

CREATE INDEX IDX\_VILLE\_IDPAYS ON VILLE (IDPAYS);

CREATE INDEX IDX\_VILLE\_IDCODEPOSTAL ON VILLE (IDCODEPOSTAL);

### Clés alternatives :

***--CATEGORIE PRODUIT : Cost 0.00..1.20***

CREATE INDEX IDX\_CATEGORIE\_PRODUIT\_NOMCATEGORIE ON CATEGORIE\_PRODUIT (NOMCATEGORIE);

*La recherche se faisant sur le nom de la catégorie, nous avons donc créer un index nomcategorie.*

*Voici la requête SQL :*

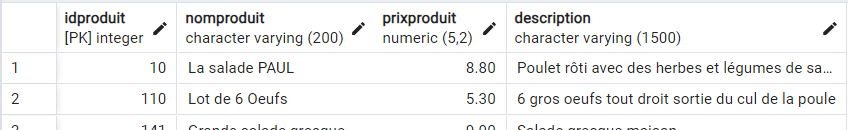
SELECT P.IDPRODUIT, P.NOMPRODUIT, P.PRIXPRODUIT, P.DESCRIPTION

FROM PRODUIT P

JOIN A\_3 A ON P.IDPRODUIT = A.IDPRODUIT

JOIN CATEGORIE\_PRODUIT C ON A.IDCATEGORIE = C.IDCATEGORIE

WHERE C.NOMCATEGORIE = 'Fruits, Légumes et Produits frais';



***--CLIENT :  Cost de 0.00..7.20***

CREATE INDEX IDX\_CLIENT\_NOM\_PRENOM\_USER ON CLIENT (NOMUSER, PRENOMUSER);

*La recherche se faisant sur le nom et prénom du client, nous avons donc créer un index nomuser et prenomuser du client*

*Voici la requête SQL :*

SELECT IDCLIENT, NOMUSER, PRENOMUSER,DATENAISSANCE, EMAILUSER

FROM CLIENT

WHERE PRENOMUSER = 'Emma' AND NOMUSER = 'Leclerc';

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

***--CODE POSTAL :*** *.* ***Cost de 0.00..1.40***

CREATE INDEX IDX\_CODE\_POSTAL\_CODEPOSTAL ON CODE\_POSTAL (CODEPOSTAL);

*La recherche se faisant sur le code postal, nous avons donc créer un index sur codepostal.*

*Voici la requête SQL :*

SELECT CODEPOSTAL, IDPAYS

FROM CODE\_POSTAL WHERE CODEPOSTAL = '75000';

Une image contenant texte, Police, capture d’écran, ligne

Description générée automatiquement

***--COMMANDE : Cost de 0.00..1.70***

CREATE INDEX IDX\_COMMANDE\_STATUT ON COMMANDE (STATUTCOMMANDE);

*La recherche se faisant sur le statut de la commande, nous avons donc créer un index sur statutcommande.*

*Voici la requête SQL :*

SELECT IDCOMMANDE, IDPANIER, IDLIVREUR, IDADRESSE, PRIXCOMMANDE, TEMPSCOMMANDE, STATUTCOMMANDE

FROM COMMANDE

WHERE STATUTCOMMANDE = 'Livrée'; Une image contenant texte, Police, ligne, nombre

Description générée automatiquement

***--COURSE :  Cost de 0.00..1.64***

CREATE INDEX IDX\_COURSE\_STATUTCOURSE ON COURSE (STATUTCOURSE);

*La recherche se faisant sur le statut de la course, nous avons donc créer un index sur statutcourse.*

*Voici la requête SQL :*

SELECT IDCOURSE, IDCB, IDADRESSE, DATECOURSE, STATUTCOURSE, HEURECOURSE, PRIXCOURSE, NOTECOURSE, COMMENTAIRECOURSE

FROM COURSE

WHERE STATUTCOURSE = 'En attente';

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

***--COURSIER :  Cost de 0.00..2.70***

CREATE INDEX IDX\_COURSIER\_NOM\_PRENOM\_USER ON COURSIER (NOMUSER, PRENOMUSER);

*La recherche se faisant sur le nom et prénom du coursier, nous avons donc créer un index sur nomuser et prenomuser du coursier*

*Voici la requête SQL :*

Une image contenant texte, Police, ligne, capture d’écran

Description générée automatiquement

CREATE INDEX IDX\_COURSIER\_DATEDEBUTACTIVITE ON COURSIER (DATEDEBUTACTIVITE);

*La recherche se faisant sur la date de début d’activité du coursier, nous avons donc créer un index sur datedebutactivite.*

*Voici la requête SQL :*

SELECT IDCOURSIER, NOMUSER, PRENOMUSER, DATEDEBUTACTIVITE

FROM COURSIER

WHERE DATEDEBUTACTIVITE >= '2022-01-01'

  AND DATEDEBUTACTIVITE <= '2022-12-31';

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

***--ENTREPRISE : Cost de 0.00..1.40***

CREATE INDEX IDX\_ENTREPRISE\_NOMENTREPRISE ON ENTREPRISE (NOMENTREPRISE);

*La recherche se faisant sur le nom de l’entreprise, nous avons donc créer un index sur le nomentreprise.*

*Voici la requête SQL :*

SELECT \* FROM ENTREPRISE WHERE NOMENTREPRISE = 'Delta Industries';



CREATE INDEX IDX\_ENTREPRISE\_SIRETENTREPRISE ON ENTREPRISE (SIRETENTREPRISE);

*La recherche se faisant sur le numéro de Siret de l’entreprise, nous avons donc créer un index sur siretentreprise.*

*Voici la requête SQL :*

SELECT \* FROM ENTREPRISE WHERE SIRETENTREPRISE LIKE '1%';

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement ***--ENTRETIEN : Cost de 0.00..1.70***

CREATE INDEX IDX\_ENTRETIEN\_DATEENTRETIEN ON ENTRETIEN (DATEENTRETIEN);

*La recherche se faisant sur la date de l’entretien, nous avons donc créer un index sur datenetretien.*

*Voici la requête SQL :*

SELECT IDENTRETIEN, DATEENTRETIEN

FROM ENTRETIEN

WHERE DATEENTRETIEN BETWEEN '2025-01-01' AND '2025-01-10';

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

CREATE INDEX IDX\_ENTRETIEN\_STATUT ON ENTRETIEN (STATUS);

*La recherche se faisant sur le statut de l’entretien, nous avons donc créer un index sur status*

*Voici la requête SQL :*

SELECT IDENTRETIEN, DATEENTRETIEN, STATUS

FROM ENTRETIEN

WHERE STATUS = 'Planifié';

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

CREATE INDEX IDX\_ENTRETIEN\_RESULTAT ON ENTRETIEN (RESULTAT);

*La recherche se faisant sur le résultat de l’entretien, nous avons donc créer un index sur resultat.*

*Voici la requête SQL :*

SELECT IDENTRETIEN, IDCOURSIER, DATEENTRETIEN, STATUS, RESULTAT

FROM ENTRETIEN WHERE STATUS = 'Terminée'

Une image contenant texte, Police, ligne, nombre

Description générée automatiquement

***--ETABLISSEMENT : Cost de 0.00..2.45***

CREATE INDEX IDX\_ETABLISSEMENT\_TYPEETABLISSEMENT ON ETABLISSEMENT (TYPEETABLISSEMENT);

*La recherche se faisant sur le type de l’établissement, nous avons donc créer un index sur typeetablissement.*

*Voici la requête SQL :*

SELECT E.IDETABLISSEMENT, E.NOMETABLISSEMENT, E.DESCRIPTION, E.LIVRAISON, E.AEMPORTER

FROM ETABLISSEMENT E

WHERE E.TYPEETABLISSEMENT = 'Restaurant';

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

CREATE INDEX IDX\_ETABLISSEMENT\_NOMETABLISSEMENT ON ETABLISSEMENT (NOMETABLISSEMENT);

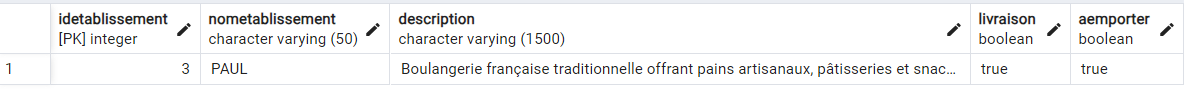
*La recherche se faisant sur le nom de l’établissement, nous avons donc créer un index sur nometablissement.*

*Voici la requête SQL :*

SELECT E.IDETABLISSEMENT, E.NOMETABLISSEMENT, E.DESCRIPTION, E.LIVRAISON, E.AEMPORTER

FROM ETABLISSEMENT E

WHERE E.NOMETABLISSEMENT = 'PAUL';



CREATE INDEX IDX\_ETABLISSEMENT\_LIVRAISON\_AEMPORTER ON ETABLISSEMENT (LIVRAISON, AEMPORTER);

*La recherche se faisant sur les colonnes livraison et aemporter simultanément, nous avons donc créé un index combiné sur ces deux colonnes.*

*Voici la requête SQL :*

SELECT IDETABLISSEMENT, NOMETABLISSEMENT, DESCRIPTION ,LIVRAISON, AEMPORTER

FROM ETABLISSEMENT

WHERE LIVRAISON = TRUE AND AEMPORTER = FALSE;

Une image contenant texte, Police, ligne, nombre

Description générée automatiquement

***--HORAIRES : Cost de 0.00..7.15***

CREATE INDEX IDX\_HORAIRES\_JOURSEMAINE\_HORAIRESOUVERTURE\_HORAIRESFERMETURE ON HORAIRES (JOURSEMAINE, HORAIRESOUVERTURE, HORAIRESFERMETURE);

*La recherche se faisant sur les colonnes joursemaine, horairesouverture et horairesfermeture, nous avons donc créé un index combiné nommé IDX\_HORAIRES\_JOURSEMAINE\_HORAIRESOUVERTURE\_HORAIRESFERMETURE sur ces trois colonnes.*

*Voici la requête SQL :*

SELECT E.IDETABLISSEMENT, E.NOMETABLISSEMENT, E.DESCRIPTION, H.HORAIRESOUVERTURE, H.HORAIRESFERMETURE

FROM ETABLISSEMENT E

JOIN HORAIRES H ON E.IDETABLISSEMENT = H.IDETABLISSEMENT

WHERE H.JOURSEMAINE = 'Mardi'

  AND H.HORAIRESOUVERTURE = '08:00:00'

  AND H.HORAIRESFERMETURE = '22:00:00';

Une image contenant texte, Police, ligne, capture d’écran

Description générée automatiquement

***--HORAIRES COURSIER : Cost de 0.00..9.90***

CREATE INDEX IDX\_HORAIRES\_COURSIER\_JOURSEMAINE\_HEUREDEBUT\_HEUREFIN ON HORAIRES\_COURSIER (JOURSEMAINE, HEUREDEBUT,HEUREFIN);

*La recherche se faisant sur les colonnes joursemaine, heuredebut et heurefin, nous avons donc créé un index combiné nommé IDX\_HORAIRES\_COURSIER\_JOURSEMAINE\_HEUREDEBUT\_HEUREFIN sur ces trois colonnes.*

*Voici la requête SQL :*

SELECT C.IDCOURSIER, C.NOMUSER, C.PRENOMUSER, H.JOURSEMAINE,H.HEUREDEBUT, H.HEUREFIN

FROM COURSIER C

JOIN HORAIRES\_COURSIER H ON C.IDCOURSIER = H.IDHORAIRES\_COURSIER

WHERE H.JOURSEMAINE = 'Mardi'

AND H.HEUREDEBUT = '08:00:00'

AND H.HEUREFIN = '18:00:00'

Une image contenant texte, Police, ligne, nombre

Description générée automatiquement

***--HORAIRES LIVREUR: Cost de 0.00..1.42***

CREATE INDEX IDX\_HORAIRES\_LIVREUR\_JOURSEMAINE\_HEUREDEBUT\_HEUREFIN ON HORAIRES\_LIVREUR (JOURSEMAINE, HEUREDEBUT,HEUREFIN);

*La recherche se faisant sur les colonnes joursemaine, heuredebut et heurefin de la table horaires\_livreur nous avons donc créé un index combiné nommé IDX\_HORAIRES\_COURSIER\_JOURSEMAINE\_HEUREDEBUT\_HEUREFIN sur ces trois colonnes.*

*Voici la requête SQL :*

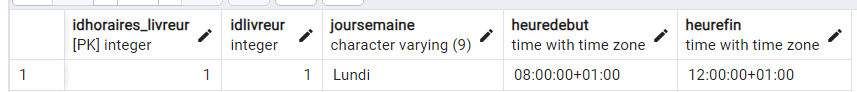
SELECT \*

FROM HORAIRES\_LIVREUR

WHERE JOURSEMAINE = 'Lundi'

AND HEUREDEBUT = '08:00:00'

AND HEUREFIN = '12:00:00';



***--FACTURE: Cost de 0.00..22.20***

CREATE INDEX IDX\_FACTURE\_DATEFACTURE ON FACTURE (DATEFACTURE);

*La recherche se faisant sur la date de la facture de la course, nous avons donc créé un index sur datefacture*

*Voici la requête SQL :*

SELECT \*

FROM FACTURE\_COURSE

WHERE DATEFACTURE BETWEEN '2024-01-01' AND '2024-12-31';

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

***--LIEU\_FAVORI : Cost de 0.00..1.20***

CREATE INDEX IDX\_LIEU\_FAVORI\_NOMLIEU ON LIEU\_FAVORI (NOMLIEU);

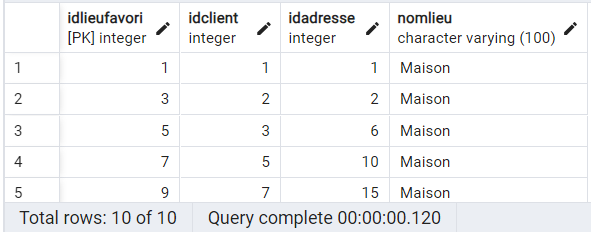
*La recherche se faisant sur le nom du lieu favoris du client, nous avons donc créé un index sur nomlieu de la table lieu\_favors.*

*Voici la requête SQL  :*

SELECT \*

FROM LIEU\_FAVORI

WHERE NOMLIEU = 'Maison';



***--LIVREUR : Cost de 0.00..1.21***

CREATE INDEX IDX\_LIVREUR\_NOM\_PRENOM ON LIVREUR (NOMUSER, PRENOMUSER);

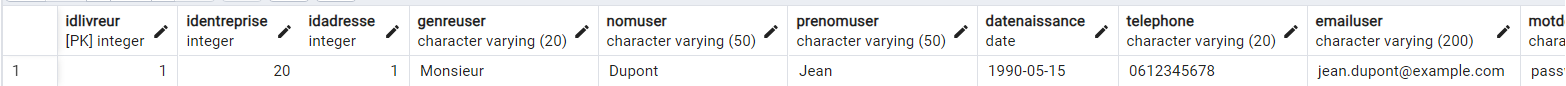
*La recherche se faisant sur le nom et prénom du livreur, nous avons donc créé un index sur nomuser et prenomuser de la table livreur.*

*Voici la requête SQL :*

SELECT \*

FROM LIVREUR

WHERE NOMUSER = 'Dupont' AND PRENOMUSER = 'Jean';



***--PAYS : Cost de 0.00..1.20***

CREATE INDEX IDX\_PAYS\_NOMPAYS ON PAYS (NOMPAYS);

*La recherche se faisant sur le nom du pays, nous avons donc créé un index sur nompays*

*Voici la requête SQL :*

SELECT \* FROM PAYS WHERE NOMPAYS = 'France'

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

***--PRODUIT : Cost de 0.00..7.55***

CREATE INDEX IDX\_PRODUIT\_NOMPRODUIT ON PRODUIT (NOMPRODUIT);

*La recherche se faisant sur le nom du produit, nous avons donc créé un index sur nomproduit*

*Voici la requête SQL :*

SELECT \* FROM PRODUIT WHERE NOMPRODUIT = 'Empanada carne'

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

***--RESERVATION : Cost de 0.00..2.14***

CREATE INDEX IDX\_RESERVATION\_DATERESERVATION\_HEURERESERVATION ON RESERVATION (DATERESERVATION, HEURERESERVATION);

*La recherche se faisant sur les colonnes datereservation et heurereservation nous avons donc créé un index combiné nommé IDX\_RESERVATION\_DATE\_HEURE\_RESERVATION sur ces trois colonnes.*

*Voici la requête SQL :*

SELECT \*

FROM RESERVATION

WHERE DATERESERVATION = '2024-11-24'

  AND HEURERESERVATION BETWEEN '09:00:00' AND ' 14:00:00';

Une image contenant texte, Police, nombre, ligne

Description générée automatiquement

***--RESPONSABLE\_ENSEIGNE : Cost de 0.00..1.45***

CREATE INDEX IDX\_RESPONSABLE\_ENSEIGNE\_NOM\_PRENOM ON RESPONSABLE\_ENSEIGNE (NOMUSER, PRENOMUSER);

*La recherche se faisant sur le nom et prénom du responsable d’enseigne, nous avons donc créé un index sur nomuser et prenomuser de la table responsable\_enseigne.*

*Voici la requête SQL :*

SELECT \*

FROM RESPONSABLE\_ENSEIGNE

WHERE NOMUSER = 'Yılmaz' AND PRENOMUSER = 'Ahmet';



***--RESTAURATEUR : Cost de 0.00..1.45***

CREATE INDEX IDX\_RESTAURATEUR\_NOM\_PRENOM ON RESTAURATEUR (NOMUSER, PRENOMUSER);

*La recherche se faisant sur le nom et prénom du restaurateur, nous avons donc créé un index sur nomuser et prenomuser de la table restaurateur.*

*Voici la requête SQL :*

SELECT \*

FROM RESTAURATEUR

WHERE NOMUSER = 'Durand' AND PRENOMUSER = 'Michel';



***--TYPE\_PRESTATION : Cost de 0.00..1.07***

CREATE INDEX IDX\_TYPE\_PRESTATION\_LIBELLEPRESTATION ON TYPE\_PRESTATION (LIBELLEPRESTATION);

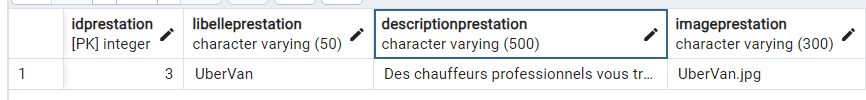
*La recherche se faisant sur le libellé du type de prestation, nous avons donc créé un index libelleprestation*

*Voici la requête SQL :*

SELECT \*

FROM TYPE\_PRESTATION

WHERE LIBELLEPRESTATION = 'UberVan'

 ***--CATEGORIE\_PRESTATION : Cost de 0.00..1.46***

CREATE INDEX IDX\_CATEGORIE\_PRESTATION\_LIBELLECATEGORIEPRESTATION ON CATEGORIE\_PRESTATION (LIBELLECATEGORIEPRESTATION);

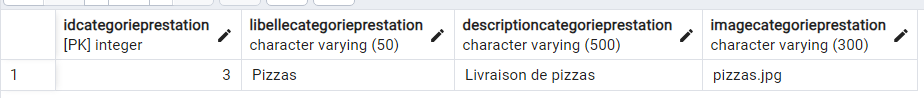
*La recherche se faisant sur le libellé du type de catégorie de prestation, nous avons donc créé un index libellecategorieprestation*

*Voici la requête SQL :*

SELECT \*

FROM CATEGORIE\_PRESTATION

WHERE LIBELLECATEGORIEPRESTATION = 'Pizzas'

  ***--VEHICULE : Cost de 0.00..1.21***

CREATE INDEX IDX\_VEHICULE\_STATUT ON VEHICULE (ESTELECTRIQUE, ESTCONFORTABLE, ESTRECENT, ESTLUXUEUX);

*La recherche se faisant sur les colonnes estelectrique, estconfortable, estrecent et estluxueux, nous avons donc créé un index combiné nommé IDX\_VEHICULE\_STATUT sur ces quatre colonnes.*

*Voici la requête SQL :*

SELECT IDVEHICULE, IMMATRICULATION, MARQUE, MODELE, COULEUR,

    CASE WHEN ESTELECTRIQUE THEN 'Électrique' ELSE 'Non Électrique' END AS Statut\_Electrique,

    CASE WHEN ESTCONFORTABLE THEN 'Confortable' ELSE 'Non Confortable' END AS Statut\_Confort,

    CASE WHEN ESTRECENT THEN 'Recent' ELSE 'Non-Recent' END AS Statut\_Recent,

    CASE WHEN ESTLUXUEUX THEN 'Luxueux' ELSE 'Non Luxueux' END AS Statut\_Luxe

FROM VEHICULE

Une image contenant texte, Police, nombre, ligne

Description générée automatiquement

***--VILLE : Cost de 0.00..1.40***

CREATE INDEX IDX\_VILLE\_NOMVILLE ON VILLE (NOMVILLE);

*La recherche se faisant sur le nom de la ville, nous avons donc créé un index nomville.*

*Voici la requête SQL :*

SELECT \*

FROM VILLE

WHERE NOMVILLE = 'Paris'

Une image contenant texte, Police, ligne, nombre

Description générée automatiquement

Volume des tables et des index :

Sur 243 attributs, il y a 100 index : 40% de la base sont indexé.

# Procédures Stockées

Nous avons ajouté des procédures pour automatiser certaines opérations.

**Exemple - Enregistrer une commande** :

CREATE OR REPLACE FUNCTION inserer\_commande(p\_idclient INT, p\_date TIMESTAMP)

RETURNS VOID AS $$

BEGIN

    INSERT INTO COMMANDE (IDCLIENT, DATE\_COMMANDE) VALUES (p\_idclient, p\_date);

END;

$$ LANGUAGE plpgsql;

# Triggers

### -- 1. Empêcher les commandes de plus de 500€

CREATE OR REPLACE FUNCTION check\_commande\_prix()

RETURNS TRIGGER AS $$

BEGIN

    IF NEW.PRIXCOMMANDE > 500 THEN

        RAISE EXCEPTION 'Le montant d''une commande ne peut pas dépasser 500€';

    END IF;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg\_check\_commande\_prix

BEFORE INSERT OR UPDATE ON COMMANDE

FOR EACH ROW

EXECUTE FUNCTION check\_commande\_prix();

*--Le but de ce trigger est d’éviter qu’une commande ait un montant supérieur à 500€, pour garantir une politique de prix raisonnable et éviter des erreurs de saisie ou de facturation.*

-- 2. Vérifier que les coursiers et livreurs ont au moins 18 ans

CREATE OR REPLACE FUNCTION check\_age()

RETURNS TRIGGER AS $$

BEGIN

    IF NEW.DATENAISSANCE > CURRENT\_DATE - INTERVAL '18 years' THEN

        RAISE EXCEPTION 'Les coursiers et livreurs doivent avoir au moins 18 ans';

    END IF;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg\_check\_age\_coursier

BEFORE INSERT OR UPDATE ON COURSIER

FOR EACH ROW

EXECUTE FUNCTION check\_age();

CREATE TRIGGER trg\_check\_age\_livreur

BEFORE INSERT OR UPDATE ON LIVREUR

FOR EACH ROW

EXECUTE FUNCTION check\_age();

*--Le but de ce trigger est de s’assurer que les coursiers et livreurs sont majeurs donc respecter l’âge légal qui est 18 ans ou plus.*

-- 3. Valider le paiement des courses et commandes

CREATE OR REPLACE FUNCTION check\_payment()

RETURNS TRIGGER AS $$

BEGIN

    IF NEW.STATUTCOMMANDE = 'Paiement validé' AND NEW.IDCB IS NULL THEN

        RAISE EXCEPTION 'Une commande validée doit avoir un moyen de paiement';

    END IF;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg\_check\_payment\_commande

BEFORE INSERT OR UPDATE ON COMMANDE

FOR EACH ROW

EXECUTE FUNCTION check\_payment();

*--Le but de ce trigger est de vérifier que toute les commande marqué comme « Paiement validé » dispose bien d’un moyen de paiement enregistré.*

-- 4. Mise à jour de la note moyenne du coursier ou livreur après évaluation

CREATE OR REPLACE FUNCTION update\_avg\_rating()

RETURNS TRIGGER AS $$

BEGIN

    IF NEW.NOTECOURSE IS NOT NULL THEN

        UPDATE COURSIER

        SET NOTEMOYENNE = (

            SELECT AVG(NOTECOURSE) FROM COURSE WHERE IDCOURSIER = NEW.IDCOURSIER AND NOTECOURSE IS NOT NULL

        )

        WHERE IDCOURSIER = NEW.IDCOURSIER;

    END IF;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg\_update\_avg\_rating

AFTER INSERT OR UPDATE ON COURSE

FOR EACH ROW

EXECUTE FUNCTION update\_avg\_rating();

*--Le but de ce trigger est de mettre à jour automatiquement la note moyenne d’un coursier lorsqu’une nouvelle note est ajouté.*

-- 5. Vérifier qu'un coursier ne prenne pas deux courses en même temps

CREATE OR REPLACE FUNCTION check\_coursier\_availability()

RETURNS TRIGGER AS $$

BEGIN

    IF (SELECT COUNT(\*) FROM COURSE WHERE IDCOURSIER = NEW.IDCOURSIER AND STATUTCOURSE = 'En cours') > 0 THEN

        RAISE EXCEPTION 'Un coursier ne peut pas prendre deux courses en même temps';

    END IF;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg\_check\_coursier\_availability

BEFORE INSERT OR UPDATE ON COURSE

FOR EACH ROW

EXECUTE FUNCTION check\_coursier\_availability();

*--Le but de ce trigger est d’empêcher qu’un coursier accepte une nouvelle course alors qu’il en a déjà une en cours..*

-- 6. Empêcher une réservation de vélo si le vélo n'est pas disponible

CREATE OR REPLACE FUNCTION check\_velo\_disponibilite()

RETURNS TRIGGER AS $$

BEGIN

    IF (SELECT ESTDISPONIBLE FROM VELO WHERE IDVELO = NEW.IDVELO) = FALSE THEN

        RAISE EXCEPTION 'Ce vélo est déjà réservé ou indisponible';

    END IF;

    RETURN NEW;

END;

$$ LANGUAGE plpgsql;

CREATE TRIGGER trg\_check\_velo\_disponibilite

BEFORE INSERT ON VELO\_RESERVATION

FOR EACH ROW

EXECUTE FUNCTION check\_velo\_disponibilite();

*--Le but de ce trigger est de vérifier qu’un vélo est disponible avant d’accepter une nouvelle réservation.*

# Conclusion