DELIVERABLE 2:

1)Updated version of the ER Diagram :

We deleted the relationship “Gere” between the entities **ChefDeProjet** and **Facture** and we added a relationship “Possede” between the entities **Entite** and **Facture** for the following reason: **Entite** and **Facture** have the same **ChefDeProjet.**

2) Schemas:

Commande (numeroCommande: string, service: string, typeAchatPO: string, uniteOperationelle: string, devise: string, montantCommande: float, montantReceptionne: float, acheteur: string, codeAP: integer, nom: string)

Fournisseur (codeAP: integer, nomFournisseur: string, siteFournisseur: string)

Facture (identifiantGED: string, numeroDeFacture: string, montantFacture: float, intervenant: string, nombreDeJours: integer, contractAdmin: string, typeDeBlockage: string, numeroCommande: string, nomEntite: string)

ChefDeProjet (nom: string, nomEntite: string)

Entite (nomEntite: string)

FDS :

numeroCommande -> service, typeAchatPO, uniteOperationelle, devise, montantCommande, montantReceptionne, acheteur, codeAP, nom.

codeAP -> nomFournisseur, siteFournisseur.

identifiantGED -> numeroDeFacture, montantFacture, intervenant, nombreDeJours, contractAdmin, typeDeBlockage, echeance, numeroCommande, nomEntite.

nom -> nomEntite.

BCNF :

For a relation to be in BCNF, every left hand side attribute of the FD should be a superkey or the FD is already trivial.

In the **Commande** relation, the primary key is numeroCommande, which uniquely determines all other attributes in the relation. Therefore, it is in BCNF.

In the **Fournisseur** relation, the primary key is codeAP, which uniquely determines both nomFournisseur and siteFournisseur. Therefore, it is in BCNF.

In the **Facture** relation, the primary key is identifiantGED, which uniquely determines all other attributes in the relation. Therefore, it is in BCNF.

In the **ChefDeProjet** relation, the primary key is nom, which uniquely determines nomEntite. Therefore, it is in BCNF.

In the **Entite** relation, the primary key is nomEntite, which uniquely determines all other attributes in the relation. Therefore, it is in BCNF.

Overall, the schemas are already in BCNF.

3) Minimal cover set:

Minimal cover : { numeroCommande -> service, numeroCommande -> typeAchatPO, numeroCommande -> uniteOperationelle, numeroCommande -> devise, numeroCommande -> numeroCommande -> montantCommande, numeroCommande -> montantReceptionne, numeroCommande -> acheteur, numeroCommande -> codeAP, numeroCommande -> nom,

identifiantGED -> numeroDeFacture, identifiantGED -> montantFacture, identifiantGED -> intervenant, identifiantGED -> nombreDeJours, identifiantGED -> contractAdmin, identifiantGED -> typeDeBlockage, identifiantGED -> echeance, identifiantGED -> numeroCommande, identifiantGED -> nomEntite,

nom -> nomEntite}.

Explanation:

In **Commande**, numeroCommande is the only candidate key and determines all non-key attributes. The functional dependency codeAP -> { nomFournisseur, siteFournisseur } ensures that each codeAP value is associated with a unique pair of nomFournisseur and siteFournisseur values.

In **Fournisseur**, codeAP is the only candidate key and determines all attributes.

In **Facture**, identifiantGED is the only candidate key and determines all non-key attributes. The foreign key numeroCommande is also included to enforce referential integrity. The functional dependency nom -> nomEntite ensures that each nom value is associated with a unique nomEntite value.

In **ChefDeProjet**, nom is the only candidate key and determines nomEntite.

In **Entite**, nomEntite is the only candidate key and does not determine any other attribute.

4) SQL Tables Creation:

CREATE TABLE Commande (

numeroCommande VARCHAR(255) PRIMARY KEY NOT NULL,

service VARCHAR(255) NOT NULL,

typeAchatPO VARCHAR(255) NOT NULL,

uniteOperationelle VARCHAR(255) NOT NULL,

devise VARCHAR(255) NOT NULL,

montantCommande FLOAT NOT NULL,

montantReceptionne FLOAT NOT NULL,

acheteur VARCHAR(255) NOT NULL,

codeAP INTEGER NOT NULL,

nom VARCHAR(255) NOT NULL,

FOREIGN KEY (codeAP) REFERENCES Fournisseur(codeAP),

FOREIGN KEY (nom) REFERENCES ChefDeProjet (nom)

);

CREATE TABLE Fournisseur (

codeAP INTEGER PRIMARY KEY NOT NULL,

nomFournisseur VARCHAR(255) NOT NULL,

siteFournisseur VARCHAR(255) NOT NULL

);

CREATE TABLE Facture (

identifiantGED VARCHAR(255) PRIMARY KEY NOT NULL,

numeroDeFacture VARCHAR(255) NOT NULL,

montantFacture FLOAT NOT NULL,

intervenant VARCHAR(255) NOT NULL,

nombreDeJours INTEGER NOT NULL,

contractAdmin VARCHAR(255) NOT NULL,

typeDeBlockage VARCHAR(255) NOT NULL,

numeroCommande VARCHAR(255) NOT NULL,

nomEntite VARCHAR(255) NOT NULL,

FOREIGN KEY (numeroCommande) REFERENCES Commande(numeroCommande),

FOREIGN KEY (nomEntite) REFERENCES Entite(nomEntite)

);

CREATE TABLE ChefDeProjet (

nom VARCHAR(255) PRIMARY KEY NOT NULL,

nomEntite VARCHAR(255) NOT NULL,

FOREIGN KEY (nomEntite) REFERENCES Entite(nomEntite)

);

CREATE TABLE Entite (

nomEntite VARCHAR(255) PRIMARY KEY NOT NULL

);