



Université Cote d’Azur / FDS

UE3 - Immersion BD et SQL Oracle

Projet : Gestion d’un cabinet médical

Etudiants :

- **PIERRE** Bob Charlemagne
- **DOUILLY** Rodely
- **MILORME** Pierre Rubens
- **SURLIN** Djimy

Professeur : WEDTER Jérôme

9 mars 2024

Table des matières

1	Projet MCD MERISE, SQL2, PLSQL	2
1.1	Choix du sujet	2
1.2	Spécification, Analyse et Conception	2
1.2.1	Description du sujet	2
1.2.2	Description textuelle des requêtes de mise à jour	2
1.2.3	Description textuelle des requêtes de suppression . . .	4
1.2.4	Description textuelle des requêtes de consultation . . .	4
1.3	Dictionnaire de données MERISE	8
1.4	Description textuelle des associations	8
1.5	Définition du Modèle Entité-Association MERISE	9
1.6	La définition du modèle logique de Données ou schéma relationnel	9
1.7	Spécification des traitements avec des packages PLSQL	9
1.8	Spécification des triggers	9
1.8.1	Trigger TG_CTRL_RENDEZ_VOUS_DOUBLE	10
1.8.2	Trigger TG_HISTORIQUE_FACTURE	10

1 Projet MCD MERISE, SQL2, PLSQL

1.1 Choix du sujet

Système de Gestion d'un cabinet médical.

1.2 Spécification, Analyse et Conception

1.2.1 Description du sujet

Cette application pour un cabinet médical permet la gestion complète des patients, des médecins, des rendez-vous, des consultations, des prescriptions, des examens et de la facturation. Les patients peuvent être enregistrés avec leurs détails personnels, tandis que les médecins sont répertoriés avec leur spécialité respective. Les rendez-vous entre patients et médecins sont programmés, enregistrés dans la table RENDEZ-VOUS. Chaque consultation est consignée dans la table CONSULTATION, associée à un patient, un médecin et une facture. Les prescriptions médicales sont enregistrées dans la table PRESCRIPTION, liées à la consultation correspondante. Les détails des examens sont stockés dans la table EXAMEN, également liés à la consultation. Chaque consultation génère une facture, enregistrée dans la table FACTURE avec le montant total à payer. Des contraintes de clé étrangère garantissent l'intégrité des données et les relations entre les tables.

En résumé, cette application fournit un système complet pour gérer les opérations quotidiennes d'un cabinet médical, optimisant le suivi des patients et la gestion des consultations et des facturations.

1.2.2 Description textuelle des requêtes de mise à jour

— 2 requêtes impliquant 1 table

1. **REQUÊTE 1** : Cette requête modifie la date de naissance du

patient dont l'adresse mail est thomas.leclerc@email.com. La nouvelle date de naissance sera 22-AUG-1986. Seules les lignes correspondant à l'adresse mail spécifiée seront affectées.

2. **REQUÊTE 2** : Cette requête modifie l'adresse du patient dont l'identifiant est 1. La nouvelle adresse sera '90, DELMAS 75'. Seule la ligne correspondant à l'identifiant spécifié sera affectée.

— **2 requêtes impliquant 2 tables**

1. **REQUÊTE 1** : Cette requête SQL permet de modifier la date de rendez-vous des patients dont le rendez-vous est prévu entre le 14-FEB-2024 et le 18-FEB-2024 dans la table virtuelle issue de la jointure entre PATIENT et RENDEZ-VOUS. La nouvelle date de rendez-vous sera le 01-MAR-24.
2. **REQUÊTE 2** : Cette requête parcourt toutes les factures dont le montant total est inférieur à 200 et pour lesquelles un patient correspondant existe. Pour chacune de ces factures, elle augmente le montant total de 10%.

— **2 requêtes impliquant plus de 2 tables**

1. **REQUÊTE 1** : Cette requête met à jour la colonne DETAILS_PRESCRIPTION de la table PRESCRIPTION pour les enregistrements associés à des consultations de patients ayant un identifiant de patient égal à 1 et où la date de consultation est le 5 février 2024. La nouvelle valeur de la colonne DETAILS_PRESCRIPTION sera 'Zinoboost'.
2. **REQUÊTE 2** : Cette requête met à jour la colonne DETAILS_EXAMEN de la table EXAMEN pour les enregistrements associés à des consultations de patients ayant un identifiant de patient égal à 1 et où la date de consultation est le 5 février 2024. La nouvelle valeur de la colonne DETAILS_EXAMEN

sera 'HAC1'.

1.2.3 Description textuelle des requêtes de suppression

— 2 requêtes impliquant 1 table

1. **REQUÊTE 1 :** Cette requête supprime toutes les lignes de la table EXAMEN où la valeur de la colonne DATE_EXAMEN est égale à la date du 5 février 2024.
2. **REQUÊTE 2 :** Cette requête supprime toutes les lignes de la table PRESCRIPTION où la valeur de la colonne DATE_PRESCRIPTION est égale à la date du 5 février 2024.

— 2 requêtes impliquant 2 tables

1. **REQUÊTE 1 :** Cette requête supprime tous les rendez-vous des patients dont l'adresse e-mail est 'thomas.leclerc@email.com'.
2. **REQUÊTE 2 :** Cette requête supprime tous les rendez-vous du patient dont l'identifiant est 3.

— 2 requêtes impliquant plus de 2 tables

1. **REQUÊTE 1 :** Cette requête supprime tous les examens associés à une consultation qui a eu lieu le 5 février 2024 et qui est liée à un patient dont l'identifiant est 1.
2. **REQUÊTE 2 :** Cette requête supprime tous les examens associés à une consultation qui a eu lieu le 6 février 2024 et qui est liée à un patient dont l'identifiant est 2.

1.2.4 Description textuelle des requêtes de consultation

- #### — 5 requêtes impliquant 1 table dont 1 avec un GROUP BY et 1 avec un ORDER By

1. **REQUÊTE 1 :** Cette requête récupère toutes les informations (toutes les colonnes) stockées dans la table PATIENT, ce qui signifie qu'elle retournera toutes les lignes de cette table.
 2. **REQUÊTE 2 :** Cette requête récupère toutes les informations (toutes les colonnes) stockées dans la table CONSULTATION, ce qui signifie qu'elle retournera toutes les lignes de cette table.
 3. **REQUÊTE 3 :** Cette requête compte le nombre de lignes dans chaque groupe de données regroupées selon les valeurs uniques de la colonne DETAILS_EXAMEN. Chaque groupe dans le résultat final représente une valeur unique de DETAILS_EXAMEN, et le nombre de lignes dans chaque groupe est renvoyé.
 4. **REQUÊTE 4 :** Cette requête compte le nombre de lignes dans chaque groupe de données regroupées selon les valeurs uniques de la colonne DETAILS_EXAMEN, et les présente dans l'ordre croissant basé sur ces valeurs..
 5. **REQUÊTE 5 :** cette requête renvoie le nombre total de factures, la somme totale des montants de toutes les factures pour chaque patient, regroupées par ID_PATIENT_, et les présente dans l'ordre croissant des ID_PATIENT_.
- **5 requêtes impliquant 2 tables avec jointures internes dont 1 externe + 1 GROUP BY + 1 tri**
1. **REQUÊTE 1 :** Cette requête retourne toutes les colonnes des consultations, jointes avec les informations des patients correspondants où les ID_PATIENT_ sont égaux dans les deux tables. Cela permet d'obtenir des données combinées sur les consultations et les patients dans une seule table résultante.
 2. **REQUÊTE 2 :** Cette requête retourne toutes les colonnes des

factures, jointes avec les informations des patients correspondants où les `ID_PATIENT` sont égaux dans les deux tables. Cela permet d'obtenir des données combinées sur les factures et les patients dans une seule table résultante.

3. **REQUÊTE 3 :** Cette requête renvoie toutes les colonnes des consultations, jointes avec les informations des patients correspondants, et les ordonne par date de consultation croissante. Cela permet d'obtenir une liste de consultations associées à leurs patients, triées par date de consultation.
 4. **REQUÊTE 4 :** Cette requête retourne l'identifiant du patient, son adresse e-mail, et la somme des montants totaux de ses factures, regroupés par identifiant de patient et e-mail, et triés par identifiant de patient puis par e-mail. Cela permet d'obtenir une vue agrégée des montants totaux de factures pour chaque patient avec leurs adresses e-mail correspondantes.
 5. **REQUÊTE 5 :** Cette requête retourne toutes les colonnes des consultations et des patients, incluant tous les patients et seulement les consultations qui leur sont associées, triées par date de consultation croissante. Les consultations sans patients associés apparaîtront avec des valeurs `NULL` dans les colonnes correspondantes de la table "CONSULTATION".
- **5 requêtes impliquant plus de 2 tables avec jointures internes dont 1 externe + 1 GROUP BY + 1 tri**
1. **REQUÊTE 1 :** cette requête retourne toutes les colonnes des consultations, des patients et des examens associés, où chaque consultation est liée à son patient correspondant via la jointure avec la table "PATIENT", et chaque consultation est liée à ses examens correspondants via la jointure avec la table "EXAMEN".

2. **REQUÊTE 2 :** Cette requête retourne toutes les colonnes des factures, des patients et des consultations associées, où chaque facture est liée à son patient correspondant via la jointure avec la table "PATIENT", et chaque facture est liée à sa consultation correspondante via la jointure avec la table "CONSULTATION".
3. **REQUÊTE 3 :** Cette requête renvoie toutes les colonnes des consultations, des patients et des examens associés, où chaque consultation est liée à son patient correspondant via la jointure avec la table "PATIENT", et chaque consultation est liée à ses examens correspondants via la jointure avec la table "EXAMEN". Les résultats sont ensuite triés par date de consultation croissante.
4. **REQUÊTE 4 :** Cette requête retourne l'identifiant du patient, son adresse e-mail, la somme des montants totaux de ses factures, ainsi que les dates de la facture et de la consultation correspondantes, regroupés par identifiant de patient, adresse e-mail, date de facture et date de consultation, et triés dans cet ordre. Cela permet d'obtenir une vue agrégée des montants totaux de factures pour chaque patient, avec les détails des factures et des consultations.
5. **REQUÊTE 5 :** Cette requête retourne toutes les colonnes des consultations, des examens et des patients, où chaque consultation est liée à ses examens correspondants et chaque consultation est également liée à son patient correspondant. Les résultats sont triés par date de consultation croissante. Les patients pour lesquels il n'y a pas de consultation correspondante apparaîtront avec des valeurs NULL dans les colonnes correspondantes de la table "CONSULTATION".

1.3 Dictionnaire de données MERISE

Le tableau à la figure 1 présente le dictionnaire de données MERISE.

1.4 Description textuelle des associations

Dans cette section sont décrites les associations entre les différentes entités du modèle. Une association permet de mettre en relation deux ou plusieurs entités :

- **Association "Effectuer" entre MEDECIN et CONSULTATION**

Cette association indique que chaque consultation est effectuée par un médecin. Un médecin peut effectuer plusieurs consultations, mais une consultation est effectuée par un seul médecin à la fois.

- **Association "Inclure" entre FACTURE et CONSULTATION** Cette association représente le fait qu'une facture inclut une consultation. Chaque consultation peut être incluse dans une seule facture, mais une facture peut inclure plusieurs consultations.

- **Association "Passer" entre PATIENT et CONSULTATION** Cette association signifie qu'un patient passe une consultation. Chaque consultation est passée par un seul patient, mais un patient peut passer plusieurs consultations.

- **Association "Contenir" entre CONSULTATION et EXAMEN** Cette association indique que chaque consultation peut contenir plusieurs examens. Chaque examen est associé à une seule consultation.

- **Association "Recevoir" entre PATIENT et FACTURE** Cette association représente le fait qu'un patient peut recevoir une fac-

ture. Chaque facture est destinée à un seul patient, mais un patient peut recevoir plusieurs factures.

- **Association "Contenir" entre CONSULTATION et PRESCRIPTION** Cette association signifie qu'une consultation peut contenir plusieurs prescriptions. Chaque prescription est associée à une seule consultation.

- **Association "Rendez-vous" entre PATIENT et MEDECIN**

Cette association indique que chaque rendez-vous est entre un patient et un médecin. Chaque rendez-vous est pris par un patient avec un médecin spécifique.

1.5 Définition du Modèle Entité-Association MERISE

La figure 2 présente le modèle Modèle Entité-Association MERISE.

1.6 La définition du modèle logique de Données ou schéma relationnel

La figure 3 présente le modèle Modèle Entité-Association MERISE.

1.7 Spécification des traitements avec des packages PLSQL

Voir le fichier correspondant :

Package_plsql_BobCharlemagnePierre_RodelyDouilly_PierreRubensMilorme_DjimySurlin..

1.8 Spécification des triggers

Voici les définitions textuelles des deux triggers :

1.8.1 Trigger TG_CTRL_RENDEZ_VOUS_DOUBLE

— **Objectif**

Empêcher la création de rendez-vous en doublon pour un même patient avec un même médecin à la même date.

— **Déclenchement**

Avant l’insertion ou la mise à jour dans la table RENDEZ_VOUS.

— **Actions**

1. Le trigger vérifie s’il existe déjà un rendez-vous pour le nouveau patient avec le nouveau médecin à la nouvelle date spécifiée.
2. Si un rendez-vous en doublon est trouvé, une erreur est levée pour signaler que le rendez-vous existe déjà pour ce patient avec ce médecin à cette date.

1.8.2 Trigger TG_HISTORIQUE_FACTURE

— **Objectif**

Maintenir un historique des modifications apportées aux factures.

— **Déclenchement**

Après l’insertion, la mise à jour ou la suppression dans la table FACTURE.

— **Actions**

1. Si une nouvelle facture est insérée (INSERTING), les détails de la facture sont insérés dans la table HISTORIQUE_FACTURE avec l’action ‘INSERTION’.
2. Si une facture est mise à jour et que la colonne MONTANT_TOTAL est modifiée (UPDATING (‘MONTANT_TOTAL’)), les anciennes et nouvelles valeurs du montant ainsi que la date de la modification sont insérées dans la table HISTORIQUE_FACTURE avec l’action ‘MODIFICATION’.

3. Si une facture est supprimée (DELETING), les détails de la facture avant sa suppression sont insérés dans la table HISTORIQUE_FACTURE avec l'action 'SUPPRESSION'. De plus, toute référence à cette facture dans la table CONSULTATION est supprimée en mettant à jour la colonne ID_FACTURE_ à NULL.

FIGURE 1 – Dictionnaire de données MERISE

Titre	Caractéristiques
CONSULTATION	Description : Enregistre les détails d'une consultation médicale entre un médecin et un patient.
	Format de données : <ul style="list-style-type: none"> — ID_CONSULTATION_ : Entier — ID_FACTURE_ : Entier — ID_MEDECIN_ : Entier — ID_PATIENT_ : Entier — DATE_CONSULTATION : Date
	Type : <ul style="list-style-type: none"> — ID_CONSULTATION_ : NUMBER(38) — ID_FACTURE_ : NUMBER(38) — ID_MEDECIN_ : NUMBER(38) — ID_PATIENT_ : NUMBER(38) — DATE_CONSULTATION : DATE
	Identifiant : ID_CONSULTATION_
	Contraintes : <ul style="list-style-type: none"> — ID_CONSULTATION_ IS NOT NULL — ID_FACTURE_ IS NOT NULL — ID_MEDECIN_ IS NOT NULL — ID_PATIENT_ IS NOT NULL — DATE_CONSULTATION IS NOT NULL — Clé primaire (PK_CONSULTATION) sur ID_CONSULTATION_ — Contrainte de clé étrangère (FK_CONSULTA_EFFECTUER_MEDECIN) sur ID_MEDECIN_ référençant MEDECIN(ID_MEDECIN_) — Contrainte de clé étrangère (FK_CONSULTA_INCLURE_FACTURE) sur ID_FACTURE_ référençant FACTURE(ID_FACTURE_) — Contrainte de clé étrangère (FK_CONSULTA_PASSER_PATIENT) sur ID_PATIENT_ référençant PATIENT(ID_PATIENT_)

Continued on next page

Table 1: Dictionnaire de données MERISE (Continued)

Titre	Caractéristiques
EXAMEN	Description : Enregistre les détails des examens médicaux effectués lors d'une consultation.
	Format de données : <ul style="list-style-type: none"> — ID_EXAMEN_ : Entier — ID_CONSULTATION_ : Entier — DETAILS_EXAMEN : Chaîne de caractères — DATE_EXAMEN : Date
	Type : <ul style="list-style-type: none"> — ID_EXAMEN_ : NUMBER(38) — ID_CONSULTATION_ : NUMBER(38) — DETAILS_EXAMEN : VARCHAR2(100) — DATE_EXAMEN : DATE
	Identifiant : ID_EXAMEN_
	Contraintes : <ul style="list-style-type: none"> — ID_EXAMEN_ IS NOT NULL — ID_CONSULTATION_ IS NOT NULL — DETAILS_EXAMEN IS NOT NULL — DATE_EXAMEN IS NOT NULL — Clé primaire (PK_EXAMEN) sur ID_EXAMEN_ — Contrainte de clé étrangère (FK_EXAMEN_CONTENIR_CONSULTA) sur ID_CONSULTATION_ référençant CONSULTATION(ID_CONSULTATION_)

Continued on next page

Table 1: Dictionnaire de données MERISE (Continued)

Titre	Caractéristiques
FACTURE	Description : Enregistre les détails des factures générées suite à une consultation.
	Format de données : <ul style="list-style-type: none"> — ID_FACTURE_ : Entier — ID_PATIENT_ : Entier — MONTANT_TOTAL : Nombre décimal — DATE_FACTURE : Date
	Type : <ul style="list-style-type: none"> — ID_FACTURE_ : NUMBER(38) — ID_PATIENT_ : NUMBER(38) — MONTANT_TOTAL : NUMBER(15,2) — DATE_FACTURE : DATE
	Identifiant : ID_FACTURE_
	Contraintes : <ul style="list-style-type: none"> — ID_FACTURE_ IS NOT NULL — ID_PATIENT_ IS NOT NULL — MONTANT_TOTAL IS NOT NULL — DATE_FACTURE IS NOT NULL — Clé primaire (PK_FACTURE) sur ID_FACTURE_ — Contrainte de clé étrangère (FK_FACTURE.RECEVOIR_PATIENT) sur ID_PATIENT_ référençant PATIENT(ID_PATIENT_)

Continued on next page

Table 1: Dictionnaire de données MERISE (Continued)

Titre	Caractéristiques
MEDECIN	Description : Enregistre les détails des médecins.
	Format de données : <ul style="list-style-type: none"> — ID_MEDECIN_ : Entier — NOM : Chaîne de caractères — PRENOM : Chaîne de caractères — SPECIALITE : Chaîne de caractères — TELEPHONE : Chaîne de caractères — EMAIL : Chaîne de caractères
	Type : <ul style="list-style-type: none"> — ID_MEDECIN_ : NUMBER(38) — NOM : VARCHAR2(50) — PRENOM : VARCHAR2(50) — SPECIALITE : VARCHAR2(50) — TELEPHONE : VARCHAR2(8) — EMAIL : VARCHAR2(50)
	Identifiant : ID_MEDECIN_
	Contraintes : <ul style="list-style-type: none"> — ID_MEDECIN_ IS NOT NULL — NOM IS NOT NULL — PRENOM IS NOT NULL — SPECIALITE IS NOT NULL — TELEPHONE IS NOT NULL — EMAIL IS NOT NULL — Clé primaire (PK_MEDECIN) sur ID_MEDECIN_

Continued on next page

Table 1: Dictionnaire de données MERISE (Continued)

Titre	Caractéristiques
PATIENT	Description : Enregistre les détails des patients.
	Format de données : <ul style="list-style-type: none"> — ID_PATIENT_ : Entier — NOM : Chaîne de caractères — PRENOM : Chaîne de caractères — ADRESSE : Chaîne de caractères — EMAIL : Chaîne de caractères — DATE_NAISSAINCE : Date
	Type : <ul style="list-style-type: none"> — ID_PATIENT_ : NUMBER(38) — NOM : VARCHAR2(50) — PRENOM : VARCHAR2(50) — ADRESSE : VARCHAR2(100) — EMAIL : VARCHAR2(50) — DATE_NAISSAINCE : DATE
	Identifiant : ID_PATIENT_
	Contraintes : <ul style="list-style-type: none"> — ID_PATIENT_ IS NOT NULL — NOM IS NOT NULL — PRENOM IS NOT NULL — DATE_NAISSAINCE IS NOT NULL — Clé primaire (PK_PATIENT) sur ID_PATIENT_

Continued on next page

Table 1: Dictionnaire de données MERISE (Continued)

Titre	Caractéristiques
PRESCRIPTION	Description : Enregistre les détails des prescriptions médicales effectuées lors d'une consultation.
	Format de données : <ul style="list-style-type: none"> — ID_PRESCRIPTION_ : Entier — ID_CONSULTATION_ : Entier — DETAILS_PRESCRIPTION : Chaîne de caractères — DATE_PRESCRIPTION : Date
	Type : <ul style="list-style-type: none"> — ID_PRESCRIPTION_ : NUMBER(38) — ID_CONSULTATION_ : NUMBER(38) — DETAILS_PRESCRIPTION : VARCHAR2(100) — DATE_PRESCRIPTION : DATE
	Identifiant : ID_PRESCRIPTION_
	Contraintes : <ul style="list-style-type: none"> — ID_PRESCRIPTION_ IS NOT NULL — ID_CONSULTATION_ IS NOT NULL — DETAILS_PRESCRIPTION IS NOT NULL — DATE_PRESCRIPTION IS NOT NULL — Clé primaire (PK_PRESCRIPTION) sur ID_PRESCRIPTION_ — Contrainte de clé étrangère (FK_PRESCRIP_CONTENIR_CONSULTA) sur ID_CONSULTATION_ référençant CONSULTATION(ID_CONSULTATION_)

Continued on next page

Table 1: Dictionnaire de données MERISE (Continued)

Titre	Caractéristiques
RENDEZ.VOUS	Description : Enregistre les rendez-vous pris entre un patient et un médecin.
	Format de données : <ul style="list-style-type: none"> — ID_PATIENT_ : Entier — ID_MEDECIN_ : Entier — DATE_RENDEZ_VOUS : Date
	Type : <ul style="list-style-type: none"> — ID_PATIENT_ : NUMBER(38) — ID_MEDECIN_ : NUMBER(38) — DATE_RENDEZ_VOUS : DATE
	Identifiant : (ID_PATIENT_, ID_MEDECIN_)
	Contraintes : <ul style="list-style-type: none"> — ID_PATIENT_ IS NOT NULL — ID_MEDECIN_ IS NOT NULL — DATE_RENDEZ_VOUS IS NOT NULL — Clé primaire composée (PK.RENDEZ.VOUS) sur (ID_PATIENT_, ID_MEDECIN_) — Contrainte de clé étrangère (FK.RENDEZ.V.RENDEZ.VO_MEDECIN) sur (ID_MEDECIN_) référençant MEDECIN (ID_MEDECIN_) — Contrainte de clé étrangère (FK.RENDEZ.V.RENDEZ.VO_PATIENT) sur (ID_PATIENT_) référençant PATIENT (ID_PATIENT_)

FIGURE 2 – Définition du Modèle Entité-Association MERISE

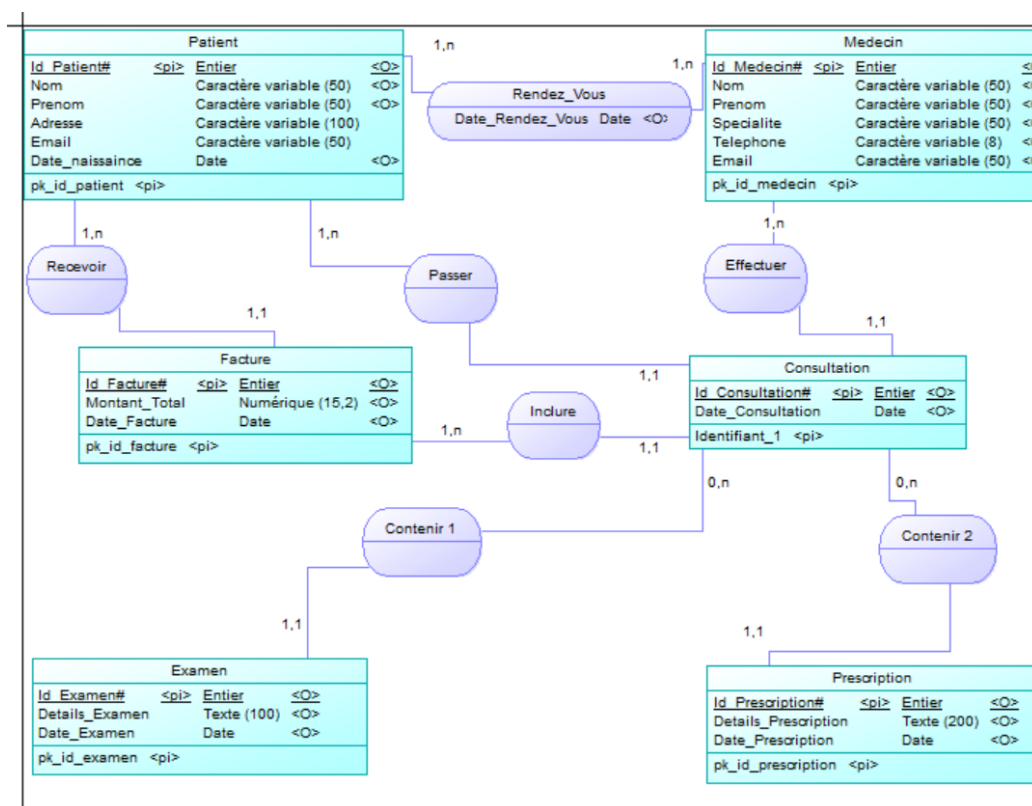


FIGURE 3 – La définition du modèle logique de Données ou schéma relationnel

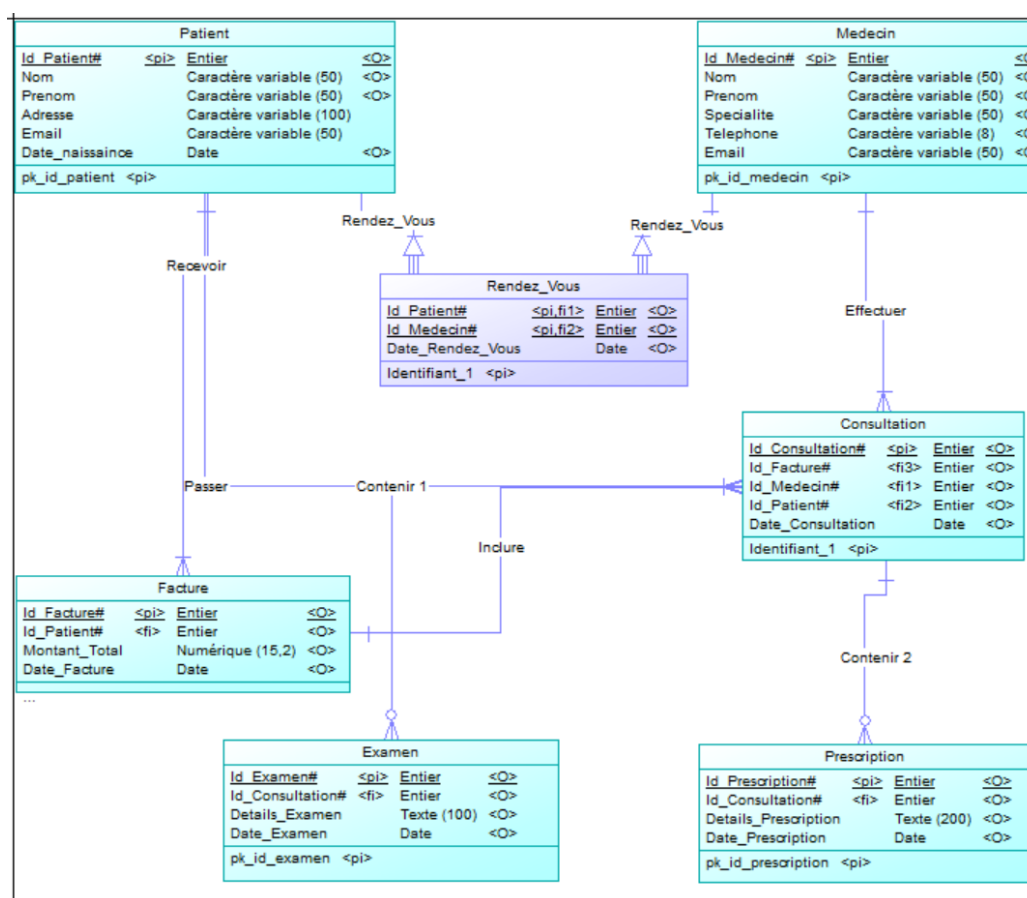


FIGURE 4 – La définition du modèle physique de Données

