



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

28 février 2024

Table des matières

1	Spécification, Analyse et Conception	3
1.1	Description textuelle des requêtes de mise à jour	3
1.1.1	2 requêtes impliquant 1 table	3
1.1.2	2 requêtes impliquant 2 tables	3
1.1.3	2 requêtes impliquant plus de 2 tables	3
1.2	Description textuelle des requêtes de suppression	4
1.2.1	2 requêtes impliquant 1 table	4
1.2.2	2 requêtes impliquant 2 tables	4
1.2.3	2 requêtes impliquant plus de 2 tables	5
1.3	Description textuelle des requêtes de consultation	5
1.3.1	5 requêtes impliquant 1 table dont 1 avec un GROUP BY et 1 avec un ORDER By	5
1.3.2	5 requêtes impliquant 2 tables avec jointures internes dont 1 externe + 1 GROUP BY + 1 tri	6
1.3.3	5 requêtes impliquant plus de 2 tables avec jointures internes dont 1 externe + 1 GROUP BY + 1 tri	7
1.4	Dictionnaire de données MERISE	9
1.5	Description textuelle des associations	16
1.5.1	Association "Effectuer" entre MEDECIN et CONSULTATION	16
1.5.2	Association "Inclure" entre FACTURE et CONSULTATION	17
1.5.3	Association "Passer" entre PATIENT et CONSULTATION	17
1.5.4	Association "Contenir" entre CONSULTATION et EXAMEN	17
1.5.5	Association "Recevoir" entre PATIENT et FACTURE	17
1.5.6	Association "Contenir" entre CONSULTATION et PRESCRIPTION	17
1.5.7	Association "Rendez-vous" entre PATIENT et MEDECIN	17
1.6	Définition du Modèle Entité-Association MERISE	17
1.7	La définition du modèle logique de Données ou schéma relationnel	17
1.8	Spécification des traitement avec des packages PLSQL	17
1.9	Spécification des triggers	17
1.9.1	Trigger "TG_CTRL_RENDEZ_VOUS_DOUBLE"	18
1.9.2	Trigger "TG_HISTORIQUE_FACTURE"	18

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 Spécification, Analyse et Conception

1.1 Description textuelle des requêtes de mise à jour

1.1.1 2 requêtes impliquant 1 table

1. **Requête 1 :** Cette requête effectue une mise à jour dans la table "PATIENT". Elle modifie la valeur de la colonne "DATE_NAISSANCE" pour l'entrée correspondant à l'email thomas.leclerc@email.com'. La nouvelle valeur de "DATE_NAISSANCE" est définie comme étant le 22 août 1986, convertie en format de date à partir de la chaîne '22-AUG-1986' utilisant la fonction TO_DATE('22-AUG-1986','DD-MON-YYYY'). En résumé, cette requête met à jour la date de naissance du patient dont l'email est 'thomas.leclerc@email.com' pour être le 22 août 1986.
2. **Requête 2 :** Cette requête effectue une mise à jour dans la table "PATIENT". Elle modifie la valeur de la colonne "ADRESSE" pour l'entrée correspondant à l'identifiant du patient égal à 1. La nouvelle valeur de "ADRESSE" est définie comme étant '90, DELMAS 75'. En résumé, cette requête met à jour l'adresse du patient dont l'identifiant est 1 pour être '90, DELMAS 75'.

1.1.2 2 requêtes impliquant 2 tables

1. **Requête 1 :** Cette requête effectue une mise à jour sur un sous-ensemble de données résultant d'une jointure entre les tables "PATIENT" (alias P) et "RENDEZ-VOUS" (alias R), où les deux tables sont liées par la colonne "ID_PATIENT_". Une condition WHERE filtre les résultats pour inclure uniquement les rendez-vous ayant lieu entre le 14 février 2024 et le 18 février 2024. Une sous-requête SELECT crée une vue temporaire X avec les colonnes "ID_PATIENT_", "NOM", "PRENOM" et "DATE_RENDEZ-VOUS" pour les rendez-vous sélectionnés. Ensuite, la mise à jour est effectuée sur cette vue temporaire X, modifiant la colonne "DATE_RENDEZ-VOUS" pour tous les rendez-vous sélectionnés, en les déplaçant au 1er mars 2024.
2. **Requête 2 :** Cette requête effectue une mise à jour sur un sous-ensemble de données résultant d'une jointure entre les tables "FACTURE" (alias F) et "PATIENT" (alias P), où les deux tables sont liées par la colonne "ID_PATIENT_". De plus, elle inclut une condition supplémentaire pour sélectionner les factures dont le "MONTANT_TOTAL" est inférieur à 200. Ensuite, la mise à jour est effectuée sur ce sous-ensemble de données, où le "MONTANT_TOTAL" de chaque facture sélectionnée est augmenté de 10 % (en ajoutant 10 % du montant total initial à celui-ci).

1.1.3 2 requêtes impliquant plus de 2 tables

1. **Requête 1 :** Cette requête effectue une mise à jour sur la table "PRESCRIPTION" (alias R), où elle met à jour la colonne "DETAILS_PRESCRIPTION" pour les prescriptions associées à certaines consultations. La sous-requête SELECT sélectionne les identifiants des

consultations (ID_CONSULTATION_) qui correspondent à des consultations ayant eu lieu le 5 février 2024 et associées à un patient spécifique (dont l'identifiant est 1). Ensuite, la mise à jour est appliquée sur les prescriptions correspondant aux consultations sélectionnées, où la colonne "DETAILS_PRESCRIPTION" est définie comme étant 'Zinoboost'.

En résumé, cette requête met à jour les détails des prescriptions associées aux consultations d'un patient spécifique ayant eu lieu le 5 février 2024, en les changeant pour 'Zinoboost'.

2. **Requête 2 :** Cette requête effectue une mise à jour sur la table "EXAMEN" (alias R), où elle met à jour la colonne "DETAILS_EXAMEN" pour les examens associés à certaines consultations. La sous-requête SELECT sélectionne les identifiants des consultations (ID_CONSULTATION_) qui correspondent à des consultations ayant eu lieu le 5 février 2024 et associées à un patient spécifique (dont l'identifiant est 1). Ensuite, la mise à jour est appliquée sur les examens correspondant aux consultations sélectionnées, où la colonne "DETAILS_EXAMEN" est définie comme étant 'HAC1'.
- En résumé, cette requête met à jour les détails des examens associés aux consultations d'un patient spécifique ayant eu lieu le 5 février 2024, en les changeant pour 'HAC1'.

1.2 Description textuelle des requêtes de suppression

1.2.1 2 requêtes impliquant 1 table

1. **Requête 1 :** Cette requête DELETE supprime toutes les entrées de la table "EXAMEN" où la date de l'examen (colonne DATE_EXAMEN) est égale au 5 février 2024. La date est spécifiée dans le format 'DD-MM-YY', et elle est convertie en format de date interne par la fonction TO_DATE(). En résumé, cette requête supprime tous les enregistrements d'examens qui ont eu lieu le 5 février 2024.
 2. **Requête 2 :** Cette requête DELETE supprime toutes les entrées de la table "PRESCRIPTION" où la date de prescription (colonne DATE_PRESCRIPTION) est égale au 5 février 2024. La date est spécifiée dans le format 'DD-MM-YY' et elle est convertie en format de date interne par la fonction TO_DATE().
- En résumé, cette requête supprime toutes les prescriptions datées du 5 février 2024.

1.2.2 2 requêtes impliquant 2 tables

1. **Requête 1 :** Cette requête DELETE supprime tous les rendez-vous de la table "RENDEZ_VOUS" pour lesquels l'identifiant du patient (ID_PATIENT_) correspond à celui des patients ayant l'adresse e-mail 'thomas.leclerc@email.com'. Une sous-requête SELECT récupère l'identifiant du patient (ID_PATIENT_) à partir de la table "PATIENT" où l'adresse e-mail est 'thomas.leclerc@email.com'. En résumé, cette requête supprime tous les rendez-vous associés au patient ayant l'adresse e-mail spécifiée.
2. **Requête 2 :** Cette requête DELETE supprime tous les rendez-vous de la table "RENDEZ_VOUS" pour lesquels l'identifiant du patient (ID_PATIENT_) correspond à 3. Une sous-requête SELECT récupère l'identifiant du patient (ID_PATIENT_) à partir de la table

"PATIENT" où l'identifiant du patient est égal à 3. En résumé, cette requête supprime tous les rendez-vous associés au patient dont l'identifiant est 3.

1.2.3 2 requêtes impliquant plus de 2 tables

1. **Requête 1** : Cette requête DELETE supprime les enregistrements de la table "EXAMEN" pour lesquels l'identifiant de la consultation (ID_CONSULTATION_) correspond à une consultation ayant eu lieu le 5 février 2024 et impliquant un patient spécifique ayant un identifiant égal à 1. La sous-requête SELECT récupère les identifiants des consultations (ID_CONSULTATION_) à partir de la table "CONSULTATION" où la date de consultation est le 5 février 2024 et où l'identifiant du patient (ID_PATIENT_) est égal à 1.
En résumé, cette requête supprime tous les enregistrements d'examens associés aux consultations du patient ayant l'identifiant 1 et ayant eu lieu le 5 février 2024.
2. **Requête 2** : Cette requête DELETE supprime les enregistrements de la table "EXAMEN" pour lesquels l'identifiant de la consultation (ID_CONSULTATION_) correspond à une consultation ayant eu lieu le 6 février 2024 et impliquant un patient spécifique ayant un identifiant égal à 2. La sous-requête SELECT récupère les identifiants des consultations (ID_CONSULTATION_) à partir de la table "CONSULTATION" où la date de consultation est le 6 février 2024 et où l'identifiant du patient (ID_PATIENT_) est égal à 2.
En résumé, cette requête supprime tous les enregistrements d'examens associés aux consultations du patient ayant l'identifiant 2 et ayant eu lieu le 6 février 2024.

1.3 Description textuelle des requêtes de consultation

1.3.1 5 requêtes impliquant 1 table dont 1 avec un GROUP BY et 1 avec un ORDER By

1. **Requête 1** : Cette requête SELECT récupère toutes les données de la table "PATIENT". Elle retourne toutes les colonnes et toutes les lignes de la table "PATIENT", fournissant ainsi une vue complète de toutes les informations stockées dans cette table. En résumé, elle affiche toutes les données sur les patients qui sont enregistrées dans la table.
2. **Requête 2** : Cette requête SELECT récupère toutes les données de la table "CONSULTATION". Elle retourne toutes les colonnes et toutes les lignes de la table "CONSULTATION", fournissant ainsi une vue complète de toutes les informations stockées dans cette table. En résumé, elle affiche toutes les données sur les consultations qui sont enregistrées dans la table.
3. **Requête 3** : Cette requête SELECT compte le nombre d'occurrences de chaque valeur unique dans la colonne "DETAILS_EXAMEN" de la table "EXAMEN" et les groupe par ces valeurs. Elle retournera donc deux colonnes : Une première colonne, contiendra le nombre d'occurrences de chaque valeur unique. Une deuxième colonne contiendra les valeurs uniques de la colonne "DETAILS_EXAMEN". En résumé, cette requête affiche le nombre de fois que chaque détail d'examen spécifique apparaît dans la table "EXAMEN".

4. **Requête 4 :** Cette requête SELECT compte le nombre d'occurrences de chaque valeur unique dans la colonne "DETAILS_EXAMEN" de la table "EXAMEN" et les groupe par ces valeurs. Ensuite, elle trie les résultats par ordre croissant des valeurs de la colonne "DETAILS_EXAMEN". Elle retournera donc deux colonnes : La première colonne, COUNT(*), contiendra le nombre d'occurrences de chaque valeur unique. La deuxième colonne, DETAILS_EXAMEN, contiendra les valeurs uniques de la colonne "DETAILS_EXAMEN". En résumé, cette requête affiche le nombre de fois que chaque détail d'examen spécifique apparaît dans la table "EXAMEN", trié par ordre croissant des détails d'examen.

5. **Requête 5 :** Cette requête SELECT récupère les données suivantes de la table "FACTURE" : ID_PATIENT_ : l'identifiant du patient, COUNT(*) : le nombre total de factures pour chaque patient, SUM(MONTANT_TOTAL) : la somme totale des montants des factures pour chaque patient. Elle groupe ces données par l'identifiant du patient (ID_PATIENT_), calculant ainsi le nombre total de factures et la somme totale des montants de factures pour chaque patient distinct. Ensuite, elle ordonne les résultats par ordre croissant des identifiants de patient (ID_PATIENT_).

En résumé, cette requête fournit le nombre total de factures et la somme totale des montants de factures pour chaque patient, triés par identifiant de patient.

1.3.2 5 requêtes impliquant 2 tables avec jointures internes dont 1 externe + 1 GROUP BY + 1 tri

1. **Requête 1 :** Cette requête SELECT effectue une jointure interne entre deux tables : "CONSULTATION" (alias C) et "PATIENT" (alias P). Elle récupère toutes les colonnes de ces deux tables pour les lignes où l'identifiant du patient (ID_PATIENT_) dans la table "CONSULTATION" correspond à l'identifiant du patient dans la table "PATIENT". En résumé, cette requête affiche toutes les colonnes des consultations et des patients associés, pour lesquels il existe une correspondance entre l'identifiant du patient dans la table "CONSULTATION" et l'identifiant du patient dans la table "PATIENT".

2. **Requête 2 :** Cette requête SELECT effectue une jointure interne entre deux tables : "FACTURE" (alias F) et "PATIENT" (alias P). Elle récupère toutes les colonnes de ces deux tables pour les lignes où l'identifiant du patient (ID_PATIENT_) dans la table "FACTURE" correspond à l'identifiant du patient dans la table "PATIENT". En résumé, cette requête affiche toutes les colonnes des factures et des patients associés, pour lesquels il existe une correspondance entre l'identifiant du patient dans la table "FACTURE" et l'identifiant du patient dans la table "PATIENT".

3. **Requête 3 :** Cette requête SELECT effectue une jointure interne entre deux tables : "CONSULTATION" (alias C) et "PATIENT" (alias P). Elle récupère toutes les colonnes de ces deux tables pour les lignes où l'identifiant du patient (ID_PATIENT_) dans la table "CONSULTATION" correspond à l'identifiant du patient dans la table "PATIENT". Ensuite, elle ordonne les résultats par ordre croissant de la date de consultation (DATE_CONSULTATION) de la table "CONSULTATION". En résumé, cette requête affiche toutes les colonnes des

consultations et des patients associés, pour lesquels il existe une correspondance entre l'identifiant du patient dans la table "CONSULTATION" et l'identifiant du patient dans la table "PATIENT", et les trie par ordre croissant de la date de consultation.

4. **Requête 4 :** Cette requête SELECT effectue une jointure interne entre deux tables : "FACTURE" (alias F) et "PATIENT" (alias P). Elle récupère l'identifiant du patient (ID_PATIENT_) et l'adresse e-mail (EMAIL) de la table "PATIENT", ainsi que la somme des montants totaux des factures (MONTANT_TOTAL) de la table "FACTURE" associées à chaque patient. Elle groupe les résultats par l'identifiant du patient (F.ID_PATIENT_) et l'adresse e-mail du patient (P.EMAIL), calculant ainsi la somme totale des montants des factures pour chaque patient distinct. Ensuite, elle ordonne les résultats par ordre croissant de l'identifiant du patient (F.ID_PATIENT_) et par ordre alphabétique de l'adresse e-mail du patient (P.EMAIL). En résumé, cette requête affiche l'identifiant du patient, son adresse e-mail et la somme totale des montants des factures pour chaque patient, triés par identifiant de patient et par adresse e-mail.
5. **Requête 5 :** Cette requête SELECT effectue une jointure droite (RIGHT JOIN) entre deux tables : "CONSULTATION" (alias C) et "PATIENT" (alias P). Elle récupère toutes les colonnes de ces deux tables, y compris les lignes de la table "PATIENT" pour lesquelles il n'y a pas de correspondance dans la table "CONSULTATION". La condition de jointure spécifie que l'identifiant du patient (ID_PATIENT_) dans la table "CONSULTATION" doit correspondre à l'identifiant du patient dans la table "PATIENT". Ensuite, elle ordonne les résultats par ordre croissant de la date de consultation (DATE_CONSULTATION) de la table "CONSULTATION". En résumé, cette requête affiche toutes les colonnes de la table "CONSULTATION" et de la table "PATIENT", y compris les lignes de la table "PATIENT" qui n'ont pas de correspondance dans la table "CONSULTATION", triées par ordre croissant de la date de consultation.

1.3.3 5 requêtes impliquant plus de 2 tables avec jointures internes dont 1 externe + 1 GROUP BY + 1 tri

1. **Requête 1 :** Cette requête SELECT effectue une jointure interne entre trois tables : "CONSULTATION" (alias C), "PATIENT" (alias P), et "EXAMEN" (alias E). Elle sélectionne toutes les colonnes de chaque table. La première jointure INNER JOIN relie les tables "CONSULTATION" et "PATIENT" sur la base de l'identifiant du patient (ID_PATIENT_). La deuxième jointure INNER JOIN relie les tables "CONSULTATION" et "EXAMEN" sur la base de l'identifiant de la consultation (ID_CONSULTATION_). En résultat, elle retourne toutes les colonnes des consultations, des patients et des examens qui sont associés, c'est-à-dire qu'elle retourne des informations sur les consultations et les patients associés à ces consultations, ainsi que des informations sur les examens qui sont associés à ces mêmes consultations. Notez que cela suppose qu'il existe des correspondances dans toutes les tables pour chaque consultation (c'est-à-dire qu'il y a des patients et des examens associés à chaque consultation). Si des correspondances sont manquantes, cela ne les inclura pas dans les résultats.
2. **Requête 2 :** Cette requête SELECT effectue une jointure interne entre trois tables : "FAC-

TURE" (alias F), "PATIENT" (alias P), et "CONSULTATION" (alias C). Elle sélectionne toutes les colonnes de chaque table. La première jointure INNER JOIN relie les tables "FACTURE" et "PATIENT" sur la base de l'identifiant du patient (ID_PATIENT_). La deuxième jointure INNER JOIN relie les tables "FACTURE" et "CONSULTATION" sur la base de l'identifiant de la facture (ID_FACTURE_). En résultat, elle retourne toutes les colonnes des factures, des patients et des consultations qui sont associées, c'est-à-dire qu'elle retourne des informations sur les factures et les patients associés à ces factures, ainsi que des informations sur les consultations qui sont associées à ces mêmes factures. Notez que cela suppose qu'il existe des correspondances dans toutes les tables pour chaque facture (c'est-à-dire qu'il y a des patients et des consultations associés à chaque facture). Si des correspondances sont manquantes, cela ne les inclura pas dans les résultats.

3. **Requête 3 :** Cette requête SELECT effectue une jointure interne entre trois tables : "CONSULTATION" (alias C), "PATIENT" (alias P), et "EXAMEN" (alias E). Elle sélectionne toutes les colonnes de chaque table.

La première jointure INNER JOIN relie les tables "CONSULTATION" et "PATIENT" sur la base de l'identifiant du patient (ID_PATIENT_).

La deuxième jointure INNER JOIN relie les tables "CONSULTATION" et "EXAMEN" sur la base de l'identifiant de la consultation (ID_CONSULTATION_).

En résultat, elle retourne toutes les colonnes des consultations, des patients et des examens qui sont associés, triées par ordre croissant de la date de consultation (DATE_CONSULTATION).

Notez que cela suppose qu'il existe des correspondances dans toutes les tables pour chaque consultation (c'est-à-dire qu'il y a des patients et des examens associés à chaque consultation). Si des correspondances sont manquantes, cela ne les inclura pas dans les résultats.

4. **Requête 4 :** Cette requête SELECT effectue une jointure interne entre trois tables : "FACTURE" (alias F), "PATIENT" (alias P), et "CONSULTATION" (alias C). Elle sélectionne plusieurs colonnes de chaque table, notamment l'identifiant du patient (ID_PATIENT_) de la table "FACTURE", l'e-mail du patient (EMAIL) de la table "PATIENT", la somme des montants totaux des factures (MONTANT_TOTAL) de la table "FACTURE", la date de facturation (DATE_FACTURE) de la table "FACTURE", et la date de consultation (DATE_CONSULTATION) de la table "CONSULTATION".

Les jointures INNER JOIN relient les tables "FACTURE" et "PATIENT" sur la base de l'identifiant du patient (ID_PATIENT_), et les tables "FACTURE" et "CONSULTATION" sur la base de l'identifiant de la facture (ID_FACTURE_).

La clause GROUP BY regroupe les résultats par l'identifiant du patient (F.ID_PATIENT_), l'e-mail du patient (P.EMAIL), la date de facturation (F.DATE_FACTURE) et la date de consultation (C.DATE_CONSULTATION), calculant ainsi la somme totale des montants des factures pour chaque combinaison unique de ces colonnes.

Ensuite, les résultats sont triés par ordre croissant de l'identifiant du patient (F.ID_PATIENT_), l'e-mail du patient (P.EMAIL), la date de facturation (F.DATE_FACTURE) et la date de consultation (C.DATE_CONSULTATION).

Notez que si plusieurs consultations sont associées à une même facture, cette requête peut produire des résultats inattendus, car elle regroupe par la date de consultation également. Cela peut entraîner l'agrégation de plusieurs lignes de consultation pour une seule facture, ce qui peut ne pas être souhaité dans certains cas.

5. **Requête 5 :** Cette requête SELECT effectue une jointure entre trois tables : "CONSULTATION" (alias C), "EXAMEN" (alias E), et "PATIENT" (alias P). Elle sélectionne toutes les colonnes de chaque table.

La première jointure INNER JOIN relie les tables "CONSULTATION" et "EXAMEN" sur la base de l'identifiant de la consultation (ID_CONSULTATION_). La deuxième jointure RIGHT JOIN relie les tables "EXAMEN" et "PATIENT" sur la base de l'identifiant du patient (ID_PATIENT_). Un RIGHT JOIN signifie que toutes les lignes de la table de droite ("PATIENT") sont incluses dans le résultat, et les lignes de la table de gauche ("EXAMEN") ne sont incluses que si elles correspondent à une ligne dans la table de droite. La clause ORDER BY trie les résultats par ordre croissant de la date de consultation (DATE_CONSULTATION) de la table "CONSULTATION". En résumé, cette requête retourne toutes les colonnes des consultations, des examens et des patients, où les examens sont associés aux consultations, et les consultations sont associées aux patients. Les résultats sont triés par ordre croissant de la date de consultation.

1.4 Dictionnaire de données MERISE

Table 1: Dictionnaire de données MERISE

Titre	Caractéristiques
CONSULTATION	<p>Description : Enregistre les détails d'une consultation médicale entre un médecin et un patient.</p> <hr/> <p>Format de données :</p> <ul style="list-style-type: none"> — ID_CONSULTATION_ : Entier — ID_FACTURE_ : Entier — ID_MEDECIN_ : Entier — ID_PATIENT_ : Entier — DATE_CONSULTATION : Date <hr/> <p>Type :</p> <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 <hr/> <p>Identifiant : ID_CONSULTATION_</p> <hr/> <p>Contraintes :</p> <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_)

1.5 Description textuelle des associations

Sont décrites ci-dessous les associations entre les différentes entités du modèle.

1.5.1 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.

1.5.2 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.

1.5.3 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.

1.5.4 Association "Contenir" entre CONSULTATION et EXAMEN

Cette association indique que chaque consultation peut contenir plusieurs examens. Chaque examen est associé à une seule consultation.

1.5.5 Association "Recevoir" entre PATIENT et FACTURE

Cette association représente le fait qu'un patient peut recevoir une facture. Chaque facture est destinée à un seul patient, mais un patient peut recevoir plusieurs factures.

1.5.6 Association "Contenir" entre CONSULTATION et PRESCRIPTION

Cette association signifie qu'une consultation peut contenir plusieurs prescriptions. Chaque prescription est associée à une seule consultation.

1.5.7 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.

1.6 Définition du Modèle Entité-Association MERISE

La figure 1 présente le modèle Modèle Entité-Association MERISE.

1.7 La définition du modèle logique de Données ou schéma relationnel

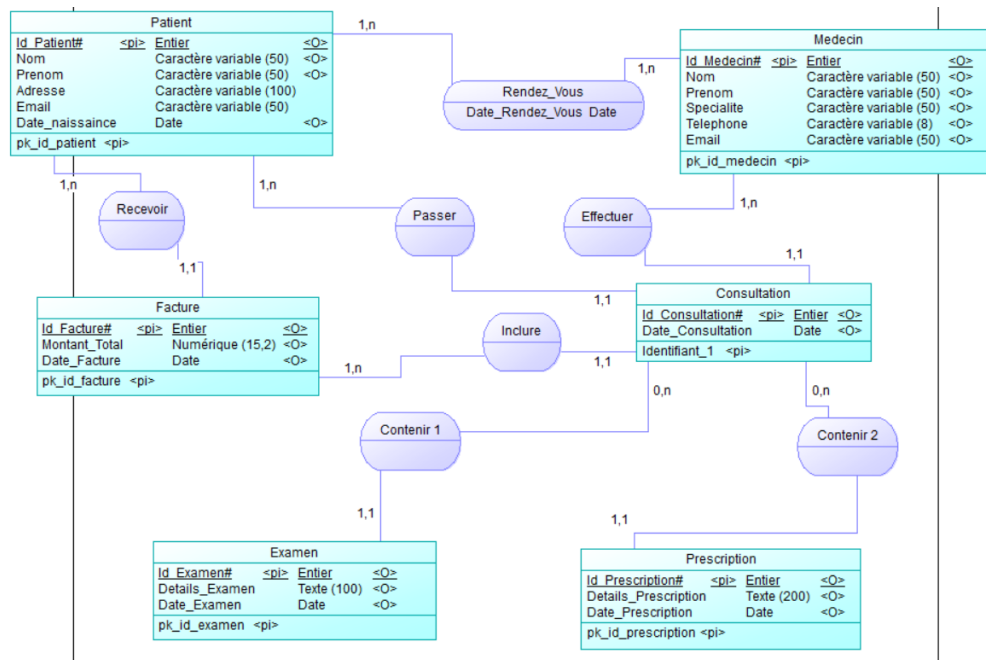
La figure 2 présente le modèle Modèle Entité-Association MERISE.

1.8 Spécification des traitement avec des packages PLSQL

1.9 Spécification des triggers

Voici les définitions textuelles des deux triggers :

FIGURE 1 – Définition du Modèle Entité-Association MERISE



1.9.1 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.

— Action

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.

1.9.2 Trigger "TG_HISTORIQUE_FACTURE"

— Objectif

Maintenir un historique des modifications apportées aux factures.

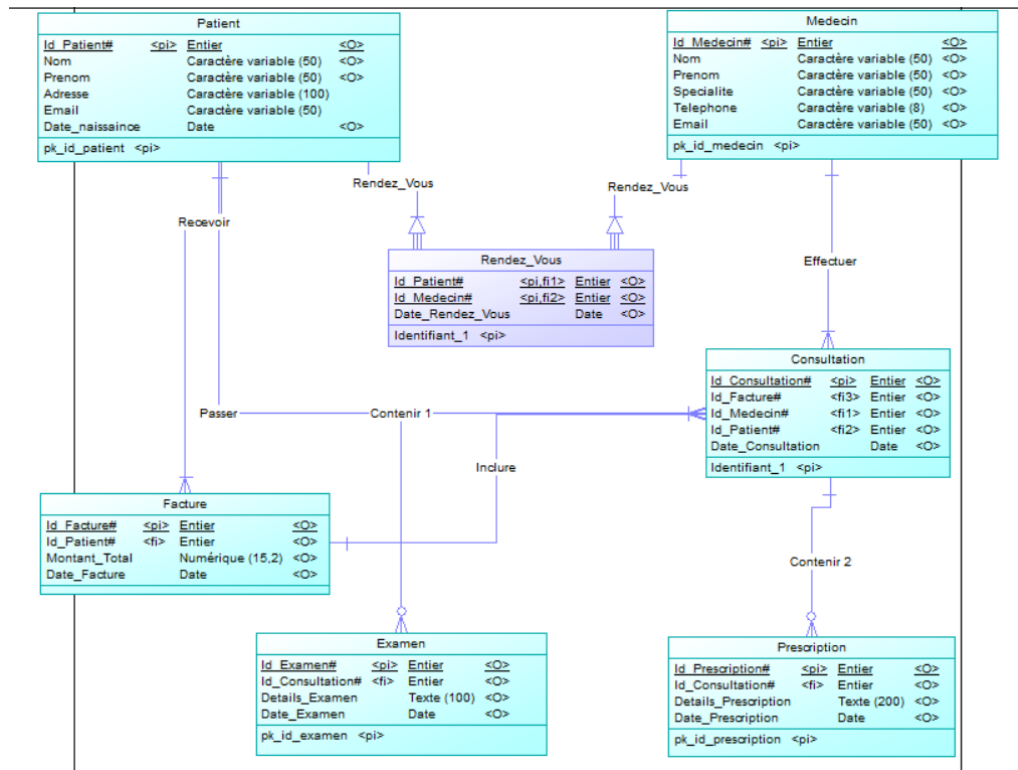
— Déclenchement

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

— Action

1. Si une nouvelle facture est insérée (INSERTING), les détails de la facture sont insérés dans la table HISTORIQUE_FACTURE avec l'action 'INSERTION'.

FIGURE 2 – La définition du modèle logique de Données ou schéma relationnel



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.