

TP de SQL

Titouan Castro
Nathan Domenichini

22 novembre 2024

1 Création de la base

Question 1 : Proposer le schéma EA correspondant à ce schéma relationnel.

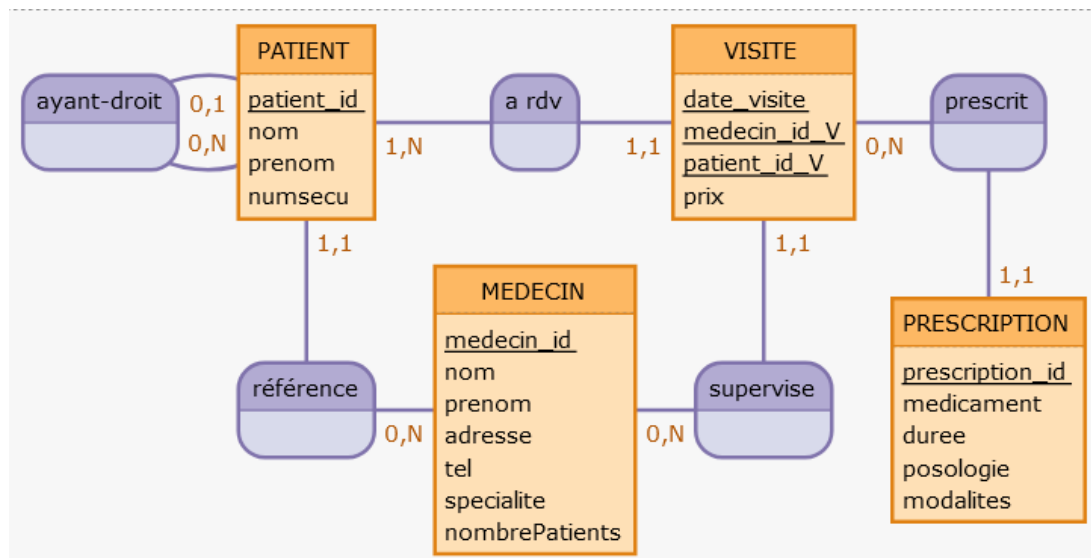


FIGURE 1 – Schéma relationnel

Question 2 : Proposer une implémentation du schéma relationnel en SQL

Nous avons créé nos tables, le script `creation_tables.sql` est joint également. Pour avoir des jeux de données, nous avons utilisé quatre fichiers CSV, et un script Python permettant de transformer les fichiers CSV en INSERT. Dans le .zip est joint ce script python, et aussi tous les insert dans `sql_insert.sql`. Voici le script Python :

```

import csv

def readCSV(table_name: str) -> str:
    res = ''
    csv_file = f'{table_name}.csv'
    table_name = table_name.upper()

    with open(csv_file, 'r') as file:
        reader = csv.reader(file)
        headers = next(reader, None)
        if not headers:
            return res

        for row in reader:
            values = ', '.join(
                'NULL' if value == '' else
                f"'{value}'" if headers[i] == 'tel' else
                f"TO_DATE('{value}', 'YYYY-MM-DD')" if headers[i] == '
                    date_visite' else
                value if value.isdecimal() else
                f"'{value}'"
                for i, value in enumerate(row)
            )
            res += f'INSERT INTO {table_name} ({", ".join(headers)})
                VALUES ({values});\n'
        res += '\n'
    return res

if __name__ == '__main__':
    csv_to_read = ['medecin', 'patient', 'visite', 'prescription']
    final_string = ''.join(readCSV(csv_file) for csv_file in csv_to_read)

    with open('sql_insert.sql', 'w') as file:
        file.write(final_string)

```

```

CREATE TABLE MEDECIN (
  medecin_id    NUMBER(10) NOT NULL PRIMARY KEY,
  nom            VARCHAR(20),
  prenom         VARCHAR(20),
  adresse        VARCHAR(42),
  tel            VARCHAR(10),
  specialite     VARCHAR(20),
  nombrePatients NUMBER(10)
);

CREATE TABLE PATIENT (
  patient_id    NUMBER(10) NOT NULL PRIMARY KEY,
  nom            VARCHAR(20),
  prenom         VARCHAR(20),
  numsecu       NUMBER(13),
  rattachement  NUMBER(10) REFERENCES PATIENT(patient_id),
  medecin_referent  NUMBER(10) REFERENCES MEDECIN(medecin_id)
);

CREATE TABLE VISITE (
  medecin_id    NUMBER(10) REFERENCES MEDECIN(medecin_id),
  patient_id    NUMBER(10) REFERENCES PATIENT(patient_id),
  date_visite   DATE NOT NULL,
  prix          NUMBER(10),
  PRIMARY KEY (date_visite, medecin_id, patient_id)
);

CREATE TABLE PRESCRIPTION (
  prescription_id VARCHAR(20) NOT NULL PRIMARY KEY,
  patient_id      NUMBER(10) NOT NULL,
  medecin_id      NUMBER(10) NOT NULL,
  medicament      VARCHAR(20),
  duree           NUMBER(10), -- Nombre de jours
  posologie       VARCHAR(20),
  modalites       VARCHAR(20),
  date_visite     DATE NOT NULL,
  FOREIGN KEY (date_visite, medecin_id, patient_id) REFERENCES VISITE(
    date_visite, medecin_id, patient_id)
);

-- Trigger pour mettre a jour les nombres de patients apres la
-- modification du mdecin referent
CREATE OR REPLACE TRIGGER update_medecin_referent
AFTER UPDATE OF medecin_referent ON PATIENT
FOR EACH ROW
BEGIN
  -- Si le medecin referent a change (comparaison des anciens et nouveaux
  -- medecins)
  IF :OLD.medecin_referent IS NOT NULL AND :NEW.medecin_referent != :OLD.
    medecin_referent THEN

```

```

-- Decrementation du nombre de patients pour l'ancien medecin rfrent
UPDATE MEDECIN
SET nombrePatients = nombrePatients - 1
WHERE medecin_id = :OLD.medecin_referent;

-- Incrementation du nombre de patients pour le nouveau medecin
referent
UPDATE MEDECIN
SET nombrePatients = nombrePatients + 1
WHERE medecin_id = :NEW.medecin_referent;
END IF;
END;

```

2 Mise à jour de la base

Question 2 : Supprimer les patients n'ayant jamais consulté

On veut supprimer $\pi_{patient_id}(PATIENT) - \pi_{patient_id}(VISITE)$.

```

-- Premiere solution
DELETE FROM patient
WHERE NOT EXISTS (
    SELECT patient_id
    FROM visite
    WHERE visite.patient_id = patient.patient_id
);

-- Deuxieme solution
DELETE FROM patient
WHERE patient_id NOT IN (
    SELECT patient_id
    FROM visite
);

```

Question 3 : Ajouter une visite et les prescriptions correspondantes

```

INSERT INTO VISITE (medecin_id, patient_id, date_visite, prix) VALUES
(3, 2, TO_DATE('2024-12-31', 'YYYY-MM-DD'), 30);
INSERT INTO PRESCRIPTION (prescription_id, patient_id, medecin_id,
    medicament, duree, posologie, modalites, date_visite) VALUES (5, 2,
    3, 'doliprane', 9, '16 cachets', 8, TO_DATE('2024-12-31', 'YYYY-MM-DD
    '));
INSERT INTO PRESCRIPTION (prescription_id, patient_id, medecin_id,
    medicament, duree, posologie, modalites, date_visite) VALUES (6, 2,
    3, 'aspegic', 2, '8 cachets', 2, TO_DATE('2024-12-31', 'YYYY-MM-DD'))
;

```

Question 4 : Affecter un médecin référent à un patient

```
-- Premiere solution
UPDATE PATIENT
SET medecin_referent = 2
WHERE nom = 'Dupont';

-- Deuxieme solution
UPDATE PATIENT
SET medecin_referent = 3
WHERE patient_id = 2;
```

Question 5 : Détacher un patient de l'assuré auquel il est rattaché

```
-- Premiere solution
UPDATE PATIENT
SET rattachement= NULL
WHERE patient_id = 2;

-- Deuxieme solution
UPDATE PATIENT
SET rattachement= NULL
WHERE nom = 'Dupont';
```

3 Consultation de la base

Question 1 : Lister les médecins "généralistes" ordonnés par nombre de patients croissant

$R_1 = \sigma_{specialite="generaliste"}(MEDECIN)$

```
-- Premiere solution
SELECT *
FROM MEDECIN
WHERE specialite='generaliste'
ORDER BY nombrePatients ASC;

-- Deuxieme solution
SELECT *
FROM MEDECIN
WHERE specialite='generaliste'
ORDER BY 7 ASC;
```

Question 2 : Lister les prescriptions de "amoxicillin" ayant une durée supérieure à 7 jours

$R_2 = \sigma_{medicament="amoxicillin", duree > 7}(MEDECIN)$

```
SELECT *
FROM PRESCRIPTION
WHERE medicament = 'amoxicillin' AND duree > 7;
```

Question 3 : Lister les noms des patients et les noms des médecins qu'ils ont visité le "10/11/2023"

$$\begin{aligned}
 R_{31} &= \sigma_{date="10/11/2023"}(VISITE) \\
 R_{32} &= R_{31} \bowtie_{VISITE.patient_id=PATIENT.patient_id} PATIENT \\
 R_{33} &= R_{32} \bowtie_{MEDECIN.medecin_id=PATIENT.medecin_referent} MEDECIN \\
 R_3 &= \pi_{PATIENT.nom, MEDECIN.nom}(R_{33})
 \end{aligned}$$

```

-- Premiere solution
SELECT p.nom AS patient_nom, m.nom AS medecin_nom
FROM VISITE v
JOIN PATIENT p ON v.patient_id = p.patient_id
JOIN MEDECIN m ON v.medecin_id = m.medecin_id
WHERE v.date_visite = TO_DATE('10/11/2023', 'DD/MM/YYYY');

-- Deuxieme solution
SELECT p.nom AS nom_patient, m.nom AS nom_medecin
FROM PATIENT p, MEDECIN m, (SELECT * FROM VISITE WHERE date_visite =
    TO_DATE('10/11/2023', 'DD/MM/YYYY')) v
WHERE p.patient_id = v.patient_id AND m.medecin_id = v.medecin;

```

Question 4 : Affichez pour chaque ayant-droit, les numéros de sécurité sociale, nom et prénom de l'assuré auquel il est rattaché

$$\begin{aligned}
 R_4 &= PATIENT \bowtie_{PATIENT.rattachement=PATIENT.patient_id} PATIENT \\
 R &= \pi_{ad.patient_id, ad.nom, ad.prenom, p.numsecu, p.nom, p.prenom}(R_4)
 \end{aligned}$$

```

-- Premiere solution
SELECT
    ad.patient_id AS ayant_droit_id,
    ad.nom AS ayant_droit_nom,
    ad.prenom AS ayant_droit_prenom,
    p.numsecu AS assure_numsecu,
    p.nom AS assure_nom,
    p.prenom AS assure_prenom
FROM PATIENT ad
JOIN PATIENT p ON ad.rattachement = p.patient_id;

-- Deuxieme solution
SELECT
    p1.nom AS nom_ayant_droit, p1.prenom AS prenom_ayant_droit,
    p2.numsecu AS numsecu_assure,
    p2.nom AS nom_assure, p2.prenom AS prenom_assure
FROM PATIENT p1, PATIENT p2
WHERE p1.patient_id != p1.rattachement AND p1.rattachement = p2.patient_id;

```

Question 5 : Affichez les noms des patients et de leur médecin référent, triés sur le nom du médecin puis le nom du patient

$$R_{51} = PATIENT \bowtie_{PATIENT.medecin_referent=MEDECIN.medecin_id} MEDECIN$$

$$R_5 = \pi_{\substack{p.nom, \\ p.prenom, \\ mref.nom, \\ mref.prenom}} (R_{51})$$

```
-- Premiere solution
SELECT p.nom AS nom_patient,
       p.prenom AS prenom_patient,
       mref.nom AS nom_medecin_ref,
       mref.prenom AS prenom_medecin_ref
FROM PATIENT p
JOIN MEDECIN mref ON mref.medecin_id = p.medecin_referent
ORDER BY nom_medecin_ref, nom_patient;

-- Deuxieme solution
SELECT PATIENT.nom AS nom_patient, MEDECIN.nom AS nom_medecin_referent
FROM PATIENT, MEDECIN
WHERE MEDECIN.medecin_id=PATIENT.medecin_referent
ORDER BY PATIENT.nom, MEDECIN.nom;
```

Question 6 : Lister les assurés qui n'ont pas d'ayant-droit puis le nom du patient

$$R_{61} = \pi_{rattachement}(PATIENT)$$

$$R_{62} = \pi_{patient_id}(PATIENT)$$

$$R_{63} = R_{62} - R_{61}$$

$$R_6 = R_{63} \bowtie_{R_{63}.rattachement=PATIENT.patient_id} MEDECIN$$

```
-- Premiere solution
SELECT
  p.patient_id,
  p.nom,
  p.prenom
FROM PATIENT p
WHERE p.patient_id
IN (SELECT rattachement FROM PATIENT WHERE rattachement IS NULL);

-- Deuxieme solution
SELECT p1.patient_id, p1.nom, p1.prenom
FROM PATIENT p1
WHERE p1.patient_id NOT IN (
  SELECT DISTINCT p2.rattachement
  FROM PATIENT p2
  WHERE p2.rattachement IS NOT NULL
);
```

Question 7 : Lister les médecins qui ne sont pas "généralistes "ni "cardiologues"

$$R_7 = \sigma_{\text{specialite} \neq \text{'generaliste'} \wedge \text{specialite} \neq \text{'cardiologue'}}(\text{MEDECIN})$$

```
-- Premiere solution
SELECT * FROM medecin m
WHERE m.specialite != 'generaliste'
AND m.specialite != 'cardiologue';

-- Deuxieme solution
SELECT * FROM MEDECIN
WHERE specialite NOT IN ('generaliste', 'cardiologue');
```

Question 8 : Lister les patients qui n'ont jamais consulté leur médecin référent

$$R_{81} = \sigma_{\text{patient}}(\text{VISITE})$$

$$R_{82} = \pi_{\text{patient_id}}(\text{PATIENT} \bowtie_{\text{PATIENT.medecin_referent}=\text{VISITE.medecin_id}} \text{VISITE})$$

$$R_{83} = R_{81} - R_{82}$$

$$R_8 = \pi_{\text{p.patient_id}, \text{p.nom}, \text{p.prenom}}(R_{83})$$

```
-- Premiere solution
SELECT p.patient_id, p.nom, p.prenom
FROM PATIENT p
WHERE p.medecin_referent IS NOT NULL
AND p.patient_id NOT IN (
    SELECT DISTINCT v.patient_id
    FROM VISITE v
    WHERE v.medecin_id = p.medecin_referent
);

-- Deuxieme solution
SELECT p.patient_id, p.nom, p.prenom
FROM PATIENT p
WHERE NOT EXISTS (
    SELECT 1
    FROM VISITE v
    WHERE v.patient\_id = p.patient_id
    AND v.medecin = p.medecin_referent
);
```


Question 9 : Lister les patients qui ont consulté tous les médecins "généralistes" "nantais"

$$R_{91} = \pi_{patient, medecin}(VISITE)$$

$$R_{92} = \pi_{medecin_id}(\sigma_{specialite='generaliste'}(MEDECIN))$$

$$R_9 = R_{91} \div R_{92}$$

```
-- Premiere solution
SELECT * FROM PATIENT p WHERE NOT EXISTS (
    SELECT m.medecin_id
    FROM MEDECIN m
    WHERE m.specialite = 'generaliste'
    AND m.adresse = 'Nantes'
    AND m.medecin_id NOT IN (
        SELECT v.medecin_id
        FROM VISITE v
        WHERE v.patient_id = p.patient_id
    )
);

-- Deuxieme solution
SELECT p.nom, p.prenom
FROM PATIENT p
JOIN VISITE v ON p.patient_id = v.patient
JOIN MEDECIN m ON v.medecin = m.medecin_id
WHERE m.specialite = 'generaliste'
AND m.adresse = 'Nantes'
GROUP BY p.patient_id, p.nom, p.prenom
HAVING COUNT(DISTINCT m.medecin_id) = (
    SELECT COUNT(DISTINCT m.medecin_id)
    FROM MEDECIN m
    WHERE m.specialite = 'generaliste'
    AND m.adresse = 'Nantes'
);
```

Question 10 : Déterminez le nombre total de patients ayant un médecin référant

```
-- Premiere solution
SELECT COUNT(DISTINCT patient_id) AS total_patients_avec_referent
FROM PATIENT
WHERE medecin_referent IS NOT NULL;

-- Deuxieme solution
SELECT COUNT(DISTINCT p.patient_id) AS total_patients_avec_referent
FROM PATIENT p
JOIN MEDECIN m ON p.medecin_referent = m.medecin_id;
```

Question 11 : Lister le nombre de médecins par spécialité pour les 3 spécialités comptant le plus de médecins

```
-- Premiere solution
SELECT
    specialite,
    COUNT(*) as nb_medecin
FROM MEDECIN
GROUP BY specialite
ORDER BY nb_medecin DESC
FETCH FIRST 3 ROWS ONLY;

-- Deuxieme solution
SELECT specialite, COUNT(*) AS nombre_medecins
FROM MEDECIN
GROUP BY specialite
HAVING COUNT(*) > 0
ORDER BY nombre_medecins DESC
FETCH FIRST 3 ROWS ONLY;
```

Question 12 : Lister les personnes qui ont plus de deux ayants-droit

```
-- Premiere solution
SELECT
    p.nom,
    p.prenom
FROM PATIENT p
WHERE (
    SELECT COUNT(*)
    FROM PATIENT
    WHERE rattachement = p.patient_id) > 2;

-- Deuxieme solution
SELECT
    p.nom,
    p.prenom,
    COUNT(p2.patient_id) AS nb_ayant_droit
FROM
    PATIENT p
JOIN
    PATIENT p2 ON p2.rattachement = p.patient_id
GROUP BY
    p.patient_id, p.nom, p.prenom
HAVING
    COUNT(p2.patient_id) > 2;
```

Question 13 : Lister les personnes qui ont plus de cinq ayants-droit

```
-- Premiere solution
SELECT
    p.nom,
    p.prenom
FROM PATIENT p
WHERE (
    SELECT COUNT(*)
    FROM PATIENT
    WHERE rattachement = p.patient_id) > 5;

-- Deuxieme solution
SELECT
    p.nom,
    p.prenom,
    COUNT(p2.patient_id) AS nb_ayant_droit
FROM PATIENT p
JOIN PATIENT p2 ON p2.rattachement = p.patient_id
GROUP BY
    p.patient_id, p.nom, p.prenom
HAVING
    COUNT(p2.patient_id) > 5;
```

Question 14 : Lister les personnes qui ont exactement un ayant-droit

```
-- Premiere solution
SELECT
    p.nom,
    p.prenom
FROM PATIENT p
WHERE (
    SELECT COUNT(*)
    FROM PATIENT
    WHERE rattachement = p.patient_id) = 1;

-- Deuxieme solution
SELECT
    p.nom,
    p.prenom,
    COUNT(p2.patient_id) AS nb_ayant_droit
FROM
    PATIENT p
JOIN
    PATIENT p2 ON p2.rattachement = p.patient_id
GROUP BY
    p.patient_id, p.nom, p.prenom
HAVING
    COUNT(p2.patient_id) = 1;
```

Question 15 : Lister les patients dont toutes les visites concernent des "généralistes" ou des "cardiologues"

$$R_{151} = \pi_{patient_id}(PATIENT)$$

$$R_{152} = \pi_{patient}(VISITE \bowtie_{\substack{VISITE.medecin_id=MEDECIN.medecin_id, \\ \wedge MEDECIN.specialite \neq 'generaliste', \\ \wedge MEDECIN.specialite \neq 'cardiologue'}} MEDECIN)$$

$$R_{15} = R_{151} - R_{152}$$

```
SELECT * FROM patient p WHERE NOT EXISTS(
  SELECT m.medecin_id
  FROM MEDECIN m
  WHERE m.specialite NOT IN ('generaliste','cardiologue')
    AND m.medecin_id IN (
      SELECT v.medecin_id
      FROM VISITE v
      WHERE v.patient_id = p.patient_id))
AND EXISTS (
  SELECT 1
  FROM VISITE v
  WHERE v.patient_id = p.patient_id
);
```

-- Deuxieme solution

```
SELECT p.patient_id, p.nom, p.prenom
FROM PATIENT p
JOIN VISITE v ON p.patient_id = v.patient_id
JOIN MEDECIN m ON v.medecin_id = m.medecin_id
WHERE m.specialite IN ('generaliste', 'cardiologue')
GROUP BY p.patient_id, p.nom, p.prenom
HAVING COUNT(DISTINCT m.specialite) =
  (SELECT COUNT(DISTINCT m2.specialite)
   FROM VISITE v2
   JOIN MEDECIN m2 ON v2.medecin_id = m2.medecin_id
   WHERE v2.patient_id = p.patient_id);
```

Question 16 : Affichez les noms de tous les patients et pour les ayants droit, le nom du patient auquel ils sont rattachés

$$R_{161} = \pi_{p1.nom,p2.nom}(PATIENT \bowtie_{PATIENT.rattachement=PATIENT.patient_id} PATIENT)$$

$$R_{162} = \pi_{nom,NULL}(\sigma_{rattachement=NULL}(PATIENT))$$

$$R_{16} = R_{161} \cup R_{162}$$

```
SELECT
  p.patient_id AS patient_id,
  p.nom AS patient_nom,
  p.prenom AS patient_prenom,
  a.nom AS rattachement_nom,
  a.prenom AS rattachement_prenom
FROM PATIENT p
LEFT JOIN PATIENT a ON p.rattachement = a.patient_id;
```

Question 17 : Affichez les prescriptions qui ne contiennent pas de "doliprane" ni d' "aspegic"

$$R_{17} = PRESCRIPTION - \sigma_{\substack{\text{medicament}='doliprane' \\ \vee \text{medicament}='aspegic'}}(PRESCRIPTION)$$

```
-- Premiere solution
SELECT * FROM PRESCRIPTION p
WHERE p.medicament NOT IN ('doliprane', 'aspegic');

-- Deuxieme solution
SELECT * FROM PRESCRIPTION p
WHERE p.medicament!='doliprane' and p.medicament!='aspegic';
```

Question 18 : Affichez les visites où a été prescrit "doliprane" ou "aspegic"

$$R_{181} = \sigma_{\substack{\text{medicament}='doliprane' \\ \vee \text{medicament}='aspegic'}}(PRESCRIPTION)$$

$$R_{182} = \pi_{\text{medecin}, \text{patient}, \text{date_visite}}(R_{181})$$

$$R_{18} = R_{182} \bowtie_{R_{182}.\text{date_visite}=VISITE.\text{date_visite}} VISITE$$

```
SELECT * FROM
    VISITE v,
    PRESCRIPTION p
WHERE v.patient_id=p.patient_id
AND v.medecin_id=p.medecin_id
AND v.date_visite=p.date_visite
AND (p.medicament='doliprane' or p.medicament='aspegic');
```

Question 19 : Affichez les visites où ont été prescrits "doliprane" et "aspégic"

$$R_{191} = \pi_{\text{medecin}, \text{patient}, \text{date_visite}}(\sigma_{\text{medicament}='doliprane'}(PRESCRIPTION))$$

$$R_{192} = \pi_{\text{medecin}, \text{patient}, \text{date_visite}}(\sigma_{\text{medicament}='aspegic'}(PRESCRIPTION))$$

$$R_{193} = R_{191} \cap R_{192}$$

$$R_{19} = VISITE \bowtie_{VISITE.\text{date_visite}=R_{193}.\text{date_visite}} R_{193}$$

```
SELECT v.*
FROM VISITE v
WHERE EXISTS (
    SELECT pr.prescription_id
    FROM PRESCRIPTION pr
    WHERE pr.medecin_id = v.medecin_id AND pr.patient_id = v.patient_id
        AND pr.date_visite = v.date_visite
        AND pr.medicament = 'doliprane'
)
AND EXISTS (
    SELECT pr.prescription_id
    FROM PRESCRIPTION pr
    WHERE pr.medecin_id = v.medecin_id AND pr.patient_id = v.patient_id
        AND pr.date_visite = v.date_visite
        AND pr.medicament = 'aspegic'
);
```

Question 20 : Affchez les noms des médicaments prescrits par tous les médecins "nantais"

$$R_{201} = \pi_{medecin_id}(\sigma_{adresse='Nantes'}(MEDECIN))$$

$$R_{202} = \pi_{medicament,medecin_id}((\sigma_{adresse='Nantes'}(MEDECIN)) \bowtie PRESCRIPTION)$$

$$R_{20} = \pi_{medicament}(R_{202}) \div \pi_{medecin_id}(R_{201})$$

```
SELECT DISTINCT pr.medicament
FROM PRESCRIPTION pr
WHERE NOT EXISTS (
    SELECT m.medecin_id
    FROM MEDECIN m
    WHERE m.adresse = 'Nantes'
    AND m.medecin_id NOT IN (
        SELECT pr2.medecin_id
        FROM PRESCRIPTION pr2
        WHERE pr2.medicament = pr.medicament
    )
);
```