### IUT de Lille

BUT SD 2024-2025

Semestre 1

### **SAE 1-01 :** **Création de reporting à partir de données stockées dans un SGBD relationnel**

#### **Étude de cas : Clinique vétérinaire**

Ce projet a pour objectif de vous familiariser avec la gestion et l'analyse de données au sein d'une base de données relationnelle et de créer des reportings sur ACCESS. Vous travaillerez sur une base de données de gestion d'une clinique vétérinaire, comprenant des informations sur les animaux, les clients, les visites, les traitements et les médicaments administrés.

#### **Schéma de la base de données**

La base de données relationnelle « Clinique » comprend les tables suivantes :

1. **Animal** : Informations sur les animaux traités par la clinique  
   *(IdAnimal, NumCli, NomAnimal, TypeAnimal, Race, DN, Sexe, Couleur, Stérilisation, Taille, Poids, Décédé, Photo, Commentaires)*
2. **Client** : Informations sur les clients de la clinique  
   *(NumClient, TypeClient, NomClient, Rue, Localité, Département, CodePostal, Téléphone, SiteWeb, ClientDepuis, Remise, Balance)*
3. **Medicament** : Informations sur les médicaments disponibles et leur coût  
   *(CodeMedi, NomMed, PrixMed)*
4. **Traitement** : Détails des traitements proposés par la clinique  
   *(CodeTrai, NomTrai, PrixTrai)*
5. **Visite** : Informations sur les visites réalisées pour chaque animal  
   *(NumVis, IdAnimal, DateVisite, TypeSuivi, DateSuivi, TypePaiement, DatePaiement, Envoi, Détails)*
6. **DetailVisite** : Détail des traitements et médicaments administrés lors des visites  
   *(NumVis, NumLig, TypeVisite, CodeTrait, CodeMedi)*
7. **ListeAnimaux** : Liste des types d'animaux pris en charge par la clinique  
   *(TypeAnimal)*

### **Partie 1 : Exploration et extraction des données (Requêtes SQL)**

1. Pour chaque table, donner sa clé primaire et ses clés étrangères (faites un tableau).

|  |  |  |
| --- | --- | --- |
| **Table** | **Clé primaire** | **Clés étrangères** |
|  |  |  |
| DetailVisite | VisNum  VisLig  ( couple de clé primaire) | VisNum |

|  |  |  |
| --- | --- | --- |
| **Table** | **Clé primaire** | **Clés étrangères** |
| Traitement | CodeTrai | VisTrait |

|  |  |  |
| --- | --- | --- |
| **Table** | **Clé primaire** | **Clés étrangères** |
| Medicament | MedCode | VisMed |

|  |  |  |
| --- | --- | --- |
| **Table** | **Clé primaire** | **Clés étrangères** |
| Visite | VisNum | VisNum |

|  |  |  |
| --- | --- | --- |
| **Table** | **Clé primaire** | **Clés étrangères** |
| Animal | Anild | visAni |

|  |  |  |
| --- | --- | --- |
| **Table** | **Clé primaire** | **Clés étrangères** |
| Client | CliNum | Anicli |

|  |  |  |
| --- | --- | --- |
| **Table** | **Clé primaire** | **Clés étrangères** |
| Paiement | PaiTypeCode | CliType, VisTypepaie  VisTypepaie |

1. Iplémenter le schéma relationnel sous ACCESS

### 

### **Partie 2 : Exploration et extraction des données (Requêtes SQL : ordre SELECT)**

Vous êtes chargés de rédiger et exécuter les requêtes SQL suivantes pour explorer et extraire des informations à partir des données de la clinique. Ces requêtes doivent permettre d’analyser les données sur les animaux, les clients, les visites, les traitements et les médicaments. Pour vos tests, vous pouvez ajouter des données dans les différentes tables pour vérifier que vos requêtes fonctionnent bien dans différents cas.

1. Lister tous les clients avec leur nom, localité et département.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT CliNom, CliLocalite, CliDep  FROM client; |  |

1. Afficher les informations de tous les médicaments disponibles avec leurs prix.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT MedNom, MedPrix  FROM medicament; |  |

1. Obtenir les noms et prix des traitements proposés par la clinique.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT NomTrai, PrixTrai  FROM traitement; |  |

1. Lister toutes les visites effectuées pour chaque animal, en montrant la date de visite et le type de suivi.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT AniNom, VisSuivi, VisDate  FROM visite INNER JOIN Animal ON visite.VisAni = Animal.AniId; |  |

1. Lister les informations de chaque animal, y compris la race, le sexe et la date de naissance.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT AniNom, AniRace, AniSexe, AniDN  FROM Animal; |  |

1. Récupérer les noms des clients et des animaux qu’ils possèdent.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT client.CliNom, animal.AniNom  FROM client INNER JOIN animal ON client.CliNum =animal.Anicli; |  |

1. Afficher les noms de tous les traitements administrés aux animaux de type « Chat ».

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT traitement.NomTrai, animal.AniType  FROM ((traitement INNER JOIN detailvisite ON traitement.CodeTrai = detailvisite.VisTrait) INNER JOIN visite ON detailvisite.VisNum = visite.VisNum) INNER JOIN animal ON visite.VisAni = animal.AniId  WHERE animal.AniType = "Chat"; |  |

1. Afficher les informations des animaux mâles.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT \*  FROM Animal  WHERE AniSexe = "M"; |  |

1. Récupérer la liste de tous les clients ayant un site web.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT CliSiteWeb  FROM client  WHERE CliSiteWeb is not null; |  |

1. Obtenir la liste des animaux stérilisés.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT AniNom, AniSteril  FROM animal  WHERE AniSteril = True; |  |

1. Quels sont les médicaments et les dates de prescription aux chiens ainsi que les différents numéro et nom de chien ? (Donner deux solutions si c’est possible : une avec imbrication de bloc et l’autre sans.)

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT medicament.MedNom, visite.VisDate, animal.AniId, animal.AniNom  FROM ((medicament INNER JOIN detailvisite ON medicament.medcode = detailvisite.vismed) INNER JOIN visite ON detailvisite.visnum = visite.visnum) INNER JOIN animal ON visite.visani = animal.AniId  WHERE AniType = "Chien"; |  |

1. Quels sont les animaux n’ayant jamais eu de traitement ?

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT animal.AniId  FROM (animal INNER JOIN visite ON animal.AniId = visite.visani) INNER JOIN detailvisite ON visite.visnum = detailvisite.visnum  WHERE detailVisite.vismed = "0000"; |  |

1. Quels sont les animaux n’ayant pas eu de visite ?

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT animal.AniNom, visite.VisDate  FROM visite RIGHT JOIN animal ON visite.visani = animal.AniId  WHERE visite.VisDate is null; |  |

1. Donner pour chaque animal, son nom, sa race, et les différentes dates de visite à la clinique :

* a. Faire figurer dans le résultat de la requête que les animaux qui ont eu au moins une visite.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT Animal.AniNom, Animal.AniRace, COUNT(Visite.VisDate) AS ["Nb visite"], Visite.VisDate  FROM Animal LEFT JOIN Visite ON Animal.AniId = Visite.VisAni  GROUP BY Animal.AniNom, Animal.AniRace, Visite.VisDate  HAVING COUNT (Visite.VisDate) >= 1; |  |

* b. Faire figurer dans le résultat de la requête tous les animaux.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT animal.AniNom, animal.AniRace, visite.VisDate  FROM animal LEFT JOIN visite ON visite.VisAni = animal.AniId; |  |

1. Lister les traitements coûtant plus d’un prix (de votre choix).

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT NomTrai, PrixTrai  FROM traitement  WHERE PrixTrai > 100; |  |

1. Récupérer tous les clients ayant une remise supérieure à une remise donnée (de votre choix).

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT CliNom, CliRemise  FROM client  WHERE CliRemise>0.2; |  |

1. Afficher les animaux décédés avec leur nom, type et date de naissance.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT AniNom, AniType, AniDN, AniDecede  FROM animal  WHERE AniDecede = True; |  |

1. Lister les clients enregistrés dans la clinique depuis plus de cinq ans.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT CliNum, CliNom, Clidepuis  FROM client  WHERE Clidepuis<DATEADD("yyyy",-5,NOW()); |  |

1. Trouver tous les animaux de type « Chien » pesant plus d’une valeur donnée.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT AniType, Poids  FROM Animal  WHERE AniType = "Chien" and Poids > 20; |  |

1. Afficher les médicaments dont le prix est inférieur à la moyenne des prix.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT MedNom, MedPrix  FROM medicament  WHERE MedPrix < (SELECT AVG(MedPrix) FROM medicament); |  |

1. Récupérer les visites pour lesquelles le paiement a été effectué en liquide.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT visite.VisNum, paiement.PaiTypeCode, paiement.PaiTypeLib  FROM paiement INNER JOIN visite ON paiement.PaiTypeCode = visite.VisTypepaie  WHERE paiement.PaiTypeCode = 2; |  |

1. Lister les animaux avec un commentaire associé dans leur fiche.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT AniNom, AniCommentaires  FROM Animal  WHERE AniCommentaires is not null; |  |

1. Trouver les animaux ayant des propriétaires dans un département de votre choix.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT animal.AniNom, client.CliDep  FROM animal INNER JOIN client ON animal.AniCli = client.CliNum  WHERE client.CliDep like "34"; |  |

1. Compter le nombre total d'animaux traités par la clinique.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT count(VisAni) AS ["Total animal"]  FROM Visite; |  |

1. Calculer le coût total des médicaments disponibles.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT sum(MedPrix)  FROM medicament; |  |

1. Récupérer le nombre de clients par département.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT CliDep AS ["Departement"], count(CliNum) AS ["Nombre de clients"]  FROM client  GROUP BY CliDep; |  |

1. Calculer la durée moyenne de vie des animaux dans la base.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT avg(datediff("yyyy", anidn,date())) AS moyenne\_de\_vie\_des\_animaux  FROM Animal; |  |

1. Trouver le nombre total de visites réalisées pour chaque animal.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT animal.AniNom, count(VisDate)as ["total\_des\_visites"]  FROM animal INNER JOIN visite ON animal.AniId = visite.visani  GROUP BY animal.AniNom; |  |

1. Afficher le nombre de médicaments administrés par visite.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT animal.AniNom, count(visite.VisDate) AS ["Nb de visite"]  FROM visite INNER JOIN animal ON visite.visani = animal.AniId  GROUP BY animal.AniNom; |  |

1. Calculer le nombre moyen de traitements par visite pour chaque animal.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT animal.AniNom, visite.VisDate, count(NomTrai) AS ["Nb moyen de visite par traitement"]  FROM ((traitement INNER JOIN detailvisite ON traitement.codetrai = detailvisite.vistrait) INNER JOIN visite ON detailvisite.Visnum = visite.Visnum) INNER JOIN animal ON visite.VisAni = animal.AniId  GROUP BY animal.AniNom, visite.VisDate;  Version 2  SELECT AniNom, avg(NombreDeTraitements)  FROM (select animal.AniNom, visite.VisDate, count(NomTrai) as NombreDeTraitements  ((traitement inner join detailvisite on traitement.codetrai = detailvisite.vistrait) inner join visite on detailvisite.Visnum = visite.Visnum) inner join animal on visite.VisAni = animal.  group by animal.AniNom, visite.VisDate) AS TraitementsParVisite  GROUP BY AniNom; |  |

1. Calculer le chiffre d'affaires total généré par les traitements.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT ROUND(sum((1-Cliremise)\*prixTrai),2) AS Ca\_Traitement  FROM (((client INNER JOIN animal ON Client.CliNum = Animal.AniCli) INNER JOIN Visite ON Animal.AniId = Visite.VisAni) INNER JOIN DetailVisite ON Visite.Visnum = DetailVisite.Visnum) INNER JOIN traitement ON DetailVisite.VisTrait = traitement.CodeTrai; |  |

1. Afficher la somme des remises accordées à chaque client.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT Client.CliNum, ROUND(SUM((Medicament.MedPrix-((1-Client.CliRemise))\*Medicament.MedPrix))+SUM((Traitement.PrixTrai-(1-Client.CliRemise)\*Traitement.PrixTrai)),2) AS ["Somme total"]  FROM ((((Client INNER JOIN Animal ON Client.CliNum = Animal.AniCli) INNER JOIN Visite ON Animal.AniId = Visite.VisAni) INNER JOIN DetailVisite ON Visite.VisNum = DetailVisite.VisNum) INNER JOIN Medicament ON DetailVisite.VisMed = Medicament.MedCode) INNER JOIN Traitement ON DetailVisite.VisTrait = Traitement.CodeTrai  GROUP BY Client.CliNum; |  |

1. Récupérer le coût total des médicaments administrés aux chats.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT AniType, sum(MedPrix)  FROM ((medicament INNER JOIN detailvisite ON medicament.MedCode = detailvisite.VisMed) INNER JOIN visite ON detailvisite.VisNum = visite.VisNum) INNER JOIN animal ON visite.VisAni = animal.AniId  GROUP BY AniType  HAVING AniType = "Chat"; |  |

1. Afficher les noms des animaux et les noms de leurs propriétaires.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT AniNom as ["nom\_animaux "], CliNom as ["nom\_propriétaires"]  FROM animal INNER JOIN client ON animal.AniCli =client.CliNum; |  |

1. Récupérer les noms des clients et les types d'animaux qu'ils possèdent.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT CliNom as ["noms\_animaux "] , AniType as ["types \_animaux "]  FROM animal INNER JOIN client ON animal.AniCli =client.CliNum; |  |

1. Afficher les noms des animaux et les traitements qu’ils ont reçus.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT animal.AniNom, traitement.NomTrai  FROM ((traitement INNER JOIN detailvisite ON traitement.codetrai = detailvisite.vistrait) INNER JOIN visite ON detailvisite.Visnum = visite.Visnum) INNER JOIN animal ON visite.VisAni = animal.AniId; |  |

1. Obtenir le nom de chaque client et le nombre de visites qu’il a effectuées.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT client.CliNom, count(VisDate)as ["nombre\_de\_visites"]  FROM (client INNER JOIN animal ON client.CliNum = animal.AniCli) INNER JOIN visite ON animal.AniId = visite.VisAni  GROUP BY client.CliNom; |  |

1. Lister les médicaments prescrits lors des visites, en incluant les détails de chaque visite.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT medicament.MedNom, visite.VisDate, detailvisite.VisType, visite.VisSuivi, visite.VisTypepaie, detailvisite.VisTrait  FROM (medicament INNER JOIN detailvisite ON medicament.MedCode = detailvisite.VisMed) INNER JOIN visite ON detailvisite.VisNum = visite.VisNum; |  |

1. Afficher les clients avec leur localité et le nombre d'animaux qu'ils possèdent.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT client.CliNom, client.CliLocalite, count(AniId) AS ["Nombre d'animaux"]  FROM animal INNER JOIN client ON animal.AniCli =client.CliNum  GROUP BY client.CliNom, client.CliLocalite; |  |

1. Récupérer les noms des clients et les visites effectuées pour chacun de leurs animaux.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT client.CliNom, visite.VisDate, animal.AniId,animal.AniNom  FROM (client INNER JOIN animal ON client.CliNum =animal.AniCli) INNER JOIN visite ON animal.AniId = visite.VisAni; |  |

1. Afficher les détails de toutes les visites avec les traitements administrés et médicaments utilisés.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT Visite.VisDate, DetailVisite.VisNum, DetailVisite.VisLig, DetailVisite.VisType, DetailVisite.VisMed, Traitement.NomTrai, Medicament.MedNom  FROM Visite INNER JOIN (Medicament INNER JOIN (Traitement INNER JOIN DetailVisite ON Traitement.CodeTrai = DetailVisite.VisTrait) ON Medicament.MedCode = DetailVisite.VisMed) ON Visite.VisNum = DetailVisite.VisNum; |  |

1. Obtenir la liste des clients avec le montant total de leurs factures de visites.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT Client.CliNum, CCUR(ROUND(SUM((Traitement.PrixTrai+Medicament.MedPrix)\*(1-Client.CliRemise)),2)) AS [" Facture"]  FROM ((((Client INNER JOIN Animal ON Client.CliNum = Animal.AniCli) INNER JOIN Visite ON ANimal.AniId = Visite.VisAni) INNER JOIN DetailVisite ON Visite.VisNum = DetailVisite.VisNum) INNER JOIN Traitement ON DetailVisite.VisTrait = Traitement.CodeTrai) INNER JOIN Medicament ON DetailVisite.VisMed = Medicament.MedCode  GROUP BY Client.CliNum; |  |

1. Récupérer la liste des visites et des médicaments prescrits pour chaque visite.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT visite.VisNum, medicament.MedNom  FROM (medicament INNER JOIN detailvisite ON medicament.MedCode = detailvisite.VisMed) INNER JOIN visite ON detailvisite.VisNum = visite.VisNum  GROUP BY visite.VisNum, medicament.MedNom; |  |

1. Trouver les noms des clients dont le nom commence par la lettre « M ».

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT Client.CliNom AS NomClient  FROM Client  WHERE Client.CliNom LIKE "M\*"; |  |

1. Afficher les médicaments administrés à des animaux d’un type de votre choix.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT distinct medicament.medNom, animal.anitype  FROM ((animal INNER JOIN visite ON animal.aniId = visite.visani) INNER JOIN detailvisite ON visite.visnum = detailvisite.visnum) INNER JOIN medicament ON detailvisite.vismed = medicament.medcode  WHERE animal.anitype = "Chien"; |  |

1. Lister les visites de suivi effectuées après le premier semestre 2020.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT visite.VisNum, visite.VisAni, visite.VisDate, DetailVisite.VisType, visite.VisDateSuivi  FROM Visite INNER JOIN DetailVisite ON Visite.VisNum = DetailVisite.VisNum  WHERE visite.VisDate > #2020-06-30#; |  |

1. Récupérer les clients ayant au moins deux animaux.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT client.CliNum, client.CliNom, count(AniNom) AS ["Nonbre d'animaux"]  FROM client INNER JOIN animal ON client.clinum = animal.anicli  GROUP BY client.CliNum, client.CliNom  HAVING count(AniNom) >= 2; |  |

1. Afficher les clients ayant une balance de compte négative.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT Client.CliNum  FROM (Client INNER JOIN Animal ON Client.CliNum = Animal.AniCli) INNER JOIN Visite ON Animal.AniId = Visite.VisAni  WHERE VisDatePaiement IS NULL; |  |

1. Trouver tous les animaux ayant reçu un traitement spécifique de votre choix.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT animal.AniNom, traitement.NomTrai  FROM ((animal INNER JOIN visite ON animal.aniId = visite.visani) INNER JOIN detailvisite ON visite.visnum = detailvisite.visnum) INNER JOIN traitement ON detailvisite.vistrait = traitement.codetrai  WHERE traitement.NomTrai = "Amputation d'un membre"; |  |

1. Lister les animaux nés avant une année de votre choix.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT AniId, AniNom, AniDN  FROM animal  WHERE AniDN < #2015-01-01#; |  |

1. Trouver les clients dont les animaux ont été traités avec un médicament spécifique.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT client.CliNom, medicament.MedNom  FROM (((client INNER JOIN animal ON client.CliNum = animal.anicli) INNER JOIN visite ON animal.aniId = visite.visani) INNER JOIN detailvisite ON visite.visnum = detailvisite.visnum) INNER JOIN medicament ON detailvisite.vismed = medicament.medcode  WHERE medicament.MedNom = "Aspirine - 100 mg"; |  |

1. Afficher les clients ayant effectué au moins une visite au cours du dernier mois.

|  |  |
| --- | --- |
| Requête | Résultat |
| select CliNum  from ((client  inner join animal on client.CliNum = animal.AniCli)  inner join visite on animal.AniId = visite.VisAni)  where VisDate >= DateAdd("m", -1, Date()); |  |

1. Obtenir la liste des clients qui n'ont effectué aucune visite depuis leur enregistrement.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT CliNom, CliNum, count(visdate)  FROM (client LEFT JOIN animal ON client.clinum = animal.anicli) LEFT JOIN visite ON animal.AniId = visite.Visani  GROUP BY CliNom,CliNum  HAVING count(visdate)= 0; |  |

1. Trouver les clients ayant effectué le plus grand nombre de visites (top 5).

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT TOP 5 CliNom, count(visdate)  FROM (client INNER JOIN animal ON client.clinum = animal.anicli) INNER JOIN visite ON animal.AniId = visite.Visani  GROUP BY CliNom  ORDER BY count(visdate) DESC; |  |

1. Requête pour déterminer les animaux ayant eu plus de trois traitements différents en un an.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT A.AniId, A.AniNom, COUNT(\*) AS NbTraitements, YEAR(V.VisDate) AS Annee  FROM (Animal AS A  INNER JOIN Visite AS V ON A.AniId = V.VisAni)  INNER JOIN (SELECT DISTINCT DV.VisNum, DV.VisTrait  FROM DetailVisite AS DV) AS UniqueDetails  ON V.VisNum = UniqueDetails.VisNum  GROUP BY A.AniId, A.AniNom, YEAR(V.VisDate)  HAVING COUNT(\*) > 3; |  |

1. Calcul du chiffre d’affaires par trimestre de l'année 2021, pour chaque type de traitement.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT Traitement.NomTrai AS TypeTraitement, FORMAT(Visite.VisDate,"yyyy 'T'Q") AS Trimestre, SUM(Traitement.PrixTrai\*1-cliremise) AS ChiffreAffaires  FROM Client INNER JOIN (Animal INNER JOIN (Visite INNER JOIN (Traitement INNER JOIN DetailVisite ON Traitement.CodeTrai = DetailVisite.VisTrait) ON Visite.VisNum = DetailVisite.VisNum) ON Animal.AniId = Visite.VisAni) ON Client.CliNum = Animal.AniCli  WHERE YEAR(Visite.VisDate) = 2021  GROUP BY Traitement.NomTrai, FORMAT(Visite.VisDate,"yyyy 'T'Q")  ORDER BY Traitement.NomTrai, FORMAT(Visite.VisDate,"yyyy 'T'Q"); |  |

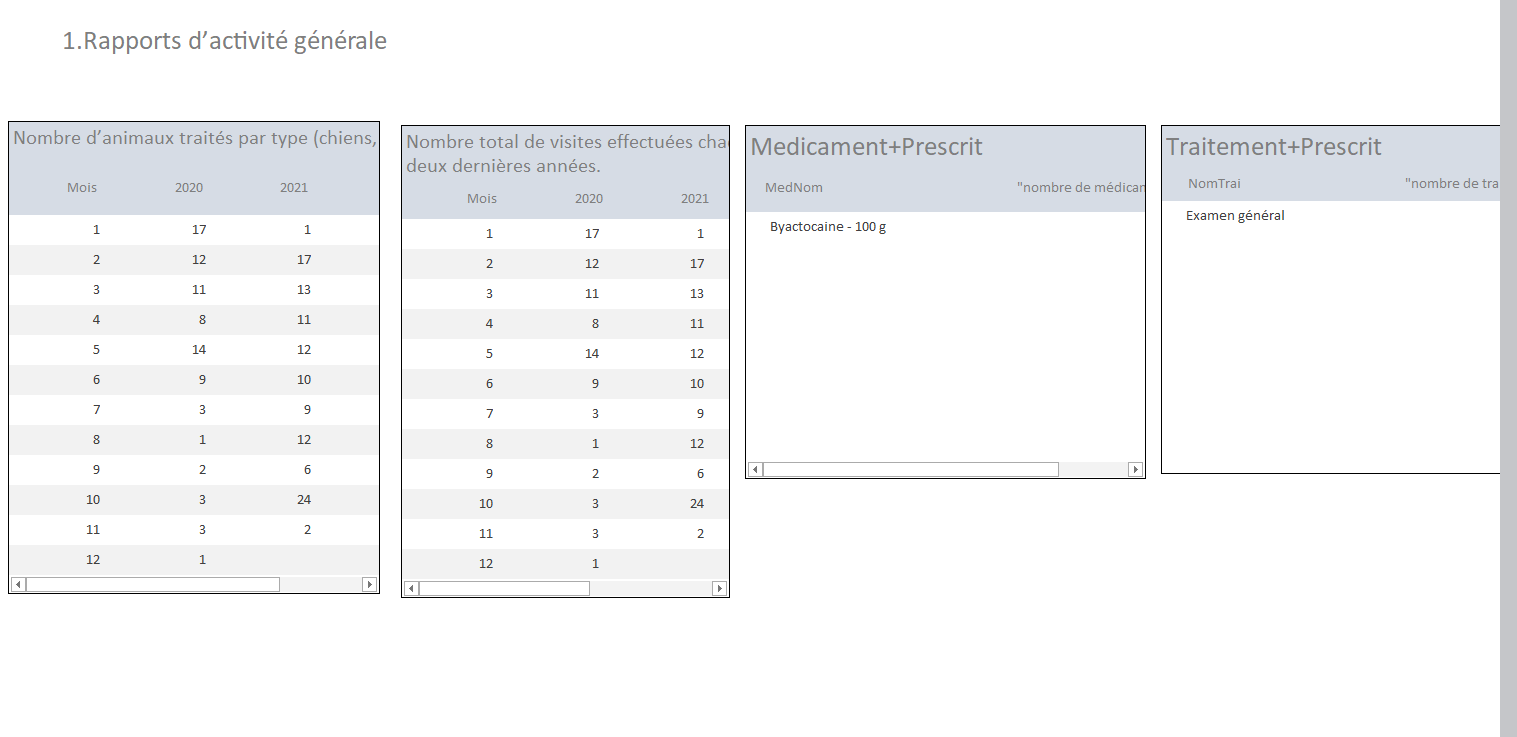
1. Requête donnant la liste des animaux ayant pris le même médicament plus de trois fois dans l'année.

|  |  |
| --- | --- |
| Requête | Résultat |
|  |  |

1. Requête pour calculer le coût total des traitements administrés sur une période donnée.

|  |  |
| --- | --- |
| Requête | Résultat |
| SELECT SUM(Traitement.PrixTrai) AS CoutTotalTraitement  FROM Visite INNER JOIN (Traitement INNER JOIN DetailVisite ON Traitement.CodeTrai = DetailVisite.VisTrait) ON Visite.VisNum = DetailVisite.VisNum  WHERE Visite.VisDate BETWEEN #2021-01-01# AND #2021-12-31#  ORDER BY SUM(Traitement.PrixTrai) DESC; |  |

1. **Rapports d’activité générale**



### **Partie 2 : Reporting sur Access**

Après avoir extrait les données via SQL, créez des reportings de suivi de l'activité de la clinique dans Access. Le directeur de la clinique souhaite obtenir les rapports suivants :

1. **Rapports d’activité générale**
   * Nombre d’animaux traités par type (chiens, chats, etc.).
   * Nombre total de visites effectuées chaque mois pour les deux dernières années.
   * Médicaments et traitements les plus prescrits.
2. **Rapports financiers**
   * Chiffre d’affaires total généré par client.
   * Chiffre d’affaires par type de traitement et médicament.
   * Coût moyen des traitements par catégorie d'animal.
3. **Évolution des indicateurs dans le temps**
   * Suivi mensuel du nombre de visites et des dépenses par client.
   * Nombre de traitements par trimestre et par année, avec calcul d’évolution.
4. **Rapports de fidélité client** 
   * Liste des clients avec leur ancienneté dans la clinique et le nombre de visites.
   * Clients ayant bénéficié de remises et le montant total de leur remise.

Ces rapports doivent être présenté sous forme de tableaux et graphiques croisés dynamiques pour faciliter leur analyse. Deux de ces rapports doivent inclure une mesure d’objectif cible (par exemple, une progression d’au moins 5 % sur l’année précédente).

### **Livrables**

1. **Notepad++**: Un fichier de travail contenant les requêtes SQL.
2. **Access** : Un fichier Access contenant toutes les requêtes et les reportings demandés.
3. **Word** : Un document reprenant les requêtes avec capture d’écran des résultats, les états avec les explications ainsi que l’interprétation des résultats pour la gestion de la clinique.

Cette SAE vous permettra de développer des compétences en SQL et en reporting sur Access pour des analyses approfondies et du reporting.

**Partie 2 : Reporting sur Access**

Excel

31/12/2024 2h