

SGBDR ORACLE

TP – B.D BALNEO

Introduction :

L'objectif du TP est d'appliquer les concepts de création/manipulation de base de données vue en cours à travers un cas pratique de mise en situation réel.

Notre cas : BALNEO met en scène une coopérative de propriétaire d'une station balnéaire.

Notre objectif est de synthétiser une base de données pour la coopérative afin qu'elle automatise la gestion des services de réservations et facturation.

Le TP s'organise en plusieurs séances. Tout d'abord la synthèse du MCD/MPD Balnéo. Puis la création de la BD et enfin le développement de scripts exécutant des requêtes spécifiques de consultation.

I) Séance1

Architecture de la base de données

Le document explication cas Balnéo et le script de création de la BD nous ont servis de base de départ. Grâce à ces informations nous avons pu définir les différentes Tables de la BD

Tables existantes dans le script de création :

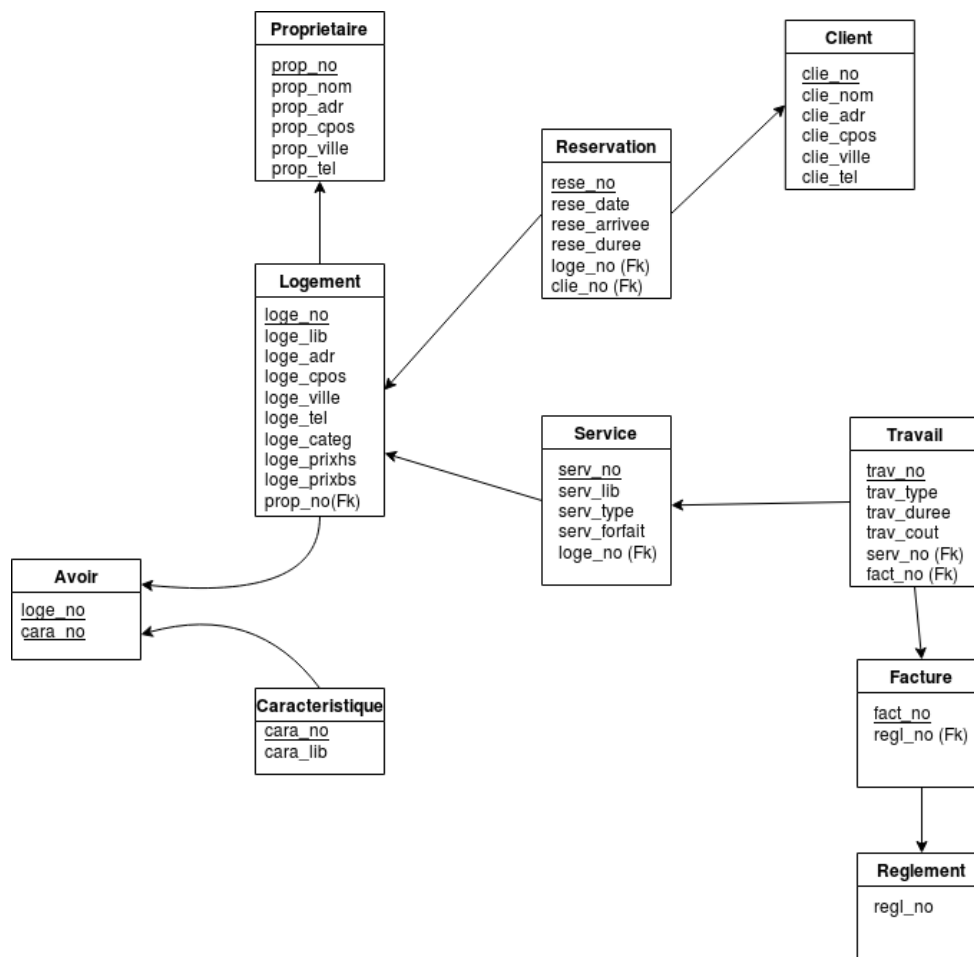
- ⑩ Propriétaire
- ⑩ Logement
- ⑩ Service
- ⑩ Client
- ⑩ Reservation
- ⑩ Avoir
- ⑩ Caracteristique

Tables rajoutés à la suite des explications du cas Balnéo :

- ⑩ Travail
- ⑩ Facture
- ⑩ Reglement

En complément de ces informations, nous nous sommes inspirés du script d'insertion pour concevoir le MPD de la BD.

Modèle physique de donnée (MPD)

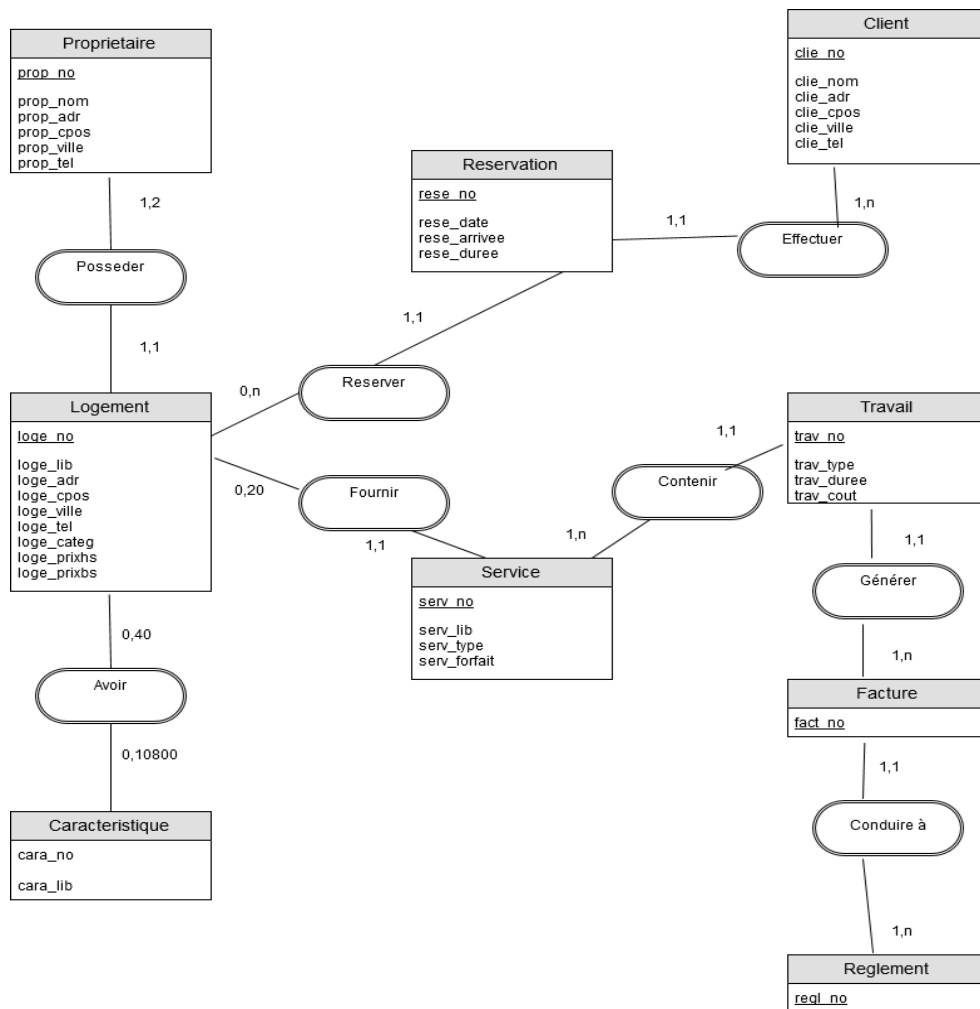


- ⑩ Les clés primaires des tables sont soulignées et les clés étrangères sont suivies d'un (Fk).
- ⑩ Après demande de renseignements, on nous a assurés que le script de Création était complet en ce qui concernés les tables déjà inscrites. Nous n'avions qu'à rajouter les tables manquantes.
- ⑩ De ce fait nous nous sommes appuyés sur les Pk et Fk pour établir l'architecture global.
- ⑩ Nous avons y rajouter les tables Travail, Facture et Règlement.
- ⑩ Les tables Facture et Reglement ne contiennent que des Pk ou Fk contrairement à la table Travail contenant plusieurs champs demandés par la suite.

Ce premier modèle nous permet d'examiner les différentes associations :

- La table Avoir apparaît comme une table de jointure (car Pk composé de Fk) ce qui traduit une association de type -..n / -..n
- Une table possédant une Fk étant la Pk d'une table qui lui est lié, apparaît comme le « maitre » de cette table et traduit une relation -..n / -,,1
- La table facture est « l'esclave » de la table Travail et non Service car la Pk de Facture n'était pas présente dans la table Service.
- ⑩ A partir de ces informations nous pouvons établir le MCD avec les associations et cardinalités correspondantes.

Modèle conceptuel de donnée (MPD)



Les cardinalités numériques sont données d'après l'explication Balneo :

- ⑩ Un propriétaire peut posséder maximum 2 logement.
- ⑩ Un logement peut avoir jusqu'à 40 caractéristiques.
- ⑩ Une caractéristique peut être commune à tout les logements, soit 10800 max.
- ⑩ Un peut avoir jusqu'à 20 services différents.

Maintenant que les schémas ont été complétés nous pouvons nous intéresser aux scripts SQL. En effet les schémas nous permettent de visualiser l'architecture global de la base de données. Maintenant nous allons créer notre Base de données sur SQL Plus.

Architecture de la base de données

Présentation du support

Nous travaillons sur la base de données Oracle. Un compte à notre nom à été créé. Nous utilisons le terminal SQL*Plus permettant de saisir des lignes de commande ou script SQL afin d'accéder à notre BD. Cela permet de dialoguer avec le SGBDR.

Durant le TP, nous codions notre script SQL sur Notepad++ puis nous l'exécutons via SQL*Plus.

Création de la base de données

Un script de création de la base de données nous a été fourni sur moodle. Celui-ci est censé être correct. Cependant il manque les contraintes d'intégrités et les tables que nous avons rajouter. De plus nous avons rajoutés les **NOT NULL**. Puis on corrige si le SGBDR nous renvoie des erreurs.

⑩ Ajout de table

Sur le script « Creation_balneo.sql » nous avons rajouter les tables Facture, Reglement, Travail selon la syntaxe suivante :

```
/*=====*/
/* Table : Facture */
/*=====*/

create table Facture (
    fact_no          NUMBER(4) not null
    ,
    regl_no          NUMBER(4) not null
    ,
    constraint PK_Facture primary key(fact_no)
    ,
    constraint FK_Facture_Reglement foreign key(regl_no) references Reglement(regl_no)
);
```

- Toujours avec la commande **CREATE TABLE** *nomtable*(...) pour créer la table.
- Puis déclaration des champs suivis de son type (NUMBER(X);VARCHAR(X);DATE;etc..).
- On rajoute pour chacune de nos valeurs le NOT NULL afin d'être toujours obliger d'insérer une valeur. Cela nous éviteras des erreurs dans la suite du TP.

⑩ Ajout des contraintes d'intégrités

Toujours en s'appuyant sur la capture précédente :

```
constraint PK_Facture primary key(fact_no)
constraint FK_Facture_Reglement foreign key(regl_no) references Reglement(regl_no)
```

On utilise le terme **constraint** *nomContrainte* pour définir la contrainte. Puis on définit son type et sa cible :

- **primary key** pour clé primaire puis on indique son champ correspondant
- **foreign key** pour clé étrangère en l'indiquant dans la table puis avec le **references** on indique sur quel table il renvoie et quel champs dans cette même table.

• Erreurs rencontrés

```
create table Travail <
*
ERREUR ó la ligne 1 :
ORA-00955: ce nom d'objet existe d'já
```

On a rencontré cette erreur car on a relancé le script de création de la BD. Alors que les tables ont déjà été créées. Pour résoudre ce problème nous avons supprimé nos tables avec :

DROP TABLE *matable*

Puis on relance le script.

⑩ Voir Annexe 1 pour le script

Création de la base de données

Un second script d'insertion a été fourni sur Moodle. Celui va permettre de remplir de lignes nos tables. Cependant ce script est composé de plusieurs erreurs qui seront à corriger. De plus nous devrons y rajouter des valeurs pour les tables que nous avons créées afin d'effectuer nos futures tests.

⑩ Ajout de valeurs dans nos tables

```
/*=====*/
/* Table : Travail */
/* trav_no N4, serv_no VA50, fact_no VA50, trav_type N6,2, trav_duree N6,2 ,trav_cout N6,2 */
/*=====*/

Insert into Travail values (1, 1, 1, 'pose d un parpaing', 2, 400, '15/09/2019');
Insert into Travail values (2, 2, 2, 'pose un mur en PVC', 1, 100, '01/10/2019');
Insert into Travail values (3, 5, 1, 'beton', 20, 4000, '15/09/2019');
Insert into Travail values (4, 5, 2, 'electricite', 10, 1000, '01/10/2018');
```

La syntaxe est composée de **INSERT INTO** maTable **VALUES**. On remplis notre table avec 4 lignes et on y insère des dates courantes 2019 afin de répondre aux futures exigences.

⑩ Erreurs rencontrées