Informatique PT. 2024 - 2025

Mathématiques & Informatique - F. Gaunard
`http://frederic.gaunard.com`
Lycée Voltaire, Paris 11e.

5 Travaux pratiques : Solution

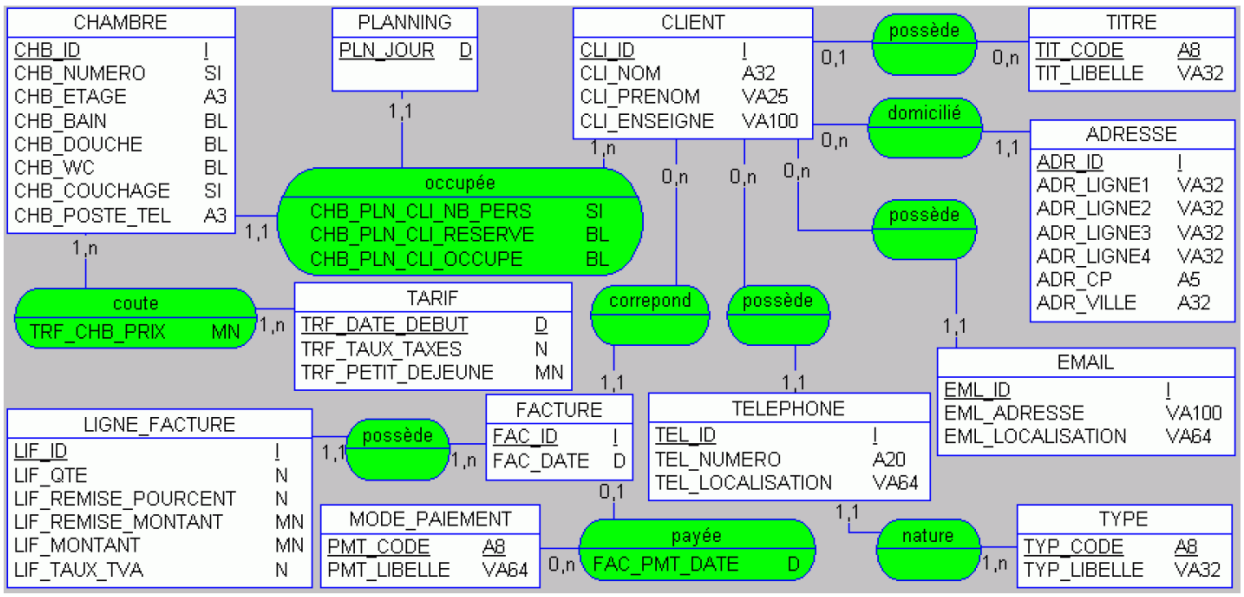
Un sujet de T.P proposé par **Nicolas Boucher**.

La base de donnée est à télécharger **ici** et à ouvrir avec SQLite.

Dans tout ce T.P, on retrouvera la terminologie *Entité*.
Une entité est un modèle de données qui se concrétise, une fois la modélisation réalisée (notamment après avoir déterminé de quelles données on a besoin), par une table (aussi appelée relation dans le cours).

Le contexte

On donne le schéma d'une base de données d'un hôtel :



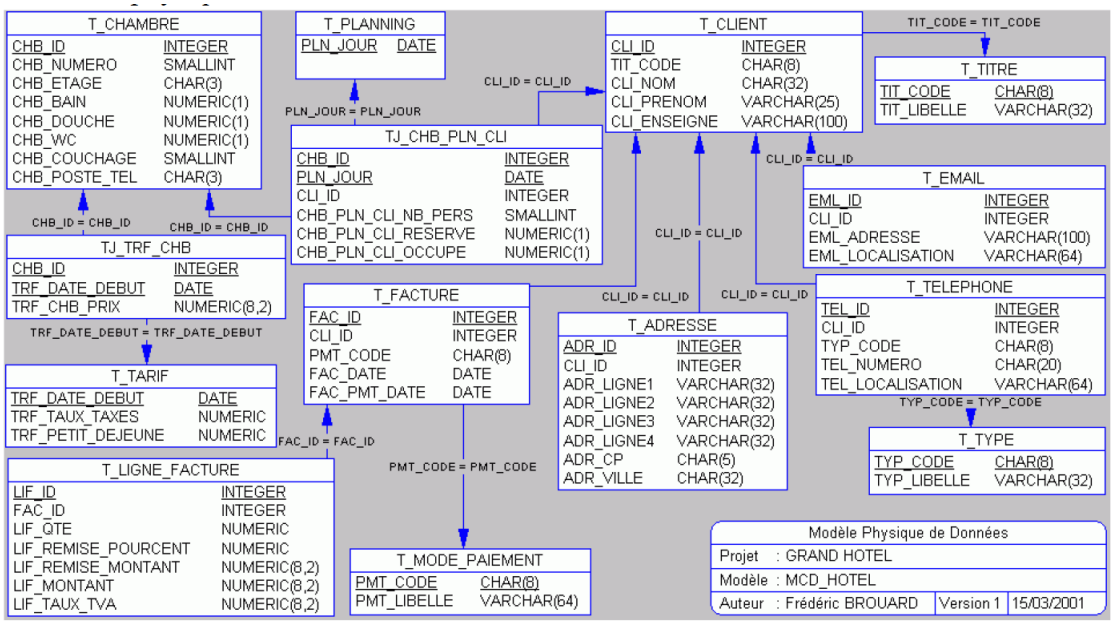
La plupart des clés sont des entiers (I) qui pourront être auto générés par exemple par autoincrément.
Pour certaines entités, notamment celles servant de références à la saisie (`MODE_PAIEMENT`, `TYPE`, `TITRE`) la clé est une chaîne de caractères.
Enfin pour les entités `TARIF` et `PLANNING`, il a été choisi une date comme clé.
Chaque entité est repérée à l'aide d'un î (code de 3 lettres) qui sert de préfixe pour chaque attribut.
Exemple : `CHB` pour `CHAMBRE`, `LIF` pour `LIGNE_FACTURE`, etc... Les booléens seront représentés par des valeurs numériques 0 (faux) et 1 (vrai), chaque attribut ayant obligatoirement une valeur par défaut.
L'association "occupée" permet de connaître la réservation ou l'occupation d'une chambre (une chambre peut avoir été réservée mais pas occupée), c'est pourquoi cette association possède les attributs `NB_PERS` (nombre de personnes : entier) `RESERVE` (réservée : booléen) et `OCCUPE` (occupe : booléen).

Une chambre à une date donnée, ne peut être occupée que par un seul client. Mais un client peut occuper plusieurs chambres à la même date ou la même chambre à différentes dates, voire même plusieurs chambres à plusieurs dates...

Définition des entités :

- ✗ Entité `CLIENT` : Un client peut avoir plusieurs adresses, plusieurs numéros de téléphone et plusieurs e-mail. Pour le téléphone, comme pour l'e-mail, l'attribut 'localisation' permet de savoir si le téléphone est situé au domicile, à l'entreprise, etc...
- ✗ L'entité `TITRE` permet de donner un titre à une personne, parmi les valeurs '`M.`' (monsieur), '`Mme.`' (madame) et '`Melle.`' (mademoiselle).
- ✗ L'entité `TYPE` permet de connaître le type de téléphone, parmi les valeurs '`TEL`' (téléphone), '`FAX`' (télécopie) et '`GSM`' (portable).
- ✗ L'entité `MODE_PAIEMENT` permet de connaître le genre de paiement, parmi les valeurs '`ESP`' (espèces), '`CHQ`' (chèque), '`CB`' (carte bancaire). L'association "payée" intègre la date du paiement d'une facture.

Le modèle physique de cette base de données est le suivant :



Il permet de faire apparaitre les clés primaires et les clés étrangères entre les différentes tables (relations) de la base de données.

Remarques préliminaires:

- ✗ Lorsqu'il y a des redondances, on pourra chercher à simplifier l'affichage du résultat avec la fonction `DISTINCT` (uniquement lorsque cela est nécessaire).
- ✗ Lorsqu'on utilisera une jointure, il faudra ne pas oublier la correspondance des clés entre tables avec la syntaxe `ON TABLE1.CLE1=TABLE2.CLE1` (on remarquera que les clés devant correspondre ont le même nom entre chaque table).
- ✗ Vous pouvez renommer un attribut (colonne) avec le mot clef `AS`, dans le résultat de la requête SQL.
- ✗ Vous pouvez rajouter autant de colonnes que vous le désirez en utilisant le même mot clef.
- ✗ En principe l'opérateur `AS` sert à donner un nom à de nouvelles colonnes créées par la requête.
- ✗ La syntaxe pour renommer une colonne de colonne1 à c1, d'une table contenant deux colonnes colonne c1 et colonne c2, est la suivante:

```
SELECT colonne1 AS c1, colonne2
FROM table
```

- ✗ La syntaxe pour renommer une table dans une requête est la suivante:

```
SELECT *
FROM 'nom_table' AS t1
```

Travail demandé - Solution des requêtes

Donner les requêtes permettant de :

1. Trouver les noms et prénoms des clients dont le titre est 'Mme.'.

```
SELECT CLI_NOM, CLI_PRENOM
FROM T_CLIENT
WHERE TIT_CODE = 'Mme.'
```

2. Donner la liste des identifiants des chambres dont le prix TRF_CHB_PRIX est supérieur à 400?.

```
SELECT DISTINCT CHB_ID
FROM TJ_TRF_CHB
WHERE TRF_CHB_PRIX>400
```

3. Classer par ordre alphabétique les clients de sexe masculin.

```
SELECT CLI_NOM, CLI_PRENOM
FROM T_CLIENT
WHERE  TIT_CODE = 'M.'
ORDER BY CLI_NOM, CLI_PRENOM
```

4. Donner les noms des clients et leurs numéros de téléphone.

```
SELECT CLI_NOM, TEL_NUMERO
FROM T_CLIENT JOIN T_TELEPHONE ON T_CLIENT.CLI_ID=T_TELEPHONE.CLI_ID
```

5. Identifier la ou les chambres dont le prix est maximum.

```
SELECT CHB_ID
FROM TJ_TRF_CHB
WHERE TRF_CHB_PRIX=(SELECT MAX(TRF_CHB_PRIX) FROM TJ_TRF_CHB)
```

6. Donner l'adresse e-mail de la personne ayant réservé la chambre 6 le 28 Janvier 2001.
Remarque: les dates sont notées sous le format: Année-mois-jour, soit pour le 28 janvier 2001 on aura 2001-01-28.

```
SELECT EML_ADRESSE
FROM T_EMAIL JOIN T_CLIENT JOIN TJ_CHB_PLN_CLI JOIN T_CHAMBRE
ON (T_CHAMBRE.CHB_ID= TJ_CHB_PLN_CLI.CHB_ID AND T_EMAIL.CLI_ID=T_CLIENT.CLI_ID AND
T_CLIENT.CLI_ID=TJ_CHB_PLN_CLI.CLI_ID)
WHERE CHB_NUMERO=6 AND PLN_JOUR="2001-01-28"
```

7. Donner le nombre de chambre par étage.

```
SELECT COUNT(*) AS NOMBRE, CHB_ETAGE
FROM T_CHAMBRE
GROUP BY CHB_ETAGE
```

8. Donner la liste des identifiants des chambres occupées et non réservées le 28 Janvier 2001.

```
SELECT CHB_ID
FROM TJ_CHB_PLN_CLI
WHERE CHB_PLN_CLI_OCCUPE=1 AND PLN_JOUR="2001-01-28"
AND CHB_PLN_CLI_RESERVE=0
```

9. Donner le nombre de chambres occupées et non réservées le 28 janvier 2001.

```
SELECT COUNT(*) AS NOMBRE
FROM TJ_CHB_PLN_CLI
WHERE CHB_PLN_CLI_OCCUPE=1 AND PLN_JOUR="2001-01-28"
AND CHB_PLN_CLI_RESERVE=0
```

10. Donner le nom du client occupant la chambre 19 le 28 Janvier 2001.

```
SELECT CLI_NOM
FROM TJ_CHB_PLN_CLI JOIN T_CLIENT ON
TJ_CHB_PLN_CLI.CLI_ID=T_CLIENT.CLI_ID
WHERE CHB_ID=19 and PLN_JOUR='2001-01-28'
```

11. Donner le nombre de chambres occupées par étage le 28 janvier 2001.

```
SELECT COUNT(*) AS NOMBRE, CHB_ETAGE
FROM T_CHAMBRE JOIN TJ_CHB_PLN_CLI ON
T_CHAMBRE.CHB_ID=TJ_CHB_PLN_CLI.CHB_ID
WHERE CHB_PLN_CLI_OCCUPE=1 AND PLN_JOUR="2001-01-28"
GROUP BY CHB_ETAGE
```

12. Donner le montant total qu'a rapporté la chambre 8 depuis la création de la base de donnée.

```
SELECT SUM(LIF_MONTANT)
FROM T_LIGNE_FACTURE JOIN T_FACTURE JOIN T_CLIENT JOIN
TJ_CHB_PLN_CLI ON (T_LIGNE_FACTURE.FAC_ID=T_FACTURE.FAC_ID AND
T_FACTURE.CLI_ID=T_CLIENT.CLI_ID AND
T_CLIENT.CLI_ID=TJ_CHB_PLN_CLI.CLI_ID)
WHERE CHB_ID=8
```