



## Module Base de données Corrigé de l'examen de moyenne durée

### Partie théorique (04 points)

1- Donner 2 avantages d'un SGBD par rapport à un système de gestion de fichiers classique.

- Intégrité des données (l'information est stockée une seule fois)
- Sécurité de l'accès aux données
- Instructions de traitement puissantes : SQL

2- Quelle est la différence entre une clé candidate et une clé primaire ?

- Clé candidate : clé susceptible d'être une clé primaire ou sous ensemble minimal d'attributs qui permet d'identifier chacun des tuples (lignes) d'une relation
- Clé primaire : clé choisie comme identifiant privilégié parmi les clés candidates

3- Donner un exemple de clé étrangère en indiquant son utilité.

- EMPLOYE (matricule\_nom, numero\_dept)
- DEPARTEMENT (numero\_dept, libelle)
- numero\_dept dans EMPLOYE est une clé étrangère de numero\_dept de DEPARTEMENT  
Il assure la contrainte d'intégrité référentielle

4- Donner les acronymes des termes suivants :

- a. SGBD : Système de Gestion des Bases de Données
- b. DF ; Dépendance Fonctionnelle
- c. SQL : Structured Query Language
- d. E-A : Entité Association

### Exercice N° 01 (04 points)

On considère une relation R (A, B, C, D, E, G, H)

Soit F l'ensemble des dépendances fonctionnelles (DF) associé à R :

| A, B → C      | B → D      | C, D → E      | C, E → G      | C, E → H      | G → A

- 1- Démontrer que (B, G) est une clé de R.
- 2- La relation R est-elle en deuxième forme normale ? En troisième forme normale ?
- 3- Donner si besoin une décomposition de R en relations en troisième forme normale. Les relations obtenues sont-elles toutes en BCNF (forme normale de Boyce-Codd) ?



### Corrigé type Exercice N° 04

#### 1. Pour monter que B,G est une clé nous devons montrer que :

B,G → A

B,G → B

B,G → C

B,G → D

B,G → E

B,G → G

B,G → H

- Par énoncé on a

| A, B → C 1 | B → D 2 | C, D → E 3 | C, E → G 4 | C, E → H 5 | G → A 6

- Par réflexivité nous obtenons

B,G → B 7

B,G → G 8

- Par augmentation de 2 par G nous obtenons B,G → D,G et par distributivité nous obtenons B,G → D (9) et B,G → G

- Par augmentation de 6 par B nous obtenons B,G → B,A et par distributivité nous obtenons B,G → A (10) et B,G → B

- Par augmentation de 6 par B nous obtenons B,G → B,A et par transitivité avec 1 nous obtenons B,G → C (11).

- Par union de 11 et 9 nous obtenons B,G → C,D et par transitivité avec 2 nous obtenons B,G → E (12)

- Par union de 11 et 12 nous obtenons B,G → C,E et par transitivité avec 5 nous obtenons B,G → H (13)

- A partir de 7,8,9,10,11,12,13 nous pouvons confirmer que B,G est la clé de R donc R s'écrit comme suit ; R (B,G, A,C, D, E, H).

#### 2. La relation R est-elle en deuxième forme normale ? En troisième forme normale ?

- Supposons que les valeurs des attributs sont atomiques donc R est 1FN.

- R n'est pas en 2FN parqu'il existe une partie de la clé (B, G) qui détermine une partie non clé (D, A) par les DFs 2 et 6.

- Comme R n'est pas en 2FN donc R n'est pas en 3FN.

#### 3. Décomposition de la relation R pour la rendre en 2FN.

R1 (B,G, C,E, H).

R2(B,D)

R3(G,A)

On remarque que R2 et R3 sont 3 FN et en BCNF

R1 (B,G, C,E, H) n'est pas en 3FN parce qu'il existe une partie non clé (C,E) qui détermine une partie non clé.

Décomposition de R1 pour la rendre en 3FN

R11 (B,G, C,E).

R12(C,E,H)

Donc R11 et R12, R2, R3 sont en BCNF.



une relation est FNBC si, pour toute dépendance fonctionnelle  $X \rightarrow A$  s'appliquant sur R avec A non inclus dans X, X est clé (primaire ou candidate) de R.

**Exercice N° 02 (04 points)**

Soit le modèle relationnel suivant :

AVION ( numero\_immat, localisation, id\_type )

TYPE ( id\_type, nom, poids, capacite, rayon\_action )

EXPERTISE ( id\_type, matricule, date\_debut, date\_fin )

PERSONNE ( matricule, nom, adresse, tel, salaire, examen\_medical, type\_personne )

où type\_personne = TECHNICIEN ou PILOTE

VOL ( numero\_vol, numero\_immat, matricule, ville\_dep, ville\_arr, h\_dep, h\_arr )

TEST ( numero\_test, nom, seuil, date, etat, numero\_immat )

**En utilisant le SQL effectuer les opérations suivantes :**

1- Donner la requête SQL de création de la table AVION

```
create table AVION (  
    numero_immat varchar(20) not null,  
    localisation varchar(50) not null,  
    id_type integer not null,  
    primary key (numero_immat),  
    foreign key(id_type) references type(id_type)  
);
```

2- Donner la requête SQL de remplissage d'une ligne pour cette même table

```
insert into AVION values ( '1234XPZ', 'Bordeaux', 2 );
```

3- Donner la liste des pilotes en service et les villes de départ de leurs vols.

- `select matricule, ville_dep from vol ;`

ou

- `select vol.matricule, ville_dep from vol, pilote  
where vol.matricule = pilote.matricule ;`

4- Donner par type d'avion, le nombre de vols au départ d'ALGER



```
• select t.nom, count(*)  
  from vol v, avion a, type t  
 |where a.id_type = t.id_type  
  and a.numero_immat = v.numero_immat  
  and ville_dep = 'ALGER'  
  group by t.id_type ;
```

5- les numéros d'immatriculation et le type des avions dont la moyenne de la durée des vols est inférieure à 45 minutes. On supposera que tous les vols ont lieu dans la même journée.

```
• select a.numero_immat, t.nom, avg((h_arr-h_dep)*60) as moyenne_min  
  from vol v, avion a, type t  
  where v.numero_immat = a.numero_immat  
  and a.id_type = t.id_type  
  group by a.numero_immat, t.id_type  
  having moyenne_min < 45;
```

### Exercice N° 03(08 points)

Nous sommes sur le centre de production d'un constructeur automobile. Deux chaînes effectuent le montage. Les véhicules sont ensuite transportés chez les distributeurs par un service livraison/logistique. Les employés sont répartis dans des bâtiments parfois éloignées les uns des autres. On veut gérer, à l'aide d'une base de données relationnelle, l'accès aux différents parkings. On définit, pour chaque parking, les bâtiments qui sont accessibles à partir de ce parking. L'attribution des places de parking se fera en fonction du lieu d'affectation de l'employé. L'attribution dépend également de la marque du véhicule : certains parkings sont interdits aux véhicules de marques concurrentes.

Les employés peuvent obtenir des autorisations exceptionnelles de parking, par exemple s'ils participent à une réunion dans un autre bâtiment que leur bâtiment habituel. On ne souhaite pas gérer le calendrier des réunions avec les salles correspondantes. La pertinence d'une demande exceptionnelle reste donc en dehors du système informatisé. Pour suivre la bonne affectation des autorisations exceptionnelles, on enregistre l'utilisation réelle de la place attribuée.

On suppose que :

- chaque bâtiment est proche d'au moins un parking,
- pour toute marque, il y a au moins un parking accessible,
- un employé n'est basé à un moment donné que dans un seul bâtiment,
- un seul propriétaire du véhicule est déclaré,
- un employé ne peut pas déclarer plusieurs véhicules,
- le système propose une ou plusieurs affectations, s'appuyant sur les règles en vigueur,
- toute affectation doit être confirmée manuellement.

a) Proposez une modélisation conceptuelle entité-association de cette base en précisant vos hypothèses complémentaires éventuelles.

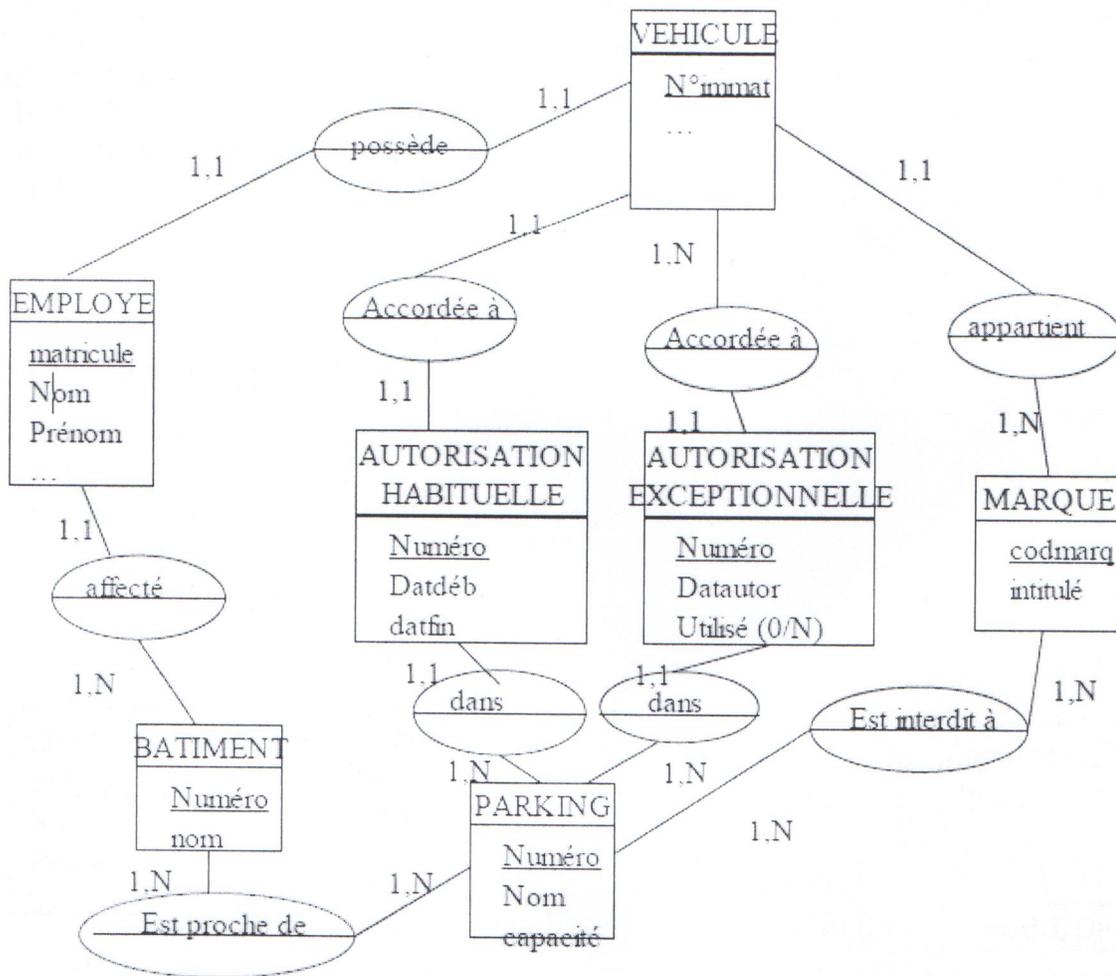


b) En expliquant la méthode utilisée, traduisez ce schéma en un ensemble de tables relationnelles.

c) Exprimer en SQL :

- 1- La liste des employés.
- 2- Le nombre d'employés par bâtiment
- 3- Le nombre de véhicule par parking.

**Corrigé type Exercice N° 04**





Hypothèses complémentaires :

- chaque véhicule a une et une seule autorisation habituelle de parking
- un parking peut être proche de plusieurs bâtiments

EMPLOYE(matricule, nom, prénom, n°bâtiment, n°autorhabit, n°immat, codmarq)

BATIMENT(numbâtiment, nombâtiment)

PARKING (numeroparking, nomparking, capacité)

MARQUE (codmarq, intitulé)

AUTORHABIT(n°autorhabit, datdeb, n°parking)

AUTOREXCEPT(n°autorexcept, datautor, n°immat, n°parking)

INTERDICTION (n°parking, codmarq)

PROXIMITE (n°parking, n°bâtiment)

Requête SQL :

1- La liste des employés.

**Select \* form employe**

2- Le nombre d'employés par bâtiment

**Select count(\*) , numbâtiment from employe group by numbâtiment**

3- Le nombre de véhicule par parking.

4- **Select count(\*) , numeroparking from parking group by numeroparking.**