|  |  |
| --- | --- |
| logo_IUT_Annecy_RVB | **LP DIM - Bases de données – Correction TD2**  LMD & LDD (Partie 1) |

**Q1 :**

**Liste des propriétés et entités :**

|  |  |  |
| --- | --- | --- |
| Terme | Type | Commentaire |
| Personne | Entité | =Client |
| Sommet | Entité |  |
| Client | Entité |  |
| Nom sommet | Propriété |  |
| Altitude sommet | Propriété |  |
| Référence IGN | Propriété |  |
| Titre carte | Propriété |  |
| Carte | Entité |  |
| Itinéraire | Entité |  |
| Titre itinéraire | Propriété |  |
| Code difficulté itinéraire | Propriété |  |
| Orientation itinéraire | Entité | Car plusieurs orientations sont possibles (N, E,…) et on souhaite avoir une liste déroulante |
| Dénivelé itinéraire | Propriété |  |
| Temps de parcours théorique itinéraire | Propriété |  |
| Sortie | Entité |  |
| Date sortie | Propriété |  |
| Météo sortie | Propriété |  |
| Commentaire sortie | Propriété |  |
| Durée itinéraire par sortie | Propriété |  |
| Nom client | Propriété |  |
| Prénom client | Propriété |  |
| Adresse client | Propriété | Il faudra décomposer l’adresse. |
| Téléphone fixe client | Propriété |  |
| Mail client | Propriété |  |
| Date naissance | Propriété | Et non pas âge |
| Participant | ? | Entité ou propriété ? |
| N° sommet | Propriété |  |
| N° itinéraire | Propriété |  |
| Difficulté | Entité | Car plusieurs difficultés sont possibles (AD, AD-,…) |

**Dans un MCD, on sort en général tout => sortir Orientation et Difficulté !**

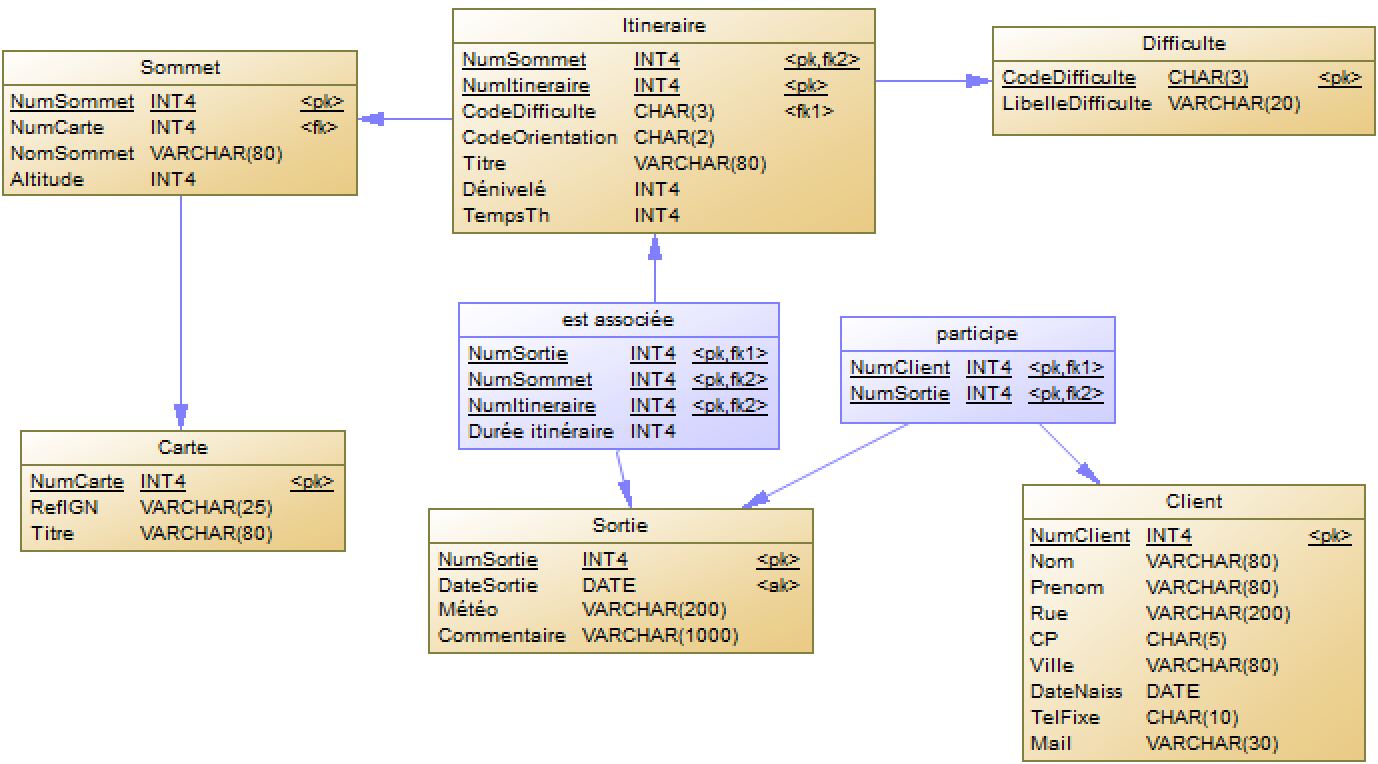
**MCD :**

****

* Penser à décomposer l’adresse en rue, CP et Ville car une propriété doit être atomique (non décomposable).
* Dans un MCD, on sort Orientation et Difficulté.
* (1,1) : identifiant relatif. L’entité Itinéraire est dépendante de l’entité Sommet. Le n° de l’itinéraire est relatif au sommet.

**MLDR :**

Consiste à créer le modèle relationnel.

****

**Remarque : INT4 ou INTEGER**

**Questions à (se) poser :**

* **Pourquoi le téléphone est-il de type char ?** Sinon le 0 de début disparaît (idem pour un code postal français).
* **Pourquoi le CP est-il de type char et non varchar ?** Un caractère de longueur fixe tient moins de place mémoire qu’un caractère de longueur variable. En outre, un CP ne peut contenir que 5 caractères (en l’occurrence 5 chiffres) ni plus ni moins.
* **Combien y a-t-il de clés primaires dans la table Participe ?** 1 seule composée de 2 champs !

**MPD :**

**Actions possibles :**

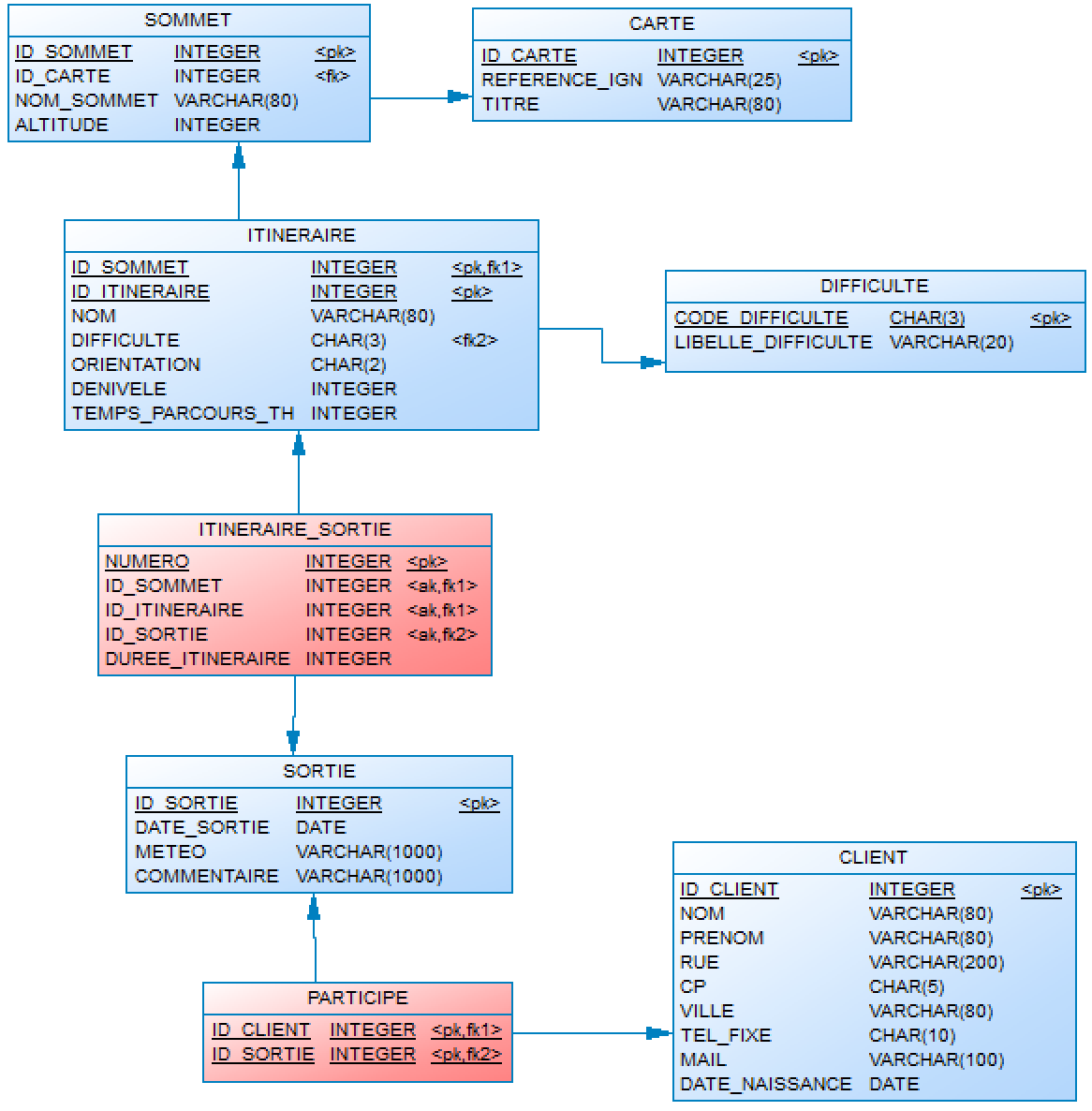
* Renommage des tables et/ou colonnes (peut également être fait à l’étape précédente) : supprimer les accents, espaces, caractères spéciaux dans les noms des colonnes et tables. Il est coutumier d’utiliser des substantifs (noms communs) pour qualifier une table. Le nom de la table PARTICIPE a volontairement été laissé ainsi afin de la renommer ultérieurement.
* Conservation ou non des tables de référence ?
  + Il est normal de se poser une question concernant les niveaux de difficulté des itinéraires : est-ce que cette liste est susceptible d'évoluer ou bien est-elle figée ? De même pour les orientations ? Les difficultés peuvent évoluer au fil du temps, même si ces modifications sont rares, elles ne sont pas impossibles. Il est donc nécessaire de créer une table pour gérer les difficultés.
  + Le même type de question doit être posé en ce qui concerne l’orientation des itinéraires. Cette fois-ci, il est logique de penser que ces orientations sont pleinement définies et qu'il n'y aura pas d'ajout ou suppression. Les différentes orientations pour un itinéraire vont donc être gérées par une contrainte de type check (contrainte de validation) définie sur la colonne.

*Remarque : dans PowerAMC, pour ne pas générer une entité, il suffit au niveau du MCD de décocher « Générer » (propriété de l’entité).*

**Questions à (se) poser :**

* **Pourquoi doit-on toujours conserver une table de référence si les valeurs de ses champs ne sont pas figées ?**
  + - Pour éviter les doublons sur des champs de type caractère variable (ou caractère) dans la table contenant la clé étrangère **=>** gain en volumétrie (seul le n° ou le code est doublonné, et non le libellé, en général plus long).
    - Pour éviter les erreurs de saisie (une seule saisie réalisée dans la table de référence).
    - Pour avoir la possibilité de disposer d’une liste déroulante dans l’application cliente et donc de sélectionner la valeur à partir de la table difficulté, sans avoir à la saisir…
* Table « EST ASSOCIEE » (à renommer en ITINERAIRE\_SORTIE) : il est préférable de limiter les clés primaires portant sur plusieurs champs à 2 champs au maximum pour éviter de charger de gros index en mémoire lors des jointures. On créé un champ « Numéro » qui devient clé primaire. Ce champ s’incrémente automatiquement à chaque association itinéraire / sortie. On créé un index unique (ou clé unique) sur les champs (ID\_SOMMET, ID\_ITINERAIRE et ID\_SORTIE) que l’on force à NOT NULL.

**ATTENTION : une clé primaire ne peut jamais être NULL contrairement à une clé unique => penser à forcer à NOT NULL l’ensemble des champs composant la clé unique.**

****

Tables **CARTE, CLIENT, DIFFICULTE, ITINERAIRE, ITINERAIRE\_SORTIE, PARTICIPE, SOMMET, SORTIE.**

Table **CARTE :**

|  |  |  |
| --- | --- | --- |
| ID\_CARTE | integer | Clé primaire. |
| REFERENCE\_IGN | varchar(25) |  |
| TITRE | varchar(80) |  |

Tous les champs sont obligatoires.

Table **CLIENT :**

|  |  |  |
| --- | --- | --- |
| ID\_CLIENT | integer | Clé primaire. |
| NOM | varchar(80) |  |
| PRENOM | varchar(80) |  |
| RUE | varchar(200) |  |
| CP | char(5) |  |
| VILLE | varchar(80) |  |
| TEL\_FIXE | char(10) |  |
| MAIL | varchar(30) |  |
| DATE\_NAISSANCE | date |  |

Tous les champs sont obligatoires, sauf MAIL.

*Remarque : on peut utiliser integer ou int4.*

Table **DIFFICULTE :**

|  |  |  |
| --- | --- | --- |
| CODE\_DIFFICULTE | char(3) | Clé primaire |
| LIBELLE\_DIFFICULTE | varchar(20) |  |

Tous les champs sont obligatoires.

Table **ITINERAIRE :**

|  |  |  |
| --- | --- | --- |
| ID\_SOMMET | integer | Participe à la clé primaire.  Clé étrangère qui référence la table des sommets. |
| ID\_ITINERAIRE | integer | Participe à la clé primaire. |
| NOM | varchar(80) |  |
| DIFFICULTE | char(3) | Clé étrangère qui référence la table des difficultés. |
| ORIENTATION | char(2) | Contrainte de validation pour permettre uniquement les valeurs suivantes : 'N', 'NE', 'E', 'SE', 'S', 'SO', 'O', 'NO'. |
| DENIVELE | integer |  |
| TEMPS\_PARCOURS\_TH | integer |  |

Tous les champs sont obligatoires.

Table **ITINERAIRE\_SORTIE :**

|  |  |  |
| --- | --- | --- |
| NUMERO | integer | Clé primaire. |
| ID\_SOMMET | integer | Participe à la clé unique.  Participe à la clé étrangère qui référence la table des itinéraires (et non des sommets !). |
| ID\_ITINERAIRE | integer | Participe à la clé unique.  Participe à la clé étrangère qui référence la table des itinéraires. |
| ID\_SORTIE | integer | Participe à la clé unique.  Clé étrangère qui référence la table des sorties. |
| DUREE\_ITINERAIRE | integer |  |

Tous les champs sont obligatoires.

Table **PARTICIPE :**

|  |  |  |
| --- | --- | --- |
| ID\_CLIENT | integer | Participe à la clé primaire.  Clé étrangère qui référence la table des clients. |
| ID\_SORTIE | integer | Participe à la clé primaire.  Clé étrangère qui référence la table des sorties. |

Tous les champs sont obligatoires.

Table **SOMMET :**

|  |  |  |
| --- | --- | --- |
| ID\_SOMMET | integer | Clé primaire. |
| ID\_CARTE | integer | Clé étrangère qui référence la table des cartes. |
| NOM\_SOMMET | varchar(80) |  |
| ALTITUDE | integer |  |

Tous les champs sont obligatoires.

Table **SORTIE :**

|  |  |  |
| --- | --- | --- |
| ID\_SORTIE | integer | Clé primaire. |
| DATE\_SORTIE | Date |  |
| METEO | varchar(1000) |  |
| COMMENTAIRE | varchar(1000) |  |

Tous les champs sont obligatoires sauf METEO et COMMENTAIRE.

**Q2 :**

drop table if exists CARTE cascade;drop table if exists CLIENT cascade;drop table if exists DIFFICULTE cascade;drop table if exists ITINERAIRE cascade;drop table if exists ITINERAIRE\_SORTIE cascade;drop table if exists PARTICIPE cascade;

drop table if exists PARTICIPATION cascade;

drop table if exists SOMMET cascade;

drop table if exists SORTIE cascade;

/\*==============================================================\*/

/\* Table : CARTE \*/

/\*==============================================================\*/

create table CARTE (

ID\_CARTE INTEGER not null,

REFERENCE\_IGN VARCHAR(25) not null,

TITRE VARCHAR(80) not null,

constraint CARTE\_PK primary key (ID\_CARTE)

);

/\*==============================================================\*/

/\* Table : CLIENT \*/

/\*==============================================================\*/

create table CLIENT (

ID\_CLIENT INTEGER not null,

NOM VARCHAR(80) not null,

PRENOM VARCHAR(80) not null,

RUE VARCHAR(200) not null,

CP CHAR(5) not null,

VILLE VARCHAR(80) not null,

TEL\_FIXE CHAR(10) not null,

MAIL VARCHAR(30),

DATE\_NAISSANCE DATE not null,

constraint CLIENT\_PK primary key (ID\_CLIENT)

);

/\*==============================================================\*/

/\* Table : DIFFICULTE \*/

/\*==============================================================\*/

create table DIFFICULTE (

CODE\_DIFFICULTE CHAR(3) not null,

LIBELLE\_DIFFICULTE VARCHAR(20) not null,

constraint DIFFICULTE\_PK primary key (CODE\_DIFFICULTE)

);

/\*==============================================================\*/

/\* Table : ITINERAIRE \*/

/\*==============================================================\*/

create table ITINERAIRE (

ID\_SOMMET INTEGER not null,

ID\_ITINERAIRE INTEGER not null,

NOM VARCHAR(80) not null,

DIFFICULTE CHAR(3) not null,

ORIENTATION CHAR(2) not null

constraint ITINERAIRE\_ORIENTATION\_CK check (orientation IN ('N','NE','E','SE','S','SO','O', 'NO')),

DENIVELE INTEGER not null,

TEMPS\_PARCOURS\_TH INTEGER not null,

constraint ITINERAIRE\_PK primary key (ID\_SOMMET, ID\_ITINERAIRE)

);

/\*==============================================================\*/

/\* Table : ITINERAIRE\_SORTIE \*/

/\*==============================================================\*/

create table ITINERAIRE\_SORTIE (

NUMERO INTEGER not null,

ID\_SOMMET INTEGER not null,

ID\_ITINERAIRE INTEGER not null, ID\_SORTIE INTEGER not null, DUREE\_ITINERAIRE INTEGER not null, constraint ITINERAIRE\_SORTIE\_PK primary key (NUMERO),

constraint ITINERAIRE\_SORTIE\_UK unique (

ID\_SOMMET,

ID\_ITINERAIRE,

ID\_SORTIE

)

);

/\*==============================================================\*/

/\* Table : PARTICIPE \*/

/\*==============================================================\*/

create table PARTICIPE (

ID\_CLIENT INTEGER not null,

ID\_SORTIE INTEGER not null,

Constraint PARTICIPE\_PK primary key (ID\_CLIENT, ID\_SORTIE)

);

/\*==============================================================\*/

/\* Table : SOMMET \*/

/\*==============================================================\*/

create table SOMMET (

ID\_SOMMET INTEGER not null,

ID\_CARTE INTEGER not null,

NOM\_SOMMET VARCHAR(80) not null,

ALTITUDE INTEGER not null,

constraint SOMMET\_PK primary key (ID\_SOMMET)

);

/\*==============================================================\*/

/\* Table : SORTIE \*/

/\*==============================================================\*/

create table SORTIE (

ID\_SORTIE INTEGER not null,

DATE\_SORTIE DATE not null,

METEO VARCHAR(1000),

COMMENTAIRE VARCHAR(1000),

constraint SORTIE\_PK primary key (ID\_SORTIE)

);

alter table ITINERAIRE

add constraint ITINERAIRE\_SOMMET\_FK foreign key (ID\_SOMMET)

references SOMMET (ID\_SOMMET);

alter table ITINERAIRE

add constraint ITINERAIRE\_DIFFICULTE\_FK foreign key (DIFFICULTE)

references DIFFICULTE (CODE\_DIFFICULTE);

alter table ITINERAIRE\_SORTIE

add constraint ITINERAIRE\_SORTIE\_ITI\_FK foreign key (ID\_SOMMET, ID\_ITINERAIRE)

references ITINERAIRE (ID\_SOMMET, ID\_ITINERAIRE);

alter table ITINERAIRE\_SORTIE

add constraint ITINERAIRE\_SORTIE\_SORTIE\_FK foreign key (ID\_SORTIE)

references SORTIE (ID\_SORTIE);

alter table PARTICIPE

add constraint PARTICIPE\_CLIENT\_FK foreign key (ID\_CLIENT)

references CLIENT (ID\_CLIENT);

alter table PARTICIPE

add constraint PARTICIPE\_SORTIE\_FK foreign key (ID\_SORTIE)

references SORTIE (ID\_SORTIE);

alter table SOMMET

add constraint SOMMET\_CARTE\_FK foreign key (ID\_CARTE)

references CARTE (ID\_CARTE);

**Q5 :**

Create sequence CLIENT\_SEQ OWNED BY client.id\_client;

Select \* from information\_schema.sequences;

**Q6 :**

Alter table CLIENT add constraint CLIENT\_UQ unique (NOM, PRENOM, TEL\_FIXE);

**Q7 :**

Alter table CLIENT add MOBILE char(10);

Alter table CLIENT add constraint CLIENT\_MOBILE\_CK check (MOBILE like '06%' or MOBILE like '07%' or MOBILE IS NULL); -- et non (MOBILE like '06%' ~~OR '07%'~~)

**Q8 :**

Clés étrangères :

- Table SOMMET : colonne ID\_CARTE.

- Table ITINERAIRE : ID\_SOMMET, DIFFICULTE.

- Table ITINERAIRE\_SORTIE : (ID\_SOMMET, ID\_ITINERAIRE), ID\_SORTIE

- Table PARTICIPE : ID\_CLIENT, ID\_SORTIE

create index SOMMET\_IDCARTE\_IX on SOMMET (ID\_CARTE);

create index ITINERAIRE\_SOMMET\_IX on ITINERAIRE (ID\_SOMMET);

create index ITINERAIRE\_DIFFICULTE\_IX on ITINERAIRE (DIFFICULTE);

**create index ITINERAIRE\_SORTIE\_SOM\_ITI\_IX on ITINERAIRE\_SORTIE (ID\_SOMMET,ID\_ITINERAIRE);**

create index ITINERAIRE\_SORTIE\_IDSORTIE\_IX on ITINERAIRE\_SORTIE (ID\_SORTIE);

create index PARTICIPE\_IDCLIENT\_IX on PARTICIPE (ID\_CLIENT);

create index PARTICIPE\_IDSORTIE\_IX on PARTICIPE (ID\_SORTIE);

**Q9 :**

**Q10 :**

Alter table SORTIE alter column DATE\_SORTIE SET default current\_date;

**Q11 :**

Tout d'abord, supprimer la contrainte de clé étrangère qui existe entre les tables PARTICIPE et SORTIE.

Alter table PARTICIPE drop constraint PARTICIPE\_SORTIE\_FK;

Puis il faut définir une nouvelle contrainte de référence entre les deux tables, avec cette fois-ci l'option ON DELETE CASCADE.

Alter table PARTICIPE

add constraint PARTICIPE\_SORTIE\_FK foreign key (ID\_SORTIE)

references SORTIE (ID\_SORTIE) **on delete cascade**;

select \* from information\_schema.referential\_constraints;

*"postgres";"public";"*PARTICIPE\_SORTIE\_FK*";"postgres";"public";"pk\_sortie";"NONE";"NO ACTION";"CASCADE"*

**Q12 :**

Alter table PARTICIPE rename to PARTICIPATION;