

Devoir d'informatique

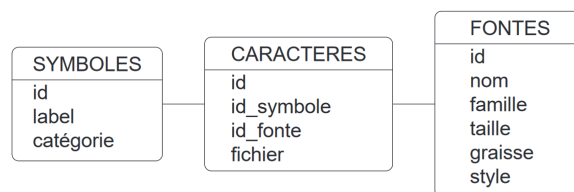
Durée : 2 heures

1 Reconnaissance de caractères

L'exercice proposé s'inscrit dans un sujet plus complet ayant pour objet la reconnaissance de caractères lors de la numérisation de documents.

Il faut alors comparer les caractères numérisés à une base de données.

Cette base de données contient des informations sur chaque caractère selon le type de fonte, la taille, la graisse... Trois tables sont utilisées.



La table SYMBOLES contient les attributs suivants :

- **id** : identifiant d'un symbole (entier), clé primaire ;
- **label** : nom du symbole ("A", "a", "1", "é", " !" ...) (chaîne de caractères) ;
- **catégorie** : parmi majuscule, minuscule, chiffre, spécial (dont accent) (chaîne de caractères).

La table CARACTERES contient les attributs suivants :

- **id** : identifiant d'un caractère (entier), clé primaire ;
- **id_symbole** : identifiant du nom du symbole (entier) ;
- **id_fonte** : identifiant du type de fonte (entier) ;
- **fichier** : nom du fichier image correspondant (chaîne de caractères).

La table FONTES contient les attributs suivants :

- **id** : identifiant d'une fonte (entier), clé primaire ;
- **nom** : nom de la fonte ("Arial", "Times new roman", "Calibri", "Zurich", ...) (chaîne de caractères) ;
- **famille** : nom de la famille dont fait partie la fonte ("humane", "garalde", "réale", "didone", "scripte", ...) (chaîne de caractères) ;
- **taille** : dimension en hauteur des caractères en pixels (entier) ;
- **graisse** : type de graisse ("léger", "normal", "gras", "noir", ...) (chaîne de caractères) ;
- **style** : type de style ("romain", "italique", "ombré", "décoratif", ...) (chaîne de caractères).

Q1. Écrire une requête SQL permettant d'extraire les identifiants des fontes dont le nom est "Zurich", de style "romain" et dont la taille est comprise entre 10 et 16 pixels.

Q2. Écrire une requête SQL permettant d'extraire tous les noms de fichiers des caractères qui correspondent au symbole de label "A".

Q3. Écrire une requête SQL permettant d'indiquer le nombre de caractères correspondant à la fonte "Zurich", de style "romain" et dont la taille est comprise entre 10 et 16 pixels groupés selon les labels des symboles.

2 Données géographiques

Les fichiers contenant des photos possèdent des informations sur celles-ci, elles sont communément appelées données exif.

Entre autres, on y trouve :

- la date où elle a été prise ;
- les coordonnées GPS de l'endroit où elle a été prise.

Le but de cet exercice est de manipuler les données associées à des photos.

Les tables utiles pour répondre aux questions sont données ci-dessous.

Nom	Date	Latitude	Longitude	Id
Photo10	2020/06/11	35937	-1149686	1
Photo11	2020/06/22	35937	-1149685	2
Photo24	2020/06/17	35937	-1149686	3
Photo15	2020/06/17	35938	-1149686	4
Photo33	2020/06/16	35939	-1149688	5
⋮	⋮	⋮	⋮	⋮

Table Photos

Id	Dossier	NomPhoto
1	C : \Images \LasVegas \Bellagio	Photo10
2	C : \Images \LasVegas \Palazzo	Photo11
3	C : \Images \LasVegas \Flamingo	Photo24
⋮	⋮	⋮
10	C : \Images \LasVegas \Flamingo	Photo40
11	C : \Images \LasVegas \Palazzo	Photo30
12	C : \Images \LasVegas \Flamingo	Photo15
⋮	⋮	⋮

Table Dir

Table	Champ	Type
Photos	Nom	Chaîne de caractères
Photos	Date	Chaîne de caractères
Photos	Latitude	Entier
Photos	Longitude	Entier
Photos	Id	Entier
Dir	Id	Entier
Dir	NomPhoto	Chaîne de caractères
Dir	Dossier	Chaîne de caractères

Type des données par Table

On remarquera que dans la table Photos, les latitudes et longitudes sont des entiers : une division par 10 000 permet d'avoir la latitude et la longitude réelle.

On notera que dans la table Photos, Date correspond à la date de création (année/mois/jour) de photos présentes dans le dossier (colonne nommée Dossier). On précise qu'à un nom de photo correspond une unique photo et que chaque dossier ne contient que des photos.

- Q1.** Écrire la requête SQL qui permet d'avoir le nom, la latitude et la longitude des photos faites le 2020/06/16.
- Q2.** Écrire la requête SQL qui permet d'avoir le nom des photos faites le 2020/06/16 et entre les latitudes 35935 et 35940.
- Q3.** Écrire la requête SQL qui permet de trouver le nom des photos prises le 2020/06/17 et situées dans le dossier C:\Images\Las Vegas\Bellagio.
- Q4.** Écrire la requête SQL qui permet de compter le nombre de photos situées dans le dossier C:\Images\Las Vegas\Bellagios.
- Q5.** Écrire la requête SQL qui permet de compter le nombre de photos situées dans chaque dossier. Cette requête reverra donc la table composée d'une première colonne dont le nom est le nom du dossier et d'une seconde colonne qui donne le nombre de photo pour le dossier en question.
- Q6.** Écrire la requête SQL correspondant à la question précédente mais en limitant l'affichage aux dossiers contenant plus de 50 photos en les classant en partant du dossier le moins rempli vers le dossier le plus rempli.

3 Gestion d'une entreprise

Une entreprise possède une base de données nommée `Gestion_Entreprise` constituée de trois tables : `clients`, `produits` et `ventes`. Les contenus de ces tables se trouvent à la fin de cet exercice.

La **table "clients"** est constituée de 5 champs :

- `id` : de type `INTEGER` – clé primaire auto-incrémentée ;
- `num_secu` : de type `INTEGER` – entier de 15 chiffres ;
- `nom` : de type `TEXT` ;
- `prenom` : de type `TEXT` ;
- `num_CB` : de type `INTEGER`.

La **table "produits"** est constituée de 5 champs :

- `id` : de type `INTEGER` – clé primaire auto-incrémentée ;
- `ref_produit` : de type `INTEGER` ;
- `nom_produit` : de type `TEXT` ;
- `qrcode` : de type `TEXT` ;
- `prix` : de type `DECIMAL`.

La **table "ventes"** est constituée de 3 champs :

- `date` : de type `TEXT` ;
- `ref_produit` : de type `INTEGER` – clé étrangère, pointe vers la clé primaire `id` de la table `produits` ;
- `num_client` : de type `INTEGER` – clé étrangère, pointe vers la clé primaire `id` de la table `clients`.

Les dates dans cette table sont définies par une chaîne de 10 caractères suivant le format année-mois-jour. Exemple de dates : "2019-06-01", "2022-12-30".

- Q1.** Écrire, en SQL, la requête (1) qui permet d'obtenir le numéro de carte de crédit de toutes les personnes référencées dans la base de données de l'entreprise dont le numéro de sécurité sociale commence par 2. On utilisera le caractère "_" comme le séparateur des milliers. Par exemple 10000000 sera réécrit comme 10_000_000.
- Q2.** Écrire, en SQL, la requête (2) permettant d'obtenir le nom et le prénom de toutes les personnes ayant effectué un achat avec un résultat sans doublon.
- Q3.** Écrire, en SQL, la requête (3) qui permet d'obtenir les produits associés à chaque numéro de carte de crédit du

client et qui ont été vendus entre le 1 juin 2020 et le 30 juillet 2020. On rappelle que SQL compare les variables de type TEXT grâce à l'ordre lexicographique : par exemple "13-06-1989 < 13-07-1999" est vrai.

Contenu de la base de données :

Table "clients"

id	num_secu	nom	prenom	num_CB
1	286128817863441	Eldyn	Sophie	6767342589219928
2	298082934500890	Gomez	Maria	2324563490665454
3	298082934500896	Ruiza	Flor	9889454573204522
4	109086723487917	Kovitz	Boris	6789543778653678
5	175105642102321	Mottreff	Erwan	4745342178563217
6	189027511732543	Settin	Michel	7856432167453492
7	191017511318196	Valérie	Georges	8787564521392354
...

Table "produits"

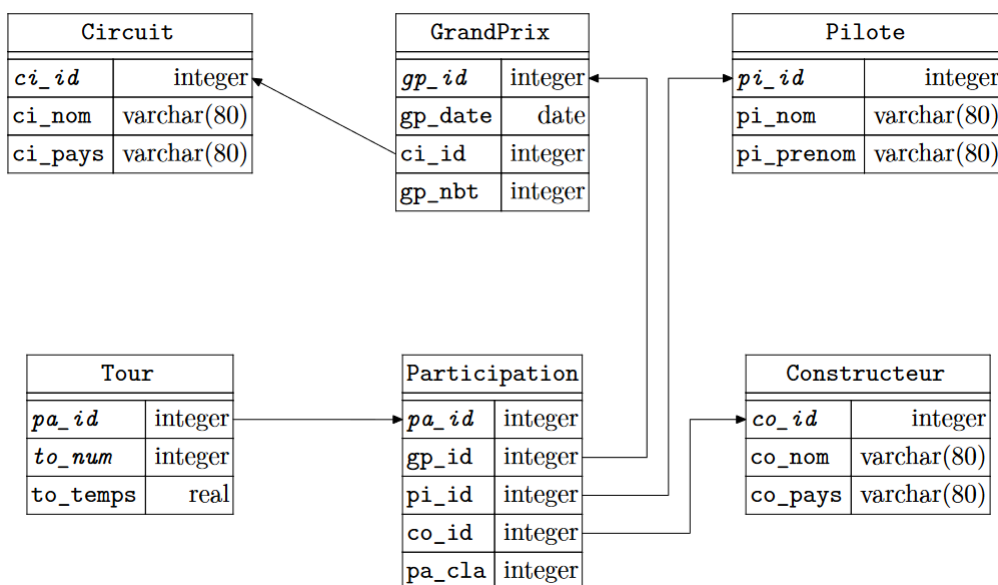
id	ref_produit	nom_produit	qrcode	prix
1	27	Buffet chêne	27.jpg	320
2	102	Chaise rustique	102.jpg	65
3	453	Table ronde	453.jpg	75
4	756	Table ovale	756.jpg	120
5	921	Coffret Bali	921.jpg	170
...

Table "ventes"

date	ref_produit	num_client
2020-05-15	2	2
2020-06-17	3	1
2020-06-21	3	7
2020-07-19	4	5
2020-08-19	5	5
2020-09-05	4	6
...

4 Résultats de championnat de formule 1

Depuis la création du championnat du monde de formule 1 en 1950, la fédération internationale de l'automobile (FIA) conserve l'ensemble des résultats des différents grands prix. Ces différentes informations sont stockées dans une base de données relationnelle dont un modèle physique simplifié est schématisé ci-dessous.



Cette base comporte les six tables listées ci-dessous avec la description de leurs colonnes :

- la table **Circuit** de circuits utilisés pour le championnat du monde de formule 1

- ci_id identifiant (entier arbitraire) du circuit (clef primaire)
- ci_nom nom du circuit (généralement le nom de la ville ou de la région dans laquelle il se trouve)
- ci_pays pays dans lequel se trouve le circuit
- la table GrandPrix des courses comptant pour le championnat du monde
 - gp_id identifiant (entier arbitraire) du grand prix (clef primaire)
 - gp_date date de la course
 - ci_id circuit sur lequel s'est déroulé la course
 - gp_nbt nombre de tours de circuit
- la table Pilote des pilotes de formule 1
 - pi_id identifiant (entier arbitraire) du pilote (clef primaire)
 - pi_nom nom de famille du pilote
 - pi_prenom prénom du pilote
- la table Constructeur des écuries de formule 1
 - co_id identifiant (entier arbitraire) de l'écurie (clef primaire)
 - co_nom nom de l'écurie
 - co_pays pays d'origine de l'écurie
- la table Participation des participants (couple pilote, constructeur) pour une course donnée
 - pa_id identifiant (entier arbitraire) de la participation (clef primaire)
 - gp_id la course
 - pi_id le pilote
 - co_id le constructeur
 - pa_cla le classement à l'arrivée de ce participant ou NULL s'il n'a pas terminé la course ou a été disqualifié
- la table Tour des temps mis par un participant pour chaque tour de course, de clef primaire (pa_id, to_num)
 - pa_id identifiant (entier arbitraire) de la participation
 - to_num numéro du tour (de 1 à gp_nbt)
 - to_temps le temps (en secondes) pour le tour correspondant

Q1. Écrire une requête SQL qui liste, par ordre chronologique, la date et le nom du circuit de toutes les courses qui se sont déroulées en France (ci_pays = 'France').

Q2. Écrire une requête SQL qui liste, pour chaque course de l'année 2021, le nom du circuit, le nom du pilote gagnant et son temps de course.

Q3. Écrire une requête SQL qui affiche, dans le sens décroissant, le nom des dix pilotes ayant le plus de victoire et le nombre de ces victoires.

Q4. Expliquer le résultat de la requête suivante

```

1 1 SELECT ci_nom, pi_nom, gp_date, to_temps
2 2 FROM (SELECT ci_id, ci_nom, MIN(to_temps) AS mtt
3 3   FROM Circuit
4 4   JOIN GrandPrix ON GrandPrix.ci_id = Circuit.ci_id
5 5   JOIN Participation ON Participation.gp_id = GrandPrix.gp_id
6 6   JOIN Tour ON Tour.pa_id = Participation.pa_id
7 7 GROUP BY
8 8   ci_id, ci_nom
9 9 ) AS rdc
10 10 JOIN GrandPrix ON GrandPrix.ci_id = rdc
11 11 JOIN Participation ON Participation.gp_id = GrandPrix.gp_id
12 12 JOIN Pilote ON Pilote.pi_id = Participation.pi_id

```

```
13 JOIN Tour ON Tour.pa_id = Participation.pa_id AND to_tmps = mtt
14 ORDER BY
15 ci_nom
```

5 Reconnaissance de caractères

La typographie est l’art d’assembler des caractères afin de composer des pages en vue de leur impression ou de leur affichage sur un écran, en respectant des règles visuelles qui rendent un texte agréable à lire.

Elle requiert des efforts importants, avantageusement simplifiés par le recours à l’outil informatique.

Voici la définition de quelques termes utiles pour la suite :

- un **caractère** est un signe graphique d’un système d’écriture, par exemple le caractère latin *a majuscule* « A ». Le standard Unicode donne à chaque caractère un nom et un identifiant numérique, appelé *point de code*, que nous appellerons ci-après simplement *code*. Le code de « A » dans la représentation Unicode est 65.
- un **glyphe** est un dessin particulier représentant un caractère, par exemple pour le caractère latin a majuscule : A (roman) A (italique) A (caligraphié) A (gras), A (courrier). . .
- une **police de caractères** est un ensemble coordonné de glyphes incluant différentes variantes (style roman ou italique, grasse. . .) et permettant de représenter un texte complet dans un système d’écriture.
- une **famille** est un groupe de polices.

Une base de données stocke les informations liées aux polices de caractères dans 4 tables ou relations.

Famille décrit les familles de polices, avec `fid` la clé primaire entière et `fnom` leur nom.

Police décrit les polices de caractères disponibles, avec `pid` la clé primaire entière, `pnom` le nom de la police et `fid` de numéro de sa famille.

Caractere décrit les caractères, avec `code` la clé primaire entière, `car` le caractère lui-même, `cnom` le nom du caractère.

Glyphe décrit les glyphes disponibles, avec `gid` la clé primaire entière, `code` le code du caractère correspondant au glyphe, `pid` le numéro de la police à laquelle le glyphe appartient, `groman` un booléen vrai pour du roman et faux pour de l’italique et `gdesc` la description vectorielle du glyphe.

Voici un extrait du contenu de ces tables.

Famille		Police			Caractere		
fid	fnom	pid	pnom	fid	code	car	cnom
1	Humane	1	Centaur	1	65	A	lettre majuscule latine a
2	Garalde	2	Garamond	2	66	B	lettre majuscule latine b
3	Réale	3	Times New Roman	3
4	Didone	4	Computer Modern	4	97	a	lettre minuscule latine a
5	Mécane	98	b	lettre minuscule latine b
6	Linéale	21	Triangle	6	99	c	lettre minuscule latine c
...

Glyphe					
gid	code	pid	groman	gdesc	
1	65	20	True	[[[0, 0], [1, 2], [2, 0]], [[0.5, 1], [1.5, 1]]]	
2	65	20	False	[[[0, 0], [2, 2], [2, 0]], [[1, 1], [2, 1]]]	
...	
501	97	21	True	[[[0, 0], [0.5, 1], [1, 0], [0, 0]]]	
502	98	21	True	[[[0, 2], [0, 0], [1, 0.5], [0, 1]]]	
503	99	21	True	[[[1, 1], [0, 0.5], [1, 0]]]	
504	100	21	True	[[[1, 2], [1, 0], [0, 0.5], [1, 1]]]	
...	

Q1 Proposer une requête en SQL sur cette base de données pour compter le nombre de glyphes en roman (cf. description

précédente).

Q2 Proposer une requête en SQL afin d'extraire la description vectorielle du caractère A dans la police nommée Helvetica en italique.

Q3 Proposer une requête en SQL pour extraire les noms des familles, classés par ordre alphabétique, ainsi que leur nombre de polices en limitant l'affichage aux familles ayant au moins deux polices.