



I - Principe de la Base de Données

I.1 Vocabulaire sur les données structurées

Chaque **ligne** est l'un des **enregistrements / objets / n-uplets**.

Si la première ligne contient les noms des attributs ont la nomme **en-tête**.

Chaque **colonne** comporte les valeurs associées à un même **descripteur / attribut**.

L'ensemble des informations ayant la même structure d'attributs se nomme une **collection** ou une **table**.

La case à l'intersection d'une ligne et d'une colonne se nomme un **champ**.

Le contenu d'une case se nomme sa **valeur**.

I.2 Base de données

Une **base de données** est un **ensemble structuré stockant des données intègres**.

On peut **insérer, trier, projeter** des colonnes, **filtrer** des lignes, ou **faire des jointures** entre les tables (les *assembler* entre elles) en respectant les **propriétés suivantes** :

- | | |
|--|--|
| <ol style="list-style-type: none"> 1. Intégrité des données 2. Persistance des données 3. Sécurisation des accès 4. Efficacité de traitement | <ol style="list-style-type: none"> 5. Accès concurrents 6. Indépendance entre logique et physique. 7. Interface directe ou service API |
|--|--|

II - Historique

III - Relation (voir DOC 1)

3.1 - Contrainte d'intégrité de DOMAINE

Chaque attribut est défini par son nom ET par son domaine de valeurs autorisées.

La valeur **NULL** signale l'absence d'info ou qu'elle n'a aucun sens pour cet enregistrement.

On peut imposer des **CONTRAINTES d'intégrité de DOMAINE** qui empêchent d'enregistrer une donnée si elle n'a pas le type attendu : tout ce qui rentre dans le système est analysé AVANT insertion.

3.2 - Domaines disponibles en SQLITE

1. TEXT : string et caractère unique
2. NUMERIC pour un nombre.
3. INTEGER pour les entiers.
4. REAL pour les flottants.
5. BLOB pour le reste

Dans les versions plus évoluées :

CHAR(40) : exactement 40 caractères
 VARCHAR(40) : jusqu'à 40 caractères
 DATE : une date au format jour, mois, année
 DATETIME : pareil mais à l'horaire hh:mm:ss

3.3 - Schéma relationnel d'une table

Une **relation** est définie avec **son nom** et la **liste de ses attributs** (nom et domaine)

salarie(id:int, nom:text, age:int, adresse:text, salaire:int)

L'attribut souligné est la clé primaire (voir ci-dessous).

3.4 Contrainte d'UNICITE (liée à la valeur de la clé primaire/primary key pk)

La **relation** peut être vue comme un **ensemble de n-uplets** : chaque n-uplet doit donc être différent des autres.

La **clé primaire** est un attribut permettant d'identifier et différencier un n-uplet parmi tous les n-uplets de la relation. Il s'agit le plus souvent d'un attribut de type INTEGER et nommé id. La clé primaire peut-être

1. un attribut (comme un numero_etudiant) ou
2. l'association de plusieurs attributs (comme (numero_de_secu, nom, prenom)).

La **contrainte d'UNICITE** impose qu'on puisse toujours distinguer deux n-uplets d'une relation, au moins par la clé primaire. **Une clé primaire NE PEUT PAS VALOIR NULL et doit être différente de toutes les autres.**

Un SGBD refuse d'insérer un n-uplet si sa clé primaire est déjà celle d'une autre n-uplet.

Dans le cadre d'un , on peut charger le SGBD de choisir automatiquement le nouvel id d'un n-uplet qu'on veut insérer. On peut alors lui fournir NULL comme valeur d'id, et il choisira lui-même une valeur qui n'existe pas encore.

On identifie les clés primaires **en les soulignant** dans un schéma relationnel abstrait, en plaçant une clé dorée sur l'interface graphique phpMyAdmin, parfois en plaçant le nom de la relation en suffixe (ex **id_livre**), parfois en plaçant pk en préfixe (ex **pk_id_livre**)

IV - Clés étrangères (Voir Doc 2)

Clés primaires et clés étrangères permettent de réaliser des jointures / jonctions entre plusieurs tables.

Les **clés étrangères** respectent la **contrainte de REFERENCE** : ils ont une valeur qui doit correspondre à NULL ou à une valeur de clé primaire réelle dans une autre table.

Suivre une clé étrangère permet de ramener de nouveaux attributs qu'on va rajouter à la sélection actuelle.

V - Schéma relationnel d'une base de données (voir Doc 3)

Le **schéma relationnel d'une base de données** est l'ensemble :

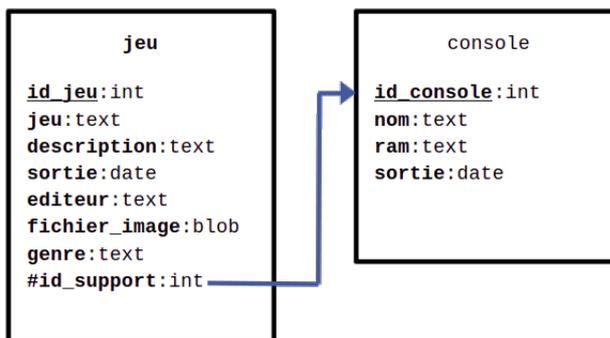
- des schémas relationnels de chaque table
- des liaisons clés primaires-étrangères entre ces relations.

On **souligne** les clés primaires.

On **place #** devant les noms des clés étrangères.

On indique par un moyen quelconque vers quelle clé primaire pointe la clé étrangère.

DOCUMENT 3



A CONNAITRE PAR COEUR pour le BAC :

Nous avons vu aujourd'hui **3 CONTRAINTES D'INTEGRITE :**

→ contrainte de DOMAINE : les valeurs des attributs doivent appartenir au domaine imposé.

→ contrainte d'UNICITE : clés primaires toutes différentes, NULL interdit.

→ contrainte de REFERENCE : clés étrangères référencent une valeur de clé primaire, ou NULL.

DOCUMENT 1

Q01 :

Q03 :

En-tête

id	Jeu	Description	Année de sortie	Editeur	Support	Genre	Capture d'écran
1	Flight Simulator	Simulation de pilotage d'avion avec des pixels gros comme des camions !	1980	subLOGIC	Apple 2	Simulateur	
2	DONKEY KONG	Un méchant singe géant a capturé la fiancée de JumpMan (qui portera plus tard le nom de Mario). Guide JumpMan pour qu'il délivre sa dulcinée.	1981	Nintendo	Arcade	Plate-formes	
3	PITFALL!	Pitfall Harry, un explorateur, ramasse des objets éparpillés dans la jungle.	1982	Activision	Atari 2600	Plate-formes	
4	PITFALL!	Pitfall Harry, un explorateur, ramasse des objets éparpillés dans la jungle.	1983	Activision	Commodore 64 (C64)	Plate-formes	
5	Boulder Dash	Rockford, mineur téméraire, ramasse des diamants et tente de ne pas se faire écraser par les énormes pierres instables.	1983	First Star Software	Commodore 64 (C64)	Plate-formes	

Q04 :
Q05 :

Q02 :

DOCUMENT 2

relation jeu

Clé Primaire								Clé étrangère
id	jeu	description	sortie	editeur	fichier_image	genre	id_support	
1	Flight Simulator	Simulation de pilotage d'avion avec des pixels gros...	1980	subLOGIC		Simulateur	1	
7	DONKEY KONG	Un méchant singe géant a capturé la fiancée de Jump...	1981	Nintendo		Plate-formes	0	
8	PITFALL!	Pitfall Harry, un explorateur, ramasse des objets ...	1982	Activision		Plate-formes	2	
9	PITFALL!	Pitfall Harry, un explorateur, ramasse des objets ...	1983	Activision		Plate-formes	3	
10	Boulder Dash	Rockford, mineur téméraire, ramasse des diamants e...	1983	First Star Software		Plate-formes	3	

relation console

Clé Primaire			
id	nom	RAM	date_sortie
1	Apple II	4 Ko	1977-06-10
2	Atari 2600	4 Ko	1977-10-14
3	C64	64 Ko	1982-08-01