

Dictionnaire

Middleware (M)	Source Excel (E)	Source BD (B)	Critères
Album (Al) Artiste (Ar) Genre (Ge) Subgenre (Sg)		AlbumListe (L) Artiste (A) Genre (G)	
Al.year : type = INT	E.year : type = INT	L.année : type = INT	valeur(E.year)==valeur(B.L.année) == valeur(M.Al.year)
Al.number : type = INT	E.number : type = INT	L.id-album : type = INT	valeur(E.number)==valeur(B.L.id-album)==valeur(M.al.number)
Al.title : type = String	E.album : type = String	L.titre : type = INT	valeur(E.album)==valeur(B.L.titre)==valeur(M.Al.title)
Ar.id-artist : type = String Al.id-artist : type = String		L.id-artiste : type = INT A.id-artiste : type = INT	valeur(M.Ar.id-artist)==valeur(B.A.id-artist)==valeur(M.Al.id-artist)==valeur(B.L.id-artiste) id-artiste n'existe pas dans E Clé étrangère
Ar.name : type = String	E.artiste : type = String	A.nom : type = String	valeur(E.artiste)==valeur(B.A.nom)==valeur(M.Ar.name)
Al.id-genre : type = INT Ge.id-genre : type = INT		L.id-genre : type = INT G.id-genre : type = INT	valeur(M.Al.id-genre)==valeur(B.L.id-genre)==valeur(M.Ge.id-genre)==valeur(B.G.id-genre) id-genre n'existe pas dans E Clé étrangère
Ge.designation : type = String	E.genre : type = String	G.libelle : type = String	valeur(E.genre)==valeur(B.G.libelle)==valeur(M.Ge.designation)
Al.id-subgenre : type = INT Sg.id-subgenre : type = INT			id-subgenre n'existe ni dans E, ni dans B
Sg.designation : type = String	E.subgenre : type = String		valeur(E.subgenre)==valeur(M.Sg.designation) Peut être null

Al.source			source n'existe ni dans E, ni dans B
-----------	--	--	--------------------------------------

Requêtes SQL liées aux demandes :

1. Afficher pour chaque année, le nombre total d'albums pour le genre rock, puis pour les genres autres que Rock pour les années avant et après 1970

MySQL (antérieur à 1970) :

- Partie Rock : `SELECT année, COUNT(`id-album`) as nb FROM albumliste a, genre g WHERE a.`id-genre`=g.`id-genre` AND g.libelle='Rock' GROUP BY année ORDER BY année`
- Partie Non-Rock : `SELECT année, COUNT(`id-album`) as nb FROM albumliste a, genre g WHERE a.`id-genre`=g.`id-genre` AND g.libelle<>'Rock' GROUP BY année ORDER BY année`

2. Afficher par année, la liste des genres et sous genres d'albums sortis.

MySQL (antérieur à 1970) :

- `SELECT année, libelle FROM albumliste a, genre g WHERE a.`id-genre`=g.`id-genre` ORDER BY année, libelle`

3. Afficher le nb de genre d'albums sortis par année par ordre décroissant

MySQL (antérieur à 1970) :

- `SELECT année, COUNT(DISTINCT(`id-genre`)) as nb FROM albumliste a GROUP BY année ORDER BY nb desc`