
Bases de données

Version 1.0

Alain Taquet

Septembre 2017

Table des matières

1	TD 1	1
1.1	La base Sport	1
1.1.1	Relations	1
1.1.2	Tri	2
1.1.3	Recherche dans une table	2
1.1.4	Filtres	2
1.2	Introduction aux requêtes	2
1.2.1	Requête Monotable	2
1.2.2	Requête avec jointure	3
1.3	Requêtes simples	4
2	TD 2	5
2.1	Requêtes simples.	5
2.2	Champs calculés.	6
2.3	Un premier regroupement.	7
3	TD 3	8
3.1	Regroupements	8
4	TD 4	10
4.1	Sous-requêtes	10
5	TD 5	13
5.1	Autojointures	13
6	Annales Access : Gowalla	16
6.1	La base Gowalla	16
6.2	Questions	19
7	Annales Access : Agence	21
7.1	La base Agence	21
7.2	Questions	22
8	TD Select	24
8.1	La clause SELECT	24

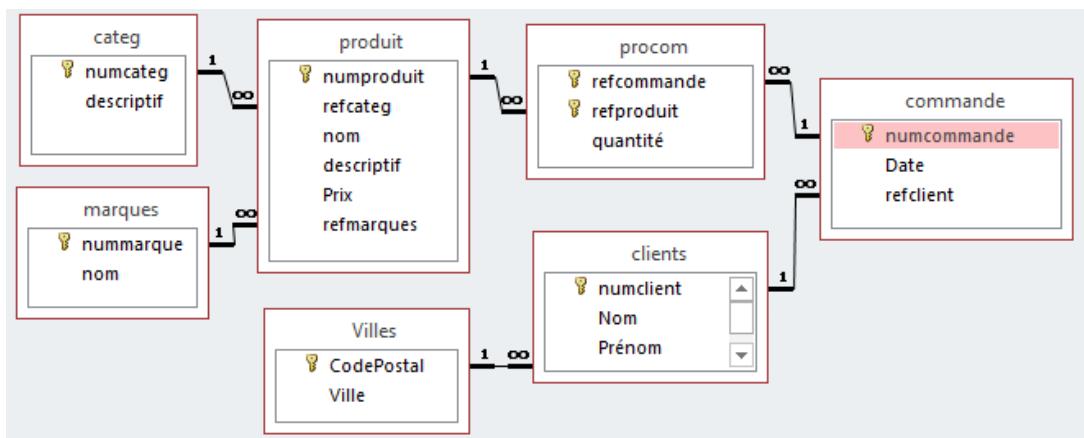
8.1.1	Projection	25
8.1.2	Lignes distinctes	25
8.1.3	Champs calculés - alias	25
8.2	La clause WHERE	26
8.2.1	Filtrage	26
8.2.2	Caractères génériques	26
8.3	La clause ORDER BY	27
8.4	Les fonctions	28
8.4.1	Opérations sur les chaînes de caractères	28
8.4.2	Opérateur conditionnel	29
8.4.3	Fonctions statistiques	30
9	TD Jointures	31
9.1	Jointure croisée	31
9.2	Jointure interne	31
9.2.1	Avec condition de selection	32
9.2.2	Auto-jointures	33
9.2.3	Sous requêtes	34
9.2.4	Jointures généralisées	35
9.3	Jointure externe	36
10	TD Regroupements	37
10.1	La clause GROUP BY	37
10.2	Sémantique du GROUP BY	37
10.3	GROUP BY HAVING	38
10.4	Groupement avec jointure externe	41
11	TD Révisions	42
11.1	La base	42
11.2	Questions	42
11.3	Questions Bis	44
12	Cinema1	47
12.1	La base	47
12.2	Questions	47
13	Cinema2	50
13.1	Questions	50
14	Cinema3	54
14.1	Questions	54
15	Correction TD 1	58
16	Correction TD 2	60
17	Correction TD 3	64
18	Correction TD 4	68

19 Correction TD 5	75
19.1 Jointures gauches	78
20 Correction Annales Access : Gowalla	83
21 Correction Annales Access : Agence	87
22 Correction TD Select	92
23 Correction TD Jointures	95
24 Correction TD Regroupements	100
25 Correction TD Revisions	106
26 Correction Cinema1	111
27 Correction Cinema2	114
28 Correction Cinema3	118

1.1 La base Sport

Un magasin de sport souhaite informatiser les commandes effectuées par ses clients. Les produits ont un code, un nom, un descriptif et un prix et sont classés selon des marques et des catégories. Les clients ont un nom, un prénom et habitent une ville référencée à partir du code postal. Les clients effectuent des commandes d'un ou plusieurs produits en quantité variable, plusieurs commandes peuvent être faites par le même client à une même date.

1.1.1 Relations



- Passez en mode création de la table clients : Sélectionnez la table clients -> modifier ou sélectionnez la table dans la fenêtre relation puis activez le menu contextuel
- Observez les différents champs en cliquant dans la colonne “nom du champ”, vérifiez que la clé primaire est noté en gras dans la fenêtre “relations”
- Faites de même pour les autres tables. Quelle est la clé de “procom” ?
- Sélectionnez la relation entre clients et villes (double clic ou menu contextuel) Quelle est la relation ?

1.1.2 Tri

- Affichez la liste des clients dans l'ordre alphabétique des noms
- Triez la table dans l'ordre des références des clients

1.1.3 Recherche dans une table

- Recherchez le client Paldir

1.1.4 Filtres

- Affichez la liste des clients habitant le Nord
- Affichez tous les enregistrements (menu enregistrement ou entonnoir)
- Recherchez le nombre de clients dont le nom commence par “Ta”
- Recherchez le nombre de clients dont le nom se termine par “ie”
- Recherchez le nombre de clients dont le nom contient un “y”
- Recherchez le nombre de clients dont le nom contient 5 lettres
- Recherchez le nombre de clients dont le nom commence par “Ta” et qui habitent le Nord
- Recherchez le nombre de clients dont le nom commence par “Ta” et qui habitent le Nord ou le Pas de Calais
- Recherchez le nombre de commandes effectuées en janvier 97
- Quels sont les produits qui ont été commandés au cours de la commande numéro 1 ?
- Par qui a été faite cette commande ?
- Combien ce client a-t-il fait de commandes ?
- Où habite ce client ?

1.2 Introduction aux requêtes

1.2.1 Requête Monotable

Affichez le prénom du client Paldir

- Choisir dans la rubrique REQUETES, le bouton nouveau, puis Requête vierge (les champs affichés sont repérés par une croix dans un carré)
- Ajouter la table clients
- Faire glisser le champ Nom jusqu'à la première colonne de la ligne Champ (ou double clic)
- Vérifier que le champ ne sera pas affiché
- Faire de même avec le champ Prénom et vérifier qu'il sera affiché
- Dans la zone Critères du champ nom ajouter : Paldir
- Enregistrer la requête sous le nom Requête1
- Exécuter la requête en cliquant sur l'icône Executer (point d'exclamation)
- Revenir en mode création s'il faut apporter des modifications (bouton équerre)

Formalisme

#	requete Paldir
tables ouvertes	clients
jointures	
champs affichés	clients.Prénom
critères	clients.Nom="Paldir"
regroupements	
tris	

SQL

```
SELECT clients.Prenom
FROM clients
WHERE clients.Nom="Paldir"
```

1.2.2 Requête avec jointure

Affichez le client Paldir avec sa ville d'habitation

- Choisir dans la rubrique REQUETES, le bouton nouveau, puis Requête vierge (les champs à afficher sont repérés par une croix dans un carré)
- Ajouter les tables clients et villes
- Faire glisser le champ Nom jusqu'à la première colonne de la ligne Champ
- Vérifier que le champ sera affiché
- Faire de même avec le champ Prénom puis avec le champ Ville
- Dans la zone Critères du champ nom ajouter : Paldir
- Enregistrer la requête sous le nom Requête2
- Exécuter la requête en cliquant sur l'icône Executer (point d'exclamation)
- Revenir en mode création s'il faut apporter des modifications (bouton équerre).

Formalisme

#	requete Paldir
tables ouvertes	clients, villes
jointures	villes.CodePostal = clients.Code_Postal
champs affichés	clients.Nom, villes.Ville
critères	clients.Nom = "paldir"
regroupements	
tris	

SQL

```
SELECT clients.Nom, Villes.Ville
FROM Villes JOIN clients ON villes.CodePostal = clients.Code_Postal
WHERE clients.Nom = "paldir"
```

1.3 Requêtes simples

r3 : Affichez les noms et prénoms des clients dont le nom commence par “Ta”.

#	Nom	Prénom
1	Talonion	Thomas
2	Tabille	Raoul
3	Tabaniol	Alphonse
.	.	.
13	Taijouhai Parmillier	Betty Barbara Noël Kathy Tess Sandra Lucille Ella Vick

r4 : Recherchez les clients dont le nom commence par “Ta” et qui habitent le Nord

#	Nom	Prénom
1	Taitrajeune	Thérèse
2	Tamair	Annick
3	Tamune Tablette D'Heuchokola	Jean
4	Tafraise	Théramène

r5 : Recherchez les clients dont le nom commence par “Ta” et qui habitent le Nord ou le Pas de Calais

#	Nom	Prénom
1	Taitrajeune	Thérèse
2	Tart'inn	Kimberley
3	Tamair	Annick
.	.	.
6	Taijouhai Parmillier	Betty Barbara Noël Kathy Tess Sandra Lucille Ella Vick

r6 : Recherchez les clients dont le nom commence par “Ta” et qui n’habitent ni le Nord ni le Pas de Calais

#	Nom	Prénom
1	Alavie	Irenée
2	Aulet	Thierry
3	Talonion	Thomas
.	.	.
586	Céjoli	Edouard Nehneh

TD 2

2.1 Requêtes simples.

r6a : Afficher les noms des clients dont le nom commence par Ta et qui ont effectué une commande (On n'affichera que le nom du client) Vous devez trouver 5 réponses. Quel est le rôle de la jointure ? Comparez avec la requête 3.

#	Nom
1	Talonion
2	Tabille
3	Tabille
.	.
5	Taijouhai Parmillier

r6b : Même requête que la 6a, allez dans les propriétés de la requête et dans le champ « valeurs distinctes » affichez « oui » Combien trouvez vous de réponses ? Comparez avec la requête 6a. Comparez le code sql des requêtes 6a et 6b.

#	Nom
1	Tabille
2	Taijouhai Parmillier
3	Talonion

r7 : Afficher les noms des clients qui ont effectué une commande en janvier 97

#	Nom
1	Adlaplace
2	Aicteur
3	Alavie
.	.
21	Use

r8 : Afficher le nom du client qui a effectué la commande 1 ainsi que les produits commandés

#	clients.nom	produit.nom
1	Alavie	Diamond lady
2	Alavie	Aventure 65
3	Alavie	Super guide tradition
4	Alavie	Charlotte

r9 : Afficher les numéros de commande et les produits commandés par un client dont le code sera saisi au clavier

#	refclient	nom
1	1	Diamond lady
2	1	Aventure 65
3	1	Super guide tradition
.	.	.
8	1	Alpinist

r10 : Afficher les produits commandés par un client dont le code et le code du numéro de commande seront saisis au clavier

#	nom
1	Diamond lady
2	Aventure 65
3	Super guide tradition
4	Charlotte

2.2 Champs calculés.

r11 : Commencez par afficher la quantité et le prix des produits de la commande 1. Positionnez le curseur dans “Champ” d’une nouvelle colonne puis cliquez sur Générer. Générez l’expression procom.quantité * produit.Prix. Modifiez l’entête de colonne : Total au lieu de expr1. Ecrivez le formalisme de la requête.

#	quantité	Prix	Total
1	3	550,00 €	1 650,00 €
2	1	949,00 €	949,00 €
3	1	1 312,00 €	1 312,00 €
4	1	144,00 €	144,00 €

r12 : La TVA est de 19,6% et les prix des produits sont donnés tva incluse. Pour chaque produit commandé dans la commande 1 affichez le montant de la TVA, le montant hors taxe et le montant TTC. Commencez par calculer le montant hors taxe et le montant TTC puis calculez le montant de la TVA.

#	quantité	Prix	TTC	HT	TVA
1	3	550,00 €	1 650,00 €	1 375,00 €	275,00 €
2	1	949,00 €	949,00 €	790,83 €	158,17 €
3	1	1 312,00 €	1 312,00 €	1 093,33 €	218,67 €
4	1	144,00 €	144,00 €	120,00 €	24,00 €

r13 : Une réduction de 10% est accordée pour chaque produit commandé en quantité supérieure ou égale à 3. Pour chaque produit commandé dans la commande 1 affichez le nom du produit, le total sans réduction, le montant de la réduction et le montant avec réduction (On ne tient plus compte de la question 12).

#	nom	ssReduc	Reduc	AvecReduc
1	Diamond lady	1 650,00 €	165,00 €	1 485,00 €
2	Aventure 65	949,00 €	0,00 €	949,00 €
3	Super guide tradition	1 312,00 €	0,00 €	1 312,00 €
4	Charlotte	144,00 €	0,00 €	144,00 €

r14 : La direction décide d'accorder une remise dépendant de la catégorie des produits. Il y a 6 catégories, si le code de la catégorie est inférieur ou égal à 3, la remise est de 10%, si le code est égal à 4 ou 5 la remise est de 15%, s'il s'agit de sac à dos (catégorie 6) la remise est de 20 %. Pour la commande 1 affichez le nom du produit, le code de la catégorie et le total compte tenu de la remise.

#	nom	refcateg	Total
1	Diamond lady	1	1 485,00 €
2	Aventure 65	6	759,20 €
3	Super guide tradition	3	1 180,80 €
4	Charlotte	4	122,40 €

2.3 Un premier regroupement.

r15 : Affichez le nombre de produits par marque par ordre alphabétique.

#	nom	nbProduits
1	Boreal	1
2	cebe	2
3	cimalp	1
.	.	.
23	vertical	1

TD 3

3.1 Regroupements

r18 : Affichez le nombre de produits dans la base.

#	nbProduits
1	31

r19 : Affichez le maximum, le minimum et la moyenne des prix des produits.

#	maxPrix	minPrix	moyPrix
1	1 450,00 €	49,00 €	461,10 €

r20 : Affichez le nombre d'articles vendus.

#	nbArticles
1	353

r21 : Affichez le nombre de produits différents commandés par le client Alavie par commande.

#	numcommande	nbProduits
1	1	4
2	14	4

r22 : Affichez le nombre d'articles commandés par le client Alavie par commande.

#	numcommande	nbArticles
1	1	6
2	14	6

r23 : Affichez le montant et la date des commandes effectuées par le client Alavie.

#	Montant	Date
1	4 055,00 €	22/01/1997
2	4 736,00 €	02/02/1997

r24 : Affichez le montant et la date des commandes effectuées par le client Alavie avec une réduction de 20% sur l'article de ref 15

#	Date	Montant
1	22/01/1997	3865,2
2	02/02/1997	4546,2

r25 : Affichez le montant et la date des commandes effectuées par un client dont le code sera saisi au clavier

#	Monatnt	Date
1	4 055,00 €	22/01/1997
2	4 736,00 €	02/02/1997

r26 : Affichez le nom et prénom des clients dont le total des commandes est de plus de 5000 €

#	Nom
1	Alavie
2	Daicacahuett'Livair
3	Juskalapporte
.	.
6	Yala Couvertur Kigratt

r27 : Affichez le nom et prénom des clients qui ont effectué une commande de plus de 5000 €

#	Nom
1	Daicacahuett'Livair
2	Juskalapporte
3	Yala Couvertur Kigratt

r28 : Affichez par ordre alphabétique inverse les produits qui ont été vendus en quantité ≥ 15

#	nom
1	Tube-stick indice 8
2	Super Khumbu
3	K2 Rosso
.	.
6	Alpinist

r29 : Affichez par ordre alphabétique les produits qui ont été vendus en quantité ≥ 15 et dont le prix unitaire est > 900 €

#	nom
1	Aventure 65
2	K2 Rosso

4.1 Sous-requêtes

r31 : Affichez le nombre de produits par marque

#	nom	nbProduits
1	Boreal	1
2	cebe	2
3	cimalp	1
.	.	.
23	vertical	1

r32 : Affichez le nombre de produits par catégorie

#	descriptif	nbProduits
1	Chaussons d'escalade	6
2	Chaussures d'alpinisme	6
3	Lunettes de glacier	4
.	.	.
6	Vêtements de grimpe	4

r33 : Affichez le nombre d'articles vendus par catégorie

#	descriptif	nbVendus
1	Chaussons d'escalade	61
2	Chaussures d'alpinisme	75
3	Lunettes de glacier	41
.	.	.
6	Vêtements de grimpe	45

r34 : Affichez les noms des articles de prix supérieur à la moyenne des prix

#	nom
1	Diamond lady
2	Manga
3	Miura
.	.
13	Nepal top

r35 : Affichez le nom et le prénom des clients qui ont acheté des produits de plus de 3 marques différentes

#	Nom
1	Al Poubayl
2	Alavie
3	Baul
.	.
11	Touténe

r36 : Affichez le montant de la première commande effectuée par un client dont le code sera saisi au clavier. S'il y a plusieurs commandes le même jour on les affichera toutes.

#	Montant
1	4 055,00 €

r37 : Affichez le nom et prénom des clients qui n'ont pas acheté de produits de moins de 1000 euros. Les clients n'ayant rien commandé ne seront pas inclus.

#	Nom	Prénom
1	Citation	Félix
2	Deuf	John
3	Diotte	Kelly
.	.	.
8	Ticoli	Thor

r38 : Affichez dans l'ordre alphabétique le nom et prénom des clients qui ont acheté au moins un des produits acheté par Alavie

#	Nom
1	Aboucepartou
2	Adlaplace
3	Al Poubayl
.	.
42	Yala Couvertur Kigratt

r39 : Affichez les nom et prénom des clients qui habitent la même ville que Peuleurido en utilisant une sous-requête

#	Nom
1	Deulisse
2	Attan
3	Sanfraper
.	.
13	Yoraitoux

r39Bis : Même question sans utiliser de sous-requêtes

#	Nom
1	Deulisse
2	Attan
3	Sanfraper
.	.
13	Yoraitoux

r40 : Affichez les nom et prénom des clients qui ont commandé le même jour qu'Alavie en utilisant une sous-requête

#	Nom
1	Aulet
2	Bienlepeti
3	Fran
.	.
6	Use

r40Bis : Même question sans utiliser de sous-requêtes

#	Nom
1	Aulet
2	Bienlepeti
3	Fran
.	.
6	Use

r41 : Affichez les nom et prénom des clients qui ont commandé le même MOIS qu'Alavie en utilisant une sous-requête

#	Nom
1	Adlaplace
2	Aicteur
3	Am'Servir
.	.
31	Vaigette

r41Bis : Même question sans utiliser de sous-requêtes

#	Nom
1	Adlaplace
2	Aicteur
3	Am'Servir
.	.
31	Vaigette

TD 5

5.1 Autojointures

r39 : Affichez les nom et prénom des clients qui habitent la même ville que Peuleurido en utilisant une sous-requête

#	Nom
1	Deulisse
2	Attan
3	Sanfraper
.	.
13	Yoraitoux

r39Bis : Même question sans utiliser de sous-requêtes

#	Nom
1	Deulisse
2	Attan
3	Sanfraper
.	.
13	Yoraitoux

r40 : Affichez les nom et prénom des clients qui ont commandé le même jour qu'Alavie en utilisant une sous-requête

#	Nom
1	Aulet
2	Bienlepeti
3	Fran
.	.
6	Use

r40Bis : Même question sans utiliser de sous-requêtes

#	Nom
1	Aulet
2	Bienlepeti
3	Fran
.	.
6	Use

r41 : Affichez les nom et prénom des clients qui ont commandé le même MOIS qu'Alavie en utilisant une sous-requête

#	Nom
1	Adlaplace
2	Aicteur
3	Am'Servir
.	.
31	Vaigette

r41Bis : Même question sans utiliser de sous-requêtes

#	Nom
1	Adlaplace
2	Aicteur
3	Am'Servir
.	.
31	Vaigette

r43 : Affichez le nombre de produits par marques dans l'ordre croissant des marques

#	nom	nbProduits
1	aigle	0
2	Boreal	1
3	cebe	2
.	.	.
26	vertical	1

r44 : 586 clients sont enregistrés dans la base de données, 109 ont effectés une commande. Affichez par ordre alphabétique les noms des clients qui n'ont rien commandé.

#	Nom	Prénom
1	Adessin	Blanche
2	Adlvy Yadlespoir	Tanguy
3	Adonaideux	Hermann
.	.	.
477	Zieubleux	Bruno

r45 : Retour sur la requête 37 : “Affichez le nom et prénom des clients qui n'ont pas acheté de produits de moins de 1000 euros.” Recommencez en incluant les noms et prénoms des clients n'ayant rien commandé et en les affichant par ordre alphabétique.

#	Nom
1	Duman
2	Comaindieu
3	Adessin
.	.
485	Céjoli

r47 : Affichez les noms des clients qui ont effectués en août une commande de montant supérieur à celle qui a été la plus grande en janvier

#	Nom
1	Juskalapporte

r48 : Nom du client qui a acheté le produit le plus cher en janvier

#	Prénom	Nom
1	Félix	Citation

r49 : Quels sont les clients qui ont acheté le produit le plus cher acheté par Pige ? (3 réponses)

#	Nom	Prénom
1	Adrouille Toultan	Yves
2	Dumonde	Séraphin
3	Yala Couvertur Kigratt	Sandra

r50 : Nombre de commandes par clients (s'ils n'ont pas fait de commande on doit indiquer 0)

#	Nom	Prénom	nbCommandes
1	Aboucepartou	Thècle	1
2	Adessin	Blanche	0
3	Adlaplace	Andreï	1
.	.	.	.
586	Zieubleux	Bruno	0

Annales Access : Gowalla

6.1 La base Gowalla

- La base Gowalla se trouve ici : gowalla.mdb .
- La correction Gowalla au format access se trouve ici : corr-gowalla1.mdb .

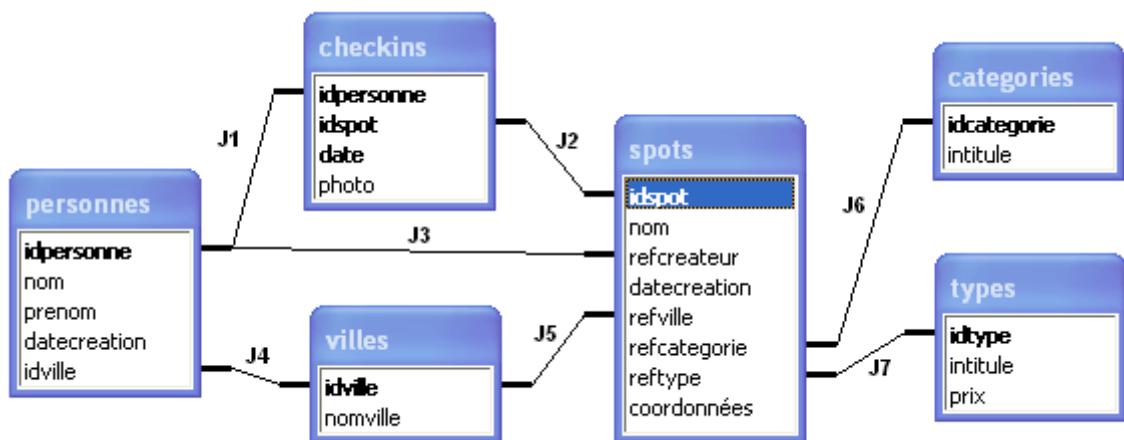


Fig. 6.1 – Relations

Gowalla, Foursquare, DisMoiOu sont des réseaux sociaux mobiles basés sur la géolocalisation (GPS) qui permettent de référencer des lieux et de les partager avec les autres membres. Dès que quelqu'un a créé un point d'intérêt (spot) les autres membres peuvent y laisser un "check-in" éventuellement associé d'un commentaire et d'une photo.

On supposera ici que les spots sont créés par une et une seule personne et qu'au moment de leur création la référence de la ville dans laquelle ils se situent est mise à jour. Ces spots sont répartis en catégories (Bar, Librairie, Hotel ...) et en types (simple, commercial, vip). Les spots simples sont gratuits et contribuent à la diffusion de l'application. Les spots de type commercial ou vip sont payants avec un abonnement par mois de 5€ pour le commercial et 30€ pour le vip. Un traitement particulier leur est associé lors de la présentation des spots à l'utilisateur.

Les check-in sont enregistrés dans la table "checkins", ils sont laissés par une personne et sur un spot. La date du check-in est enregistrée ainsi que le nom de la photo s'il y en a une.

		
Passport	Spots	Trips
	Le 30 3 meters	
	Café Coton 6 meters	
	Paul 12 meters	
	Orange 14 meters	
	Meert 16 meters	
	Le Dracir 17 meters	
	Coq Hardi 21 meters	
	Le Beffroi 24 meters	

Fig. 6.2 – spots

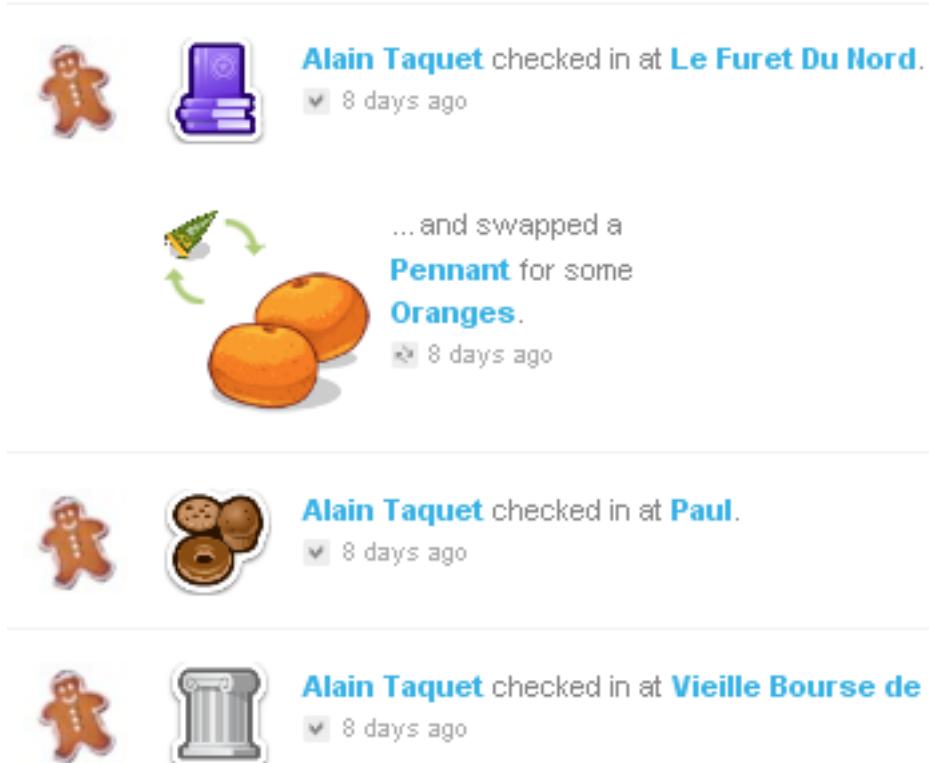


Fig. 6.3 – checkins

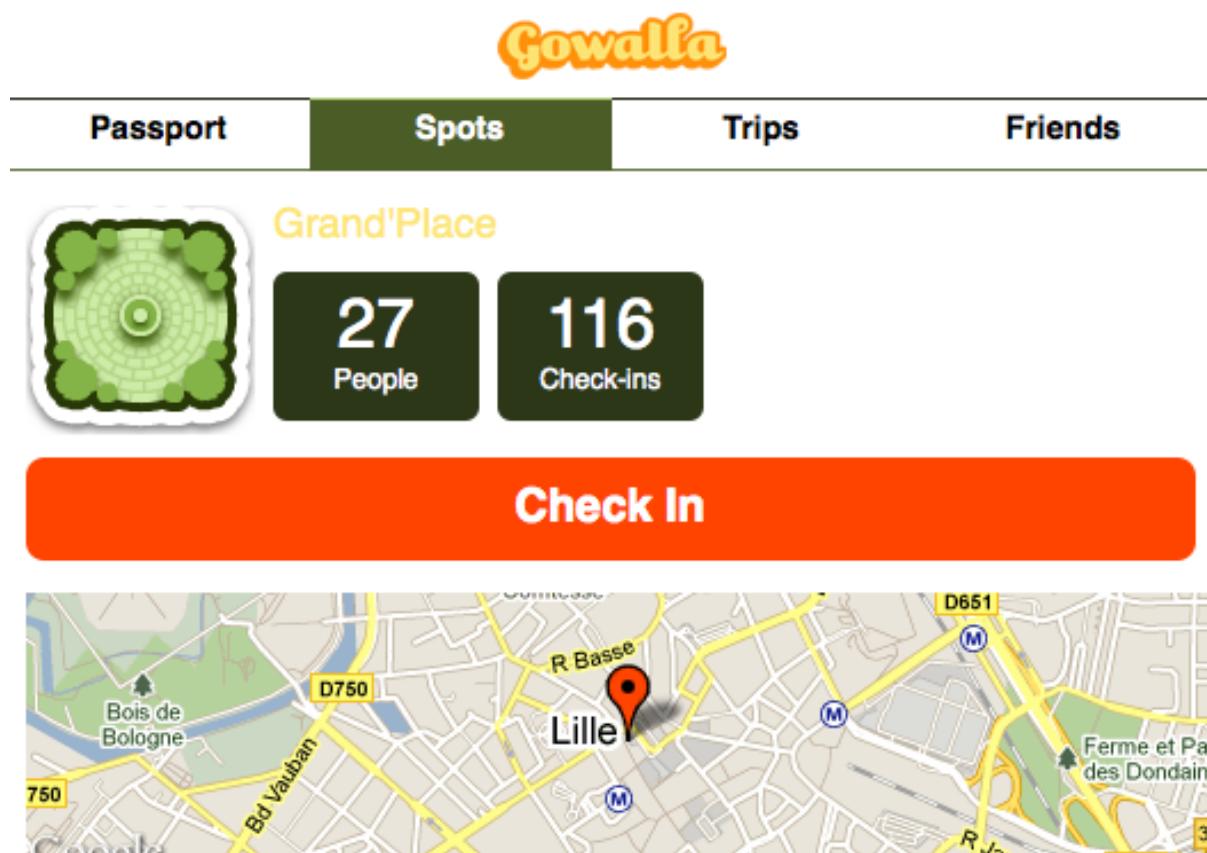


Fig. 6.4 – check

Pour les 4 premières questions vous renseignerez les jointures, à partir de la question 5 vous indiquerez J1, J2 etc pour les jointures “normales”. Pour chaque question vous indiquerez en plus du formalisme le nombre de résultats obtenus ainsi que les 3 premières lignes.

6.2 Questions

r1 : Afficher les noms des spots dans l'ordre alphabétique.

#	nom
1	Amorino
2	Balthazar
3	Berthillon
.	.
21	UGC Lille

r2 : Afficher les noms des personnes habitant Lille dans l'ordre alphabétique.

#	nom
1	BLEU
2	COUVERT
3	HOSSECOURT
4	TASSIEPA

r3 : Afficher le nom de la catégorie dont le code est saisi au clavier, on choisira 12.

#	intitule	idcategorie
1	Bar	1

r4 : Afficher les noms des spots de catégorie Librairie situés à Lille dans l'ordre alphabétique.

#	nom
1	Fnac

r5 : Combien y a-t-il de spots par type (intitulé). Dans cette question on n'affichera pas la valeur 0.

#	intitule	CompteDeidspot
1	commercial	8
2	simple	6
3	vip	7

r6 : Chaque check-in sur un spot commercial de catégorie Librairie est rémunéré 0,1€, quelle est la recette pour ce type de spot ? (check-in uniquement)

#	Expr1
1	1,1

r7 : Quelle est la moyenne en nombre de jours entre la date de création du spot et ses check-in ?

#	nom	Expr1
1	Amorino	78,75
2	Balthazar	146,666666666667
3	Berthillon	177,714285714286
.	.	.
20	UGC Lille	173,428571428571

r8 : Afficher, dans l'ordre alphabétique, les noms des personnes qui ont fait un check-in au même spots que ceux de Ness. (sans sous-requête)

#	nom
1	BLEU
2	CHANJEPA
3	CITATION
.	.
19	TASSIEPA

r9 : Quelle est la ville où il y a le plus grand nombre de spots ?

#	nomville
1	Paris

r10 : Le prix des abonnements des spots de type vip augmente de 10%, calculer le montant total des abonnements par ville.

#	nomville	Expr1
1	Lille	76
2	New York	76
3	Paris	86
4	Stockholm	33

r11 : Quel est le nombre de spots sans check-in ?

#	CompteDeidspot
1	1

Annales Access : Agence

7.1 La base Agence

- La base de données Agence à récupérer : AgenceVoyages.mdb .
- La correction Agence au format access se trouve ici : CorrAgenceVoyages.mdb .

Une agence de voyages veut informatiser la gestion des voyages qu'elle propose (itinéraires, monuments et sites visités, réservations etc ...)

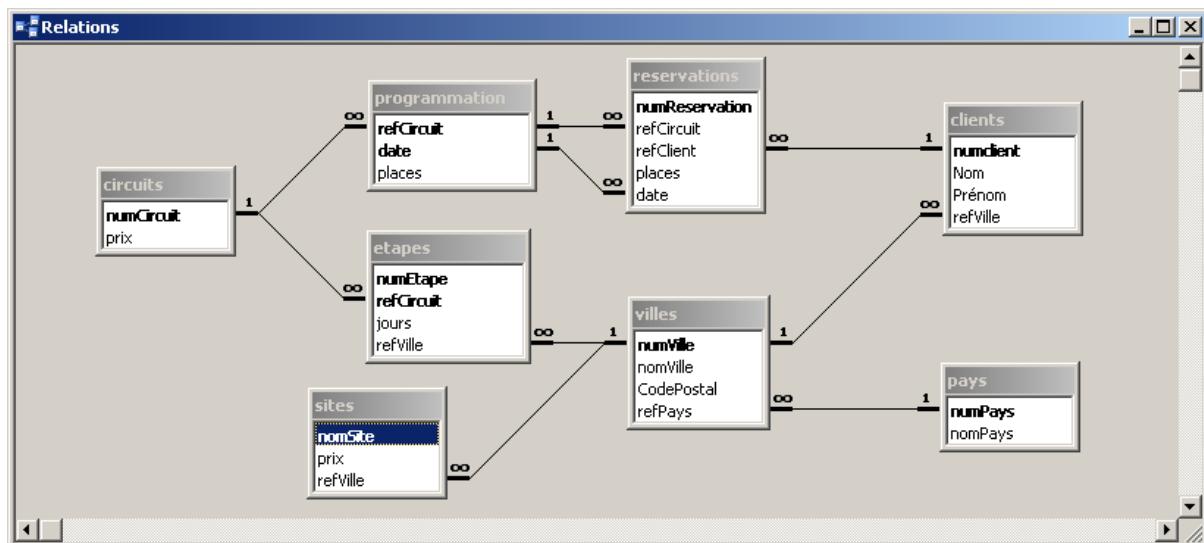


Fig. 7.1 – Relations

- Un circuit est identifié par un numéro et une séquence d'étapes. Une étape se déroule pendant un nombre de jours dans une ville donnée. Au cours de cette étape tous les sites de la ville, s'il y en a, sont visités.
- Un même circuit ne passe jamais plusieurs fois par la même ville. Les villes et les sites sont identifiées par leur nom.
- Un circuit peut être programmé plusieurs fois à des dates différentes. A chacune de ces programmations on associe un nombre de places. Deux programmations d'un même circuit peuvent avoir des nombres de places différents. Par contre, à moins d'une réduction exceptionnelle, le prix d'un circuit est toujours le même quelle que soit sa date de programmation. La durée d'un circuit est égale à la somme des durées de chacune des étapes.

- Une réservation, identifiée par un numéro, est effectuée pour le compte d'un client et concerne une programmation d'un circuit. Plusieurs places pour la même programmation du même circuit peuvent être réservées en une seule fois.

7.2 Questions

r1 : Quelles sont les villes de la base de données situées en France dans l'ordre alphabétique ?

#	nomVille
1	Amiens
2	Arras
3	Auchel
.	.
58	Wambrechies

r2 : Quels sont les sites situés en Italie ?

#	nomSite
1	La Basilique San Marco
2	Le Grand Canal
3	Le Palais des Doges
.	.
5	Piazza San Marco

r3 : Quels sont les villes visités en France ? (avec un site)

#	nomVille
1	Port Cotton
2	Port Maria
3	Villeneuve d'Ascq

r4 : Quels sont les numéros des circuits programmés au 1^o semestre 2012 dont le prix de base (sans tenir compte du prix de visite des sites) est compris entre 2000 € et 5000 € ?

#	numCircuit
1	5
2	6
3	7

r5 : Quel est le nombre d'étapes par numéro de circuit ?

#	refCircuit	CompteDenumEtape
1	1	1
2	2	1
3	3	1
.	.	.
10	10	8

r6 : Quel est le prix à payer par « Talonion » par circuit (sans tenir compte du prix des sites) ?

#	refCircuit	Expr1
1	2	3 480,00 €
2	9	57 000,00 €

r7 : Les circuits du mois de novembre subissent une baisse de 20%.Quel est le prix à payer par Duman par circuit et par date ?

#	numCircuit	Date	Expr1
1	5	07/11/2012	94256
2	2	15/04/2012	5800

r8 : Quels sont les clients qui rencontreront Duman au cours d'un circuit ? (on ne fera qu'une seule requête)

#	Nom
1	Leupoisson
2	Talonion
3	Use

r9 : Le prix total de chaque circuit comprend le prix de base et le prix de visite de chaque site. Calculer le prix total de chaque circuit pour les circuits qui ont des visites.

#	numCircuit	Expr1
1	1	1 360,00 €
2	2	1 360,00 €
3	3	1 250,00 €
.	.	.
8	9	6 000,00 €

r10 : Quels sont les numéros de circuit qui passent par toutes les villes d'un pays.

#	refCircuit
1	3
2	4
3	5
.	.
6	10

r11 : Quels sont les numéros des circuits qui font des étapes sans site

#	refCircuit
1	7
2	8
3	9
4	10

TD Select

Les requêtes présentées ici extraient des valeurs d'une seule table en ignorant totalement les autres tables de la base.

La clause SELECT indique les valeurs à obtenir pour chaque ligne, le FROM indique la table à utiliser pour obtenir ces résultats et le WHERE les conditions à remplir par les valeurs. Les lignes peuvent être triées avec la clause ORDER BY.

Les valeurs obtenues peuvent être les valeurs d'un champ de la table mais peuvent aussi être calculées au moyen d'expressions ou de fonctions portant sur plusieurs champs d'une même ligne.

La clause select produit une table qui peut être le résultat demandé mais qui peut aussi servir de point de départ à une autre clause select, on obtiendra alors des sous requêtes vues au chapitre suivant.

La base se trouve ici : `sport.s3db`.

8.1 La clause SELECT

La clause SELECT indique les valeurs qui constituent chaque ligne du résultat. Le FROM indique la ou les tables utilisées.

```
SELECT x, y
FROM T;
```

On peut également préfixer les noms des champs par le nom de la table : `T.x` Ce sera très important dans les prochains chapitres lorsque des champs de plusieurs tables auront le même nom.

```
SELECT T.x, T.y
FROM T;
```

8.1.1 Projection

r1 : Affichez les noms des clients

#	nom
1	Alavie
2	Aulet
3	Talonion
.	.
586	Céjoli

8.1.2 Lignes distinctes

r2 : Affichez les codes postaux des clients

#	codepostal
1	59100
2	02000
3	92230
.	.
586	74440

On remarque que les codes postaux sont affichés plusieurs fois

On affiche une seule fois les lignes avec DISTINCT :

```
SELECT DISTINCT x, y
FROM t;
```

r3 : Affichez les codes postaux des clients une seule fois

#	codepostal
1	59100
2	02000
3	92230
.	.
48	29000

8.1.3 Champs calculés - alias

On peut faire des opérations

```
SELECT 1 + 1;
```

r4 : Afficher les noms et le prix HT des produits (tva de 20%)

#	nom	prix / 1.2
1	Ninja baby	195.833333333333
2	Monkey	211.666666666667
3	Clean	166.666666666667
.	.	.
31	Cremes et gel indice 25	82.5

On peut nommer (ou renommer) les colonnes avec AS :

```
SELECT x AS y
FROM T;
```

r5 : Afficher les noms et le prix HT des produits

#	nom	HT
1	Ninja baby	195.833333333333
2	Monkey	211.666666666667
3	Clean	166.666666666667
.	.	.
31	Cremes et gel indice 25	82.5

8.2 La clause WHERE

La clause WHERE indique les conditions que doivent satisfaire les lignes qui donnent le résultat.

```
SELECT x, y
FROM T
WHERE conditions;
```

8.2.1 Filtrage

* Afficher les noms des produits de plus de 1000 €.

```
SELECT nom
FROM produits
WHERE prix > 1000;
```

On peut utiliser les opérateurs de comparaison usuels : =, ==, <, >, >=, <=, <>, les opérateurs logiques : OR, AND, NOT

Attention aux priorités. Ordre : comparaison, puis NOT, puis AND, puis OR.

r6 : Recherchez les clients dont le code postal est 59100

#	nom	prenom
1	Alavie	Irenée
2	Aibaitiz	Alfred
3	Deussan	Anne-Alice
.	.	.
13	Laladouchaifroide	Ulla

8.2.2 Caractères génériques

L'opérateur **LIKE** permet de faire des recherches en utilisant un modèle.

– % : n’importequelle suite de caractères (éventuellement vide)

– _ : n’importe-quel caractère (exactement un)

On utilise ce modèle avec LIKE et pas l’égalité (sinon vous cherchez le caractère %) :

```
SELECT x
FROM T
WHERE y LIKE modele;
```

r7 : Recherchez les clients dont le nom commence par Ta et qui habitent le Nord

#	nom	prenom
1	Taitraijeune	Thérèse
2	Tamair	Annick
3	Tamune Tablette D’Heuchokola	Jean
4	Tafraise	Théramène

r8 : Recherchez les clients dont le nom commence par Ta et qui habitent le Nord ou le Pas de Calais

#	nom	prenom
1	Taitraijeune	Thérèse
2	Tart’inn	Kimberley
3	Tamair	Annick
.	.	.
6	Taijouhai Parmillier	Betty Barbara Noël Kathy Tess Sandra Lucille Ella Vick

r9 : Recherchez les clients dont le nom commence par Ta et qui n’habitent ni le Nord ni le Pas de Calais

#	nom	prenom
1	Talonion	Thomas
2	Tabille	Raoul
3	Tabaniol	Alphonse
.	.	.
7	Tabaton	Jean

8.3 La clause ORDER BY

Order by permet de trier les lignes, par défaut les résultats sont dans l’ordre croissant (ASC) on peut inverser en utilisant DESC après le nom de la colonne :

```
SELECT x, y
FROM T
ORDER BY x ASC, y DESC;
```

r10 : Affichez les noms des clients dans l’ordre alphabétique des noms

#	nom
1	Aboucepartou
2	Adessin
3	Adlaplace
.	.
586	Zieubleux

r11 : Affichez les numéros de tous les produits dans l'ordre des numproduit

#	numproduit
1	2
2	3
3	4
.	.
31	32

r12 : Affichez les numéros de marques et les noms des produits de la catégorie 3 dans l'ordre decroissant des numéros de marques et l'ordre alphabétique des noms

#	refmarque	nom
1	17	Super guide tradition
2	16	Vetta
3	15	Alpinist
.	.	.
6	5	Super Khumbu

r13 : Afficher les noms et le prix HT des produits dans l'ordre croissant des prix HT (tva de 20%)

#	nom	HT
1	Tube-stick indice 4	40.8333333333333
2	Tube-stick indice 8	40.8333333333333
3	Tube-stick indice 15	41.6666666666667
.	.	.
31	Nepal top	1208.33333333333

r14 : Afficher les noms, le prix HT, le montant de la TVA et le prix TTC des produits dans l'ordre des noms

#	nom	HT	TVA	TTC
1	Alpinist	729.166666666667	145.833333333333	875
2	Aventure 65	790.833333333333	158.1666666666667	949
3	Charlotte	120.0	24.0	144
.
31	canard	120.0	24.0	144

8.4 Les fonctions

8.4.1 Opérations sur les chaines de caractères

Les Fonctions habituelles de texte sont présentes : upper(x), lower(x), substr(x,debut,nb)

* Afficher les numéros de départements d'où proviennent les clients

```
SELECT DISTINCT substr(codepostal,1,2) AS departement
FROM clients;
```

r15 : Afficher les départements des clients dans l'ordre croissant

#	dpt
1	01
2	02
3	13
.	.
17	93

r16 : Afficher les années des dates des commandes de la base

#	anneesCommandes
1	1995
2	1996
3	1997

8.4.2 Opérateur conditionnel

La commande CASE ... WHEN ... END permet d'utiliser des conditions de type si/sinon comme dans les langages de programmation. On peut l'utiliser dans n'importe quelle clause (select, where, order by ou having). Il existe 2 syntaxes, on utilisera cette syntaxe :

```
CASE
  WHEN cond1 THEN res1
  WHEN cond2 THEN res2
  ELSE res3
END -- AS y
```

– Afficher le nom des clients et leur ville. Si le code postal est 59000 ou 59800 on affichera ‘Lille’ sinon ‘autre’ :

```
SELECT nom,
CASE
  WHEN codepostal = '59000' THEN 'Lille'
  WHEN codepostal = '59800' THEN 'Lille'
  ELSE 'autre'
END AS ville
FROM clients;
```

– Même question avec un or :

```
SELECT nom,
CASE
  WHEN codepostal== '59000' OR codepostal=='59800' THEN 'Lille'
  ELSE 'autre'
END AS ville
FROM clients;
```

r17 : Les produits de la marque 3 subissent une hausse de 5%, ceux de la marque 4 une hausse de 10%, afficher les nouveaux prix des produits

#	nom	nouveauPrix
1	Ninja baby	235
2	Monkey	254
3	Clean	210.0
.	.	.
31	Cremes et gel indice 25	99

8.4.3 Fonctions statistiques

Les fonctions statistiques calculent des valeurs calculées sur l'ensemble des lignes. On trouve les fonctions habituelles count, avg (moyenne), sum, min, max

r18 : Affichez le nombre de produits dans la base.

#	nb
1	31

r19 : Affichez le maximum, le minimum et la moyenne des prix des produits.

#	maxPrix	minPrix	moyPrix
1	1450	49	461.096774193548

r20 : Affichez le nombre d'articles vendus.

#	nb
1	353

r21 : Calculer le nombre de marques

#	nb
1	26

r22 : Calculer le nombre de marques associées à des produits

#	nb
1	23

TD Jointures

La jointure permet une liaison entre 2 tables et d'utiliser tous les champs.

On peut distinguer les jointures croisées, les jointures internes et les jointures externes.

Les jointures peuvent s'écrire avec le prédicat JOIN ou dans le WHERE dans le cas des jointures internes. On utilisera ici le Where pour les jointures externes et le Join pour les jointures internes.

9.1 Jointure croisée

La jointure croisée réalise le produit cartésien (ou produit relationnel) des 2 tables. Chaque ligne d'une table est mise en relation avec toutes les lignes de l'autre table. Le nombre de lignes obtenu est égal à $\text{card}(T1) * \text{card}(T2)$. On crée cette jointure en indiquant le nom des tables dans le FROM.

```
SELECT x
FROM T1, T2
```

– Produit cartésien des tables villes et clients

```
SELECT ville, nom
FROM villes, clients
```

9.2 Jointure interne

La jointure interne fait correspondre les lignes des 2 tables en fonction d'un prédicat (condition) portant sur les colonnes des tables. On indique la condition de jointure dans le WHERE comme pour les conditions de filtrage. On peut considérer qu'il s'agit d'un filtre sur le produit cartésien des 2 tables. On verra plus loin qu'on peut utiliser l'opérateur JOIN

```
SELECT x
FROM T1, T2
WHERE T1.a = T2.b;
```

- Pour chaque ville afficher les noms des clients qui y habitent. On doit effectuer la jointure entre les codes postaux des 2 tables, les noms des champs étant identiques il faut les prefixer.

```
SELECT ville, nom
FROM villes, clients
WHERE villes.codepostal = clients.codepostal
ORDER BY ville;
```

On remarque que certaines villes n'apparaissent pas, personne n'habitant ces villes leur code postal n'est pas présent dans la table personnes et les lignes ne peuvent pas être liées. Pour afficher ces villes il faut utiliser une jointure externe.

- r1 : Pour chaque produit, afficher sa marque et son nom dans l'ordre des marques puis des produits**

#	nom	nom
1	Boreal	Ninja baby
2	Climbing shoes	Clean
3	EB	Monkey
.	.	.
31	vertical	Grimpeur

9.2.1 Avec condition de selection

On peut ajouter des conditions de filtrage comme dans tout SELECT. On les ajoutera avec un AND.

```
SELECT x
FROM T1, T2
WHERE T1.a = T2.b
AND conditions;
```

- r2 : Affichez les noms des clients de Lille**

#	nom
1	Deulisse
2	Peuleurido
3	Attan
.	.
14	Yoraitoux

- r3 : Affichez les noms des clients de Lille et Roubaix**

#	nom
1	Alavie
2	Deulisse
3	Peuleurido
.	.
27	Yoraitoux

- r4 : Afficher le nom du client qui a effectué la commande 1**

#	nom
1	Alavie

r5 : Affichez les noms des produits qui ont été achetés en quantité > 2 pour au moins une commande

#	nom
1	Diamond lady
2	Monkey
3	Super Khumbu
.	.
21	Palette

r6 : Pour chaque produit commandé dans la commande 1 affichez son nom et le prix à payer en fonction de la quantité

#	nom	totalLigne
1	Diamond lady	1650
2	Aventure 65	949
3	Super guide tradition	1312
4	Charlotte	144

r7 : Afficher le montant de la commande 1

#	totalCommande
1	4055

r9 : Affichez le nom et prénom des clients qui ont acheté de produits de plus de 1000 euros

#	nom	prenom
1	Alavie	Irenée
2	Duman	Mariette
3	Touplein Dirl Placar Chépalapenne D'Inacater	Nadja
.	.	.
26	Diotte	Kelly

r10 : Une réduction de 10% est accordée pour chaque produit commandé en quantité supérieure ou égale à 3. Pour chaque produit commandé dans la commande 1 affichez le nom du produit, le total sans réduction, le montant de la réduction et le montant avec réduction

#	nom	sansReduc	Reduc	avecReduc
1	Diamond lady	1650	165.0	1485.0
2	Aventure 65	949	0	949
3	Super guide tradition	1312	0	1312
4	Charlotte	144	0	144

r11 : Afficher le montant de la commande 1 en tenant compte de la réduction de la question précédente

#	montant
1	3890.0

9.2.2 Auto-jointures

Une table peut être jointe avec elle-même, il faut dans ce cas renommer les tables en indiquant leur alias après leur nom.

```
SELECT x
FROM T T1, T T2
WHERE T1.a = T2.b;
```

- Affichez le nom des produits de la même marque que le produit Miura.

```
select P2.nom
from produits P1, produits P2
where P1.refmarque = P2.refmarque
  and P1.nom = 'Miura'
  and P2.nom != 'Miura';
```

r12 : Affichez dans l'ordre alphabétique le nom et prénom des clients qui ont acheté au moins un des produits acheté par Alavie

#	nom	prenom
1	Aboucepartou	Thècle
2	Adlaplace	Andreï
3	Al Poubayl	Ahmed
.	.	.
42	Yala Couvertur Kigratt	Sandra

r13 : Affichez les nom et prénom des clients qui habitent la même ville que Peuleurido

#	nom	prenom
1	Attan	Charles
2	Baymole	Rémi
3	Deuf	John
.	.	.
13	Zettofrais	Mélanie

9.2.3 Sous requêtes

Les sous requêtes produisent des tables, on peut donc les utiliser comme les autres tables de la base et les placer dans le FROM. On place les sous requêtes dans le FROM et les conditions de jointure dans le WHERE. Pour nommer la sous requête il suffit d'indiquer son nom après son code.

```
SELECT x
FROM T1, (SELECT y FROM T2) ssreq
WHERE T1.y op ssreq.y;
```

- Affichez les noms et prénoms des clients qui habitent la même ville que Peuleurido avec une sous requête

```
select nom
from clients, villes,
  (select ville
   from villes, clients
   where villes.codepostal = clients.codepostal
```

```

and nom = 'Peuleurido') srl
where villes.codepostal = clients.codepostal
and villes.ville = srl.ville
and nom != 'Peuleurido';

```

r14 : Nom du client qui a acheté le produit le plus cher en janvier (de n'importe quelle année)

#	nom	prenom
1	Citation	Félix

r15 : Quels sont les clients qui ont acheté le produit le plus cher acheté par Pige

#	nom
1	Dumonde
2	Yala Couvertur Kigratt
3	Adrouille Toultan

9.2.4 Jointures généralisées

Une jointure ne se limite pas à une simple égalité (équi-jointure) on peut utiliser des expressions :

```

SELECT x
FROM T1, T2
WHERE T1.a > T2.b;

```

– nom des produits qui coutent plus cher que le produit ‘Manga’

```

-- avec une autojointure
select P2.nom
from produits P1, produits P2
where P1.nom = 'Manga'
and P2.prix > P1.prix;

-- avec une sous requête
select nom
from produits,
  (select prix from produits
   where nom = 'Manga') srl
where produits.prix > srl.prix;

```

r17 : Affichez les nom et prénom des clients qui ont effectué des commandes entre le jour de la première commande d’Alavie et le jour de sa dernière commande

#	nom
1	Aulet
2	Talonion
3	Leupoisson
.	.
8	Seiglandieu

9.3 Jointure externe

La jointure externe permet d'afficher des lignes pour lesquelles il n'y a pas correspondance et qui sont donc éliminées avec une jointure interne. On peut écrire LEFT JOIN ou LEFT OUTER JOIN

```
SELECT *
  FROM T1 LEFT JOIN T2 ON T1.a = T2.b;
```

On veut par exemple afficher les marques et les produits associés, la requête avec une jointure interne ne fait pas apparaître la marque aigle :

```
select marques.nom, produits.nom
  from marques, produits
 where nummarque = refmarque
 order by marques.nom;
```

Avec une jointure gauche on obtient toutes les marques même si nummarque de la marque aigle n'apparaît pas dans les produits. Toutes les lignes de la table de gauche (avant le join) sont affichées, si une valeur n'est pas présente dans la table de droite elle est remplacée par null, on le teste avec is null.

```
select marques.nom, produits.nom
  from marques left join produits on nummarque = refmarque
 order by marques.nom;
```

r18 : Affichez par ordre alphabétique les noms des clients qui n'ont rien commandé

#	nom
1	Adessin
2	Adlvy Yadlespoir
3	Adonaideux
.	.
477	Zieubleux

r20 : Affichez par ordre alphabétique les noms des clients qui ont acheté des produits mais pas TOUS les produits achetés par le client 1

#	nom
1	Aulet
2	Talonion
3	Tabille
.	.
97	Bussursène

TD Regroupements

Les regroupements permettent de grouper des lignes et d'effectuer des sous totaux (fonctions statistiques) sur ceux ci à la manière d'un sous total dans un tableur ou d'un tableau croisé, une seule ligne étant affiché par groupe. Les champs affichés doivent être regroupés donc on doit les trouver dans le GROUP BY. Les principales fonctions statistiques sont COUNT, COUNT(DISTINCT x), SUM, AVG, MIN, MAX.

10.1 La clause GROUP BY

```
SELECT x, fonction(y)
FROM T
GROUP BY x
```

- Affichez le nombre de produits par marque (sans la valeur 0)

```
select marques.nom, count(numproduit) as nbProduits
from marques, produits
where nummarque = refmarque
group by marques.nom
order by marques.nom;
```

10.2 Sémantique du GROUP BY

On commence par faire la requête sans GROUP BY avec les conditions dans le WHERE.

On groupe ensemble toutes les lignes qui sont identiques sur la ou les colonnes précisées dans le GROUP BY.

Pour chacun de ces groupes de lignes, on produit une seule ligne, qui contient ces valeurs identiques pour les colonnes du GROUP BY, et le résultat des opérations statistiques pour les autres.

r1 : Affichez le nombre de produits par catégorie

#	nom	nbProduits
1	Chaussons d_escalade	6
2	Chaussures d_alpinisme	6
3	Lunettes de glacier	4
.	.	.
6	Vetements de grimpe	4

r2 : Affichez le nombre de produits vendus par catégorie

#	nom	nbProduits
1	Chaussons d_escalade	61
2	Chaussures d_alpinisme	75
3	Lunettes de glacier	41
.	.	.
6	Vetements de grimpe	45

r3 : Affichez le nombre de commandes par an

#	annee	nbCommandes
1	1995	12
2	1996	71
3	1997	37

r4 : Affichez le nombre de produits différents commandés par le client Alavie par commande

#	numcommande	nbProduits
1	1	4
2	14	4

r5 : Affichez le nombre d'articles commandés par le client Alavie par commande

#	numcommande	nbArticles
1	1	6
2	14	6

r6 : Affichez le montant et la date des commandes effectuées par le client Tabille

#	date	montant
1	1996/12/22	126
2	1996/12/22	3111
3	1997/02/22	1955

10.3 GROUP BY HAVING

On peut filtrer le résultat de deux façons : filtrer la ou les tables sur lesquelles on effectue les opérations ou filtrer les lignes résultantes des groupements.

Si on veut filtrer les tables avant de grouper on veut alors filtrer suivant les champs ou les opérations calculées sur une seule ligne, par exemple substr pour calculer le département ou quantité * prix. On utilisera un WHERE comme pour toutes les requêtes vues précédemment.

Si par contre on veut filtrer après le groupement, donc sur le résultat d'une opération statistique comme count, il est clair que le filtre doit s'effectuer après le GROUP BY et pas avant au niveau

du WHERE. La clause HAVING placée après le GROUP BY permet d'écrire les conditions du filtre.

```
SELECT x, fonction(y) AS resultat
FROM T
WHERE condition sur x
GROUP BY x
HAVING condition sur resultat
```

- Affichez les noms des clients qui ont fait plus de 2 commandes

```
select nom
from clients, commandes
where numclient = refclient
group by nom, numclient
having count(numcommande) > 2;
```

On commence par faire la requête sans GROUP BY avec les conditions dans le WHERE.

On groupe ensemble toutes les lignes qui sont identiques sur la ou les colonnes précisées dans le GROUP BY.

Pour chacun de ces groupes de lignes, on produit une seule ligne, qui contient ces valeurs identiques pour les colonnes du GROUP BY, et le résultat des opérations statistiques pour les autres.

On applique les filtres présents dans le HAVING sur les résultats obtenus par chacun des groupes

r7 : Affichez le nom et le prénom des clients qui ont acheté des produits de plus de 3 marques

#	nom	prenom
1	Al Poubayl	Ahmed
2	Alavie	Irenée
3	Baul	Gérald
.	.	.
11	Touténe	Hervé

r8 : Affichez le nom et prénom des clients dont le total des commandes est de plus de 5000 €

#	nom	prenom
1	Alavie	Irenée
2	Daicacahuett'Livair	Yvan
3	Juskalapporte	Hercule
.	.	.
6	Yala Couvertur Kigratt	Sandra

r9 : Affichez le nom et prénom des clients qui ont effectué au moins une commande de plus de 5000 €

#	nom	prenom
1	Daicacahuett'Livair	Yvan
2	Juskalapporte	Hercule
3	Yala Couvertur Kigratt	Sandra

On peut utiliser des sous requêtes

r10 : Affichez le montant de la plus grande commande

#	maxMontant
1	6515

r11 : Affichez les noms et prénoms des clients qui ont achetés plus que Tabille (au total)

#	nom	prenom
1	Alavie	Irenée
2	Daicacahuett'Livair	Yvan
3	Juskalapporte	Hercule
.	.	.
5	Yala Couvertur Kigratt	Sandra

r12 : Affichez le nom et prénom du client qui a effectué la plus grande commande

#	nom	prenom
1	Juskalapporte	Hercule

r13 : Affichez le total des montants des commandes par an

#	annee	montant
1	1995	9009
2	1996	96147
3	1997	61085

r14 : Affichez l'année pour laquelle le total des montants des commandes est le plus faible

#	annee
1	1995

r15 : Affichez le montant des plus grandes commandes par client dans l'ordre décroissant des montants

#	nom	maxMontant
1	Juskalapporte	6515
2	Daicacahuett'Livair	5835
3	Yala Couvertur Kigratt	5225
.	.	.
109	Plupersonne	49

r16 : Affichez les noms des clients qui ont effectués en août 1996 une commande de montant supérieur à celle qui a été la plus grande en janvier 1996

#	nom	prenom
1	Juskalapporte	Hercule

10.4 Groupement avec jointure externe

Lorsqu'on effectue une jointure interne entre 2 tables on obtient les lignes pour lesquelles une association est possible. En regroupant suivant un champ on obtient donc au minimum une ligne, la valeur minimale d'un count sera 1. La jointure externe permet d'obtenir les lignes d'une table pour lesquelles il n'y a pas d'association, la valeur minimale d'un count pourra donc être égale à 0.

r17 : Affichez le nombre de produits par marque avec la valeur 0

#	nom	nbProduits
1	Boreal	1
2	Climbing shoes	1
3	EB	1
.	.	.
26	vertical	1

TD Révisions

11.1 La base

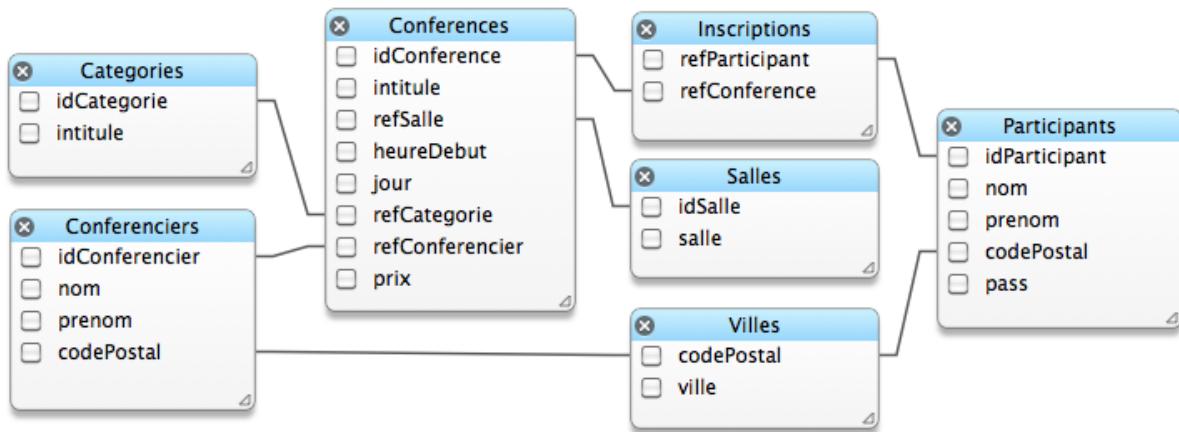


Fig. 11.1 – Relations

- La base de données : `conferences.s3db` .

11.2 Questions

r1 : Affichez le nom et le prénom du conférencier de code 1

#	nom	prenom
1	Time Is On My	Said

r2 : Affichez la liste des conferences de catégorie ‘Java Core’ et ‘Cloud’ dans l’ordre alphabétique.

#	intitule
1	Cassandra
2	Hadoop
3	Hbase
.	.
7	iText

r3 : Affichez le nom et prenom des conferenciers du nord ou du pas-de-calais dans l'ordre alphabétique.

#	nom	prenom
1	Bofrere	Edmond
2	Time Is On My	Said
3	Toutene	Herve

r4 : Affichez l'intitule des catégories pour lesquelles il n'y a pas de conférences.

#	intitule
1	Architecture

r5 : Affichez l'intitule et le nombre de participants des conferences dans l'ordre decroissant du nombre de participants puis dans l'ordre alphabétique des conférences.

#	intitule	NbParticipants
1	Cassandra	8
2	Hbase	7
3	Spring Roo	6
.	.	.
15	Android	1

r6 : Affichez le nom des categories qui ont au moins 2 conférences à 14h (pas forcemment le même jour).

#	intitule
1	Web Frameworks
2	Cloud

r7 : Avec une AUTOJOINTURE, affichez le nom des participants qui assistent aux memes conférences qu'Alavie dans l'ordre alphabétique.

#	nom
1	Aidelaine
2	Duman
3	Leupoisson
.	.
9	Use

r8 : Avec une SOUS REQUETE, affichez l'intitulé des conférences qui ont lieu dans la même salle que 'MongoDB'.

#	intitule
1	Cassandra
2	Hadoop
3	Hbase

r9 : Affichez l'intitule et le nombre de participants de la conférences qui a le plus de participants.

#	intitule	NbParticipants
1	Cassandra	8

r10 : Affichez le nombre de conférences par salle dans l'ordre alphabétique des salles avec la valeur 0 s'il n'y a pas de conférence dans une salle.

#	salle	nbConfs
1	Room1	0
2	Room2	0
3	Room3	0
.	.	.
7	Room9	4

r11 : Les participants vip ont un pass qui donne lieu a une reduction de 30% sur toutes les conférences, affichez le montant sans réduction (brut) et le montant avec réduction (net) à régler par participant.

#	nom	brut	net
1	Adessin	105	73.5
2	Aidelaine	145	101.5
3	Alavie	100	70.0
.	.	.	.
20	Use	120	120

11.3 Questions Bis

r1b : Affichez l'intitulé de la conférence de code 1

#	intitule
1	Smack

r2b : Affichez la liste des conferences qui ont lieu dans les salles ‘Room5’ et ‘Room9’ dans l’ordre alphabétique

#	intitule
1	Cassandra
2	Groovy
3	Hadoop
.	.
8	Wicket

r3b : Affichez le nom et prenom des participants du nord ou du pas-de-calais dans l’ordre alphabétique.

#	nom	prenom
1	Adessin	Blanche
2	Aidelaine	Gilles
3	Alavie	Irenee
.	.	.
18	Use	Jacques

r4b : Affichez les salles pour lesquelles il n’y a pas de conférences.

#	salle
1	Room1
2	Room2
3	Room3

r5b : Affichez le nom, le prénom des participants et leur nombre d'inscriptions aux conférences ds l'ordre decroissant du nombre d'inscriptions puis dans l'ordre alphabétique des participants.

#	nom	prenom	NbConferences
1	Aidelaine	Gilles	4
2	Leifeilei	Olivier	4
3	Leupoisson	Benoit	4
.	.	.	.
20	Alor	Marthe	1

r6b : Affichez le nom des participants qui se sont inscrits à au moins 2 conférences le 15/11/2010.

#	nom
1	Alavie
2	Aulet
3	Talonion
.	.
9	Aidelaine

r7b : Avec une AUTOJOINTURE, affichez le nom des conférences qui ont lieu dans la même salle que 'MongoDB'.

#	intitule
1	Cassandra
2	Hadoop
3	Hbase

r8b : Avec une SOUS REQUETE, affichez le nom des participants qui assistent aux memes conférences qu'Alavie dans l'ordre alphabétique.

#	nom
1	Aidelaine
2	Aidelaine
3	Duman
.	.
10	Use

r9b : Affichez l'intitule et le nombre de conférences de la catégorie qui a le plus de conférence.

#	intitule	NBCConferences
1	Web Frameworks	4
2	Cloud	4

r10b : Affichez le nombre de conférences par catégorie dans l'ordre alphabétique des catégories avec la valeur 0 s'il n'y a pas de conférence dans une catégorie.

#	intitule	nbConfs
1	Architecture	0
2	Cloud	4
3	Desktop	2
.	.	.
8	Web Frameworks	4

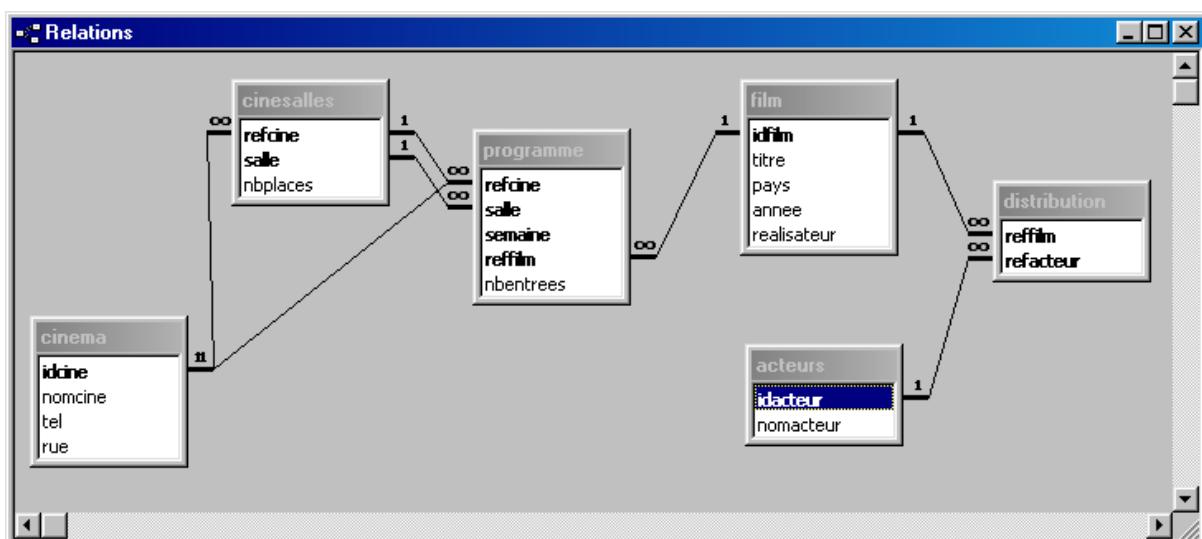
r11b : La conférence de code 2 voit son prix diminuer de 30%, affichez le montant avec réduction à régler par participant.

#	nom	brut	net
1	Adessin	105	105
2	Aidelaine	145	134.5
3	Alavie	100	100
.	.	.	.
20	Use	120	120

Cinema1

12.1 La base

- La base Cinema se trouve ici : cinema.s3db .



12.2 Questions

r1 : Afficher les noms des cinémas dans l'ordre alphabétique

#	nomcine
1	gaumont
2	la nef
3	le club
.	.
10	vox

r2 : Titre, année de sortie et réalisateur des films français

#	titre	annee	realisateur
1	La baie des anges	1962	Demy
2	Mon cher sujet	1989	Mieville
3	Baxter	1989	Boivin
.	.	.	.
14	Les visiteurs	1993	3. Clavier

r3 : Titres des films avec leur réalisateur, dans l'ordre alphabétique des titres de film

#	titre	realisateur
1	Annee du soleil calme	Krzysztof
2	Appocalypse now	Coppola
3	Baxter	Boivin
.	.	.
33	Veuve mais pas trop	Truffaut

r4 : Nom des réalisateurs qui commencent par un T

#	realisateur
1	Truffaut

r5 : Noms et numéros des acteurs du film “Cop”

#	nomacteur
1	Eastwood
2	Woods
3	Deneuve
.	.
5	Haid

r6 : Années de sortie des films tournés avec Gabin dans l'ordre des dates

#	annee
1	1962
2	1989

r7 : Films à l'affiche de la quatrième semaine dans chaque salle du cinéma gaumont

#	salle	titre
1	1	Mon cher sujet
2	2	Annee du soleil calme
3	3	Un poisson nomme wanda
.	.	.
6	6	La boca del lobo

r8 : Acteurs qui ont tourné avec Truffaut comme réalisateur

#	idacteur	nomacteur
1	23	Pfeiffer
2	19	Modine
3	29	Stockwell
.	.	.
5	24	Piccoli

r9 : Adresses des cinémas où passe un film réalisé par Crichton

#	nomcine	rue
1	gaumont	alsace lorraine
2	la nef	edouard rey
3	ugc royal	clot bey
.	.	.
5	les 6 rex	saint jacques

r10 : Rues, salles et nombres de places des cinémas où sont joués des films sortis en 1984 et dans lesquels joue Delon

#	rue	salle	nbplaces
1	alsace lorraine	5	100
2	saint jacques	1	400
3	strasbourg	1	200
4	grenette	6	100

r11 : Titres et semaines de programmation des films dans lesquels jouent Deneuve ou Eastwood

#	titre	semaine
1	Cop	1
2	Cop	2
3	Cop	3
.	.	.
8	La derniere cible	4

r12 : Pour chaque réalisateur, nom des acteurs qui ont tourné sous sa direction en 1989, trié par ordre alphabétique de nom de réalisateur puis par ordre alphabétique inverse de nom d'acteur.

#	realisateur	nomacteur
1	Boivin	Spiesser
2	Boivin	Piccoli
3	Boivin	Mercure
.	.	.
30	Truffaut	Modine

13.1 Questions

r1 : nb de films par acteur

#	nomacteur	nbFilms
1	Bueno	1
2	Clarkson	1
3	Cleese	1
.	.	.
34	Woods	1

r2 : Afficher pour le réalisateur Mieville, le nombre de films, en nommant nbreFilms le titre de la colonne.

#	nbreFilms
1	1

r3 : Afficher pour chaque réalisateur, le nombre de films, en commençant par celui qui a réalisé le plus de films.

#	realisateur	nbFilms
1	Parker	3
2	Besson	2
3	Carne	2
.	.	.
27	Stone	1

r4 : Afficher pour chaque réalisateur, le nombre de films, pour les réalisateurs qui ont fait plus d'un film.

#	realisateur	nbFilms
1	Besson	2
2	Carne	2
3	Coppola	2
.	.	.
5	Truffaut	2

r5 : Nombre de semaines de programmation de chaque film.

#	titre	nbSemaines
1	Annee du soleil calme	4
2	Baxter	4
3	Cop	4
.	.	.
9	Veuve mais pas trop	5

r6 : Titre des films qui passaient la semaine 4, avec le nombre de cinémas dans lesquels ils étaient programmés, limités à ceux programmés dans au moins 4 cinémas, triés dans l'ordre du nombre de cinémas dans lesquels ils étaient programmés.

#	titre	nbCinemas
1	Annee du soleil calme	5
2	La baie des anges	5
3	Un poisson nomme wanda	5
.	.	.
8	Mon cher sujet	4

r7 : Avec une sous-requête, afficher les films qui ont été tournés la même année que Nikita”

#	titre
1	Annee du soleil calme
2	Cop
3	La boca del lobo
.	.
8	Cyrano de Bergerac

r8 : Box office (=les films dans l'ordre décroissant du nombre d'entrées) de la semaine 4 (titre, réalisateur, nombre d'entrées).

#	titre	realisateur	nbTotalEntrees
1	Un poisson nomme wanda	Crichton	3013
2	Annee du soleil calme	Krzysztof	2679
3	Veuve mais pas trop	Truffaut	2364
.	.	.	.
9	Baxter	Boivin	101

r9 : Avec une auto jointure, afficher les films qui ont été tournés la même année que Nikita

#	titre
1	Annee du soleil calme
2	Baxter
3	Cop
.	.
8	Veuve mais pas trop

r10 : Avec une aujointure Acteurs partenaires de Deneuve

#	nomacteur
1	Eastwood
2	Haid
3	Warren
.	.
7	Hunt

r11 : Réaliseurs ayant joué dans des films qu'ils n'ont pas réalisés eux-même

#	nomacteur
1	Eastwood
2	Truffaut

r12 : Réalisateur et noms des films précédés dans la programmation par un film dans lequel joue Pfeiffer

#	nomacteur	nbPartenaires
1	Delon	9
2	Piccoli	9
3	Deneuve	7
.	.	.
34	Vega	1

r13 : Avec une sous requête Acteurs partenaires de Deneuve

#	nomacteur
1	Eastwood
2	Woods
3	Warren
.	.
7	Delon

r14 : Liste des acteurs avec le nombre de leurs partenaires (c'est à dire le nombre des acteurs avec lesquels ils ont joué dans au moins un film), dans l'ordre décroissant du nombre de partenaires.

#	nomacteur	partenaires
1	Delon	9
2	Piccoli	9
3	Deneuve	7
.	.	.
34	Vega	1

r15 : Titre, année et pays des films pour lesquels il existe un film réalisé la même année dans le même pays.

#	titre	annee	pays
1	La dernière cible	1984	USA
2	Birdy	1984	USA
3	Mon cher sujet	1989	France
.	.	.	.
15	JFK	1991	USA

r16 : Analyse de la concurrence : pour chaque paire de cinémas (C1, C2), nombre de films identiques programmés simultanément. La liste doit être triée par C1 puis par nombre de films simultanés (dans l'ordre décroissant), puis par C2.

#	nomcine	Communs	nomcine
1	gaumont	7	gaumont
2	gaumont	5	la nef
3	gaumont	5	le club
.	.	.	.
64	ugc royal	2	les dauphins

r17 : Noms des cinémas ayant programmé un même film dans des salles différentes.

#	idcine	nomcine
1	1	le club
2	7	ugc royal

r18 : Pour chaque pays, donnez le nombre moyen d'acteurs par film.

#	pays	x
1	Espagne	2
2	Pologne	4
3	USA	4
4	France	5

r19 : Pour chaque année, noms des cinémas ayant eu à l'affiche tous les films sortis cette année.

#	annee	nomcine
1	1962	le club
2	1962	les dauphins
3	1962	les 6 rex
.	.	.
5	1962	pathe grenette

Cinema3

14.1 Questions

r1 : films qui ont au moins 6 acteurs

#	reffilm	count(refacteur)
1	8	6

r2 : nombre de films par decennie

#	decennie	nbFilms
1	30	1
2	40	1
3	60	3
.	.	.
6	90	8

r3 : les acteurs sont payes 200000 euros et 0.5 euro par entree, calculer le salaire total des acteurs

#	nomacteur	salaire
1	Bueno	201360.0
2	Clarkson	201758.5
3	Cleese	208952.5
.	.	.
34	Woods	203754.0

r4 : pourcentage de remplissage des salles du cinema gaumont la semaine 4

#	salle	pourcentage
1	1	9
2	2	51
3	3	47
.	.	.
6	6	41

r5 : films qui n'ont pas ete programmes

#	titre
1	Nikita
2	L'amant
3	La belle histoire
.	.
24	Les visiteurs

r6 : cinema qui n'ont rien programme la semaine 4

#	nomcine
1	lux
2	vox

r7 : film qui a fait le plus grand nombre d'entrees

#	titre
1	Un poisson nomme wanda

r8 : films dont le nombre d'entrees est superieur a la moyenne des entrees par semaine

#	titre
1	Annee du soleil calme
2	Cop
3	Un poisson nomme wanda
.	.
7	Veuve mais pas trop

r9 : films dont le nombre d'entrées est égal à +/- 5 % de la moyenne des entrées par semaine

#	titre
1	Cop
2	Un poisson nomme wanda
3	Mon cher sujet
4	Veuve mais pas trop

r10 : les realisateurs sont payes 100000 euros avec une subvention de 50000 euros s'ils réalisent un film en France, calculer le salaire total des realisateurs

#	realisateur	salaire
1	Annaud	150000
2	Besson	300000
3	Blier	150000
.	.	.
27	Truffaut	250000

r11 : nombre de films par mois et par cinema, on notera Jamais si un cinema ne programme aucun film

#	nomcine	mois	nbFilms
1	gaumont	Janvier	7
2	la nef	Janvier	6
3	le club	Janvier	7
.	.	.	.
12	vox	Jamais	0

r12 : realisateur qui a fait le + grand nombre de films

#	realisateur
1	Parker

r13 : acteur le plus rentable (+ grand nombre d'entrées)

#	nomacteur
1	Piccoli

r14 : realisateurs qui ne sont pas acteurs ss jointure gauche

#	realisateur
1	Krzysztof
2	Harris
3	Lombardi
.	.
25	3. Clavier

r15 : realisateurs qui ne sont pas acteurs avec jointure gauche

#	realisateur
1	Krzysztof
2	Harris
3	Lombardi
.	.
25	3. Clavier

r16 : par cinema liste des films qui ont un nb d'entrées inférieur à la moitié du nb de place tt salles confondues la semaine 4

#	nomcine	titre
1	le club	La baie des anges
2	les dauphins	Annee du soleil calme
3	les dauphins	La baie des anges
.	.	.
11	le melies	Veuve mais pas trop

r17 : nb de films par première lettre (A :2) : substr(champ,debut,nb)

#	initiale	nb
1	A	2
2	B	2
3	C	2
.	.	.
17	V	1

r18 : realisateur qui a tourné avec le plus grand nombre d'acteurs

#	realisateur
1	Boivin

r19 : les acteurs ont un intérressement de 0.5 euro par entrée d'un film si le nombre d'entrées de ce film dépasse 10000, pour chaque acteur ayant obtenu des intéressesments afficher leur montant total

#	nomacteur	interessement
1	Cleese	8952.5
2	Curtis	8952.5
3	Gabin	5452.0
.	.	.
8	Wilson	5452.0

r20 : le tarif des entrées des cinémas est de 6€ sauf ‘le melies’ : 5€ et ‘le club’ : 3€. Quelle est la recette totale des cinemas

#	nomcine	recette
1	gaumont	138160
2	la nef	152032
3	le club	17202
.	.	.
8	ugc royal	53848

r21 : pour ‘le club’, par salle et par film, afficher ‘abandonner’ si le taux d’occupation de la semaine 4 est inférieur à 20% des places de la salle, ‘ouvrir’ s’il est supérieur à 80% sinon ‘continuer’

#	salle	titre	action
1	1	Veuve mais pas trop	abandonner
2	2	La boca del lobo	abandonner
3	3	La baie des anges	ouvrir
.	.	.	.
5	5	Cop	continuer

r22 : par cinema et par semaine, liste des salles sans programmation

#	nomcine	salle
1	lux	1
2	les 6 rex	6

r23 : acteurs qui n’ont pas tourné de films au USA mais qui en ont tourné en Espagne

#	nomacteur
1	Bueno
2	Vega

Correction TD 1

r3 : Affichez les noms et prénoms des clients dont le nom commence par “Ta”.

#	r3
tables ouvertes	clients
jointures	
champs affiches	clients.Nom clients.Prénom
criteres	clients.Nom = “Ta*”
regroupe par	
trie par	

r4 : Recherchez les clients dont le nom commence par “Ta” et qui habitent le Nord

#	r4
tables ouvertes	clients
jointures	
champs affiches	clients.Nom clients.Prénom
criteres	clients.Nom = “Ta*” clients.Code_Postal = “59*”
regroupe par	
trie par	

r5 : Recherchez les clients dont le nom commence par “Ta” et qui habitent le Nord ou le Pas de Calais

#	r5
tables ouvertes	clients
jointures	
champs affiches	clients.Nom clients.Prénom
criteres	(clients.Nom = “Ta*” ET clients.Code_Postal = “59*”) OU (clients.Nom = “Ta*” ET clients.Code_Postal = “62*”)
regroupe par	
trie par	

r6 : Recherchez les clients dont le nom commence par “Ta” et qui n’habitent ni le Nord ni le Pas de Calais

#	r6
tables ouvertes	clients
jointures	
champs affiches	clients.Nom clients.Prénom
criteres	clients.Code_Postal <> “59*” ET clients.Code_Postal <> “62*”
regroupe par	
trie par	

Correction TD 2

r6a : Afficher les noms des clients dont le nom commence par Ta et qui ont effectué une commande (On n'affichera que le nom du client) Vous devez trouver 5 réponses. Quel est le rôle de la jointure ? Comparez avec la requête 3.

#	r6a
tables ouvertes	clients, commande
jointures	J5
champs affiches	clients.Nom
criteres	clients.Nom = "Ta*"
regroupe par	
trie par	

r6b : Même requête que la 6a, allez dans les propriétés de la requête et dans le champ « valeurs distinctes » affichez « oui » Combien trouvez vous de réponses ? Comparez avec la requête 6a. Comparez le code sql des requêtes 6a et 6b.

#	r6b
tables ouvertes	clients, commande
jointures	J5
champs affiches	Valeurs distinctes clients.Nom
criteres	clients.Nom = "Ta*"
regroupe par	
trie par	

r7 : Afficher les noms des clients qui ont effectué une commande en janvier 97

#	r7
tables ouvertes	clients, commande
jointures	J5
champs affiches	Valeurs distinctes clients.Nom
criteres	commande.Date = "/01/"
regroupe par	
trie par	

r8 : Afficher le nom du client qui a effectué la commande 1 ainsi que les produits com-

mandés

#	r8
tables ouvertes	produit, clients, commande, procom
jointures	J5 J4 J3
champs affiches	clients.Nom produit.nom
criteres	commande.numcommande = 1
regroupe par	
trie par	

r9 : Afficher les numéros de commande et les produits commandés par un client dont le code sera saisi au clavier

#	r9
tables ouvertes	produit, commande, procom
jointures	J4 J3
champs affiches	commande.refclient produit.nom
criteres	commande.refclient = code client ?
regroupe par	
trie par	

r10 : Afficher les produits commandés par un client dont le code et le code du numéro de commande seront saisis au clavier

#	r10
tables ouvertes	produit, commande, procom
jointures	J4 J3
champs affiches	produit.nom
criteres	commande.refclient = code client ? commande.numcommande = code commande ?
regroupe par	
trie par	

r11 : Commencez par afficher la quantité et le prix des produits de la commande 1. Positionnez le curseur dans “Champ” d’une nouvelle colonne puis cliquez sur Générer. Générez l’expression procom.quantité * produit.Prix. Modifiez l’entête de colonne : Total au lieu de expr1. Ecrivez le formalisme de la requête.

#	r11
tables ouvertes	produit, procom
jointures	J3
champs affiches	procom.quantité produit.Prix Total : procom.quantité*produit.Prix
criteres	procom.refcommande = 1
regroupe par	
trie par	

r12 : La TVA est de 19.6% et les prix des produits sont donnés tva incluse. Pour chaque produit commandé dans la commande 1 affichez le montant de la TVA, le montant hors taxe et le montant TTC. Commencez par calculer le montant hors taxe et le montant TTC puis calculez le montant de la TVA.

#	r12
tables ouvertes	produit, procom
jointures	J3
champs affiches	procom.quantité produit.Prix TTC : procom.quantité*produit.Prix HT : TTC/1.2 TVA : TTC-HT
criteres	procom.refcommande = 1
regroupe par	
trie par	

r13 : Une réduction de 10% est accordée pour chaque produit commandé en quantité supérieure ou égale à 3. Pour chaque produit commandé dans la commande 1 affichez le nom du produit, le total sans réduction, le montant de la réduction et le montant avec réduction (On ne tient plus compte de la question 12).

#	r13
tables ouvertes	produit, procom
jointures	J3
champs affiches	produit.nom ssReduc : procom.quantité*produit.Prix Reduc : vraifaux (procom.quantité>=3,0,1,0)*ssReduc AvecReduc : ssReduc-Reduc
criteres	procom.refcommande = 1
regroupe par	
trie par	

r14 : La direction décide d'accorder une remise dépendant de la catégorie des produits. Il y a 6 catégories, si le code de la catégorie est inférieur ou égal à 3, la remise est de 10%, si le code est égal à 4 ou 5 la remise est de 15%, s'il s'agit de sac à dos (catégorie 6) la remise est de 20 %. Pour la commande 1 affichez le nom du produit, le code de la catégorie et le total compte tenu de la remise.

#	r14
tables ouvertes	produit, procom
jointures	J3
champs affiches	produit.nom produit.refcateg Total : vraifaux (produit.refcateg<=3,0.9, vraifaux (produit.refcateg<=5,0.85,0.8))*procom.quantité*produit.Prix
criteres	procom.refcommande = 1
regroupe par	
trie par	

r15 : Affichez le nombre de produits par marque par ordre alphabétique.

#	r15
tables ouvertes	marques, produit
jointures	J2s
champs affiches	marques.nom nbProduits : Compte(produit.numproduit)
criteres	
regroupe par	marques.nom
trie par	marques.nom

Correction TD 3

r18 : Affichez le nombre de produits dans la base.

#	r18
tables ouvertes	produit
jointures	
champs affiches	nbProduits : Compte(produit.numproduit)
critères	
regroupe par	
trie par	

r19 : Affichez le maximum, le minimum et la moyenne des prix des produits.

#	r19
tables ouvertes	produit
jointures	
champs affiches	maxPrix : Max(produit.Prix) minPrix : Min(produit.Prix) moyPrix : Moyenne(produit.Prix)
critères	
regroupe par	
trie par	

r20 : Affichez le nombre d'articles vendus.

#	r20
tables ouvertes	procom
jointures	
champs affiches	nbArticles : Somme(procom.quantité)
critères	
regroupe par	
trie par	

r21 : Affichez le nombre de produits différents commandés par le client Alavie par commande.

#	r21
tables ouvertes	clients, commande, procom
jointures	J5 J4
champs affiches	commande.numcommande nbProduits : Compte(procom.refproduit)
criteres	clients.Nom = "alavie"
regroupe par	commande.numcommande
trie par	

r22 : Affichez le nombre d'articles commandés par le client Alavie par commande.

#	r22
tables ouvertes	clients, commande, procom
jointures	J5 J4
champs affiches	commande.numcommande nbArticles : Somme(procom.quantité)
criteres	clients.Nom = "Alavie"
regroupe par	commande.numcommande
trie par	

r23 : Affichez le montant et la date des commandes effectuées par le client Alavie.

#	r23
tables ouvertes	produit, clients, commande, procom
jointures	J5 J4 J3
champs affiches	Montant : Somme(procom.quantité*produit.Prix) commande.Date
criteres	clients.Nom = "Alavie"
regroupe par	commande.Date commande.numcommande
trie par	

r24 : Affichez le montant et la date des commandes effectuées par le client Alavie avec une réduction de 20% sur l'article de ref 15

#	r24
tables ouvertes	produit, clients, commande, procom
jointures	J5 J4 J3
champs affiches	commande.Date Montant : Somme(procom.quantité*produit.Prix* vraifaux (produit.numproduit = 15,0.8,1))
criteres	clients.Nom = "Alavie"
regroupe par	commande.numcommande commande.Date
trie par	

r25 : Affichez le montant et la date des commandes effectuées par un client dont le code sera saisi au clavier

#	r25
tables ouvertes	produit, commande, procom
jointures	J4 J3
champs affiches	Valeurs distinctes Monatnt : Somme(procom.quantité*produit.Prix) commande.Date
criteres	commande.refclient = code client ?
regroupe par	commande.numcommande commande.Date
trie par	

r26 : Affichez le nom et prénom des clients dont le total des commandes est de plus de 5000 €

#	r26
tables ouvertes	produit, clients, commande, procom
jointures	J5 J4 J3
champs affiches	Valeurs distinctes clients.Nom
criteres	Somme(procom.quantité*produit.Prix)>5000
regroupe par	clients.Nom
trie par	

r27 : Affichez le nom et prénom des clients qui ont effectué une commande de plus de 5000 €

#	r27
tables ouvertes	produit, clients, commande, procom
jointures	J5 J4 J3
champs affiches	Valeurs distinctes clients.Nom
criteres	Somme(procom.quantité*produit.Prix)>5000
regroupe par	clients.Nom commande.numcommande
trie par	

r28 : Affichez par ordre alphabétique inverse les produits qui ont été vendus en quantité >= 15

#	r28
tables ouvertes	produit, procom
jointures	J3
champs affiches	produit.nom
criteres	Somme(procom.quantité)>=15
regroupe par	produit.nom
trie par	produit.nom DESC

r29 : Affichez par ordre alphabétique les produits qui ont été vendus en quantité >= 15 et dont le prix unitaire est > 900 €

#	r29
tables ouvertes	produit, procom
jointures	J3
champs affiches	produit.nom
criteres	produit.Prix>900 Somme(procom.quantité)>=15
regroupe par	produit.nom
trie par	produit.nom

Correction TD 4

r31 : Affichez le nombre de produits par marque

#	r31
tables ouvertes	marques, produit
jointures	J2s
champs affiches	marques.nom nbProduits : Compte(produit.numproduit)
criteres	
regroupe par	marques.nom
trie par	

r32 : Affichez le nombre de produits par catégorie

#	r32
tables ouvertes	categ, produit
jointures	J1
champs affiches	categ.descriptif nbProduits : Compte(produit.numproduit)
criteres	
regroupe par	categ.descriptif
trie par	

r33 : Affichez le nombre d'articles vendus par catégorie

#	r33
tables ouvertes	categ, produit, procom
jointures	J1 J3
champs affiches	categ.descriptif nbVendus : Somme(procom.quantité)
criteres	
regroupe par	categ.descriptif
trie par	

r34 : Affichez les noms des articles de prix supérieur à la moyenne des prix

r34a : Moyenne des prix

#	r34a
tables ouvertes	produit
jointures	
champs affiches	moyPrix : Moyenne(produit.Prix)
criteres	
regroupe par	
trie par	

r34 : Requête finale

#	r34
tables ouvertes	produit, r34a
jointures	
champs affiches	produit.nom
criteres	produit.Prix > r34a.moyPrix
regroupe par	
trie par	

r35 : Affichez le nom et le prénom des clients qui ont acheté des produits de plus de 3 marques différentes

r35a : Marques différentes par client

#	r35a
tables ouvertes	produit, commande, procom
jointures	J4 J3
champs affiches	Valeurs distinctes commande.refclient produit.refmarques
criteres	
regroupe par	
trie par	

r35 : Requête finale

#	r35
tables ouvertes	clients, r35a
jointures	clients.numclient = r35a.refclient
champs affiches	clients.Nom
criteres	Compte(r35a.refmarques) > 3
regroupe par	clients.Nom
trie par	

r36 : Affichez le montant de la première commande effectuée par un client dont le code sera saisi au clavier. S'il y a plusieurs commandes le même jour on les affichera toutes.

r36a : Date de première commande du client

#	r36a
tables ouvertes	commande
jointures	
champs affiches	MinDeDate : Min(commande.Date) commande.refclient
criteres	commande.refclient = code client ?
regroupe par	commande.refclient
trie par	

r36 : Requête finale

#	r36
tables ouvertes	r36a, produit, commande, procom
jointures	J4 J3 r36a.refclient = commande.refclient r36a.MinDeDate = commande.Date
champs affiches	Montant : Somme(procom.quantité*produit.Prix)
criteres	
regroupe par	
trie par	

r37 : Affichez le nom et prénom des clients qui n'ont pas acheté de produits de moins de 1000 euros. Les clients n'ayant rien commandé ne seront pas inclus.

#	r37
tables ouvertes	produit, clients, commande, procom
jointures	J5 J4 J3
champs affiches	clients.Nom clients.Prénom
criteres	Min(produit.Prix)>=1000
regroupe par	clients.Nom clients.Prénom
trie par	

r38 : Affichez dans l'ordre alphabétique le nom et prénom des clients qui ont acheté au moins un des produits acheté par Alavie

r38a : Produits achetés par Alavie

#	r38a
tables ouvertes	clients, commande, procom
jointures	J5 J4
champs affiches	procom.refproduit
criteres	clients.Nom = "Alavie"
regroupe par	
trie par	

r38 : Requête finale

#	r38
tables ouvertes	clients, commande, procom, r38a
jointures	J5 J4 procom.refproduit = r38a.refproduit
champs affiches	Valeurs distinctes clients.Nom
criteres	clients.Nom <> "Alavie"
regroupe par	
trie par	clients.Nom

Autojointures

r39 : Affichez les nom et prénom des clients qui habitent la même ville que Peuleurido en utilisant une sous-requête

r39a : Ville de Peuleurido

#	r39a
tables ouvertes	Villes, clients
jointures	J6
champs affiches	Villes.Ville
criteres	clients.Nom = "Peuleurido"
regroupe par	
trie par	

r39 : Requête finale

#	r39
tables ouvertes	Villes, clients, r39a
jointures	J6 Villes.Ville = r39a.Ville
champs affiches	clients.Nom
criteres	clients.Nom <> "Peuleurido"
regroupe par	
trie par	

r39Bis : Même question sans utiliser de sous-requêtes

#	r39Bis
tables ouvertes	Villes, clients, Villes AS Villes_1, clients AS clients_1
jointures	J6 J_6 Villes.Ville = Villes_1.Ville
champs affiches	clients_1.Nom
criteres	clients.Nom = “Peuleurido” clients_1.Nom <> “Peuleurido”
regroupe par	
trie par	

r40 : Affichez les nom et prénom des clients qui ont commandé le même jour qu'Alavie en utilisant une sous-requête

r40a : Date des commandes d'alavie

#	r40a
tables ouvertes	clients, commande
jointures	J5
champs affiches	commande.Date
criteres	clients.Nom = “Alavie”
regroupe par	
trie par	

r40 : Requête finale

#	r40
tables ouvertes	clients, commande, r40a
jointures	J5 commande.Date = r40a.Date
champs affiches	Valeurs distinctes clients.Nom
criteres	clients.Nom <> “Alavie”
regroupe par	
trie par	

r40Bis : Même question sans utiliser de sous-requêtes

#	r40Bis
tables ouvertes	clients, commande, commande AS commande_1, clients AS clients_1
jointures	J5 commande.Date = commande_1.Date J_5
champs affiches	Valeurs distinctes clients_1.Nom
criteres	clients.Nom = "Alavie" clients_1.Nom <> "Alavie"
regroupe par	
trie par	

r41 : Affichez les nom et prénom des clients qui ont commandé le même MOIS qu'Alavie en utilisant une sous-requête

r41a : Mois des commandes d'alavie

#	r41a
tables ouvertes	clients, commande
jointures	J5
champs affiches	moisAlavie : mois(commande.Date)
criteres	clients.Nom = "Alavie"
regroupe par	
trie par	

r41 : Requête finale

#	r41
tables ouvertes	r41a, clients, commande
jointures	J5
champs affiches	Valeurs distinctes clients.Nom
criteres	clients.Nom <> "Alavie" mois(commande.Date)=r41a.moisAlavie
regroupe par	
trie par	

r41Bis : Même question sans utiliser de sous-requêtes

#	r41Bis
tables ouvertes	clients, commande, commande AS commande_1
jointures	J5, clients AS clients_1 clients_1.numclient = commande_1.refclient
champs affiches	Valeurs distinctes clients_1.Nom
criteres	clients.Nom = "Alavie" clients_1.Nom <> "Alavie" mois(commande.Date)=mois(commande_1.Date)
regroupe par	
trie par	

Correction TD 5

r39 : Affichez les nom et prénom des clients qui habitent la même ville que Peuleurido en utilisant une sous-requête

r39a : Ville de Peuleurido

#	r39a
tables ouvertes	Villes, clients
jointures	J6
champs affiches	Villes.Ville
criteres	clients.Nom = “Peuleurido”
regroupe par	
trie par	

r39 : Requête finale

#	r39
tables ouvertes	Villes, clients, r39a
jointures	J6 Villes.Ville = r39a.Ville
champs affiches	clients.Nom
criteres	clients.Nom <> “Peuleurido”
regroupe par	
trie par	

r39Bis : Même question sans utiliser de sous-requêtes

#	r39Bis
tables ouvertes	Villes, clients, Villes AS Villes_1, clients AS clients_1
jointures	J6 J_6 Villes.Ville = Villes_1.Ville
champs affiches	clients_1.Nom
criteres	clients.Nom = "Peuleurido" clients_1.Nom <> "Peuleurido"
regroupe par	
trie par	

r40 : Affichez les nom et prénom des clients qui ont commandé le même jour qu'Alavie en utilisant une sous-requête

r40a : Date des commandes d'alavie

#	r40a
tables ouvertes	clients, commande
jointures	J5
champs affiches	commande.Date
criteres	clients.Nom = "Alavie"
regroupe par	
trie par	

r40 : Requête finale

#	r40
tables ouvertes	clients, commande, r40a
jointures	J5 commande.Date = r40a.Date
champs affiches	Valeurs distinctes clients.Nom
criteres	clients.Nom <> "Alavie"
regroupe par	
trie par	

r40Bis : Même question sans utiliser de sous-requêtes

#	r40Bis
tables ouvertes	clients, commande, commande AS commande_1, clients AS clients_1
jointures	J5 commande.Date = commande_1.Date J_5
champs affiches	Valeurs distinctes clients_1.Nom
criteres	clients.Nom = "Alavie" clients_1.Nom <> "Alavie"
regroupe par	
trie par	

r41 : Affichez les nom et prénom des clients qui ont commandé le même MOIS qu'Alavie en utilisant une sous-requête

r41a : Mois des commandes d'alavie

#	r41a
tables ouvertes	clients, commande
jointures	J5
champs affiches	moisAlavie : mois(commande.Date)
criteres	clients.Nom = "Alavie"
regroupe par	
trie par	

r41 : Requête finale

#	r41
tables ouvertes	r41a, clients, commande
jointures	J5
champs affiches	Valeurs distinctes clients.Nom
criteres	clients.Nom <> "Alavie" mois(commande.Date)=r41a.moisAlavie
regroupe par	
trie par	

r41Bis : Même question sans utiliser de sous-requêtes

#	r41Bis
tables ouvertes	clients, commande, commande AS commande_1
jointures	J5, clients AS clients_1 clients_1.numclient = commande_1.refclient
champs affiches	Valeurs distinctes clients_1.Nom
criteres	clients.Nom = "Alavie" clients_1.Nom <> "Alavie" mois(commande.Date)=mois(commande_1.Date)
regroupe par	
trie par	

19.1 Jointures gauches

r43 : Affichez le nombre de produits par marques dans l'ordre croissant des marques

#	r43
tables ouvertes	marques , produit
jointures	J2s (J gauche)
champs affiches	marques.nom nbProduits : Compte(produit.numproduit)
criteres	
regroupe par	marques.nom
trie par	

r44 : 586 clients sont enregistrés dans la base de données, 109 ont effectés une commande. Affichez par ordre alphabétique les noms des clients qui n'ont rien commandé.

#	r44
tables ouvertes	clients , commande
jointures	J5 (J gauche)
champs affiches	clients.Nom clients.Prenom
criteres	commande.numcommande Is Null
regroupe par	
trie par	clients.Nom

r45 : Retour sur la requête 37 : "Affichez le nom et prénom des clients qui n'ont pas acheté de produits de moins de 1000 euros." Recommencez en incluant les noms et prénoms des clients n'ayant rien commandé et en les affichant par ordre alphabétique.

r45a : Par client le prix du produit le moins cher

#	r45a
tables ouvertes	produit, commande, procom
jointures	J4 J3
champs affiches	commande.refclient minPrix : Min(produit.Prix)
criteres	
regroupe par	commande.refclient
trie par	

r45 : Requête finale

#	r45
tables ouvertes	clients , r45a
jointures	clients.numclient = r45a.refclient (J gauche)
champs affiches	clients.Nom
criteres	(r45a.minPrix Is Null) OU (r45a.minPrix>1000)
regroupe par	
trie par	

r47 : Affichez les noms des clients qui ont effectués en août une commande de montant supérieur à celle qui a été la plus grande en janvier

r47a : Montant des commandes de janvier

#	r47a
tables ouvertes	produit, commande, procom
jointures	J4 J3
champs affiches	Montant : Somme(procom.quantité*produit.Prix)
criteres	commande.Date = “/01/“
regroupe par	commande.Date commande.numcommande
trie par	

r47b : Le montant le plus grand des commandes de janvier

#	r47b
tables ouvertes	r47a
jointures	
champs affiches	maxJanvier : Max(r47a.Montant)
criteres	
regroupe par	
trie par	

r47c : Par client montant des commandes d'août

#	r47c
tables ouvertes	produit, clients, commande, procom
jointures	J5 J4 J3
champs affiches	clients.Nom Montant : Somme(procom.quantité*produit.Prix)
criteres	commande.Date = "/08/"
regroupe par	clients.Nom
trie par	

r47 : Requête finale

#	r47
tables ouvertes	r47c, r47b
jointures	
champs affiches	Valeurs distinctes r47c.Nom
criteres	r47c.Montant>r47b.maxJanvier
regroupe par	
trie par	

r48 : Nom du client qui a acheté le produit le plus cher en janvier

r48a : Produits le plus cher

#	r48a
tables ouvertes	produit
jointures	
champs affiches	maxPrix : Max(produit.Prix)
criteres	
regroupe par	
trie par	

r48 : Requête finale

#	r48
tables ouvertes	produit, clients, commande, procom, r48a
jointures	J5 J4 J3 produit.Prix = r48a.maxPrix
champs affiches	Valeurs distinctes clients.Prénom clients.Nom
criteres	commande.Date = "/01/"
regroupe par	
trie par	

r49 : Quels sont les clients qui ont acheté le produit le plus cher acheté par Pige ? (3 réponses)

r49a : Produits acheté par Pige

#	r49a
tables ouvertes	produit, clients, commande, procom
jointures	J5 J4 J3
champs affiches	produit.numproduit produit.Prix
criteres	clients.Nom = "Pige"
regroupe par	
trie par	

r49b : Le plus cher

#	r49b
tables ouvertes	r49a
jointures	
champs affiches	maxPige : Max(r49a.Prix)
criteres	
regroupe par	
trie par	

Requête finale

#	r49
tables ouvertes	r49b, produit, clients, commande, procom, r49a
jointures	J5 J4 J3 produit.numproduit = r49a.numproduit r49b.maxPige = r49a.Prix
champs affiches	Valeurs distinctes clients.Nom clients.Prénom
criteres	clients.Nom <> "Pige"
regroupe par	
trie par	

r50 : Nombre de commandes par clients (s'ils n'ont pas fait de commande on doit indiquer 0

#	r50
tables ouvertes	clients , commande
jointures	J5 (J gauche)
champs affiches	clients.Nom clients.Prénom nbCommandes : Compte(commande.numcommande)
criteres	
regroupe par	clients.Nom clients.Prénom
trie par	

Correction Annales Access : Gowalla

r1 : Afficher les noms des spots dans l'ordre alphabétique.

#	r1
tables ouvertes	spots
jointures	
champs affiches	spots.nom
criteres	
regroupe par	
trie par	spots.nom

r2 : Afficher les noms des personnes habitant Lille dans l'ordre alphabétique.

#	r2
tables ouvertes	villes, personnes
jointures	villes.idville = personnes.idville
champs affiches	personnes.nom
criteres	villes.nomville = "Lille"
regroupe par	
trie par	personnes.nom

r3 : Afficher le nom de la catégorie dont le code est saisi au clavier, on choisira 12.

#	r3
tables ouvertes	categories
jointures	
champs affiches	categories.intitule categories.idcategorie
criteres	categories.idcategorie = quel code ?
regroupe par	
trie par	

r4 : Afficher les noms des spots de catégorie Librairie situés à Lille dans l'ordre alphabétique.

#	r4
tables ouvertes	categories, villes, spots
jointures	villes.idville = spots.refville categories.idcategorie = spots.refcategorie
champs affiches	spots.nom
criteres	villes.nomville = "Lille" categories.intitule = "Librairie"
regroupe par	
trie par	spots.nom

r5 : Combien y a-t-il de spots par type (intitulé). Dans cette question on n'affichera pas la valeur 0.

#	r5
tables ouvertes	types, spots
jointures	types.idtype = spots.reftype
champs affiches	types.intitule CompteDeidspot : Compte(spots.idspot)
criteres	
regroupe par	types.intitule
trie par	

r6 : Chaque check-in sur un spot commercial de catégorie Librairie est rémunéré 0,1€, quelle est la recette pour ce type de spot ? (check-in uniquement)

#	r6
tables ouvertes	types, categories, spots, checkins
jointures	spots.idspot = checkins.idspot categories.idcategorie = spots.refcategorie types.idtype = spots.reftype types.intitule = "commercial"
champs affiches	Expr1 : Somme(0.1)
criteres	categories.intitule = "Librairie"
regroupe par	categories.intitule types.intitule
trie par	

r7 : Quelle est la moyenne en nombre de jours entre la date de création du spot et ses check-in ?

#	r7
tables ouvertes	spots, checkins
jointures	spots.idspot = checkins.idspot
champs affiches	spots.nom Expr1 : Moyenne(checkins.date-spots.datecreation)
criteres	
regroupe par	spots.nom
trie par	

r8 : Afficher, dans l'ordre alphabétique, les noms des personnes qui ont fait un check-in

au même spots que ceux de Ness. (sans sous-requête)

#	r8
tables ouvertes	personnes, checkins, checkins AS checkins_1, personnes AS personnes_1
jointures	personnes.idpersonne = checkins.idpersonne checkins_1.idpersonne = personnes_1.idpersonne checkins.idspot = checkins_1.idspot
champs affiches	Valeurs distinctes personnes_1.nom
criteres	personnes.nom = "Ness" personnes_1.nom <> "Ness"
regroupe par	
trie par	personnes_1.nom

r9 : Quelle est la ville où il y a le plus grand nombre de spots ?

gowalla.r9a

#	r9a
tables ouvertes	villes, spots
jointures	villes.idville = spots.refville
champs affiches	villes.nomville CompteDeidspot : Compte(spots.idspot)
criteres	
regroupe par	villes.nomville
trie par	

r9b

#	r9b
tables ouvertes	r9a
jointures	
champs affiches	MaxDeCompteDeidspot : Max(r9a.CompteDeidspot)
criteres	
regroupe par	
trie par	

Requête finale

#	r9
tables ouvertes	r9a, r9b
jointures	r9a.CompteDeidspot = r9b.MaxDeCompteDeidspot
champs affiches	r9a.nomville
criteres	
regroupe par	
trie par	

r10 : Le prix des abonnements des spots de type vip augmente de 10%, calculer le montant total des abonnements par ville.

#	r10
tables ouvertes	types, villes, spots
jointures	villes.idville = spots.refville types.idtype = spots.reftype
champs affiches	villes.nomville Expr1 : Somme(vraifaux (types.intitule = "vip",1.1,1)*types.prix)
criteres	
regroupe par	villes.nomville
trie par	

r11 : Quel est le nombre de spots sans check-in ?

#	r11
tables ouvertes	spots , checkins
jointures	spots.idspot = checkins.idspot (J gauche)
champs affiches	CompteDeidspot : Compte(spots.idspot)
criteres	checkins.idpersonneIsNull
regroupe par	checkins.idpersonne
trie par	

Correction Annales Access : Agence

r1 : Quelles sont les villes de la base de données situées en France dans l'ordre alphabétique ?

#	r1
tables ouvertes	pays, villes
jointures	pays.numPays = villes.refPays
champs affiches	villes.nomVille
criteres	pays.nomPays = "France"
regroupe par	
trie par	villes.nomVille

r2 : Quels sont les sites situés en Italie ?

#	r2
tables ouvertes	pays, villes, sites
jointures	pays.numPays = villes.refPays villes.numVille = sites.refVille
champs affiches	sites.nomSite
criteres	pays.nomPays = "Italie"
regroupe par	
trie par	

r3 : Quels sont les villes visités en France ? (avec un site)

#	r3
tables ouvertes	pays, villes, sites
jointures	pays.numPays = villes.refPays villes.numVille = sites.refVille
champs affiches	Valeurs distinctes villes.nomVille
criteres	pays.nomPays = "France"
regroupe par	
trie par	

r4 : Quels sont les numéros des circuits programmés au 1^o semestre 2012 dont le prix de base (sans tenir compte du prix de visite des sites) est compris entre 2000 € et 5000 € ?

#	r4
tables ouvertes	circuits, programmation
jointures	circuits.numCircuit = programmation.refCircuit
champs affiches	Valeurs distinctes circuits.numCircuit
criteres	circuits.prix > 2000 ET circuits.prix < 5000 mois(date) < 4
regroupe par	
trie par	

r5 : Quel est le nombre d'étapes par numéro de circuit ?

#	r5
tables ouvertes	etapes
jointures	
champs affiches	etapes.refCircuit CompteDenumEtape : Compte(etapes.numEtape)
criteres	
regroupe par	etapes.refCircuit
trie par	

r6 : Quel est le prix à payer par « Talonion » par circuit (sans tenir compte du prix des sites) ?

#	r6
tables ouvertes	circuits, programmation, clients, reservations
jointures	circuits.numCircuit = programmation.refCircuit clients.numclient = reservations.refClient programmation.date = reservations.date programmation.refCircuit = reservations.refCircuit
champs affiches	programmation.refCircuit Expr1 : Somme(reservations.places*circuits.prix)
criteres	clients.Nom = "Talonion"
regroupe par	programmation.refCircuit
trie par	

r7 : Les circuits du mois de novembre subissent une baisse de 20%. Quel est le prix à payer par Duman par circuit et par date ?

#	r7
tables ouvertes	circuits, programmation, clients, reservations
jointures	circuits.numCircuit = programmation.refCircuit clients.numclient = reservations.refClient programmation.date = reservations.date programmation.refCircuit = reservations.refCircuit
champs affiches	circuits.numCircuit programmation.Date Expr1 : vraifaux (mois(programmation.date)=11,0,8,1)*circuits.prix*reservations.prix
criteres	clients.Nom = "Duman"
regroupe par	
trie par	

r8 : Quels sont les clients qui rencontreront Duman au cours d'un circuit ? (on ne fera qu'une seule requête)

#	r8
tables ouvertes	clients, reservations, clients AS clients_1, reservations AS reservations_1
jointures	clients_1.numclient = reservations_1.refClient reservations.refCircuit = reservations_1.refCircuit clients.numclient = reservations.refClient
champs affiches	Valeurs distinctes clients_1.Nom
criteres	clients_1.Nom <> "Duman" clients.Nom = "Duman"
regroupe par	
trie par	

r9 : Le prix total de chaque circuit comprend le prix de base et le prix de visite de chaque site. Calculer le prix total de chaque circuit pour les circuits qui ont des visites.

Somme des prix des visites de site par circuit

#	r9a
tables ouvertes	etapes, sites
jointures	etapes.refVille = sites.refVille
champs affiches	etapes.refCircuit SommeDeprix : Somme(sites.prix)
criteres	
regroupe par	etapes.refCircuit
trie par	

Requête finale

#	r9
tables ouvertes	circuits, r9a
jointures	circuits.numCircuit = r9a.refCircuit
champs affiches	circuits.numCircuit Expr1 : circuits.prix+r9a.SommeDeprix
criteres	
regroupe par	
trie par	

La question 10 est modifiée pour qu'elle soit plus simple : on cherche les villes et PAS les villes à visiter

r10 : Quels sont les numéros de circuit qui passent par toutes les villes d'un pays.

Nombre de ville par ref de pays

#	r10a
tables ouvertes	villes
jointures	
champs affiches	villes.refPays CompteDenumVille : Compte(villes.numVille)
criteres	
regroupe par	villes.refPays
trie par	

Nombre de villes par circuit et par pays

#	r10b
tables ouvertes	villes, etapes
jointures	villes.numVille = etapes.refVille
champs affiches	etapes.refCircuit villes.refPays CompteDenumVille : Compte(villes.numVille)
criteres	
regroupe par	etapes.refCircuit villes.refPays
trie par	

Requête finale

#	r10
tables ouvertes	r10b, r10a
jointures	r10b.CompteDenumVille = r10a.CompteDenumVille r10b.refPays = r10a.refPays
champs affiches	r10b.refCircuit
criteres	
regroupe par	
trie par	

r11 : Quels sont les numéros des circuits qui font des étapes sans site

#	r11
tables ouvertes	etapes , sites
jointures	etapes.refVille = sites.refVille (J gauche)
champs affiches	Valeurs distinctes etapes.refCircuit
criteres	sites.nomSite Is Null
regroupe par	
trie par	

Correction TD Select

r1 : Affichez les noms des clients

```
select nom  
from clients;
```

r2 : Affichez les codes postaux des clients

```
select codepostal  
from clients;
```

r3 : Affichez les codes postaux des clients une seule fois

```
select distinct codepostal  
from clients;
```

r4 : Afficher les noms et le prix HT des produits (tva de 20%)

```
select nom, prix / 1.2  
from produits;
```

r5 : Afficher les noms et le prix HT des produits

```
select nom, prix / 1.2 as HT  
from produits;
```

r6 : Recherchez les clients dont le code postal est 59100

```
select nom, prenom  
from clients  
where codepostal = '59100';
```

r7 : Recherchez les clients dont le nom commence par Ta et qui habitent le Nord

```
select nom, prenom  
from clients  
where nom like 'Ta%'  
and codepostal like '59%';
```

r8 : Recherchez les clients dont le nom commence par Ta et qui habitent le Nord ou le Pas de Calais

```
select nom, prenom
from clients
where nom like 'Ta%'
  and (codepostal like '59%'
  or codepostal like '62%');
```

r9 : Recherchez les clients dont le nom commence par Ta et qui n'habitent ni le Nord ni le Pas de Calais

```
select nom, prenom
from clients
where nom like 'Ta%'
  and codepostal not like '59%'
  and codepostal not like '62%';
```

r10 : Affichez les noms des clients dans l'ordre alphabétique des noms

```
select nom
from clients
order by nom;
```

r11 : Affichez les numéros de tous les produits dans l'ordre des numproduit

```
select numproduit
from produits
order by numproduit;
```

r12 : Affichez les numéros de marques et les noms des produits de la catégorie 3 dans l'ordre décroissant des numéros de marques et l'ordre alphabétique des noms

```
select refmarque, nom
from produits
where refcateg = 3
order by refmarque desc, nom;
```

r13 : Afficher les noms et le prix HT des produits dans l'ordre croissant des prix HT (tva de 20%)

```
select nom, prix / 1.2 as HT
from produits
order by HT;
```

r14 : Afficher les noms, le prix HT, le montant de la TVA et le prix TTC des produits dans l'ordre des noms

```
select nom, prix / 1.2 as HT, prix * 0.2/1.2 as TVA, prix_
  ↳as TTC
from produits
order by nom;
```

r15 : Afficher les départements des clients dans l'ordre croissant

```
select distinct substr(codepostal,1,2) as dpt
from clients
order by dpt;
```

r16 : Afficher les années des dates des commandes de la base

```
select distinct substr(date,1,4) as anneesCommandes
from commandes
order by anneesCommandes;
```

r17 : Les produits de la marque 3 subissent une hausse de 5%, ceux de la marque 4 une hausse de 10%, afficher les nouveaux prix des produits

```
select nom,
case
  when refmarque = 3 then prix * 1.05
  when refmarque = 4 then prix * 1.1
  else prix
end as nouveauPrix
from produits;
```

r18 : Affichez le nombre de produits dans la base.

```
select count(numproduit) as nb from produits;
```

r19 : Affichez le maximum, le minimum et la moyenne des prix des produits.

```
select max(prix) as maxPrix, min(prix) as minPrix,
avg(prix) as moyPrix
from produits;
```

r20 : Affichez le nombre d'articles vendus.

```
select sum(quantite) as nb
from procom;
```

r21 : Calculer le nombre de marques

```
select count(nummarque) as nb
from marques;
```

r22 : Calculer le nombre de marques associées à des produits

```
select count(distinct refmarque) as nb
from produits;
```

Correction TD Jointures

r1 : Pour chaque produit, afficher sa marque et son nom dans l'ordre des marques puis des produits

```
select marques.nom, produits.nom
from marques, produits
where nummarque = refmarque
order by marques.nom, produits.nom;
```

r2 : Affichez les noms des clients de Lille

```
select nom
from clients, villes
where clients.codepostal = villes.codepostal
  and ville = 'Lille';
```

r3 : Affichez les noms des clients de Lille et Roubaix

```
select nom
from clients, villes
where clients.codepostal = villes.codepostal
  and (ville = 'Lille' or ville = 'Roubaix');
```

r4 : Afficher le nom du client qui a effectué la commande 1

```
select nom
from clients, commandes
where numclient = refclient
  and numcommande = 1;
```

r5 : Affichez les noms des produits qui ont été achetés en quantité > 2 pour au moins une commande

```
select distinct nom
from produits, procom
where numproduit = refproduit
  and quantite > 2;
```

r6 : Pour chaque produit commandé dans la commande 1 affichez son nom et le prix à payer en fonction de la quantité

```
select nom, quantite * prix as totalLigne
from procom, produits
where numproduit = refproduit
  and refcommande = 1;
```

r7 : Afficher le montant de la commande 1

```
select sum(quantite * prix) as totalCommande
from procom, produits
where numproduit = refproduit
  and refcommande = 1;
```

r9 : Affichez le nom et prénom des clients qui ont acheté de produits de plus de 1000 euros

```
select distinct clients.nom, clients.prenom
from clients, commandes, procom, produits
where numclient = refclient
  and numcommande = refcommande
  and numproduit = refproduit
  and prix > 1000;
```

r10 : Une réduction de 10% est accordée pour chaque produit commandé en quantité supérieure ou égale à 3. Pour chaque produit commandé dans la commande 1 affichez le nom du produit, le total sans réduction, le montant de la réduction et le montant avec réduction

```
select produits.nom, quantite*prix as sansReduc,
  case
    when quantite >= 3 then quantite*prix*0.1
    else 0
  end as Reduc,
  case
    when quantite >= 3 then quantite*prix*0.9
    else quantite*prix
  end as avecReduc
from procom, produits
where numproduit = refproduit
  and refcommande = 1;
```

r11 : Afficher le montant de la commande 1 en tenant compte de la réduction de la question précédente

```
select sum(quantite*prix *
  case
    when quantite >= 3 then 0.9
    else 1
  end) as montant
from procom, produits
```

```
where numproduit = refproduit
  and refcommande = 1;
```

r12 : Affichez dans l'ordre alphabétique le nom et prénom des clients qui ont acheté au moins un des produits acheté par Alavie

```
select distinct C12.nom, C12.prenom
from clients C11, commandes C1, procom P1, clients C12,
  commandes C2, procom P2
where C11.numclient = C1.refclient
  and C1.numcommande = P1.refcommande
  and C12.numclient = C2.refclient
  and C2.numcommande = P2.refcommande
  and C11.nom = 'Alavie'
  and P1.refproduit = P2.refproduit
  and C12.nom != 'Alavie'
order by C12.nom;
```

r13 : Affichez les nom et prénom des clients qui habitent la même ville que Peuleurido

```
select C2.nom, C2.prenom
from clients C1, villes V1, villes V2, clients C2
where C1.codepostal = V1.codepostal
  and V1.ville = V2.ville
  and C2.codepostal = V2.codepostal
  and C1.nom = 'Peuleurido'
  and C2.nom != 'Peuleurido'
order by C2.nom, C2.prenom;
```

r14 : Nom du client qui a acheté le produit le plus cher en janvier (de n'importe quelle année)

```
select clients.nom, clients.prenom
from clients, commandes, procom, produits,
  (select max(prix) as maxPrix from produits) srl
where numclient = refclient
  and numcommande = refcommande
  and numproduit = refproduit
  and prix = srl.maxPrix
  and date like '%/01/%';
```

r15 : Quels sont les clients qui ont acheté le produit le plus cher acheté par Pige

```
select nom
from procom, commandes, clients,
  (select numproduit, prix
   from produits, commandes, procom, clients
   where numproduit = refproduit
     and numcommande = refcommande
     and numclient = refclient
     and clients.nom = 'Pige') srl,
```

```

(select max(prix) as maxprix from (
    select prix
    from produits, commandes, procom, clients
    where numproduit = refproduit
    and numcommande = refcommande
    and numclient = refclient
    and clients.nom = 'Pige')) sr2
where numcommande = refcommande
and refclient = numclient
and sr1.numproduit = procom.refproduit
and sr1.prix = sr2.maxprix
and nom != 'Pige';

```

r17 : Affichez les nom et prénom des clients qui ont effectué des commandes entre le jour de la première commande d'Alavie et le jour de sa dernière commande

```

select distinct nom
from clients, commandes,
(select min(date) as minDate,
     max(date) as maxDate
  from clients, commandes
  where numclient = refclient
  and nom = 'Alavie') sr1
where numclient = refclient
and date >= minDate
and date <= maxDate
and nom != 'Alavie';

```

r18 : Affichez par ordre alphabétique les noms des clients qui n'ont rien commandé

```

select nom
from clients left join commandes on numclient = refclient
where numcommande is null
order by nom;

```

r20 : Affichez par ordre alphabétique les noms des clients qui ont acheté des produits mais pas TOUS les produits achetés par le client 1

```

select distinct nom
from clients,
(select refclient, refproduit
  from clients, procom, commandes
  where refcommande = numcommande
  and refclient = numclient) sr1
left join
(select refproduit
  from procom, commandes
  where refcommande = numcommande
  and refclient = 1) sr2
on sr1.refproduit = sr2.refproduit
where sr2.refproduit is null

```

```
and numclient = srl.refclient
and numclient != 1;
```

Correction TD Regroupements

r1 : Affichez le nombre de produits par catégorie

```
select categ.nom, count(numproduit) as nbProduits
from categ, produits
where numcateg = refcateg
group by categ.nom
order by categ.nom;
```

r2 : Affichez le nombre de produits vendus par catégorie

```
select categ.nom, sum(quantite) as nbProduits
from categ, produits, procom
where numcateg = refcateg
  and numproduit = refproduit
group by categ.nom
order by categ.nom;
```

r3 : Affichez le nombre de commandes par an

```
select substr(date, 1, 4) as annee, count(numcommande) as nbCommandes
from commandes
group by annee
order by annee;
```

r4 : Affichez le nombre de produits différents commandés par le client Alavie par commande

```
select numcommande, count(refproduit) as nbProduits
from clients, commandes, procom
where numclient = refclient
  and numcommande = refcommande
  and nom = 'Alavie'
group by numcommande;
```

r5 : Affichez le nombre d'articles commandés par le client Alavie par commande

```

select numcommande, sum(quantite) as nbArticles
from clients, commandes, procom
where numclient = refclient
  and numcommande = refcommande
  and nom = 'Alavie'
group by numcommande;

```

r6 : Affichez le montant et la date des commandes effectuées par le client Tabille

```

select date, sum(quantite * prix) as montant
from clients, commandes, procom, produits
where numclient = refclient
  and numcommande = refcommande
  and numproduit = refproduit
  and clients.nom = 'Tabille'
group by date, numcommande;

```

r7 : Affichez le nom et le prénom des clients qui ont acheté des produits de plus de 3 marques

```

select clients.nom, clients.prenom
from clients, commandes, procom, produits
where numclient = refclient
  and numcommande = refcommande
  and numproduit = refproduit
group by clients.nom, clients.prenom -- en toute rigueur
  ↪numclient
having count(distinct refmarque) > 3;

```

r8 : Affichez le nom et prénom des clients dont le total des commandes est de plus de 5000 €

```

select clients.nom, clients.prenom
from clients, commandes, procom, produits
where numclient = refclient
  and numcommande = refcommande
  and numproduit = refproduit
group by clients.nom, clients.prenom
having sum(quantite * prix) > 5000;

```

r9 : Affichez le nom et prénom des clients qui ont effectué au moins une commande de plus de 5000 €

```

select distinct clients.nom, clients.prenom
from clients, commandes, procom, produits
where numclient = refclient
  and numcommande = refcommande
  and numproduit = refproduit
group by numcommande, clients.nom, clients.prenom
having sum(quantite * prix) > 5000;

```

r10 : Affichez le montant de la plus grande commande

```
select max(montant) as maxMontant
from
  (select sum(quantite * prix) as montant
  from procom, produits
  where numproduit = refproduit
  group by refcommande);
```

r11 : Affichez les noms et prénoms des clients qui ont achetés plus que Tabille (au total)

```
-- Avec un having

select clients.nom, clients.prenom
from clients, commandes, procom, produits
where numclient = refclient
  and numcommande = refcommande
  and numproduit = refproduit
group by clients.nom, clients.prenom
having sum(quantite*prix) >
  (select sum(quantite*prix)
  from clients, commandes, procom, produits
  where numclient = refclient
    and numcommande = refcommande
    and numproduit = refproduit
    and clients.nom = 'Tabille');

-- Avec deux sous-requetes

select nom, prenom
from
  clients,
  (select refclient, sum(quantite*prix) as montant
  from commandes, procom, produits
  where numcommande = refcommande
    and numproduit = refproduit
  group by refclient) sr1,
  --
  (select sum(quantite*prix) as montant
  from clients, commandes, procom, produits
  where numclient = refclient
    and numcommande = refcommande
    and numproduit = refproduit
    and clients.nom = 'Tabille') sr2

where sr1.montant > sr2.montant
  and clients.numclient = sr1.refclient;
```

r12 : Affichez le nom et prénom du client qui a effectué la plus grande commande

```
-- Avec un having

select distinct clients.nom, clients.prenom
from clients, commandes, procom, produits
where numclient = refclient
  and numcommande = refcommande
  and numproduit = refproduit
group by numcommande, clients.nom, clients.prenom
having sum(quantite * prix) =
  (select max(montant)
from
  (select sum(quantite * prix) as montant
from procom, produits
where numproduit = refproduit
group by refcommande));

-- Avec deux sous-requetes

select distinct clients.nom, clients.prenom
from
  clients,
  --
  (select refclient, sum(quantite * prix) as montant
from commandes, procom, produits
where numcommande = refcommande
  and numproduit = refproduit
group by numcommande, refclient) srl,
  --
  (select max(montant) as maxMontant
from
  (select sum(quantite * prix) as montant
from procom, produits
where numproduit = refproduit
group by refcommande)) sr2

where srl.montant = sr2.maxMontant
  and numclient = srl.refclient;
```

r13 : Affichez le total des montants des commandes par an

```
select substr(date,1,4) as annee, sum(quantite * prix) as montant
from commandes, procom, produits
where numcommande = refcommande
  and numproduit = refproduit
group by annee;
```

r14 : Affichez l'année pour laquelle le total des montants des commandes est le plus faible

```
select annee
from
```

```

(select substr(date,1,4) as annee, sum(quantite * prix) as montant
from commandes, procom, produits
where numcommande = refcommande
and numproduit = refproduit
group by annee) srl
where montant =
(select min(montant)
from
(select sum(quantite * prix) as montant
from commandes, procom, produits
where numcommande = refcommande
and numproduit = refproduit
group by substr(date,1,4)));

```

r15 : Affichez le montant des plus grandes commandes par client dans l'ordre décroissant des montants

```

select nom, max(montant) as maxMontant
from
(select clients.nom as nom, sum(quantite * prix) as montant
from clients, commandes, procom, produits
where numclient = refclient
and numcommande = refcommande
and numproduit = refproduit
group by clients.nom, numcommande) srl
group by nom
order by maxMontant desc;

```

r16 : Affichez les noms des clients qui ont effectués en août 1996 une commande de montant supérieur à celle qui a été la plus grande en janvier 1996

```

select clients.nom, clients.prenom
from clients, commandes, procom, produits
where numclient = refclient
and numcommande = refcommande
and numproduit = refproduit
and date like '1996/08/%'
group by numcommande, clients.nom, clients.prenom
having sum(quantite * prix) >=
(select max(montant)
from
(select sum(quantite * prix) as montant
from commandes, procom, produits
where numcommande = refcommande
and numproduit = refproduit
and date like '1996/01/%'
group by numcommande));

```

r17 : Affichez le nombre de produits par marque avec la valeur 0

```
select marques.nom, count(numproduit) as nbProduits
from marques left join produits on nummarque = refmarque
group by marques.nom
order by marques.nom;
```

Correction TD Révisions

r1 : Affichez le nom et le prénom du conférencier de code 1

```
select nom, prenom
from Conferenciers
where idConferencier = 1;
```

r2 : Affichez la liste des conférences de catégorie ‘Java Core’ et ‘Cloud’ dans l’ordre alphabétique.

```
select Conferences.intitule
from Conferences, Categories
where idCategorie = refCategorie
  and (Categories.intitule = 'Java Core'
    or Categories.intitule = 'Cloud')
order by Conferences.intitule;
```

r3 : Affichez le nom et prénom des conférenciers du nord ou du pas-de-calais dans l’ordre alphabétique.

```
select nom, prenom
from Conferenciers
where (codePostal like '59%'
  or codePostal like '62%')
order by nom;
```

r4 : Affichez l’intitule des catégories pour lesquelles il n’y a pas de conférences.

```
select Categories.intitule
from Categories left join Conferences on idCategorie = refCategorie
where refCategorie is null;
```

r5 : Affichez l’intitule et le nombre de participants des conférences dans l’ordre décroissant du nombre de participants puis dans l’ordre alphabétique des conférences.

```

select intitule, count(refParticipant) as NbParticipants
from Conferences, Inscriptions
where idConference = refConference
group by idConference, intitule
order by NbParticipants desc, intitule;

```

r6 : Affichez le nom des catégories qui ont au moins 2 conférences à 14h (pas forcément le même jour).

```

select Categories.Intitule
from Conferences, Categories
where idCategorie = refCategorie
  and heureDebut = 14
group by refCategorie
having count(idConference) >= 2;

```

r7 : Avec une AUTOJOINTURE, affichez le nom des participants qui assistent aux mêmes conférences qu'Alavie dans l'ordre alphabétique.

```

select distinct p2.nom
from Participants p1, Inscriptions i1, Participants p2, ↴
  ↴Inscriptions i2
where p1.idParticipant = i1.refParticipant
  and i1.refConference = i2.refConference
  and p2.idParticipant = i2.refParticipant
  and p1.nom = 'Alavie'
  and p2.nom != 'Alavie'
order by p2.nom;

```

r8 : Avec une SOUS REQUETE, affichez l'intitulé des conférences qui ont lieu dans la même salle que 'MongoDB'.

```

select intitule
from Conferences,
  (select refSalle
   from Conferences
   where intitule = 'MongoDB') srl
where Conferences.refSalle = srl.refSalle
and intitule != 'MongoDB';

```

r9 : Affichez l'intitulé et le nombre de participants de la conférences qui a le plus de participants.

```

select intitule, NbParticipants
from Conferences,
  (select refConference, count(refParticipant) as ↴
   ↴NbParticipants
   from Inscriptions
   group by refConference) srl,

```

```

(select max(NbParticipants) as MaxNb
from (select count(refParticipant) as NbParticipants
      from Inscriptions
      group by refConference) ) sr2

where NbParticipants = MaxNb
  and refConference = idConference;

```

r10 : Affichez le nombre de conférences par salle dans l'ordre alphabétique des salles avec la valeur 0 s'il n'y a pas de conférence dans une salle.

```

select salle, count(idConference) as nbConf
from Salles left join Conferences on idSalle = refSalle
group by salle
order by salle;

```

r11 : Les participants vip ont un pass qui donne lieu a une reduction de 30 % sur toutes les conférences, affichez le montant sans réduction (brut) et le montant avec réduction (net) à régler par participant.

```

select nom, sum(prix) as brut,
       sum(
         (case
           when pass then 0.7*prix
           else prix
         end)) as net
from Participants, Inscriptions, Conferences
where idParticipant = refParticipant
  and idConference = refConference
group by nom;

```

r1b : Affichez l'intitulé de la conférence de code 1

```

select intitule
from Conferences
where idConference = 1;

```

r2b : Affichez la liste des conferences qui ont lieu dans les salles 'Room5' et 'Room9' dans l'ordre alphabétique

```

select intitule
from Conferences, Salles
where idSalle = refSalle
  and (salle = 'Room5' or salle = 'Room9')
order by intitule;

```

r3b : Affichez le nom et prenom des participants du nord ou du pas-de-calais dans l'ordre alphabétique.

```
select nom, prenom
from Participants
where (codePostal like '59%'
      or codePostal like '62%')
order by nom;
```

r4b : Affichez les salles pour lesquelles il n'y a pas de conférences.

```
select salle
from Salles left join Conferences on idSalle = refSalle
where refSalle is null;
```

r5b : Affichez le nom, le prénom des participants et leur nombre d'inscriptions aux conférences ds l'ordre decroissant du nombre d'inscriptions puis dans l'ordre alphabétique des participants.

```
select nom, prenom, count(refConference) as NbConferences
from Participants, Inscriptions
where idParticipant = refParticipant
group by idParticipant, nom, prenom
order by NbConferences desc, nom;
```

r6b : Affichez le nom des participants qui se sont inscrits à au moins 2 conférences le 15/11/2010.

```
select nom
from Participants, Inscriptions, Conferences
where idParticipant = refParticipant
  and refConference = idConference
  and jour = '2010-11-15'
group by idParticipant
having count(idConference) >= 2;
```

r7b : Avec une AUTOJOINTURE, affichez le nom des conférences qui ont lieu dans la même salle que 'MongoDB'.

```
select c2.intitule
from Conferences c1, Conferences c2
where c1.refSalle = c2.refSalle
  and c1.intitule = 'MongoDB'
  and c2.intitule != 'MongoDB'
order by c2.intitule;
```

r8b : Avec une SOUS REQUETE, affichez le nom des participants qui assistent aux memes conférences qu'Alavie dans l'ordre alphabétique.

```
select nom
from Participants, Inscriptions,
  (select refConference
   from Participants, Inscriptions
   where idParticipant = refParticipant
```

```

    and nom = 'Alavie') srl
where idParticipant = refParticipant
    and Inscriptions.refConference = srl.refConference
    and nom != 'Alavie'
order by nom;

```

r9b : Affichez l'intitule et le nombre de conférences de la catégorie qui a le plus de conférence.

```

select intitule, NBConferences
from Categories,

    (select refCategorie, count(idConference) as NBConferences
from Conferences
group by refCategorie) srl,

    (select max(NBConferences) as MaxNb
from (select count(idConference) as NBConferences
from Conferences
group by refCategorie) sr2

where NBConferences = MaxNb
    and refCategorie = idCategorie;

```

r10b : Affichez le nombre de conférences par catégorie dans l'ordre alphabétique des catégories avec la valeur 0 s'il n'y a pas de conférence dans une catégorie.

```

select Categories.intitule, count(idConference) as nbConfs
from Categories left join Conferences on idCategorie =_
    ↳refCategorie
group by Categories.intitule
order by Categories.intitule;

```

r11b : La conférence de code 2 voit son prix diminuer de 30%, affichez le montant avec réduction à régler par participant.

```

select nom, sum(prix) as brut,
    sum(
        (case
            when idConference = 2 then 0.7*prix
            else prix
            end) as net
from Participants, Inscriptions, Conferences
where idParticipant = refParticipant
    and idConference = refConference
group by nom;

```

Correction Cinema1

r1 : Afficher les noms des cinémas dans l'ordre alphabétique

```
select cinema.nomcine
from cinema
order by cinema.nomcine;
```

r2 : Titre, année de sortie et réalisateur des films français

```
select film.titre, film.annee, film.realisateur
from film
where film.pays = 'France';
```

r3 : Titres des films avec leur réalisateur, dans l'ordre alphabétique des titres de film

```
select film.titre, film.realisateur
from film
order by film.titre;
```

r4 : Nom des réalisateurs qui commencent par un T

```
select distinct film.realisateur
from film
where film.realisateur like 'T%';
```

r5 : Noms et numéros des acteurs du film “Cop”

```
select acteurs.nomacteur
from film, distribution, acteurs
where film.idfilm = distribution.reffilm and
distribution.refacteur = acteurs.idacteur and
film.titre = 'Cop';
```

r6 : Années de sortie des films tournés avec Gabin dans l'ordre des dates

```
select distinct film.annee
from film, distribution, acteurs
where film.idfilm = distribution.reffilm and
```

```

distribution.refacteur = acteurs.idacteur and
acteurs.nomacteur = 'Gabin'
order by film.annee;

```

r7 : Films à l'affiche de la quatrième semaine dans chaque salle du cinéma gaumont

```

SELECT programme.salle, film.titre
FROM programme, film, cinema
WHERE film.idfilm = programme.reffilm and
cinema.idcine = programme.refcine and
programme.semaine = 4 and
cinema.nomcine = 'gaumont';

```

r8 : Acteurs qui ont tourné avec Truffaut comme réalisateur

```

SELECT DISTINCT acteurs.idacteur, acteurs.nomacteur
FROM acteurs, film, distribution
WHERE distribution.reffilm = film.idfilm and
distribution.refacteur = acteurs.idacteur and
film.realisateur = 'Truffaut';

```

r9 : Adresses des cinémas où passe un film réalisé par Crichton

```

SELECT DISTINCT cinema.nomcine, cinema.rue
FROM cinema, film, programme
WHERE programme.reffilm = film.idfilm and
programme.refcine = cinema.idcine and
film.realisateur = 'Crichton';

```

r10 : Rues, salles et nombres de places des cinémas où sont joués des films sortis en 1984 et dans lesquels joue Delon

```

SELECT DISTINCT cinema.rue, cinesalles.salle, cinesalles.
↳ nbplaces
FROM cinesalles, cinema, programme, film, distribution, ↳
↳ acteurs
WHERE cinesalles.refcine = cinema.idcine and
cinesalles.refcine = programme.refcine and
cinesalles.salle = programme.salle and
programme.reffilm = film.idfilm and
programme.reffilm = distribution.reffilm and
acteurs.idacteur = distribution.refacteur and
acteurs.nomacteur = 'Delon' and
film.annee = 1984;

```

r11 : Titres et semaines de programmation des films dans lesquels jouent Deneuve ou Eastwood

```

SELECT DISTINCT film.titre, programme.semaine
FROM programme, distribution, film, acteurs
WHERE programme.reffilm = film.idfilm AND

```

```
distribution.reffilm = film.idfilm AND
acteurs.idacteur = distribution.refacteur AND
(acteurs.nomacteur = 'Eastwood' OR acteurs.nomacteur UN
DET = 'Deneuve')
ORDER BY film.titre;
```

r12 : Pour chaque réalisateur, nom des acteurs qui ont tourné sous sa direction en 1989, trié par ordre alphabétique de nom de réalisateur puis par ordre alphabétique inverse de nom d'acteur.

```
SELECT DISTINCT film.realisateur, acteurs.nomacteur
FROM film, distribution, acteurs
WHERE film.idfilm = distribution.reffilm AND
      acteurs.idacteur = distribution.refacteur AND
      film.annee = 1989
ORDER BY film.realisateur, acteurs.nomacteur DESC;
```

Correction Cinema2

r1 : nb de films par acteur

```
select acteurs.nomacteur, count(distribution.reffilm) as nbFilms
from acteurs, distribution
where acteurs.idacteur = distribution.refacteur
group by acteurs.nomacteur;
```

r2 : Afficher pour le réalisateur Mieville, le nombre de films, en nommant nbreFilms le titre de la colonne.

```
SELECT COUNT(idfilm) as nbreFilms
FROM film
WHERE realisateur = 'Mieville';
```

r3 : Afficher pour chaque réalisateur, le nombre de films, en commençant par celui qui a réalisé le plus de films.

```
SELECT realisateur, COUNT(idfilm) as nbFilms
FROM film
GROUP BY realisateur
ORDER BY nbFilms DESC;
```

r4 : Afficher pour chaque réalisateur, le nombre de films, pour les réalisateurs qui ont fait plus d'un film.

```
SELECT film.realisateur, COUNT(film.idfilm) as nbFilms
FROM film
GROUP BY film.realisateur
HAVING nbFilms > 1;
```

r5 : Nombre de semaines de programmation de chaque film.

```
SELECT titre, COUNT(DISTINCT semaine) as nbSemaines
FROM film, programme
WHERE reffilm = idfilm
GROUP BY titre;
```

r6 : Titre des films qui passaient la semaine 4, avec le nombre de cinémas dans lesquels ils étaient programmés, limités à ceux programmés dans au moins 4 cinémas, triés dans l'ordre du nombre de cinémas dans lesquels ils étaient programmés.

```
SELECT film.titre, COUNT(DISTINCT programme.refcine) as ↴nbCinemas
FROM programme, film
WHERE film.idfilm = programme.reffilm
      AND programme.semaine = 4
GROUP BY film.titre
      HAVING nbCinemas >= 4
ORDER BY nbCinemas desc;
```

r7 : Avec une sous-requête, afficher les films qui ont été tournés la même année que Nikita”

```
select film.titre
from film
where film.annee = (select film.annee from film where film.
      titre = 'Nikita')
and film.titre <> 'Nikita' ;
```

r8 : Box office (=les films dans l'ordre décroissant du nombre d'entrées) de la semaine 4 (titre, réalisateur, nombre d'entrées).

```
SELECT titre, realisateur, SUM(nbentrees) as nbTotalEntrees
FROM film, programme
WHERE semaine = 4
      AND reffilm = idfilm
GROUP BY titre, realisateur
ORDER BY nbTotalEntrees DESC;
```

r9 : Avec une auto jointure, afficher les films qui ont été tournés la même année que Nikita

```
select f2.titre
from film f1, film f2
where f1.annee = f2. annee
and f1.titre = 'Nikita'
and f2.titre <> 'Nikita' ;
```

r10 : Avec une aujointure Acteurs partenaires de Deneuve

```
SELECT DISTINCT a2.nomacteur
FROM acteurs a1, distribution d1, distribution d2, acteurs ↴a2
WHERE a1.idacteur = d1.refacteur and
      d1.reffilm = d2.reffilm and
      d2.refacteur = a2.idacteur and
      a1.nomacteur = 'Deneuve' and
      a2.nomacteur <> 'Deneuve' ;
```

r11 : Réaliseurs ayant joué dans des films qu'ils n'ont pas réalisés eux-même

```
select distinct(acteurs.nomacteur)
from film f1, film f2, distribution, acteurs
where f1.idfilm = distribution.reffilm
and distribution.refacteur = acteurs.idacteur
and acteurs.nomacteur = f2.realisateur
and f1.realisateur <> acteurs.nomacteur;
```

r12 : Réalisateur et noms des films précédés dans la programmation par un film dans lequel joue Pfeiffer

```
select acteurs.nomacteur, count(distinct d2.refacteur) -1 ↴
    ↵as nbPartenaires
from acteurs, distribution d1, distribution d2
where acteurs.idacteur = d1.refacteur
and d1.reffilm = d2.reffilm
group by acteurs.nomacteur
order by nbPartenaires desc;
```

r13 : Avec une sous requête Acteurs partenaires de Deneuve

```
select distinct nomacteur
from acteurs, distribution
where idacteur = refacteur
and reffilm in -- attention!!! in different de egal
    (select reffilm
        from acteurs, distribution
        where idacteur = refacteur
        and nomacteur = 'Deneuve')
and nomacteur <> 'Deneuve';
```

r14 : Liste des acteurs avec le nombre de leurs partenaires (c'est à dire le nombre des acteurs avec lesquels ils ont joué dans au moins un film), dans l'ordre décroissant du nombre de partenaires.

```
SELECT nomacteur, COUNT(DISTINCT d2.refacteur) - 1 ↴ as ↴
    ↵partenaires
FROM acteurs, distribution d1, distribution d2
WHERE idacteur = d1.refacteur AND
    d1.reffilm = d2.reffilm
GROUP BY idacteur, nomacteur
ORDER BY COUNT(DISTINCT d2.refacteur) DESC;
```

r15 : Titre, année et pays des films pour lesquels il existe un film réalisé la même année dans le même pays.

```
SELECT DISTINCT f1.titre, f1.annee, f1.pays
FROM film f1, film f2
WHERE f1.annee = f2.annee AND
    f1.pays = f2.pays AND
    f1.idfilm <> f2.idfilm
ORDER BY f1.annee, f1.pays;
```

r16 : Analyse de la concurrence : pour chaque paire de cinémas (C1, C2), nombre de films identiques programmés simultanément. La liste doit être triée par C1 puis par nombre de films simultanés (dans l'ordre décroissant), puis par C2.

```
SELECT c1.nomcine, COUNT(DISTINCT pl.refffilm) AS Communs,
       ↳c2.nomcine
  FROM programme pl, programme p2, cinema c1, cinema c2
 WHERE pl.refcine = c1.idcine AND
       p2.refcine = c2.idcine AND
       pl.semaine = p2.semaine AND
       pl.refffilm = p2.refffilm
GROUP BY c1.nomcine, c2.nomcine
ORDER BY c1.nomcine, communs DESC, c2.nomcine;
```

r17 : Noms des cinémas ayant programmé un même film dans des salles différentes.

```
SELECT DISTINCT idcine, nomcine
  FROM programme, cinema
 WHERE refcine = idcine
GROUP BY idcine, nomcine, refffilm
HAVING COUNT(DISTINCT salle) > 1;
```

r18 : Pour chaque pays, donnez le nombre moyen d'acteurs par film.

```
SELECT pays, COUNT(refacteur)/COUNT(DISTINCT idfilm) as x
  FROM distribution, film
 WHERE idfilm = refffilm
GROUP BY pays
ORDER BY x;
```

r19 : Pour chaque année, noms des cinémas ayant eu à l'affiche tous les films sortis cette année.

```
SELECT srl.annee, nomcine
  FROM (SELECT annee, refcine, COUNT(DISTINCT programme.
       ↳refffilm) AS c
        FROM programme, film
       WHERE refffilm = idfilm
GROUP BY annee, refcine) srl,
 (SELECT annee, COUNT(titre) AS c
        FROM film
GROUP BY annee) sr2,
 cinema
 WHERE srl.annee = sr2.annee AND
       srl.c = sr2.c AND
       srl.refcine = cinema.idcine
ORDER BY srl.annee;
```

Correction Cinema3

r1 : films qui ont au moins 6 acteurs

```
select reffilm , count(refacteur) from distribution
group by reffilm
having count(refacteur) >= 6;
```

r2 : nombre de films par decennie

```
select (annee-1900)/10*10 as decennie, count(idfilm) as ↴nbFilms
from film
group by decennie;
```

r3 : les acteurs sont payes 200000 euros et 0.5 euro par entree, calculer le salaire total des acteurs

```
select nomacteur, 200000 + sum(nbentrees*0.5) as salaire
from acteurs, distribution, programme
where idacteur = refacteur
and distribution.reffilm = programme.reffilm
group by nomacteur;
```

r4 : pourcentage de remplissage des salles du cinema gaumont la semaine 4

```
select programme.salle, (nbentrees*100) / (7* nbplaces) as ↴pourcentage
from programme, cinesalles, cinema
where programme.refcine = cinesalles.refcine
and programme.salle = cinesalles.salle
and programme.refcine = idcine
and semaine = 4
and nomcine = 'gaumont';
```

r5 : films qui n'ont pas ete programmes

```
select titre
from film left join programme on reffilm = idfilm
where reffilm is null;
```

r6 : cinema qui n'ont rien programme la semaine 4

```
select nomcine
from cinema left join
  (select refcine from programme where semaine = 4)
  on idcine = refcine
where refcine is null;
```

r7 : film qui a fait le plus grand nombre d'entrees

```
select titre from
film,
  (select reffilm, sum(nbentrees) as nbentrees
   from programme group by reffilm) srl ,
  (select max(nbentrees) as maxnbentrees from
   (select sum(nbentrees) as nbentrees
    from programme
    group by reffilm)) sr2
where nbentrees = maxnbentrees
  and reffilm = idfilm;
```

r8 : films dont le nombre d'entrees est superieur a la moyenne des entrees par semaine

```
select distinct titre
from programme, film,
  (select semaine, avg(nbentrees) as moyenne from
   (select semaine, sum(nbentrees) as nbentrees
    from programme
    group by semaine, reffilm )
   group by semaine) srl
where programme.semaine = srl.semaine
  and idfilm = reffilm
group by reffilm, programme.semaine
having sum(nbentrees) > moyenne      ;
```

r9 : films dont le nombre d'entrées est égal à +/- 5 % de la moyenne des entrées par semaine

```
select distinct titre
from programme, film,
  (select semaine, avg(nbentrees) as moyenne from
   (select semaine, sum(nbentrees) as nbentrees
    from programme
    group by semaine, reffilm )
   group by semaine) srl
where programme.semaine = srl.semaine
  and idfilm = reffilm
group by reffilm, programme.semaine
having sum(nbentrees) > moyenne * 0.95
  and sum(nbentrees) < moyenne * 1.05;
```

r10 : les réalisateurs sont payés 100000 euros avec une subvention de 50000 euros s'ils réalisent un film en France, calculer le salaire total des réalisateurs

```
select realisateur,
       sum( 100000 + (case when pays = 'France' then 50000
                           else 0 end)) as salaire
  from film
 group by realisateur;
```

r11 : nombre de films par mois et par cinéma, on notera Jamais si un cinéma ne programme aucun film

```
select nomcine,
       (case when semaine is null then 'Jamais'
             when semaine <= 4 then 'Janvier'
             else 'Février' end) as mois,
       count( distinct reffilm ) as nbFilms
  from cinema left join programme on refcine = idcine
 group by nomcine,mois ;
```

r12 : réalisateur qui a fait le + grand nombre de films

```
select srl.realisateur
  from (select count(idfilm) as nb,realisateur from film
        group by realisateur) srl,
       (select max(nb) as nbMax
         from (select count(idfilm) as nb from film group by
               realisateur) sr2
  where srl.nb = sr2.nbMax;
```

r13 : acteur le plus rentable (+ grand nombre d'entrées)

```
select nomacteur
  from acteurs,
       (select count(nbentrees) as nb,refacteur
         from distribution d,programme p
        where d.reffilm = p.reffilm
        group by refacteur) srl,
       (select max(nb) as nbMax
         from (select count(nbentrees) as nb
               from distribution d,programme p
              where d.reffilm = p.reffilm
              group by refacteur) sr2
  where srl.nb = sr2.nbMax and
        refacteur = idacteur;
```

r14 : réalisateurs qui ne sont pas acteurs ss jointure gauche

```
select distinct realisateur
  from film
 where realisateur not in (select nomacteur from acteurs);
```

r15 : réalisateurs qui ne sont pas acteurs avec jointure gauche

```
select distinct realisateur
from film left join acteurs on realisateur = nomacteur
where nomacteur is null;
```

r16 : par cinema liste des films qui ont un nb d'entrées inférieur à la moitié du nb de place
tt salles confondues la semaine 4

```
select nomcine, titre
from cinema, film,
(select p.refcine as refcine, reffilm, sum(nbplaces) ↴
 ↪as nbplaces
from cinesalles cs, programme p
where cs.refcine = p.refcine and semaine = 4
group by p.refcine, reffilm) srl,
(select refcine, reffilm, sum(nbentrees) as nbentrees
from programme
where semaine = 4
group by reffilm, refcine) sr2
where srl.refcine = sr2.refcine and
srl.reffilm = sr2.reffilm and
nbentrees >= nbplaces / 2 and
idcine = srl.refcine and
idfilm = srl.reffilm;
```

r17 : nb de films par première lettre (A :2) : substr(champ,debut,nb)

```
select initiale, count(initiale) as nb from
(select substr(titre,1,1) as initiale from film)
group by initiale;
```

r18 : réalisateur qui a tourné avec le plus grand nombre d'acteurs

```
select realisateur from
(select realisateur, count(distinct refacteur) as nb
from film,distribution
where idfilm = reffilm
group by realisateur) srl,
(select max(nb) as maxnb from
(select count(distinct refacteur) as nb
from film,distribution
where idfilm = reffilm
group by realisateur) sr2
where nb = maxnb;
```

r19 : les acteurs ont un intérêsement de 0.5 euro par entrée d'un film si le nombre d'entrées de ce film dépasse 10000, pour chaque acteur ayant obtenu des intérêssements afficher leur montant total

```

select nomacteur,
sum(nbentrees*(case
when distribution.reffilm in
(select reffilm from programme
group by reffilm having sum(nbentrees) > 10000)
then 0.5 else 0 end)) as interessement
from acteurs, distribution, programme
where idacteur = refacteur
and distribution.reffilm = programme.reffilm
group by nomacteur
having interessement > 0;

```

r20 : le tarif des entrées des cinémas est de 6€ sauf ‘le melies’ : 5€ et ‘le club’ : 3€. Quelle est la recette totale des cinemas

```

select nomcine,
sum(nbentrees)*(case
when nomcine = 'le melies' then 5
when nomcine = 'le club' then 3
else 8 end) as recette
from cinema, programme
where refcine = idcine
group by nomcine;

```

r21 : pour ‘le club’, par salle et par film, afficher ‘abandonner’ si le taux d’occupation de la semaine 4 est inférieur à 20% des places de la salle, ‘ouvrir’ s’il est supérieur à 80% sinon ‘continuer’

```

select p.salle, titre,
(case when (nbentrees*100)/(7* nbplaces) < 20 then
abandonner
when (nbentrees*100)/(7* nbplaces) > 80 then 'ouvrir'
else 'continuer' end) as action
from programme p, cinesalles cs, cinema,film
where p.refcine = cs.refcine
and p.salle = cs.salle
and cs.refcine = idcine
and idfilm = reffilm
and nomcine = 'le club'
and semaine = 4
group by p.salle, reffilm;

```

r22 : par cinema et par semaine, liste des salles sans programmation

```

select nomcine, cs.salle
from cinema,cinesalles cs left join
(select refcine, salle from programme where semaine = 4 srl
on cs.refcine = srl.refcine and cs.salle = srl.salle
where idcine = cs.refcine and
srl.refcine is null;

```

r23 : acteurs qui n'ont pas tourné de films au USA mais qui en ont tourné en Espagne

```
select distinct nomacteur
from acteurs, distribution, film
where idfilm = reffilm
  and idacteur = refacteur
  and pays = 'Espagne'
  and idacteur not in (
    select refacteur from distribution, film
    where idfilm = reffilm
    and pays = 'USA');
```