PREMIERE PARTIE : ALGEBRE RELATIONNELLE

= Projection = SELECT   
 = Selection = WHERE

Projection

a) N\_CI (CLIENTS)

*SELECT* N\_CI *FROM* CLIENTS;

Arite (a) = 1 *(Nombre de colonnes)*

Card (a) = 14 *(Nombre d’enregistrements)*

b) Nom (CATEGORIES)

*SELECT* nom *FROM* CATEGORIES;

Arite (b) = 1

Card (b) = 8

c) N\_CI (FOURNISSEURS)

Arite (c) = 0

Card (c) = 8

*Erreur car N\_cl n’appartient pas à Fournisseurs*

Sélection

d) VILLE = ‘Lille’ (CLIENTS)   
  
*SELECT* \* *FROM* CLIENTS *WHERE* VILLE = ‘Lille’;  
  
Arite (d) = Arite(CLIENTS) = 6

Card (d) card (CLIENTS)

e) VILLE = ‘Paris’ (Catégories)   
  
*Erreur car il y a une condition à la place d’une liste de colonne*

f) pays = ‘France’ ( N\_Fou (FOURNISSEURS))

*Erreur car la condition n’est pas exécutable dans la sélection*

g) N\_Fou ( pays = ‘France’ (FOURNISSEURS))

*SELECT* N\_FOU *FROM* FOURNISSEURS *WHERE* pays = ‘France’;

Arite (g) = 1

Card (g) = 2

Intersection = et

h) fonction (CLIENT) fonction (FOURNISSEUR)

Arite (h) = 1

Card (h) Card(A) + Card(B) = 3

i) fonction, société (CLIENT) fonction (FOURNISSEUR)

*Erreur de Syntaxe*

j)pays (CLIENT) pays (FOURNISSEUR)

Arite (j) = 1

Card (j) = 4

Union = ou

k) fonction (CLIENT) fonction (FOURNISSEUR)

Arite (k) = 1

Card (k) Card(A) + Card(B) = 8

*SELECT* fonction *FROM* client UNION *SELECT* fonction *FROM* fournisseur

l) fonction, société (CLIENT) fonction (FOURNISSEUR)

*Erreur de Syntaxe*

m)pays (CLIENT) pays (FOURNISSEUR)

Arite (m) = 1

Card (m) Card(A) + Card(B) = 7

*SELECT* pays *FROM* client UNION *SELECT* pays *FROM* fournisseur

Différence

o) fonction, société (CLIENT) - fonction (FOURNISSEUR)

*Erreur de Syntaxe*

p) pays (CLIENT) - pays (FOURNISSEUR)

Arite (p) = 1

Card (p) = Card(A) - Card(B) = 2 (Canada, Allemagne) (c’est client qu’on soustrait à fournisseur, si c’était l’inverse, on aurait eu 1 = Australie)

*SELECT* pays *FROM* client EXCEPT *SELECT* pays *FROM* fournisseur

q) pays (FOURNISSEUR) - pays (CLIENT)

Arite (q) = 1

Card (q) = Card(A) - Card(B) = 1 (Australie) (c’est fournisseur qu’on soustrait à client, si c’était l’inverse, on aurait eu 2 = Canada, Allemagne)

*SELECT* pays *FROM* fournisseur EXCEPT *SELECT* pays *FROM* client

Produit

r) nom (PRODUITS) X (CATEGORIES)

Arite (r) = 4 (nom (PRODUITS) = 1, (CATEGORIES) = 3)

Card (r) = Card(Produit) x Card(catégorie) = 120

Jointures

s) nom (PRODUITS) |\/| (CATEGORIES)

*Faux car il manque N\_Cat et la jointure*

t) nom, N\_Cat (PRODUITS) |\/| Produits.N\_Cat = Categorie.N\_Cat et prix > 30€ (CATEGORIES)

nom, N\_Cat, prix > 30 (PRODUITS) |\/| Produits.N\_Cat = Categorie.N\_Cat et prix > 30€ (CATEGORIES)

Arite (v) = 5 (nom, N\_cat (PRODUITS) = 2, (CATEGORIES) = 3)

Card (v) =

u) nom, N\_Cat (PRODUITS) |\/| N\_Cat (CATEGORIES)

v) nom (PRODUITS) |\/| N\_Cat (CATEGORIES)

Arite (v) = 4 (nom (PRODUITS) = 1, (CATEGORIES) = 3)

Card (v) = 15

DEUXIEME PARTIE : EXERCICE

SELECT

1) Afficher le contenu de la table Employe

*SELECT* \*

*FROM* employés ;

2) Donner les différents pays des fournisseurs

*SELECT DISTINCT* (pays)

*FROM* Fournisseur ;

pays(Fournisseur)

3) Donner les noms des produits qui coûtent plus que 30 €

*SELECT* nom

*FROM* produits

*WHERE* prix > 30 ;

4) Donner les noms des produits qui coûtent plus que 30 € en ordonnant le résultat par ordre décroissant des prix

*SELECT* nom

*FROM* produits

*WHERE* prix > 30

*ORDER BY* prix DESC ;

5) Donner les noms des produits de la catégorie 3 qui coûtent plus de 10€

*SELECT* nom

*FROM* produits

*WHERE* prix > 10

*AND* N\_Cat = 3 ;

pays( prix>10 et N\_Cat = 3(Fournisseur))

6) Donner les noms des employés qui habitent la France ou le Royaume-Uni

*SELECT* nom

*FROM* employés

*WHERE* pays = ‘France’

*OR* pays = ‘R-U’

nom( pays = ‘France’ ou pays = ‘R-U’(Employes))

7) Donner les deux produits les moins chers

*SELECT TOP* 2

*FROM* produit

*ORDER BY* prix ASC ;

8) Donner les deux produits les plus chers

*SELECT* TOP 2

*FROM* produit

*ORDER BY* prix DESC ;

9) Donner la liste des fournisseurs (societe, ville, pays) en ordonnant le résultat par pays, puis par ville puis par nom de fournisseur

*SELECT* societe, ville, pays

*FROM* fournisseur

*ORDER BY* pays, ville, societe

10) Donner les noms des clients dont le nom comporte un N en 3e position

*SELECT* nom

*FROM* clients

*WHERE* nom LIKE ‘\_\_N%’ ;

\_ = 1 seul caractère non vide

% = plusieurs caractère pouvant être vide

Union, Intersection, Différence

11) Donner la liste des pays où se trouve un client ou un fournisseur

*SELECT* pays

*FROM* client

*UNION*

*SELECT* pays

*FROM* fournisseur

pays(Client) pays(Fournisseur)

12) Donner la liste des pays où se trouve un client et un fournisseur

*SELECT* pays

*FROM* client

*INTERSECT*

*SELECT* pays

*FROM* fournisseur

pays(Client) pays(Fournisseur)

12bis) *SELECT* pays

*FROM* client

*WHER*E pays *IN* (*SELECT pays FROM fournisseur*)

13) Donner la liste des pays où se trouve un client mais pas de fournisseur

*SELECT* pays

*FROM* client

*EXCEPT* *(MySQL/SQL) ou MINUS (Oracle)*

*SELECT* pays

*FROM* fournisseur

pays(Client) - pays(Fournisseur)

13bis) *SELECT* pays

*FROM* client

*WHERE* pays *NOT IN* (*SELECT pays FROM fournisseur*)

JOINTURE

14) Donner pour chaque produit son nom et le nom de sa catégorie

*SELECT* P.nom as Nom, C.nom as Catégorie

*FROM* Produit P, Categorie C

*WHERE* P.N\_cat = C.N\_cat

nom, N\_Cat(Produits *JOINTURE* N\_Cat Categorie)

15) Donner pour chaque employé son nom et le numéro des commandes don’t il s’est occupés

*SELECT* E.nom as Employé, C.N\_com as Commande

*FROM* employe E, commande C

*WHERE* E.N\_Emp = C.N\_Emp

Nom, N\_com(employe *JOINTURE* N\_emp Commande)

15 = 18 (15 = From/Inner Join/On, 18 = From/Where)

16) Donner le nom des produits qui ont été déjà commandé avec une quantité > 20

*SELECT DISTINCT* P.Nom as Produit

*FROM* produit P, detailscommande DC

*WHERE* P.N\_pr = DC.N\_pr

*AND* Quantite > 20

nom( quantite>20 (Produits JOINTURE N\_Pr Detailscommande))

17) Donner le nom des clients qui ont commandé des produits provenant d’un pays étranger, ainsi que leur pays, les produits commandés et les pays de provenance de ces produits.

*SELECT* C.societe, C.pays, P.nom, F.pays

*FROM* Client C, Fournisseur F, Produit P, Commande Co, Detailscommande DC

*WHERE* C.N\_CL = Co.N\_CL

*AND* Co.N\_Com = DC.N\_Com

*AND* DC.N\_pr = P.N\_Pr

*AND* P.N\_fou = F.N\_fou

*AND* F.pays <> C.pays

societe, C.pays, nom, F.pays( F.pays C.pays (Client JOINTURE N\_Cl ( Commande JOINTURE N\_com (Detailscommande JOINTURE N\_pr (Produit JOINTURE N\_fou fournisseur)))))

Jointure usuelle (interne)

18) Donner pour chaque employé qui s’est occupé d’au moins une commande son nom et le numéro des commandes dont il s’est occupés

*SELECT* E.nom as Employé, C.N\_com as Commande

*FROM* employe E

*INNER JOIN* commande C

*ON* E.N\_Emp = C.N\_Emp

Nom, N\_com(employe *JOINTURE* N\_emp Commande)

18 = 15 (18 = From/Where, 15 = From/Inner Join/On)

19) Donner la liste des clients (noms) qui ont passé au moins une commande avec la liste des noms produits qu’ils ont commandés

Méthode 1 :

*SELECT C.societe, P.nom*

*FROM Client C*

*INNER JOIN Commande Co*

*INNER JOIN Detailscommande DC*

*INNER JOIN Produit P*

*ON DC.N\_pr = P.N\_pr*

*ON Co.N\_com = DC.N\_com*

*ON Co.N\_cl = C.N\_cl;*

Méthode 2 :

*SELECT* C.societe, P.nom

*FROM* Client C *INNER JOIN* Commande Co *ON* Co.N\_cl = C.N\_cl

*INNER JOIN* Detailscommande DC *ON* Co.N\_com = DC.N\_com

*INNER JOIN* Produit P *ON* DC.N\_pr = P.N\_pr;

Méthode 3 :

SELECT C.societe, P.nom

FROM Client C INNER JOIN (Commande Co INNER JOIN (Detailscommande DC INNER JOIN Produit P ON DC.N\_pr = P.N\_pr )ON Co.N\_com = DC.N\_com)ON Co.N\_cl = C.N\_cl

Societe, Nom(Clients *JOINTURE* N\_Cl (Commande JOINTURE N\_com (Detailscommande JOINTURE N\_Pr Produit)))

Jointure externes

20) Donner la liste des employés, avec, s’ils se sont chargés de commandes, les numerous de commandes don’t il se sont occupés

LEFT

*SELECT* E.Nom, Co.N\_com

*FROM* Employe E

*LEFT JOIN* Commande Co

*ON* E.N\_Emp = Co.N\_Emp

Nom, N\_com(Employe *JOINTURE* N\_emp Commande)

RIGHT

*SELECT* E.Nom, Co.N\_com

*FROM* Commande Co

*RIGHT JOIN* Employe E

*ON* E.N\_Emp = Co.N\_Emp

Nom, N\_com(Commande *JOINTURE* N\_emp Employe)

21) Donner la liste de tous les clients (qui ont déjà commandé ou non) avec la liste des noms de produits qu’ils ont commandé

RIGHT

SELECT C.N\_cl, P.nom

FROM Detailscommande DC

INNER JOIN Commande Co ON Co.N\_com = DC.N\_com

INNER JOIN Produit P ON P.N\_pr = DC.N\_pr

RIGHT JOIN Client C ON C.N\_CL = Co.N\_Cl;

LEFT

SELECT C.societe, P.Nom

FROM Client C

LEFT JOIN (Commande Co

INNER JOIN Detailscommande DC ON Co.N\_com = DC.N\_com

INNER JOIN Produit P ON DC.N\_pr = P.N\_pr)

ON C.N\_cl = Co.N\_cl;

Right : societe, nom(((Detailscommande *JOINTURE* N\_Com Commande) JOINTURE N\_pr Produit ) JOINTURE N\_cl Client)

Left : societe, nom(Client JOINTURE N\_cl (Commande JOINTURE N\_Com(Detailscommande JOINTURE N\_Pr Produit)))

22) Quel est le nombre de catégories de produit ?

*SELECT COUNT*(\*)

*FROM* Produit

Fonction de groupe (Statistiques)

23) Quel est le nombre de produits de la catégorie « boisson » ?

*SELECT COUNT*(\*)

*FROM* Produit

*WHERE* N\_Cat = 1 ;

24) Calculer le délai moyen entre la date de commande et la date d’envoi de la commande

25) Quel est le nombre de pays distincts où se trouve un fournisseur ?

*SELECT COUNT*(*DISTINCT* pays)

*FROM* Fournisseur

26) Donner les produits (avec leur prix) dont le prix est supérieur à la moyenne des prix des produits de la base

Regroupement (agrégation)

27) Afficher, pour chaque pays où se trouve un fournisseur, le nombre de fournisseurs qui s’y trouvent

28) Afficher pour chaque produit (N\_Pr), la quantité totale commandée, la quantité moyenne commandée, et le nombre de fois où il a été commandé.

Clause HAVING

29) Afficher les numéros de produits qui ont été commandés au moins deux fois

30) Donner les pays où il y a au moins 3 fournisseurs (1er solution utiliser la requête 27, 2e solution sans passer par la requête 27)

31) Donner les numéros des employés qui se sont chargés de plus de 2 commandes ainsi que le nombre des commandes pour chacun d’eux

Jointures et Groupes

32) Donner pour chaque employé, le prix total des commandes dont il s’est chargé

33) Donner pour chaque employé, le nombre de commandes dont il s’est chargé

34) Donner pour chaque employé, le prix total des commandes ainsi que le nombre des commandes dont il s’est chargé

35) Quels sont les produits qui ont été commandés au moins deux fois ?

36) Donner le nombre de ces produits

TROISIEME PARTIE : CREATION DE LA BASE SOUS ORACLE

*CREATE TABLE* **Employe** ( **N\_Emp** INT *PRIMARY KEY*,

Nom VARCHAR(20) *NOT NULL*,

Prenom VARCHAR(20),

DN Date,

Pays VARCHAR(20));

*CREATE TABLE* **Categorie** ( **N\_Cat** INT *PRIMARY KEY*,

Nom VARCHAR(20) *NOT NULL*,

Description VARCHAR(100));

*CREATE TABLE* **Client** ( **N\_CL** INT *PRIMARY KEY*,

Societe VARCHAR(50) *NOT NULL*,

Contact VARCHAR(20),

Fonction VARCHAR(30),

Ville VARCHAR(30),

Pays VARCHAR(20));

*CREATE TABLE* **Produit** ( **N\_Pr** INT *PRIMARY KEY*,

Nom VARCHAR(20) *NOT NULL*,

**N\_Fou** INT *REFERENCES* **Fournisseur**(**N\_Fou**),

**N\_Cat** INT *REFERENCES* **Categorie**(**N\_Cat**),

Prix DECIMAL(6,2),

Stock INT *CHECK*(Stock >= 0),

Indisp INT *CHECK*(Indisp IN(0,1)));

CHECK = Condition. Pour stock, la valeur doit être égale ou supérieur à 0 sinon la valeur ne s’ajoute pas. Pour Indisp, la valeur doit être soit 0, soit 1, sinon la valeur ne s’ajoute pas.

*CREATE TABLE* **Fournisseur** ( **N\_Fou** INT *PRIMARY KEY*,

Societe VARCHAR(50) *NOT NULL*,

Contact VARCHAR(20),

Fonction VARCHAR(30),

Ville VARCHAR(30),

Pays VARCHAR(20));

*CREATE TABLE* **Commande** ( **N\_Com** INT *PRIMARY KEY*,

**N\_CL** *REFERENCES* **Client**(N\_CL),

**N\_Emp** *REFERENCES* **Employe**(**N\_Emp**),

Date\_C Date,

Date\_L Date,

Date\_E Date,

Ville\_Liv VARCHAR(30));

*CREATE TABLE* **DetailsCommande** ( **N\_Com** INT *REFERENCES* **Commande**(**N\_Com**),

**N\_Pr** INT *REFERENCES* **Produit**(**N\_Pr**),

Quantite INT *DEFAULT* 0,

Remise Decimal(5,2),

*PRIMARY KEY*(**N\_Com, N\_Pr**)) ;

DEFAULT = Valeur par default. Ici Quantité sera 0 si aucune valeur n’est saisie.

*BEGIN*

*INSERT INTO* **Employe**

*VALUES* (1, 'Leblanc', 'Marie', TO\_DATE('08/12/1948', 'dd/mm/yyyy'), 'France');

*INSERT INTO* **Employe**

*VALUES* (2, 'Brachman', 'Andrew', TO\_DATE('19/02/1952', 'dd/mm/yyyy'), 'France');

*INSERT INTO* **Employe**

*VALUES* (3, 'Leclerc', 'Janet', TO\_DATE('30/08/1963', 'dd/mm/yyyy'), 'France');

*INSERT INTO* **Employe**

*VALUES* (4, 'Durant', 'Margaret', TO\_DATE('19/09/1937', 'dd/mm/yyyy'), 'France');

*INSERT INTO* **Employe**

*VALUES* (5, 'Dupont', 'Steven', TO\_DATE('04/03/1955', 'dd/mm/yyyy'), 'Royaume-Uni');

*INSERT INTO* **Employe**

*VALUES* (6, 'Delahaye', 'Michael', TO\_DATE('02/07/1963', 'dd/mm/yyyy'), 'Royaume-Uni');

*INSERT INTO* **Employe**

*VALUES* (7, 'King', 'Robert', TO\_DATE('29/05/1960', 'dd/mm/yyyy'), 'Royaume-Uni');

*INSERT INTO* **Employe**

*VALUES* (8, 'Callahan', 'Laura', TO\_DATE('09/01/1958', 'dd/mm/yyyy'), 'Etats-Unis');

*INSERT INTO* **Employe**

*VALUES* (9, 'Dodsworth', 'Anne', TO\_DATE('27/01/1966', 'dd/mm/yyyy'), 'Royaume-Uni');

*END*

*BEGIN*

*INSERT INTO* **Categorie**

*VALUES* (1, 'Boissons', 'Boissons, cafés, thés, bières');

*INSERT INTO* **Categorie**

*VALUES* (2, 'Condiments', 'Sauces, assaisonnements et épices');

*INSERT INTO* **Categorie**

*VALUES* (3, 'Desserts', 'Desserts et friandises');

*INSERT INTO* **Categorie**

*VALUES* (4, 'Produits laitiers', 'Fromages');

*INSERT INTO* **Categorie**

*VALUES* (5, 'Pâtes et céréales', 'Pains, biscuits, pâtes et céréales');

*INSERT INTO* **Categorie**

*VALUES* (6, 'Viandes', 'Viandes préparées');

*INSERT INTO* **Categorie**

*VALUES* (7, 'Produits secs', 'Fruits secs, raisins, autres');

*INSERT INTO* **Categorie**

*VALUES* (8, 'Poissons et fruits de mer', 'Poissons, fruits de mer, escargots');

*END*

*BEGIN*

*INSERT INTO* **Fournisseur**

*VALUES* (1, 'Exotic Liquids', 'Charlotte Cooper', 'Assistant export', 'London', 'Royaume-Uni');

*INSERT INTO* **Fournisseur**

*VALUES* (2, 'New Orleans Cajun Delights', 'Shelley Burke', 'Acheteur', 'New Orleans', 'Etats-Unis');

*INSERT INTO* **Fournisseur**

*VALUES* (3, 'Grandma Kelly''s Homestead', 'Regina Murphy', 'Représentant(e)', 'Ann Arbor', 'Etats-Unis');

*INSERT INTO* **Fournisseur**

*VALUES* (7, 'Pavlovla, Ltd.', 'Ian Devling', 'Directeur du marketing', 'Melbourne', 'Australie');

*INSERT INTO* **Fournisseur**

*VALUES* (8, 'Specialty Biscuits, Ltd.', 'Peter Wilson', 'Représentant(e)', 'Manchester', 'Royaume-Uni');

*INSERT INTO* **Fournisseur**

*VALUES* (26, 'Pasta Buttini s.r.l.', 'Giovanni Giudici', 'Acheteur', 'Salerno', 'Italie');

*INSERT INTO* **Fournisseur**

*VALUES* (27, 'Escargots Nouveaux', 'Marie Delamare', 'Chef des ventes', 'Montceau', 'France');

*INSERT INTO* **Fournisseur**

*VALUES* (28, 'Gai pâturage', 'Eliane Noz', 'Représentant(e)', 'Annecy', 'France');

*END*

*BEGIN*

*INSERT INTO* **Produit**

*VALUES* (1, 'Chai', 1, 1, 13.00, 39, 0);

*INSERT INTO* **Produit**

*VALUES* (2, 'Chang', 1, 1, 14.48, 17, 0);

*INSERT INTO* **Produit**

*VALUES* (4, 'Chef Anton''s Cajun Seasoning', 2, 2, 16.77, 53, 0);

*INSERT INTO* **Produit**

*VALUES* (5, 'Chef Anton''s Gumbo Mix', 2, 2, 16.27, 0, 1);

*INSERT INTO* **Produit**

*VALUES* (6, 'Grandma''s Boysenberry Spread', 3, 2, 19.00, 120, 0);

*INSERT INTO* **Produit**

*VALUES* (8, 'Northwoods Cranberry Sauce', 3, 2, 30.49, 6, 0);

*INSERT INTO* **Produit**

*VALUES* (16, 'Pavlova', 7, 3, 13.25, 29, 0);

*INSERT INTO* **Produit**

*VALUES* (17, 'Alice Mutton', 7, 6, 29.73, 0, 1);

*INSERT INTO* **Produit**

*VALUES* (19, 'Teatime Chocolate Biscuits', 8, 3, 12.00, 25, 0);

*INSERT INTO* **Produit**

*VALUES* (20, 'Sir Rodney''s Marmalade', 8, 3, 7.01, 40, 0);

*INSERT INTO* **Produit**

*VALUES* (56, 'Gnocchi di nonna Alice', 26, 5, 28.97, 21, 0);

*INSERT INTO* **Produit**

*VALUES* (58, 'Escargots de Bourgogne', 27, 8, 10.25, 62, 0);

*INSERT INTO* **Produit**

*VALUES* (59, 'Raclette Courdavault', 28, 4, 41.00, 79, 0);

*INSERT INTO* **Produit**

*VALUES* (60, 'Camemmbert Pierrot', 28, 4, 25.50, 19, 0);

*INSERT INTO* **Produit**

*VALUES* (68, 'Scottish Longbreads', 8, 3, 9.50, 6, 0);

*END*

*BEGIN*

*INSERT INTO* **Client**

*VALUES* ('BLONP', 'Blondel père et fils', 'Frédériques Cileaux' ,'Directeur du marketing', 'Strasbourg', 'France');

*INSERT INTO* **Client**

*VALUES* ('BONAP', 'Bon app''', 'Laurence Lebihan', 'Propriétaire', 'Marseille', 'France');

*INSERT INTO* **Client**

*VALUES* ('BOTTM', 'Bottom-Dollar Markets', 'Elizabeth Lincoln', 'Chef comptable', 'Tsawassen', 'Canada');

*INSERT INTO* **Client**

*VALUES* ('FOLIG', 'Folies gourmandes', 'Martine Rancé', 'Représentant(e)', 'Lille', 'France');

*INSERT INTO* **Client**

*VALUES* ('GREAL', 'Great Lakes Food Market', 'Howard Snyder', 'Directeur du marketing', 'Eugene', 'Etats-Unis');

*INSERT INTO* **Client**

*VALUES* ('KOENE', 'Könglich Essen', 'Philip Cramer', 'Assistant(e) de ventes', 'Brandenburg', 'Allemagne');

*INSERT INTO* **Client**

*VALUES* ('LACOR', 'La corne d''abondance', 'Daniel Tonini', 'Représentant(e)', 'Versailles', 'France');

*INSERT INTO* **Client**

*VALUES* ('LAMAI', 'La maison d''Asie', 'Annette Roulet', 'Chef des ventes', 'Toulouse', 'France');

*INSERT INTO* **Client**

*VALUES* ('LONEP', 'Lonesome Pine Restaurant', 'Fran Wilson', 'Chef des ventes', 'Portland', 'Etats-Unis');

*INSERT INTO* **Client**

*VALUES* ('MAGAA', 'Magazzini Alimentari Riunlti', 'Giovanni Rovelli', 'Directeur du marketing', 'Bergamo', 'Italie');

*INSERT INTO* **Client**

*VALUES* ('PARIS', 'Paris spécialités', 'Marie Bertrand', 'Propriétaire', 'Paris', 'France');

*INSERT INTO* **Client**

*VALUES* ('SEVES', 'Seven Seas Imports', 'Hari Kumar', 'Chef des ventes', 'London', 'Royaume-Uni');

*INSERT INTO* **Client**

*VALUES* ('VICTE', 'Victuailles en stock', 'Mary Saveley', 'Assistant(e) des ventes', 'Lyon', 'France');

*INSERT INTO* **Client**

*VALUES* ('VINET', 'vins et alcools Chevalier', 'Paul Henriot', 'Chef comptable', 'Reims', 'France');

*END*

*BEGIN*

*INSERT INTO* **Commande**

*VALUES* (10265, 'BLONP', 2, TO\_DATE('22/07/2006', 'dd/mm/yyyy'), TO\_DATE('19/08/2006', 'dd/mm/yyyy'), TO\_DATE('09/08/2006', 'dd/mm/yyyy'), 'Strasbourg');

*INSERT INTO* **Commande**

*VALUES* (10334, 'VICTE', 8, TO\_DATE('18/10/2006', 'dd/mm/yyyy'), TO\_DATE('15/11/2006', 'dd/mm/yyyy'), TO\_DATE('25/10/2006', 'dd/mm/yyyy'), 'Lyon');

*INSERT INTO* **Commande**

*VALUES* (10431, 'BOTTM', 4, TO\_DATE('27/01/2007', 'dd/mm/yyyy'), TO\_DATE('10/02/2007', 'dd/mm/yyyy'), TO\_DATE('04/02/2007', 'dd/mm/yyyy'), 'Tsawassen');

*INSERT INTO* **Commande**

*VALUES* (10436, 'BLONP', 3, TO\_DATE('02/02/2007', 'dd/mm/yyyy'), TO\_DATE('02/03/2007', 'dd/mm/yyyy'), TO\_DATE('08/02/2007', 'dd/mm/yyyy'), 'Strasbourg');

*INSERT INTO* **Commande**

*VALUES* (10454, 'LAMAI', 4, TO\_DATE('18/02/2007', 'dd/mm/yyyy'), TO\_DATE('18/03/2007', 'dd/mm/yyyy'), TO\_DATE('22/02/2007', 'dd/mm/yyyy'), 'Toulouse');

*INSERT INTO* **Commande**

*VALUES* (10457, 'KOENE', 2, TO\_DATE('22/02/2007', 'dd/mm/yyyy'), TO\_DATE('22/03/2007', 'dd/mm/yyyy'), TO\_DATE('28/02/2007', 'dd/mm/yyyy'), 'Bandenburg');

*INSERT INTO* **Commande**

*VALUES* (10511, 'BONAP', 4, TO\_DATE('15/04/2007', 'dd/mm/yyyy'), TO\_DATE('13/05/2007', 'dd/mm/yyyy'), TO\_DATE('18/04/2007', 'dd/mm/yyyy'), 'Marseille');

*INSERT INTO* **Commande**

*VALUES* (10523, 'SEVES', 7, TO\_DATE('28/04/2007', 'dd/mm/yyyy'), TO\_DATE('26/05/2007', 'dd/mm/yyyy'), TO\_DATE('27/05/2007', 'dd/mm/yyyy'), 'London');

*INSERT INTO* **Commande**

*VALUES* (10681, 'GREAL', 3, TO\_DATE('22/09/2007', 'dd/mm/yyyy'), TO\_DATE('20/10/2007', 'dd/mm/yyyy'), TO\_DATE('27/09/2007', 'dd/mm/yyyy'), 'Eugene');

*END*

*BEGIN*

*INSERT INTO* **DetailsCommande**

*VALUES* (10265, 17, 30, 0);

*INSERT INTO* **DetailsCommande**

*VALUES* (10334, 68, 10, 0);

*INSERT INTO* **DetailsCommande**

*VALUES* (10431, 17, 50, 25);

*INSERT INTO* **DetailsCommande**

*VALUES* (10436, 56, 40, 10);

*INSERT INTO* **DetailsCommande**

*VALUES* (10454, 16, 20, 20);

*INSERT INTO* **DetailsCommande**

*VALUES* (10457, 59, 36, 0);

*INSERT INTO* **DetailsCommande**

*VALUES* (10511, 4, 50, 15);

*INSERT INTO* **DetailsCommande**

*VALUES* (10511, 8, 10, 15);

*INSERT INTO* **DetailsCommande**

*VALUES* (10523, 17, 25, 10);

*INSERT INTO* **DetailsCommande**

*VALUES* (10523, 20, 15, 10);

*INSERT INTO* **DetailsCommande**

*VALUES* (10681, 19, 30, 10);

*INSERT INTO* **DetailsCommande**

*VALUES* (10681, 20, 20, 0);

*INSERT INTO* **DetailsCommande**

*VALUES* (10681, 68, 30, 20);

*END*