

# La commande SELECT sur une table



Recherche d'informations

DISTINCT

GROUP BY

HAVING

ORDER

30/06/2010

XH

1



## Syntaxes

### ■ Syntaxe générale de *requêteSQL*

```
selectSQL |  
(requêteSQL) {UNION|INTERSECT|EXCEPT} (requêteSQL)
```

### ■ Syntaxe du *selectSQL*

```
SELECT      {[ALL|DISTINCT] expression [AS nomColonne]  
            [,expression [AS nomColonne]]...} | *  
FROM        table [AS nomTable [(nomColonne[,nomColonne]]...)]  
            [,table [AS nomTable [(nomColonne[,nomColonne]]...)]...]  
[WHERE      conditionSQL]  
[GROUP BY   nomColonne [,nomColonne]...]  
[HAVING     conditionSQL]  
[ORDER BY   nomColonne [ASC|DESC] [,nomColonne[ASC|DESC]]...]
```

■ ...

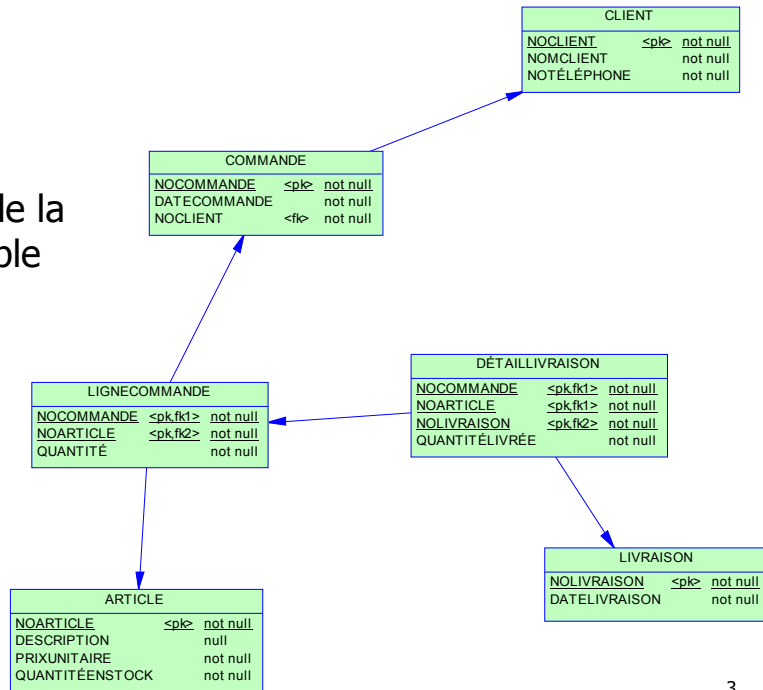
XH

2

## Schéma de la BD exemple

XH

3



## Une commande et ses lignes

<b>ABC Office Supplies Co.</b> 1 Market Street San Francisco, CA 94101		Date: 12/31/ 98		<b>INVOICE</b> No. 2346	
<b>Bill to:</b>  Major Corp. PO Box 2236 Richmond, VA 23220			<b>Ship to:</b>  Major Corp., Main Receiving 1 Park Ave. Richmond, VA 23209		
Your Order No. 1123	Cust. Ref. No.	Order Date 8/12/ 97	Terms Net 30 Days		
Qty	Unit	Item No.	Description	Unit Price	Total Price
20	DZ	41358	Pencils	9.55	191.00
10	DZ	22211	Ball Point Pens	6.05	60.50

XH

4



## Exemple: la base de données Livraison (script Oracle)

XH

```
CREATE TABLE Client
(noClient      INTEGER      NOT NULL,
nomClient     VARCHAR(20)   NOT NULL,
noTéléphone   VARCHAR(15)   NOT NULL,
PRIMARY KEY   (noClient)
)

CREATE TABLE Article
(noArticle     INTEGER      NOT NULL,
description    VARCHAR(20)   NOT NULL,
prixUnitaire  DECIMAL(10,2) NOT NULL,
quantitéEnStock INTEGER      DEFAULT 0 NOT NULL
CHECK (quantitéEnStock >= 0),
PRIMARY KEY   (noArticle)
)

CREATE TABLE Commande
(noCommande    INTEGER      NOT NULL,
dateCommande  DATE          NOT NULL,
noClient      INTEGER      NOT NULL,
PRIMARY KEY   (noCommande),
FOREIGN KEY   (noClient) REFERENCES Client
)

CREATE TABLE LigneCommande
(noCommande    INTEGER      NOT NULL,
noArticle     INTEGER      NOT NULL,
quantité      INTEGER      NOT NULL
CHECK (quantité > 0),
PRIMARY KEY   (noCommande, noArticle),
FOREIGN KEY   (noCommande) REFERENCES Commande,
FOREIGN KEY   (noArticle) REFERENCES Article
)

CREATE TABLE Livraison
(noLivraison   INTEGER      NOT NULL,
dateLivraison DATE          NOT NULL,
PRIMARY KEY   (noLivraison)
)

CREATE TABLE DétailLivraison
(noLivraison   INTEGER      NOT NULL,
noCommande    INTEGER      NOT NULL,
noArticle     INTEGER      NOT NULL,
quantitéLivrée INTEGER      NOT NULL
CHECK (quantitéLivrée > 0),
PRIMARY KEY   (noLivraison, noCommande, noArticle),
FOREIGN KEY   (noLivraison) REFERENCES Livraison,
```

3



## Projection d'une table et la clause ALL (par défaut)

```
SELECT  noClient, dateCommande
FROM    Commande
```

- Produire les *noClient* et *dateCommande* de toutes les *Commandes*

noClient	dateCommande
10	01/06/2000
20	02/06/2000
10	02/06/2000
10	05/07/2000
30	09/07/2000
20	09/07/2000
40	15/07/2000
40	15/07/2000

Multi-ensemble !

```
SELECT  ALL noClient, dateCommande
FROM    Commande
```

XH

6



# Clause DISTINCT

```
SELECT DISTINCT noClient, dateCommande
FROM Commande
```

- Produire les *noClient* et *dateCommande* de toutes les *Commandes*

noClient	dateCommande
10	01/06/2000
20	02/06/2000
10	02/06/2000
10	05/07/2000
30	09/07/2000
20	09/07/2000
40	15/07/2000

$$\pi_{noClient, dateCommande} (Commande)$$

XH

7



# WHERE (Restriction sur une table)

- Sélectionner les *Articles* dont le prix est inférieur à 20.00 E et le numéro est supérieur à 30

```
SELECT *
FROM Article
WHERE prixUnitaire < 20 AND noArticle > 30
```

noArticle	description	prixUnitaire
60	Erable argenté	15.99
70	Herbe à puce	10.99
95	Génévrier	15.99

$$\sigma_{prixUnitaire < 20.00 \text{ ET } noArticle > 30} (Article)$$

XH

8



## Syntaxe de *conditionSQL*

```
{conditionSimple  
(conditionSQL) |  
NOT(conditionSQL) |  
conditionSQL AND conditionSQL |  
conditionSQL OR conditionSQL}
```

- Sous-syntaxe (incomplète) de la *conditionSimple* :

```
{expression {=<|>|<=|>=|<>} expression|  
expression BETWEEN expression AND expression|  
expression {IS NULL |IS NOT NULL}|  
expression {IN |NOT IN} listeConstantes|  
expression {LIKE |NOT LIKE} patron}
```

XH

9



## ConditionSQL BETWEEN

- Sélectionner les *Commandes* du mois de juin de l'année 2000

```
SELECT      *  
FROM        Commande  
WHERE       dateCommande BETWEEN '01/06/2000' AND '30/06/2000'
```

```
SELECT      *  
FROM        Commande  
WHERE       dateCommande >= '01/06/2000' AND  
            dateCommande <='30/06/2000'
```

XH

10



## ConditionSQL IN

- Sélectionner les *Commandes* du *Client* dont le *noClient* est 10 ou 40 ou 80

```
SELECT      *  
FROM        Commande  
WHERE       noClient IN (10, 40, 80)
```

```
SELECT      *  
FROM        Commande  
WHERE       noClient = 10 OR noClient = 40 OR noClient = 80
```



## ConditionSQL LIKE

- Sélectionner les *Clients* dont le *nomClient* contient le mot *Le*

```
SELECT      *  
FROM        Client  
WHERE       nomClient LIKE '%Le%'
```

- 2ième lettre du *nomClient* = *o* et dernière lettre est un *k*

```
SELECT      *  
FROM        Client  
WHERE       nomClient LIKE '_o%k'
```



## ConditionSQL IS NOT NULL

- Sélectionner les *Articles* dont la description n'est pas une valeur nulle

```
SELECT      *  
FROM        Article  
WHERE       description IS NOT NULL
```



## Restriction-projection sur une table

- Produire les *noClient* et *dateCommande* des *Commandes* dont la date est supérieure au *05/07/2000*

```
SELECT      noClient, dateCommande  
FROM        Commande  
WHERE       dateCommande > '05/07/2000'
```

noClient	dateCommande
30	09/07/2000
20	09/07/2000
40	15/07/2000
40	15/07/2000



## Expressions générales sur les colonnes

```
SELECT noArticle, prixUnitaire, prixUnitaire*1.15 AS prixPlusTaxe
FROM Article
```

- La liste des *noArticle* avec le *prixUnitaire* avant et après inclusion de la taxe de 15%

noArticle	prixUnitaire	prixPlusTaxe
10	10.99	12.64
20	12.99	14.94
40	25.99	29.89
50	22.99	26.44
60	15.99	18.39
70	10.99	12.64
80	26.99	31.04
81	25.99	29.89
90	25.99	29.89
95	15.99	18.39

XH

15



## Expressions (suite)

- Produire le détail de chacun des *Articles* commandés dans la *Commande* n°1 incluant le prix total avant et après la taxe de 15% pour chacun des *Articles* commandés

```
SELECT L.noArticle, quantité, prixUnitaire, prixUnitaire*quantité AS total,
       prixUnitaire*quantité*1.15 AS totalPlusTaxe
FROM   LigneCommande AS L, Article AS A
WHERE  L.noArticle = A.noArticle AND
       L.noCommande = 1
```

noArticle	quantité	prixUnitaire	total	totalPlusTaxe
10	10	10.99	109.90	126.38
70	5	10.99	54.95	63.19
90	1	25.99	25.99	29.89

XH

16





## Expression sur colonne du WHERE

- Les *Articles* dont le *prixUnitaire* incluant la taxe de 15% est inférieur à 16.00 E

```
SELECT noArticle, prixUnitaire, prixUnitaire*1.15 AS prixPlusTaxe
FROM Article
WHERE prixUnitaire*1.15 < 16
```

noArticle	prixUnitaire	prixPlusTaxe
10	10.99	12.64
20	12.99	14.94
70	10.99	12.64



## Opérateurs arithmétiques et concaténation

Symbole	Signification
+	Somme
-	Différence
*	Produit
/	Division
	Concaténation de chaîne (SQL2)

- Conversions automatiques entre types compatibles



## Priorité (en ordre décroissant)

+, - (unaire)
*, /
+, -,
=, !=, <, >, <=, >=, IS NULL, LIKE, BETWEEN, IN, SIMILAR
NOT
AND
OR



## END

