

Université **IBM i**

19 et 20 novembre 2024

IBM Innovation Studio Paris

**S22 – Regrouper, accumuler,
puis diviser à nouveau avec SQL**

19 novembre 16:00 - 17:00

Birgitta Hauser

Birgitta Hauser – Modernization – Education – Consulting on IBM i
Hauser@ModEdCon.com

📶 **uii2024**
#ibmi
#uii2024

IBM

common
FRANCE

2

Landsberg am Lech



17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 3

IBM i

Université IBM i – 19 et 20 novembre 2024

IBM Champion since 2020



3

1

Agenda

Regroupement multidimensionnel – avec (Sous-)Totaux et Cumuls

- Fonction d'agrégation: GROUPING()

Instruction CASE WHEN ... END (si ... alors ... autrement)

- Utilisée pour générer des tableaux croisés dynamiques

Fonction d'agrégation LISTAGG(): agréger plusieurs valeurs dans une seule chaîne de caractères

- Trier la liste des valeurs
- Supprimer les doublons

Fonction de Tableau SPLIT(): Diviser un texte en plusieurs valeurs en fonction d'un séparateur

Services SQL IFS_READ... pour accéder à des fichiers situés dans l'IFS

Découper un fichier *.csv situé dans l'IFS

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 4

IBM Champion since 2020



4

GROUP BY

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 5

IBM i

Université IBM i – 19 et 20 novembre 2024

IBM Champion since 2020



5

2

L'instruction GROUP BY

GROUP BY est utilisé pour regrouper des données identiques

- Souvent utilisé en liaison avec des **fonctions d'agrégation** pour cumuler les données

Indication dans une instruction SELECT

- Après les conditions **WHERE**
- Après l'indication **FROM** (S'il n'y a pas des conditions **WHERE**)



L'instruction GROUP BY- Syntaxe

```
SELECT List Columns/Expression, AggregateFunctions
FROM Schema.Table or Vue
GROUP BY List Columns/Expressions
```

Agréger les données de plusieurs lignes

- **Toutes** les colonnes/expressions **sans** fonction d'agrégation (Grouping Expression) **doivent** être répétées après le **GROUP BY**
- **Les noms générés** pour des colonnes/expressions dans la liste **SELECT** ne sont **pas admis**
→ même **SUB-SELECT**

```
Select Year(SalesDate) SalesYear, CustNo,
Sum(Amount) Total, Count(*) Positions
From Sales
Group By Year(SalesDate), CustNo
Order By SalesYear, CustNo;
```

SALESYEAR	CUSTNO	TOTAL	POSITIONS
2008	10001	115,00	2
2008	10002	1350,00	1
2008	10003	535,00	3

• Grouping Expression doit être **répétée** dans l'instruction **GROUP BY**

• L'instruction **ORDER BY** est en dehors du Sub-Select
→ **Noms générés** peuvent être utilisés



L'instruction GROUP BY - Exemples

```

Select Year(SalesDate) SalesYear, CustNo,
       Count(*) "NbrRows", Sum(Amount) "Total",
       Cast(Avg(Amount) as Dec(11, 2)) "Average",
       Min(Amount) "Minimum", Max(Amount) "Maximum"
From Sales
Where ItemNo Between '5100' and '5300'
Group By Year(SalesDate), CustNo
Order By CustNO, SalesYear;
    
```

- **Grouping Expression**
- **Différent** de l'ordre des colonnes dans les instructions GROUP BY et ORDER BY
- Condition WHERE – usage de **Colonnes** qui **ne sont pas** dans la Grouping Expression

SALESYEAR	CUSTNO	NbrRows	Total	Average	Minimum	Maximum
2008	10001	2	115,00	57,50	55,00	60,00
2009	10001	15	2634,20	175,61	20,00	450,85
2010	10001	2	281,94	140,97	140,97	140,97
2008	10002	1	1350,00	1350,00	1350,00	1350,00
2009	10002	3	489,90	163,30	20,00	444,50
2008	10003	1	350,00	350,00	350,00	350,00
2009	10003	3	853,10	284,36	180,00	444,50
2009	10004	8	1963,55	245,44	150,00	431,80
2008	10005	1	310,00	310,00	310,00	310,00
2009	10005	3	904,45	301,48	225,00	393,70
2010	10006	4	2247,30	561,82	495,00	628,65



Regroupement multidimensionnel



Regroupement multidimensionnel

Extensions supplémentaires pour l'instruction GROUP BY:

- RollUp(Grouping Columns/Fields/Expressions)
- Cube(Grouping Columns/Fields/Expressions)
- Grouping Sets(Grouping Columns/Fields/Expressions)

CUBE and ROLLUP peuvent être utilisés dans des GROUPING SET()s



Regroupement multidimensionnel – Extension ROLLUP()

Extension RollUp()

- Génération de **sous-totaux** basés sur les colonnes indiquées dans le **RollUp()**

Exemple: **GROUP BY ROLLUP(Année, Mois, Jour)**

- Génération de **sous-totaux suivants**:
 - Sous-Total par Année, Mois, Jour
 - Sous-Total par Année, Mois,
 - Sous-Total par Année
 - Total Général
- **L'ordre des résultats dépend** des indications dans l'instruction **ORDER BY**



Regroupement multidimensionnel – Extension ROLLUP() - Exemple

```
Select Year(SalesDate) SalesYear, CustNo, Sum(Amount) Total
From Sales
Group By RollUp(Year(SalesDate), CustNo)
Order By SalesYear, CustNo
```

SALESYEAR	CUSTNO	TOTAL
2008	10001	115,00
2008	10002	1350,00
2008	10003	535,00
2008	10004	470,00
2008	10005	310,00
2008	-	2780,00
2009	10001	2634,20
2009	10002	1636,25
2009	10003	4589,86
2009	10004	2673,95
2009	10005	3741,95
2009	-	15276,21
2010	10001	281,94
2010	10003	1555,75
2010	10006	19425,70
2010	-	21263,39
-	-	39319,60

• Chiffres d'affaires par an et par client

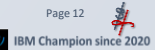
• Sous-Totaux par:
 ▪ Année/Client
 ▪ Année

• Total Général

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 12



Regroupement multidimensionnel – Extension CUBE()

Extension CUBE

- Génération des **Sous-Totaux** pour **chaque composition possible** des colonnes indiquées dans l'instruction **GROUP BY**
 → Tous les Sous-Totaux générés par RollUp() + "cross tabulation"

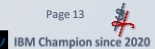
Exemple: **Group By Cube(Année, Client, Article)**

- Génération de **sous-totaux suivants**:
 - Sous-Total par Année, Client, Article
 - Sous-Total par Année, Client
 - Sous-Total par Année, Article
 - Sous-Total par Client, Article
 - Sous-Total par Année
 - Sous-Total par Client
 - Sous-Total par Article
 - Total Général
- **L'ordre des résultats dépend** des indications dans l'instruction **ORDER BY**

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 13



Regroupement multidimensionnel – Extension Cube() versus RollUp() - Exemples

```
select Year(SalesDate) SalesYear, CustNo, ItemNo,
       Sum(Amount) Total
From Sales
where SalesDate between '2008-10-01'
              and '2009-03-31'
       and CustNo in ('10001', '10002')
Group By Cube(Year(SalesDate), CustNo, Item)
Order By SalesYear, CustNo, Item
```

SALESYEAR	CUSTNO	ITEMNO	TOTAL
2008	10001	King,Stephen - Es	115,00
2008	10001	-	115,00
2008	10002	King,Stephen - Es	1350,00
2008	10002	-	1350,00
2008	-	King,Stephen - Es	1465,00
2008	-	-	1465,00
2009	10001	Grisham,John - Die Akte	180,00
2009	10001	King,Stephen - Drei	160,00
2009	10001	-	340,00
2009	-	Grisham,John - Die Akte	180,00
2009	-	King,Stephen - Drei	160,00
2009	-	-	340,00
-	10001	Grisham,John - Die Akte	180,00
-	10001	King,Stephen - Drei	160,00
-	10001	King,Stephen - Es	115,00
-	10001	-	455,00
-	10002	King,Stephen - Es	1350,00
-	10002	-	1350,00
-	-	Grisham,John - Die Akte	180,00
-	-	King,Stephen - Drei	160,00
-	-	King,Stephen - Es	1465,00
-	-	-	1805,00

```
select Year(SalesDate) SalesYear, CustNo, Item,
       Sum(Amount) Total
From Sales
where SalesDate between '2008-10-01'
              and '2009-03-31'
       and CustNo in ('10001', '10002')
Group By RollUp(Year(SalesDate), CustNo, Item)
Order By SalesYear, CustNo, Item
```

SALESYEAR	CUSTNO	ITEM	TOTAL
2008	10001	King,Stephen - Es	115,00
2008	10001	-	115,00
2008	10002	King,Stephen - Es	1350,00
2008	10002	-	1350,00
2008	-	-	1465,00
2009	10001	Grisham,John - Die Akte	180,00
2009	10001	King,Stephen - Drei	160,00
2009	10001	-	340,00
2009	-	-	340,00
-	-	-	1805,00

• Cross Tabulation



Regroupement multidimensionnel – Extension Grouping Sets ()

Grouping Sets() Extension

- Définition et création de **n'importe quel** niveau de regroupement/sous-total

Exemple: Grouping Sets((Année, Client), (Année, Article), (Année), ())

- Génération de **sous-totaux** suivants:
 - Sous-Total par Année, Client
 - Sous-Total par Année, Article
 - Sous-Total par Année
 - Total Général
- **L'ordre des résultats dépend** des indications dans l'instruction **ORDER BY**



Regroupement multidimensionnel – Extension Grouping Sets() - Exemple

```
select Year(SalesDate) SalesYear, CustNo, Item, Sum(Amount) Total
From Sales
Where CustNo between 10001 and 10003
Group By Grouping Sets((Year(SalesDate), CustNo),
                       (Year(SalesDate), Item),
                       (Year(SalesDate)), ())
Order By 1, 3, 4, 2
```

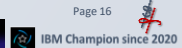
SALESYEAR	CUSTNO	ITEM	TOTAL
2008	10001	-	115,00
2008	10003	-	535,00
2008	10002	-	1350,00
2008	-	-	2000,00
2009	-	Grisham,John - Die Akte	896,65
2009	-	King,Stephen - Drei	1261,70
2009	-	King,Stephen - Es	1818,85
2009	-	King,Stephen - Shining	4883,11
2009	10002	-	1636,25
2009	10001	-	2634,20
2009	10003	-	4589,86
2009	-	-	8860,31
2010	-	King,Stephen - Es	281,94
2010	-	King,Stephen - Shining	1555,75
2010	10001	-	281,94
2010	10003	-	1555,75
2010	-	-	1837,69
-	-	-	12698,00

- Sous-Total par Année/Article
- Sous-Total par Année/Client
- Sous-Total par Année
- Total Général

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 16



Regroupement multidimensionnel – Fonction d'agrégation GROUPING()

Grouping(ColumnName)

Fonction d'agrégation: Grouping(Column)

- Ne peut être utilisé qu'en liaison avec le regroupement multidimensionnel

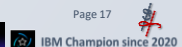
Vérifier les valeurs NULL dans les lignes de (sous-)totaux

- **1** = Valeur NULL dans la colonne indiquée de la ligne des Sous-Totaux
- **0** = Pas de Valeur NULL dans la colonne indiquée de la ligne des Sous-Totaux

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 17



Regroupement multidimensionnel – Fonction d’agrégation GROUPING() - Exemple

```
With x as (Select Year(SalesDate) SalesYear, CustNo, Amount
           from Sales)
Select Case When Grouping(SalesYear) = 1
           Then 'Grand Total'
           When Grouping(CustNo) = 1
           Then 'Total Year ' concat Varchar(SalesYear)
           Else '' End,
       SalesYear, CustNo, Sum(Amount) as Total
from x
Group By Rollup(SalesYear, CustNo)
Order By SalesYear, CustNo
```

00001	SALESYEAR	CUSTNO	TOTAL
	2008	10001	115,00
	2008	10002	1350,00
	2008	10003	535,00
	2008	10004	470,00
	2008	10005	310,00
Total Year 2008	2008 -		2780,00
	2009	10001	2634,20
	2009	10002	1636,25
	2009	10003	4389,86
	2009	10004	2673,95
	2009	10005	3741,95
Total Year 2009	2009 -		15276,21
	2010	10001	281,94
	2010	10003	1555,75
	2010	10006	19425,70
Total Year 2010	2010 -		21263,39
Grand Total	--		39319,60

- Générer des textes pour les sous-totaux à la base des informations retournées par la fonction d’agrégation GROUPING()

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 18

IBM Champion since 2020



Regroupement multidimensionnel – Fonction d’agrégation GROUPING() - Exemple

- Affichage des sous-totaux et total général uniquement

```
With x as (Select Year(SalesDate) SalesYear, CustNo, Amount
           from Sales)
Select Case When Grouping(SalesYear) = 1
           Then 'Grand Total'
           When Grouping(CustNo) = 1
           Then 'Total Year ' concat Varchar(SalesYear)
           Else '' End,
       SalesYear, CustNo, Sum(Amount) as Total
from x
Group By Rollup(SalesYear, CustNo)
Having Grouping(CustNo) = 1
Order By SalesYear, CustNo
```

00001	SALESYEAR	CUSTNO	TOTAL
Total Year 2008	2008 -		2780,00
Total Year 2009	2009 -		15276,21
Total Year 2010	2010 -		21263,39
Grand Total	--		39319,60

- Retourner les différents sous-totaux pris en compte par la fonction d’agrégation GROUPING() dans l’instruction Having

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 19

IBM Champion since 2020



Expression Case en relation avec des Fonctions d'agrégat

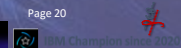
17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 20

IBM i

Université IBM i – 19 et 20 novembre 2024



20

Expression CASE

Permet d'agir **différemment** dans des situations **différentes**

- Type d'une requête IF-THEN-ELSE (Si-alors-autrement)

```
CASE Condition
  WHEN Value1 THEN ThisValue
  WHEN Value2 THEN ThatValue
  ELSE OtherValue
END
```

```
Select Case Month(SalesDate)
  When 1 Then 'Janvier'
  When 2 Then 'Février'
  Else 'Autre Mois'
  End
From Sales
```

```
CASE WHEN Condition1 THEN ThisValue
  WHEN Condition2 THEN ThatValue
  ELSE OtherValue
END
```

```
Select Case When Amount < 100 Then 'BAS'
  When Amount < 500 Then 'Moyen'
  When Amount >= 500 Then 'Elevé'
  End
From Sales
```

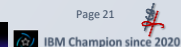
17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 21

IBM i

Université IBM i – 19 et 20 novembre 2024



21

Expression CASE

Afficher des valeurs différentes basée sur la valeur d'origine d'une colonne

```

Select Status, case Status when 'CP' then 'Completed'
                        when 'CL' then 'Cancelled'
                        when 'EN' then 'Entered'
                        when 'PD' then 'Partly Delivered'
                        else 'Unknown Status' end as StsText,
a.*
from OrderDetX a
    
```

En fonction de la condition, afficher des valeurs de colonnes différents

```

Select OrderQty, DelQty, Status,
       case Status when 'CP' then DelQty
              when 'CL' then OrderQty
              when 'EN' then OrderQty
              when 'PD' then DelQty
              else 0 end as Quantity,
a.*
From OrderDetX;
    
```

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 22



Fonction d'agrégation et expression CASE - Tableau croisé dynamique

```

Select Trim(CustNo) CustNo, Year(SalesDate) SalesYear,
sum(case when Month(SalesDate)= 1 then Amount else 0 end) as Jan,
sum(case when Month(SalesDate)= 2 then Amount else 0 end) as Feb,
sum(case when Month(SalesDate)= 3 then Amount else 0 end) as Mar,
sum(case when Month(SalesDate)= 4 then Amount else 0 end) as Apr,
sum(case when Month(SalesDate)= 5 then Amount else 0 end) as May,
sum(case when Month(SalesDate)= 6 then Amount else 0 end) as Jun,
sum(case when Month(SalesDate)= 7 then Amount else 0 end) as Jul,
sum(case when Month(SalesDate)= 8 then Amount else 0 end) as Aug,
sum(case when Month(SalesDate)= 9 then Amount else 0 end) as Sep,
sum(case when Month(SalesDate)= 10 then Amount else 0 end) as Oct,
sum(case when Month(SalesDate)= 11 then Amount else 0 end) as Nov,
sum(case when Month(SalesDate)= 12 then Amount else 0 end) as Dec,
sum(Amount) as Total
from Sales
group by CustNo, Year(SalesDate)
order by CustNo, SalesYear
    
```

CUSTNO	SALES YEAR	JAN	FEB	MAR	APR	MAY	JUN	JUL	AUG	SEP	OCT	NOV	DEC	TOTAL
10001	2008	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
10001	2009	160,00	0,00	180,00	299,40	0,00	0,00	240,40	814,60	63,50	425,45	450,85	0,00	2634,20
10001	2010	281,94	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	281,94
10002	2008	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
10002	2009	0,00	0,00	0,00	0,00	0,00	20,00	250,00	280,40	0,00	762,00	323,85	0,00	1636,25
10003	2008	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
10003	2009	333,00	200,00	200,00	1057,91	0,00	0,00	254,00	1749,00	224,45	0,00	571,50	0,00	4589,86
10003	2010	1355,75	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	1355,75
10004	2008	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
10004	2009	340,00	0,00	50,00	1121,55	0,00	155,00	220,00	450,85	57,15	0,00	279,40	0,00	2673,95
10005	2008	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
10005	2009	0,00	225,00	1250,00	0,00	0,00	285,75	0,00	1587,50	0,00	393,70	0,00	0,00	3741,95
10006	2010	1831,50	1100,00	1100,00	0,00	0,00	0,00	1397,00	9619,50	1234,46	0,00	3143,24	0,00	19425,70

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 23



Expression CASE – Analyse ABC

```

Select CustNo, Case When Sum(Amount) > 10000 Then 'A'
                  When Sum(Amount) > 4000 Then 'B'
                  Else 'C' End "ABC",

                  Case When Sum(Amount) > 10000
                        then Sum(Amount) Else 0 End "A Customer",
                  Case When Sum(Amount) between 4000 and 10000
                        Then Sum(Amount) Else 0 End "B Customer",
                  Case When Sum(Amount) < 4000
                        Then Sum(Amount) Else 0 End "C Customer"

From Sales
Group by CustNo
Order By Sum(Amount) Desc;;
    
```

- Client A: Total > 10000
- Client B: Total > 4000

CUSTNO	ABC	A Customer	B Customer	C Customer
10006	A	19425,70	0,00	0,00
10003	B	0,00	6680,61	0,00
10005	B	0,00	4051,95	0,00
10004	C	0,00	0,00	3143,95
10001	C	0,00	0,00	3031,14
10002	C	0,00	0,00	2986,25



Vues

17.11.2024
Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser
Page 25
IBM Champion since 2020


Vue – Fichier logique sans clé

Une vue est créée par l'instruction SQL **CREATE VIEW**

- Une vue décrit comment accéder aux données par l'instruction **SELECT** enregistrée
- Une vue **ne contient pas** de données
- Une vue est l'équivalent d'un **fichier logique sans clé**



Vue – Fichier logique sans clé

Pourquoi et quand utiliser?

- S'il faut **renommer** des colonnes ou en **créer** de nouvelles
- S'il faut convertir **des données** → par exemple des champs numériques pour dates
- Si l'on veut contrôler l'accès **aux données sensibles**
 - Créer une vue sans les colonnes de données sensibles
 - Si une autorisation d'accès spéciale est nécessaire
- **Déplacer la logique d'entreprise dans la base de données**
 - Une vue peut être utilisée dans des instructions SQL partout où un tableau peut être utilisé
 - **Masquer la complexité** aux utilisateurs/programmeurs
 - **Réduction du code source**, n'importe quel que soit la langue de programmation utilisé.
- **Indépendance vis-à-vis des changements dans les fichiers physiques/tableaux SQL**



Vue – Fichier logique sans clé

Peut contenir tout ce qui est possible dans une instruction **SELECT ...** **sauf** l'instruction **ORDER BY**

- la sélection de colonnes et la création de nouvelles colonnes
- Tous les types des expressions **JOIN** (Inner, Left/Right/Full Outer, Exception, Cross Join)
- Conditions **WHERE**
- **GROUP BY** (y compris le regroupement multidimensionnel) ainsi que l'instruction **HAVING**
- **Fonctions scalaires** / Fonctions (tableaux) définies par les utilisateurs i.e. User Defined (Table) Functions
- Expressions **CASE**
- **UNION / EXCEPT / INTERSECT**
- **Common Table Expressions** (CTE) / **Sub-Selects** imbriqués
- CTEs **Récurrents** et **Hierarchical Query Clauses**
- Utilisation des **Variables Globales SQL**

Une vue peut être construite sur une autre vue → Vues **imbriquées**

- Requêtes complexes peuvent être divisées en plusieurs étapes logiques

Toujours sans clé → Access Path Maintenance ***REBLD**

- Même des milliers de vues peuvent exister **sans aucune diminution de performance** !

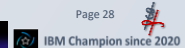
17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 28

IBM i

Université IBM i – 19 et 20 novembre 2024



28

LISTAGG – Fonction d'agrégation

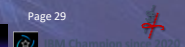
17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 29

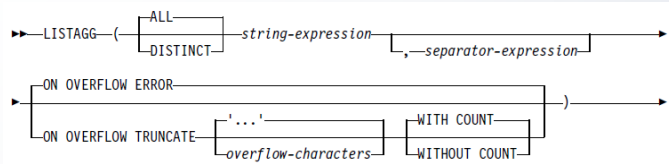
IBM i

Université IBM i – 19 et 20 novembre 2024



29

Fonction d'agrégation LISTAGG()



Agrégation d'un ensemble d'éléments de chaînes en une seule chaîne

- Les valeurs des expressions de chaînes de chaque ligne sont **concaténées**
- Un **séparateur**, qui est inséré entre les **différentes valeurs de lignes** peut également être indiqué
- **DISTINCT** peut être spécifié dans la fonction pour **éliminer les doublons**

ON OVERFLOW TRUNCATE → peut être indiqué en option

- sans cette indication, des **erreurs se produisent** si les chaînes générées **dépassent la longueur maximale**
 - Longueur maximale: Type de donnée: Char/VarChar: 4.000 Caractères Graphic/VarGraph: 2.000 Caractères CLOB, DBLOB: 1 MB
 - Overflow characters: Texte pour la suite peut être indiqué
 - With/Without Count: Afficher Nombre de lignes manquantes peut être indiqué

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 30



Fonction d'agrégation LISTAGG()

```
Select Company, OrderNo, ListAgg(Trim(ItemNo), ', ') Items
From OrderDetX d
Group By Company, OrderNo
Order By Company, OrderNo;
```

COMPANY	ORDERNO	ITEMS
10	BNR2009-10-15/1	CF001, CF003, HG001, BS002
10	BNR2009-10-15/2	HG001, HG002, CF001, CF002, BS003
10	BNR2009-10-20/2	CF003, CF004, CF005, BS001, BS002, BS003
10	BNR2009-10-23/7	BS001, HG001, HG002
10	BNR2009-10-30/2	BS001, BS002, BS003
10	BNR2009-12-15/1	CF001, CF003, HG001, BS002
10	BNR2009-12-15/2	HG001, HG002, CF001,
10	BNR2009-12-20/2	CF003, CF004, CF005,
10	BNR2009-12-23/7	BS001, HG001, HG002
10	BNR2009-12-30/2	BS001, BS002, BS003

- Cumuler les détails de la commande par entreprise et par commande
 - Concaténer tous les numéros d'article d'une commande dans une chaîne de caractères
- Numéros d'articles par Order dans une **séquence indéfini**

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser



Fonction d'agrégation LISTAGG()

Exemple: Liste des Commandes avec la description de chaque article

```

Select Company, OrderNo,
      ListAgg(Trim(ItemNo) concat ' ' concat Trim(Descript), ', ') Items
from OrderDetX d join ItemMastX i Using (Company, ItemNo)
Group By Company, OrderNo
Order By Company, OrderNo;

```

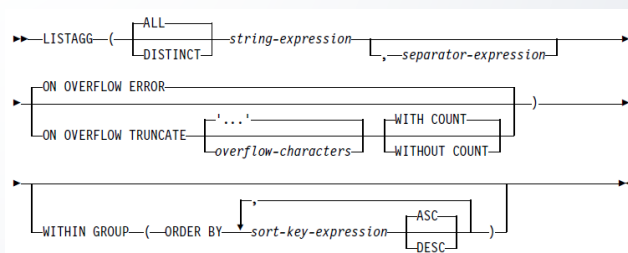
COMPANY	ORDERNO	ITEMS
10	BNR2009-10-15/1	CF001 Champagne flute 1, CF003 Champagne flute 3, HG001 Hunting gun ...
10	BNR2009-10-15/2	HG001 Hunting gun 1, HG002 Hunting gun 2, CF001 Champagne flute 1, C...
10	BNR2009-10-20/2	CF003 Champagne flute 3, CF004 Champagne flute 4, CF005 Champagne fl...
10	BNR2009-10-23/7	BS001 Book shelves 1, HG001 Hunting gun 1, HG002 Hunting gun 2
10	BNR2009-10-30/2	BS001 Book shelves 1, BS002 Book shelves 2, BS003 Book shelves 3
10	BNR2009-12-15/1	CF001 Champagne flute 1, CF003 Champagne flute 3, HG001 Hunting gun ...
10	BNR2009-12-15/2	HG001 Hunting gun 1, HG002 Hunting gun 2, CF001 Champagne flute 1, C...
10	BNR2009-12-20/2	CF003 Champagne flute 3, CF004 Champagne flute 4, CF005 Champagne fl...
10	BNR2009-12-23/7	BS001 Book shelves 1, HG001 Hunting gun 1, HG002 Hunting gun 2
10	BNR2009-12-30/2	BS001 Book shelves 1, BS002 Book shelves 2, BS003 Book shelves 3

- Liste des commandes avec la description des articles
 - Joindre le tableau des articles
 - Le numéro d'article et la description sont liés dans la fonction d'agrégation LISTAGG().

17.11.2024



LISTAGG – Fonction d'agrégation – 2^{ème} Partie



WITHIN GROUP (Order By ...)

- Permet de trier les valeurs de l'expression de la chaîne de caractères
- dans l'ordre ascendant ou descendant

```

Select Company, OrderNo,
      ListAgg(Trim(ItemNo), ', ') Within Group (Order By ItemNo) ListItems
From OrderDetX
Group By Company, OrderNo

```

17.11.2024



Fonction d'agrégation LISTAGG()

Exemple: LISTAGG() avec articles triée

```
Select Company, OrderNo,
       LISTAGG(Trim(ItemNo), ',')
       Within Group (Order By ItemNo) Items
From OrderDetX D
Group By Company, OrderNo
Order By Company, OrderNo;
```

• Trier le résultat par le numéro d'article

COMPANY	ORDERNO	ITEMS
10BNR2009-10-15/1		BS002, CF001, CF003, HG001
10BNR2009-10-15/2		BS003, CF001, CF002, HG001, HG002
10BNR2009-10-20/2		BS001, BS002, BS003, CF003, CF004, CF005
10BNR2009-10-23/7		BS001, HG001, HG002

```
Select Company, OrderNo,
       ListAgg(Trim(ItemNo) concat ' ' concat Trim(Descript), ',')
       Within Group (Order By Descript) Items
from OrderDetX d join ItemMastX i Using (Company, ItemNo)
Group By Company, OrderNo
Order By Company, OrderNo;
```

• Trier le résultat par la description de l'article

COMPANY	ORDERNO	ITEMS
10BNR2009-10-15/1		BS002 Book shelves 2, CF001 Champagne flute 1, CF003 Champagn...
10BNR2009-10-15/2		BS003 Book shelves 3, CF001 Champagne flute 1, CF002 Champagn...
10BNR2009-10-20/2		BS001 Book shelves 1, BS002 Book shelves 2, BS003 Book shelve...



Fonction d'agrégation LISTAGG() – Exemples

```
Select CustNo, ListAgg(Trim(ItemNo), ',') within Group (Order By ItemNo)
from Sales
Group By CustNo;
```

CUSTNO	ITEMS
00002	
10001	5100, 5100, 5100, 5100, 5100, 5100, 5100, 5100, 5100, 5100...
10002	5100, 5100, 5200, 5200, 5400, 5400, 5400, 5400
10003	5200, 5200, 5200, 5200, 5400, 5400, 5400, 5400, 5400, 5400...
10004	5200, 5200, 5300, 5300, 5300, 5300, 5300, 5300, 5400, 5400...
10005	5200, 5200, 5300, 5300, 5400, 5400
10006	5200, 5200, 5200, 5200, 5400, 5400, 5400, 5400, 5400, 5400...

• Chiffres d’Affaire:
Liste les Numéros d’Article par Client:
Chaque client peut avoir commandé le même article plusieurs fois, c’est à dire chaque article peut être **affiché plusieurs fois**

```
Select CustNo,
       ListAgg(Distinct Trim(ItemNo), ',') within Group (Order By ItemNo)
from Sales
```

CUSTNO	ITEMS
00002	
10001	5100, 5200, 5300
10002	5100, 5200, 5400
10003	5200, 5400
10004	5200, 5300, 5400
10005	5200, 5300, 5400
10006	5200, 5400

• Chiffres d’Affaire:
Liste les Numéros d’Article par Client:
En spécifiant **DISTINCT** dans la fonction d’agrégat LISTAGG(), chaque article n'est affiché qu'**une seule fois**.



Fonction de Tableau SPLIT() dans le schéma SYSTOOLS

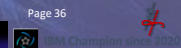
17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 36

IBM i

Université IBM i – 19 et 20 novembre 2024



36

Fonction de Tableau SPLIT()

```
SYSTOOLS.SPLIT(Input_List, Delimiter)
```

La fonction de table SPLIT se trouve dans le schéma **SYSTOOLS**

Renvoie un tableau contenant **une ligne pour chaque élément** de la liste d'entrée

- *Input_List*: Chaîne contenant **la liste des éléments** à déconstruire
Les éléments de la liste d'entrée doivent être **séparés par le délimiteur**
- *Delimiter*: **Séparateur** entre les éléments (peut être plus qu'un caractère)

Le tableau de retour se compose de **2 colonnes**

- **ORDINAL_POSITION** Integer Position relative de l'élément dans la chaîne d'entrée
- **ELEMENT** CLOB(2G) Valeur de l'élément

Note: (*from documentation*) this function is provided in the SYSTOOLS schema as an **Example of how to break a string apart at a delimiting character** by using an SQL table function. **Creating customized versions of this table function to better suit a specific need is encouraged.** Use the Insert Generated SQL feature in ACS to extract the source for this function. Then modify it and create a new function.

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 37

IBM i

Université IBM i – 19 et 20 novembre 2024



37

Fonction de Tableau SPLIT() – Exemples – Découpage d’une série de caractères séparés par des virgules

```
Select *
From Table(SysTools.Split('A, B, C, D, E, V, W, X, Y, Z', ','));
```

ORDINAL_POSITION	ELEMENT
1	A
2	B
3	C
4	D
5	E
6	V
7	W
8	X
9	Y
10	Z

- Découpage d'une chaîne contenant une série de caractères séparés par des virgules

```
With x (MyList) as (Values('A, B, C, X'), ('V, W, X, Y, Z'))
Select MyList, a.*
from x cross join Table(SysTools.Split(MyList, ',')) a
Order By MyList, Ordinal_Position;
```

MYLIST	ORDINAL_POSITION	ELEMENT
A, B, C, X		1 A
A, B, C, X		2 B
A, B, C, X		3 C
A, B, C, X		4 X
V, W, X, Y, Z		1 V
V, W, X, Y, Z		2 W
V, W, X, Y, Z		3 X
V, W, X, Y, Z		4 Y
V, W, X, Y, Z		5 Z

- Découpage de 2 chaînes en lignes:
 1. A, B, C, X
 2. V, W, X, Y, Z
- La Common Table Expression X et la fonction de tableau SPLIT() doivent être reliées par CROSS JOIN



Lecture des fichiers IFS avec Services SQL



Fonctions de Tableau IFS_READ, IFS_READ_BINARY, IFS_READ_UTF8 Lecture des Fichiers situés dans l'IFS

IFS_READ	$\left\{ \begin{array}{l} \text{Path_Name} \\ \text{Maximum_Line_Length} \\ \text{End_Of_Line} \end{array} \right\}$	=> IFSFileName
IFS_READ_BINARY		=> MaximumCharactersPerLine
IFS_READ_UTF8		=> EndOfLineCharacters

Lecture es données d'un fichier IFS (Integrated File System)

- **IFS_READ** les données renvoyées sont converties en **EBCDIC**
- **IFS_READ_BINARY** les données renvoyées **ne sont pas converties**
- **IFS_READ_UTF8** les données renvoyées sont converties en **UTF-8**

Les données peuvent être renvoyées comme **une seule chaîne de caractères** ou être **divisées en plusieurs lignes**.

→ Il dépend de l'indication dans le **paramètre END_OF_LINE**

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 40



IBM i

Université IBM i – 19 et 20 novembre 2024

IBM Champion since 2020

40

Fonctions de Tableau IFS_READ, IFS_READ_BINARY, IFS_READ_UTF8 Lecture des Fichiers situés dans l'IFS

IFS_READ	$\left\{ \begin{array}{l} \text{Path_Name} \\ \text{Maximum_Line_Length} \\ \text{End_Of_Line} \end{array} \right\}$	=> IFSFileName
IFS_READ_BINARY		=> MaximumCharactersPerLine
IFS_READ_UTF8		=> EndOfLineCharacters

Paramètres:

- **PATH_NAME** Fichier IFS → chemin et nom du fichier absolu ou relatif
- **MAXIMUM_LINE_LENGTH** Défaut: 2 GB → > 0 ou pas délivré
- **END_OF_LINE** Retour à la ligne → après le MAXIMUM_LINE_LENGTH des Caractères
ou avec le END_OF_LINE Caractère affiché
END_OF_LINE Caractères sont **supprimés**

Valeurs spéciales: **ANY** (Default)
CR, CRLF, LF, LFCR
NONE (requis pour l'IFS_READ_BINARY)

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 41



IBM i

Université IBM i – 19 et 20 novembre 2024

IBM Champion since 2020

41

Fonctions de Tableau IFS_READ, IFS_READ_BINARY, IFS_READ_UTF8 Lecture des Fichiers situés dans l'IFS

IFS_READ	$\left\{ \begin{array}{l} \text{Path_Name} \\ \text{Maximum_Line_Length} \\ \text{End_Of_Line} \end{array} \right\}$	=> IFSFileName
IFS_READ_BINARY		=> MaximumCharactersPerLine
IFS_READ_UTF8		=> EndOfLineCharacters

Tableau retourné:

- **LINE_NUMBER** Position relative de la ligne dans le fichier IFS
- **LINE**

IFS_READ	CLOB(2GB)	CCSID de l'unité de travail actuel
IFS_READ_BINARY	BLOB(2GB)	sans conversion
IFS_READ_UTF8	CLOB(2GB)	CCSID 1208 = UTF-8

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 42



IBM i

Université IBM i – 19 et 20 novembre 2024



IBM Champion since 2020

42

Lecture d'un Fichier IFS ou d'un membre d'un fichier physique source

```
Select Line_Number, Line
from Table(IFS_READ_UTF8(Path_Name => '/home/Hauser/Examples/WriteWithSQL.txt') a);
```

LINE_NUMBER	LINE
1	Always desire to learn something useful.
2	Success has a simple formula: do your best, and people may like it.

• Lecture du fichier IFS
/home/Hauser/Examples/WriteWithSQL.txt

```
Select Line_Number, Line
from Table(IFS_READ(Path_Name => '/qsys.lib/hsrrrpg.lib/qembsql.file/IFSFILE1.mbr') a);
```

LINE_NUMBER	LINE
1	CTL-Opt DatFmt(*ISO) Option(*NoDebugIO);
2	/If Defined (*CRTBNDRPG)
3	CTL-Opt ActGrp('*NEW');
4	/EndIf
5	//-----
6	// Prototyping / Reference Fields/Data Structures
7	//-----
8	//-----
9	// Global Variables
10	//-----
11	DCL-DS PGMSPS PSDS Qualified;

• Lecture du membre du fichier physique source
HSRRRPG/QEMBSQL(IFSFILE1)

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 43



IBM i

Université IBM i – 19 et 20 novembre 2024



IBM Champion since 2020

43

Lecture d'un Fichier IFS avec SQL - Exemple

```
Select * from Table(IFS_READ_UTF8('/home/Hauser/Employee1.csv')) x;
```

LINE_NUMBER	LINE
1	"EMPLOYEEENO", "NAME", "FIRSTNAME", "ADDRESS", "ZIPCODE", "CITY", "COUNTRY"
2	10, "Meier und Sohn", "", "Industriestr. 3-13", "80333", "Muenchen", "DE"
3	20, "Bauer", "Herrmann", "Wald-und-Wiesen-Weg. 4", "63128", "Dietzenbach", "DE"
4	40, "Hauser", "Birgitta", "Koenigsteiner Allee 59", "63128", "Dietzenbach", "DE"
5	60, "Lehmann", "Maria", "Schwarzwaldstr. 26", "77880", "Sasbach", "DE"
6	50, "Burger", "Emil", "Nelkenweg 21", "86916", "Kaufering", "DE"
7	80, "Miller", "Katrin", "Am Lech 35", "86916", "Kaufering", "DE"

- Lecture du fichier /home/Hauser/Employee1.csv en convertissant le résultat en UTF-8
- Retour à la ligne est effectué à chaque caractère de End_Of_Line Character



Accéder et décomposer un fichier *.csv
situé dans l'IFS



Lecture d'un fichier *.csv avec SQL et division en colonnes - Exemple

```
With x as (Select * from Table(IFS_READ_UTF8('/home/Hauser/Employee1.csv'))),
y as (Select x.*, Ordinal_Position ColKey, Trim(B '' from Element) ColInfo
      from x cross join Table(SysTools.Split(Line, ',')))
Select Line_Number,
      Min(Case When ColKey = 1 Then ColInfo End) EmployeeNo,
      Min(Case When ColKey = 2 Then ColInfo End) Name,
      Min(Case When ColKey = 3 Then ColInfo End) FirstName,
      Min(Case When ColKey = 4 Then ColInfo End) Address,
      Min(Case When ColKey = 5 Then ColInfo End) Country,
      Min(Case When ColKey = 6 Then ColInfo End) ZipCode,
      Min(Case When ColKey = 7 Then ColInfo End) City
From y
Where Line_Number > 1
Group By Line_Number
Order By Line_Number
```

• Fichier *.csv dans l'IFS

- CTE x: Lire le fichier IFS avec le service IFS_READ_UTF8() et le diviser en lignes
- CTE y: Diviser les lignes en colonnes avec le service SPLIT()
- Final Select: Agréger les colonnes

LINE_NUMBER	EMPLOYEEENO	NAME	FIRSTNAME	ADDRESS	COUNTRY	ZIPCODE	CITY
210		Meier und Sohn		Industriestr. 3-13	80333	Muenchen	DE
320		Bauer	Herrmann	Wald-und-Wiesen-Weg. 4	63128	Dietzenbach	DE
440		Hauser	Birgitta	Koenigsteiner Allee 59	63128	Dietzenbach	DE
560		Lehmann	Maria	Schwarzwaldstr. 26	77880	Sasbach	DE
650		Burger	Emil	Nelkenweg 21	86916	Kaufering	DE
780		Miller	Katrin	Am Lech 35	86916	Kaufering	DE

Vue – Exemple: Lire et diviser un fichier *.csv directement depuis l'IFS

```
With x as (Select * from Table(IFS_READ_UTF8('/home/Hauser/Employee1.csv'))),
y as (Select x.*, Ordinal_Position ColKey,
      Trim(B '' from Element) as ColInfo
      from x cross join Table(SysTools.Split(Line, ',')))
Select Line_Number,
      Min(Case When ColKey = 1 Then ColInfo End) EmployeeNo,
      Min(Case When ColKey = 2 Then ColInfo End) Name,
      Min(Case When ColKey = 3 Then ColInfo End) FirstName,
      Min(Case When ColKey = 4 Then ColInfo End) Address,
      Min(Case When ColKey = 5 Then ColInfo End) ZipCode,
      Min(Case When ColKey = 6 Then ColInfo End) City,
      Min(Case When ColKey = 7 Then ColInfo End) Country
From y
Where Line_Number > 1
Group By Line_Number; Select * from Emplcsvv02;
```

• SQL Vue: traiter directement les données *.csv depuis le fichier IFS /home/Hauser/Employee1.csv

• Traiter les données *.csv à la base de la vue EMPLCSVV02 → Dans la vue: les données sont divisées en colonnes

LINE_NUMBER	EMPLOYEEENO	NAME	FIRSTNAME	ADDRESS	ZIPCODE	CITY	COUNTRY
210		Meier und Sohn		Industriestr. 3-13	80333	Muenchen	DE
320		Bauer	Herrmann	Wald-und-Wiesen-Weg. 4	63128	Dietzenbach	DE
440		Hauser	Birgitta	Koenigsteiner Allee 59	63128	Dietzenbach	DE
560		Lehmann	Maria	Schwarzwaldstr. 26	77880	Sasbach	DE
650		Burger	Emil	Nelkenweg 21	86916	Kaufering	DE
780		Miller	Katrin	Am Lech 35	86916	Kaufering	DE

Variables Globales SQL

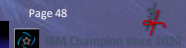
17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 48

IBM i

Université IBM i – 19 et 20 novembre 2024



48

Variables Globales SQL - Qu'est-ce qu'une variable globale SQL ?

Une variable globale SQL est un objet permanent (*SRVPGM) situé dans un schéma

- Une variable globale SQL est créée avec l'instruction SQL **CREATE VARIABLE**
- Une variable globale SQL peut être créée avec n'importe quel type de données SQL (sauf ROWID et DATALINK)
- Une variable globale SQL ne peut pas être créée dans le schéma QTEMP

Le contenu est géré par unité de travail (job / connexion)

- La même variable peut prendre différentes valeurs dans différents jobs / connexions

L'initialisation aura lieu lors de la première utilisation dans un job/connexion

- Peut être à la base de
 - Une valeur constante
 - Un registre spécial
 - Une autre variable globale SQL
 - Une fonction scalaire ou une fonction définie par l'utilisateur (UDF)
 - Une SQL requête qui ne retourne qu'une seule valeur (une seule ligne et une seule colonne)

Peut être utilisée dans les définitions des vues et des triggers

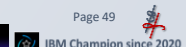
17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 49

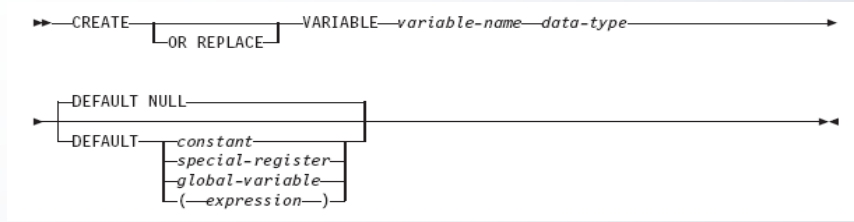
IBM i

Université IBM i – 19 et 20 novembre 2024



49

Variable Globale SQL



Instruction SQL: CREATE OR REPLACE VARIABLE

- Le nom de la variable (128 caractères maximum) doit être indiqué
- Le type SQL de la variable (sauf ROWID et DATALINK) doit être indiqué
- La valeur par défaut peut être indiquée (sans indication => valeur NULL)



Variable Globale SQL – Créer de Variables Globales nouvelles - Exemples

```

Create Variable HSCCOMMON10.GblBossIdFix Integer
Default 101;

Create Variable HSCCOMMON10.GblBossId Integer
Default (Select Employee
        from Staff
        Where Boss = 0
        Fetch First Row Only);
    
```

• GblBossIdFix:	Variable globale au type Integer La valeur par défaut = 101
• GblBossId:	Variable globale au type Integer La valeur par défaut est déterminée à la base d'une requête SQL → Employé sans employé supérieur



Utilisation de variables globales SQL dans des vues

```
Create Variable HSRRRPG.GblIFSFile VarChar(1024) Default '';  
  
Create Or Replace View HSRRRPG.SALESIFSV01 as  
With x as (Select * from Table(IFS_READ_UTF8(GblIFSFile))),  
y as (Select x.*, Ordinal_Position ColKey,  
Trim(B '' from Element) as ColInfo  
from x cross join Table(SysTools.Split(Line, ',')))  
Select Line_Number,  
Min(Case When ColKey = 1 Then Trim(ColInfo) End) CustNo,  
Min(Case When ColKey = 2 Then Trim(ColInfo) End) ItemNo,  
Min(Case When ColKey = 3 Then Trim(ColInfo) End) Descript,  
Min(Case When ColKey = 4 Then Date(ColInfo) End) SalesDate,  
Min(Case When ColKey = 5 Then Dec(ColInfo, 11, 2) End) Amount  
From y  
Where Line_Number > 1  
Group By Line_Number;
```

```
Set GblIFSFile = '/home/Hauser/Sales2015.csv';  
Select * from SALESIFSV01;
```

LINE_NUMBER	CUSTNO	ITEMNO	DESCRIPT	SALESDATE	AMOUNT
2	10001	BK5100	King,Stephen - Es	2015-01-04	140,97
3	10006	BK5400	King,Stephen - Shining	2015-01-29	915,75
4	10006	BK5400	King,Stephen - Shining	2015-02-08	550,00
5	10006	BK5400	King,Stephen - Shining	2015-03-15	550,00
6	10006	BK5200	King,Stephen - Drei	2015-09-04	495,00
7	10006	BK5400	King,Stephen - Shining	2015-08-26	742,50
8	10006	BK5400	King,Stephen - Shining	2015-08-19	3368,75
9	10006	BK5400	King,Stephen - Shining	2015-09-08	122,23
10	10006	BK5400	King,Stephen - Shining	2015-07-13	698,50
11	10006	BK5400	King,Stephen - Shining	2015-08-18	698,50
12	10006	BK5200	King,Stephen - Drei	2015-11-16	628,65
13	10006	BK5400	King,Stephen - Shining	2015-11-07	942,97

- Définition de la variable globale: GblIFSFile
- Définition de la vue SALESIFSV01 contenant la variable globale GblIFSFile

- Modifier la valeur de la SQL variable globale GblIFSFile
- Exécuter la vue SalesIFSV01 qui contient la variable globale SQL GblIFSFile



Questions?

17.11.2024 Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser Page 53 IBM Champion since 2020



References

IBM i information center

- SQL Reference
https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_75/db2/rbafzpdf.pdf?Vue=kc
- PDF Files for Database
https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_75/rzatd/rzatdprintable.htm
- Database Information Finder
https://www.ibm.com/support/knowledgecenter/ssw_ibm_i_75/rzatd/rzatdfinder.htm

IBM i Tutorials, Demos, and SQL Examples

<https://www.ibm.com/support/pages/ibm-i-tutorials-demos-and-sql-Exemples>

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 55

IBM Champion since 2020



Biographie brève: Birgitta Hauser

Birgitta Hauser **Diplom-Betriebswirt (BA)** **Database and Software Architect**

Diplômée en gestion d'entreprise, Birgitta Hauser a d'abord travaillé plusieurs années dans le contrôle de gestion avant de se tourner vers la programmation (RPG) sur AS/400. Aujourd'hui, elle travaille encore quelque fois comme programmeur sur l' IBM i. Son travail se concentre toutefois sur la modernisation et l'optimisation des applications IBM i existantes, en particulier des bases de données, ainsi que sur l'intégration de nouvelles technologies.

Depuis 2020, Birgitta travaille à son compte et assiste ses clients dans des projets de modernisation d'applications et de bases de données et de l'optimisation des performances SQL sur l'IBM i et Db2 for i.

De plus, Birgitta donne régulièrement des cours pour des programmeurs IBM i (RPG/CL) et des spécialiste Db2 for i et des utilisateurs SQL.

Depuis 2002, Birgitta intervient régulièrement lors de conférences des COMMON User Groups en Allemagne, dans d'autres pays européens, ainsi qu'aux États-Unis et au Canada.

Birgitta est co-auteur de 2 IBM Redbooks, ainsi que de plusieurs articles spécialisés pour IBM DeveloperWorks and IT-Jungle. Elle écrit régulièrement des articles spécialisés (RPG/SQL) pour le ITP-Verlag (daison d'édition allemande)

En 2015, Birgitta a reçu la bourse d'études John Earl Speaker. En 2018, elle a reçu la bourse d'études commémorative Al Barsa.

Depuis 2020 elle est un IBM Champion.

17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 56

IBM Champion since 2020



Un grand merci à

Holger Scherer – RZKH Rechenzentrum Kreuznach

- Pour la mise à disposition d'un système IBM i permettant la création des exemples, c'est-à-dire du code utilisé dans mes présentations.
- <http://www.rzkh.de>



• Your data is save! ... in the bunker

Natanaël Bonnet – Gaia / Volubis

- Pour avoir révisé mes slides en français



17.11.2024

Université IBM i 2024 - S22 - Regrouper, accumuler et puis diviser à nouveau avec SQL - Birgitta Hauser

Page 57

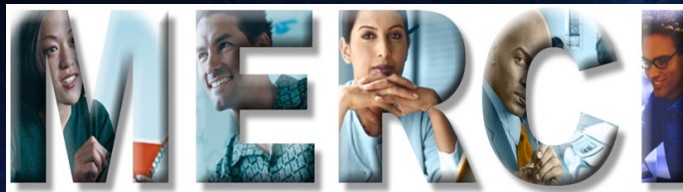


IBM i

Université IBM i – 19 et 20 novembre 2024

IBM Champion since 2020

57



Regrouper, accumuler et puis diviser à nouveau avec SQL
Yes, i can!

Si vous êtes intéressé par des classes individuelles plus détaillées, sur place ou à distance, veuillez me contacter directement

Birgitta Hauser – Modernization – Education – Consulting on IBM i
Diplom-Betriebswirt (BA)
Database and Software Architect
IBM Champion since 2020

Hauser@ModEdCon.com / Hauser@SSS-Software.de
<https://ModEdCon.com>



IBM i

Université IBM i – 19 et 20 novembre 2024

IBM Champion since 2020

58