

SQL BITS

LISTAGG and LATERAL JOIN

LISTAGG – IBM

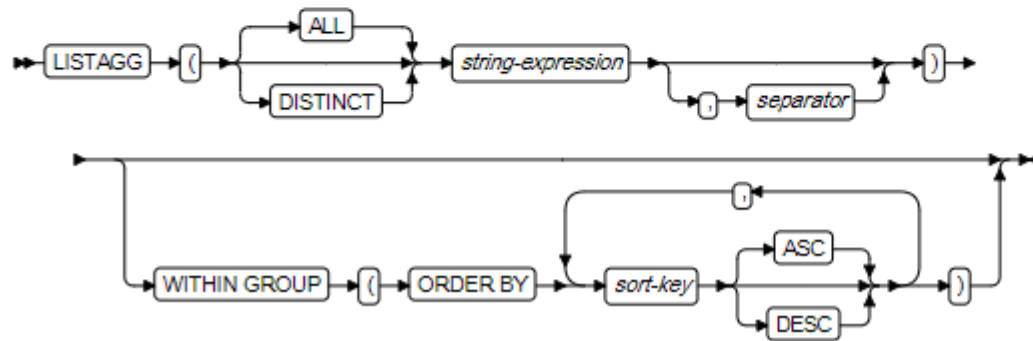
- <https://www.ibm.com/support/knowledgecenter/en/SS6NHC/com.ibm.swg.im.dashdb.sql.ref.doc/doc/r0058709.html>

LISTAGG aggregate function

☰ Table of contents

🔗 Change product

The LISTAGG function aggregates a set of string elements into one string by concatenating the strings. Optionally, a separator string can be provided which is inserted between contiguous input strings.



LISTAGG – simple example

```

7 SELECT WORKDEPT,
8     LISTAGG(LASTNAME, ', ') WITHIN GROUP(ORDER BY LASTNAME)
9     AS EMPLOYEES
10 FROM   employee
11
12 GROUP BY WORKDEPT;

```

SELECT * FROM employee ORDER BY workDept, lastName

File Edit View

EMPNO	FIRSTNME	MIDINIT	LASTNAME	WORKDEPT	PHONENO	HIREDATE	JOB	EDLEVEL	SEX	BIRTHDATE	SALARY	BONUS	COMM
000010	CHRISTINE	I	HAAS	A00	3978	1965-01-01	PRES		18 F	1933-08-24	63827.50	1000.00	4220.00
200010	DIAN	J	HEMMINGER	A00	3978	1965-01-01	SALESREP		18 F	1933-08-14	46500.00	1000.00	4220.00
000110	VINCENZO	G	LUCCHESSI	A00	3490	1958-05-16	SALESREP		19 M	1929-11-05	46500.00	900.00	3720.00
000120	SEAN		ODCONNELL	A00	2167	1963-12-05	CLERK		14 M	1942-10-18	29250.00	600.00	2340.00
200120	GREG		ORLANDO	A00	2167	1972-05-05	CLERK		14 M	1942-10-18	29250.00	600.00	2340.00
000020	MICHAEL	L	THOMPSON	B01	3476	1973-10-10	MANAGER		18 M	1948-02-02	49912.50	800.00	3300.00
000030	SALLY	A	KWAN	C01	4738	1975-04-05	MANAGER		20 F	1941-05-11	46282.50	800.00	3060.00
200140	KIM	N	NATZ	C01	1793	1976-12-15	ANALYST		18 F	1946-01-19	28420.00	600.00	2274.00
000140	HEATHER	A	NICHOLLS	C01	1793	1976-12-15	ANALYST		18 F	1946-01-19	28420.00	600.00	2274.00
000130	DOLORES	M	QUINTANA	C01	4578	1971-07-28	ANALYST		16 F	1925-09-15	23800.00	500.00	1904.00
000170	MASATOSHI	J	YOSHIMURA	D11	2890	1978-09-15	DESIGNER		16 M	1951-01-05	24680.00	500.00	1974.00
000150	BRUCE		ADAMSON	D11	4510	1972-02-12	DESIGNER		16 M	1947-05-17	25280.00	500.00	2022.00
000200	DAVID		BROWN	D11	4501	1966-03-03	DESIGNER		16 M	1941-05-29	27740.00	600.00	2217.00
300001	JOHN	A	HIRKO	D11	1234	2017-05-11	CLERK		12 M	1970-01-01	25000.00	1000.00	2500.00
200220	REBA	K	JOHN	D11	672	1968-08-29	DESIGNER		18 F	1948-03-19	29840.00	600.00	2387.00
000210	WILLIAM	T	JONES	D11	942	1979-04-11	DESIGNER		17 M	1953-02-23	18270.00	400.00	1462.00
000220	JENNIFER	K	LUTZ	D11	672	1968-08-29	DESIGNER		18 F	1948-03-19	29840.00	600.00	2387.00
000160	ELIZABETH	R	PINKA	D11	3782	1977-10-11	DESIGNER		17 F	1955-04-12	22250.00	400.00	1780.00
000180	MARILYN	S	SCOUTTEN	D11	1682	1973-07-07	DESIGNER		17 F	1949-02-21	21340.00	500.00	1707.00
000060	IRVING	F	STERN	D11	6423	1973-09-14	MANAGER		16 M	1945-07-07	39022.50	500.00	2580.00
000190	JAMES	H	WALKER	D11	2986	1974-07-26	DESIGNER		16 M	1952-06-25	20450.00	400.00	1636.00
200170	KIYOSHI		YAMAMOTO	D11	2890	1978-09-15	DESIGNER		16 M	1951-01-05	24680.00	500.00	1974.00
000230	JAMES	J	JEFFERSON	D21	2094	1966-11-21	CLERK		14 M	1935-05-30	22180.00	400.00	1774.00

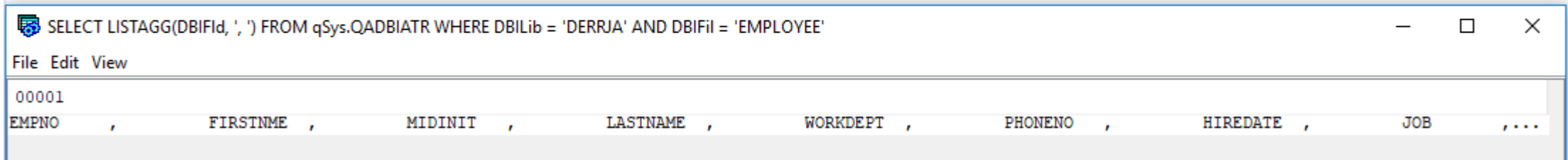
SELECT WORKDEPT, LISTAGG(LASTNAME, ', ') WITHIN GROUP(ORDER BY LASTNAME) AS EMPLOYEES FROM employee ...

File Edit View

WORKDEPT	EMPLOYEES
A00	HAAS, HEMMINGER, LUCCHESSI, ODCONNELL, ORLANDO
B01	THOMPSON
C01	KWAN, NATZ, NICHOLLS, QUINTANA
D11	YOSHIMURA, ADAMSON, BROWN, HIRKO, JOHN, JONES, ...
D21	JEFFERSON, JOHNSON, MARINO, MONTEVERDE, PEREZ, P...
E01	GEYER
E11	HENDERSON, PARKER, SCHNEIDER, SCHWARTZ, SETRIGHT...
E21	ALONZO, GOUNOT, LEE, MEHTA, SPENSER, WONG

LISTAGG – create comma separated

```
SELECT LISTAGG(DBIFld, ', ')
FROM qSys.QADBIATR
WHERE DBILib = 'DERRJA'
AND DBIFil = 'EMPLOYEE' ;
```

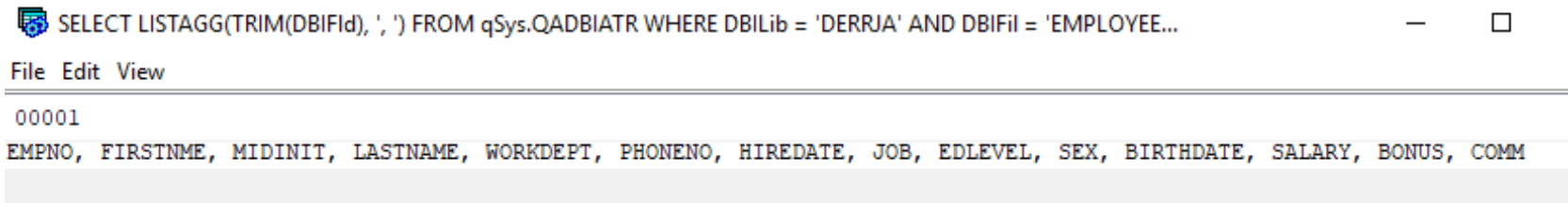


```
SELECT LISTAGG(DBIFld, ', ') FROM qSys.QADBIATR WHERE DBILib = 'DERRJA' AND DBIFil = 'EMPLOYEE'
```

File Edit View

```
00001
EMPNO , FIRSTNME , MIDINIT , LASTNAME , WORKDEPT , PHONENO , HIREDATE , JOB , ...
```

```
SELECT LISTAGG(TRIM(DBIFld), ', ')
FROM qSys.QADBIATR
WHERE DBILib = 'DERRJA'
AND DBIFil = 'EMPLOYEE' ;
```



```
SELECT LISTAGG(TRIM(DBIFld), ', ') FROM qSys.QADBIATR WHERE DBILib = 'DERRJA' AND DBIFil = 'EMPLOYEE...'
```

File Edit View

```
00001
EMPNO, FIRSTNME, MIDINIT, LASTNAME, WORKDEPT, PHONENO, HIREDATE, JOB, EDLEVEL, SEX, BIRTHDATE, SALARY, BONUS, COMM
```

LISTAGG – modified “actual” business case

- BUSINESS CASE was to show a list of account data per household. Data was required in an array-type view.

```

17 SELECT      BUD.household
18             ,LISTAGG(BUD.officers, ', ')              WITHIN GROUP(ORDER BY BUD.officers)          AS officers
19             ,LISTAGG(BUD.account || ':' || BUD.acType, ', ') WITHIN GROUP(ORDER BY BUD.account || ':' || BUD.acType) AS type
20             ,SUM(cBal)                               AS CurrentBalance
21 FROM        BUDetail_T BUD
22 GROUP BY   BUD.Household
23 HAVING     SUM(cBal) >= 500000;

```

SELECT BUD.household ,LISTAGG(BUD.officers, ', ') WITHIN GROUP(ORDER BY BUD.officers) AS officers ,L...

File Edit View

HOUSEHOLD	OFFICERS	TYPE	CURRENTBALANCE
HEAL...	040 , 040	1066084501:D, 1066092601:D	908575.15
ONET...	040 , 040 , 040 , 040 , 040 , 040 , 040	1016275601:D, 1017647101:D, 1017649801:D, 103351...	5302469.18
POBO...	040 , 040 , 040	100001901:D, 1019816501:D, 1029309501:D	3155413.28
POBO...	LNS , LNS , LNS	3303028801:D, 3303028802:D, 3303028809:S	777304.01
POBO...	013 , 040 , 040 , 040 , 040 , 040 , 040 , 040 ...	1029263301:D, 1029270601:D, 1029285401:D, 102929...	7064145.54
POBO...	040	112758201:D	3885413.62
POBO...	018 , 018 , 018 , 018 , 018 , 018	1522561901:D, 1522561901:D, 1522561910:S, 330491...	737650.57
POBO...	CAP , 001 , 007 , 007	144353001:D, 7000357926:T, 7011370609:S, 7012425...	558429.80

LATERAL – simple

```
SELECT      d.deptno,  
            d.deptname,  
            empinfo.avgсал,  
            empinfo.empcount  
FROM        department d,  
LATERAL    (SELECT      AVG(e.salary) AS avgсал,  
              COUNT(*) AS empcount  
            FROM        employee e  
            WHERE       e.workdept=d.deptno) AS empinfo;
```

From <https://www.ibm.com/support/knowledgecenter/SSGU8G_12.1.0/com.ibm.sqls.doc/ids_sql_0254.htm?view=embed>

LATERAL JOIN

Product

```
CREATE TABLE DerrTest.Product
(
  productId      CHAR(20)
, listPrice     DECIMAL(10, 2)
);
INSERT INTO DerrTest.Product
VALUES
  ('TOOL_SMALL'      , 10.00)
, ('TOOL_MED'       , 20.00)
, ('TOOL_LARGE'     , 30.00)
, ('TOOL_HUGE'      , 60.00)
, ('WIDGET_SMALL'   , 100.00)
, ('WIDGET_MED'     , 200.00)
, ('WIDGET_LARGE'   , 300.00)
, ('WIDGET_HUGE'    , 800.00)
, ('TUIT_SMALL'     , 27.00)
, ('TUIT_MED'       , 38.00)
, ('TUIT_LARGE'     , 49.00)
, ('TUIT_HUGE'      , 70.00) ;
```


SalesOrderDetail

```
CREATE TABLE DerrTest.SalesOrderDetail
(
  orderNumber    INT
, orderDate      DATE
, productId      CHAR(20)
, orderQty       INT
, unitPrice      DECIMAL(10 , 2)
, unitPriceDiscount DECIMAL( 5 , 4)
);
INSERT INTO DerrTest.SalesOrderDetail
VALUES
  ( 1, '2019-01-10', 'WIDGET_SMALL', 1, 100.00, 0)
, ( 1, '2019-01-10', 'WIDGET_MED', 2, 200.00, .1000)
, ( 1, '2019-01-10', 'WIDGET_LARGE', 3, 300.00, .1500)
, ( 1, '2019-01-10', 'TOOL_SMALL', 3, 10.00, .0050)
, ( 1, '2019-01-10', 'TOOL_MED', 2, 20.00, .0100)
, ( 1, '2019-01-10', 'TOOL_LARGE', 1, 30.00, .0200)
, ( 2, '2019-01-20', 'WIDGET_MED', 3, 200.00, .1000)
, ( 2, '2019-01-20', 'WIDGET_MED', 3, 200.00, .1000)
, ( 2, '2019-01-20', 'WIDGET_MED', 3, 200.00, .1000)
, ( 3, '2019-02-10', 'WIDGET_SMALL', 1, 100.00, 0)
, ( 3, '2019-02-11', 'WIDGET_MED', 2, 200.00, .1000)
, ( 3, '2019-02-11', 'WIDGET_LARGE', 3, 300.00, .1500)
, ( 4, '2019-01-11', 'WIDGET_SMALL', 4, 100.00, 0)
, ( 4, '2019-01-11', 'WIDGET_MED', 3, 200.00, .1000)
, ( 4, '2019-01-11', 'WIDGET_LARGE', 2, 300.00, .1500)
, ( 4, '2019-01-11', 'TOOL_SMALL', 1, 10.00, .0050)
, ( 5, '2019-01-12', 'TOOL_MED', 11, 20.00, .0100)
, ( 5, '2019-01-12', 'TOOL_LARGE', 2, 30.00, .0200)
, ( 6, '2019-01-22', 'WIDGET_MED', 10, 200.00, .1000)
, ( 6, '2019-01-22', 'WIDGET_MED', 100, 200.00, .1000)
, ( 6, '2019-01-22', 'WIDGET_MED', 7, 200.00, .1000)
, ( 7, '2019-02-13', 'WIDGET_SMALL', 6, 100.00, 0)
, ( 8, '2019-02-13', 'WIDGET_MED', 5, 200.00, .1000)
, ( 8, '2019-02-13', 'WIDGET_LARGE', 4, 300.00, .1500);
```

LATERAL JOIN

<https://www.itjungle.com/2016/08/02/fhg080216-story03/>

```
101 SELECT SOD.*
102        ,calcs.ExtPrice
103        ,CAST(calcs.DiscountPrice AS DECIMAL(10 , 2)) AS discountPrice , p.*
104 FROM SalesOrderDetail sod
105 CROSS JOIN LATERAL (
106        VALUES(sod.OrderQty*sod.UnitPrice,
107               sod.OrderQty*sod.UnitPrice * (1-sod.UnitPriceDiscount))) calcs (ExtPrice,DiscountPrice)
108 JOIN Product p
109 ON p.ProductId = sod.ProductId
110 AND calcs.ExtPrice >= calcs.DiscountPrice --<<< Simply to demonstrate the repetition
111 WHERE ExtPrice-DiscountPrice > 10.00
112 ORDER BY ExtPrice-DiscountPrice DESC;
```

 SELECT SOD.* ,calcs.ExtPrice ,CAST(calcs.DiscountPrice AS DECIMAL(10, 2)) AS discountPrice , p.* FR...

File Edit View

ORDERNUMBER	ORDERDATE	PRODUCTID	ORDERQTY	UNITPRICE	UNITPRICEDISCOUNT	EXTPRICE	DISCOUNTPRICE	PRODUCTID	LISTPRICE
6	2019-01-22	WIDGET_MED	100	200.00	0.1000	20000.00	18000.00	WIDGET_MED	200.00
6	2019-01-22	WIDGET_MED	10	200.00	0.1000	2000.00	1800.00	WIDGET_MED	200.00
8	2019-02-13	WIDGET_LARGE	4	300.00	0.1500	1200.00	1020.00	WIDGET_LARGE	300.00
6	2019-01-22	WIDGET_MED	7	200.00	0.1000	1400.00	1260.00	WIDGET_MED	200.00
1	2019-01-10	WIDGET_LARGE	3	300.00	0.1500	900.00	765.00	WIDGET_LARGE	300.00
3	2019-02-11	WIDGET_LARGE	3	300.00	0.1500	900.00	765.00	WIDGET_LARGE	300.00
8	2019-02-13	WIDGET_MED	5	200.00	0.1000	1000.00	900.00	WIDGET_MED	200.00
4	2019-01-11	WIDGET_LARGE	2	300.00	0.1500	600.00	510.00	WIDGET_LARGE	300.00
2	2019-01-20	WIDGET_MED	3	200.00	0.1000	600.00	540.00	WIDGET_MED	200.00
2	2019-01-20	WIDGET_MED	3	200.00	0.1000	600.00	540.00	WIDGET_MED	200.00
2	2019-01-20	WIDGET_MED	3	200.00	0.1000	600.00	540.00	WIDGET_MED	200.00
4	2019-01-11	WIDGET_MED	3	200.00	0.1000	600.00	540.00	WIDGET_MED	200.00
1	2019-01-10	WIDGET_MED	2	200.00	0.1000	400.00	360.00	WIDGET_MED	200.00
3	2019-02-11	WIDGET_MED	2	200.00	0.1000	400.00	360.00	WIDGET_MED	200.00

ANALYSIS

Used in this way, the calculated fields "Calcs.ExtPrice" and "Calcs.DiscountPrice" are defined once, and used multiple times. While confusing, it avoids redundancy.

"Calcs" is an ALIAS of the VALUES expression, which returns two fields: ExtPrice and DiscountPrice. This LATERAL is actually a CROSS JOIN (cartesian), but the single row in the right side of the join is recalculated for every row on the left!

This behaves as if the columns "Calcs.ExtPrice" and "Calcs.DiscountPrice" exists in SOD.

```
SELECT SOD.*
   ,calcs.ExtPrice
   ,CAST(calcs.DiscountPrice AS DECIMAL(10 ,2)) AS discountPrice, p.*
FROM SalesOrderDetail sod

CROSS JOIN LATERAL (
  VALUES(sod.OrderQty * sod.UnitPrice,
         sod.OrderQty * sod.UnitPrice * (1-sod.UnitPriceDiscount)) calcs
  (ExtPrice DiscountPrice)

  JOIN Product p ON p.ProductId=sod.ProductId
                 AND p.ListPrice<calcs.DiscountPrice
  WHERE ExtPrice-DiscountPrice>10.00
  ORDER BY ExtPrice-DiscountPrice DESC
```

|
Analysis:
The expression

```
CROSS JOIN LATERAL (
  VALUES(sod.OrderQty * sod.UnitPrice,
         sod.OrderQty * sod.UnitPrice * (1-sod.UnitPriceDiscount)) calcs
  (ExtPrice DiscountPrice)
```

Is calculated for EVERY SOD. row

- The VALUES returns two columns
- VALUES expression is aliased to the name Calcs