



**EBook Gratis**

# APRENDIZAJE

## db2

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#db2**

# Tabla de contenido

Acerca de.....	1
<b>Capítulo 1: Empezando con db2.....</b>	<b>2</b>
Observaciones.....	2
Examples.....	2
Instalación.....	2
<b>Capítulo 2: Analizar los valores separados por comas en una columna en varias filas.....</b>	<b>3</b>
Sintaxis.....	3
Examples.....	3
Consulta recursiva para analizar valores separados por comas.....	3
<b>Capítulo 3: Combine varias filas en un solo valor separado por comas.....</b>	<b>6</b>
Examples.....	6
Usando la función listagg ().....	6
<b>Capítulo 4: Copia tabla con o sin datos.....</b>	<b>7</b>
Introducción.....	7
Sintaxis.....	7
Examples.....	7
Copiar tabla con datos.....	7
Copiar tabla sin datos.....	7
Copiar tabla con la cláusula where.....	7
<b>Capítulo 5: dashDB.....</b>	<b>8</b>
Introducción.....	8
Observaciones.....	8
<b>Para obtener más información sobre dashDB, puede probar estas fuentes:.....</b>	<b>8</b>
Información general:.....	8
Más documentación.....	8
Examples.....	8
Una selección básica de SQL: enumera todas las filas de una tabla.....	8
<b>Capítulo 6: Donde no en columnas múltiples.....</b>	<b>9</b>
Examples.....	9
Filtra múltiples combinaciones de valores.....	9

<b>Capítulo 7: Hola Mundo;</b> .....	<b>10</b>
Examples.....	10
Creando una base de datos en DB2.....	10
Conectando a una base de datos en DB2.....	10
Crear una tabla en DB2 llamada "empleado".....	10
Insertando una fila en una tabla de DB2.....	11
muestra de consulta de selección;.....	11
<b>Capítulo 8: Información de restricción de DB2</b> .....	<b>13</b>
Introducción.....	13
Examples.....	13
Obtener restricciones basadas en el nombre de la columna.....	13
Obtener información de restricciones basada en el nombre de la restricción.....	13
<b>Capítulo 9: Insertar en una tabla seleccionando de otra tabla</b> .....	<b>15</b>
Examples.....	15
Insertar en una tabla seleccionando de otra tabla.....	15
<b>Capítulo 10: Mesa plegable</b> .....	<b>16</b>
Introducción.....	16
Examples.....	16
Sintaxis básica de la tabla de caída.....	16
<b>Creditos</b> .....	<b>17</b>

---

## Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [db2](#)

It is an unofficial and free db2 ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official db2.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# Capítulo 1: Empezando con db2

## Observaciones

Esta sección proporciona una descripción general de qué es db2 y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema importante dentro de db2 y vincular a los temas relacionados. Dado que la Documentación para db2 es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

## Examples

### Instalación

Hay diferentes sabores de DB2. Uno de ellos es LUW: Linux, UNIX y Windows.

DB2 LUW en Linux / UNIX se puede instalar con o sin root. Cuando se instala con root, puede crear diferentes instancias asociándolas a diferentes usuarios.

Al instalar DB2 LUW sin privilegios de root, puede instalar DB2 en su directorio de inicio y su usuario será automáticamente la única instancia que puede tener esta instalación. La instancia no se iniciará automáticamente cada vez que la máquina arranque, al menos usted configura eso.

Una vez que haya descargado los binarios (desde Fixpack Central, Passport Advantage o un CD), puede extraer los archivos. Habrá un directorio que describe la edición de DB2 que usa (Expc, server\_t, etc.), y en ese directorio encontrará `db2setup` y `db2_install`. Una de ellas es para la instalación gráfica, la otra es para la instalación de texto a través del archivo de respuestas.

Lea [Empezando con db2 en línea](https://riptutorial.com/es/db2/topic/5617/empezando-con-db2): <https://riptutorial.com/es/db2/topic/5617/empezando-con-db2>

# Capítulo 2: Analizar los valores separados por comas en una columna en varias filas

## Sintaxis

- CON *CTE\_name* (*COLUMN\_NAME* [...]) AS (
- SELECCIONAR *column\_name* [, ...] FROM *base\_table*
- UNION TODO
- SELECCIONAR *column\_name* [, ...] FROM *CTE\_name*
- DONDE < *condición limitante de recursión* >
- )
- SELECCIONAR *column\_name* [, ...] FROM *CTE\_name*

## Examples

### Consulta recursiva para analizar valores separados por comas

Aunque almacenar múltiples valores en una sola columna viola las reglas de normalización, a veces uno tiene que lidiar con tablas heredadas mal diseñadas. Una consulta recursiva puede ayudar a convertir valores separados por comas en filas distintas.

Cree una tabla mal diseñada de muestra e inserte algunos datos:

```
create table projects (name varchar(10), members varchar(1000));  
  
insert into projects (name, members) values ('Luna', '1, 3, 4'), ('Terra', '2,3,5');
```

Compruebe lo que tenemos:

```
select * from projects;
```

dará salida

```
NAME          MEMBERS  
-----  
Luna          1, 3, 4  
Terra         2,3,5  
  
2 record(s) selected.
```

Use una expresión de tabla común (CTE) para extraer recursivamente cada valor separado por comas de los `MEMBERS` en su propia fila:

```
WITH parse (lvl, name, member, tail) AS (  
  SELECT 1, name,
```

```

CASE WHEN LOCATE(',',members) > 0
      THEN TRIM(LEFT(members, LOCATE(',',members)-1))
      ELSE TRIM(members)
END,
CASE WHEN LOCATE(',',members) > 0
      THEN SUBSTR(members, LOCATE(',',members)+1)
      ELSE ''
END
FROM projects
UNION ALL
SELECT lvl + 1, name,
       CASE WHEN LOCATE(',', tail) > 0
            THEN TRIM(LEFT(tail, LOCATE(',', tail)-1))
            ELSE TRIM(tail)
       END,
       CASE WHEN LOCATE(',', tail) > 0
            THEN SUBSTR(tail, LOCATE(',', tail)+1)
            ELSE ''
       END
FROM parse
WHERE lvl < 100 AND tail != ''
SELECT name, integer(member) member FROM parse
ORDER BY 1

```

## volverá

NAME	MEMBER
Luna	1
Luna	3
Luna	4
Terra	2
Terra	3
Terra	5

6 record(s) selected.

El resultado devuelto por el CTE se puede utilizar como una tabla regular, por ejemplo, uniéndolo a otra tabla. Por ejemplo, cree una tabla de búsqueda de empleados:

```

create table employees (id integer, name varchar(20));
insert into employees (id, name) values (1, 'John'), (2, 'Peter'),
                                       (3, 'Venkat'), (4, 'Mishka'), (5, 'Xiao');

```

## Entonces la siguiente consulta

```

WITH parse (lvl, name, member, tail) AS (
  SELECT 1, name,
         CASE WHEN LOCATE(',',members) > 0
              THEN TRIM(LEFT(members, LOCATE(',',members)-1))
              ELSE TRIM(members)
         END,
         CASE WHEN LOCATE(',',members) > 0
              THEN SUBSTR(members, LOCATE(',',members)+1)
              ELSE ''
         END
  FROM projects
)

```

```

UNION ALL
SELECT lvl + 1, name,
       CASE WHEN LOCATE(',', tail) > 0
            THEN TRIM(LEFT(tail, LOCATE(',', tail)-1))
            ELSE TRIM(tail)
       END,
       CASE WHEN LOCATE(',', tail) > 0
            THEN SUBSTR(tail, LOCATE(',', tail)+1)
            ELSE ''
       END
FROM parse
WHERE lvl < 100 AND tail != ''
SELECT p.name "Project name", e.name "Member name"
FROM parse p
INNER JOIN employees e
ON e.id = integer(p.member)
ORDER BY 1, 2

```

volverá

```

Project name Member name
-----
Luna         John
Luna         Mishka
Luna         Venkat
Terra        Peter
Terra        Venkat
Terra        Xiao

6 record(s) selected.

```

Lea [Analizar los valores separados por comas en una columna en varias filas en línea:](https://riptutorial.com/es/db2/topic/6751/analizar-los-valores-separados-por-comas-en-una-columna-en-varias-filas)  
<https://riptutorial.com/es/db2/topic/6751/analizar-los-valores-separados-por-comas-en-una-columna-en-varias-filas>



---

# Capítulo 3: Combine varias filas en un solo valor separado por comas

## Examples

### Usando la función listagg ()

Digamos que tiene una tabla de préstamos y otra tabla relacionada de parcelas, donde cada préstamo puede tener una o más parcelas asociadas. Si desea que una consulta muestre cada préstamo y una lista de todas sus parcelas asociadas, pero solo desea que cada préstamo aparezca una vez, entonces podría usar algo como esto:

```
select
  loan.loannumber,
  parcel_agg.p_list as parcel_list
from
  schema.loan loan
  left join
    ( select loannumber, listagg(parcelnum, ', ') from schema.parcel parcel group by loannumber
    ) parcel_agg on parcel_agg.loannumber = loan.loannumber
```

Lea [Combine varias filas en un solo valor separado por comas en línea](https://riptutorial.com/es/db2/topic/5940/combine-varias-filas-en-un-solo-valor-separado-por-comas):

<https://riptutorial.com/es/db2/topic/5940/combine-varias-filas-en-un-solo-valor-separado-por-comas>

---

# Capítulo 4: Copia tabla con o sin datos

## Introducción

Ejemplo de cómo copiar la estructura de tabla existente con / sin datos

## Sintaxis

1. CREAR TABLA schemaName.table AS (SELECCIONAR columnas DE schemaName.table) CON DATOS

## Examples

### Copiar tabla con datos

```
CREATE TABLE myschema.tableNew AS (  
    SELECT *  
    FROM myschema.tableOld  
) WITH DATA
```

### Copiar tabla sin datos

```
CREATE TABLE myschema.tableNew AS (  
    SELECT *  
    FROM myschema.tableOld  
) WITHOUT DATA
```

### Copiar tabla con la cláusula where

```
CREATE TABLE myschema.tableNew AS (  
    SELECT *  
    FROM myschema.tableOld  
    WHERE column1 = 'myCriteria'  
) WITH DATA
```

Lea Copia tabla con o sin datos en línea: <https://riptutorial.com/es/db2/topic/8778/copia-tabla-con-o-sin-datos>

---

# Capítulo 5: dashDB

## Introducción

IBM dashDB es una familia de bases de datos SQL. Su motor es una mezcla de DB2, Netezza y el [motor en memoria BLU](#) . Cualquier controlador DB2 soportado funcionará; es un reemplazo para DB2.

Las 3 líneas de productos para dashDB son: (1) IBM dashDB for Transactions: Una base de datos SQL en la nube totalmente administrada y de propósito general. (2) IBM dashDB for Analytics: un almacén de datos en la nube SQL. (3) IBM dashDB Local: una versión local que puede instalar en su propio hardware.

## Observaciones

---

# Para obtener más información sobre dashDB, puede probar estas fuentes:

## Información general:

- Sitio web principal: <http://dashdb.com/>
- [dashDB para Transacciones Bluemix Page](#)
- [DashDB for Analytics Bluemix Page](#)

## Más documentación

- [DashDB docs en Bluemix](#)
- [IBM Knowledge Center para dashDB](#)
- [Referencia de API para dashDB para Analytics](#)

## Examples

Una selección básica de SQL: enumera todas las filas de una tabla.

```
SELECCIONAR * DE MI_TABLE
```

Lea dashDB en línea: <https://riptutorial.com/es/db2/topic/8673/dashdb>

---

# Capítulo 6: Donde no en columnas múltiples

## Examples

### Filtra múltiples combinaciones de valores

Digamos que desea filtrar una consulta por dos columnas, pero solo ciertas combinaciones de esas columnas. Por ejemplo, está bien tener la cuenta 60400 con la referencia JE, pero no puede tener la cuenta 60400 con la referencia ED, pero puede tener la cuenta 60500 con la referencia ED.

```
select * from schema.table where (acct, ref) not in
( values
  (60400, 'ED'),
  (60600, 'ED'),
  (60701, 'ED'),
  (70400, 'ED'),
  (70500, 'ED'),
  (70600, 'ED'),
  (80800, 'ED')
);
```

Lea Donde no en columnas múltiples en línea: <https://riptutorial.com/es/db2/topic/6349/donde-no-en-columnas-multiples>

# Capítulo 7: Hola Mundo;

## Examples

### Creando una base de datos en DB2

```
CREATE DATABASE SAMPLEDB;
```

Esto creará una nueva base de datos llamada sampledb.

### Conectando a una base de datos en DB2

```
CONNECT TO SAMPLEDB;
```

Desde la línea de comando (db2clp, terminal, db2cmd) puede escribir:

```
db2 CONNECT TO SAMPLEDB
```

### Crear una tabla en DB2 llamada "empleado"

La siguiente declaración creará una nueva tabla llamada empleado:

```
CREATE TABLE EMPLOYEE (  
    EMPNO      CHAR(6)          NOT NULL,  
    FIRSTNME   VARCHAR(12)     NOT NULL,  
    LASTNAME   VARCHAR(15)     NOT NULL,  
    SALARY     DECIMAL(9,2)      ,  
    PRIMARY KEY (EMPNO)  
)
```

Esto creará una nueva tabla llamada empleado. La tabla tendrá una clave primaria en la columna EMPNO . Las tres primeras columnas no pueden tener un valor nulo y son texto. El cuarto puede tener nulos y es un número.

Puede crear esta tabla desde db2clp (Linux, UNIX, MacOS) de esta manera (encerrando la declaración entre comillas):

```
db2 "CREATE TABLE EMPLOYEE (  
    EMPNO      CHAR(6)          NOT NULL,  
    FIRSTNME   VARCHAR(12)     NOT NULL,  
    LASTNAME   VARCHAR(15)     NOT NULL,  
    SALARY     DECIMAL(9,2)      ,  
    PRIMARY KEY (EMPNO)  
) "
```

En Linux / UNIX, también puedes escapar de los caracteres especiales con barra diagonal invertida, pero esto podría ser más difícil de escribir:

```
db2 CREATE TABLE EMPLOYEE \(\ \
EMPNO CHAR\ (6\) NOT NULL, \
FIRSTNME VARCHAR\ (12\) NOT NULL, \
LASTNAME VARCHAR\ (15\) NOT NULL, \
SALARY DECIMAL\ (9,2\), \
PRIMARY KEY \ (EMPNO\)\
\)
```

## Insertando una fila en una tabla de DB2

Supongamos que vamos a insertar filas en la tabla creada anteriormente.

Podemos nombrar explícitamente las columnas a las que vamos a dar los valores y su orden:

```
INSERT INTO EMPLOYEE (EMPNO, FIRSTNME, LASTNAME, SALARY)
VALUES ( '123456', 'Ali', 'Veli', 100000);
```

Si conocemos el orden y vamos a poner valores para todas las columnas, podemos escribir:

```
INSERT INTO EMPLOYEE
VALUES ( '123456', 'Ali', 'Veli', 100000);
```

Cuando usamos el db2clp, necesitamos poner comillas debido al paréntesis (sin punto y coma al final):

```
db2 "INSERT INTO EMPLOYEE (EMPNO, FIRSTNME, LASTNAME, SALARY)
VALUES ( '123456', 'Ali', 'Veli', 100000)"
```

## muestra de consulta de selección;

```
SELECT 'HELLO WORLD' FROM SYSIBM.SYSDUMMY1;

1
-----
Hello World

1 record(s) selected.
```

"La tabla SYSIBM.SYSDUMMY1 contiene una fila. La tabla se usa para sentencias de SQL en las que se requiere una referencia de tabla, pero el contenido de la tabla no es importante"

Esta tabla tiene una sola columna. El nombre de la columna es IBMREQD. El valor predeterminado es Y.

```
SELECT * FROM SYSIBM.SYSDUMMY1;
IBMREQD
-----
Y

1 record(s) selected.
```

Lea Hola Mundo; en línea: <https://riptutorial.com/es/db2/topic/5995/hola-mundo->

# Capítulo 8: Información de restricción de DB2

## Introducción

Esta documentación ayudará a cualquiera que esté buscando todos los Contratos en una columna de una tabla. La consulta se puede modificar para encontrar la tabla / columnas según el nombre de la restricción.

## Examples

### Obtener restricciones basadas en el nombre de la columna

```
select cst.constraint_schema, cst.constraint_name,
       fk.table_name, fk.ordinal_position, fk.column_name,
       pk.table_name, pk.column_name
from qsys2.syscst cst join qsys2.syskeycst fk
  on fk.constraint_schema = cst.constraint_schema
  and fk.constraint_name = cst.constraint_name
join qsys2.sysrefcst ref
  on ref.constraint_schema = cst.constraint_schema
  and ref.constraint_name = cst.constraint_name
join qsys2.syskeycst pk
  on pk.constraint_schema = ref.unique_constraint_schema
  and pk.constraint_name = ref.unique_constraint_name
where cst.constraint_type = 'FOREIGN KEY'
  and fk.ordinal_position = pk.ordinal_position
  and pk.table_name = 'PRIMARYTABLE'
  and pk.column_name = 'EMPID'

order by cst.constraint_schema, cst.constraint_name
```

### Obtener información de restricciones basada en el nombre de la restricción

```
select cst.constraint_schema, cst.constraint_name,
       fk.table_name, fk.ordinal_position, fk.column_name,
       pk.table_name, pk.column_name
from qsys2.syscst cst join qsys2.syskeycst fk
  on fk.constraint_schema = cst.constraint_schema
  and fk.constraint_name = cst.constraint_name
join qsys2.sysrefcst ref
  on ref.constraint_schema = cst.constraint_schema
  and ref.constraint_name = cst.constraint_name
join qsys2.syskeycst pk
  on pk.constraint_schema = ref.unique_constraint_schema
  and pk.constraint_name = ref.unique_constraint_name
where fk.ordinal_position = pk.ordinal_position
  and cst.constraint_name = 'CST NAME'

order by cst.constraint_schema, cst.constraint_name
```

Lea Información de restricción de DB2 en línea:



<https://riptutorial.com/es/db2/topic/9381/informacion-de-restriccion-de-db2>

---

# Capítulo 9: Insertar en una tabla seleccionando de otra tabla

## Examples

### Insertar en una tabla seleccionando de otra tabla

```
insert into schema.table (field1, field2)
  select 'Static Value', foreignField from schema.otherTable;
```

Lea [Insertar en una tabla seleccionando de otra tabla en línea](https://riptutorial.com/es/db2/topic/6306/insertar-en-una-tabla-seleccionando-de-otra-tabla):

<https://riptutorial.com/es/db2/topic/6306/insertar-en-una-tabla-seleccionando-de-otra-tabla>

---

# Capítulo 10: Mesa plegable

## Introducción

Instrucciones básicas para soltar una tabla en DB2.

## Examples

### Sintaxis básica de la tabla de caída

```
db2 connect to {databaseName}
db2 drop table {schema}.{table}
db2 connect reset
```

El esquema no es necesario si coincide con el nombre de usuario actual. El prefijo "db2" no es necesario si ya se encuentra en un indicador de comandos de DB2.

Lea Mesa plegable en línea: <https://riptutorial.com/es/db2/topic/10115/mesa-plegable>

# Creditos

S. No	Capítulos	Contributors
1	Empezando con db2	<a href="#">AngocA</a> , <a href="#">Community</a> , <a href="#">J. Allen</a> , <a href="#">Surya Sg</a>
2	Analizar los valores separados por comas en una columna en varias filas	<a href="#">mustaccio</a>
3	Combine varias filas en un solo valor separado por comas	<a href="#">J. Allen</a> , <a href="#">Stephen Leppik</a>
4	Copia tabla con o sin datos	<a href="#">uSeruSher</a>
5	dashDB	<a href="#">SilentSteel</a>
6	Donde no en columnas múltiples	<a href="#">J. Allen</a>
7	Hola Mundo;	<a href="#">AngocA</a> , <a href="#">Mustafa DOGRU</a> , <a href="#">Wieland</a>
8	Información de restricción de DB2	<a href="#">uSeruSher</a>
9	Insertar en una tabla seleccionando de otra tabla	<a href="#">AngocA</a> , <a href="#">J. Allen</a>
10	Mesa plegable	<a href="#">J. Allen</a>