FREE eBook

LEARNING db2

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



Table of Contents

About1
Chapter 1: Getting started with db2
Remarks
Examples
Installation2
Chapter 2: Combine multiple rows into a single comma separated value
Examples
Using the listagg() function
Chapter 3: Copy table with or without data 4
Introduction4
Syntax
Examples4
Copy Table With Data
Copy Table without data4
Copy Table with where clause4
Chapter 4: dashDB
Introduction
Remarks
Remarks
Remarks
Remarks .5 For more information about dashDB, you can try these sources: .5 General information: .5
Remarks .5 For more information about dashDB, you can try these sources: .5 General information: .5 More Documentation .5
Remarks .5 For more information about dashDB, you can try these sources: .5 General information: .5 More Documentation .5 Examples .5
Remarks 5 For more information about dashDB, you can try these sources: 5 General information: 5 More Documentation 5 Examples 5 A basic SQL select- List all rows in a table. 5
Remarks 5 For more information about dashDB, you can try these sources: 5 General information: 5 More Documentation. 5 Examples. 5 A basic SQL select- List all rows in a table. 5 Chapter 5: DB2 Constraint Information 6
Remarks .5 For more information about dashDB, you can try these sources: .5 General information: .5 More Documentation. .5 Examples. .5 A basic SQL select- List all rows in a table. .5 Chapter 5: DB2 Constraint Information .6
Remarks .5 For more information about dashDB, you can try these sources: .5 General information: .5 More Documentation. .5 Examples. .5 A basic SQL select- List all rows in a table. .5 Chapter 5: DB2 Constraint Information .6 Introduction. .6 Examples. .6
Remarks .5 For more information about dashDB, you can try these sources: .5 General information: .5 More Documentation .5 Examples. .5 A basic SQL select- List all rows in a table. .5 Chapter 5: DB2 Constraint Information .6 Introduction .6 Examples. .6 Get constraints based on column name. .6

Examples	7
Basic Drop Table Syntax	7
Chapter 7: Hello World;	8
Examples	8
Creating a database in DB2	
Connecting to a database in DB2	8
Create a table in DB2 called "employee"	8
Inserting a row into a DB2 table	
sample select query;	9
Chapter 8: Insert into one table by selecting from another table	
Examples	
Insert into one table by selecting from another table	10
Chapter 9: Parse comma-separated values in a column into multiple rows	
Syntax	11
Examples	11
Recursive query to parse comma-separated values	11
Chapter 10: Where not in multiple columns	14
Examples	14
Filter out multiple combinations of values	14
Credits	15



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

Chapter 1: Getting started with db2

Remarks

This section provides an overview of what db2 is, and why a developer might want to use it.

It should also mention any large subjects within db2, and link out to the related topics. Since the Documentation for db2 is new, you may need to create initial versions of those related topics.

Examples

Installation

There different flavors of DB2. One of them is LUW: Linux, UNIX and Windows.

DB2 LUW in Linux / UNIX can be installed with or without root. When installed with root, you can create different instances associating them to different users.

When installing DB2 LUW without root privileges, you can install DB2 in your home directory and your user will be automatically the only instance this installation can have. The instance will not be started automatically each time the machine boots at least you configure that.

Once you have downloaded the binaries (from Fixpack Central, Passport Advantage, or a CD), you can extract the files. There will be a directory that describes the DB2 edition you a using (Expc, server_t, etc.), and in that directory you will find db2setup and db2_install. One of them is for graphic installation, the other is for text installation via response file.

Read Getting started with db2 online: https://riptutorial.com/db2/topic/5617/getting-started-with-db2

Chapter 2: Combine multiple rows into a single comma separated value

Examples

Using the listagg() function

Let's say you've got a table of loans, and another related table of parcels, where each loan can have one or more parcels associated with it. If you want a query to show each loan and a list of all its associated parcels, but you only want each loan to show up once, then you could use something like this:

```
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
```

Read Combine multiple rows into a single comma separated value online: https://riptutorial.com/db2/topic/5940/combine-multiple-rows-into-a-single-comma-separated-value

Chapter 3: Copy table with or without data

Introduction

Example of how to copy existing table structure with/without data

Syntax

1. CREATE TABLE schemaName.table AS (SELECT columns FROM schemaName.table) WITH DATA

Examples

Copy Table With Data

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

Copy Table without data

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

Copy Table with where clause

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

Read Copy table with or without data online: https://riptutorial.com/db2/topic/8778/copy-table-withor-without-data

Chapter 4: dashDB

Introduction

IBM dashDB is a family of SQL databases. It's engine is a blend of DB2, Netezza and the BLU inmemory engine. Any supported DB2 driver will work; it's a drop-replacement for DB2.

The 3 product lines for dashDB are: (1) IBM dashDB for Transactions: A general-purpose, fully managed cloud SQL database. (2) IBM dashDB for Analytics: A cloud SQL data warehouse. (3) IBM dashDB Local: A local version you can install on your own hardware.

Remarks

For more information about dashDB, you can try these sources:

General information:

- Main website: http://dashdb.com/
- dashDB for Transactions Bluemix Page
- dashDB for Analytics Bluemix Page

More Documentation

- dashDB docs on Bluemix
- IBM Knowledge Center for dashDB
- API reference for dashDB for Analytics

Examples

A basic SQL select- List all rows in a table.

SELECT * FROM MY_TABLE

Read dashDB online: https://riptutorial.com/db2/topic/8673/dashdb

Chapter 5: DB2 Constraint Information

Introduction

This documentation will help anyone who is looking for all the Contraints on a column of a table. The query can be modified to find the table/columns based on the constraint name.

Examples

Get constraints based on column name

```
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

Get constraints info based on constraint name

order by cst.constraint_schema, cst.constraint_name

Read DB2 Constraint Information online: https://riptutorial.com/db2/topic/9381/db2-constraint-information

Chapter 6: Drop table

Introduction

Basic instructions to drop a table in DB2.

Examples

Basic Drop Table Syntax

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

The schema is not necessary if it matches the current user name. The "db2" prefix is not necessary if you are already in a DB2 command prompt.

Read Drop table online: https://riptutorial.com/db2/topic/10115/drop-table

Chapter 7: Hello World;

Examples

Creating a database in DB2

CREATE DATABASE SAMPLEDB;

This will create a new database called sampledb.

Connecting to a database in DB2

CONNECT TO SAMPLEDB;

From the command line (db2clp, terminal, db2cmd) you can write:

db2 CONNECT TO SAMPLEDB

Create a table in DB2 called "employee"

The following statement will create a new table called employee:

```
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)

)
```

This will create a new table called employee. The table will have a primary key on EMPNO column. The first three columns cannot have a null value and they are text. The fourth one can have nulls and it is a number.

You can create this table from db2clp (Linux, UNIX, MacOS) like this (by surrounding the statement into quotes):

```
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)

)"
```

In Linux/UNIX, you can also escape the special characters with back-slash, but this could be more difficult to write:

```
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 \(EMPNO\) \
    \)
```

Inserting a row into a DB2 table

Let's suppose we are going to insert rows in the previously created table.

We can explicitly name the columns we are going to out values is and its order:

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

If we know the order and we are going to put values for all columns we can write:

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

When using the db2clp, we need to put quotes because of the parenthesis (without semicolon at the end):

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

sample select query;

```
SELECT 'HELLO WORLD' FROM SYSIBM.SYSDUMMY1;
1
------
Hello World
1 record(s) selected.
```

"The SYSIBM.SYSDUMMY1 table contains one row. The table is used for SQL statements in which a table reference is required, but the contents of the table are not important"

this table has only one column. Column Name is IBMREQD. Default value is Y.

```
SELECT * FROM SYSIBM.SYSDUMMY1;
IBMREQD
-----
Y
1 record(s) selected.
```

Read Hello World; online: https://riptutorial.com/db2/topic/5995/hello-world-

Chapter 8: Insert into one table by selecting from another table

Examples

Insert into one table by selecting from another table

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

Read Insert into one table by selecting from another table online: https://riptutorial.com/db2/topic/6306/insert-into-one-table-by-selecting-from-another-table

Chapter 9: Parse comma-separated values in a column into multiple rows

Syntax

- WITH CTE_name (column_name[,...]) AS (
- SELECT column_name[,...] FROM base_table
- UNION ALL
- SELECT column_name[,...] FROM CTE_name
- WHERE < recursion limiting condition>
-)
- SELECT column_name[,...] FROM CTE_name

Examples

Recursive query to parse comma-separated values

Although storing multiple values in a single column violates normalization rules, sometimes one has to deal with badly designed legacy tables. A recursive query can help convert comma-separated values into distinct rows.

Create a sample badly designed table and insert some data:

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

Check what we have:

select * from projects;

will output

 NAME
 MEMBERS

 Luna
 1, 3, 4

 Terra
 2, 3, 5

 2 record(s) selected.

Use a common table expression (CTE) to recursively extract each comma-separated value from MEMBERS into its own row:

```
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
```

will return

NAME I	IEMBER
Luna	1
Luna	3
Luna	4
Terra	2
Terra	3
Terra	5
6 record(s	s) selected.

The result returned by the CTE can be used as a regular table, e.g. by joining it to another table. For example, create an employee lookup table:

Then the following query

```
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
```

will return

Project name Member name -------Luna John Luna Mishka Luna Venkat Terra Peter Terra Venkat Terra Xiao 6 record(s) selected.

Read Parse comma-separated values in a column into multiple rows online: https://riptutorial.com/db2/topic/6751/parse-comma-separated-values-in-a-column-into-multiplerows

Chapter 10: Where not in multiple columns

Examples

Filter out multiple combinations of values

Lets say you want to filter a query by two columns, but only certain combinations of those columns. For example, it's OK to have account 60400 with reference JE, but you cannot have account 60400 with reference ED, but you can have account 60500 with reference 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')
);
```

Read Where not in multiple columns online: https://riptutorial.com/db2/topic/6349/where-not-inmultiple-columns

Credits

S. No	Chapters	Contributors
1	Getting started with db2	AngocA, Community, J. Allen, Surya Sg
2	Combine multiple rows into a single comma separated value	J. Allen, Stephen Leppik
3	Copy table with or without data	uSeruSher
4	dashDB	SilentSteel
5	DB2 Constraint Information	uSeruSher
6	Drop table	J. Allen
7	Hello World;	AngocA, Mustafa DOGRU, Wieland
8	Insert into one table by selecting from another table	AngocA, J. Allen
9	Parse comma- separated values in a column into multiple rows	mustaccio
10	Where not in multiple columns	J. Allen