**DB2**

**(https://www.tutorialspoint.com/db2/db2\_databases.htm)**

**DB2 INSTANCE**

This tutorial provides you the basic understanding of concepts of database, database installation and management. At the end of the tutorial you should be equipped with well understanding of database management concepts.

**Audience**

This tutorial is designed for the readers pursuing education in database management domain and all enthusiastic readers.

**Prerequisites**

This tutorial is designed and developed for absolute beginners. Though, awareness about software systems, operating systems and computer fundamentals would be beneficial.

INTRODUCTION: ---

Overview

DB2 is a database product from IBM. It is a Relational Database Management System (RDBMS). DB2 is designed to store, analyze and retrieve the data efficiently.

DB2 product is extended with the **support of Object-Oriented features** and **non-relational structures with XML.**

**Versions**

For IBM DB2, the UDB current version is 10.5 with the features of BLU Acceleration and its code name as 'Kepler'. All the versions of DB2 till today are listed below:

|  |  |
| --- | --- |
| **Version** | **Code Name** |
| 3.4 | Cobweb |
| 8.1, 8.2 | Stinger |
| 9.1 | Viper |
| 9.5 | Viper 2 |
| 9.7 | Cobra |
| 9.8 | It added features with Only PureScale |
| 10.1 | Galileo |
| 10.5 | Kepler |

**Data server editions and features**

Depending upon the requirement of needful features of DB2, the organizations select appropriate DB2 version. The following table shows DB2 server editions and their features:

|  |  |
| --- | --- |
| **Editions** | **Features** |
| Advanced Enterprise Server Edition and Enterprise Server Edition (AESE / ESE) | It is designed for mid-size to large-size business organizations. Platform - Linux, UNIX, and Windows. Table partitioning High Availability Disaster Recovery (HARD) Materialized Query Table (MQTs) Multidimensional Clustering (MDC) Connection concentrator Pure XML Backup compression Homogeneous Federations |
| Workgroup Server Edition (WSE) | It is designed for Workgroup or mid-size business organizations. Using this WSE you can work with - High Availability Disaster Recovery (HARD) Online Reorganization Pure XML Web Service Federation support DB2 Homogeneous Federations Homogeneous SQL replication Backup compression |
| Express -C | It provides all the capabilities of DB2 at zero charge. It can run on any physical or virtual systems with any size of configuration. |
| Express Edition | It is designed for entry level and mid-size business organizations. It is full featured DB2 data server. It offers only limited services. This Edition comes with - Web Service Federations DB2 homogeneous federations Homogeneous SQL Replications Backup compression |
| Enterprise Developer Edition | It offers only single application developer. It is useful to design, build and prototype the applications for deployment on any of the IBM server. The software cannot be used for developing applications |

**DB2 - Instance**

Introduction

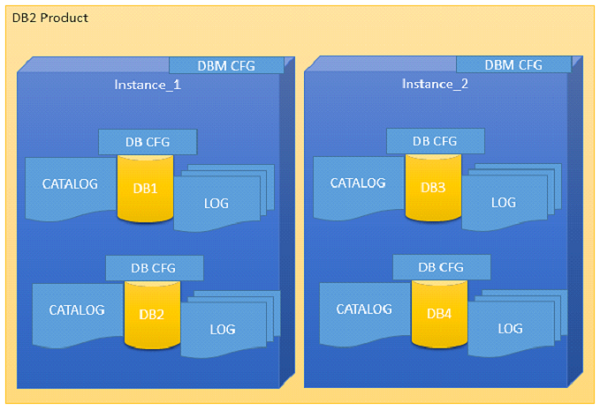
An Instance is a logical environment for DB2 Database Manager. Using instance, you can **manage databases.** Depending on our requirements, you can create multiple instances on one physical machine. The contents of Instance directory are:

* Database Manager Configuration file
* System Database Directory
* Node Directory
* Node Configuration File [db2nodes.cfg]
* Debugging files, dump files

For DB2 Database Server, **the default instance is “DB2”.** It is not possible to change the location of Instance directory after its creation. An instance can manage multiple databases. In an instance, each database has a unique name, its own set of catalog tables, configurations files, authorities and privileges.

Ezoic

Architecture of instance in DB2 product



Multiple instances

You can create multiple instances in one DB2Server on Linux, UNIX and Windows. It is possible to install multiple DB2Servers on a physical machine.

Ezoic

Creating instance on Linux

You can create multiple instances on Linux and UNIX if DB2 Server is installed as root user. An **instance can run simultaneously on Linux and UNIX independently. You can work within a single instance of the database manager at a time.**

An Instance folder contains database configuration files and folders. The Instance directory is stored at different locations on Windows depending on the operating system versions.

Listing instances

The following command is used to list instances:

db2ilist

This command lists all the instances that are available on a system.

**Syntax:**

db2ilist

**Example:**[To see how many instances are created in DB2 copy]

db2ilist

**Output:**

db2inst1

db2inst2

db2inst3

Ezoic

Instance environment commands

These commands are useful to work with arrangement of instance in the DB2 CLI.

**Get instance**

This command shows details of the currently running instance.

**Syntax:**

db2 get instance

**Example:**[To see the current instance which activated the current user]

db2 get instance

**Output:**

The current database manager instance is : db2inst1

**Set instance**

To start or stop the database manager of an instance on DB2 UDB, the following command is executed for the current instance.

**Syntax:**

set db2instance=<instance\_name>

**Example:**[ To arrange the “db2inst1” environment to current user]

set db2instance=db2inst1

db2start

Using this command, you can start an instance. Before this, you need to run “set instance”.

**Syntax:**

db2start

**Example:**[To start an instance]

db2start

**Output:**

SQL1063N DB2START processing was successful

db2stop

Using this command you can stop a running instance.

**Syntax:**

db2stop

**Output:**

SQL1064N DB2STOP processing was successful.

**Updating an instance**

**db2iupdt**

This command is used to update the instance within the same version release. Before executing this command, you need to stop the instance database manager using “db2stop” command. The

You can update an instance using following command:

**Syntax 1**: To update an instance in normal mode

db2iupdt <inst\_name>

**Example1:**. /db2iupdt db2inst2

**Syntax 2**: To update an instance in debugging mode

db2iupdt -D <inst\_name>

**Example**

db2iupdt -D db2inst2

**db2iupgrade**

On Linux or UNIX system, this command is located in DB2DIR/instance directory. In the following syntaxes, “inst\_name” indicates the previous version DB2 instance and “inst\_username” indicates the current installed version DB2 copy instance user.

**Syntax 2**:

db2iupgrade -d -k -u <inst\_username> <inst\_name>

**Example**:

db2iupgrade -d -k -u db2inst2 db2inst2

**Command Parameters:**

**-d** : Turns debugging mode on.

**-k** : Keeps the pre-upgrade instance type if it is supported in the DB2 copy, from where you are running this command.

If you are using the Super User (su) on Linux for db2iupgrade command, you must issue the “su” command with the “-” option.

**Dropping an instance**

You can drop or delete the instance, which was created by “db2icrt” command.

db2idrop

On Linux and UNIX operating system, this command is located in the DB2\_installation\_folder/instance directory.

**Syntax**: [in the following syntax, ‘inst\_username’ indicates username of instance and ‘inst\_name’ indicates instance name]

db2idrop -u <inst\_username> <inst\_name>

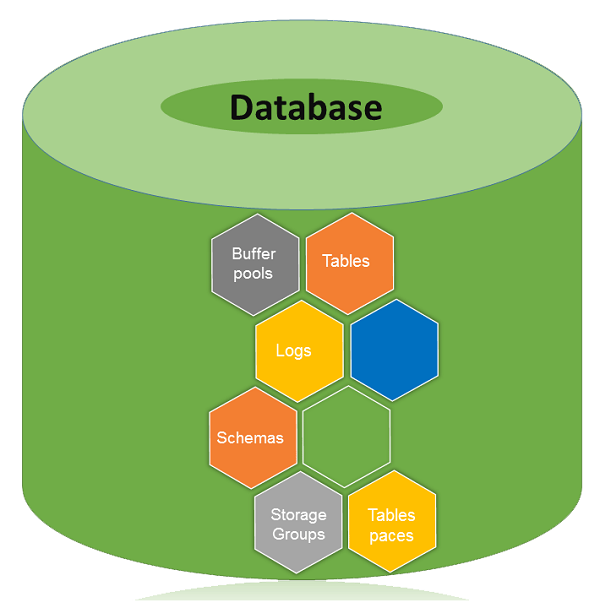
**Example**: [To drop db2inst2]

./db2idrop -u db2inst2 db2inst2

**DB2 DATABASES**

This chapter describes creating, activating and deactivating the databases with the associated syntax.

Database architecture



A database is a collection of Tables, Schemas, Bufferpools, Logs, Storage groups and Tablespaces working together to handle database operations efficiently.

**Database directory**

Database directory is an organized repository of databases. When you create a database, all the details about database are stored in a database directory, such as details of default storage devices, configuration files, and temporary tables list etc.

Partition global directory is created in the instance folder. This directory contains all global information related to the database. This partition global directory is named as NODExxxx/SQLyyy, where xxxx is the data partition number and yyy is the database token.

In the partition-global directory, a member-specific directory is created. This directory contains local database information. The member-specific directory is named as MEMBERxxxx where xxxx is a member number. DB2 Enterprise Server Edition environment runs on a single member and has only one member specific directory. This member specific directory is uniquely named as MEMBER0000.

**Partitioned global directory**

Directory Location : <instance>/NODExxx/SQLxxx

The partition-global directory contains database related files as listed below.

* Global deadlock write-to-file event monitoring files
* Table space information files [SQLSPCS.1, SQLSPCS.2]
* Storage group control files [SQLSGF.1, SQLSGF.2]
* Temporary table space container files. [/storage path//T0000011/C000000.TMP/SQL00002.MEMBER0001.TDA]
* Global Configuration file [SQLDBCONF]
* History files [DB2RHIST.ASC, DB2RHIST.BAK, DB2TSCHG.HIS, DB2TSCHG.HIS]
* Logging-related files [SQLOGCTL.GLFH.1, SQLOGCTL.GLFH.2]
* Locking files [SQLINSLK, SQLTMPLK]
* Automatic Storage containers

**Member specific directory**

Directory location : /NODExxxx/SQLxxxx/MEMBER0000

This directory contains:

* Objects associated with databases
* Buffer pool information files [SQLBP.1, SQLBP.2]
* Local event monitoring files
* Logging-related files [SQLOGCTL.LFH.1, SQLOGCTL.LFH.2, SQLOGMIR.LFH].
* Local configuration files
* Deadlocks event monitor file. The detailed deadlock events monitor files are stored in the database directory of the catalog node in case of ESE and partitioned database environment.

**Creating database**

You can create a database in instance using the “CREATE DATABASE” command. All databases are created with the default storage group “IBMSTOGROUP”, which is created at the time of creating an instance. In DB2, all the database tables are stored in “tablespace”, which use their respective storage groups.

The privileges for database are automatically set as PUBLIC [CREATETAB, BINDADD, CONNECT, IMPLICIT\_SCHEMA, and SELECT], however, if the RESTRICTIVE option is present, the privileges are not granted as PUBLIC.

**Creating non-restrictive database**

This command is used to create a non-restrictive database.

**Syntax**: [To create a new Database. ‘database\_name’ indicates a new database name, which you want to create.]

db2 create database <database name>

**Example**: [To create a new non-restrictive database with name ‘one’]

db2 create database one

**Output:**

DB20000I The CREATE DATABASE command completed successfully.

**Creating restrictive database**

Restrictive database is created on invoking this command.

**Syntax**: [In the syntax below, “db\_name” indicates the database name.]

db2 create database <db\_name> restrictive

**Example**: [To create a new restrictive database with the name ‘two’]

db2 create database two restrictive

**Creating database with different user defined location**

Create a database with default storage group “IBMSTOGROUP” on different path. Earlier, you invoked the command “create database” without any user-defined location to store or create database at a particular location. To create the database using user- defined database location, the following procedure is followed:

**Syntax**: [In the syntax below, ‘db\_name’ indicates the ‘database name’ and ‘data\_location’ indicates where have to store data in folders and ‘db\_path\_location’ indicates driver location of ‘data\_location’.]

db2 create database '<db\_name>' on '<data location>' dbpath on '<db\_path\_location>'

**Example**: [To create database named ‘four’, where data is stored in ‘data1’ and this folder is stored in ‘dbpath1’]

db2 create database four on '/data1' dbpath on '/dbpath1'

Viewing local or system database directory files

You execute this command to see the list of directories available in the current instance.

**Syntax:**

db2 list database directory

**Example:**

db2 list database directory

**Output:**

System Database Directory

Number of entries in the directory = 6

Database 1 entry:

Database alias = FOUR

Database name = FOUR

Local database directory =

/home/db2inst4/Desktop/dbpath

Database release level = f.00

Comment =

Directory entry type = Indirect

Catalog database partition number = 0

Alternate server hostname =

Alternate server port number =

Database 2 entry:

Database alias = SIX

Database name = SIX

Local database directory = /home/db2inst4

Database release level = f.00

Comment =

Directory entry type = Indirect

Catalog database partition number = 0

Alternate server hostname =

Alternate server port number =

**Activating database**

This command starts up all necessary services for a particular database so that the database is available for application.

**Syntax**:[‘db\_name’ indicates database name]

db2 activate db <db\_name>

**Example**: [Activating the database ‘one’]

db2 activate db one

**Deactivating database**

Using this command, you can stop the database services.

**Syntax:**

db2 deactivate db <db\_name>

**Example**: [To Deactivate database ‘one’]

db2 deactivate db one

**Connecting to database**

After creating a database, to put it into use, you need to connect or start database.

**Syntax:**

db2 connect to <database name>

**Example**: [To Connect Database one to current CLI]

db2 connect to one

**Output:**

Database Connection Information

Database server = DB2/LINUXX8664 10.1.0

SQL authorization ID = DB2INST4

Local database alias = ONE

**Verifying if database is restrictive**

To check if this database is restrictive or not, here is the syntax:

**Syntax**: [In the following syntax, ‘db’ indicates Database, ‘cfg’ indicates configuration, ‘db\_name’ indicates database name]

db2 get db cfg for <db\_name> | grep -i restrict

**Example**: [To check if ‘one’ database is restricted or not]

db2 get db cfg for one | grep -i restrict

**Output:**

Restrict access = NO