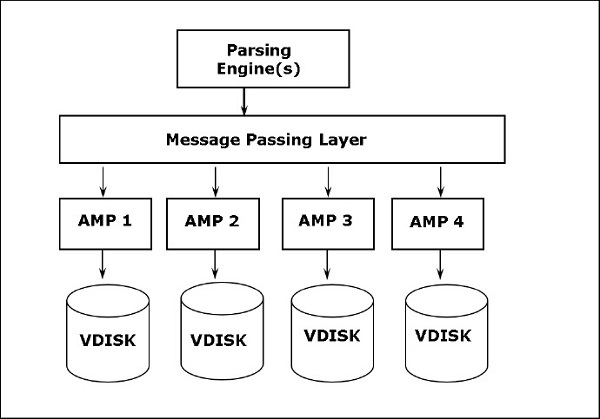
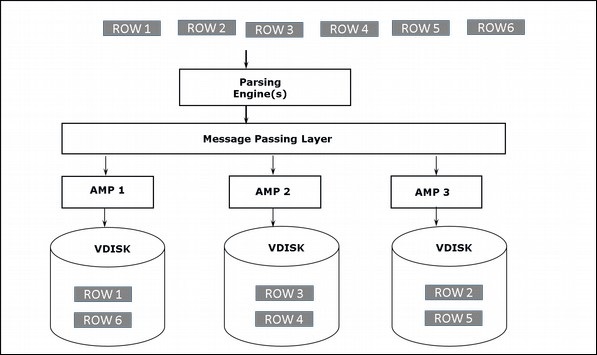
**Teradata basic understanding:**

Teradata is one of the popular Relational Database Management System. It is mainly suitable for building large scale data warehousing applications. Teradata achieves this by the concept of parallelism. It is developed by the company called Teradata..,

**Teradata - Architecture**



Storage Architecture



CREATE TABLE command is used to create tables in Teradata.

Syntax

Following is the generic syntax of CREATE TABLE statement.

CREATE <SET/MULTISET> TABLE <Tablename>

<Table Options>

<Column Definitions>

<Index Definitions>;

Example:

CREATE SET TABLE EMPLOYEE,FALLBACK (

EmployeeNo INTEGER,

FirstName VARCHAR(30),

LastName VARCHAR(30),

DOB DATE FORMAT 'YYYY-MM-DD',

JoinedDate DATE FORMAT 'YYYY-MM-DD',

DepartmentNo BYTEINT

)

UNIQUE PRIMARY INDEX ( EmployeeNo );

Insert vales in the table:

Syntax:

INSERT INTO <tablename>

(column1, column2, column3,…)

VALUES

(value1, value2, value3 …);

Example:

INSERT INTO Employee (

EmployeeNo,

FirstName,

LastName,

BirthDate,

JoinedDate,

DepartmentNo

)

VALUES (

101,

'Mike',

'James',

'1980-01-05',

'2005-03-27',

01

);

**Alter command:**

ALTER TABLE command is used to add or drop columns from an existing table. You can also use ALTER TABLE command to modify the attributes of the existing columns.

Syntax

Following is the generic syntax for ALTER TABLE.

ALTER TABLE <tablename>

ADD <columnname> <column attributes>

DROP <columnname>;

**2.Basic understanding about control-m**:

Control-M is a workload automation tool that helps organizations automate and manage their business workflows. Here’s a basic overview to get you started:

**What is Control-M?**

Control-M is a solution for **workload automation** and **job scheduling**.

**Key Features:**

1. **Job Scheduling**: Automates the execution of jobs (tasks) based on predefined schedules and dependencies.
2. **Workflow Automation**: Manages complex workflows by defining the sequence and conditions under which jobs should run.
3. **Monitoring and Alerts**: Provides real-time monitoring of job execution and sends alerts for any issues or failures.
4. [**Cross-Platform Support**: Works across multiple platforms, applications, and data sources, integrating processes running on any system into an automatic job flow1](https://documents.bmc.com/supportu/9.0.21.200/en-US/Documentation/Introduction_to.htm).

**Basic Components:**

1. **Jobs**: Execution units such as scripts or commands that perform specific tasks.
2. **Folders**: Group related jobs to manage them more easily.
3. **Workflows**: Define the sequence and dependencies of jobs to ensure they run in the correct order.
4. **Control-M Server**: The central component that manages job scheduling and execution.
5. **Control-M/Agent**: Installed on the machines where jobs will run, executing the tasks as instructed by the Control-M Server.

**Applications:**

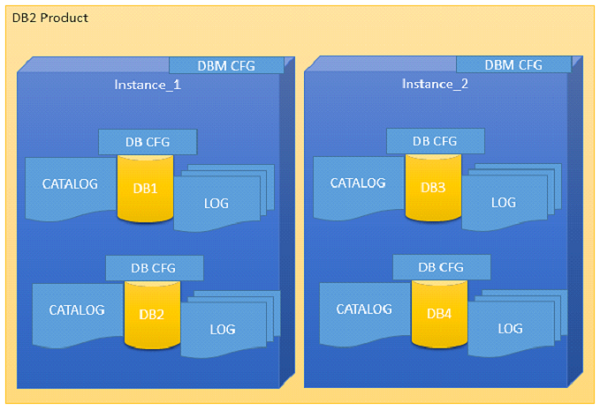
* **Data Processing**: Automates data extraction, transformation, and loading (ETL) processes.
* **Batch Processing**: Manages batch jobs in various environments, ensuring they run efficiently and on time.
* **IT Operations**: Automates routine IT tasks like backups, updates, and system maintenance

3.Db2 basic understanding:

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.

Architecture of instance in DB2 product



Instance: An instance shows the data or information that is stored in the database

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

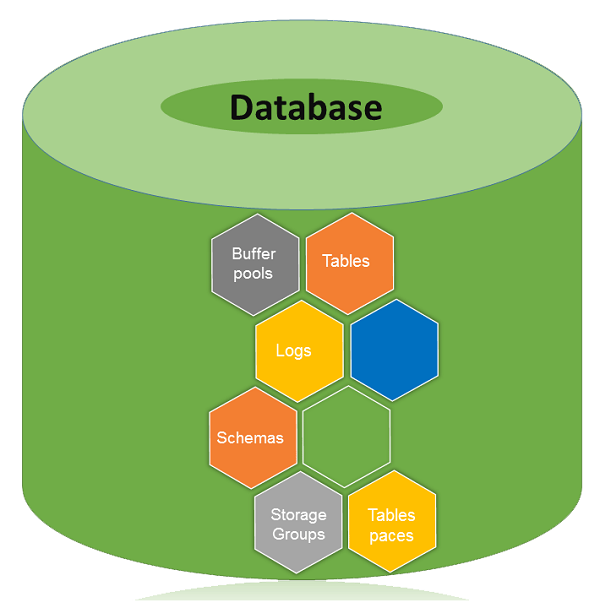
**Output:**

db2inst1

db2inst2

db2inst3

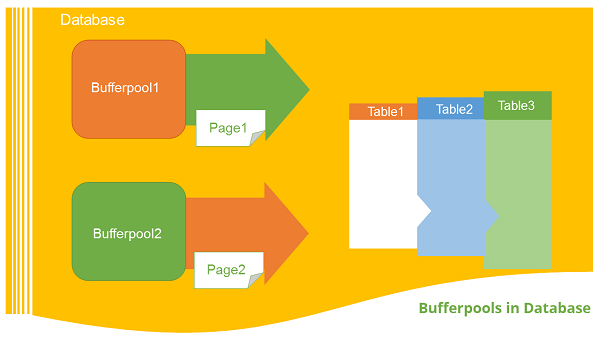
Database architecture



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

Bufferpools

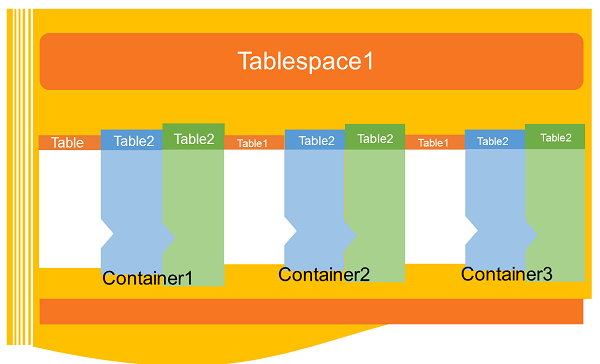
This chapter introduces you to Bufferpools in the database.



Introduction

The bufferpool is portion of a main memory space which is allocated by the database manager. The purpose of bufferpools is to cache table and index data from disk. All databases have their own bufferpools. A default bufferpool is created at the time of creation of new database. It called as “IBMDEFAULTBP”. Depending on the user requirements, it is possible to create a number of bufferpools.

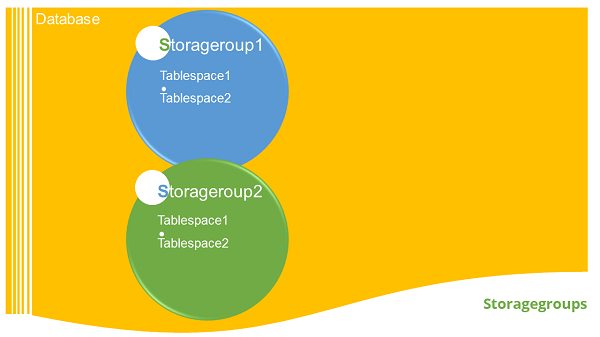
Tablespace:



Introduction

A table space is a storage structure, it contains tables, indexes, large objects, and long data. It can be used to organize data in a database into logical storage group which is related with where data stored on a system. This tablespaces are stored in database partition groups

This chapter describes the Database Storagegroups:



Introduction

A set of Storage paths to store database table or objects, is a storage group. You can assign the tablespaces to the storage group. When you create a database, all the tablespaces take default storagegorup. The default storage group for a database is ‘IBMSTOGROUP’. When you create a new database, the default storage group is active, if you pass the “AUTOMATIC STOGROUP NO” parameter at the end of “CREATE DATABASE” command. The database does not have any default storage groups.

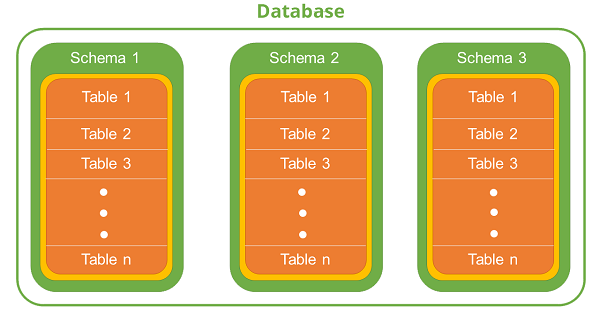
Listing instances

This chapter introduces and describes the concept of Schema:.

Introduction

A schema is a collection of named objects classified logically in the database.

In a database, you cannot create multiple database objects with same name. To do so, the schema provides a group environment. You can create multiple schemas in a database and you can create multiple database objects with same name, with different schema groups.



Creating Tables

The following syntax creates table:

**Syntax**: [To create a new table]

db2 create table <schema\_name>.<table\_name>

(column\_name column\_type....) in <tablespace\_name>

**Example**: We create a table to store “employee” details in the schema of “professional”. This table has “id, name, jobrole, joindate, salary” fields and this table data would be stored in tablespace “ts1”.

db2 create table professional.employee(id int, name

varchar(50),jobrole varchar(30),joindate date,

salary double) in ts1

**Output:**

DB20000I The SQL command completed successfully.

Listing table details:

The following syntax is used to list table details:

**Syntax**: [To see the list of tables created with schemas]

db2 select tabname, tabschema, tbspace from syscat.tables

**Example**: [To see the list of tables in the current database]

db2 select tabname, tabschema, tbspace from syscat.tables

**Output:**

TABNAME TABSCHEMA TBSPACE

------------ ------------- --------

EMPLOYEE PROFESSIONAL TS1

3 informatica basic understanding

Informatica is introduced as a software development company in the market. It provides a complete data integration solution and data management system. It launched multiple products that mainly focused on data integration.

Informatica Architecture

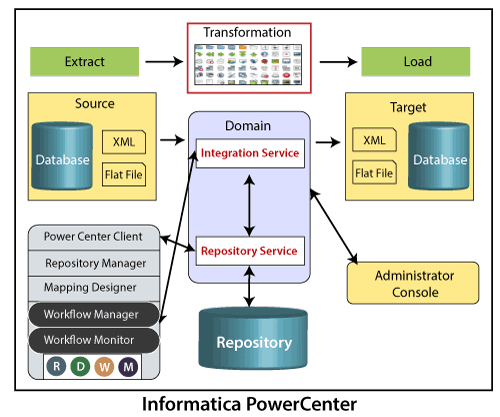
Informatica architecture is service-oriented architecture (SOA). A service-oriented architecture is defined as a group of services that communicate with each other. It means a simple data transfer during this communication, or it can be two or more services that coordinate the same activity.

The Informatica development depends upon the component-based development techniques. This technique uses the predefined components and functional units with their functionalities to get the result.

Informatica is a powerful data integration tool widely used for ETL (Extract, Transform, Load) processes. Here’s a basic overview to get you started:

**What is Informatica?**

Informatica provides a suite of data integration products, with **Informatica PowerCenter** being the most popular. It helps in extracting data from various sources, transforming it as per business requirements, and loading it into target systems like data warehouses.



**Key Components:**

1. **Repository Manager**: Manages metadata and stores information about mappings, sources, targets, and transformations.
2. **Designer**: Used to create mappings between source and target systems.
3. **Workflow Manager**: Helps in creating, scheduling, and managing workflows.
4. **Workflow Monitor**: Monitors the execution of workflows and provides details about their status and logs.

**Basic Steps in Informatica:**

1. **Creating a Mapping**: Define how data is extracted, transformed, and loaded.
2. **Creating a Workflow**: Define the sequence of tasks and their dependencies.
3. **Executing and Monitoring**: Run the workflows and monitor their progress.

**Applications:**

* **Data Warehousing**: Consolidating data from different sources into a central repository.
* **Data Migration**: Moving data from legacy systems to modern databases.
* **Data Cleansing**: Ensuring data quality by removing inconsistencies and errors.