

Db2

Change version

11.5

☒ Show full table of contents

Filter on titles

IBM Db2 version 11.5
documentation

Product overviews

Database fundamentals

Database administration

Db2 data servers

Instances

Databases

Designing databases

Creating databases

Data recovery

Instance management in a Db2
pureScale environment

Db2 workload management

Interfaces (Tools, Commands,
APIs)Developing code for accessing and
managing data

Database reference content

Db2 Connect overview

Db2 for cloud service providers

Db2 containerized deployments

Db2 Migration Service

IBM Cognos Analytics 11.1

Federation

Db2 / 11.5 /

Feedback

Product list

Databases

Last Updated: 2022-11-26

8.1

A Db2® database is a *relational database*. The *database* stores all data in tables that are related to one another. Relationships are established between tables such that data is shared and duplication is minimized.

8.1

A *relational database* is a database that is treated as a set of tables and manipulated in accordance with the relational model of data. It contains a set of objects used to store, manage, and access data. Examples of such objects are tables, views, indexes, functions, triggers, and packages. Objects can be either defined by the system (built-in objects) or defined by the user (user-defined objects).

A *distributed relational database* consists of a set of tables and other objects that are spread across different but interconnected computer systems. Each computer system has a relational database manager to manage the tables in its environment. The database managers communicate and cooperate with each other in a way that allows a given database manager to execute SQL statements on another computer system.

A *partitioned relational database* is a relational database whose data is managed across multiple database partitions. This separation of data across database partitions is transparent to most SQL statements. However, some data definition language (DDL) statements take database partition information into consideration (for example, **CREATE DATABASE PARTITION GROUP**). DDL is the subset of SQL statements used to describe data relationships in a database.

A *federated database* is a relational database whose data is stored in multiple data sources (such as separate relational databases). The data appears as if it were all in a single large database and can be accessed through traditional SQL queries. Changes to the data can be explicitly directed to the appropriate data source.

– Designing databases

When designing a database, you are modeling a real business system that contains a set of entities and their characteristics, or attributes, and the rules or relationships between those entities.

– Creating databases

You create a database using the **CREATE DATABASE** command. To create a database from a client application, call the `sqlcreate` API. All databases are created with the default storage group `IBMSTOGROUP`, unless you specify otherwise. Automatic storage managed table spaces use storage groups for their storage definitions.

– Connecting to distributed relational databases

Distributed relational databases are built on formal requester-server protocols and functions.

– Viewing the local or system database directory files

Use the **LIST DATABASE DIRECTORY** command to view the information associated with the databases that you have on your system.

– Database partitions

A *database partition* is a part of a database that consists of its own data, indexes, configuration files, and transaction logs. A database partition is sometimes called a node or a database node. A partitioned database environment is a database installation that supports the distribution of data across database partitions.

– Buffer pools

A *buffer pool* is an area of main memory that has been allocated by the database manager for the purpose of caching table and index data as it is read from disk. Every Db2 database must have a buffer pool.

– Table spaces

A *table space* is a storage structure containing tables, indexes, large objects, and long data. They are used to organize data in a database into logical storage groupings that relate to where data is stored on a system. Table spaces are stored in database partition groups.

– Storage groups

A storage group is a named set of storage paths where data can be stored. Storage groups are configured to represent different classes of storage available to your database system. You can assign table spaces to the storage group that best suits the data. Only automatic storage table spaces use storage groups.

– Schemas

A *schema* is a collection of named objects; it provides a way to group those objects logically. A schema is also a name qualifier; it provides a way to use the same natural name for several objects, and to prevent ambiguous references to those objects.

– Schema enabled for row modification tracking

– Dropping databases

Dropping a database can have far-reaching effects, because this action deletes all its objects, containers, and associated files. The dropped database is removed (uncataloged) from the database directories.

Was this topic helpful?

Yes



No

© Copyright IBM Corporation
2019, 2021