

Introduction to IBM DB2

Architecture

- Client-server system
- Server: SERVEDB,
 - Internal address (from the lab) 192.168.0.252
 - External address (from home with OpenVPN) 10.17.2.91
- Client:
 - IBM Data Studio: graphical
 - DB2 Command Window: command line

Architecture

- Servers, instances, databases: as in SQL Server
- Objects are contained in databases: table, views, stored procedures
- Objects are divided into schemas
 - As in SQL Server

IBM Data Studio

- Run IBM Data Studio
- Right Click on All Databases -> New connection to a database
- Scegliere DB2 for Windows...
- Database: SAMPLE
- Host: 192.168.0.252
- Port number: 5000
- Username: utente
- Password: Infonew1
- Click su next, next, finish

IBM Data Studio

- In Administration Explore expand 192.168.0.252
- Expand DB2
- Right click su SAMPLE-> connect
- Click on tables
- Right click on Administrator, Department -> Data-> Browse data

IBM Data Studio

- Click on the button new SQL Script
- Click on Click here to connect to a database
- `SELECT EMPNO, FIRSTNME, LASTNAME FROM ADMINISTRATOR.EMPLOYEE`
- Click on green >
- Click on Result

Authentication

- DB2 uses the username and passwords of the client or server operating system
- Types of authentication
 - SERVER: Authentication takes place on the server.
 - SERVER_ENCRYPT: Authentication takes place on the server. Passwords are encrypted at the client machine before being sent to the server.
 - CLIENT: Authentication takes place on the client machine.
 - DATA_ENCRYPT: This operates the same way as SERVER_ENCRYPT, except the data is encrypted as well.
 - Plus others

Server Configuration

- Authentication is part of the configuration of the instance
- The configuration is defined by a number of parameters

Authorities

- Authorities are made up of groups of privileges and higher-level database manager (instance-level) maintenance and utility operations
- Same as roles in SQL Server
- Instance-level Authorities:
 - SYSADM, SYSCTRL, SYSMANT, SYSMON
- Database-level Authorities:
 - DBADM, LOAD, SECADM

SYSADM

- Comparable to root authority on UNIX or Administrator authority on Windows.
- Users with SYSADM authority for a DB2 instance are able to issue any DB2 commands against that instance, any databases within the instance, and any objects within those databases.
- They also have the ability to access data within the databases and grant or revoke privileges and authorities.
- SYSADM users are the only users allowed to update the configuration

SYSADM

- SYSADM authority is assigned to all the users in a group in the operating system
- The group is controlled in the DBM CFG file via the SYSADM_GROUP parameter.
- When the instance is created, this parameter is set to Administrator on Windows. On UNIX, it is set to the primary group of the user who created the instance.
- Since SYSADM users are the only users allowed to update the configuration, they are also the only ones allowed to grant any of the SYS* authorities to other groups.

SYSCTRL

- Users with SYSCTRL authority can perform all administrative and maintenance commands within the instance.
- However, unlike SYSADM users, they cannot access any data within the databases unless they are granted the privileges required to do so.
- Allowed commands:
 - db2start/db2stop
 - db2 create/drop database
 - db2 create/drop tablespace
 - db2 backup/restore/rollforward database
 - db2 runstats (against any table)
 - db2 update db cfg for database dbname

SYSMAINT

- A user with SYSMAINT authority can issue commands that are a subset of those allowed to users with SYSCTRL authority.
- SYSMAINT users can only perform tasks related to maintenance, such as:
 - db2start/db2stop
 - db2 backup/restore/rollforward database
 - db2 runstats (against any table)
 - db2 update db cfg for database dbname

Authorities

- All instance level authorities are assigned to all the users in a group in the operating system
- The group is chosen by setting a configuration parameter

DBADM

- DBADM authority is a database-level authority rather than an instance-level authority.
- DBADM users have complete control over a database
- They can perform the following tasks:
 - db2 create/drop table
 - db2 grant/revoke (any privilege)
 - db2 runstats (any table)

DBADM

- Since DBADM authority is a database-level authority, it can be assigned to both users and groups.
- db2 create database test
- db2 connect to test
- db2 grant dbadm on database to user tst1
 - Only SYSADM can do it
- db2 grant dbadm on database to group db2grp1
 - Only SYSADM can do it

Other DB Level Authorities

- **LOAD:**
 - allows users to issue the LOAD command against a table.
 - The LOAD command is typically used as a faster alternative to insert or import commands when populating a table with large amounts of data.
- **SECADM:**
 - allows users to update the security policies of a database

What makes up a DB2 database?

- From the user's perspective, a database is a collection of tables that are usually related in some way.
- From the perspective of a database administrator it's a little more complicated than that.
- The actual database contains many of the following physical and logical objects:
 - Tables, views, indexes, schemas
 - Locks, triggers, stored procedures, packages
 - Buffer pools, log files, table spaces

Logical, physical, and performance features of a database

- Some of these objects, like tables or views, help determine how the data is organized.
- Other objects, like table spaces, refer to the physical implementation of the database.
- Finally, some objects, like buffer pools and other memory objects, only deal with how the database performance is managed.
- The DBA should first concentrate on the physical implementation of the database.
- How do you create a database and allocate the disk storage required for it?

The DB2 storage model

- DB2 has both a logical and physical storage model to handle data.
- The tables are placed into table spaces.
- A table space is used as a layer between the database and the container objects that hold the actual table data.
- A table space can contain more than one table.
- A container is a physical storage device. It can be identified by a directory name, a device name, or a file name.
- A container is assigned to a table space.

Relationships

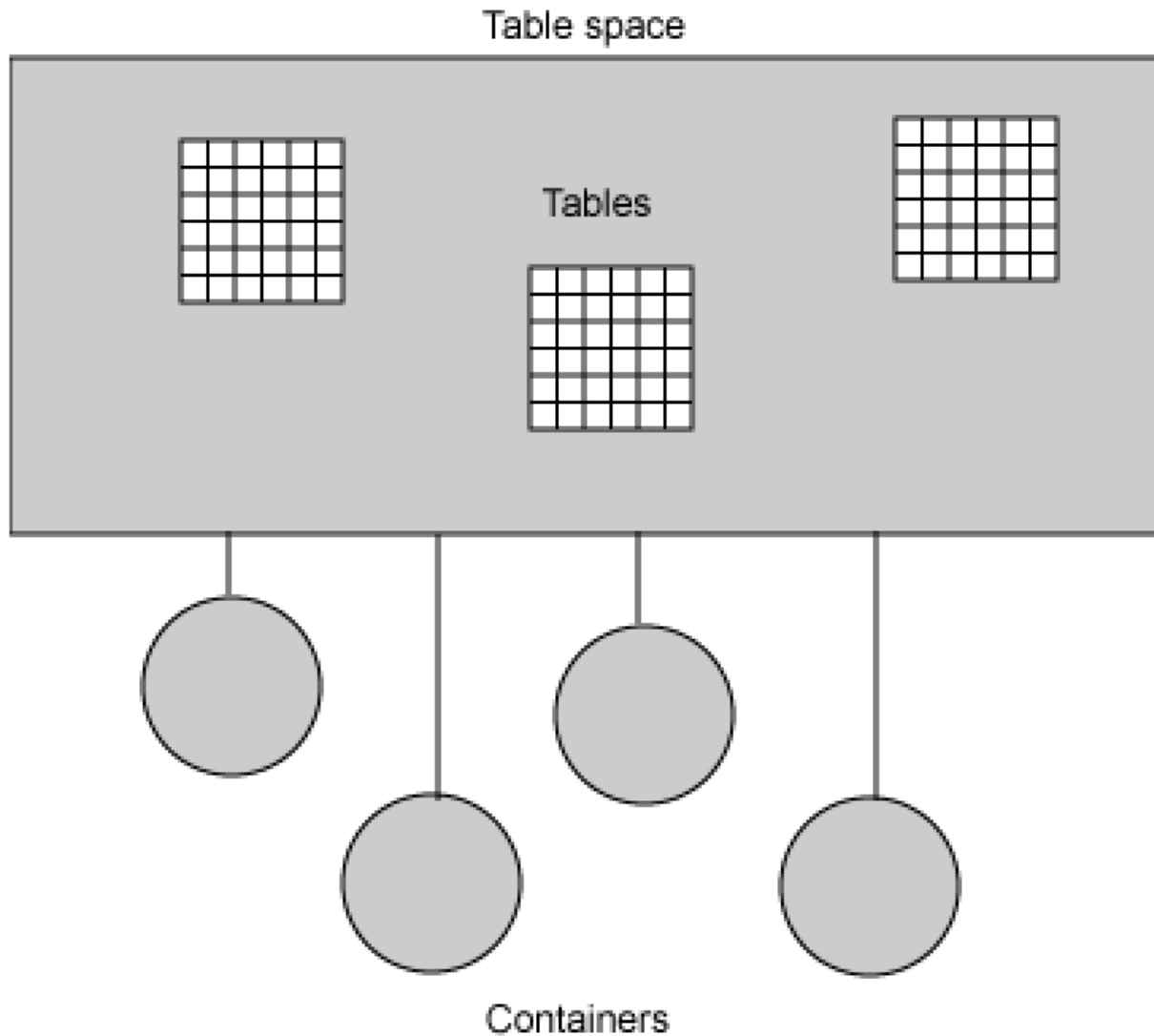
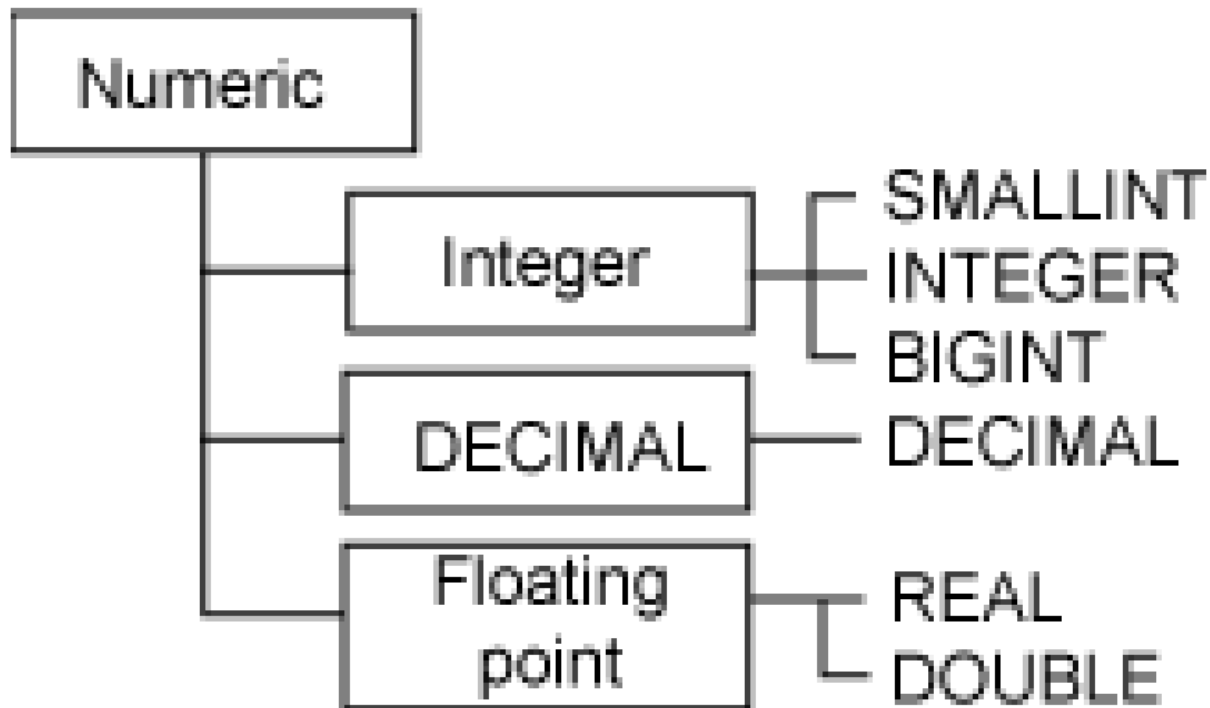


Table space

- A table space can span many containers, which means that you can get around operating system limitations that may limit the amount of data that one container can have.
- Tables, indexes, and long fields (sometimes called binary large objects or BLOBs) are mapped to a table space.

Data Types

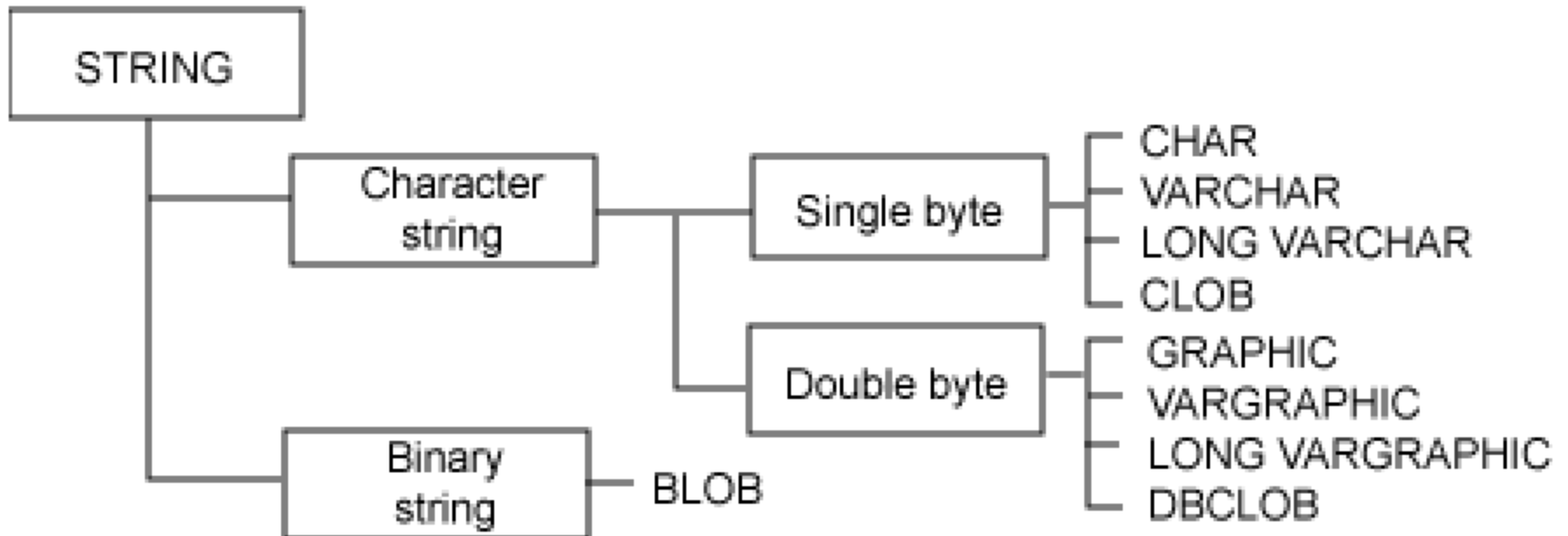
- There are four categories of built-in data types: numeric, string, datetime, and XML.



Numeric

- SMALLINT, INTEGER, and BIGINT: as SQL Server
- DECIMAL: as SQL Server: **decimal**[(p[, s])]
 - Memory: $p/2+1$ bytes
 - DECIMAL(10,2): $10/2+1=6$ bytes
- FLOAT: as SQL Server
- REAL can be defined with a length between 1 and 24 digits and requires 4 bytes of storage.
- DOUBLE can be defined with a length of between 25 and 53 digits and requires 8 bytes of storage.
- FLOAT can be used as a synonym for REAL or DOUBLE.

String Data Types



Single Byte Strings

- CHAR or CHARACTER is used to store fixed-length character strings up to 254 bytes.
- VARCHAR is used to store variable-length character strings.
 - The maximum length of a VARCHAR column is 32,672 bytes.
 - In the database, VARCHAR data only takes as much space as required.

Double Byte Strings

- GRAPHIC is used to store fixed-length double-byte character strings.
 - The maximum length of a GRAPHIC column is 127 characters.
 - Corresponds to NCHAR of SQL Server
- VARGRAPHIC is used to store variable-length double-byte character strings.
 - The maximum length of a VARGRAPHIC column is 16,336 character
 - Corresponds to NVARCHAR of SQL Server

Long Data Types

- Data types to store very long strings of data.
- The data is not stored physically with the row data in the database, which means that additional processing is required to access this data.
- Long data types can be defined up to 2GB in length. However, only the space required is actually used.

Long Data Types

- LONG VARCHAR: as VARCHAR(MAX) of SQL Server
- CLOB (character large object)
- LONG VARGRAPHIC : as NVARCHAR(MAX) of SQL Server
- DBCLOB (double-byte character large object)
- BLOB (binary large object): : as VARBINARY(MAX) of SQL Server

Date and Time

- DATE
- TIME
- TIMESTAMP: corresponds to DATETIME of SQL Server
- The values of these data types are stored in the database in an internal format; however, applications can manipulate them as strings.
- When one of these data types is retrieved, it is represented as a character string.
- Enclose the value in quotation marks when updating these data types.

Date and Time

- DB2 provides built-in functions to manipulate datetime values.
- For example, you can determine the day of the week of a date value using the DAYOFWEEK or DAYNAME functions.
- Use the DAYS function to calculate how many days lie between two dates.
- CURRENT DATE returns a string representing the current date on the system.

XML data type

- DB2 provides the XML data type to store well-formed XML documents.
- Values in XML columns are stored in an internal representation different from string data types.
- To store XML data in an XML data type column, transform the data using the XMLPARSE function.
- An XML data type value can be transformed into a serialized string value representation of the XML document using the XMLSERIALIZE function.

Creating Tables

- Connect to database PROVA on 10.17.2.91
- Creates table BOOKS<matr> in the schema UTENTE

```
CREATE TABLE BOOKS<matr> (  
    BOOKID INTEGER,  
    BOOKNAME VARCHAR(100),  
    ISBN CHAR(10)  
)
```

- Creates table MYBOOKS<matr> in the schema UTENTE with the same schema as BOOKS<matr>

```
CREATE TABLE MYBOOKS<matr> LIKE  
    BOOKS<matr>
```

Altering a Table

- Use the ALTER TABLE SQL statement to change characteristics of a table.
- For instance, you can add or drop:
 - A column
 - A primary key
 - One or more unique or referential constraints
 - One or more check constraints

```
ALTER TABLE BOOKS<matr> ADD BOOKTYPE  
CHAR(1)
```

Altering a Table

- You can also change characteristics of specific columns in a table
- The following statement changes the DATATYPE of column BOOKNAME from VARCHAR(100) to VARCHAR(200) and changes the nullability of the ISBN column to NOT NULL:

```
ALTER TABLE BOOKS<matr> ALTER BOOKNAME  
SET DATA TYPE VARCHAR(200) ALTER ISBN SET  
NOT NULL
```

Dropping a table

- The DROP TABLE statement removes a table from the database, deleting the data and the table definition.
- If there are indexes or constraints defined on the table, they are dropped as well.

```
DROP TABLE BOOKS<matr>
```

```
DROP TABLE MYBOOKS<matr>
```