Introduction to IBM DB2 version 9

Architecture

- Client-server system
- Server: SERVEDB,
  - Internal addess (from the lab) 192.168.0.252
  - External address (from home with OpenVPN) 10.17.2.91
- Client:
  - Control Center (Centro di Controllo): 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

Client Directory

- The client maintains a list of servers, instances and databases called “directory”
- It is used to assign a logical name to servers, instances and databases to be used for connecting
- To connect to an instance/database it is necessary to add it first to the directory
- Two methods
  - Graphical interface
  - Command line

Graphical Configuration

- Start “Strumenti di configurazione\Assistente di Configurazione”
- “Si desidera aggiungere un database?” ->si
- Or from the menu “Selezionati” choose “Aggiungi database con il wizard”
- Configura manualmente una connessione ad un database
- TCP/IP
- Nome host: 10.17.2.91 (192.168.0.252 from lab), Numero porta: 50000
- Nome database: SAMPLE
- Avanti
- Sistema operativo: NT
- Fine

Connection

- Start the Centro di Controllo
- Choose Vista avanzata
- Expand “Tutti i sistemi”
- Insert username utente and password Infonew1
- Expand Istanze
- Expand the underlying icon (something like NDEA4E07 (DB2))
- Expand Database
- Expand Sample
- Insert username utente and password Infonew1

Graphical Configuration (advanced)

- Start Strumenti di configurazione\Assistente di Configurazione
- Menu “VIsualizza”->”Vista avanzata”
- Sistemi tab
- Menu “Selezionati”->“Aggiungi sistema”
- Nome sistema: SERVEDB (<a piacere1>)
- Nome host: 10.17.2.91
- Nome nodo: SERVEDB (<a piacere1>)
- ok

Graphical Configuration (advanced)

- Nodi delle istanze tab
- Menu “Selezionati”->“Aggiungi nodo istanza”
- Nome sistema: SERVEDB
- Nome istanza: Rilevamento-> DB2
- Nome nodo istanza: (DB2SERV) <a piacere2>
- ok

Graphical Configuration (advanced)

- Database tab
- Menu “Selezionati”->“Aggiungi database”
- Nodo di istanza: DB2SERV (<a piacere2>)
- Nome del database: rilevamento-> SAMPLE
- Alias: SAMPLEDB (<a piacere3>)
- ok

Connection

- Start the Centro di Controllo
- Choose vista avanzata
- Expand “Tutti i sistemi”
- Insert username utente and password Infonew1
- Expand Istanze
- Expand l’icona DB2SERV (<a piacere2>)
- Expand Database
- Expand SAMPLEDB (<a piacere3>)
- Insert username utente and password Infonew1

Centro di Controllo

- Expand Tabelle
- Select EMPLOYEE
- The table schema is shown in the detail panel (lower right)
- Right click on EMPLOYEE-> Apri
- The content of the table is shown
- Right click on EMPLOYEE-> Interroga
- The command editor is opened
- You can write a SQL query
- Submit it pressing the Esegui button (play symbol)

Centro di controllo

- Right click -> Modifica
- To modify the schema of a table
- Right click -> Ridenomina
- To change the name
- Right click -> Cancella
- To delete the table (DROP)

Command Line

- Two approaches:
  - DB2 Command window: a shell window with the possibility of entering DB2 commands by preceding them by the keyword db2. Run from the Start menu or by entering db2cmd in a windows command window
  - Command Line Processor: a command interpreter that is run either from the Start menu or by entering db2 on a DB2 command window.
    - Same commands as DB2 Command Window but without the preceding db2
    - All the example are for CLP

Obtaining Help

- ? <COMMAND>
  - Shows a textual summary of the command syntax
- HELP
  - Opens a web page on the IBM servers

Connecting to an Instance

- ATTACH TO <INSTANCE-NAME>
  - Uses as username the one used for logging into the system
- ATTACH TO DB2SERV
- ATTACH TO <INSTANCE-NAME> USER <USERNAME>
- ATTACH TO DB2SERV USER UTENTE
- DETACH
  - To disconnect

Connecting to a database

- CONNECT TO <DATABASE>
  - Uses as username the one used for logging into the system
- CONNECT TO SAMPLE
- CONNECT TO <DATABASE> USER <USERNAME> USING <PASSWORD>
- CONNECT TO SAMPLE USER UTENTE USING Infonew1
- CONNECT RESET
  - To disconnect

Issuing SQL Commands

From CLP:

```
db2 => SELECT EMPNO,FIRSTNME,LASTNAME FROM
    ADMINISTRATOR.EMPLOYEE

EMPNO  FIRSTNME     LASTNAME
------ ------------ --------------
000010 CHRISTINE    HAAS
000020 MICHAEL      THOMPSON
000030 SALLY        KWAN
000050 JOHN         GEYER
000060 IRVING       STERN
000070 EVA          PULASKI
000090 EILEEN       HENDERSON
.....
```

Issuing SQL Commands

From CW:

DB2 "SELECT EMPNO,FIRSTNME,LASTNAME FROM ADMINISTRATOR.EMPLOYEE" > employee.txt\

- With CW you can redirect the input and output of the commands

Connection Configuration with the Command Line

- Use the CATALOG command:
- For the instances:
- CATALOG TCPIP4 NODE <ALIAS-NODE> REMOTE <IP-NUMBER> SERVER <PORT NUMBER>
- CATALOG TCPIP4 NODE SERVLAB REMOTE 10.17.2.91 SERVER 50000
- For the databases:
- CATALOG DATABASE <DATABASE-NAME> [AS <ALIAS>] AT NODE <NODE-NAME>
- CATALOG DATABASE SAMPLE AS SAMPLE1 AT NODE SERVLAB

Removing an Entry

- To remove an entry from the node/database directory
  - UNCATALOG NODE <NODE>
  - UNCATALOG DATABASE <DB>

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

Authentication

- Authentication is set on the server at the level of the instance for the ATTACH command and at the level of the database for the CONNECT command
- The client authentication setting in the directory entry for the instance/database being connected to must match that of the database server

Server Configuration

- Authentication is part of the configuration of the instance
- The configuration is defined by a number of parameters
- GET DBM CFG
  - Returns the list of configuration parameters together with their value

Authentication

- On SERVEDB:

GET DBM CFG

…..

Autenticazione connessione server (SRVCON_AUTH) = NOT_SPECIFIED

Autenticazione Database manager (AUTHENTICATION) = SERVER_ENCRYPT

…..

- Autenticazione Database manager: authentication for attaching to instances
- Autenticazione connessione server: authentication for connecting to databases. Not specified-> = AUTHENTICATION

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, SYSMAINT, 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 DBM CFG file.

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 (although it appears blank if you issue the command db2 get dbm cfg). 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 DBM CFG, they are also the only ones allowed to grant any of the SYS* authorities to other groups.
- The following example illustrates how to grant SYSADM authority to the group db2grp1:

```
DB2 UPDATE DBM CFG USING SYSADM_GROUP db2grp1
```

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 DBM CFG 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

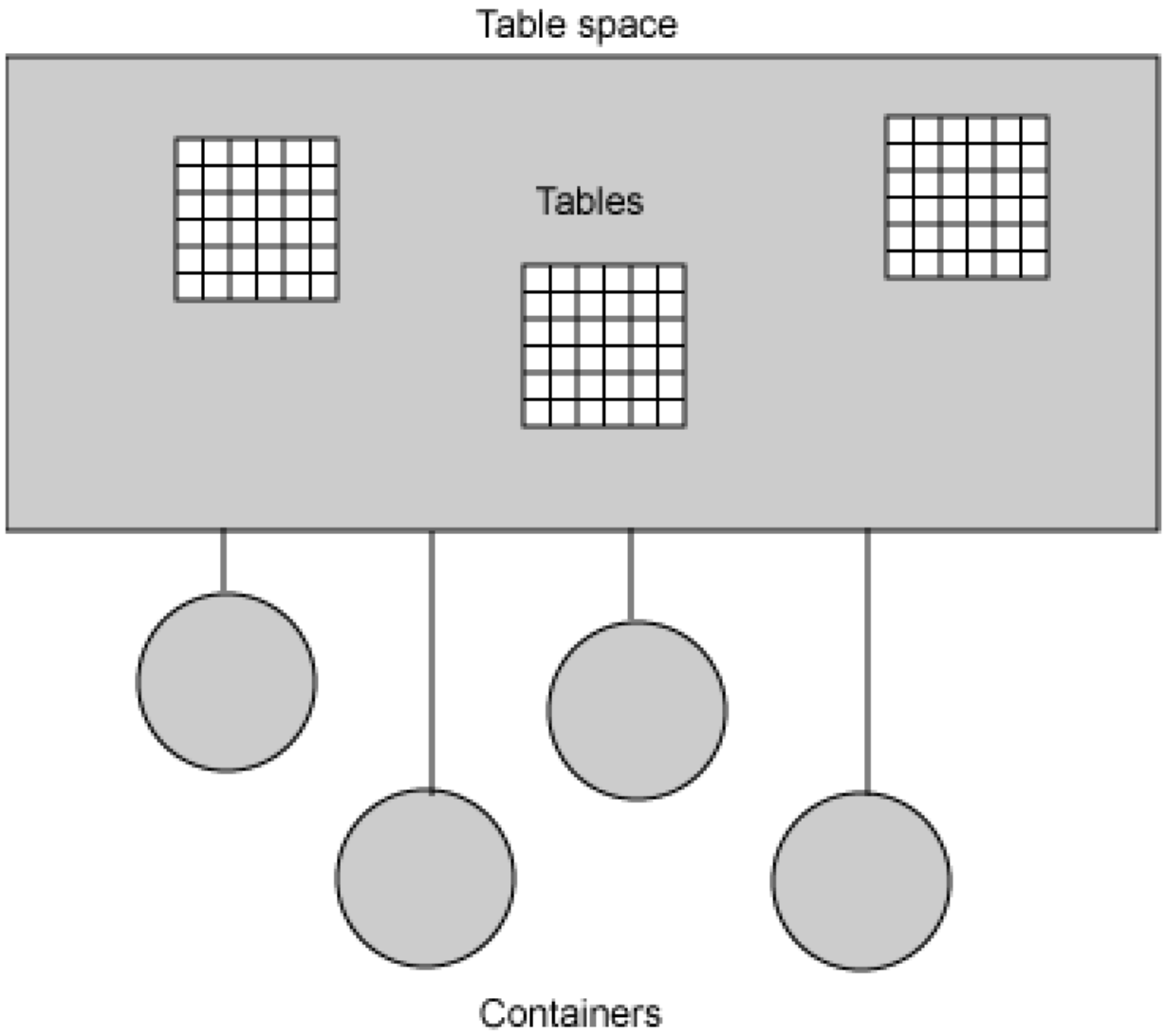


Table space

- 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.

Kinds of table spaces

- System-Managed Space (SMS): Here, the operating system's file system manager allocates and manages the space.
- Database-Managed Space (DMS): Here, the database manager controls the storage space. This table space is, essentially, an implementation of special-purpose file system designed to best meet the needs of the database manager.
- Automatic Storage With DMS: Automated storage is not really a separate type of table space, but a different way of handling DMS storage. DMS containers require more maintenance
- SMS table spaces require very little maintenance. However, SMS table spaces offer fewer optimization options and may not perform as well as DMS table spaces.

Creating a database

- When you create a database, information about the database including default information is stored in a directory hierarchy.
- The hierarchical directory structure is created for you at a location that is determined by the information you provide in the CREATE DATABASE command.
- If you do not specify the location of the directory path or drive when you create the database, the default location is used.

Creating a database

- In the directory you specify as the database path in the CREATE DATABASE command, a subdirectory that uses the name of the instance is created.
- This subdirectory ensures that databases created in different instances under the same directory do not use the same path.
- Below the instance-name subdirectory, a subdirectory named NODE0000 is created.
- This subdirectory differentiates database partitions in a logically partitioned database environment.

Creating a database

- Below the node-name directory, a subdirectory named SQL00001 is created.
- SQL00001 contains objects associated with the first database created, and subsequent databases are given higher numbers: SQL00002, and so on.
- These subdirectories differentiate databases created in this instance on the directory that you specified in the CREATE DATABASE command.

Files

- The database directory contains a number of files that are created as part of the CREATE DATABASE command.
  - Data files
  - Configuration files

Creating a database

- From Control Center, right click on databases> create database
- From CLP or CW: use CREATE DATABASE
- Utente does not have the rights
- Use the database PROVA
- Utente is the DBADM of PROVA

Creating a database example

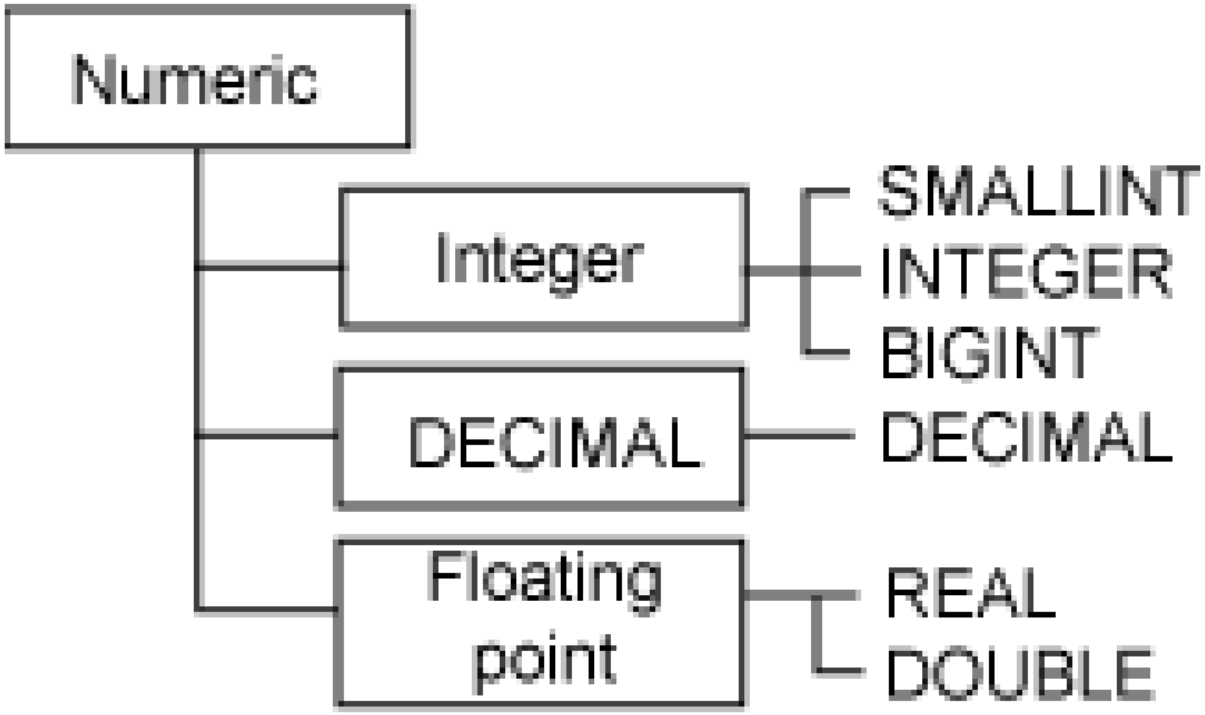
```
CREATE DATABASE DB1
CREATE DATABASE DB2 ON X:
CREATE DATABASE DB3 ON /data/path1, /data/path2
CREATE DATABASE DB4 ON D:\StoragePath
```

Connecting to PROVA

```
CATALOG DATABASE PROVA AT NODE DB2SERV
CONNECT RESET
CONNECT TO PROVA USER UTENTE
```

Data Types

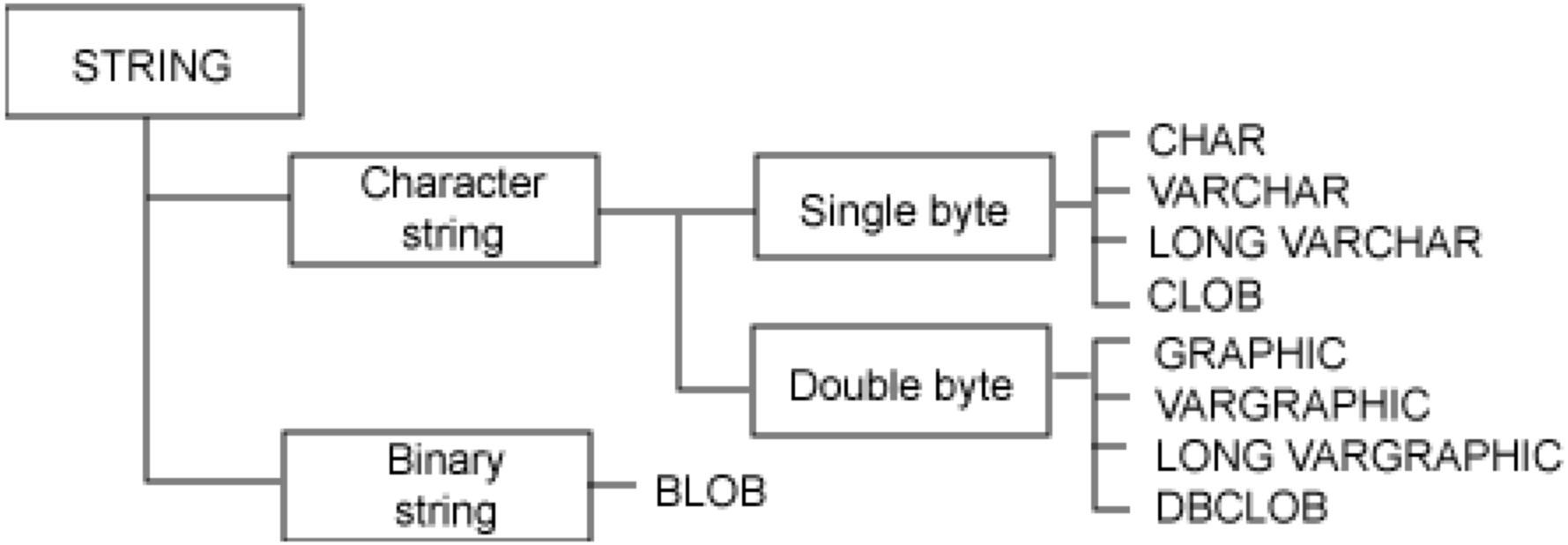
- There are four categories of built-in data types: numeric, string, datetime, and XML.



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.

Date and Time

- The format of the date and time values depends on the country code of the database, which is specified when the database is created.
- There is a single format for the TIMESTAMP data type. The string representation is
  - yyyy-mm-dd-hh.mm.ss.nnnnnn

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.

User-defined data types

- User-defined distinct:
  - Define a new data type based on a built-in type.
  - This new type has the same features of the built-in type, but you can use it to ensure that only values of the same type are compared.
- User-defined structured:
  - Create a type that consists of several columns of built-in types.
  - Then use this structured type when creating a table as the type of a column.
  - For example, you can create a structured type named ADDRESS.
  - Structured types can have subtypes in a hierarchical structure.

Complex, nontraditional data types

- Supported by DB2 Extenders, software packages separated from the DB2 server code and installed on the server and into each database that uses the data type.
- Examples:
  - DB2 Image Extender for images
  - DB2 Text Extender for texts
  - DB2 Spatial Extender for storing and analyzing spatial data

Creating Tables

- Creates table BOOKS in the schema UTENTE

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

- Use CREATE SCHEMA to create a schema
- Creates table MYBOOKS in the schema UTENTE with the same schema as BOOKS

```
CREATE TABLE MYBOOKS LIKE BOOKS
```

Creating Tables

```
CREATE TABLE <TABLE> (
…
IN <TABLESPACE>

CREATE TABLE BOOKS (
    BOOKID INTEGER,
    BOOKNAME VARCHAR(100),
    ISBN CHAR(10) )
IN TBPROVA
```

System Catalog Tables

- A database has a set of tables, called the system catalog tables, which hold information about all the objects in the database.
- DB2 provides views for the base system catalog tables.
- The catalog view SYSCAT.TABLES contains a row for each table defined in the database.
- SYSCAT.COLUMNS contains a row for each column of each table in the database.
- Look at the catalog views using SELECT statements, just like any other table in the database; however, you cannot modify the data using INSERT, UPDATE, or DELETE statements.

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 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 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
```