



IBM Software Group

# DB2 Express-C 9 overview course

*Working with database objects*

**DB2** Information Management Software

A horizontal decorative bar spanning the width of the slide, composed of various colored squares and rectangles in shades of green, yellow, red, and blue.

**ON DEMAND BUSINESS™**

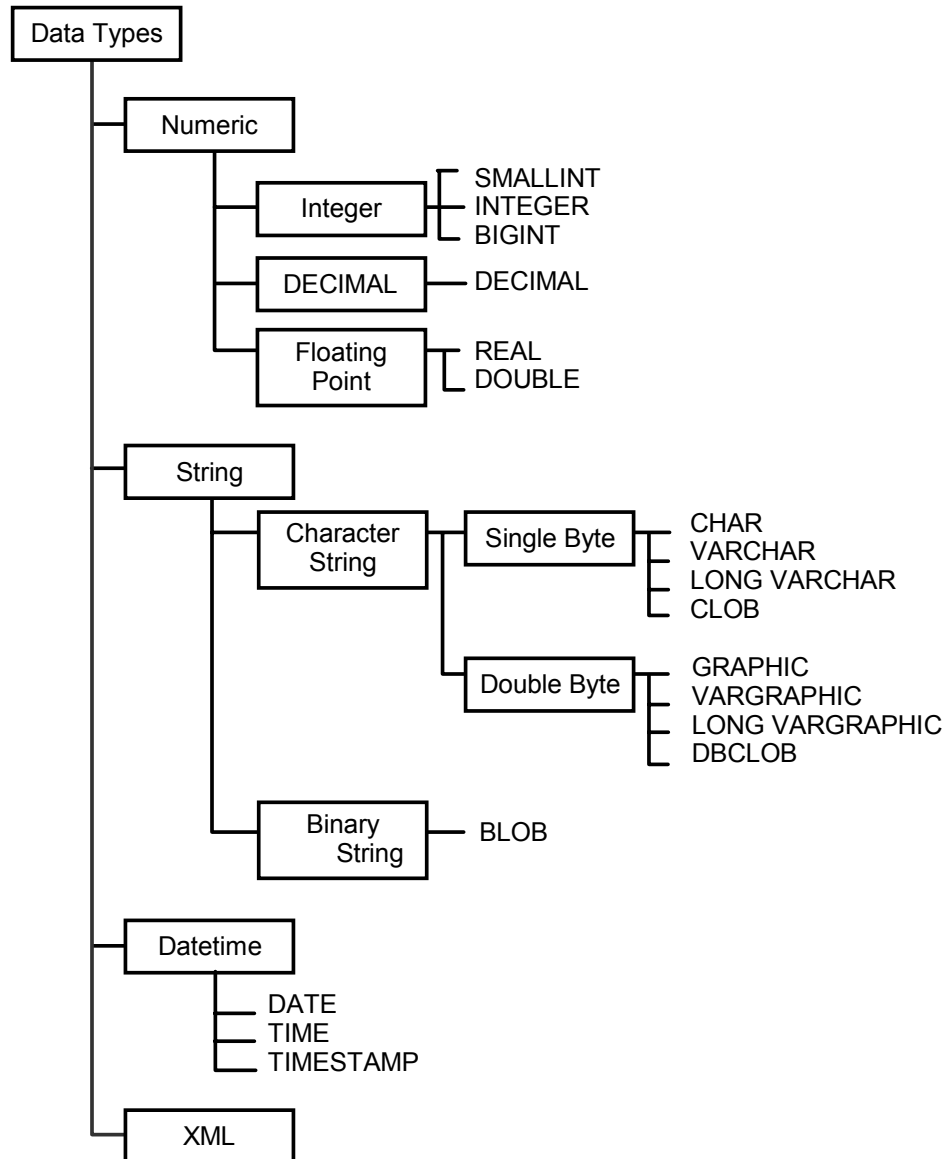
# Creating Tables

```
CREATE TABLE artists
(artno          SMALLINT      not null,
 name          VARCHAR(50)    with default 'abc',
 classification CHAR(1)       not null,
 bio           CLOB(100K)     logged,
 picture       BLOB(2M)       not logged compact
)
in mytbls1
```



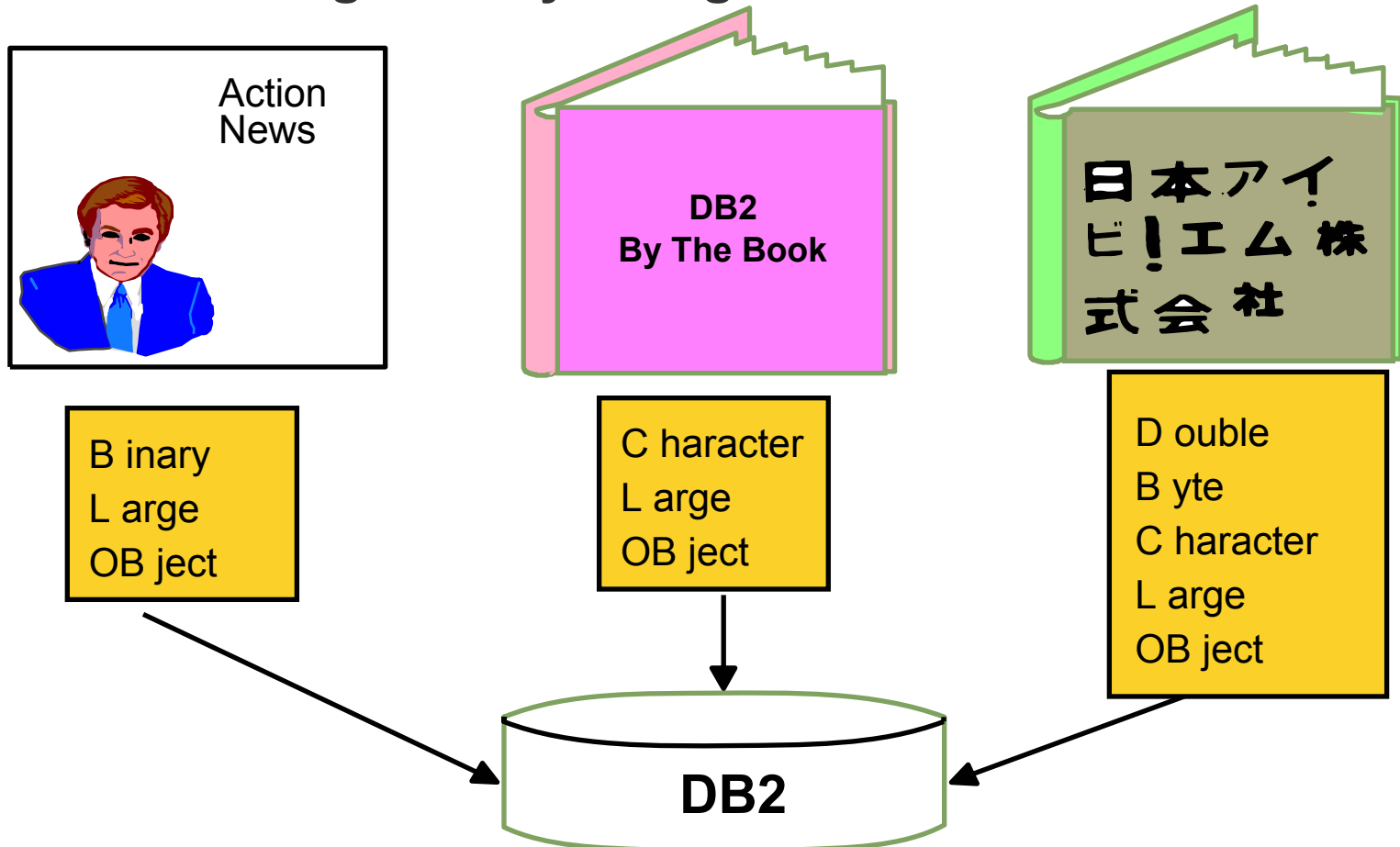
# Data Types

- DB2 Data Types



# Large Objects

- To store large character strings or files
- To store large binary strings or files



## Concept - *Schema*

- Schemas are *name spaces* for a collection of database objects
- Schemas are primarily used to:
  - ▶ Provide an indication of object ownership or relationship to an application
  - ▶ Logically group related objects together
- All database objects belong to schemas and are *qualified* by a two-part name:

**<schema\_name>.<object\_name>**

- ▶ A fully qualified object name must be unique
- When you connect to a database and create or reference an object without specifying the schema, DB2 uses the user ID you connected to the database with for the schema name



# System Catalog Tables

- Each database has its own system catalog tables/views
- These store *meta data* about the *database* objects
- You can query these tables just like any other tables
- Reside in three schemas:
  - ▶ SYSIBM - base tables, optimized for DB2
  - ▶ SYSCAT - views based on SYSIBM tables, optimized for ease of use
  - ▶ SYSSTAT - database statistics

examples:

- SYSCAT.TABLES
- SYSCAT.INDEXES
- SYSCAT.COLUMNS
- SYSCAT.FUNCTIONS
- SYSCAT.PROCEDURES



# Identity Columns

- A numeric column in a table which automatically generates a unique numeric value for each row that is inserted
- One Identity column per table maximum
- Values can be generated by DB2 always or by default
  - ▶ Generated always
    - values are always generated by DB2
    - applications are not allowed to provide an explicit value.
  - ▶ Generated by default
    - values can be explicitly provided by an application or if no value is given, then DB2 generates one
    - DB2 cannot guarantee uniqueness
    - intended for data propagation, unload/reload of a table



# SEQUENCE objects

- Unlike identity columns, sequences are independent of tables
- example:

```
CREATE SEQUENCE myseq  
START WITH 1  
INCREMENT BY 1  
NO CYCLE
```

```
INSERT INTO t1 VALUES (nextval for myseq, ...)
```

```
SELECT prevval for myseq FROM sysibm.sysdumy1
```





# Creating Views

- Data for view not stored separately
- Nested view supported
- View information kept in: SYSCAT.VIEWS, SYSCAT.VIEWDEP, SYSCAT.TABLES

```
CONNECT TO MYDB1
```

```
CREATE VIEW MYVIEW1  
AS SELECT ARTNO, NAME, CLASSIFICATION  
FROM ARTISTS
```

```
SELECT * FROM MYVIEW1
```

ARTNO	NAME	CLASSIFICATION
-----	-----	-----
10	HUMAN	A
20	MY PLANT	C
30	THE STORE	E
...		

# Creating Indexes

## ■ Index Characteristics:

- ▶ ascending or descending
- ▶ Unique or non-unique
- ▶ compound
- ▶ cluster
- ▶ bi-directional

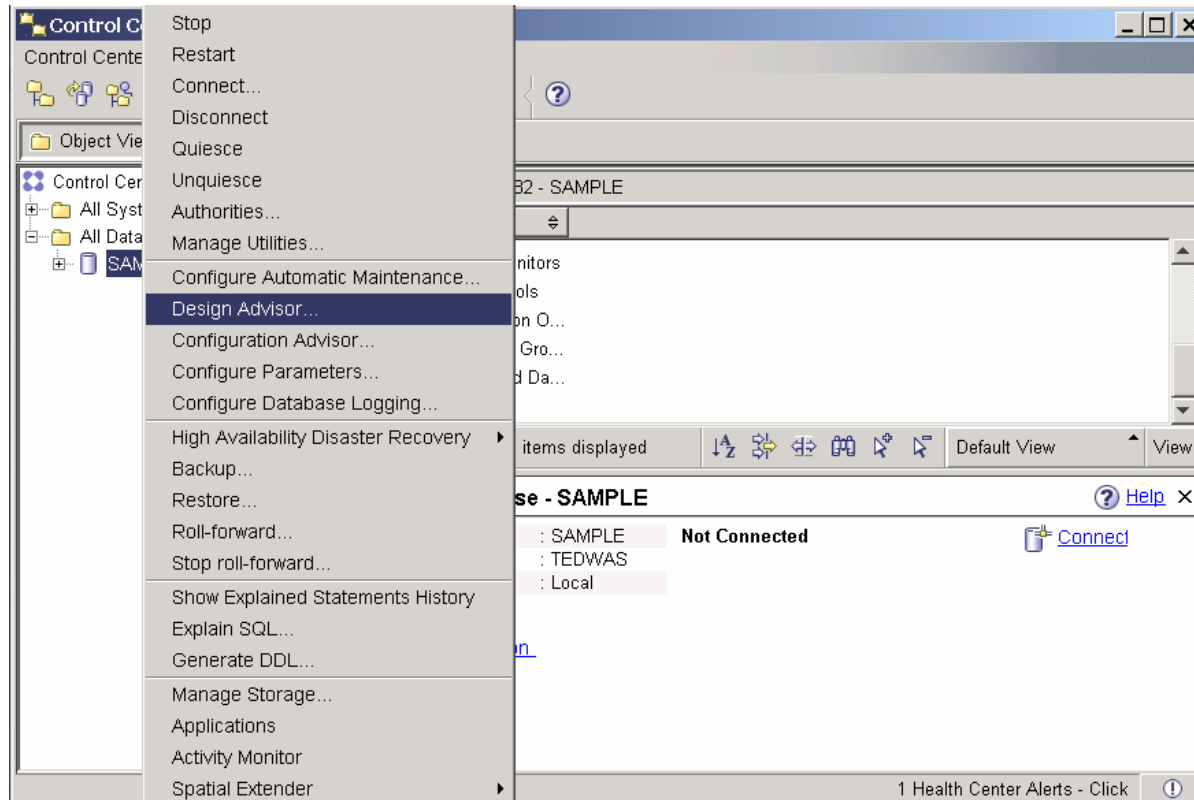
## ■ Examples:

create unique index artno\_ix on artists (artno)



# Launching the Design Advisor

- Advises you on the design of your database to optimize it for a given SQL workload



Control Center > (expand) All Databases Folder > (right-click) Database > Design Advisor

# Referential Integrity

DEPARTMENT table (Parent table)

DEPTNO (Primary key) or unique constraint	DEPTNAME	MGRNO
---	----------	-------

EMPLOYEE table (Dependent table)

EMPNO (Primary key)	FIRSTNAME	LASTNAME	WORKDEPT (Foreign key)	PHONENO
------------------------	-----------	----------	---------------------------	---------

```
create table employee (empno .....  
    primary key (empno)  
    foreign key (workdept)  
    references department on delete no action)  
in DMS01
```



# Referential Integrity Rules

## ■ Insert Rules

- ▶ Rule is implicit when a foreign key is specified.
- ▶ backout insert if not found

## ■ Delete Rules

- ▶ Restrict
  - Parent row not deleted if dependent rows are found.
- ▶ Cascade
  - Deleting row in parent table automatically deletes any related rows in dependent tables.
- ▶ No Action (default)
  - Enforces presence of parent row for every child after all other referential constraints applied
- ▶ Set Null
  - Foreign key fields set to null; other columns left unchanged.

