

PROJECT 2 (32 points)

In this project you will implement the relational database MUSEUM. You find below the description of the database and the ER model associated.

First you have to design the relational database MUSEUM based on them. Second you implement the relational database MUSEUM in your DB2 account in C3421M.

Database description.

You are supposed to design a database to keep track of information for an art museum. Assume that the following requirements were collected:

- The museum has a collection of art objects. Each art object has a unique identification, an artist (if known), a year (when it was created, if known), a title, an epoch (Renaissance, Modern, Ancient, etc.), and a description.
- The art objects are classified in the museum based on their type. The main types are painting and sculpture, plus another type they called "other" to accommodate objects that do not fall into one of the two main types.
- A painting has a paint type (oil, watercolor, etc.), material on which it is draw on (paper, canvas, wood, etc.), and style (baroque, renaissance, modern, realism, abstract, impressionism, etc.).
- A sculpture has a material from which it was created (wood, stone, iron, etc.), height, weight, and style (baroque, renaissance, modern, realism, abstract, impressionism, etc.).
- An art object in the other category has a type (print, photo, clay, etc.) and a style (modern, abstract, etc.).
- Art objects are also categorized as permanent collection that are owned by the museum (which has information on the date acquired, whether it is on display or stored, and cost) or borrowed from another collection. When an art object is borrowed, the museum must store the date when the art object is borrowed, and the date when it is returned. It is possible that the same art object is borrowed many distinct times by the museum from the same collection. We will store in the database only the collections from which the museum borrowed some art object.
- Information is kept on collections with which the museum interacts, including collection's id, name (unique), type (museum, personal, etc.), description, address, phone, and current contact person's name.
- The museum keeps track of artist's information, if known: artist's id (five digits), name, date of birth, date of death (if not living), country of birth, epoch, main style, and a short biography. We store in the database information about artists having their art objects owned or borrowed by the museum.
- Different art exhibitions occur, each having an exhibition's id, name, a place, a start date, and an end date. The museum uses to display its art objects during such exhibitions. For each art object displayed in an exhibition we need to keep track of its display start date and display end date which can be different from the start and end date of the exhibition.

Entities and relationships sets

Entity sets

Art object

a unique identification,
a year (when it was created, if known),
a title,
an epoch (Renaissance, Modern, Ancient, etc.)
a description.
a style (modern, abstract, impressionism, etc.)..

ArtID; varchar(5)
ArtYear; smallint
ArtTitle ; varchar(20)
ArtEpoch ;varchar(15)
ArtDescr ;varchar(50)
OStyle; varchar(15)

Arts objects are categorized based on their type. IsA specialization

A painting

paint type (oil, watercolor, etc.),
material on which it is draw on (paper, canvas, wood, etc.)

Ptype; varchar(15)
PMat; varchar(15)

A sculpture

material from which it was created (wood, stone, iron, etc.),
height,
weight,

SMat; varchar(15)
SHeight; Real
SWeight; Real

An art object in the other category

type (print, photo, etc.)

OType; varchar(10)

Art objects are also categorized as

permanent (that are owned by the museum)

date acquired,
on display or stored,
cost

ArtDate; Date
ArtState; varchar(7) ∈ {"DISPLAY", "STORED"}
ArtCost; Real

borrowed,

the collection (from which it was borrowed)
date borrowed
date returned.

DateBorrow; Date
DateReturn; Date

Artist

Artist Id
Name,
date of birth,
date of death (if not living),
country of birth,
epoch,
main style,
short biography

A_Id; Varchar(7)
AName; Varchar(25)
ABirth; Date
ADeath; Date (possibly Null)
ACountry; Varchar(15)
AEpoch; Varchar(20)
AStyle; Varchar(15)
ABiogrph; Varchar(15)

Exhibition

Exhibition Id
a name (unique),
a place,
a start date,
end date

E_Id; Varchar(7)
ExibName; Varchar(15)
ExibPlace; Varchar(30)
ExibStart; Date
ExibEnd; Date

Other collections

Collection Id
name (unique),
type (museum, personal, etc.),
description,

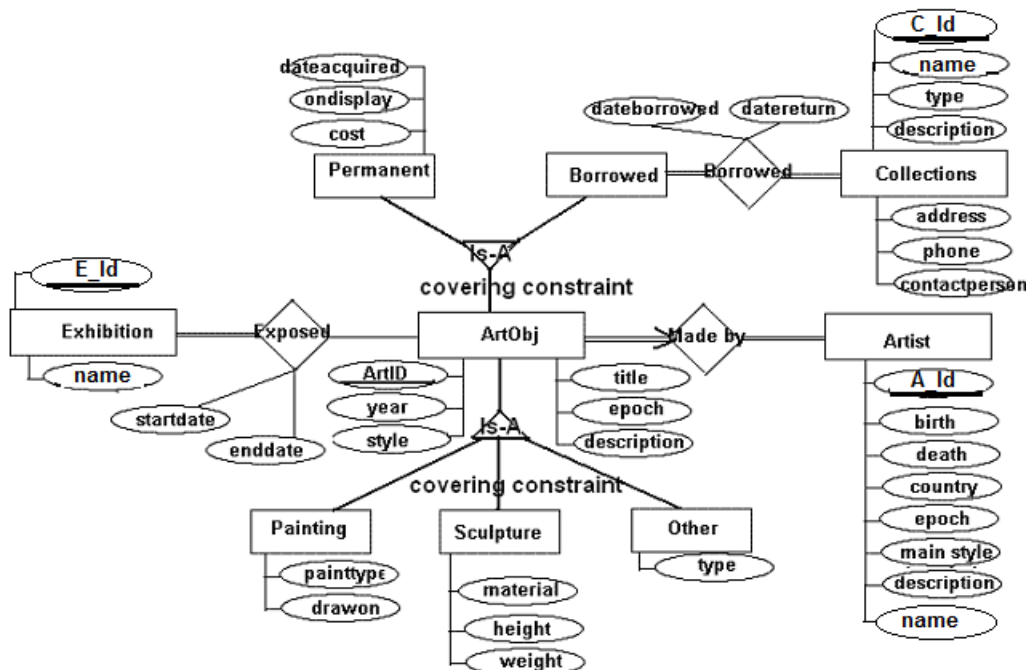
Col_Id; Varchar(7)
ColName; Varchar(15)
ColType; Varchar(20)
ColDescr; Varchar(50)

address,
phone,
current contact person.

ColAdr; Varchar(20)
ColPhone; Varchar(10)
ColContact; Varchar(20)

Relationship sets

- An art object is **made by** an artist; 1:M; key constraint on the art object side, total participation on both sides.
One makes the assumption that we have in the database only artists having their art objects in the museum (permanent or borrowed), hence artist total participation.
- The museum **borrow**s art objects from other collections; descriptive attributes: the date it was borrowed and the date it is returned; M:M ; consider only collections having art objects borrowed by the museum, hence total participation both sides. An art object can be borrowed (and returned) many times from a collection.
- ArtObj **exposed** in an exhibition; descriptive attributes: the start day and end day of the object exposed in an exhibition (which can be different from the start date and end date of the exhibition); M:M; total participation on exhibitions side, and partial participation on ArtObj side. One art object can be exposed in the same exhibition many times.



ER diagram for the MUSEUM database

You will create the relational database MUSEUM. The tables of the database must be created using the list of entity sets, relationship sets and ER diagram above.

You find associated to each attribute in the ER model an attribute name and description of its domain. Use the information above to create the tables (do not forget the rules studied in course concerning the creation of tables of a relational model from the ER model with constraints). Add constraint ON DELETE CASCADE on foreign key introduced.

Write down in the file **project2_3.doc** the ERM constraints you were not able to include in your relational model (explain why).

Data

You must follow the indications below concerning data to be inserted in your tables. The identification of tuples in tables, primary key must be as it is presented. It is your job to complete and/or associate the correct data to the right attribute. **If there is no specification of a particular value for an attribute, then it is your choice.** Replace **MK** by your initials; replace **"Your Name"** by your name.

The date is presented as: MM/DD/YYYY

Information about artists:

- A_Id="MK_A1", birth year =1657, born in Holland, death year 1702
- A_Id="MK_A2", birth year =1980, born in Canada, still alive
- A_Id="MK_A3", birth year 1978, born in Canada, still alive
- A_Id = "unknown"

Information about art objects:

- ArtId="MK01", painting, oil, artist=MK_A1, year =1700, style1, paint on canvas, borrowed from collection MK_C1 (12/23/2004 to 12/02/2005, 12/23/2005 to 11/02/2006, 2/5/2009 to 3/4/2009), exposed on exhibition MK_E01 (2/5/2009 to 3/4/2009)
- ArtId="MK02", painting, oil, artist=MK_A1, year=1710, style1, paint on wood, exposed in exhibition MK_E02 (11/24/2007 to 12/24/2007); bought in 12/30/1980, cost=5.5E6, in display
- ArtId="MK03", sculpture, artist=MK_A2, year=2000, style2, made from stone, height=2.3, weight=234.5, exposed on exhibition MK_E02 (2/2/2008 to 4/25/2008); bought in 12/12/1978, cost=2.3E4, in display
- ArtId="MK04", sculpture, artist=MK_A2, year=2000, style2, made from wood, height 12, weight=55.5, exposed on exhibition MK_E02 (2/2/2008 to 4/25/2008); bought in 12/23/2000, cost=2e5, in display
- ArtId="MK05", other, artist=MK_A2, year=2007, style3,photo, exposed on exhibition MK_E02 (2/2/2008 to 4/25/2008); bought in 12/23/2007, cost=2e5, stored
- ArtId="MK06", other, artist=MK_A2, year=2008, style3,video, exposed on exhibition MK_E03 (1/1/2009 to 2/28/2009, 3/30/2009 to 4/30/2009); bought in 12/12/2007, cost=2e7, stored
- ArtId="MK07", painting, artist unknown, year=1600, style2, oil paint on wood, borrowed from collection MK_C2 twice (2/2/2005 to 2/3/2006, 12/2/2007 to 12/12/2008), exposed at MK_E02 (12/2/2007 to 2/2/2008).
- ArtId="MK08", sculpture, artist=MK_A3, 2000, style4, steel, height=0.5, weight=2.5, borrowed from collection MK_C3 once (12/30/2006 to 12/30/2008).
- ArtId="MK09", other, artist=MK_A3, 2002, style4, photo, borrowed from collection MK_C3 during the same period as MK08.

Information about Collections

- C_Id= "MK_C1", private, Contact="Anissa Maw"
- C_Id="MK_C2", museum, Contact="Hector Berlioz"
- C_Id="MK_C3", museum, Contact= "Your name"

Information about Exhibitions

- E_Id="MK_E01", place Hamilton, 01/23/2009 to 5/4/2009
- E_Id="MK_E02", place Toronto, 11/24/2007 to 5/5/2008
- E_Id="MK_E03", place Toronto, 1/1/2009 to 5/30/2009

You have to upload 3 files: **project2_1.txt (21.5 p)**, **project2_2.txt (7.5p)**, and **project2_3.doc (3p)**

- In the file **project2_1.txt** you present the creation of all tables and all the insert data associated.
Do not forget:

update command options using v ON

update command options using z on project2_1.txt

The file project2_1.txt is not supposed to include all the mistakes during the implementation.

Please, delete everything which is not "CREATE TABLE ..." (correct) and "INSERT INTO ..." (correct).

- In the file **project2_2.txt** you will present the content of your tables; one "SELECT ..." statement for each table. Be aware that the line is 80 characters long and if your tuple has more than 80 characters you have to use **CAST() function** for some attributes in the table to display **one line per tuple**.

Do not forget:

update command options using v ON

update command options using z on project2_2.txt

- The first statement in the file **project2_2.txt** must be "LIST TABLES" (and the result).
- In the file **project2_3.doc** you write down the constraints in ERM that you were not able to include in the relational model (they will be solved later, using triggers or stored procedures).

Hint:

cast-specification:

```
|--CAST--(--+expression-----+--AS--data-type----->
      +-NULL-----+
      '-parameter-marker-'

>--+-----+--)------|
      |      (1)      |
      '-SCOPE-----+--typed-table-name-+-'
                        '-typed-view-name--'
```

Examples:

```
SELECT J_NBR, J_NAME, CAST(J_DESCR AS VARCHAR(25))AS DESCR FROM F_JOB
SELECT EMPNO, CAST(SALARY AS INTEGER)AS SALARY FROM EMPLOYEE
```