Progetto Basi di Dati

by

Nicola Ricci Maccarini & Andrea Casagrande

Dipartimento di Matematica e Informatica

Laurea Triennnale in Informatica

Università degli Studi di Ferrara

July 2024

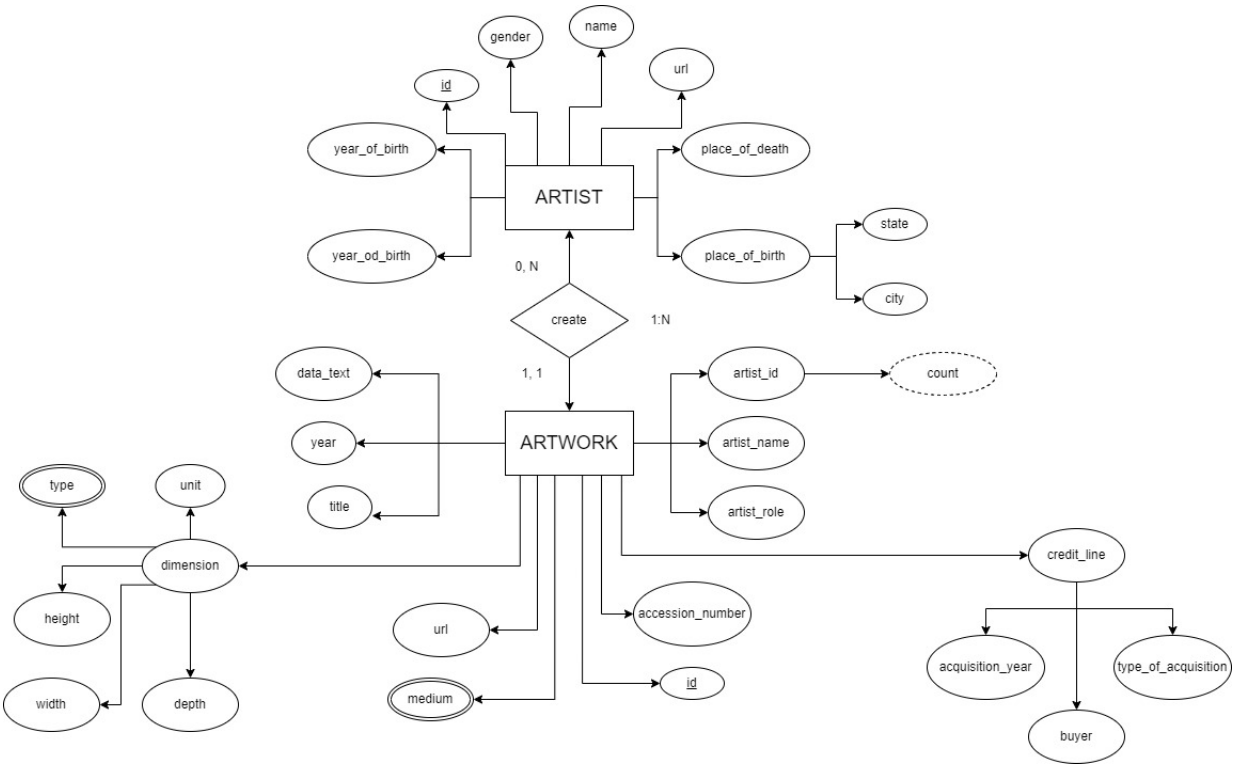
Indice

Capitolo 1

Analisi e inserimento dei dati

1.1 Schema ER

Per prima cosa abbiamo costruito il diagramma ER (Entity-Relationship).
Questo ci ha permesso di capire che dati avevamo a disposizione e come gestirli.
Lo schema comprende due entità principali: Artisti (Artist) e Opere (Artwork), la relazione tra queste due entita' e' "create", e' una relaione 1:N, ovvero, un artista puo' creare molte opere ma ogni opera appartiene a un solo artista.



1.2 Modello Relazionale

Il modello relazionale si basa esclusivamente sul modello ER:

- **Artists** (artistId, name, gender, birthYear, deathYear, birthCity, deathCity, birthState, deathState, artistUrl, artwork_id);
- **Artworks** (artworkId, accessionNumber, title, dateText, medium, creditLine, year, acquisitionYear, types, width, height, depth, units, inscription, thumbnailUrl, artworkUrl, artist_id, artistName, artistRole).

La chiave esterna artist_id in Artworks fa riferimento all'id di Artists.

Artist

ArtistId	Name	Gender	BirthYear	DeathYear	BirthCity	DeathCity	BirthState	DeathState	ArtistUrl	artworkId

Artwork

ArtworkId	AccessionNumber	Title	DateText	Medium	CreditLine	Year	AcquisitionYear	Types	Width	Height	Depth	Units	Inscription	ThumbnailUrl	artistId	ArtworkUrl	ArtistName	ArtistRole

1.3 Pulizia dei dati

Dopo aver creato il modello ER e il modello relazionale, abbiamo iniziato la costruzione del progetto scaricando i file csv che contenevano i dati grezzzisugli artisti e sulle opere e abbiamo iniziato a pulirli.

1.3.1 Pulizia del dataset Artists

- Gender: gestione del valore mancante tramite il riempimento con il carattere '-' e sostituzione di 'Male' e 'Female' con 'M' e 'F';
- `placeOfBirth` & `placeOfDeath`: gestione dei valori mancanti tramite il riempimento con 'Unknown'
- `placeOfBirth`: essendo un attributo composto lo abbiamo scomposto in `birthCity` e `birthState` basandoci sulla presenza della virgola. Nel caso in cui ci sia solo `birthCity`, `birthState` viene riempito con 'Unknown';
- `placeOfDeath`: diviso in `deathCity` e `deathState` usando la stessa logica di `placeOfBirth`;
- `yearOfBirth` & `yearOfDeath`: sostituiti i valori mancanti con 0
- Eliminazione delle colonne `placeOfBirth`, `placeOfDeath` e `dates`;
- Ordinamento delle colonne rimaste e conversione delle colonne nei tipi di dati desiderati.

1.3.2 Pulizia del dataset Artworks

- Rimozione della colonna`thumbnailUrl`;
- Verifica esistenza dell'attributo `artistId` di `Artworks` in `Artists` tramite la funzione `check_fk`;
- Units: gestione dei valori mancanti tramite il riempimento con 'mm';
- Estrazione di `acquisitionYear` dalla colonna `creditLine`
- Estrazione di widht e height dalla colonna `dimension`;
- Estrazione di type dalla colonna `dimension`;
- Gestione dei valori mancanti di Credit Line, Depth, Year, Inscription, Dimensions, Medium, Acquisition Year tramite l'inserimento di 0 nel caso l'attributo fosse un intero e 'Unknown' nel caso fosse una stringa;
- Sostituto "/www." con "/media." in `thumbnailUrl` per consentire la visualizzazione delle immagini;
- Ordinamento delle colonne rimaste e conversione delle colonne nei tipi di dati desiderati.

1.3.3 Conversione dei tipi di dato

Durante la pulizia dei dati, la conversione dei tipi di dato in quelli desiderati e' stata fatta nel `main()` in due semplici passaggi:

1. Definizione dei tipi di dati desiderati:

```
artist_desired_dtypes ={
    'id'          : int,
    'name'        : str,
    'gender'      : str,
    'yearOfBirth' : int,
    'birthCity'   : str,
    'birthState'  : str,
    'yearOfDeath' : int,
    'deathCity'   : str,
    'deathState'  : str,
    'url'         : str
}
```

2. Chiamata alla funzione `convert_dtypes(df, desired_types)` che prende in input il dataframe interessato e i tipi di dati desiderati per quel dataframe. Questa funzione scorre i nostri attributi convertendoli uno a uno al tipo di dato da noi desiderato.

1.4 Inserimento dei dati

Per l'inserimento dei dati puliti dentro il database MySQL abbiamo eseguito i seguenti passaggi:

1. Scrittura di un file MySQL per la creazione del database e delle tabelle:

   (a) Creazione del database MySQL:

   ```
   CREATE SCHEMA Museo;
   USE Museo;
   ```

   (b) Verifica e creazione delle tabelle Artists e Artworks:

   ```
   CREATE TABLE Artists (
       id INTEGER NOT NULL,
       name VARCHAR(255) NOT NULL,
       gender CHAR NOT NULL,
       yearOfBirth CHAR(4) NOT NULL,
       birthCity VARCHAR(50) NOT NULL,
       birthState VARCHAR(50) NOT NULL,
       yearOfDeath VARCHAR(4),
       deathCity VARCHAR(50),
       deathState VARCHAR(50),
       url VARCHAR(255) NOT NULL,

       PRIMARY KEY (id)
   );

   CREATE TABLE Artworks (
       id INTEGER NOT NULL,
       accession_number CHAR(7) NOT NULL,
       artist VARCHAR(255),
       artistRole VARCHAR(100),
       artistId INTEGER NOT NULL,
       title VARCHAR(2047),
       dateText VARCHAR(255),
       medium VARCHAR(255),
       creditLine VARCHAR(2047),
       year INTEGER,
       acquisitionYear INTEGER,
       types VARCHAR(100),
       width INTEGER,
       height INTEGER,
       depth INTEGER,
   ```

```
    units CHAR(2),
    inscription CHAR(15),
    thumbnailUrl VARCHAR(255),
    url VARCHAR(255),

    PRIMARY KEY (id),
    FOREIGN KEY (artistId) REFERENCES Artists(id)
);
```

2. Scrittura di un file in PHP per la connessione e l'inserimento dei dati puliti nel database MySQL:

   (a) Connessione al database:

   ```
   $link = mysqli_connect($hostname, $username, $password, $dbName);
   ```

   (b) Inserimento:

   ```
   INSERT INTO Artists VALUES (id, name, gender, yearOfBirth, ...)
   INSERT INTO Artworks (id, accession_number, artist, artistRole, ...)
   ```

Capitolo 2

Query SQL

2.1 Ricerca degli artisti

Per funzionalita', in questa parte del sito web, oltre al punto 1 del progetto (ricerca degli artisti inserendo uno o piu' parametri anche parziali), abbiamo incorporato anche il punto 2 (visualizzazione di tutte le opere per un determinato artista).
L'unione del punto 1 e del punto 2 e' stata resa possibile grazie alla creazione di una colonna contenente un bottone che, grazie al passaggio di `artistId` ci permette di visualizzare, in una pagina dedicata, una tabella, strutturalmente uguale a quella degli artisti, dove sono riportate tutte le opere dell'artista da noi selezionato.

- La tabella contenente tutti gli artisti e' ottenuta tramite la seguente query:

```
SELECT *
FROM Artists
WHERE (
    name LIKE '%$name%' AND
    gender LIKE '%$gender%' AND
    yearOfBirth LIKE '%$yearOfBirth%' AND
    yearOfDeath LIKE '%$yearOfDeath%' AND
    birthCity LIKE '%$birthCity%' AND
    birthState LIKE '%$birthState%' AND
    deathCity LIKE '%$deathCity%' AND
    deathState LIKE '%$deathState%'
)
```

- La tabella contenente tutte le opere di un determinato artista, e' ottenuta tramite la seguente query:

```
SELECT *
FROM Artworks
JOIN Artists ON Artworks.artistId = Artists.id
WHERE (Artworks.artistId=<artistId>)
ORDER BY Artworks.year ASC
```

dove `artistId` e' un informazione che otteniamo nel momento in cui clicchiamo il bottone "View Artworks" ed e' presente nella variabile `$_GET['artistId']`. Oltre alla tabella delle opere viene mostrato anche il numero delle opere fatte dall'artista selezionato grazie alla funzione `mysqli_num_rows()` che, semplicemente, restituisce il numero di righe della tabella.

2.2 Ricerca delle opere

Ricerca delle opere inserendo uno o più parametri (anche parziali), in forma libera o eventualmente guidata corredate dal nome dell'artista.
Stesso procedimento del punto 1, ma questa volta lo facciamo per le opere:

```
SELECT *
FROM Artworks
WHERE (
    accession_number LIKE '%$accession_number%' AND
    title LIKE '%$title%' AND
    medium LIKE '%$medium%' AND
    year LIKE '%$year%' AND
    acquisitionYear LIKE '%$acuisitionYear%'
)
LIMIT 15000
```

2.3 Statistiche

Ques'ultima pagina del sito web mostra unna serie di statistiche riguardanti il database:

- Numero di opere realizzate in un determinato anno

  ```
  SELECT COUNT(*)
  FROM Artworks
  WHERE (year = '$year')
  ```

- Numero di artisti nati o morti in una specifica nazione

  ```
  SELECT COUNT(*)
  FROM Artists
  WHERE (birthState = '$nation' OR deathState = '$nation')
  ```

- Artisti che hanno pubblicato la loro prima opera sotto i 18 anni

```
SELECT at.id, at.name, at.yearOfBirth
  FROM Artists at
  JOIN Artworks aw ON at.id = aw.artistId
  GROUP BY at.id, at.name, at.yearOfBirth
  HAVING (MIN(CONVERT(aw.year, SIGNED)) - CONVERT(at.yearOfBirth, SIGNED)) < 18
          AND MIN(CONVERT(aw.year, SIGNED)) > 0
  ORDER BY at.name
```

- Artisti che hanno variato di piu' nella realizzazione delle loro opere

```
SELECT at.name, COUNT(DISTINCT aw.medium) AS differentMediums
FROM Artists at
JOIN Artworks aw ON at.id = aw.artistId
GROUP BY at.id, at.name
ORDER BY differentMediums DESC
LIMIT 1
```

- L'opera con l'area piu' grande e piu' piccola di un determinato artista

```
SELECT aw.title, aw.units, aw.width * aw.height AS area
FROM Artworks aw
JOIN Artists at ON aw.artistId = at.id
WHERE at.name = '$artistName'
      AND aw.width IS NOT NULL
      AND aw.height IS NOT NULL
      AND aw.units IS NOT NULL
      AND aw.title IS NOT NULL
      AND (aw.width * aw.height) > 0
ORDER BY area ASC
LIMIT 1
```

  questa query ci fornisce il nome dell'opera con l'area minore dell'artista da noi inserito. All'inizio abbiamo usato una query piu' complicata. La lentezza della query pensata in primo piano, ci ha obbligato a semplificarla e dividerla in due parti, ne deriva una statistica composta da due sotto-query che vengono eseguite di seguito cambiando solo ASC con DESC:

```
$sql = str_replace('ASC', 'DESC', $sql);
```

  questo ci permette di trovare, con la prima query l'opera di area minima, mentre con la seconda query l'opera di area massima.

Query iniziale:

```
SELECT
    at.name,
    max_aw.title AS maxTitle,
    max_aw.units AS maxUnits,
    max_aw.width * max_aw.height AS maxArea,
    min_aw.title AS minTitle,
    min_aw.units AS minUnits,
    min_aw.width * min_aw.height AS minArea
FROM Artists at
JOIN Artworks max_aw ON at.id = max_aw.artistId
JOIN Artworks min_aw ON at.id = min_aw.artistId
WHERE
    at.name = '$minMaxArtistName'
    AND max_aw.width * max_aw.height = (
        SELECT MAX(aw.width * aw.height)
        FROM Artworks aw
        WHERE aw.artistId = at.id
    )
    AND min_aw.width * min_aw.height = (
        SELECT MIN(aw.width * aw.height)
        FROM Artworks aw
        WHERE aw.artistId = at.id AND aw.width * aw.height > 0
    )
```

Il problema di questa query, oltre ad essere inutilmente complicata, e' la lentezza: infatti, qauando inseriamo un artista con un numero elevato di opere, causa un numero enorme di combinazioni che devono essere valutate (questo per colpa delle 2 `JOIN` tra `Artists` e `Artworks`) . Dividendo in 2 la query, oltre ad un incremento di velocita' nel soddisfare la richiesta, miglioriamo anche la semplicita' della nostra interrogazione al database.