Prima di iniziare a lavorare sul database lo abbiamo rappresentato con un diagramma ER per farci un'idea del lavoro da svolgere durante la pulizia dei dati in pandas.

Diagramma ER con i valori del csv

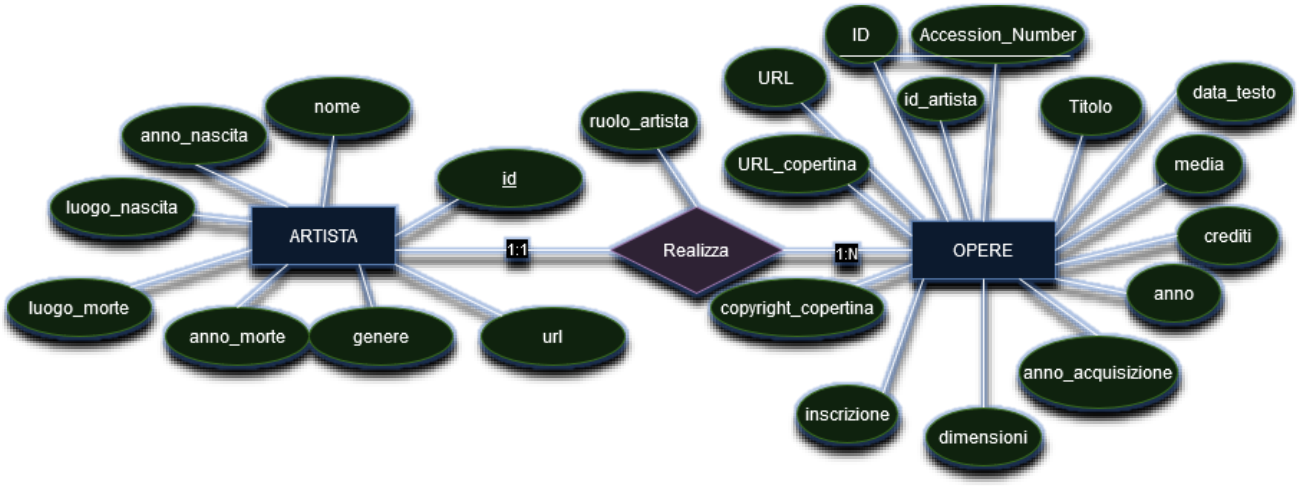
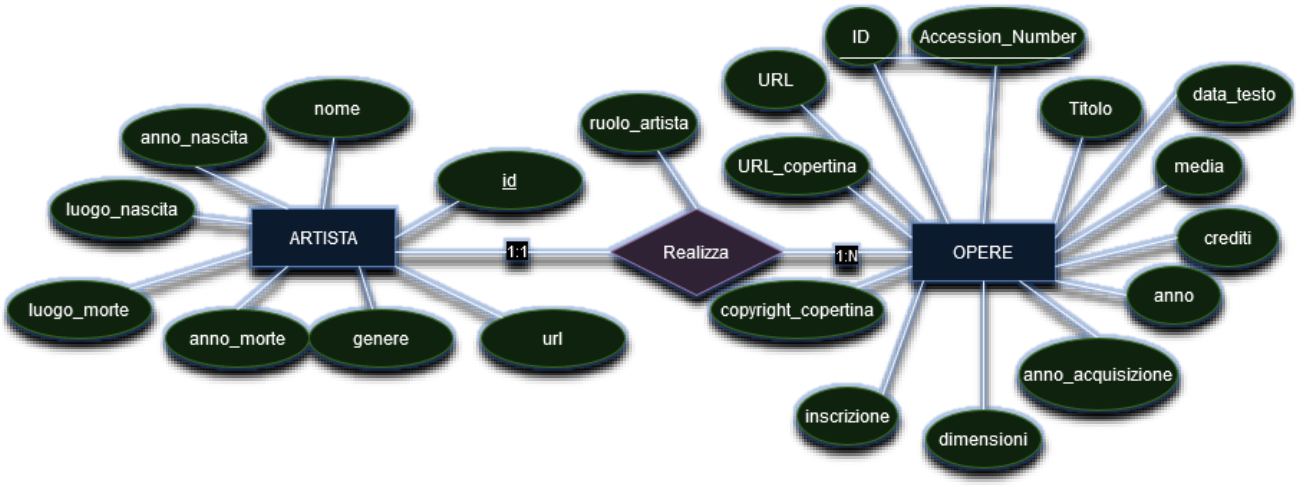


Diagramma ER dopo aver rimosso i dati derivati superflui



Oltre ad aver quindi rimosso i dati superflui abbiamo trovato delle incongruenze (alcune stringhe non erano scritte tra doppi apici, alcuni campi mancanti, le date segnate come float etc.). Abbiamo quindi deciso di uniformare il tipo di dato per colonna come ad esempio qui:

df_lavori['id'] = pd.to_numeric(df_lavori['id'], errors='coerce').astype(pd.Int32Dtype())

usando pd.to_numeric per convertire il dato in numero (float di base) e con errors='coerce' per verificare che il dato non sia mancante o non valido altrimenti sarebbe stato riempito con pd.NA (valore NULLO) , astype (pd.Int32Dtype()) invece converte il valore in intero. Stessa cosa per le stringhe df_lavori['inscription'] = df_lavori['inscription'].replace('', pd.NA).astype(pd.StringDtype()) . Per verificare se la conversione sia andata a buon fine abbiamo usato print(df_lavori.dtypes). Per verificare la presenza di duplicati abbiamo usato print(df_lavori.shape) che ci mostrava il totale delle righe e colonne del dataframe, poi df_lavori=df_lavori.drop_duplicates() per togliere i duplicati ed infine abbiamo nuovamente eseguito un print(df_lavori.shape) per vedere se il dataframe avesse cambiato dimensioni.

Per gestire meglio il database abbiamo creato un nuovo dataframe chiamato Realizza in cui abbiamo inserito: id dell'artista, ruolo _artista, id e accession_number di opere. Successivamente abbiamo eliminato le colonne del ruolo dell'artista e il suo id perché superflui nel dataframe delle opere.

Per verificare la lunghezza massima dei char per colonna in modo da creare i dati delle tabelle nel file .sql delle giuste dimensioni abbiamo implementato la seguente funzione che abbiamo cercato su internet max_lengths = df_lavori.apply(lambda col: col.astype(str).str.len().max()).

Per la creazione del database abbiamo salvato i file in un .sql che abbiamo eseguito, dopo esserci connessi da terminale, col comando source *percorso del .sql*

Per connetterci al database abbiamo creato un file connessione.php in modo tale che lo potessimo includere in ogni file che faceva richieste di query al database, senza riscrivere ogni volta il codice.

Per l'inserimento degli artisti abbiamo stabilito la connessione al database, aperto il file degli artisti_puliti.csv, scartato la prima riga contenente le intestazioni delle tabelle e abbiamo proceduto a ricavare iterativamente i dati di ogni riga.

Per evitare errori con i caratteri speciali nelle stringhe abbiamo usato real_escape_string per aggiunge backslash davanti ai caratteri speciali come apici singoli, apici doppi , backslash e byte nulli; inoltre abbiamo verificato se ci fossero stringhe vuote con un'espressione condizionale assegnando NULL per i valori vuoti o loro stessi se fossero stati presenti come stringhe. Anche per gli interi abbiamo utilizzato un'espressione condizionale: nel caso non fosse stato interpretato come numero avremmo assegnato il valore NULL, (ad esempio nei campi vuoti che avevamo pulito in pandas). Infine abbiamo scritto la query per l'inserimento $sql = "INSERT INTO ARTISTI (id, nome, genere, anno_nascita, anno_morte, luogo_nascita, luogo_morte, indirizzo_url) VALUES ($id, $name, $gender, $yearOfBirth, $yearOfDeath, $placeOfBirth, $placeOfDeath, $url)"; e abbiamo posto un controllo: una stampa per la query eseguita con successo e una per eventuali errori. E infine abbiamo chiuso il file csv e abbiamo interrotto la connessione al database. La stessa operazione l'abbiamo adotta per l'inserimento dei dati delle opere, e per la relazione REALIZZA.

Per quanto riguarda le query per cercare gli artisti e le opere (punto 1 e 3) per determinati parametri (anche parziali) abbiamo eseguito la query $sql = "SELECT * FROM ARTISTI WHERE 1=1"; e aggiunto alla variabile $sql le condizioni aggiuntive, come ad esempio $sql .= " AND id = $id; se fosse stato presente un id. Nei casi con le stringhe abbiamo utilizzato $sql .= " AND nome LIKE '%$nome%'"; dove le percentuali indicano che la variabile presa può essere preceduta da qualcosa prima o seguita da qualcos'altro.

Per la sezione che richiedeva il Nome di un'artista (punto 2) per ricercare le sue opere abbiamo usato la seguente query $sql_opere = "SELECT o.* FROM OPERE o

JOIN REALIZZA r ON o.id = r.id_opera AND o.accession_number = r.accession_number_opera

WHERE r.id_artista = (SELECT id FROM ARTISTI WHERE nome LIKE '%$nome_artista%')

ORDER BY o.media, o.anno ";}

che seleziona tutte le colonne relative alle opere, da opere, e le unisce sull'id e l'accession number con REALIZZA per poi selezionare nella sottoquery solo gli id corrispondenti al nome dell'artista, ordinati per il tipo di media e dall'anno dell'opera.

Per la sezione statistiche (punto 4) abbiamo creato una pagina a parte per selezionare il tipo di statistica. Per il numero di opere per anno (4.1) abbiamo scritto la query: SELECT COUNT(*) AS numero_opere FROM OPERE WHERE anno = $anno che conta il numero di opere in un determinato $anno e poi le mette in una colonna con intestazione 'numero_opere'.

Per il numero di artisti nati o/e morti in una determinata nazione (punto 4.2): SELECT COUNT(DISTINCT id) AS numero_artisti FROM ARTISTI WHERE luogo_nascita LIKE '%$nazione%' OR luogo_morte LIKE '%$nazione%' abbiamo usato il distinct per differenziare il caso in cui un artista nasca e muoia nella stessa nazione e venga contato due volte.

Per il numero di opere realizzate da un determinato artista (punto 4.3):

```
SELECT COUNT(*) AS numero_opere FROM OPERE O
JOIN REALIZZA R1 ON O.id = R1.id_opera AND O.accession_number = R1.accession_number_opera
JOIN ARTISTI A ON R1.id_artista = A.id
WHERE A.id = (SELECT id FROM ARTISTI WHERE nome LIKE '$nome_artista' );
```

Abbiamo chiamato la colonna che identifica il numero totale di opere per artista con 'numero_opere', da opere abbiamo unito su id e accession number OPERA e REALIZZA e l'abbiamo a sua volta unito all'id artista con l'id artista di REALIZZA, poi col where abbiamo posto nella stottoquery una condizione per la quale l'id dell'ARTISTA venisse associato al nome da noi richiesto. Non abbiamo messo la possibilità di inserimento parziale in quanto fraintendibile. (es. Ercolec che ha 102 opere ma di cui sua 1 e 101 di Ercole).

Per i tre punti a scelta noi abbiamo deciso di verificare le statistiche di :

1) Numero di opere per media (punto 4.4): `SELECT COUNT(*) AS numero_opere FROM OPERE WHERE media LIKE '$media';` dove contiamo nella colonna 'numero_opere' le opere per tipo di media richiesto dall'utente. Non abbiamo messo la possibilità di inserimento parziale in quanto fraintendibile per lo stesso motivo di prima.

2) Numero di opere acquisite da anno ad anno (punto 4.5): `SELECT COUNT(*) AS numero_opere FROM OPERE WHERE anno_acquisizione IS NOT NULL AND anno_acquisizione BETWEEN $anno_inizio AND $anno_fine;`

crea la colonna 'numero_opere' nella quale mette il totale delle opere che sono state acquisite da anno_inizio e anno_fine e i quali anni di acquisizione non siano null.

3) Numero di artisti vivi per luogo di nascita in ordine decrescente (punto 4.6): `SELECT luogo_nascita, COUNT(*) AS num_artisti_vivi FROM ARTISTI WHERE luogo_nascita IS NOT NULL AND anno_morte IS NULL GROUP BY luogo_nascita ORDER BY num_artisti_vivi DESC`

seleziona la colonna del 'luogo_nascita', e conta il numero totale di artisti dove il luogo di nascita non è nullo e l'anno di morte non è nullo, raggruppando per luogo di nascita e ordinato in ordine decrescente per il numero di artisti ancora in vita.

174668 Giuseppe Viggiano, 188296 Alessia Brina