Team 4 Project Part 2:
Litsmonian Museum Exhibit Management

By:
Emily Nguyen, Aiden Gump, Logan House, Jacob Maxwell

Contribution:
Emily: 25% | Aiden: 25% | Logan: 25% | Jacob: 25%

Litsmonian Museum Exhibit Management

There is one Litsmonian Museum. The scope of exhibit management includes artifacts, exhibits, sections, and artifact owners. Out of scope are staff members, concessions, tickets, visitors, gift shop, movies/documentaries, penny press & swirl, and exhibit sponsors.

Business Rules

1. Each artifact must have an Artifact ID, a description, a medium (oil on canvas, speech, novel, fossil, clothing, etc.), an ownership status, and a display status. Each artifact also has the foreign keys of Owner ID, Exhibit ID, and Section ID. The ownership status shall be either: owned or borrowed. The display status shall be either: on display, loaned out, or in storage. Each artifact may have a title and a section/exhibit.

2. An artifact is either a piece of artwork, a piece of literature, or an item.

3. Each artwork must have a style (artistic era). Each artwork may have a creation date and artist name.

4. Each piece of literature must have a language. Each piece of literature may have an author, creation date, transcription, and genre.

5. Each item may have a date discovered, historical/relevant date range, and creator.

6. Each exhibit will have an Exhibit ID, a name, an start date, end date, and exhibit descriptions.

7. Each section will have a Section ID and type denoted by a number (empty[0] / storage[1] / display[2]). Each section also has the foreign key of Exhibit ID. Sections will only belong to at most one hosted exhibit at a time to prevent over-booking. Sections may host exhibits.

8. Each artifact owner has a name and an Owner ID. An artifact owner can either be an organization or an individual.

Major Entity Types (and relationships)

Artifact: An item that is generally of some historical or cultural context. These items are owned by either the Litsmonion, a different museum, or private collectors. Some of the artifacts are displayed in exhibits (generally several) while some are in storage or loaned out. An artifact may belong to a specific exhibit.

Exhibit: An exhibition of several artifacts of some similar category or theme (Ex. Aztec artifacts). Exhibits are the main attractions at the Litsmonian. Exhibits can inhabit multiple sections.

Section: A section is a partitioned space in the museum to display artifacts of a certain exhibit or store artifacts:. A section can host only 1 exhibit entity. Multiple sections can host the same exhibit. Typically, museum exhibits have multiple room-like sections to walk through. There are also storage sections.

Artifact Owner: Some benefactor or organization who owns artifacts currently located at the Litsmonian. An artifact owner can be an individual (Ex. Jeff Bazos) or another museum (Ex. The Smithsonian). An artifact owner can own several artifacts.

Table and relationship assumptions

Estimated # of tables: 7

- Artifacts
- Artwork
- Literature
- Items
- Exhibits
- Sections
- Artifact Owners

Estimated # of relationships: 7

- Artifact IS_A Artwork
- Artifact IS_A Literature
- Artifact IS_A Item
- Artifact IS_IN Section
- Exhibit CONSISTS_OF Artifact(s)
- Artifact Owner OWNS Artifact(s)
- Exhibit INCLUDES Section(s)

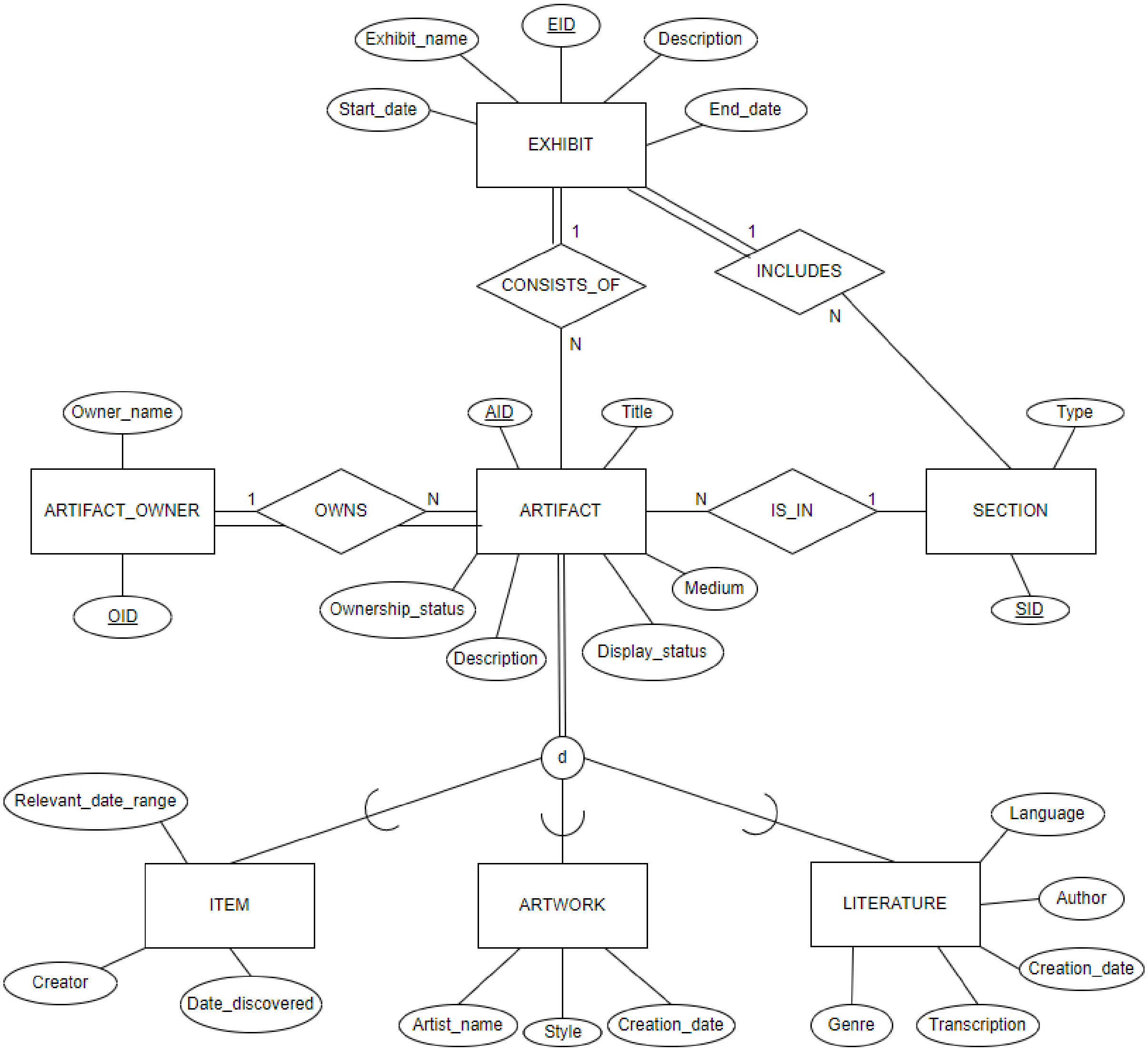
Variable assumptions

We estimate that the museum will house an estimated 80000 artifacts, rotate through 30 exhibits a year, have 20 sections (~ 15 display 5 storage), and have about 900 artifact owners.

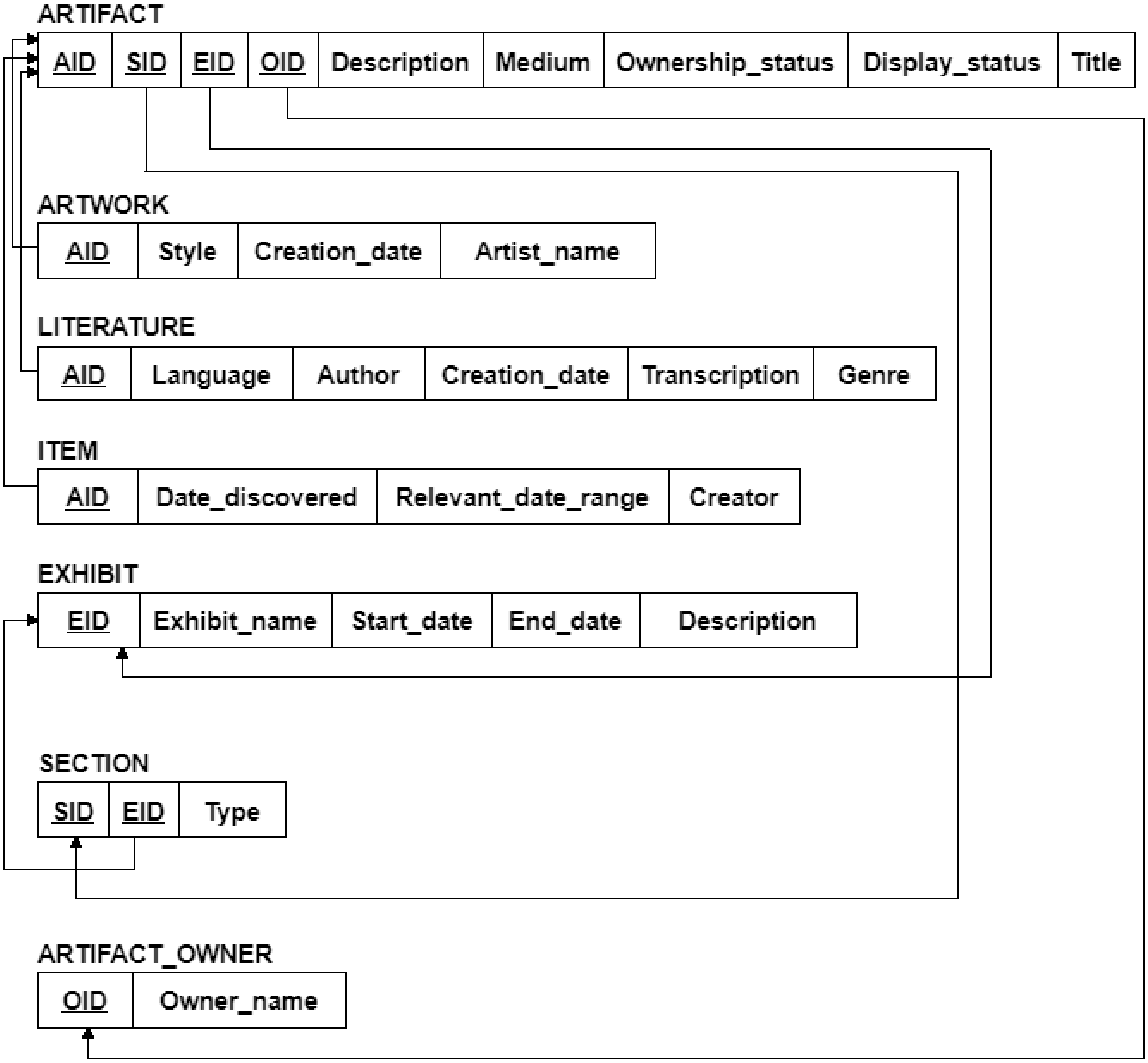
What the project will achieve:

The project will provide a usable database for the management of the contents of a museum for the eventual users. The database will keep track of information such as: artifacts, the exhibits those artifacts inhabit, which sections those artifacts are in, and the owners of each artifact. The eventual users will be able to easily access information and keep track of the statuses and locations (within the museum) of the artifacts. Some authoritative users will be able to add, modify, and delete information.

Entity-Relationship Diagram



Supporting Schema Diagram



ARTIFACT

AID	SID	EID	OID	Description	Medium	Ownership_status	Display_status	Title

ARTWORK

AID	Style	Creation_date	Artist_name

LITERATURE

AID	Language	Author	Creation_date	Transcription	Genre

ITEM

AID	Date_discovered	Relevant_date_range	Creator

EXHIBIT

EID	Exhibit_name	Start_date	End_date	Description

SECTION

SID	EID	Type

ARTIFACT_OWNER

OID	Owner_name

Data Dictionary

<table>
<tr><th>Table</th><th>Table Description</th><th>Primary Key</th><th>SuperKey</th><th>Attributes</th><th>Foreign Key</th></tr>
<tr><td>Exhibit</td><td>An exhibition of several artifacts of some similar category or theme</td><td>EID</td><td>EID, Exhibit_name</td><td>
<table>
<tr><th>Attribute</th><th>Data Type</th><th>Domain</th></tr>
<tr><td>EID</td><td>INT</td><td>Any positive integer</td></tr>
<tr><td>Exhibit_name</td><td>VARCHAR</td><td>Any string</td></tr>
<tr><td>Start_date</td><td>DATE</td><td>Any date</td></tr>
<tr><td>End_date</td><td>DATE</td><td>Any date after Start_date</td></tr>
<tr><td>Description</td><td>VARCHAR</td><td>Any string</td></tr>
</table>
</td><td>N/A</td></tr>
<tr><td>Section</td><td>A section is a partitioned space in the museum to display artifacts of a certain exhibit or store artifacts</td><td>SID</td><td>SID, Type</td><td>
<table>
<tr><th>Attribute</th><th>Data Type</th><th>Domain</th></tr>
<tr><td>SID</td><td>INT</td><td>Any positive integer</td></tr>
<tr><td>SType</td><td>ENUM</td><td>‘empty’
‘storage’
‘display’</td></tr>
<tr><td>EID</td><td>INT</td><td>Any existing EID and NULL</td></tr>
</table>
</td><td>EID</td></tr>
<tr><td>Artifact Owner</td><td>Some benefactor or organization who owns artifacts currently located at the Litsmonian</td><td>OID</td><td>OID, Owner_name</td><td>
<table>
<tr><th>Attribute</th><th>Data Type</th><th>Domain</th></tr>
<tr><td>OID</td><td>INT</td><td>Any positive integer</td></tr>
<tr><td>Owner_name</td><td>VARCHAR</td><td>Any String</td></tr>
</table>
</td><td>N/A</td></tr>
<tr><td>Artifact</td><td>An item that is generally of some historical or</td><td>AID</td><td>AID, OID, Display_ Status, Ownership</td><td>
<table>
<tr><th>Attribute</th><th>Data Type</th><th>Domain</th></tr>
</table>
</td><td>OID, EID, SID</td></tr>
</table>

<table>
<tr><td></td><td>cultural context</td><td></td><td>_status</td><td>
<table>
<tr><td>AID</td><td>INT</td><td>Any positive integer</td></tr>
<tr><td>Description</td><td>VARCHAR</td><td>Any String</td></tr>
<tr><td>Art_Medium</td><td>VARCHAR</td><td>Any String</td></tr>
<tr><td>Ownership_ status</td><td>ENUM</td><td>‘OWNED’, ‘BORROWED’</td></tr>
<tr><td>Display_ status</td><td>ENUM</td><td>‘ON_DISPLAY’, ‘LOANED_ OUT’, ‘IN_STORAGE’</td></tr>
<tr><td>Title</td><td>VARCHAR</td><td>Any String and NULL</td></tr>
<tr><td>OID</td><td>INT</td><td>An existing OID</td></tr>
<tr><td>EID</td><td>INT</td><td>An existing EID and NULL</td></tr>
<tr><td>SID</td><td>INT</td><td>An existing SID and NULL</td></tr>
</table>
</td><td></td></tr>
<tr><td>Artwork</td><td>An artifact that is considered a piece of art, for example, a painting or sketch.</td><td>AID</td><td>AID, Style</td><td>
<table>
<tr><th>Attribute</th><th>Data Type</th><th>Domain</th></tr>
<tr><td>AID</td><td>INT</td><td>An existing AID</td></tr>
<tr><td>Style</td><td>VARCHAR</td><td>Any String</td></tr>
<tr><td>Creation_ date</td><td>VARCHAR</td><td>Any day up to current and NULL</td></tr>
<tr><td>Artist_name</td><td>VARCHAR</td><td>Any String and NULL</td></tr>
</table>
</td><td>AID</td></tr>
<tr><td>Literature</td><td>An artifact that is considered a piece of literature, for example, a book or poem.</td><td>AID</td><td>AID, Language</td><td>
<table>
<tr><th>Attribute</th><th>Data Type</th><th>Domain</th></tr>
<tr><td>AID</td><td>INT</td><td>An existing AID</td></tr>
<tr><td>Lit_Language</td><td>VARCHAR</td><td>Any String</td></tr>
</table>
</td><td>AID</td></tr>
</table>

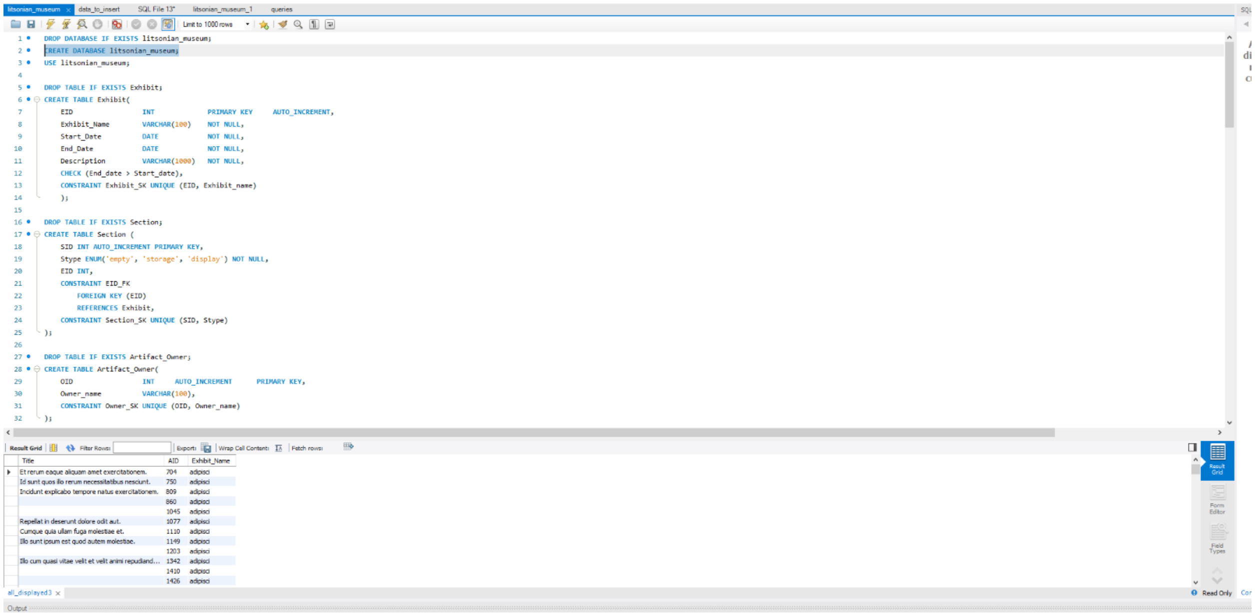
<table>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><td>Author</td><td>VARCHAR</td><td>Any String and NULL</td></tr>
<tr><td>Creation_ date</td><td>VARCHAR</td><td>Any day up to current and NULL</td></tr>
<tr><td>Transcription</td><td>VARCHAR</td><td>Any String and NULL</td></tr>
<tr><td>Genre</td><td>VARCHAR</td><td>Any String and NULL</td></tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td>All artifacts aside from artworks or pieces of literature, for example, an old cannon or a stone-age tool</td>
<td>AID</td>
<td>AID</td>
<td>
<table>
<tr><th>Attribute</th><th>Data Type</th><th>Domain</th></tr>
<tr><td>AID</td><td>INT</td><td>An existing AID</td></tr>
<tr><td>Date_ discovered</td><td>VARCHAR</td><td>Any date up to current and NULL</td></tr>
<tr><td>Relevent_date _range</td><td>VARCHAR</td><td>Roman numeral century</td></tr>
<tr><td>Creator</td><td>VARCHAR</td><td>Any String and NULL</td></tr>
</table>
</td>
<td>AID</td>
</tr>
</table>

Tools Used

ERD:	draw.io, Snipping Tool
Schema Diagram:	Google Docs, MS Paint
Data Dictionary:	Google Docs
SQL Files/Queries:	MySQL Workbench

Screenshots

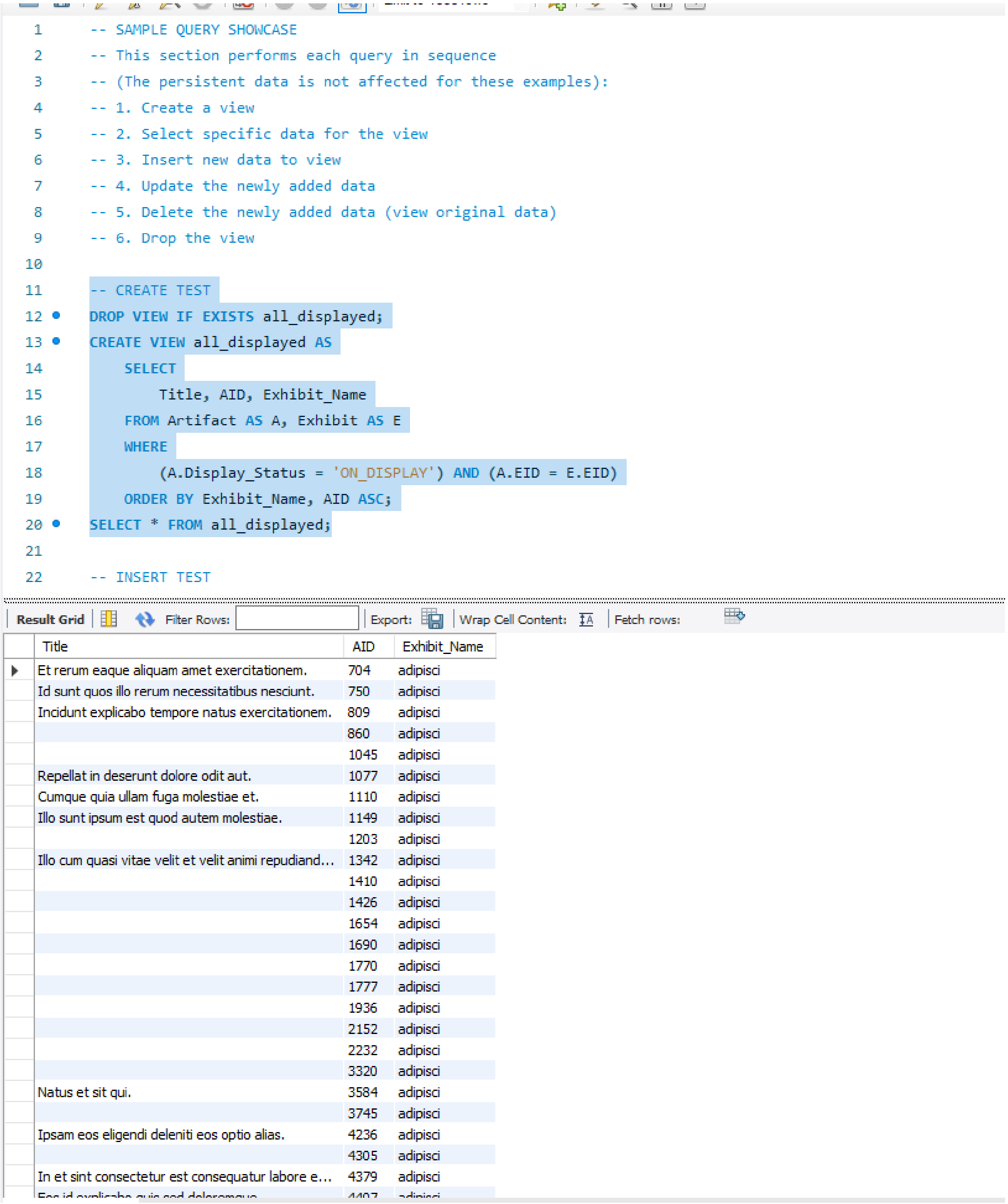
Work Area for Queries



```sql
DROP DATABASE IF EXISTS litsonian_museum;
CREATE DATABASE litsonian_museum;
USE litsonian_museum;

DROP TABLE IF EXISTS Exhibit;
CREATE TABLE Exhibit(
    EID                 INT             PRIMARY KEY     AUTO_INCREMENT,
    Exhibit_Name        VARCHAR(100)    NOT NULL,
    Start_Date          DATE            NOT NULL,
    End_Date            DATE            NOT NULL,
    Description         VARCHAR(1000)   NOT NULL,
    CHECK (End_date > Start_date),
    CONSTRAINT Exhibit_SK UNIQUE (EID, Exhibit_name)
    );

DROP TABLE IF EXISTS Section;
CREATE TABLE Section (
    SID INT AUTO_INCREMENT PRIMARY KEY,
    Stype ENUM('empty', 'storage', 'display') NOT NULL,
    EID INT,
    CONSTRAINT EID_FK
        FOREIGN KEY (EID)
        REFERENCES Exhibit,
    CONSTRAINT Section_SK UNIQUE (SID, Stype)
);

DROP TABLE IF EXISTS Artifact_Owner;
CREATE TABLE Artifact_Owner(
    OID                 INT     AUTO_INCREMENT      PRIMARY KEY,
    Owner_name          VARCHAR(100),
    CONSTRAINT Owner_SK UNIQUE (OID, Owner_name)
);
```

Title	AID	Exhibit_Name
Et rerum eaque aliquam amet exercitationem.	704	adipisci
Id sunt quos illo rerum necessitatibus nesciunt.	750	adipisci
Incidunt explicabo tempore natus exercitationem.	809	adipisci
	860	adipisci
	1045	adipisci
Repellat in deserunt dolore odit aut.	1077	adipisci
Cumque quia ullam fuga molestiae et.	1110	adipisci
Illo sunt ipsum est quod autem molestiae.	1149	adipisci
	1203	adipisci
Illo cum quasi vitae velit et velit animi repudiand...	1342	adipisci
	1410	adipisci
	1426	adipisci

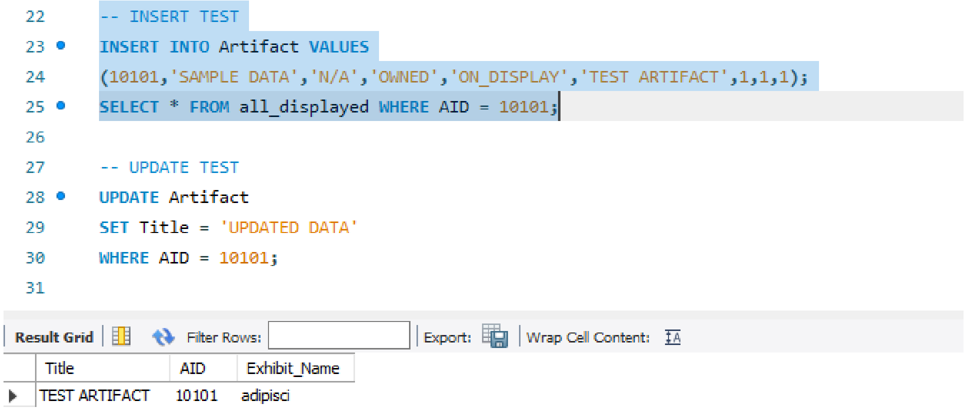
Create Example



```sql
-- SAMPLE QUERY SHOWCASE
-- This section performs each query in sequence
-- (The persistent data is not affected for these examples):
-- 1. Create a view
-- 2. Select specific data for the view
-- 3. Insert new data to view
-- 4. Update the newly added data
-- 5. Delete the newly added data (view original data)
-- 6. Drop the view

-- CREATE TEST
DROP VIEW IF EXISTS all_displayed;
CREATE VIEW all_displayed AS
    SELECT
        Title, AID, Exhibit_Name
    FROM Artifact AS A, Exhibit AS E
    WHERE
        (A.Display_Status = 'ON_DISPLAY') AND (A.EID = E.EID)
    ORDER BY Exhibit_Name, AID ASC;
SELECT * FROM all_displayed;

-- INSERT TEST
```

Title	AID	Exhibit_Name
Et rerum eaque aliquam amet exercitationem.	704	adipisci
Id sunt quos illo rerum necessitatibus nesciunt.	750	adipisci
Incidunt explicabo tempore natus exercitationem.	809	adipisci
	860	adipisci
	1045	adipisci
Repellat in deserunt dolore odit aut.	1077	adipisci
Cumque quia ullam fuga molestiae et.	1110	adipisci
Illo sunt ipsum est quod autem molestiae.	1149	adipisci
	1203	adipisci
Illo cum quasi vitae velit et velit animi repudiand...	1342	adipisci
	1410	adipisci
	1426	adipisci
	1654	adipisci
	1690	adipisci
	1770	adipisci
	1777	adipisci
	1936	adipisci
	2152	adipisci
	2232	adipisci
	3320	adipisci
Natus et sit qui.	3584	adipisci
	3745	adipisci
Ipsam eos eligendi deleniti eos optio alias.	4236	adipisci
	4305	adipisci
In et sint consectetur est consequatur labore e...	4379	adipisci
Eos id explicabo quis sed doloremque	4407	adipisci

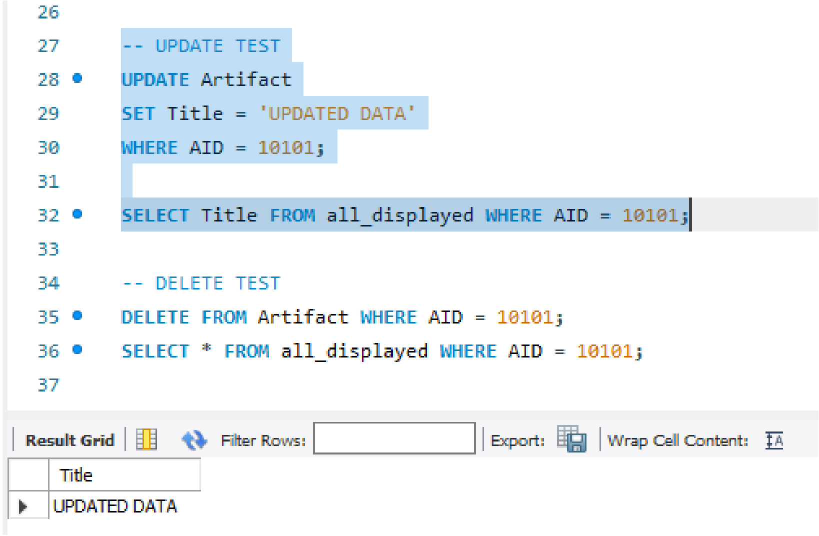
Insert Example



```
-- INSERT TEST
INSERT INTO Artifact VALUES
(10101,'SAMPLE DATA','N/A','OWNED','ON_DISPLAY','TEST ARTIFACT',1,1,1);
SELECT * FROM all_displayed WHERE AID = 10101;

-- UPDATE TEST
UPDATE Artifact
SET Title = 'UPDATED DATA'
WHERE AID = 10101;

```

Title	AID	Exhibit_Name
TEST ARTIFACT	10101	adipisci

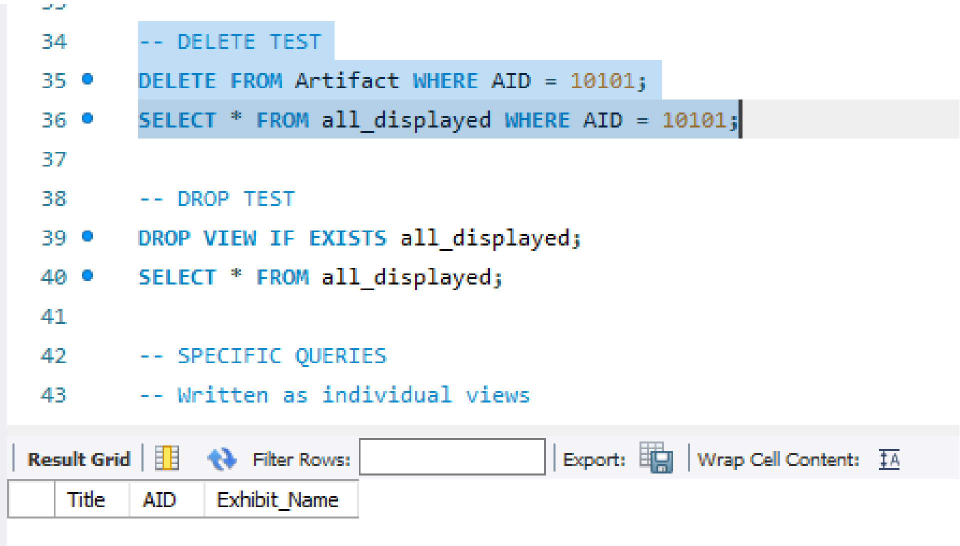
Update Example



```

-- UPDATE TEST
UPDATE Artifact
SET Title = 'UPDATED DATA'
WHERE AID = 10101;

SELECT Title FROM all_displayed WHERE AID = 10101;

-- DELETE TEST
DELETE FROM Artifact WHERE AID = 10101;
SELECT * FROM all_displayed WHERE AID = 10101;

```

Title
UPDATED DATA

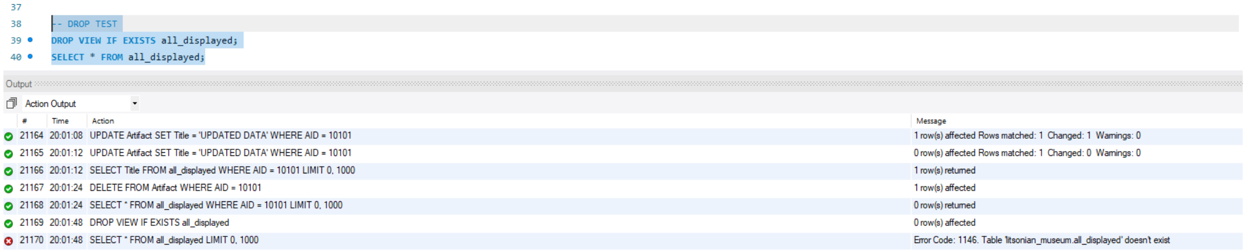
Delete Example



```
-- DELETE TEST
DELETE FROM Artifact WHERE AID = 10101;
SELECT * FROM all_displayed WHERE AID = 10101;

-- DROP TEST
DROP VIEW IF EXISTS all_displayed;
SELECT * FROM all_displayed;

-- SPECIFIC QUERIES
-- Written as individual views
```

	Title	AID	Exhibit_Name

Drop Example



```
-- DROP TEST
DROP VIEW IF EXISTS all_displayed;
SELECT * FROM all_displayed;
```

```
#      Time      Action                                                         Message
21164  20:01:08  UPDATE Artifact SET Title = 'UPDATED DATA' WHERE AID = 10101   1 row(s) affected Rows matched: 1  Changed: 1  Warnings: 0
21165  20:01:12  UPDATE Artifact SET Title = 'UPDATED DATA' WHERE AID = 10101   0 row(s) affected Rows matched: 1  Changed: 0  Warnings: 0
21166  20:01:12  SELECT Title FROM all_displayed WHERE AID = 10101 LIMIT 0, 1000   1 row(s) returned
21167  20:01:24  DELETE FROM Artifact WHERE AID = 10101                         1 row(s) affected
21168  20:01:24  SELECT * FROM all_displayed WHERE AID = 10101 LIMIT 0, 1000    0 row(s) returned
21169  20:01:48  DROP VIEW IF EXISTS all_displayed                              0 row(s) affected
21170  20:01:48  SELECT * FROM all_displayed LIMIT 0, 1000                      Error Code: 1146. Table 'itsonian_museum.all_displayed' doesn't exist
```

Log:

4/25/2022

- 20:03 - Inserted table: Exhibit
- 20:10 -Inserted table: Section
- 20:17 -Inserted table: Artifact_Owner
- 20:30 - Inserted table: Artifact
- 20:42 - Inserted table: Artwork
- 21:03 - Inserted table: Literature
- 21:22 - Inserted table: Item
- 21:24 - Propogate Exhibit with 2 tuples
- 21:29 - Propogate Section with 7 tuples
- 21:36 - Propogate Artifact_Owner with 5 tuples
- 21:40 - Propogate Artifact with 5 tuples
- 21:42 - Propogate Artwork with 1 tuple
- 21:50 - Propogate Item with 5 tuples
- 22:07 - Implemented display all view

5/5/2022

- 19:21 - Propogate Exhibit with 51 tuples
- 19:21 - Propogate Section with 40 tuples
- 19:21 - Propogate Artifact_Owner with 900 tuples
- 19:23 - Propogate Artifact with 10000 tuples
- 19:23 - Propogate Artwork with 250 tuples
- 19:25 - Propogate Item with 9500 tuples
- 19:26 - Propogate Literature with 250 tuples

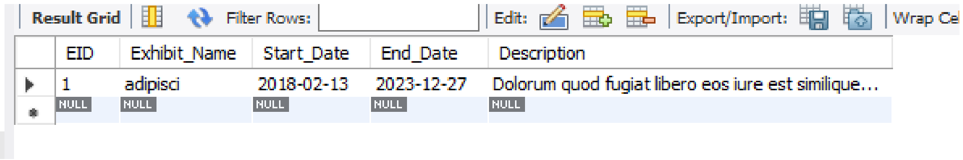
20:01 - Added queries for CREATE, SELECT, INSERT, DELETE, UPDATE, DROP, Q1, Q2, Q3, Q4, Q5

Data Generation:

Exhibit - Planned Size: 60
Section - Planned Size: 45
Artifact_Owner - Planned Size: 1000
Artifact - Planned Size: 80000
Artwork - Planned Size: 10000
Item - Planned Size: 60000
Literature - Planned Size: 10000

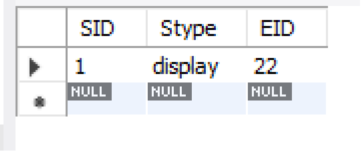
One Tuple from Each Table

Exhibit



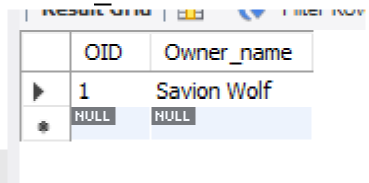
	EID	Exhibit_Name	Start_Date	End_Date	Description
▶	1	adipisci	2018-02-13	2023-12-27	Dolorum quod fugiat libero eos iure est similique...
*	NULL	NULL	NULL	NULL	NULL

Section



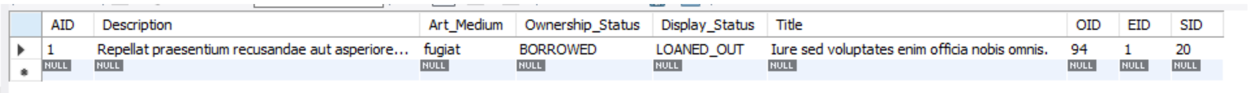
	SID	Stype	EID
▶	1	display	22
*	NULL	NULL	NULL

Artifact_Owner



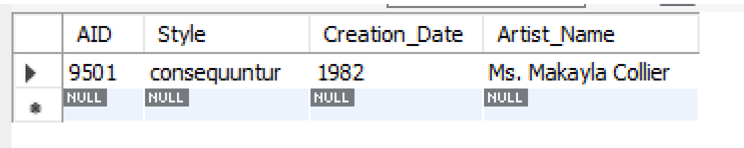
	OID	Owner_name
▶	1	Savion Wolf
*	NULL	NULL

Artifact



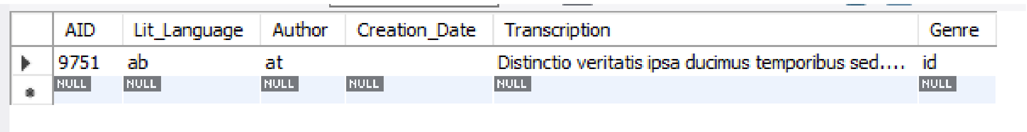
	AID	Description	Art_Medium	Ownership_Status	Display_Status	Title	OID	EID	SID
▶	1	Repellat praesentium recusandae aut asperiore...	fugiat	BORROWED	LOANED_OUT	Iure sed voluptates enim officia nobis omnis.	94	1	20
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Artwork



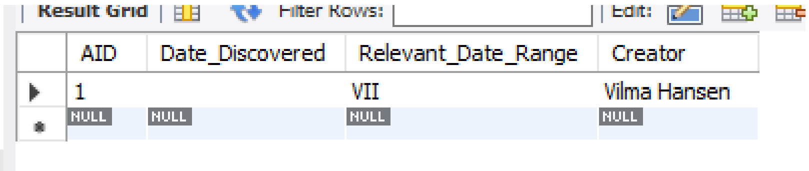
	AID	Style	Creation_Date	Artist_Name
▶	9501	consequuntur	1982	Ms. Makayla Collier
*	NULL	NULL	NULL	NULL

Literature



	AID	Lit_Language	Author	Creation_Date	Transcription	Genre
▶	9751	ab	at		Distinctio veritatis ipsa ducimus temporibus sed....	id
*	NULL	NULL	NULL	NULL	NULL	NULL

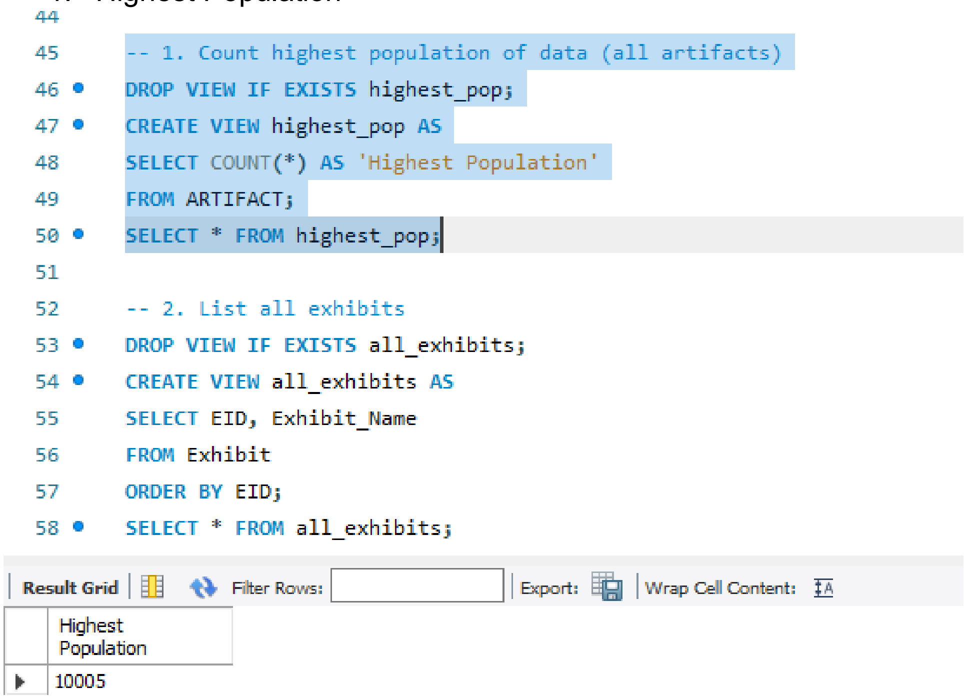
Item



	AID	Date_Discovered	Relevant_Date_Range	Creator
▶	1		VII	Vilma Hansen
*	NULL	NULL	NULL	NULL

Specific Queries

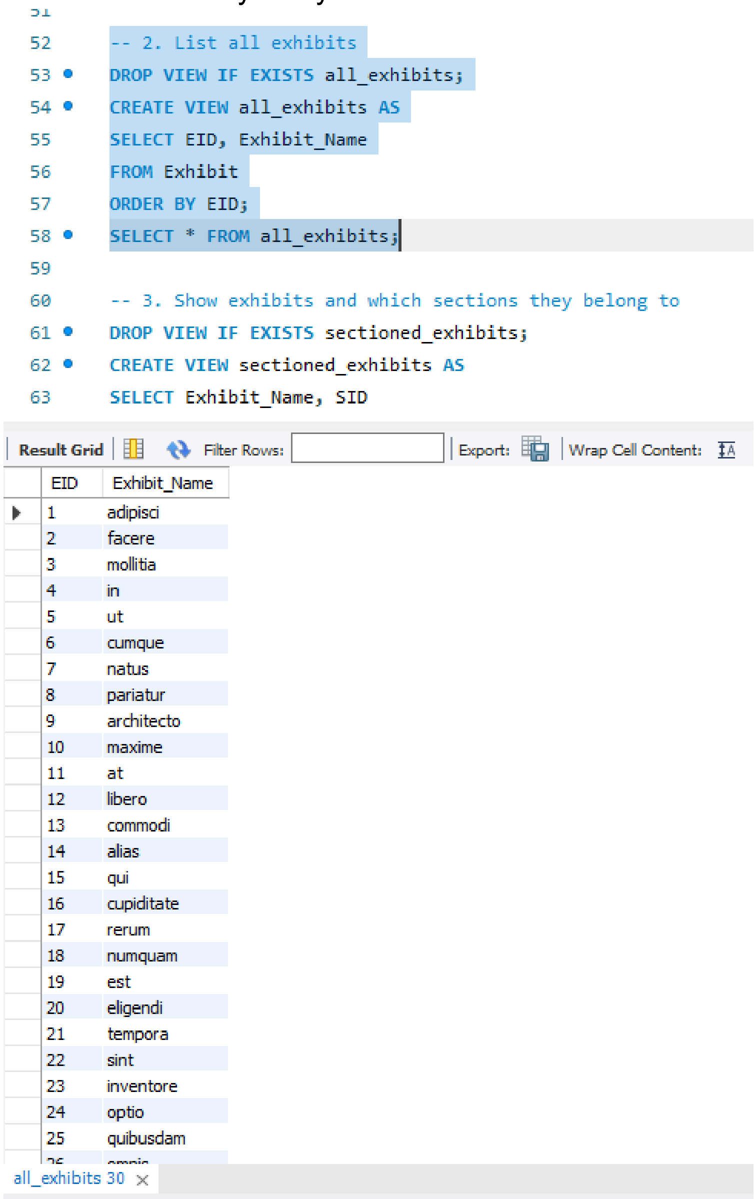
1. Highest Population



```
-- 1. Count highest population of data (all artifacts)
DROP VIEW IF EXISTS highest_pop;
CREATE VIEW highest_pop AS
SELECT COUNT(*) AS 'Highest Population'
FROM ARTIFACT;
SELECT * FROM highest_pop;

-- 2. List all exhibits
DROP VIEW IF EXISTS all_exhibits;
CREATE VIEW all_exhibits AS
SELECT EID, Exhibit_Name
FROM Exhibit
ORDER BY EID;
SELECT * FROM all_exhibits;
```

Highest Population
10005

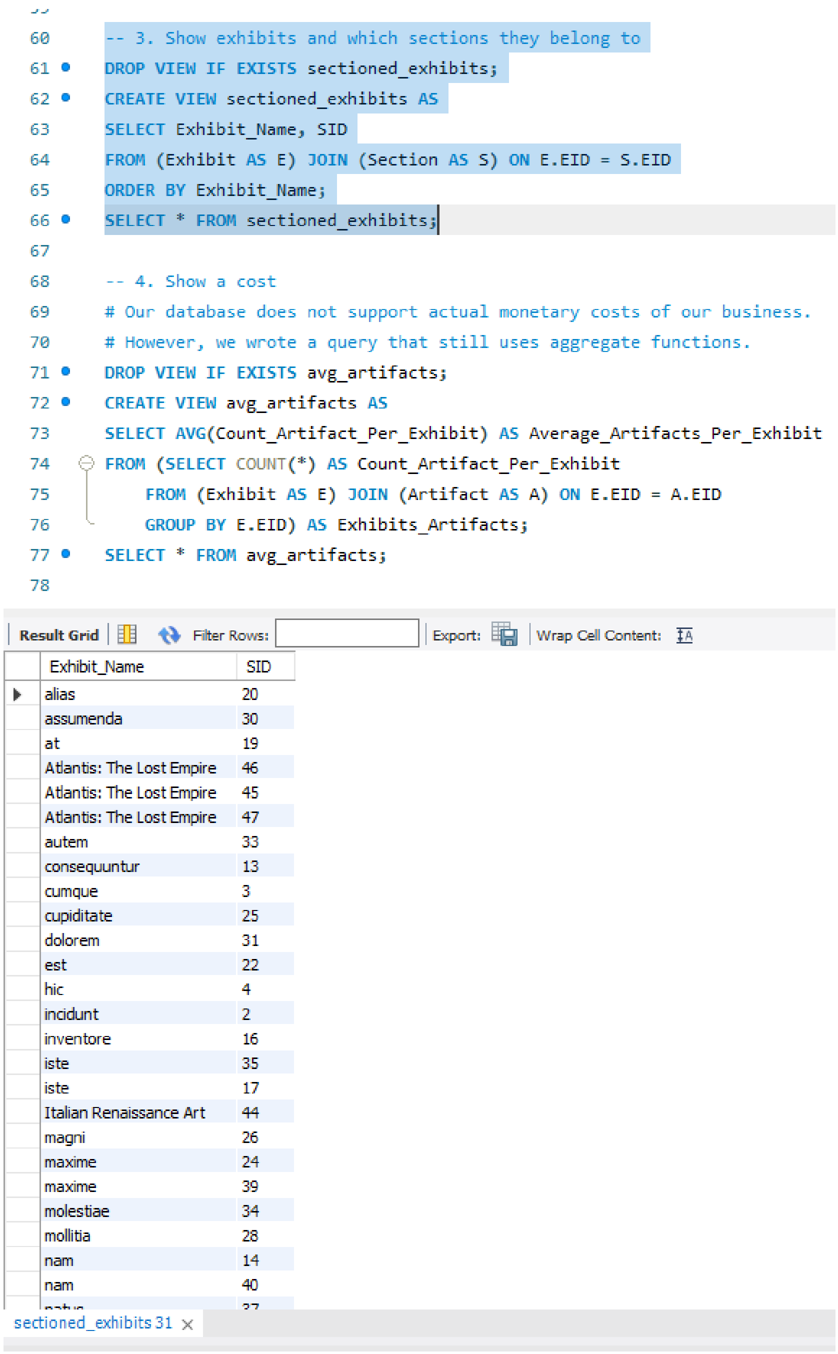
2. List of a key entity: Exhibits



```
-- 2. List all exhibits
DROP VIEW IF EXISTS all_exhibits;
CREATE VIEW all_exhibits AS
SELECT EID, Exhibit_Name
FROM Exhibit
ORDER BY EID;
SELECT * FROM all_exhibits;

-- 3. Show exhibits and which sections they belong to
DROP VIEW IF EXISTS sectioned_exhibits;
CREATE VIEW sectioned_exhibits AS
SELECT Exhibit_Name, SID
```

EID	Exhibit_Name
1	adipisci
2	facere
3	mollitia
4	in
5	ut
6	cumque
7	natus
8	pariatur
9	architecto
10	maxime
11	at
12	libero
13	commodi
14	alias
15	qui
16	cupiditate
17	rerum
18	numquam
19	est
20	eligendi
21	tempora
22	sint
23	inventore
24	optio
25	quibusdam
26	omnis

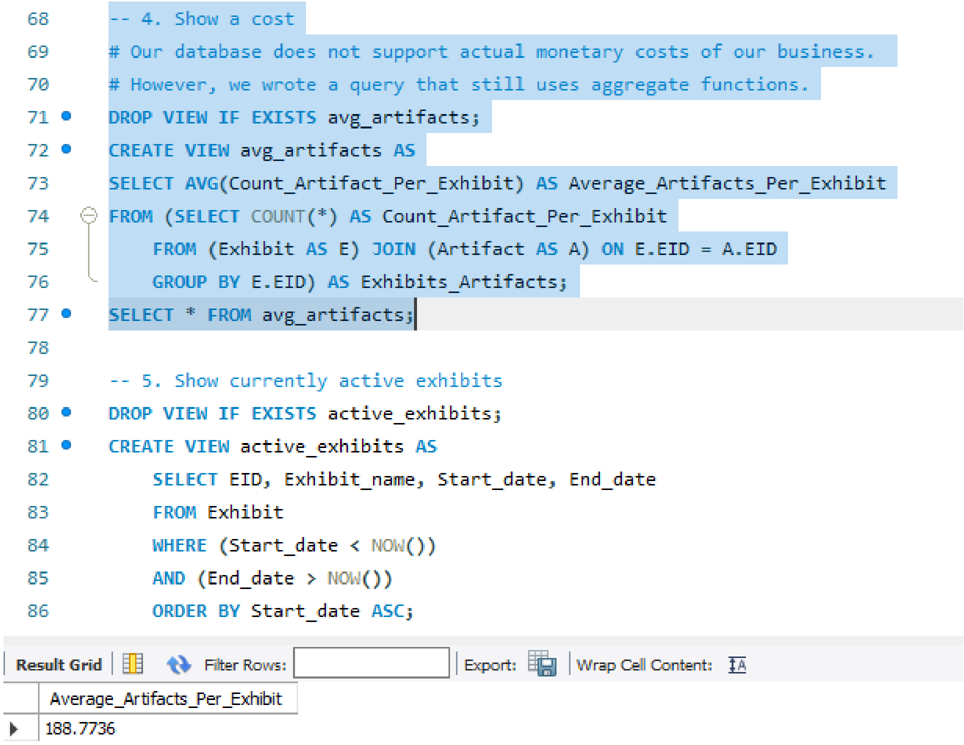
3. Joined list: Exhibits and their sections



```sql
-- 3. Show exhibits and which sections they belong to
DROP VIEW IF EXISTS sectioned_exhibits;
CREATE VIEW sectioned_exhibits AS
SELECT Exhibit_Name, SID
FROM (Exhibit AS E) JOIN (Section AS S) ON E.EID = S.EID
ORDER BY Exhibit_Name;
SELECT * FROM sectioned_exhibits;

-- 4. Show a cost
# Our database does not support actual monetary costs of our business.
# However, we wrote a query that still uses aggregate functions.
DROP VIEW IF EXISTS avg_artifacts;
CREATE VIEW avg_artifacts AS
SELECT AVG(Count_Artifact_Per_Exhibit) AS Average_Artifacts_Per_Exhibit
FROM (SELECT COUNT(*) AS Count_Artifact_Per_Exhibit
    FROM (Exhibit AS E) JOIN (Artifact AS A) ON E.EID = A.EID
    GROUP BY E.EID) AS Exhibits_Artifacts;
SELECT * FROM avg_artifacts;
```

Exhibit_Name	SID
alias	20
assumenda	30
at	19
Atlantis: The Lost Empire	46
Atlantis: The Lost Empire	45
Atlantis: The Lost Empire	47
autem	33
consequuntur	13
cumque	3
cupiditate	25
dolorem	31
est	22
hic	4
incidunt	2
inventore	16
iste	35
iste	17
Italian Renaissance Art	44
magni	26
maxime	24
maxime	39
molestiae	34
mollitia	28
nam	14
nam	40
natus	37

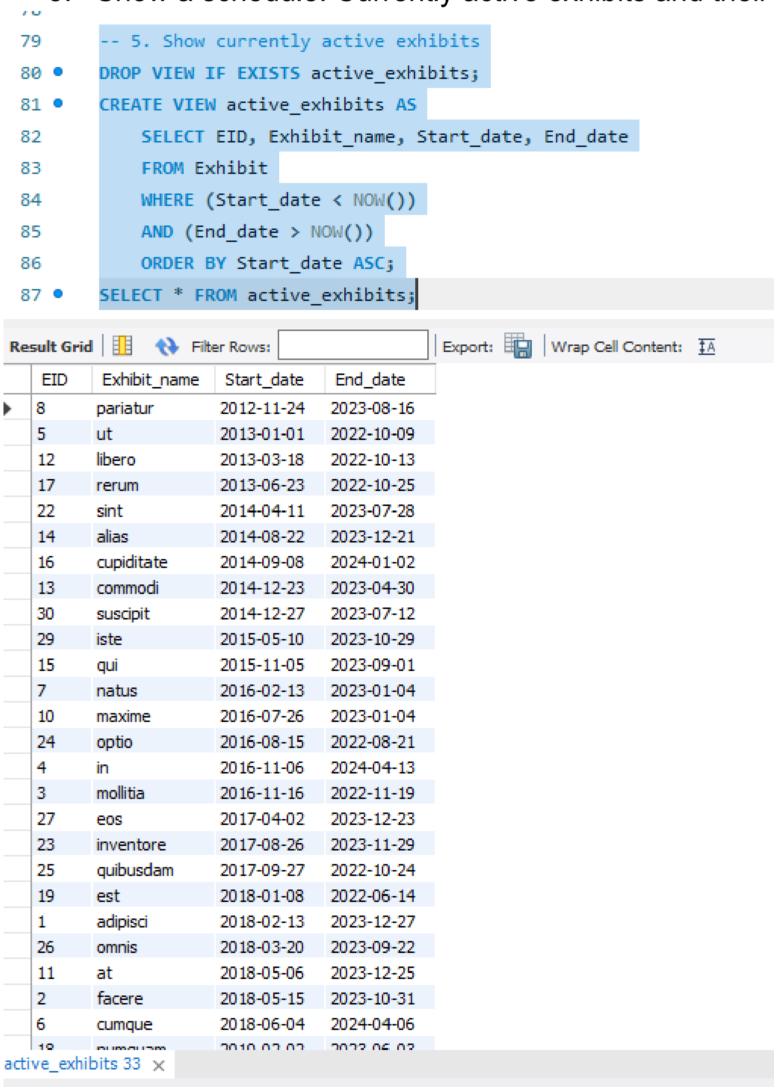
4. Show a cost: Average number of artifacts per exhibit
   (Our database doesn’t contain standard cost values)



```sql
-- 4. Show a cost
# Our database does not support actual monetary costs of our business.
# However, we wrote a query that still uses aggregate functions.
DROP VIEW IF EXISTS avg_artifacts;
CREATE VIEW avg_artifacts AS
SELECT AVG(Count_Artifact_Per_Exhibit) AS Average_Artifacts_Per_Exhibit
FROM (SELECT COUNT(*) AS Count_Artifact_Per_Exhibit
    FROM (Exhibit AS E) JOIN (Artifact AS A) ON E.EID = A.EID
    GROUP BY E.EID) AS Exhibits_Artifacts;
SELECT * FROM avg_artifacts;

-- 5. Show currently active exhibits
DROP VIEW IF EXISTS active_exhibits;
CREATE VIEW active_exhibits AS
    SELECT EID, Exhibit_name, Start_date, End_date
    FROM Exhibit
    WHERE (Start_date < NOW())
    AND (End_date > NOW())
    ORDER BY Start_date ASC;
```

Average_Artifacts_Per_Exhibit
188.7736

5. Show a schedule: Currently active exhibits and their start/end dates



```
-- 5. Show currently active exhibits
DROP VIEW IF EXISTS active_exhibits;
CREATE VIEW active_exhibits AS
    SELECT EID, Exhibit_name, Start_date, End_date
    FROM Exhibit
    WHERE (Start_date < NOW())
    AND (End_date > NOW())
    ORDER BY Start_date ASC;
SELECT * FROM active_exhibits;
```

EID	Exhibit_name	Start_date	End_date
8	pariatur	2012-11-24	2023-08-16
5	ut	2013-01-01	2022-10-09
12	libero	2013-03-18	2022-10-13
17	rerum	2013-06-23	2022-10-25
22	sint	2014-04-11	2023-07-28
14	alias	2014-08-22	2023-12-21
16	cupiditate	2014-09-08	2024-01-02
13	commodi	2014-12-23	2023-04-30
30	suscipit	2014-12-27	2023-07-12
29	iste	2015-05-10	2023-10-29
15	qui	2015-11-05	2023-09-01
7	natus	2016-02-13	2023-01-04
10	maxime	2016-07-26	2023-01-04
24	optio	2016-08-15	2022-08-21
4	in	2016-11-06	2024-04-13
3	mollitia	2016-11-16	2022-11-19
27	eos	2017-04-02	2023-12-23
23	inventore	2017-08-26	2023-11-29
25	quibusdam	2017-09-27	2022-10-24
19	est	2018-01-08	2022-06-14
1	adipisci	2018-02-13	2023-12-27
26	omnis	2018-03-20	2023-09-22
11	at	2018-05-06	2023-12-25
2	facere	2018-05-15	2023-10-31
6	cumque	2018-06-04	2024-04-06
18	numquam	2019-02-02	2023-06-03