



UNIVERSITY OF ALBERTA
ALBERTA SCHOOL OF BUSINESS

Department of Accounting and Business Analytics

BTM 211

Management Information Systems

DB Assignment – Winter 2025

Samia Musaddique Katingiri

March 14, 2025

Version [3.0]

Table of Contents

Table of Contents	1
Case Study: Alberta Aerospace Museum	3



	3
Background	3
Problem	3
Requirements:	4
Part A (15 marks):	4
Part B (15 marks):	5
Part C (20 marks):	6
Part D (50 marks):	8
Query 1 (5 marks):	8
Query 2 (5 marks):	9
Query 3 (5 marks):	10
Query 4 (5 marks):	11
Query 5 (5 marks):	12
Query 6 (5 marks):	13

Query 7 (5 marks):	14
Query 8 (5 marks):	15
Query 9 (10 marks):	16
Submission Instructions	17
Cheating and Plagiarism	18

Case Study: Alberta Aerospace Museum



Background

Established in 2018, the Alberta Aerospace Museum (AAM) has become the standard for aerospace education in Canada. While its priority lies in showcasing Canada's contribution to space exploration, historic pieces from some of the most legendary space exploration missions can be seen at this state-of-the-art institution.

Located on the outskirts of Edmonton, AAM is complete with a modern education center and its most impressive addition, a vast hangar that houses larger aircraft and ongoing restoration projects. AAM's curator, Christina Hadfield, prides herself on ensuring that the museum always offers top-of-the-line interactive exhibits that offer educational fun for everyone!

Problem

Christina is looking to implement a database for AAM so she can manage data more effectively across the museum. The database will be used not only by her but also by employees across the museum at various different levels.

Using the data model provided, create a database for AAM in SQLite.

Requirements:

Part A (15 marks):

Create the SQLite database **tables**.

Build an SQLite database of TABLES that match the official "*Alberta_Aerospace_Museum_Data_Model.pdf*" provided in the Assignments folder on eClass.

IMPORTANT: Do **NOT** use your personal data model that you built in the Data Model Assignment

1. Create each **table** with the correct name. You should create nine (9) tables.
2. Create all **attributes** for each table with the correct name and SQLite data types.
3. Implement the **primary keys** (or composite keys)
4. Implement each relationship for every **foreign key**.

Part B (15 marks):

Import data into the SQLite database.

Using the techniques you learned in the lab videos, import the provided sample data (AlbertaAerospaceMuseum_Data_2025_Winter1.xlsx) into each table.

1. Prepare a CSV (Comma Delimited) file for each table you intend to import into your SQLite database. Name each CSV file the name of the table. You should have nine (9) CSV files in total.

Note: You will need to reformat and “clean up” the sample data spreadsheet by cutting and re-pasting some of the columns from each table into its own CSV file.

2. Using SQLite Studios "Tools - Import" function, populate your database tables with the appropriate CSV files.

Hint: First import the “Parent” tables that act as “1” tables in 1-M relationships – then import the “Child” tables.

Part C (20 marks):

Insert new data rows into tables.

Using the techniques you learned in the Labs, insert new data rows into tables

1. Create two (2) new Visitors in the Visitor table. Use the following information for the new visitors.

Attribute	New Visitor #1	New Visitor #2
VisitorID	2025	5202
VisitorAge	25	52
VisitorFeedbackType	Positive	Negative
VisitorFeedback	[Insert Your Own Fake Review]	[Insert Your Own Fake Review]

2. Create two (2) new loans in the Loan table and allocate two (2) loan items from the LoanItem table to each of the new loans.. Use the following information for the new loans:

New Loan 1:

LOAN				
LoanID:	20250301			
EffectiveDate	05-30-2024			
ExpiryDate	04-15-2025			
ItemID	ArtifactID	ReturnDate	Cost	InsurancePremium
1	4753	04-14-2025	1800000	10000
2	6754	04-14-2025	5000000	10000
LoanerMuseumID	13			
LoanerSignatureDate	05-22-2024			
LoaneeMuseumID	1			
LoaneeSignatureDate	05-23-2024			

New Loan 2:

LOAN				
LoanID:	20250103			
EffectiveDate	09-04-2023			
ExpiryDate	03-25-2024			
ItemID	ArtifactID	ReturnDate	Cost	InsurancePremium
1	5219	03-22-2024	1660000	10000
2	7042	03-21-2024	1819000	10000
LoanerMuseumID	1			
LoanerSignatureDate	08-25-2023			
LoaneeMuseumID	3			
LoaneeSignatureDate	08-28-2023			

Part D (50 marks):

Query the data in the SQLite database using **Select**.

Using the techniques you learned in the labs, write a SELECT QUERY to answer the following questions based on the data in your SQLite database.

IMPORTANT: You **MUST** include your “working” SQL statements in this document. The code should not include any line numbers. We will copy these SQL code and test run with your database. Failure to submit working SQL will result in deductions.

Query 1 (5 marks):

Christina wants to pull a list of every exhibit and its description from her database. Write a query to display the following information (from left to right):

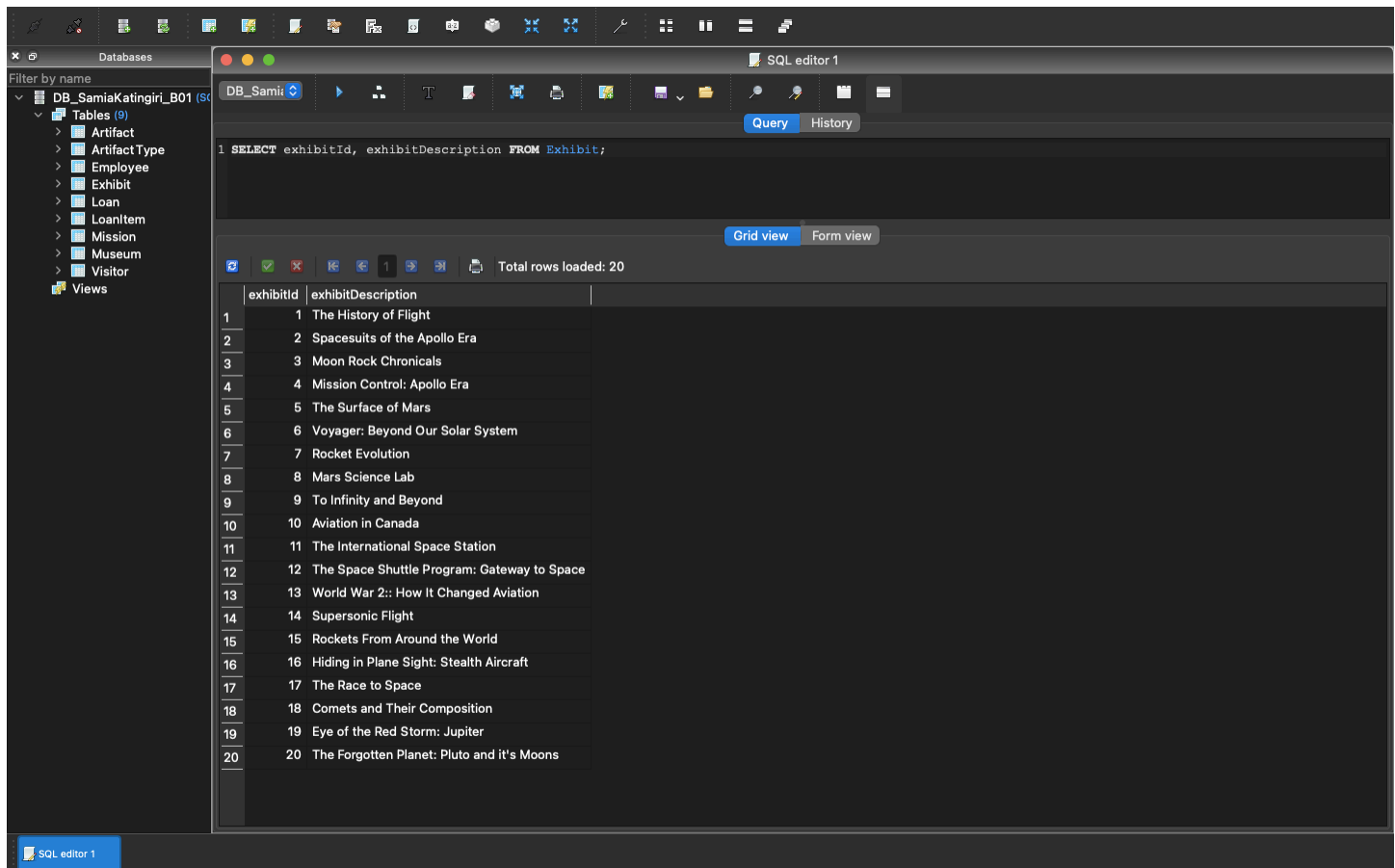
- ExhibitID
- ExhibitDescription

Important:. Do not include the line numbers when you copy and paste in the blue box.

Copy and paste your working SQL statement in the box below:

```
SELECT exhibitId, exhibitDescription FROM Exhibit;
```

Output:



exhibitId	exhibitDescription
1	The History of Flight
2	Spacesuits of the Apollo Era
3	Moon Rock Chronicals
4	Mission Control: Apollo Era
5	The Surface of Mars
6	Voyager: Beyond Our Solar System
7	Rocket Evolution
8	Mars Science Lab
9	To Infinity and Beyond
10	Aviation in Canada
11	The International Space Station
12	The Space Shuttle Program: Gateway to Space
13	World War 2:: How It Changed Aviation
14	Supersonic Flight
15	Rockets From Around the World
16	Hiding in Plane Sight: Stealth Aircraft
17	The Race to Space
18	Comets and Their Composition
19	Eye of the Red Storm: Jupiter
20	The Forgotten Planet: Pluto and it's Moons

Query 2 (5 marks):

Christina wants to see every row and column from the Visitor table. Write a query to display the following information (from left to right):

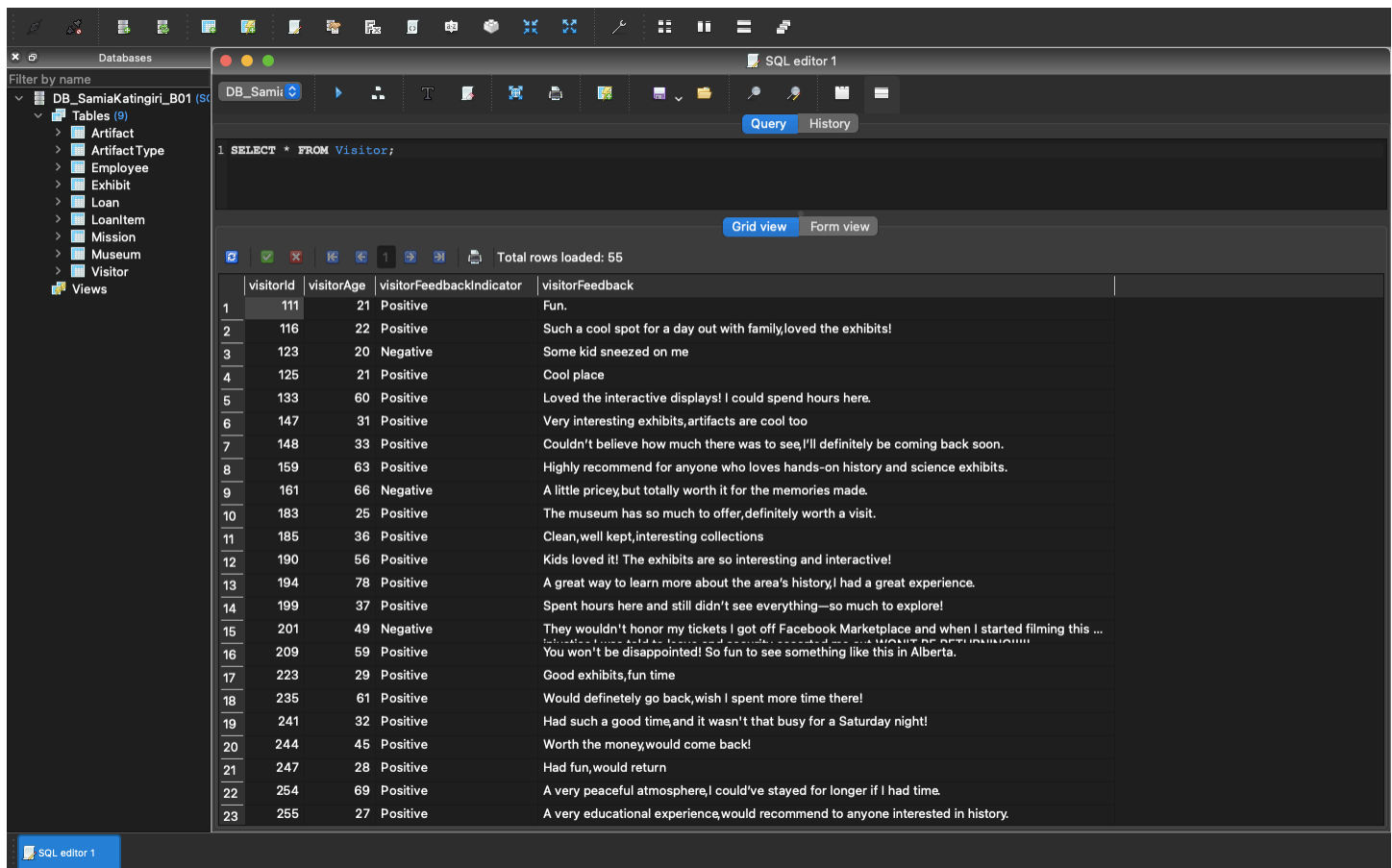
- VisitorID
- VisitorAge
- VisitorFeedbackType
- VisitorFeedback

Important:. Do not include the line numbers in the blue box.

Copy and paste your working SQL statement in the box below:

```
SELECT * FROM Visitor;
```

Output:



SQL editor 1

DB_Samir

1 SELECT * FROM Visitor;

Grid view Form view

Total rows loaded: 55

	visitorid	visitorAge	visitorFeedbackIndicator	visitorFeedback
1	111	21	Positive	Fun.
2	116	22	Positive	Such a cool spot for a day out with family,loved the exhibits!
3	123	20	Negative	Some kid sneezed on me
4	125	21	Positive	Cool place
5	133	60	Positive	Loved the interactive displays! I could spend hours here.
6	147	31	Positive	Very interesting exhibits,artifacts are cool too
7	148	33	Positive	Couldn't believe how much there was to see,I'll definitely be coming back soon.
8	159	63	Positive	Highly recommend for anyone who loves hands-on history and science exhibits.
9	161	66	Negative	A little pricey,but totally worth it for the memories made.
10	183	25	Positive	The museum has so much to offer,definitely worth a visit.
11	185	36	Positive	Clean,well kept,interesting collections
12	190	56	Positive	Kids loved it! The exhibits are so interesting and interactive!
13	194	78	Positive	A great way to learn more about the area's history,I had a great experience.
14	199	37	Positive	Spent hours here and still didn't see everything—so much to explore!
15	201	49	Negative	They wouldn't honor my tickets I got off Facebook Marketplace and when I started filming this ...
16	209	59	Positive	You won't be disappointed! So fun to see something like this in Alberta.
17	223	29	Positive	Good exhibits,fun time
18	235	61	Positive	Would definitely go back,wish I spent more time there!
19	241	32	Positive	Had such a good time,and it wasn't that busy for a Saturday night!
20	244	45	Positive	Worth the money,would come back!
21	247	28	Positive	Had fun,would return
22	254	69	Positive	A very peaceful atmosphere,I could've stayed for longer if I had time.
23	255	27	Positive	A very educational experience,would recommend to anyone interested in history.

Query 3 (5 marks):

Christina wants a list of every employee, but she only wants to include their ID, name, and job type. However, Christina wants the first names and last names to be in one column.

Write a query to display the following information (from left to right). Concatenate the employee names and rename them with the alias name provided in quotation marks:

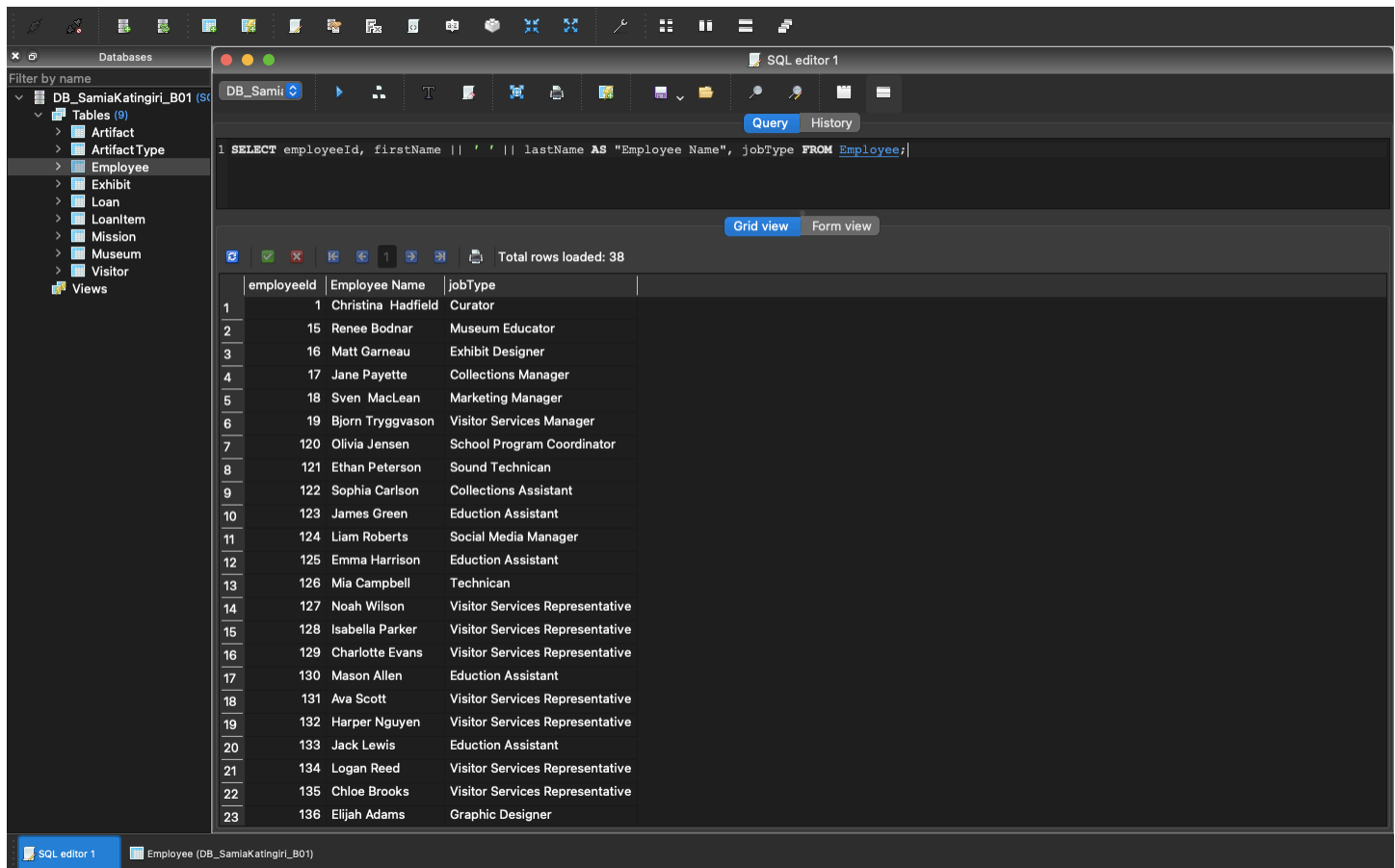
- EmployeeID
- FirstName and LastName as "Employee Name" (separate each name with a space)
- JobType

Important: Do not include the line numbers in the blue box.

Copy and paste your working SQL statement in the box below:

```
SELECT employeeId, firstName || ' ' || lastName AS "Employee Name", jobType FROM Employee;
```

Output:



The screenshot shows a SQL editor window titled "SQL editor 1" with a database connection "DB_SamiaKatingiri_B01". The query editor contains the following SQL statement:

```
1 SELECT employeeId, firstName || ' ' || lastName AS "Employee Name", jobType FROM Employee;
```

The results are displayed in a table with 38 rows. The columns are "employeeId", "Employee Name", and "jobType".

	employeeId	Employee Name	jobType
1	1	Christina Hadfield	Curator
2	15	Renee Bodnar	Museum Educator
3	16	Matt Garneau	Exhibit Designer
4	17	Jane Payette	Collections Manager
5	18	Sven MacLean	Marketing Manager
6	19	Bjorn Tryggvason	Visitor Services Manager
7	120	Olivia Jensen	School Program Coordinator
8	121	Ethan Peterson	Sound Technician
9	122	Sophia Carlson	Collections Assistant
10	123	James Green	Education Assistant
11	124	Liam Roberts	Social Media Manager
12	125	Emma Harrison	Education Assistant
13	126	Mia Campbell	Technician
14	127	Noah Wilson	Visitor Services Representative
15	128	Isabella Parker	Visitor Services Representative
16	129	Charlotte Evans	Visitor Services Representative
17	130	Mason Allen	Education Assistant
18	131	Ava Scott	Visitor Services Representative
19	132	Harper Nguyen	Visitor Services Representative
20	133	Jack Lewis	Education Assistant
21	134	Logan Reed	Visitor Services Representative
22	135	Chloe Brooks	Visitor Services Representative
23	136	Elijah Adams	Graphic Designer

Query 4 (5 marks):

Christina wants a list of the exhibit ID's, descriptions, budgets, and square-foot quantities, and she wants them to be ordered by budget in ascending order. Write a query to display the following information (from left to right):

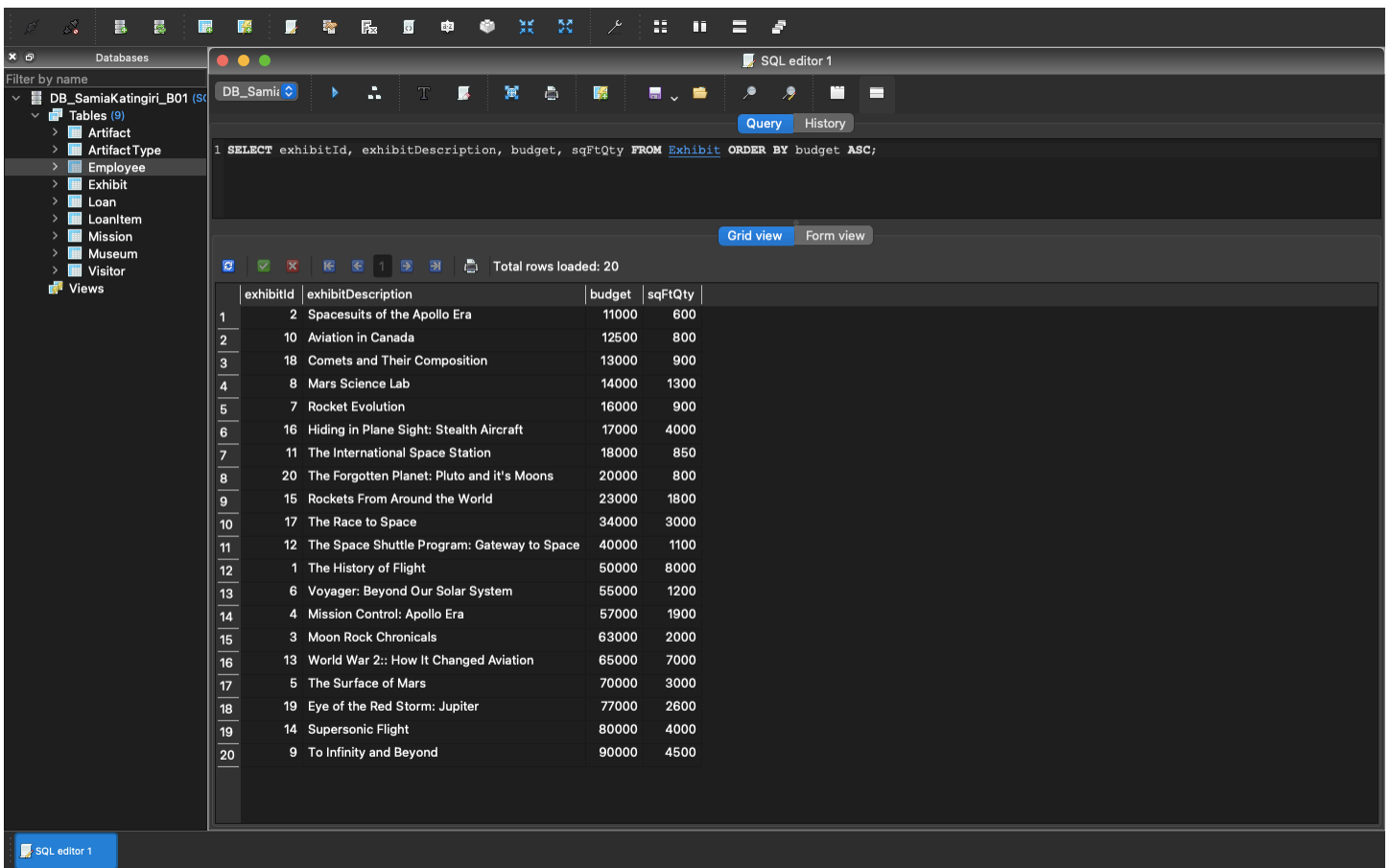
- ExhibitID
- ExhibitDescription
- Budget
- SqFtQty

Important: Do not include the line numbers in the blue box.

Copy and paste your working SQL statement in the box below:

```
SELECT exhibitId, exhibitDescription, budget, sqFtQty FROM Exhibit ORDER BY budget ASC;
```

Output:



The screenshot shows a database management tool interface. On the left, a sidebar lists databases and tables. The main area displays the SQL query: `1 SELECT exhibitId, exhibitDescription, budget, sqFtQty FROM Exhibit ORDER BY budget ASC;`. Below the query, a table shows the results of the query, ordered by budget in ascending order. The table has four columns: exhibitId, exhibitDescription, budget, and sqFtQty. The results are as follows:

	exhibitId	exhibitDescription	budget	sqFtQty
1	2	Spacesuits of the Apollo Era	11000	600
2	10	Aviation in Canada	12500	800
3	18	Comets and Their Composition	13000	900
4	8	Mars Science Lab	14000	1300
5	7	Rocket Evolution	16000	900
6	16	Hiding in Plain Sight: Stealth Aircraft	17000	4000
7	11	The International Space Station	18000	850
8	20	The Forgotten Planet: Pluto and its Moons	20000	800
9	15	Rockets From Around the World	23000	1800
10	17	The Race to Space	34000	3000
11	12	The Space Shuttle Program: Gateway to Space	40000	1100
12	1	The History of Flight	50000	8000
13	6	Voyager: Beyond Our Solar System	55000	1200
14	4	Mission Control: Apollo Era	57000	1900
15	3	Moon Rock Chronicals	63000	2000
16	13	World War 2: How It Changed Aviation	65000	7000
17	5	The Surface of Mars	70000	3000
18	19	Eye of the Red Storm: Jupiter	77000	2600
19	14	Supersonic Flight	80000	4000
20	9	To Infinity and Beyond	90000	4500

Query 5 (5 marks):

Christina wants a list of every artifact, but only if that artifact has a mission code associated with it. Write a query to display the following information (from left to right):

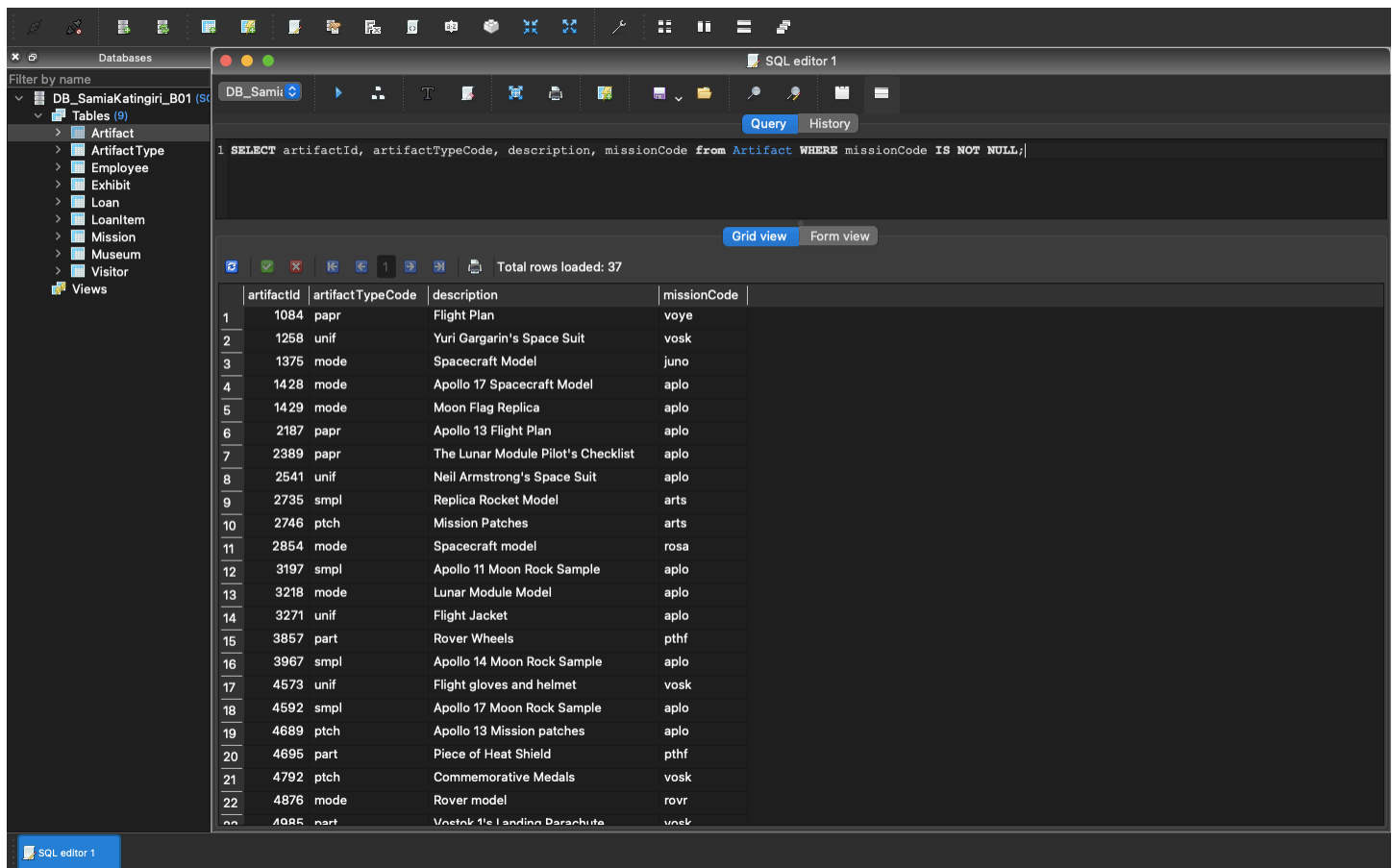
- ArtifactID
- ArtifactTypeCode
- ArtifactDescription
- MissionCode

Important: Do not include the line numbers in the blue box.

Copy and paste your working SQL statement in the box below:

```
SELECT artifactId, artifactTypeCode, description, missionCode from Artifact WHERE missionCode IS NOT NULL;
```

Output:



The screenshot shows a database management tool interface. On the left, a sidebar lists databases and tables. The main window displays the SQL query: `SELECT artifactId, artifactTypeCode, description, missionCode from Artifact WHERE missionCode IS NOT NULL;`. Below the query, a table of results is shown with 37 rows. The table has four columns: artifactId, artifactTypeCode, description, and missionCode. The results list various artifacts such as Flight Plan, Yuri Gagarin's Space Suit, Spacecraft Model, and Apollo 17 Spacecraft Model, each with a corresponding mission code.

	artifactId	artifactTypeCode	description	missionCode
1	1084	papr	Flight Plan	voye
2	1258	unif	Yuri Gagarin's Space Suit	vosk
3	1375	mode	Spacecraft Model	juno
4	1428	mode	Apollo 17 Spacecraft Model	aplo
5	1429	mode	Moon Flag Replica	aplo
6	2187	papr	Apollo 13 Flight Plan	aplo
7	2389	papr	The Lunar Module Pilot's Checklist	aplo
8	2541	unif	Neil Armstrong's Space Suit	aplo
9	2735	smpl	Replica Rocket Model	arts
10	2746	ptch	Mission Patches	arts
11	2854	mode	Spacecraft model	rosa
12	3197	smpl	Apollo 11 Moon Rock Sample	aplo
13	3218	mode	Lunar Module Model	aplo
14	3271	unif	Flight Jacket	aplo
15	3857	part	Rover Wheels	pthf
16	3967	smpl	Apollo 14 Moon Rock Sample	aplo
17	4573	unif	Flight gloves and helmet	vosk
18	4592	smpl	Apollo 17 Moon Rock Sample	aplo
19	4689	ptch	Apollo 13 Mission patches	aplo
20	4695	part	Piece of Heat Shield	pthf
21	4792	ptch	Commemorative Medals	vosk
22	4876	mode	Rover model	rovr
23	4985	part	Vostok 1's Landing Parachute	vosk

Query 6 (5 marks):

Christina wants a list of every loan item and she wants to know the total cost of that loan item. However, he only wants to see the total revenue of instruments that have an insurance premium applied. Write a query to display the following information (from left to right):

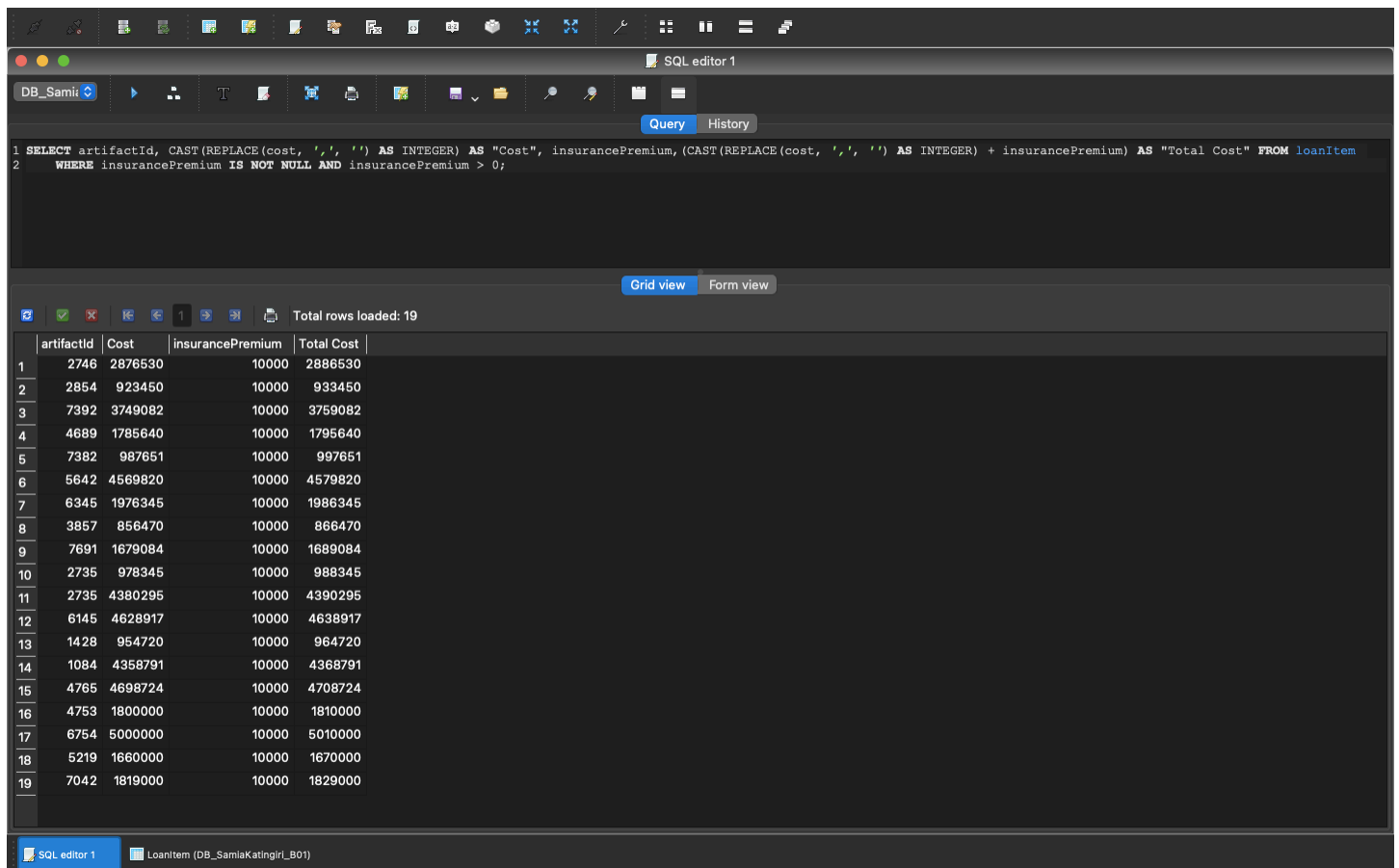
- ArtifactID
- Cost
- InsurancePremium
- A new column called "Total Cost"

Important: Do not include the line numbers in the blue box.

Copy and paste your working SQL statement in the box below:

```
SELECT artifactId, CAST(REPLACE(cost, ',', '')) AS "Cost",  
insurancePremium, (CAST(REPLACE(cost, ',', '')) AS INTEGER) + insurancePremium) AS "Total Cost" FROM  
loanItem WHERE insurancePremium IS NOT NULL AND insurancePremium > 0;
```

Output:



The screenshot shows a SQL editor window titled "SQL editor 1" with a toolbar and a query history panel. The query is as follows:

```
1 SELECT artifactId, CAST(REPLACE(cost, ',', '')) AS "Cost", insurancePremium, (CAST(REPLACE(cost, ',', '')) AS INTEGER) + insurancePremium) AS "Total Cost" FROM loanItem  
2 WHERE insurancePremium IS NOT NULL AND insurancePremium > 0;
```

The output is displayed in a table with 19 rows. The columns are artifactId, Cost, insurancePremium, and Total Cost. The total rows loaded are 19.

	artifactId	Cost	insurancePremium	Total Cost
1	2746	2876530	10000	2886530
2	2854	923450	10000	933450
3	7392	3749082	10000	3759082
4	4689	1785640	10000	1795640
5	7382	987651	10000	997651
6	5642	4569820	10000	4579820
7	6345	1976345	10000	1986345
8	3857	856470	10000	866470
9	7691	1679084	10000	1689084
10	2735	978345	10000	988345
11	2735	4380295	10000	4390295
12	6145	4628917	10000	4638917
13	1428	954720	10000	964720
14	1084	4358791	10000	4368791
15	4765	4698724	10000	4708724
16	4753	1800000	10000	1810000
17	6754	5000000	10000	5010000
18	5219	1660000	10000	1670000
19	7042	1819000	10000	1829000

Query 7 (5 marks):

Christina wants a count of the number of artifacts in descending order that she has in her database grouped by their artifact type code. Write a query to display the following information (from left to right):

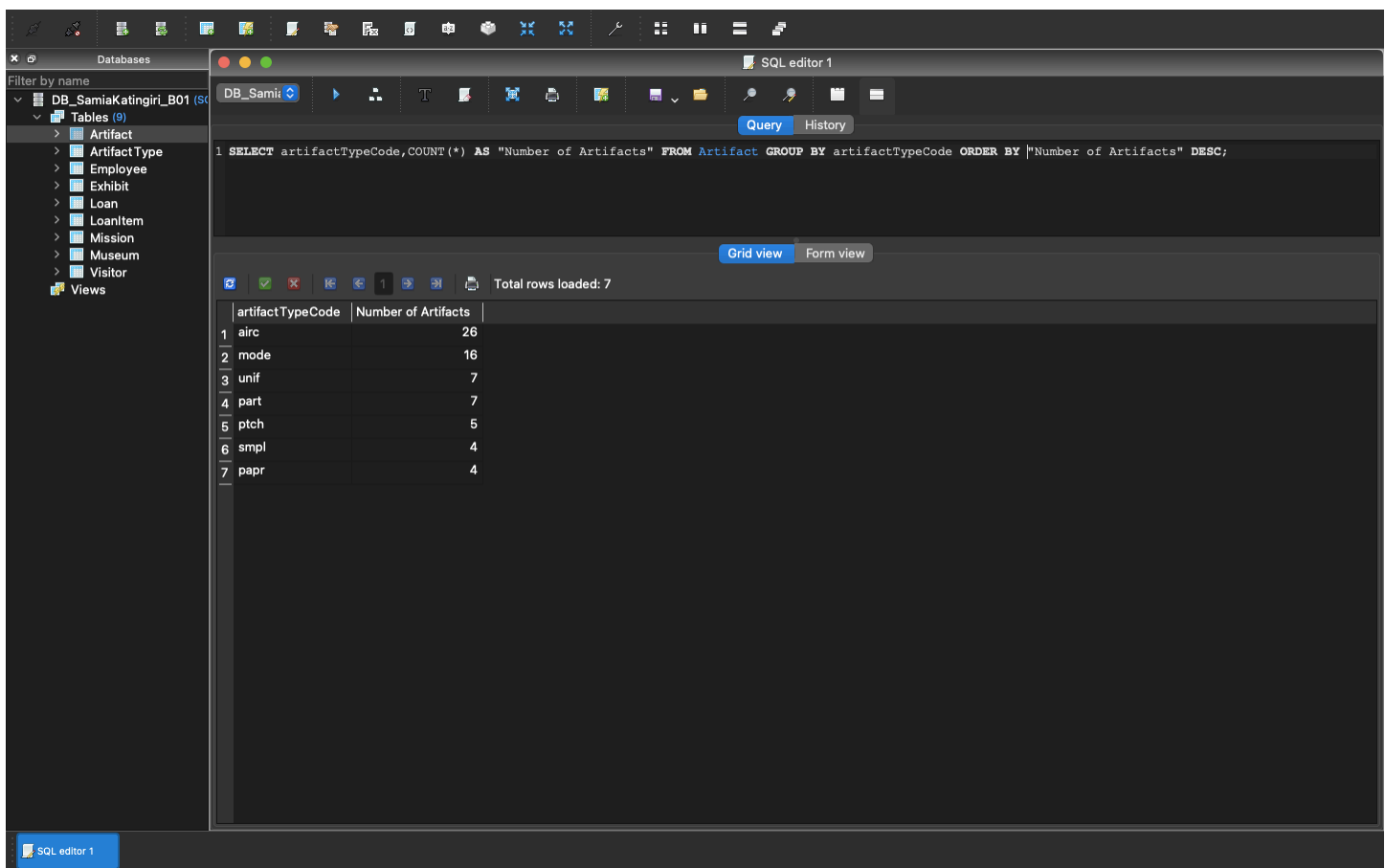
- ArtifactTypeCode
- A new column called "Number of Artifacts"

Important: Do not include the line numbers in the blue box.

Copy and paste your working SQL statement in the box below

```
SELECT artifactTypeCode,COUNT(*) AS "Number of Artifacts" FROM Artifact GROUP BY artifactTypeCode ORDER BY "Number of Artifacts" DESC;
```

Output:



The screenshot shows a database management interface with a left sidebar containing a tree view of databases and tables. The main area is split into two panes. The top pane shows the SQL query: `1 SELECT artifactTypeCode,COUNT(*) AS "Number of Artifacts" FROM Artifact GROUP BY artifactTypeCode ORDER BY "Number of Artifacts" DESC;`. The bottom pane shows the query results in a grid view. The results are as follows:

	artifactTypeCode	Number of Artifacts
1	airc	26
2	mode	16
3	unif	7
4	part	7
5	ptch	5
6	smpl	4
7	papr	4

Query 8 (5 marks):

Christina wants a list of every exhibit ID in ascending order, exhibit description, artifact ID, and artifact description. Write a query to display the following information (from left to right):

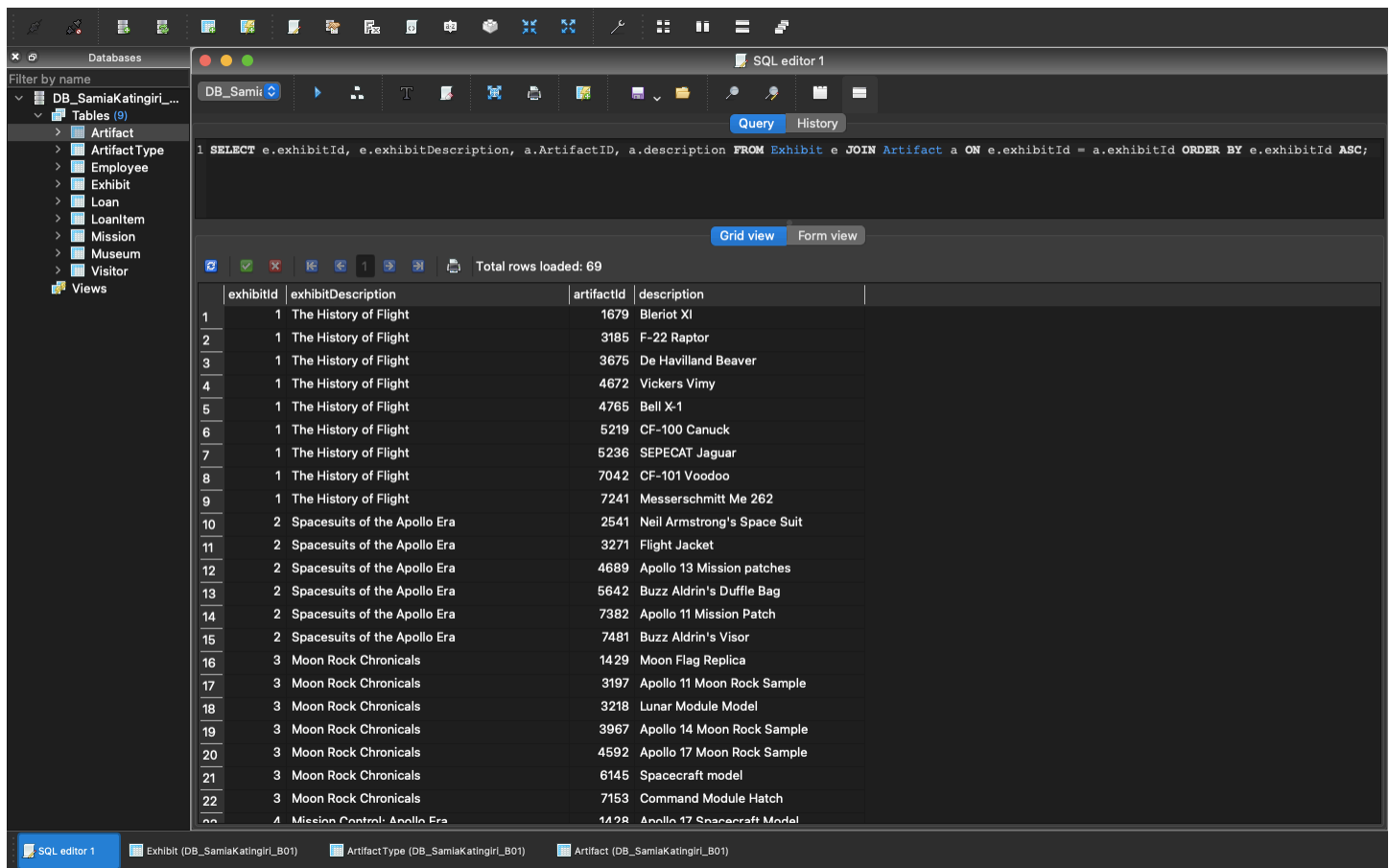
- ExhibitID
- ExhibitDescription
- ArtifactID
- ArtifactDescription

Important: Do not include the line numbers in the blue box.

Copy and paste your working SQL statement in the box below:

```
SELECT e.exhibitId, e.exhibitDescription, a.artifactId, a.description FROM Exhibit e JOIN Artifact a ON e.exhibitId = a.exhibitId ORDER BY e.exhibitId ASC;
```

Output:



The screenshot shows a SQL editor interface with a database schema on the left and a query editor on the right. The query is: `1 SELECT e.exhibitId, e.exhibitDescription, a.artifactId, a.description FROM Exhibit e JOIN Artifact a ON e.exhibitId = a.exhibitId ORDER BY e.exhibitId ASC;` The results are displayed in a grid view with 69 rows. The columns are exhibitId, exhibitDescription, artifactId, and description.

	exhibitId	exhibitDescription	artifactId	description
1	1	The History of Flight	1679	Bleriot XI
2	1	The History of Flight	3185	F-22 Raptor
3	1	The History of Flight	3675	De Havilland Beaver
4	1	The History of Flight	4672	Vickers Vimy
5	1	The History of Flight	4765	Bell X-1
6	1	The History of Flight	5219	CF-100 Canuck
7	1	The History of Flight	5236	SEPECAT Jaguar
8	1	The History of Flight	7042	CF-101 Voodoo
9	1	The History of Flight	7241	Messerschmitt Me 262
10	2	Spacesuits of the Apollo Era	2541	Neil Armstrong's Space Suit
11	2	Spacesuits of the Apollo Era	3271	Flight Jacket
12	2	Spacesuits of the Apollo Era	4689	Apollo 13 Mission patches
13	2	Spacesuits of the Apollo Era	5642	Buzz Aldrin's Duffle Bag
14	2	Spacesuits of the Apollo Era	7382	Apollo 11 Mission Patch
15	2	Spacesuits of the Apollo Era	7481	Buzz Aldrin's Visor
16	3	Moon Rock Chronicals	1429	Moon Flag Replica
17	3	Moon Rock Chronicals	3197	Apollo 11 Moon Rock Sample
18	3	Moon Rock Chronicals	3218	Lunar Module Model
19	3	Moon Rock Chronicals	3967	Apollo 14 Moon Rock Sample
20	3	Moon Rock Chronicals	4592	Apollo 17 Moon Rock Sample
21	3	Moon Rock Chronicals	6145	Spacecraft model
22	3	Moon Rock Chronicals	7153	Command Module Hatch
23	4	Mission Control: Apollo Era	1428	Apollo 17 Spacecraft Model

Query 9 (10 marks):

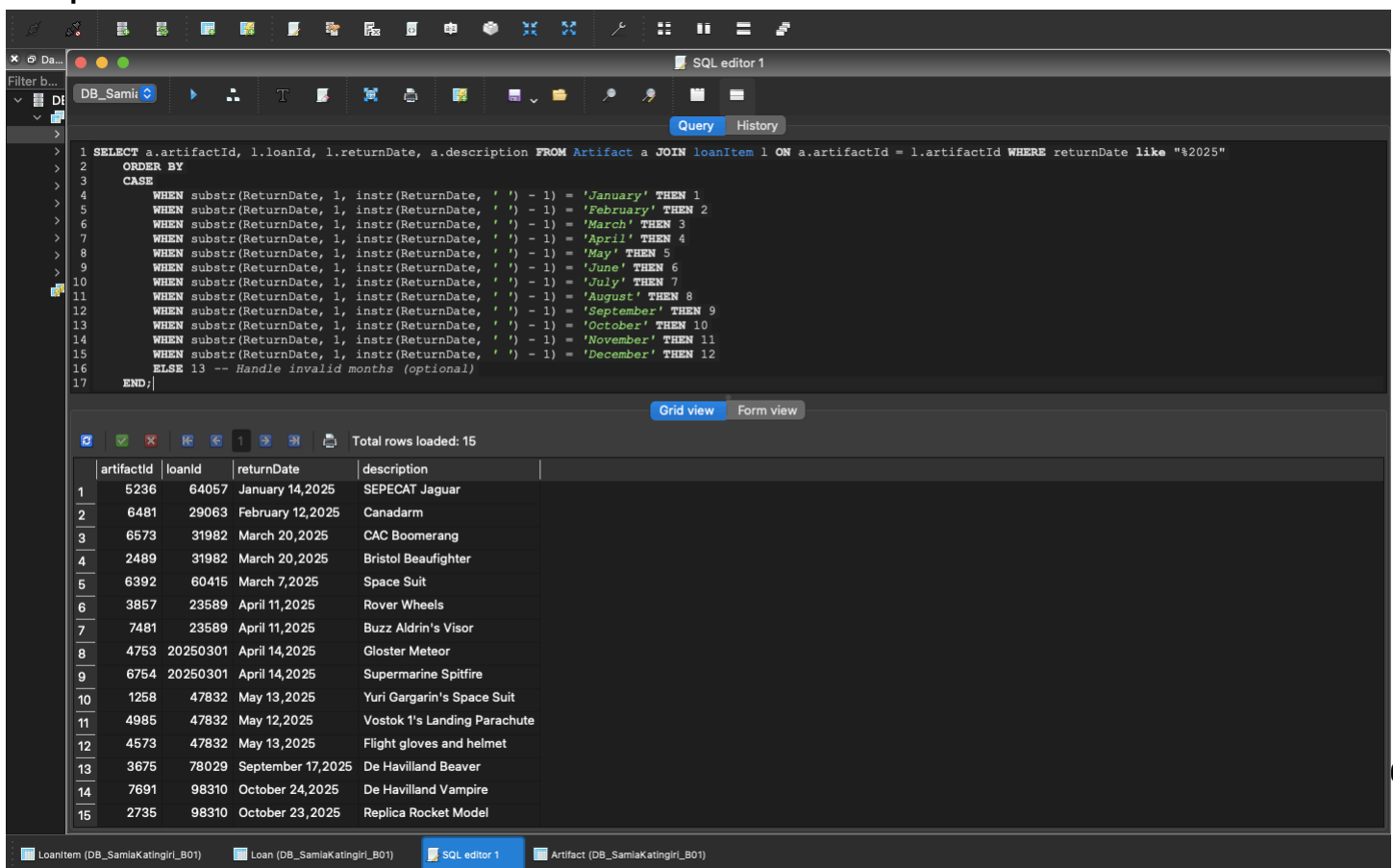
This query lists artifact ID, loan ID, return date of that artifact and its description based on order of the return date. This can be used as a reminder of the artifacts to be returned in the year 2025. The following tables are displayed/used from left-to-right:

- artifactId
- loanId
- returnDate
- artifactDescription

Copy and paste your working SQL statement in the box below:

```
SELECT a.artifactId, l.loanId, l.returnDate, a.description FROM Artifact a JOIN loanItem l ON a.artifactId = l.artifactId WHERE returnDate like "%2025"
ORDER BY
CASE
WHEN substr(ReturnDate, 1, instr(ReturnDate, ' ') - 1) = 'January' THEN 1
WHEN substr(ReturnDate, 1, instr(ReturnDate, ' ') - 1) = 'February' THEN 2
WHEN substr(ReturnDate, 1, instr(ReturnDate, ' ') - 1) = 'March' THEN 3
WHEN substr(ReturnDate, 1, instr(ReturnDate, ' ') - 1) = 'April' THEN 4
WHEN substr(ReturnDate, 1, instr(ReturnDate, ' ') - 1) = 'May' THEN 5
WHEN substr(ReturnDate, 1, instr(ReturnDate, ' ') - 1) = 'June' THEN 6
WHEN substr(ReturnDate, 1, instr(ReturnDate, ' ') - 1) = 'July' THEN 7
WHEN substr(ReturnDate, 1, instr(ReturnDate, ' ') - 1) = 'August' THEN 8
WHEN substr(ReturnDate, 1, instr(ReturnDate, ' ') - 1) = 'September' THEN 9
WHEN substr(ReturnDate, 1, instr(ReturnDate, ' ') - 1) = 'October' THEN 10
WHEN substr(ReturnDate, 1, instr(ReturnDate, ' ') - 1) = 'November' THEN 11
WHEN substr(ReturnDate, 1, instr(ReturnDate, ' ') - 1) = 'December' THEN 12
ELSE 13 -- Handle invalid months (optional)
END;
```

Output:



The screenshot shows a SQL editor interface with a query window and a results grid. The query is the same as the one provided in the previous block. The results grid displays 15 rows of data, sorted by return date. The columns are artifactId, loanId, returnDate, and description.

	artifactId	loanId	returnDate	description
1	5236	64057	January 14, 2025	SEPECAT Jaguar
2	6481	29063	February 12, 2025	Canadarm
3	6573	31982	March 20, 2025	CAC Boomerang
4	2489	31982	March 20, 2025	Bristol Beaufighter
5	6392	60415	March 7, 2025	Space Suit
6	3857	23589	April 11, 2025	Rover Wheels
7	7481	23589	April 11, 2025	Buzz Aldrin's Visor
8	4753	20250301	April 14, 2025	Gloster Meteor
9	6754	20250301	April 14, 2025	Supermarine Spitfire
10	1258	47832	May 13, 2025	Yuri Gargarin's Space Suit
11	4985	47832	May 12, 2025	Vostok 1's Landing Parachute
12	4573	47832	May 13, 2025	Flight gloves and helmet
13	3675	78029	September 17, 2025	De Havilland Beaver
14	7691	98310	October 24, 2025	De Havilland Vampire
15	2735	98310	October 23, 2025	Replica Rocket Model

Submission Instructions

Before the submission deadline, you must submit electronically the following:

- One (1) SQLite **database (.db) file**
 - **We will not accept any other file types.**
- One (1) completed DB Assignment REQUIREMENTS document with your SQL queries pasted in Section D.

File naming convention: *AssignmentCode_FirstNameLastName_LectureSection*

Example: **DB_LadyGaga_A02**

Cheating and Plagiarism

All work is to be done individually. Do not copy, in whole or in part, the work of others, including paper printouts, electronic files or computer programs. Do not use the work of others as a starting point and then modify it. All work submitted under your name must be yours and yours alone.

The University of Alberta is committed to the highest standards of academic integrity and honesty. Students are expected to be familiar with these standards regarding academic honesty and to uphold the policies of the University in this respect. Students are particularly urged to familiarize themselves with the provisions of the Code of Student Behavior (online at www.ualberta.ca/secretariat/appeals.htm) and avoid any behavior that could potentially result in suspicions of cheating, plagiarism, misrepresentation of facts and/or participation in an offence. Academic dishonesty is a serious offence and can result in suspension or expulsion from the University.