Mayur Sejpal (C2071044)

University of Law

Data Design Management

Prof. Varun Gupta

January 12th, 2023

<u>Table of Contents</u>

ASSIGNMENT DETAILS

You will need to design a database and implement your scripts for an online e-shop. The e-shop sells music records, movies and books to customers around the world. Consider the following use case specification:

- A customer submits orders for products
- A customer can place one order at a time
- An employee controls the products and is responsible for collecting the products for delivery
- The products are music records, movies, and books
- A customer can place an order using a credit card.
- The customer data includes: first name, last name, phone number, email, city, country
- The product data includes title, description, unit price, release data, language, and genre and product type

1. Design an ER diagram with the appropriate relationships and cardinality for the above use case.
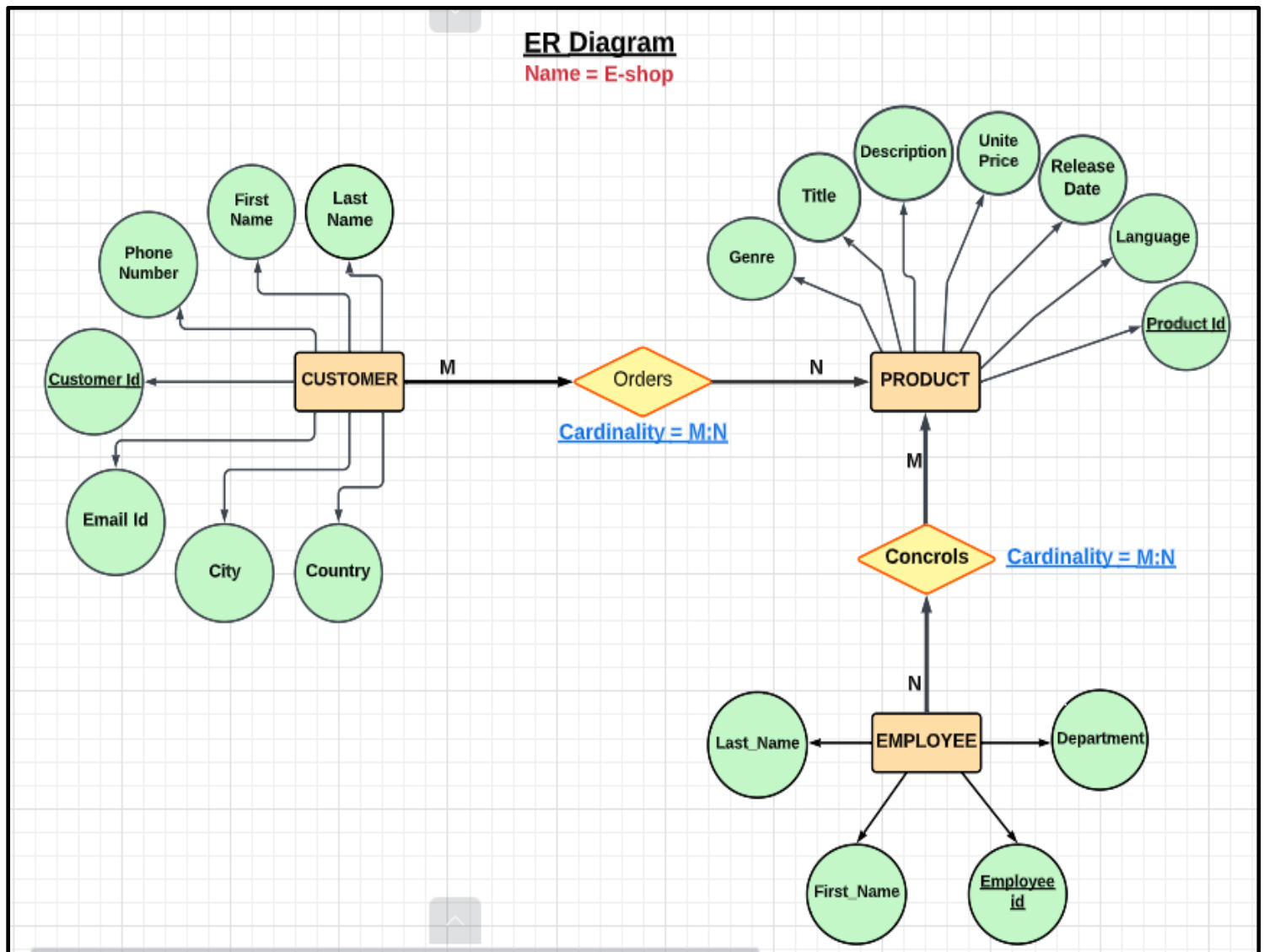
2. Develop the SQL scripts and the database for the ER diagram.

→ **CUSTOMER TABLE**

Assumption: CUSTOMER_ID is assumed as attribute as considered as primary key

Input   [ ]   ☾   ⋮   **Run SQL**

```
CREATE TABLE CUSTOMER (
  CUSTOMER_ID INTEGER PRIMARY KEY,
  FIRST_NAME VARCHAR (200),
  LAST_NAME VARCHAR (200),
  EMAIL_ID VARCHAR (200),
  CITY VARCHAR (200),
  COUNTRY VARCHAR (200),
  PHONE_NUMBER INTEGER
 );
```

→ **PRODUCT TABLE**

Assumption: PRODUCT_ID is assumed as attribute as considered as primary key

Input   [ ]   ☾   ⋮   **Run SQL**

```
CREATE TABLE PRODUCT (
  PRODUCT_ID INTEGER PRIMARY KEY,
  LANGUAGE VARCHAR (200),
  RELEASE_DATE VARCHAR (200),
  UNIT_PRICE VARCHAR (200),
  DISCREPTION VARCHAR (200),
  TITLE VARCHAR (200),
  GENRE VARCHAR (200)
 );
```

→ **EMPLOYEE TABLE**

Assumption: EMPLOYEE_ID is assumed as attribute as considered as primary key

Input   [ ]   ☾   ⋮   **Run SQL**

```
CREATE TABLE EMPLOYEE (
  EMPLOYEE_ID INTEGER PRIMARY KEY,
  FIRST_NAME VARCHAR (200),
  LAST_NAME VARCHAR (200),
  DEPARTMENT VARCHAR (200)
 );
```

DELIVERY CONTOL TABLE (Composite table)

```
Input                                          [ ]   G   :   Run SQL

CREATE TABLE DELIVERY_CONTROL (
PRODUCTS_IDS INTEGER,
EMPLOYEE_IDS INTEGER,
DELIVERY_STATUS VARCHAR (200),
FOREIGN KEY (PRODUCTS_IDS) REFERENCES PRODUCT (PRODUCT_ID),
FOREIGN KEY (EMPLOYEE_IDS) REFERENCES EMPLOYEE (EMPLOYEE_ID)
   );
```

ORDER TABLE (Composite table)

```
Input                                          [ ]   G   :   Run SQL

CREATE TABLE ORDERS (
CUSTOMER_IDS INTEGER,
PRODUCT_NOS INTEGER,
MODE_OF_PAYMENT VARCHAR (200),
QUANTITY INTEGER,
FOREIGN KEY (CUSTOMER_IDS) REFERENCES CUSTOMER (CUSTOMER_ID),
FOREIGN KEY (PRODUCT_NOS) REFERENCES PRODUCT (PRODUCT_ID)
);
```

3. Insert at least 5 records of your desired data in each table of the database, feel free to improvise in terms of the number of records.

➜ CUSTOMER

Input

```
INSERT INTO CUSTOMER (CUSTOMER_ID, PHONE_NUMBER, FIRST_NAME, LAST_NAME, EMAIL_ID, CITY, COUNTRY)
VALUES (100211, 499766117924, 'NITIN', 'THAKKAR', 'Nitin01@Gisma.com', 'PUNE', 'INDIA'),
       (200212, 478845965233, 'ARUN', 'SINGH', 'Arun.singh@yahoo.com', 'VANCOUVER', 'CANADA'),
       (300213, 917932865334, 'PANKAJ', 'KOTHARI', 'Pankaj007@gmail.com', 'CHICAGO', 'AMERICA'),
       (400214, 034478563211, 'RAJPAL', 'YADAV', 'Rajpal2022@yahoo.in', 'BEIJING', 'CHINA'),
       (600215, 499766652324, 'MASHESH', 'DESAI', 'Desai01@Gisma.com', 'PUNE', 'INDIA');
```

Output

CUSTOMER_ID	FIRST_NAME	LAST_NAME	EMAIL_ID	CITY	COUNTRY	PHONE_NUMBER
100211	NITIN	THAKKAR	Nitin01@Gisma.com	PUNE	INDIA	499766117924
200212	ARUN	SINGH	Arun.singh@yahoo.com	VANCOUVER	CANADA	478845965233
300213	PANKAJ	KOTHARI	Pankaj007@gmail.com	CHICAGO	AMERICA	917932865334
400214	RAJPAL	YADAV	Rajpal2022@yahoo.in	BEIJING	CHINA	34478563211
600215	MASHESH	DESAI	Desai01@Gisma.com	PUNE	INDIA	499766652324

➜ PRODUCT

Input   Run SQL

```
INSERT INTO PRODUCT (PRODUCT_ID, LANGUAGE, RELEASE_DATE, UNIT_PRICE, TITLE, DISCREPTION, GENRE)
VALUES (100021, 'ENGLISH', '23-09-2020', '120', 'HORROR_HUNTING', 'THE_TRAGEDY_OF_DEATH', 'MOVIE'),
       (200022, 'GERMAN', '18-11-2014', '180', 'BABLEON_SECREAT', 'THE_SECREAT_MYSTERY', 'BOOK'),
       (300023, 'HINDI', '22-07-2021', '350', 'LOVE_OF_TOMORROW', 'LOVE_SONGS', 'MUSIC'),
       (400024, 'SPANISH', '11-07-2015', '200', 'CRAZY_HUMAN', 'THE_MAN_WHO_MAD', 'DRAMA'),
       (500025, 'ENGLISH', '14-07-2008', '750', 'POPCON_BABY', 'CHILDHOOD_ENJOYMENT', 'MOVIE');
```

Output

PRODUCT_ID	LANGUAGE	RELEASE_DATE	UNIT_PRICE	DISCREPTION	TITLE	GENRE
100021	ENGLISH	23-09-2020	120	THE_TRAGEDY_OF_DEATH	HORROR_HUNTING	MOVIE
200022	GERMAN	18-11-2014	180	THE_SECREAT_MYSTERY	BABLEON_SECREAT	BOOK
300023	HINDI	22-07-2021	350	LOVE_SONGS	LOVE_OF_TOMORROW	MUSIC
400024	SPANISH	11-07-2015	200	THE_MAN_WHO_MAD	CRAZY_HUMAN	DRAMA
500025	ENGLISH	14-07-2008	750	CHILDHOOD_ENJOYMENT	POPCON_BABY	MOVIE

→ EMPLOYEE

Input				Run SQL

```
INSERT INTO EMPLOYEE (EMPLOYEE_ID, FIRST_NAME, LAST_NAME, DEPARTMENT)
VALUES (205748, 'NITUJA', 'THAKUR', 'MOVIE_RECORDS'),
       (650123, 'AJINKYA', 'SUCHAK', 'BOOKS_SECTION'),
       (310221, 'JOHN', 'ADDAM', 'MUSIC_SECTION'),
       (420999, 'RAJESHWAR', 'ALLAN', 'BOOKS_SECTION'),
       (420217, 'RICKEY', 'BELLY', 'MUSIC_SECTION');
```

Output

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT
205748	NITUJA	THAKUR	MOVIE_RECORDS
310221	JOHN	ADDAM	MUSIC_SECTION
420217	RICKEY	BELLY	MUSIC_SECTION
420999	RAJESHWAR	ALLAN	BOOKS_SECTION
650123	AJINKYA	SUCHAK	BOOKS_SECTION

→ DELIVERY CONTROL

Input				Run SQL

```
INSERT INTO DELIVERY_CONTROL (PRODUCTS_IDS, EMPLOYEE_IDS, DELIVERY_STATUS)
VALUES (100021, 205748, 'CURRENTLY_EXECUTING'),
       (200022, 650123, 'DELIVERED'),
       (300023, 310221, 'CURRENTLY_EXECUTING'),
       (400024, 420999, 'YET_TO_START'),
       (500025, 420217, 'DELIVERED');
```

DELIVERY_CONTROL

PRODUCTS_IDS	EMPLOYEE_IDS	DELIVERY_STATUS
100021	205748	CURRENTLY_EXECUTING
200022	650123	DELIVERED
300023	310221	CURRENTLY_EXECUTING
400024	420999	YET_TO_START
500025	420217	DELIVERED

→ ORDERS

Input Run SQL

```
INSERT INTO ORDERS (CUSTOMER_IDS, PRODUCT_NOS, MODE_OF_PAYMENT, QUANTITY)
VALUES (100211, 100021, 'CREDIT_CARD', 450),
       (200212, 200022, 'CASH',500),
       (300213, 300023, 'BANK_TRANSFER',100),
       (400214, 400024, 'WALLET', 350),
       (600215, 500025, 'UPI_PAYMENT',220);
```

ORDERS

CUSTOMER_IDS	PRODUCT_NOS	MODE_OF_PAYMENT	QUANTITY
100211	100021	CREDIT_CARD	450
200212	200022	CASH	500
300213	300023	BANK_TRANSFER	100
400214	400024	WALLET	350
600215	500025	UPI_PAYMENT	220

4. Provide SQL statements for the following queries: a. Extract all the customers from a specific city.

> ## Extract all the customers from a specific city.

→ SELECT * FROM CUSTOMER WHERE CITY='PUNE'

Input Run SQL

```
SELECT * FROM CUSTOMER WHERE CITY='PUNE'
```

Output

CUSTOMER_ID	FIRST_NAME	LAST_NAME	EMAIL_ID	CITY	COUNTRY	MODE_OF_PAYMENT	PHONE_NUMBER
100211	NITIN	THAKKAR	Nitin01@Gisma.com	PUNE	INDIA	CASH	499766117924
600216	MASHESH	DESAI	Desai01@Gisma.com	PUNE	INDIA	CREDIT_CARD	499766652324

➤ <u>Search for a product of a specific genre</u>

⟶ <mark>**SELECT * FROM PRODUCT WHERE GENRE='MOVIE'**</mark>

Input   ⟨⟩   ☾   ⋮   **Run SQL**

SELECT * FROM PRODUCT WHERE GENRE='MOVIE'

Output

PRODUCT_ID	LANGUAGE	RELEASE_DATE	UNIT_PRICE	DISCREPTION	TITLE	GENRE
100021	ENGLISH	23-09-2020	120	THE_TRAGEDY_OF_DEATH	HORROR_HUNTING	MOVIE
500025	ENGLISH	14-07-2008	750	CHILDHOOD_ENJOYMENT	POPCON_BABY	MOVIE

➤ <u>Count how many customers are from a specific city.</u>

<mark>**SELECT COUNT (CITY) FROM CUSTOMER WHERE CITY='PUNE'**</mark>

Input   ⟨⟩   ☾   ⋮   **Run SQL**

SELECT COUNT (CITY) FROM CUSTOMER WHERE CITY='PUNE'

Output

COUNT (CITY)
2

➢ **Calculate the average of the unit price.**

⟶ SELECT AVG (UNIT_PRICE) FROM PRODUCT

Input					Run SQL

```
SELECT AVG(UNIT_PRICE)
FROM PRODUCT
```

Output

AVG(UNIT_PRICE)
320

Output

PRODUCT_ID	LANGUAGE	RELEASE_DATE	UNIT_PRICE	DISCREPTION	TITLE	GENRE
100021	ENGLISH	23-09-2020	120	THE_TRAGEDY_OF_DEATH	HORROR_HUNTING	MOVIE
200022	GERMAN	18-11-2014	180	THE_SECREAT_MYSTERY	BABLEON_SECREAT	BOOK
300023	HINDI	22-07-2021	350	LOVE_SONGS	LOVE_OF_TOMORROW	MUSIC
400024	SPANISH	11-07-2015	200	THE_MAN_WHO_MAD	CRAZY_HUMAN	DRAMA
500025	ENGLISH	14-07-2008	750	CHILDHOOD_ENJOYMENT	POPCON_BABY	MOVIE

➢ Extract all current orders

SELECT * FROM DELIVERY_CONTROL WHERE DELIVERY_STATUS='CURRENTLY_EXECUTING'

Input   [ ]   ☾   ⋮   **Run SQL**

```
SELECT * FROM DELIVERY_CONTROL WHERE DELIVERY_STATUS='CURRENTLY_EXECUTING'
```

Output

PRODUCTS_IDS	EMPLOYEE_IDS	DELIVERY_STATUS
100021	205748	CURRENTLY_EXECUTING
300023	310221	CURRENTLY_EXECUTING

➢ Extract all orders for books that has the keyword "the" in their description

SELECT * FROM PRODUCT WHERE DISCREPTION LIKE 'THE%';

Input   [ ]   ☾   ⋮   **Run SQL**

```
SELECT * FROM PRODUCT WHERE DISCREPTION LIKE 'THE%';
```

Output

PRODUCT_ID	LANGUAGE	RELEASE_DATE	UNIT_PRICE	DISCREPTION	TITLE	GENRE
100021	ENGLISH	23-09-2020	120	THE_TRAGEDY_OF_DEATH	HORROR_HUNTING	MOVIE
200022	GERMAN	18-11-2014	180	THE_SECREAT_MYSTERY	BABLEON_SECREAT	BOOK
400024	SPANISH	11-07-2015	200	THE_MAN_WHO_MAD	CRAZY_HUMAN	DRAMA

➢ **Extract all payments with credit cards for music records.**

SELECT ORDERS.MODE_OF_PAYMENT='CREDIT_CARD', PRODUCT.GENRE='MUSIC' FROM ORDERS, PRODUCT WHERE ORDERS.PRODUCT_NOS=PRODUCT.PRODUCT_ID;

Input				R

SELECT ORDERS.MODE_OF_PAYMENT='CREDIT_CARD', PRODUCT.GENRE='MUSIC' FROM ORDERS, PRODUCT WHERE ORDERS.PRODUCT_NOS=PRODUCT.PRODUCT_ID;

Output

ORDERS.MODE_OF_PAYMENT='CREDIT_CARD'	PRODUCT.GENRE='MUSIC'
1	0
0	0
0	1
0	0
0	0

➢ Count how many employees handle music records.

➡ SELECT COUNT (DEPARTMENT) FROM EMPLOYEE WHERE DEPARTMENT='MUSIC_SECTION'

Input					Run SQL

```
SELECT COUNT (DEPARTMENT) FROM EMPLOYEE WHERE DEPARTMENT='MUSIC_SECTION'
```

Output

COUNT (DEPARTMENT)
2

➡ SELECT * FROM EMPLOYEE WHERE DEPARTMENT='MUSIC_SECTION'

Input					Run SQL

```
SELECT * FROM EMPLOYEE WHERE DEPARTMENT='MUSIC_SECTION'
```

Output

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT
310221	JOHN	ADDAM	MUSIC_SECTION
420217	RICKEY	BELLY	MUSIC_SECTION

➢ Count how many employees first name is John.

→ <mark>SELECT COUNT (FIRST_NAME) FROM EMPLOYEE WHERE FIRST_NAME='JOHN'</mark>

Input				Run SQL

```
SELECT COUNT (FIRST_NAME) FROM EMPLOYEE WHERE FIRST_NAME='JOHN'
```

Output

COUNT (FIRST_NAME)
1

→ <mark>SELECT * FROM EMPLOYEE WHERE FIRST_NAME='JOHN'</mark>

Input				Run SQL

```
SELECT * FROM EMPLOYEE WHERE FIRST_NAME='JOHN'
```

Output

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	DEPARTMENT
310221	JOHN	ADDAM	HUMAN_RESOURCE

➢ <u>**Count how many orders are in the system.**</u>

SELECT COUNT (QUANTITY) FROM ORDERS

Input				Run SQL

```
SELECT COUNT(QUANTITY) FROM ORDERS
```

Output

COUNT(QUANTITY)
5

SELECT SUM (QUANTITY) FROM ORDERS

Input				Run SQL

```
SELECT SUM(QUANTITY) FROM ORDERS
```

Output

SUM(QUANTITY)
1620

5. Briefly present your use case. – (My Assumptions)

- In the Customer Table "**CUSTOMER_ID**" is Assumed as primary Key

- In the Employee Table "**EMPLOYEE_ID**" is Assumed as primary Key

- In the Product Table "**PRODUCT_ID**" is Assumed as primary Key

- Based on Availability of the Question I have assumed cardinality between "Customer - & - Product" / "Product - & - Employee" as **M:N**

- As Cardinality is **M:N** the new Composite Table was created *Delivery Control Table* and *Order Table*

- To solve the **Question No 4. (e)** – *Extract all current order*. The new attributes were assumed with respect to this question (Delivery Status as –> Currently executing; Delivered; Yet to start) in the Delivery Control table

- In Delivery Control Table which is as Composite table the (*PRODUCT_IDS, EMPLOYEE_IDS*) is considered as Composite Key or Foreign Key in correspondent to Product as – product_id **and** Employee as – employee_id.

- In Order Table which is as Composite table the (*CUSTOMER_IDS, PRODUCT_NOS*) is considered as Composite Key or Foreign Key in correspondent to Customer as – customer_id **and** Product as – product_id.

- To solve the **Question No 4. (j)** – *Count how many order are in system.* The new attributes were assumed with respect to this question (*Quantity* and Mode of Payment – Credit card, Cash, wallet etc.) in the Order table

6. Discuss the data governance plan used in the use case

Data governance is a process for regulating the availability, usefulness, authenticity, and security of the information stored in corporate systems. It is governed by internal information policies and guidelines that also control data consumption.

The three core categories of data owners, data administrators, and data guardians often make up the advisory board of a solid data governance program. Data custodians are accountable for the safe custody, handling, and storage of data as well as the implementation of business standards. Data stewards are in responsibility of what is retained in a data field, whereas data guardians/custody are in charge of the technological environment and database structure. The titles of database administrator, data designer, and ETL programmer are typically held by data custodians.

When Data is small in size and non-critical in nature the manual data storage may be reliable but as on when the data reliability increases the cloud form data store is most considered. Data collection, storage, and use within the cloud are all streamlined by

a set of policies, regulations, and procedures known as cloud data governance. By democratizing data, this framework upholds compliance. Even as your data landscape becomes more extensive and complex, it facilitates collaboration.

Data Access Governance is a data protection technology that enables businesses to enforce policies governing access to sensitive unstructured data that is present throughout the institution. Numerous channels are available for users to access data. It can be accessed directly from a database or data warehouse, through the user interface of an application, or occasionally even while data is still in transit. To maintain and guarantee privacy, data access governance enables users to control, safeguard, and audit data use. There are two fundamental methods for accessing data that is at rest in a repository: sequential access and random access. Sequential access moves the various data on a disc using a seek operation until the desired data is located. Data is stored or retrieved using random access from any location on the disc.

Source: Stedman, Craig; Industry editor; Jack, Vaughan (2022), "*What is data governance and why does it matter*", 4th May, 2022, viewed on: 28th Dec, 2022. [URL: https://www.alation.com/blog/why-cloud-data-governance-is-critical

Under my Case study Data is computed using online SQL Coding platform, But Data is not stored anywhere, however for ER diagram Data is accessed under my profile https://lucid.app/documents#/documents?folder_id=home

➤ **References and Sources**

- ▪ For ER Diagram - https://lucid.app.com

- ▪ For SQL Coding - https://www.programiz.com/sql/online-compiler

- ▪ For Doubts or reference example - https://www.w3schools.com/sql/sql_operators.asp