CS586 Introduction to Databases
Fall 2021 Quarter

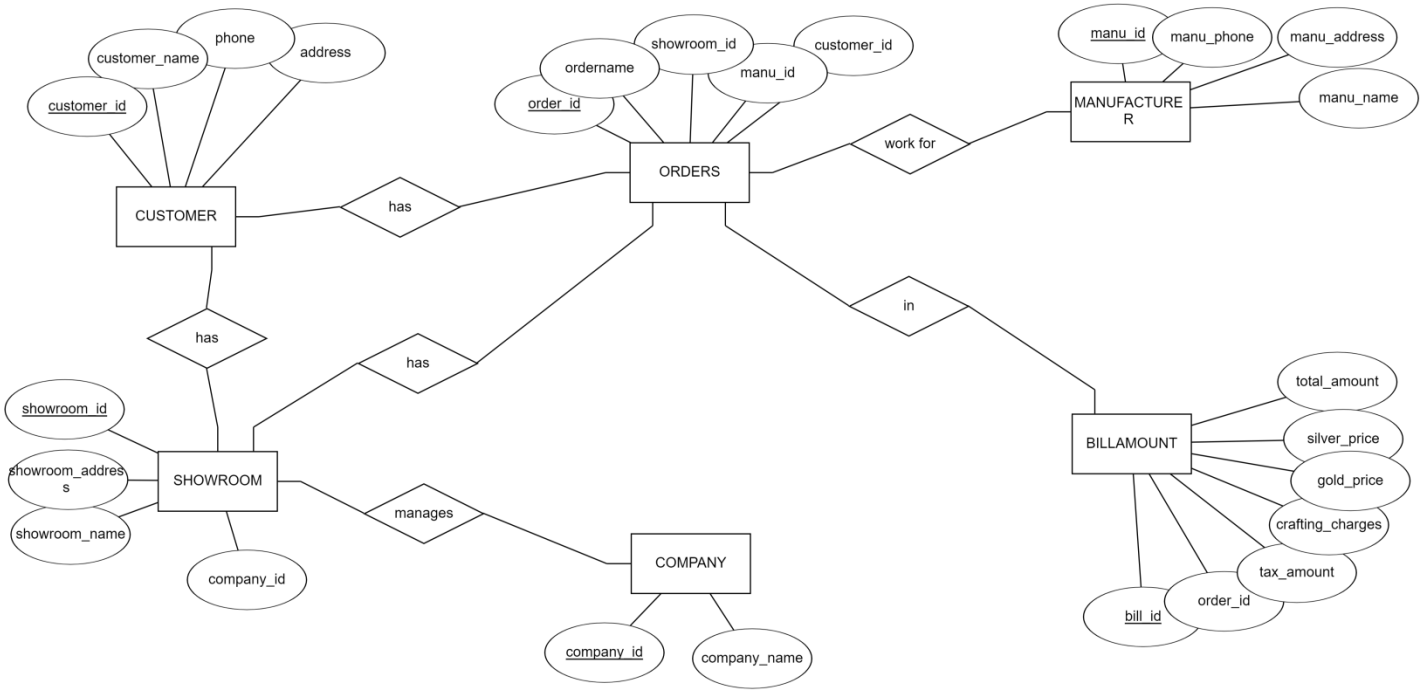
Graduate Project Database Implementation by:

Yerra Tarun Kumar
Harshvardan Rajiv Bindage.

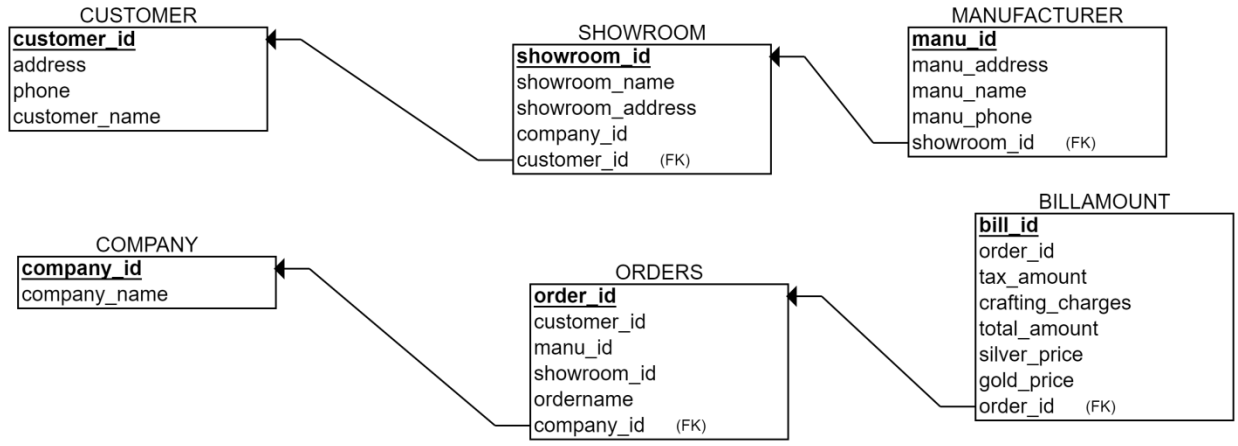
Project Overview:

- The purpose of this project is to give the Showroom a simple and user-friendly interface to coordinate all the details regarding their Customers and manufactures. In general, an company's development and practices are its intentional efforts to improve current and future performance by helping customers to place their orders easily.
- jewellery management system focuses on the basic entities of the manufacturing process. For example, the customers, manufactures, orders and showroom.
- Jewellery Management Database is a storage capacity which enable the Showroom to keep track of their Customers, Manufacturers and their respective Orders efficiently. This database provides common platform for every Showroom. This project serves as a handy tool to manage and organize the details of the Customers and their respective Orders in a systematic manner. It also helps the administrator/manager to have good control on them. The database can be referred for most of the queries regarding the same, which eliminates the use of paper to a large extent. The main advantage is the flexibility of addition of Customers, Bill Generation and Display orders undertaken by a showroom data sharing and also the permanent storage of data which can be retrieved and referred anytime.
- The Showroom of the database has a physical copy of all the details related to the Customers, Manufacturers and their respective Orders. But if this digitalized method is used, work is made easy and efficient along with the advantage of saving one's time.
- This can be applied to teachers' training, IT industry training, students' training or any sort of training in the real world as it is a generalized approach to cater the necessities of data storage, retrieval and modification operations.

ER DIAGRAM



SCHEMA DIAGRAM



TABLES & INFORMATION ABOUT ALL THE KEYS INCLUDING PRIMARY KEYS AND FOREIGN KEYS.

Customer it consists of four attributes.

(Customer_id,Customer_name,Phone,Address)

Customer_id is the foreign key to Orders table.

ORDERS it has five attributes.

(Order_id,Order_name,Showroom_id,Manu_id,Customer_id)

MANUFACTURE it has four attributes.

(Manu_id,Manu_phone,Manu_address,Manu_name)

SHOWROOM it has four attributes

(Showroom_id,Showroom_address,Showroom_name,Company_id)

Showroom_id is the foreign key to Orders table.

BILLAMOUNT it has seven attributes

(Bill_id,Order_id,Tax_amount,Crafting_charges,Gold_price,Silver_price,Total_amount)

Bill_id is the foreign key to Order table.

COMPANY it has two attributes

(Company_id,Company_name)

Company_id is the foreign key to Showroom table

CREATING TABLE QUERIES WITH ONE ROW OF THE SAMPLE DATA. (This is an Editable copy which was made in Visual Studio Code)

```
CREATE TABLE customer (
    customer_id int NOT NULL,
    customer_name varchar(255) NOT NULL,
    address varchar(255),
    phone CHAR(10),
    PRIMARY KEY (customer_id));

INSERT INTO customer (customer_id, customer_name, address, phone)
VALUES ('1', 'Erichsen', 'Portland', '7896541254');

=============================================================================================
CREATE TABLE showroom (
    showroom_id int NOT NULL,
    showroom_name varchar(255) NOT NULL,
    address varchar(255),
    PRIMARY KEY (showroom_id));


INSERT INTO showroom (showroom_id, showroom_name, address)
VALUES ('1', 'GRTGOLD', 'INDIA');
=============================================================================================

CREATE TABLE manufacturer (
    manu_id int NOT NULL,
    manu_name varchar(255) NOT NULL,
    manu_address varchar(255),
    manu_phone CHAR(10),
    PRIMARY KEY (manu_id));

INSERT INTO manufacturer (manu_id, manu_name, manu_address, manu_phone)
VALUES ('1', 'AA_FACTORIES', 'INDIA', 9874568485);
=======================================

CREATE TABLE orders (
    order_id int NOT NULL,
    customer_id int NOT NULL,
    manu_id int NOT NULL,
    showroom_id int NOT NULL,
    order_name varchar(255),
    company_id int,
    PRIMARY KEY (order_id),
    FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
    FOREIGN KEY (manu_id) REFERENCES manufacturer(manu_id),
    FOREIGN KEY (showroom_id) REFERENCES showroom(showroom_id));

INSERT INTO orders (order_id, customer_id, manu_id, showroom_id, order_name)
VALUES (1, 1, 1, 1, "Gold_Chain");

=======================================
```

```
CREATE TABLE billamount (
    bill_id int NOT NULL,
    order_id int NOT NULL,
    tax_amount int,
    crafting_charges int,
    gold_price int,
    silver_price int,
    total_bill int,
    PRIMARY KEY (bill_id),
    FOREIGN KEY (order_id) REFERENCES orders(order_id));

==

INSERT INTO billamount (bill_id, order_id, tax_amount, crafting_charges, gold_price, silver_price,
total_bill)
VALUES (1, 1, 250, 150, 500, 400, 1300);
```

A brief description of how you populated the database

- **I have populated the data on my own depending on the queries which ever I need.**
- **I have designed the database in such a realistic way that resembles the real scenario which is presently being used by the big merchanants and gold smith business.**
- **I have randomly generated the customer names, address, showroom names, manufacturer names using the https://www.mockaroo.com/ where it has a feature of generating random set of values in bulk which is fake and irrelevant to any means of data.**
- **Mockaroo has a great data mocking libraries available for almost every language and platform. But not everyone is a programmer or has time to learn a new framework. Mockaroo allows you to quickly and easily to download large amounts of randomly generated test data based on your own specs which you can then load directly into your test environment using SQL or CSV formats. No programming is required.**
- **Using the above tool I have generated the data for six tables i.e Customer, Orders,Manufacturers, Showroom, Company, Billamount.**

Questions.

1. List all the customers, showrooms,manufacturers in the database. (To avoid the huge data im showing the sample of 10 rows of data)

   Ans)

   SELECT * FROM CUSTOMER LIMIT 10;
   SELECT * FROM SHOWROOM LIMIT 10;
   SELECT * FROM MANUFACTURER LIMIT 10;

   Ans)

```
fall2021db44=> select * from customer limit 10;
 customer_id |  customer_name   |        address         |   phone
-------------+------------------+------------------------+------------
           2 | Cherilynn Stempe | 74 Acker Crossing      | 4377091858
           3 | Frannie Pevsner  | 1 Claremont Court      | 3263889003
           4 | Nikola Simek     | 61 Mallory Plaza       | 9687672840
           5 | Jobina Gouda     | 69 Russell Court       | 8178649261
           6 | Aldin Haw        | 3789 Lakewood Junction | 5395232415
           7 | Janet Bullion    | 0120 Veith Parkway     | 2322932450
           8 | Debora Bulley    | 18812 Twin Pines Plaza | 3881989676
           9 | Ailee Pretswell  | 12013 Fair Oaks Avenue | 1901682500
          10 | Myriam Pischoff  | 15605 Logan Circle     | 3329741442
          11 | Stillman Perell  | 37 Fuller Park         | 8717409572
(10 rows)
```

```
fall2021db44=> select * from showroom limit 10;
 showroom_id |        showroom_name         |   showroom_address    | company_id
-------------+------------------------------+-----------------------+------------
        3001 | Ziemann-Conroy               | 469 Vera Trail        |          1
        3002 | Robel, Windler and Osinski   | 1449 Jay Center       |          1
        3003 | Krajcik LLC                  | 94582 Vernon Crossing |          1
        3004 | Wehner Inc                   | 4 Kenwood Way         |          1
        3005 | Reilly, Swaniawski and Feest | 071 Judy Crossing     |          1
        3006 | Erdman-Stoltenberg           | 84194 Del Mar Center  |          1
        3007 | Walsh, Price and Volkman     | 016 Monument Court    |          1
        3008 | Lang and Sons                | 1329 Mockingbird Way  |          1
        3009 | Hegmann-Williamson           | 1 Prairie Rose Point  |          1
        3010 | Schaefer LLC                 | 8 Michigan Trail      |          1
(10 rows)
```

```
fall2021db44=> select * from manufacturer limit 10;
 manu_id |           manu_name            |      manu_address       | manu_phone
---------+--------------------------------+-------------------------+------------
    2001 | Gislason Inc                   | 967 Northfield Center   | 9208945632
    2002 | Weissnat Group                 | 30 Calypso Plaza        | 3474273387
    2003 | Hessel Group                   | 43085 Maywood Point     | 4591120978
    2004 | Hyatt-Rempel                   | 51610 Annamark Terrace  | 7106088931
    2005 | Emmerich, Grimes and Langworth | 17 Stoughton Alley      | 2587501075
    2006 | Schmitt-Nitzsche               | 22173 Comanche Junction | 3658262730
    2007 | Schulist Group                 | 73188 Westerfield Trail | 1932037506
    2008 | Wilkinson LLC                  | 3 Becker Avenue         | 6882513222
    2009 | Medhurst, Volkman and Mohr     | 83844 Elka Center       | 9361965584
    2010 | Nitzsche-Weber                 | 4491 Jackson Road       | 3934552034
(10 rows)
```

2. List the name of customers who has brought the ornaments greater than 5000 bill amount.
   Ans) **select customer.customer_name from customer inner join orders on orders.customer_id = customer.customer_id inner join billamount on orders.order_id = billamount.order_id where billamount.total_bill > 5000;**
   Rows Returned: 46

```
fall2021db44=> select customer.customer_name from customer inner join orders on orders.customer_id = customer.customer_id inner join billamount on orders.order_id = billamount.order_id wher
e billamount.total_bill > 5000;
    customer_name
--------------------
 Ariela Cerman
 Renard Sneesbie
 Keir Gallego
 Lazare Burgiss
 Lenora Shutler
 Gordon O'Hallagan
 Xylina Manwell
 Caye Clericoates
 Nowell Enocksson
```

3. List the order, customer name who has the highest price of the order.
   select billamount.total_bill, customer.customer_name,orders.order_name from customer inner join orders on orders.customer_id = customer.customer_id inner join billamount on orders.order_id = billamount.order_id where billamount.total_bill = (select max(billamount.total_bill) from billamount);

```
fall2021db44=> select billamount.total_bill, customer.customer_name,orders.order
_name from customer inner join orders on orders.customer_id = customer.customer_
id inner join billamount on orders.order_id = billamount.order_id where billamou
nt.total_bill = (select max(billamount.total_bill) from billamount);
 total_bill |  customer_name  | order_name
------------+-----------------+------------
       5500 | Kaylyn Milleton | Bangle
(1 row)
```

4. List all the customers who brought the ornaments at the highest gold price.
   select billamount.gold_price, customer.customer_name,orders.order_name from customer inner join orders on orders.customer_id = customer.customer_id inner join billamount on orders.order_id = billamount.order_id where billamount.gold_price = (select max(billamount.gold_price) from billamount);

```
fall2021db44=> select billamount.gold_price, customer.customer_name,orders.order_name from customer inner join orders on orders.customer_id = customer.customer_id inner join billamount on o
rders.order_id = billamount.order_id where billamount.gold_price = (select max(billamount.gold_price) from billamount);
 gold_price |  customer_name  | order_name
------------+-----------------+------------
       1502 | Kaylyn Milleton | Bangle
(1 row)
```

5. List all the jewllery that has the highest making charges

select orders.order_name, billamount.crafting_charges from billamount inner join orders on orders.order_id = billamount.order_id where gold_price = (select max(billamount.gold_price) from billamount);

```
fall2021db44=> select orders.order_name, billamount.crafting_charges from billamount inner join orders on orders.order_id = billamount.order_id where gold_price = (select max(billamount.gol
d_price) from billamount);
 order_name | crafting_charges
------------+------------------
 Bangle     |             1001
(1 row)
```

6. List all the customers who brought the ornaments at the highest silver price.

Ans).

select customer.customer_name, orders.order_name from customer inner join orders on orders.customer_id = customer.customer_id inner join billamount on orders.order_id = billamount.order_id where silver_price = (select max(billamount.silver_price) from billamount);

```
fall2021db44=> select customer.customer_name, orders.order_name from customer inner join orders on orders.customer_id = customer.customer_id inner join billamount on orders.order_id = billamount.or
der_id where silver_price = (select max(billamount.silver_price) from billamount);
  customer_name  | order_name
-----------------+------------
 Kaylyn Milleton | Bangle
(1 row)
```

7. List the showrooms which are located at specific location.

Ans).

select showroom_name, showroom_address from showroom where showroom_address = '3684 Melody Park';

```
fall2021db44=> select showroom_name, showroom_address from showroom where showroom_address = '3684 Melody Park';
       showroom_name        | showroom_address
----------------------------+------------------
 Mayer, Wisoky and Toy      | 3684 Melody Park
 Lang and Sons              | 3684 Melody Park
 Schaefer LLC               | 3684 Melody Park
 Rowe-Crooks                | 3684 Melody Park
 Keebler and Sons           | 3684 Melody Park
 Von, Hoppe and Berge       | 3684 Melody Park
 Feil, Marquardt and Wisoky | 3684 Melody Park
 Lindgren-Mayer             | 3684 Melody Park
 Robel, Windler and Osinski | 3684 Melody Park
(9 rows)
```

8. List all the manufacture names who has the orders greater than 2000.

Ans).

select manufacturer.manu_name from manufacturer inner join orders on orders.manu_id = manufacturer.manu_id inner join billamount on billamount.order_id = orders.order_id where billamount.total_bill > 2000;

```
              manu_name
-------------------------------------
 Emmerich, Kuhlman and Hirthe
 Buckridge, Schmitt and Tremblay
 Kessler-Boehm
 Mosciski-Bechtelar
 Spinka-Koch
 Kohler-Morar
 Rice-Doyle
 Flatley, Krajcik and Kihn
 Becker-Oberbrunner
 Swift Group
 Jacobson, Lesch and Schinner
 Emmerich Group
 Marquardt and Sons
 Erdman and Sons
```

9. List all the customers who has made the orders with the bill amount at the lowest price.
   Ans).

```
select customer.customer_name, orders.order_name,billamount.total_bill from customer inner join orders on orders.customer_id = customer.customer_id inner join billamount on orders.order_id = billamount.order_id where total_bill = (select min(billamount.total_bill) from billamount);
```

```
fall2021db44=> select customer.customer_name, orders.order_name,billamount.total_bill from customer inner join orders on orders.customer_id = customer.customer_id inner join billamount on orders.order_id = billamount.order_id where total_bill = (select min(billamount.total_bill) from billamount);
 customer_name | order_name | total_bill
---------------+------------+------------
 Erichsen      | Chain      |         11
(1 row)
```

10. List all the showroom names who have the orders with the name ‘Bangles’.
    Ans)

```
select showroom.showroom_name, orders.order_name from showroom inner join orders on orders.showroom_id = showroom.showroom_id where orders.order_name = 'Bangle’;
```

```
fall2021db44=> select showroom.showroom_name, orders.order_name from showroom inner join orders on orders.showroom_id = showroom.showroom_id where orders.order_name = 'Bangle'
;
         showroom_name          | order_name
--------------------------------+------------
 Walsh, Price and Volkman       | Bangle
 Bashirian, Herzog and Mohr     | Bangle
 Mills, Wunsch and Hansen       | Bangle
 Schmidt, McClure and Turcotte  | Bangle
 Toy-Medhurst                   | Bangle
 Schneider-Lindgren             | Bangle
 Bauch LLC                      | Bangle
 Rolfson, Hammes and Wyman      | Bangle
 Kutch Group                    | Bangle
 Champlin, Ruecker and Leannon  | Bangle
 Kling Inc                      | Bangle
 Davis, Schmeler and Larson     | Bangle
 Maggio, Schaefer and Dooley    | Bangle
 Howe, Fay and Kshlerin         | Bangle
 Cruickshank, Douglas and Lockman | Bangle
 Goyette, Bernier and McLaughlin | Bangle
 Kilback, Reinger and Lueilwitz | Bangle
 Farrell, Kassulke and Renner   | Bangle
 Ortiz-Wisozk                   | Bangle
 Hahn, Schultz and Kozey        | Bangle
 Hoeger LLC                     | Bangle
 Jaskolski-Rutherford           | Bangle
 Auer Inc                       | Bangle
 Koss, Morissette and Wunsch    | Bangle
 Roberts and Sons               | Bangle
 Huel, Aufderhar and Moen       | Bangle
(26 rows)

fall2021db44=>
```

11. List the orders and customer names who has the making charge greater than 500
    Select
    Ans)
    select customer.customer_name, orders.order_name from customer inner join orders on orders.customer_id = customer.customer_id inner join billamount on billamount.order_id = orders.order_id where crafting_charges = (select billamount.crafting_charges where billamount.crafting_charges > 500);

customer_name	order_name	crafting_charges
Anneliese Thornham	Little Millet	501
Barton Hairsnape	Bangle	503
Reidar Rolls	Xanthoparmelia Lichen	505
Frannie Bauman	Yerba Santa	507
Alika Passler	Barnacle Lichen	509
Janella Varley	Beet	511
Pamelina Gerrell	Snake River Twinpod	513
Rooney Padillo	San Diego Raspberry	515
Jefferson Astbury	Wax Mallow	517
Micheal Leabeater	earring	519
Jamison Highnam	Coralbush	521
Morse Gazey	Tepic Flameflower	523

12. List all the customers who made the order named ‘Chains’.
    Ans)
    select customer.customer_name from customer inner join orders on orders.customer_id = customer.customer_id inner join billamount on billamount.order_id = orders.order_id where orders.order_name = 'Chain';

```
fall2021db44=> select customer.customer_name from customer inner join orders on orders.customer_id = customer.customer_id inner join billamount on billamount.order_id = orders.order_id where orders
.order_name = 'Chain';
   customer_name
--------------------
 Aldin Haw
 Stillman Perell
 Shell Haining
 Alisander Bollen
 Samson Piell
 Serene Hymus
 Hatty Ivers
 Welby Ostridge
 Frederik Spillane
 Chickie Scase
 Bidget Klainman
 Randene Hurtado
 Erichsen
(13 rows)
```

13. List the minimum of total bill.
    Ans.) **select min(billamount.total_bill) from billamount;**

```
fall2021db44=> select min(billamount.total_bill) from billamount;
 min
-----
  11
(1 row)
```

14. List the average of the total bill
    Ans.)
    select avg(billamount.total_bill) from billamount;

```
fall2021db44=> select avg(billamount.total_bill) from billamount;
          avg
-----------------------
 2755.5000000000000000
(1 row)
```

15. List the max of the total bill
    Ans.)

    select max(billamount.total_bill) from billamount;

```
fall2021db44=> select max(billamount.total_bill) from billamount;
 max
------
 5500
(1 row)
```

16. List all the orders, bill amounts, manufacturers, showrooms and customers whose order names are earring
    Ans.)
    select customer.customer_name,orders.order_name as order_type,manufacturer.manu_name as manufacturer_name, showroom.showroom_name, billamount.total_bill from customer inner join orders on orders.customer_id = customer.customer_id inner join manufacturer on manufacturer.manu_id = orders.manu_id inner join showroom on showroom.showroom_id = orders.showroom_id inner join billamount on billamount.order_id = orders.order_id where order_name = (select orders.order_name from orders where orders.order_name = 'earring' limit 1);

```
fall2021db44=> select customer.customer_name,orders.order_name as order_type,manufacturer.manu_name as manufacturer_name, showroom.showroom_name, billamount.total_bill from customer inner join orders on orders.customer_id = customer.customer_id inner join manufacturer on manufacturer.manu_id = orders.manu_id inner join showroom on showroom.showroom_id = orders.showroom_id inner join billamount on billamount.order_id = orders.order_id where order_name = (select orders.order_name from orders where orders.order_name = 'earring' limit 1);
    customer_name    | order_type |       manufacturer_name        |          showroom_name          | total_bill
---------------------+------------+--------------------------------+---------------------------------+------------
 Kurtis Fleckness    | earring    | Little, Grimes and Gusikowski  | Osinski-Nitzsche                |        132
 Robers Maybury      | earring    | Heidenreich-Kuphal             | Jenkins-Berge                   |        990
 Rudd Jakoubek       | earring    | Zulauf, Pacocha and Torp       | Hintz, Kihn and Dooley          |       1122
 Trace Flannigan     | earring    | Stehr LLC                      | Carroll-Stamm                   |       1353
 Roch Frowing        | earring    | Bechtelar-Kuhic                | Hermiston-Daugherty             |       1452
 Kaitlin Gillani     | earring    | Dicki, Leannon and Raynor      | Corkery, Langosh and Jaskolski  |       1551
 Papagena Gabbot     | earring    | Gerhold-Zemlak                 | Corkery, Stroman and Roob       |       1716
 Zilvia Willison     | earring    | Murray Inc                     | Koepp Group                     |       1848
 Tonya Whereat       | earring    | Spinka-Koch                    | Cruickshank-Davis               |       2046
 Georgianna Tringham | earring    | Carter-Russel                  | Kohler-Schowalter               |       2178
 Salomi Abbess       | earring    | Ratke, Pollich and Mueller     | Pagac-Douglas                   |       2310
 Leyla Voaden        | earring    | Keeling-Davis                  | Upton and Sons                  |       2442
 Rutter Oldroyd      | earring    | Hilpert Inc                    | Gleason, Jones and Macejkovic   |       2574
 Papageno Lamerton   | earring    | Crooks LLC                     | Cole, Herman and Koepp          |       2717
 Micheal Leabeater   | earring    | Becker Group                   | Schultz-Price                   |       2849
 Roseline Argile     | earring    | Stroman, Lind and Skiles       | White-Ortiz                     |       2981
 Mattie McLugish     | earring    | Powlowski Group                | Miller Group                    |       3245
 Netti Janczyk       | earring    | Fritsch, Kuhn and Wiza         | Rohan, Friesen and Dach         |       3333
 Paulie Arkin        | earring    | Jacobs LLC                     | Nader Inc                       |       3465
 Jephthah Erasmus    | earring    | Miller-Wiegand                 | Feil-Wolf                       |       3597
```

17. List the total count of customers and showrooms whose orders are ear rings.
    Ans.)
    SELECT COUNT(customer.customer_name) AS customerCount, COUNT(showroom.showroom_name) AS showroomCount FROM customer JOIN orders ON (orders.customer_id = customer.customer_id AND orders.order_name = 'earring') JOIN showroom ON (orders.showroom_id = showroom.showroom_id AND orders.order_name = 'earring');

```
ers.order_name = 'earring') JOIN sho
 customercount | showroomcount
---------------+---------------
            33 |            33
(1 row)
```

18. list the total number of showrooms that have orders with order name necklace
    Ans.)
    select count(showroom.showroom_name) as showroomCount from showroom inner join orders on orders.showroom_id = showroom.showroom_id where orders.order_name = 'necklace';

```
fall2021db44=> s
 showroomcount
---------------
            16
(1 row)
```

19. list all the orders whose customers live in a specific area address.
    Ans.)
    select orders.order_name from orders inner join customer on orders.customer_id = customer.customer_id where customer.address = '3789 Lakewood Junction';

```
fall2021db44=> select  orders.order_name from orders inner join customer on orders.customer_id = customer.customer_id where customer.address = '3789 Lakewood Junction';
 order_name
------------
 Chain
(1 row)
```

20. List all the making charges, gold prices, silver prices whose total is greater than 5000 .
    Ans.)
    select billamount.crafting_charges, billamount.gold_price, billamount.silver_price from billamount where billamount.total_bill > 5000;

```
fall2021db44=> select billamount.crafting_charges, billamount.gold_price, billamount.silver_price from billamount where billamount.total_bill > 5000;
 crafting_charges | gold_price | silver_price
------------------+------------+--------------
              911 |       1367 |         1817
              913 |       1370 |         1821
              915 |       1373 |         1825
              917 |       1376 |         1829
              919 |       1379 |         1833
              921 |       1382 |         1837
              923 |       1385 |         1841
              925 |       1388 |         1845
              927 |       1391 |         1849
              929 |       1394 |         1853
              931 |       1397 |         1857
```

A listing of 5 rows from each of your tables:

select * from customer limit 5;

select * from orders limit 5;

select * from manufacturer limit 5;

select * from billamount limit 5;

select * from company limit 5;

```
fall2021db44=> select * from customer limit 5;
 customer_id |  customer_name   |        address         |   phone
-------------+------------------+------------------------+------------
           2 | Cherilynn Stempe | 74 Acker Crossing      | 4377091858
           3 | Frannie Pevsner  | 1 Claremont Court      | 3263889003
           4 | Nikola Simek     | 61 Mallory Plaza       | 9687672840
           5 | Jobina Gouda     | 69 Russell Court       | 8178649261
           6 | Aldin Haw        | 3789 Lakewood Junction | 5395232415
(5 rows)

fall2021db44=> select * from orders limit 5;
 order_id | customer_id | manu_id | showroom_id |      order_name
----------+-------------+---------+-------------+----------------------
     1001 |           1 |    2001 |        3001 | Chain
     1002 |           2 |    2002 |        3002 | Notothylas
     1003 |           3 |    2003 |        3003 | Spreading Navarretia
     1004 |           4 |    2004 |        3004 | Pterostegia
     1005 |           5 |    2005 |        3005 | necklace
(5 rows)

fall2021db44=> select * from manufacturer limit 5;
 manu_id |           manu_name            |      manu_address      | manu_phone
---------+--------------------------------+------------------------+------------
    2001 | Gislason Inc                   | 967 Northfield Center  | 9208945632
    2002 | Weissnat Group                 | 30 Calypso Plaza       | 3474273387
    2003 | Hessel Group                   | 43085 Maywood Point    | 4591120978
    2004 | Hyatt-Rempel                   | 51610 Annamark Terrace | 7106088931
    2005 | Emmerich, Grimes and Langworth | 17 Stoughton Alley     | 2587501075
(5 rows)

fall2021db44=> select * from billamount limit 5;
 bill_id | order_id | tax_amount | crafting_charges | gold_price | silver_price | total_bill
---------+----------+------------+------------------+------------+--------------+------------
    4001 |     1001 |          2 |                3 |          5 |            1 |         11
    4002 |     1002 |          4 |                5 |          8 |            5 |         22
    4003 |     1003 |          6 |                7 |         11 |            9 |         33
    4004 |     1004 |          8 |                9 |         14 |           13 |         44
    4005 |     1005 |         10 |               11 |         17 |           17 |         55
(5 rows)

fall2021db44=> select * from company limit 5;
 company_id |  company_name
------------+----------------
          1 | SEATTLE JEWLS
          2 | PHOENIX JEWLS
          3 | NEWYORK JEWLS
          4 | PORTLAND JEWLS
          5 | FLORIDA JEWLS
(5 rows)

fall2021db44=> select * from showroom limit 5;
 showroom_id |        showroom_name         |   showroom_address    | company_id
-------------+------------------------------+-----------------------+------------
        3001 | Ziemann-Conroy               | 469 Vera Trail        |          1
        3003 | Krajcik LLC                  | 94582 Vernon Crossing |          1
        3004 | Wehner Inc                   | 4 Kenwood Way         |          1
        3005 | Reilly, Swaniawski and Feest | 071 Judy Crossing     |          1
        3006 | Erdman-Stoltenberg           | 84194 Del Mar Center  |          1
(5 rows)

fall2021db44=>
```

References :

https://www.kaggle.com/

https://www.mockaroo.com/mock_apis

https://www.mockaroo.com/