CPS510: Database Systems

Assignment 10

Section: 06

Group Members:

Thomas Young-Audet, 500865018
Ivan Golovine, 500813431
Amanda Nguyen, 500761893

Art Gallery Database

Index

1. Application Description of Art Gallery Database ---------- 3

2. Entity Relationship Model ---------- 4

3. Art Gallery Database Schema Design ---------- 5

4. Demo Views and Simple Queries ---------- 10

5. Query Demonstration of Advanced Queries by Unix Shell ---------- 29

6. Database Normalization/ Functional Dependencies ---------- 53

7. 3NF Normalization Diagrams ---------- 60

8. 3NF Normalization/ BCNF with Algorithm ---------- 69

9. Application of Database by Apex Oracle UI ---------- 86

--> 9.2 Proving the SQL Database is in BCNF ---------- 96

10. Relational Algebra Notation ---------- 103

11. Concluding Remarks ---------- 106

1. Application Description of Art Gallery Database

Description:

Modern art galleries provide more than just walls to hang up art. They help artists gain exposure, sell or loan art pieces to customers, and create expositions to highlight artists and styles. A database management system is essential to the operation of these busy and involved centers of arts and culture. The art gallery convention center database management system will be used by the staff of the art gallery to find and distribute artistic pieces. It will store information about the art, it's artist and in which exposition it is being displayed. The database system will also store information about pieces of art that are being rented or sold, and to whom.

Functions:

This database application should provide organizational information to the employees of the art gallery, as well as to the customers. With this system, users can find the art pieces displayed in the gallery as well as which exhibition it is displayed in, and how long they will be there. Users can perform advanced search by providing specific characteristics the art piece falls under. When the user searches by defining the medium, size, type and time period it brings up the art pieces that fulfil the criteria. A user can also place an order to rent or buy an art piece. Administrators are the only users capable of modifying any tables.Users can also add artworks or artists in the gallery.

Information expected:

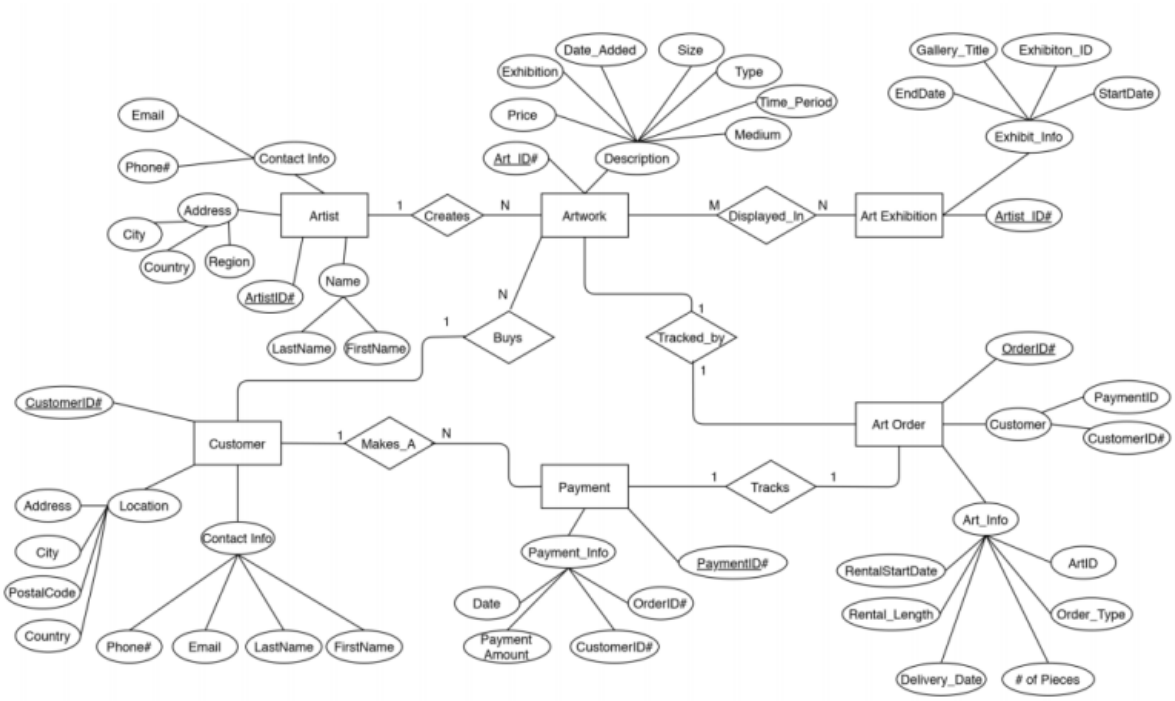
We expect our DBMS to have multiple tables such as Artist, Artwork, Art Exhibition, Customer, Rental, Payment and a Sold table. An artwork should have various attributes like an ID it’s title, medium, description of the piece, size, type, time period, and price and any additional information. Out of all the said attributes, the ID is the primary key because it is unique. A customer will have an entry in the Customer table with attributes such as ID and name, and will be able to make payments to buy or rent art pieces. All payments will be tracked in the Payment table with attributes such as a unique payment ID, price and date. Pieces of art that have been sold will be tracked in the Sold table, and those rented in the Rent table and will have the attributes of transaction ID, sale price, and date. Artists will have attributes such as Artist ID First Name, Last Name, Date of birth, works produced, Style, Address, Country and Phone number.

For example the Artwork table would have:

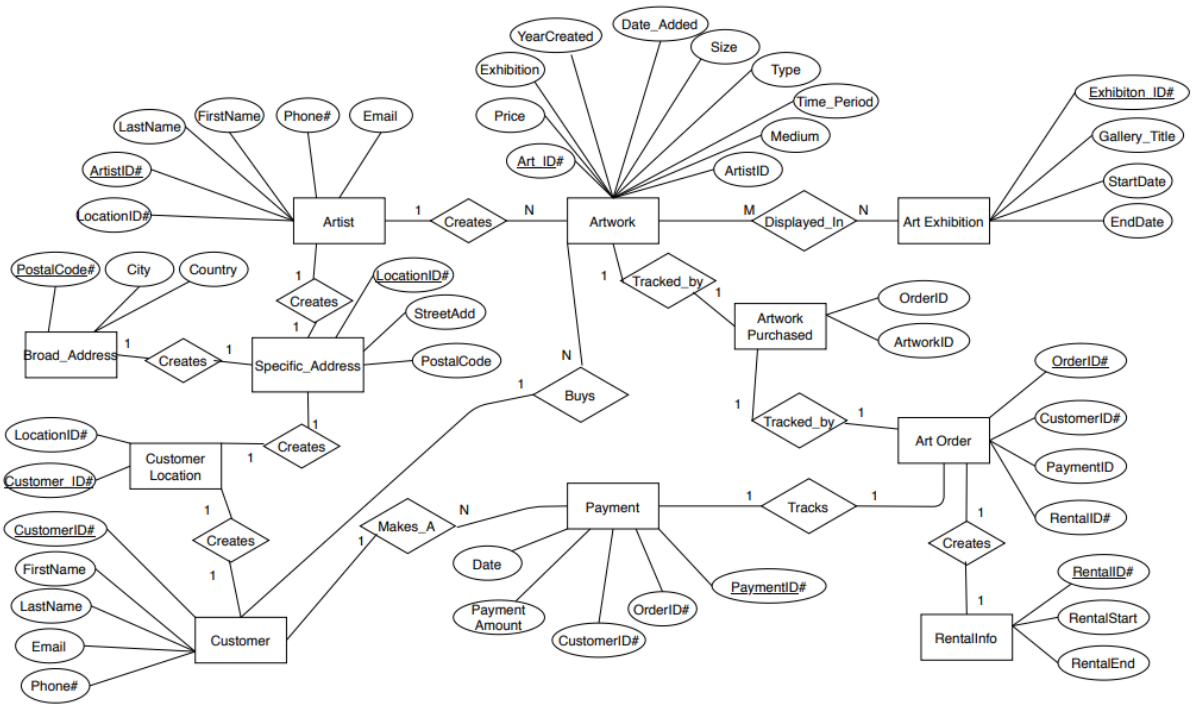
Artist_ID	Art_ID	Date added	Removal Date	Art Medium	Art Size	Art Type	Art Time Period	Price	Exhibition
John Smith	0001	Sept 13, 2020	Nov 20, 2020	Oil on canvas	33 x 40 inches	Painting	1900 - 1950	2000$	John Smith
Jane Doe	0002	Sept 13, 2020	Nov 20, 2020	Acrylic on canvas	50 x 60 inches	Painting	2000 - Present	4000$	Moder nism

2. Entity Relationship Model (Updated to reflect the database)

Old ER Diagram:



New ER Diagram to reflect database in BCNF:



3. Art Gallery Database Schema Design

```
CREATE TABLE artist (
    ArtistID NUMBER PRIMARY KEY,
    Firstname VARCHAR2(25) NOT NULL,
    Lastname VARCHAR2(25) NOT NULL,
    City VARCHAR2(20),
    Region VARCHAR2(20),
    Country VARCHAR2(30) NOT NULL,
    Email VARCHAR(40),
    PhoneNumber VARCHAR2(12))

SELECT * FROM artist ORDER BY firstname ASC;

INSERT INTO artist
VALUES (123, 'Dawn', 'Lol', 'Toronto', 'ON', 'CA', 'lol@gmail.com',
'6471234567');
INSERT INTO artist
VALUES (456, 'Smith', 'Jones', 'Scarborough', 'ON', 'CA', 'sje@gmail.com',
'4167898907');
INSERT INTO artist
VALUES (789, 'Lee', 'Li', 'Brampton', 'ON', 'CA', 'leelol@gmail.com',
'4136547890');
INSERT INTO artist
VALUES (367, 'Sally', 'Hong', 'Markham', 'ON', 'CA', 'bluesky@yahoo.ca',
'6475781265');
INSERT INTO artist
VALUES (148, 'James', 'Lone', 'Toronto', 'ON', 'CA', 'jleartist@gmail.com',
'4134563421');


CREATE TABLE artwork (
    ArtworkID NUMBER PRIMARY KEY,
    Artist VARCHAR(25),
    NameArt VARCHAR2(100),
    Exhibition VARCHAR2(100),
    DateAdded DATE NOT NULL,
    SizeArt VARCHAR2(20) NOT NULL,
    Price NUMBER,
    TypeArt VARCHAR(40),
    MediumArt VARCHAR(50),
    TimePeriod VARCHAR2(12),
    YearCreated NUMBER)

SELECT * FROM artwork ORDER BY Artist ASC;

INSERT INTO artwork
VALUES (4560,'Dawn Lol', 'uhm', 'Oldendel', '2019-12-17', 'Medium',
2345,'Painting', 'Oil on Canvas','1900s', 2016);
INSERT INTO artwork
VALUES (7891,'Smith Jones', 'Bleu', 'Tulipshaw', '2019-2-08', 'Large',
1356,'Painting', 'Acrylic on Canvas','Renaissance', 2019);
INSERT INTO artwork
```

```
VALUES (2345,'Lee Li', 'Oythus', 'Relicmilie', '2019-2-08', 'Large',
1098,'Painting', 'Watercolor on Hot Pressed Paper','Modern', 2003);
INSERT INTO artwork
VALUES (6778,'Sally Hong', 'Rackaro', 'Skipleaf', '2019-3-18', 'Large',
5000,'Statue', 'Copper','Modern', 2006);
INSERT INTO artwork
VALUES (8674,'James Lone', 'Lenveyas', 'Leafmill', '2019-3-18', 'Large',
5670,'Statue', 'Copper','Modern', 2019);

DELETE
FROM
    artwork
WHERE
    ArtworkID = 7890;

CREATE TABLE exhibition (
    ArtistID REFERENCES artist(ArtistID),
    ExhibitionID int,
    Gallery_Title VARCHAR(50),
    StartDate DATE,
    EndDate DATE)

SELECT * FROM exhibition;

INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from
artist where Firstname = 'Dawn')), 23206,'Oldendel', '2020-1-08','2020-1-
05');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from
artist where Firstname = 'Smith')), 90865,'Tulipshaw', '2019-2-08','2019-2-
15');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from
artist where Firstname = 'Lee')), 78964,'Relicmilie', '2019-2-08','2019-2-
20');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from
artist where Firstname = 'Sally')), 14532,'Skipleaf', '2019-3-18','2019-3-
25');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from
artist where Firstname = 'James')), 23256,'Leafmill', '2019-3-18','2019-3-
25');


CREATE TABLE customer (
    CustomerID NUMBER PRIMARY KEY,
    FirstName VARCHAR2(25) NOT NULL,
    LastName VARCHAR2(25) NOT NULL,
    Address VARCHAR(50),
    City VARCHAR2(20),
    PostalCode VARCHAR2(6),
    Country VARCHAR2(30),
    Email VARCHAR(40) NOT NULL,
    PhoneNumber VARCHAR2(12))
```

```sql
INSERT INTO customer
VALUES (00982,'Tom','Nook', '123 Sesame Street', 'New
York','09303','USA','randomemail@yahoo.com','212-234-3344');
INSERT INTO customer
VALUES (00983,'Timmy','Nook', '124 Sesame Street', 'New
York','09303','USA','randomemail1@yahoo.com','212-456-3646');
INSERT INTO customer
VALUES (00984,'Tommy','Nook', '125 Sesame Street', 'New
York','09303','USA','randomemail2@yahoo.com','212-789-3567');

SELECT * FROM customer ORDER BY FirstName ASC;

CREATE TABLE art_order (
    OrderID NUMBER PRIMARY KEY,
    CustomerID NUMBER REFERENCES customer(CustomerID),
    ArtID REFERENCES artwork(ArtworkID),
    RentalStartDate DATE NOT NULL,
    Rental_Length int,
    Delivery_Date DATE NOT NULL,
    NoPieces int,
    Order_Type VARCHAR(50))

ALTER TABLE art_order ADD (
PaymentID NUMBER REFERENCES payment(PaymentID));

INSERT INTO art_order
VALUES (100987,(select CustomerID from customer where CustomerID
=00982),(select ArtworkID from artwork where ArtworkID=4560),'2019-4-
25',3,'2019-4-30',1,'Rental',(select PaymentID from payment where PaymentID
=200987));
INSERT INTO art_order
VALUES (100988,(select CustomerID from customer where CustomerID
=00983),(select ArtworkID from artwork where ArtworkID=6778),'2019-4-
30',1,'2019-5-05',1,'Rental',(select PaymentID from payment where PaymentID
=200988));
INSERT INTO art_order
VALUES (100989,(select CustomerID from customer where CustomerID
=00984),(select ArtworkID from artwork where ArtworkID=8674),'2019-5-
05',1,'2019-05-10',1,'Rental',(select PaymentID from payment where PaymentID
=200989));

SELECT * FROM art_order;

CREATE TABLE payment (
    PaymentID NUMBER PRIMARY KEY,
    OrderID NUMBER REFERENCES art_order(OrderID),
    CustomerID NUMBER REFERENCES customer(CustomerID),
    PaymentAmount NUMBER,
    DatePayment DATE)

INSERT INTO payment
VALUES (200987,(select OrderID from art_order where OrderID=100987),(select
CustomerID from customer where CustomerID =00982),250,'2019-5-01');
INSERT INTO payment
VALUES (200988,(select OrderID from art_order where OrderID=100988),(select
CustomerID from customer where CustomerID =00983),450,'2019-4-29');
INSERT INTO payment
```

```
VALUES (200989,(select OrderID from art_order where OrderID=100989),(select
CustomerID from customer where CustomerID =00984),450,'2019-4-30');

SELECT * FROM payment;
```

Query 1 Result - artist entity:

```
VALUES (367, 'Sally', 'Hong', 'Markham', 'ON', 'CA', 'bluesky@yahoo.ca', '6475781265');
```

Query Result

All Rows Fetched: 5 in 0.026 seconds

	ARTISTID	FIRSTNAME	LASTNAME	CITY	REGION	COUNTRY	EMAIL	PHONENUMBER
1	123	Dawn	Lol	Toronto	ON	CA	lol@gmail.com	6471234567
2	148	James	Lone	Toronto	ON	CA	jleartist@gmail.com	4134563421
3	789	Lee	Li	Brampton	ON	CA	leelol@gmail.com	4136547890
4	367	Sally	Hong	Markham	ON	CA	bluesky@yahoo.ca	6475781265
5	456	Smith	Jones	Scarborough	ON	CA	sje@gmail.com	4167898907

Query 2 Result – artwork entity:

Query Result

All Rows Fetched: 5 in 0.014 seconds

	ARTWORKID	ARTIST	NAMEART	EXHIBITION	DATEADDED	SIZEART	PRICE	TYPEART	MEDIUMART	TIMEPERIOD	YEARCREATED
1	4560	Dawn Lol	uhm	Oldendel	19-12-17	Medium	2345	Painting	Oil on Canvas	1900s	2016
2	8674	James Lone	Lenveyas	Leafmill	19-03-18	Large	5670	Statue	Copper	Modern	2019
3	2345	Lee Li	Oythus	Relicmilie	19-02-08	Large	1098	Painting	Watercolor on Hot Pressed Paper	Modern	2003
4	6778	Sally Hong	Rackaro	Skipleaf	19-03-18	Large	5000	Statue	Copper	Modern	2006
5	7891	Smith Jones	Bleu	Tulipshaw	19-02-08	Large	1356	Painting	Acrylic on Canvas	Renaissance	2019

Query 3 Result – exhibition entity:

Query Result

All Rows Fetched: 10 in 0.02 seconds

	ARTISTID	EXHIBITIONID	GALLERY_TITLE	STARTDATE	ENDDATE
1	456	90865	Tulipshaw	19-02-08	19-02-15
2	789	78964	Relicmilie	19-02-08	19-02-20
3	367	14532	Skipleaf	19-03-18	19-03-25
4	148	23256	Leafmill	19-03-18	19-03-25
5	123	23206	Oldendel	20-01-08	20-01-05
6	456	90865	Tulipshaw	19-02-08	19-02-15
7	789	78964	Relicmilie	19-02-08	19-02-20
8	367	14532	Skipleaf	19-03-18	19-03-25
9	148	23256	Leafmill	19-03-18	19-03-25
10	123	23206	Oldendel	20-01-08	20-01-05

Query 4 Result – customer entity:

Query Result

All Rows Fetched: 3 in 0.022 seconds

	CUSTOMERID	FIRSTNAME	LASTNAME	ADDRESS	CITY	POSTALCODE	COUNTRY	EMAIL	PHONENUMBER
1	983	Timmy	Nook	124 Sesame Street	New York	09303	USA	randomemail1@yahoo.com	212-456-3646
2	982	Tom	Nook	123 Sesame Street	New York	09303	USA	randomemail@yahoo.com	212-234-3344
3	984	Tommy	Nook	125 Sesame Street	New York	09303	USA	randomemail2@yahoo.com	212-789-3567

Query 5 Result – art_order entity:

Query Result

All Rows Fetched: 3 in 0.012 seconds

	ORDERID	CUSTOMERID	ARTID	RENTALSTARTDATE	RENTAL_LENGTH	DELIVERY_DATE	NOPIECES	ORDER_TYPE	PAYMENTID
1	100987	982	4560	19-04-25	3	19-04-30	1	Rental	200987
2	100988	983	6778	19-04-30	1	19-05-05	1	Rental	200988
3	100989	984	8674	19-05-05	1	19-05-10	1	Rental	200989

Query 6 Result – payment entity:

Query Result

All Rows Fetched: 3 in 0.018 seconds

	PAYMENTID	ORDERID	CUSTOMERID	PAYMENTAMOUNT	DATEPAYMENT
1	200987	100987	982	250	19-05-01
2	200988	100988	983	450	19-04-29
3	200989	100989	984	450	19-04-30

4. Demo Views and Simple Queries

Code:

```
CREATE TABLE artist (
    ArtistID NUMBER PRIMARY KEY,
    Firstname VARCHAR2(25) NOT NULL,
    Lastname VARCHAR2(25) NOT NULL,
    City VARCHAR2(20),
    Region VARCHAR2(20),
    Country VARCHAR2(30) NOT NULL,
    Email VARCHAR(40),
    PhoneNumber VARCHAR2(12));

INSERT INTO artist
VALUES (123, 'Dawn', 'Lol', 'Toronto', 'ON', 'CA', 'lol@gmail.com',
'6471234567');
INSERT INTO artist
VALUES (456, 'Smith', 'Jones', 'Scarborough', 'ON', 'CA', 'sje@gmail.com',
'4167898907');
INSERT INTO artist
VALUES (789, 'Lee', 'Li', 'Brampton', 'ON', 'CA', 'leelol@gmail.com',
'4136547890');
INSERT INTO artist
VALUES (367, 'Sally', 'Hong', 'Markham', 'ON', 'CA', 'bluesky@yahoo.ca',
'6475781265');
INSERT INTO artist
VALUES (148, 'James', 'Lone', 'Toronto', 'ON', 'CA',
'jleartist@gmail.com', '4134563421');
INSERT INTO artist VALUES (001,'Leo','Da Vinci','Toronto', 'ON', 'CA',
'jleartist@gmail.com', '03502381921');
INSERT INTO artist VALUES (007,'Leonardo','Capablanca','Perth', 'WA',
'AUS', 'leoartsales@gmail.com', '441342634990');
INSERT INTO artist VALUES (333,'Kyle','Denise','Paris', 'IDF', 'FRA',
'slmyartkyle@gmail.com', '8896342111');
INSERT INTO artist VALUES (432,'Taylor','Octavian','Hauts-de-Seine',
'IDF', 'FRA', 'taywrkmadd@gmail.com', '99345234428');
INSERT INTO artist VALUES (578,'Sophia','Charlemagne','Rennes', 'BRE',
'FRA', '', '');
INSERT INTO artist VALUES (912,'Anthony','Constantinous','Bydgoszcz',
'KP', 'POL', 'anyforsalem@gmail.com', '90421563421');
INSERT INTO artist VALUES (754,'Kendrick','Van Gogh','Barcelona', 'CT',
```

```
'ESP', '', '');
INSERT INTO artist VALUES (332,'Louis','Picasso','Madrid', 'MD', 'ESP',
'', '');
INSERT INTO artist VALUES (719,'Seagle','Aivazovsky','Moscow', 'MOS',
'RUS', '', '');
INSERT INTO artist VALUES (900,'Tyler','Willendorf','Berlin', 'BE', 'DE',
'germartistmai@gmail.com', '8914574423');
INSERT INTO artist VALUES (891,'Pablo','Southerland','New York', 'NY',
'US', 'jennysellsart@gmail.com', '6784543234');
INSERT INTO artist VALUES (783,'Jean-Antoine','Watteau','Strasbourg',
'GES', 'FRA', '', '');
INSERT INTO artist VALUES (549,'Paul','Dillet','Arlon', 'WLX', 'BE',
'randomgall@gmail.com', '8893455678');
INSERT INTO artist VALUES (130,'Albert','Neuhuys','Istanbul', '34', 'TR',
'albertsales34@gmail.com', '1134373422');


SELECT *
FROM artist a RIGHT JOIN artwork b ON a.firstname||' '||a.lastname =
b.artist;
SELECT * FROM artist ORDER BY firstname ASC;
SELECT * FROM artist WHERE firstname LIKE 'S%';
SELECT ArtistID,Firstname, Lastname FROM artist WHERE firstname LIKE
'L%';
CREATE VIEW noArtists (total) AS SELECT count(*) from artist;
SELECT * from noArtists;
DROP VIEW noArtists;

CREATE VIEW livesInToronto AS
SELECT Firstname,Lastname, City, Region, Country
FROM artist
WHERE City='Toronto';
SELECT * FROM livesInToronto;
DROP VIEW livesInToronto;


CREATE TABLE artwork (
    ArtworkID NUMBER PRIMARY KEY,
    ArtistID NUMBER references artist(ArtistID),
    Artist VARCHAR(25),
    NameArt VARCHAR2(100),
    Exhibition VARCHAR2(100),
    DateAdded DATE NOT NULL,
    SizeArt VARCHAR2(20) NOT NULL,
    Price NUMBER,
```

```
    TypeArt VARCHAR(40),
    MediumArt VARCHAR(50),
    TimePeriod VARCHAR2(12),
    YearCreated NUMBER);

INSERT INTO artwork
VALUES (4560,(select ArtistID from artist where ArtistID = (select
ArtistID from artist where Firstname = 'Dawn')),'Dawn Lol', 'uhm',
'Oldendel', '2019-12-17', 'Medium', 2345,'Painting', 'Oil on
Canvas','1900s', 2016);
INSERT INTO artwork
VALUES (7891,(select ArtistID from artist where ArtistID = (select
ArtistID from artist where Firstname = 'Smith')),'Smith Jones', 'Bleu',
'Tulipshaw', '2019-2-08', 'Large', 1356,'Painting', 'Acrylic on
Canvas','Renaissance', 2019);
INSERT INTO artwork
VALUES (2345,(select ArtistID from artist where ArtistID = (select
ArtistID from artist where Firstname = 'Lee')),'Lee Li', 'Oythus',
'Relicmilie', '2019-2-08', 'Large', 1098,'Painting', 'Watercolor on Hot
Pressed Paper','Modern', 2003);
INSERT INTO artwork
VALUES (6778,(select ArtistID from artist where ArtistID = (select
ArtistID from artist where Firstname = 'Sally')),'Sally Hong', 'Rackaro',
'Skipleaf', '2019-3-18', 'Large', 5000,'Statue', 'Copper','Modern', 2006);
INSERT INTO artwork
VALUES (8674,(select ArtistID from artist where ArtistID = (select
ArtistID from artist where Firstname = 'James')),'James Lone', 'Lenveyas',
'Leafmill', '2019-3-18', 'Large', 5670,'Statue', 'Copper','Modern', 2019);
INSERT INTO artwork VALUES (1049,(select ArtistID from artist where
ArtistID = (select ArtistID from
artist where Firstname = 'Leo')),'Leo Da Vinci','Amour', 'Leafmill',
'2019-7-20', 'Small', 9670,'Painting', 'Oil on Canvas','Renaissance',
1490);
INSERT INTO artwork VALUES (7369,(select ArtistID from artist where
ArtistID = (select ArtistID from
artist where Firstname = 'Leonardo')),'Leonardo Capablanca','Belle Nuit',
'Oldendel', '2019-2-25', 'Medium', 5010,'Painting', 'Tempera','1950-
Present', 2000);
INSERT INTO artwork VALUES (9730,(select ArtistID from artist where
ArtistID = (select ArtistID from
artist where Firstname = 'Kyle')),'Kyle Denise','La Ciel', 'Oldendel',
'2019-10-01', 'Large', 9080,'Statue', 'Marble','1950-Present', 1990);
INSERT INTO artwork VALUES (4999,(select ArtistID from artist where
ArtistID = (select ArtistID from
artist where Firstname = 'Taylor')),'Taylor Octavian','War and Peace',
```

```
'Tulipshaw', '2019-12-19', 'Medium', 7000,'Sculpture', 'Marble','1900s',
1940);
INSERT INTO artwork VALUES (9764,(select ArtistID from artist where
ArtistID = (select ArtistID from
artist where Firstname = 'Sophia')),'Sophia Charlemagne','Lords',
'Tulipshaw', '2019-12-19', 'Large', 2323,'Painting',
'Pastel','Neoclassical', 1803);
INSERT INTO artwork VALUES (6892,(select ArtistID from artist where
ArtistID = (select ArtistID from
artist where Firstname = 'Anthony')),'Anthony Constantinous','Prince
Augusto', 'Tulipshaw', '2019-12-19', 'Large', 1089,'Painting', 'Acrylic on
Canvas','1900s', 1949);
INSERT INTO artwork VALUES (3436,(select ArtistID from artist where
ArtistID = (select ArtistID from
artist where Firstname = 'Kendrick')),'Kendrick Van Gogh','Silent Night',
'Relicmilie', '2019-2-25', 'Small', 7171,'Sculpture', 'Ivory','Medieval',
1343);
INSERT INTO artwork VALUES (5313,(select ArtistID from artist where
ArtistID = (select ArtistID from
artist where Firstname = 'Louis')),'Louis Picasso','Outers', 'Relicmilie',
'2019-2-25', 'Small', 8899,'Sculpture', 'Marble','Romantic', 1842);
INSERT INTO artwork VALUES (4335,(select ArtistID from artist where
ArtistID = (select ArtistID from
artist where Firstname = 'Seagle')),'Seagle Aivazovsky','Ganity',
'Oldendel', '2019-2-25', 'Small', 3899,'Painting', 'Oil on
Canvas','Renaissance', 1560);
INSERT INTO artwork VALUES (6604,(select ArtistID from artist where
ArtistID = (select ArtistID from
artist where Firstname = 'Tyler')),'Tyler Willendorf','Glory', 'Oldendel',
'2019-2-25', 'Small', 6899,'Sculpture', 'Marble','1950-Present', 1954);
INSERT INTO artwork VALUES (1340,(select ArtistID from artist where
ArtistID = (select ArtistID from
artist where Firstname = 'Pablo')),'Pablo Southerland','Dimension',
'Oldendel', '2019-2-25', 'Small', 9999,'Sculpture', 'Marble','1950-
Present', 2007);
INSERT INTO artwork VALUES (4949,(select ArtistID from artist where
ArtistID = (select ArtistID from
artist where Firstname = 'Jean-Antoine')),'Jean-Antoine Watteau','Calm',
'Skipleaf', '2019-4-12', 'Large', 4591,'Painting', 'Oil on
Canvas','Baroque', 1940);
INSERT INTO artwork VALUES (9814,(select ArtistID from artist where
ArtistID = (select ArtistID from
artist where Firstname = 'Paul')),'Paul Dillet','Calamity', 'Skipleaf',
'2019-4-13', 'Large', 6735,'Painting', 'Acrylic on Canvas','1950-Present',
1967);
```

```sql
INSERT INTO artwork VALUES (6645,(select ArtistID from artist where
ArtistID = (select ArtistID from
artist where Firstname = 'Albert')),'Albert Neuhuys','Reflection',
'Relicmilie', '2019-2-25', 'Small', 8899,'Sculpture',
'Gypsum','Realistic', 1895);


SELECT a.firstname, a.lastname FROM artist a, artwork w
WHERE a.firstname||' '||a.lastname = w.artist AND a.firstname LIKE 'S%'
AND w.typeart = 'Painting'
UNION(SELECT artist.firstname, artist.lastname FROM artist
WHERE firstname LIKE 'P%');

SELECT * FROM artwork ORDER BY Artist ASC;
SELECT DISTINCT MediumArt FROM artwork;
SELECT *
FROM artwork RIGHT JOIN exhibition ON (Exhibition = Gallery_Title);
SELECT * FROM artwork where MediumArt ='Copper';

CREATE VIEW artworkMediums AS
SELECT Artist,NameArt,TypeArt,MediumArt
FROM artwork
WHERE TypeArt='Painting';
SELECT * FROM artworkMediums;
DROP VIEW artworkMediums;

SELECT w.nameart, w.artist, o.orderid, c.firstname
FROM art_order o
INNER JOIN artwork w ON o.artworkid = w.artworkid
LEFT JOIN customer c ON o.customerid = c.customerid
ORDER BY w.artist ASC;

SELECT DISTINCT artist.firstname,artist.lastname, artwork.NameArt,
exhibition.StartDate
FROM artist
RIGHT JOIN artwork ON artist.firstname||' '||artist.lastname =
artwork.artist
RIGHT JOIN exhibition ON artwork.Exhibition = exhibition.gallery_title
WHERE artist.email is NOT NULL
ORDER BY artist.firstname ASC;

Select w.nameart, w.artist, o.orderid, c.firstname
FROM art_order o
INNER JOIN artwork w ON o.artworkid = w.artworkid
LEFT JOIN customer c ON o.customerid = c.customerid
```

```sql
ORDER BY w.artist ASC;

SELECT a.firstname, a.lastname FROM artist a, artwork w
WHERE a.firstname||' '||a.lastname = w.artist AND a.firstname LIKE 'S%'
AND w.typeart = 'Painting'
UNION(SELECT artist.firstname, artist.lastname FROM artist
   WHERE firstname LIKE 'P%');


SELECT 'Average Price of Sculptures', AVG(Price)
FROM  artwork
WHERE TypeArt = 'Sculpture';


DELETE
FROM
    artwork
WHERE
    ArtworkID = 7890;

CREATE TABLE exhibition (
    ArtistID int,
    FOREIGN KEY (ArtistID) REFERENCES artist(ArtistID),
    ExhibitionID int,
    Gallery_Title VARCHAR(50),
    StartDate DATE,
    EndDate DATE);

INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from artist where Firstname = 'Dawn')), 23206,'Oldendel', '2020-1-
08','2020-1-15');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from artist where Firstname = 'Smith')), 90865,'Tulipshaw', '2019-2-
08','2019-2-15');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from artist where Firstname = 'Lee')), 78964,'Relicmilie', '2019-2-
08','2019-2-20');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from artist where Firstname = 'Sally')), 14532,'Skipleaf', '2019-3-
18','2019-3-25');
INSERT INTO exhibition
```

```
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from artist where Firstname = 'James')), 23256,'Leafmill', '2019-3-
18','2019-3-25');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from
artist where Firstname = 'Leo')), 23256,'Leafmill', '2019-3-18','2019-3-
25');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from
artist where Firstname = 'Leonardo')), 23206,'Oldendel', '2020-1-
08','2020-1-
15');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from
artist where Firstname = 'Kyle')), 23206,'Oldendel', '2020-1-08','2020-1-
15');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from
artist where Firstname = 'Taylor')), 90865,'Tulipshaw', '2019-2-08','2019-
2-
15');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from
artist where Firstname = 'Sophia')), 90865,'Tulipshaw', '2019-2-08','2019-
2-
15');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from
artist where Firstname = 'Anthony')), 90865,'Tulipshaw', '2019-2-
08','2019-2-
15');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from
artist where Firstname = 'Kendrick')), 78964,'Relicmilie', '2019-2-
08','2019-2-
20');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
```

```
from
artist where Firstname = 'Louis')), 78964,'Relicmilie', '2019-2-08','2019-
2-
20');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from
artist where Firstname = 'Seagle')), 23206,'Oldendel', '2020-1-08','2020-
1-
15');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from
artist where Firstname = 'Tyler')), 23206,'Oldendel', '2020-1-08','2020-1-
15');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from
artist where Firstname = 'Pablo')), 23206,'Oldendel', '2020-1-08','2020-1-
15');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from
artist where Firstname = 'Jean-Antoine')), 14532,'Skipleaf', '2019-3-
18','2019-3-
25');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from
artist where Firstname = 'Paul')), 14532,'Skipleaf', '2019-3-18','2019-3-
25');
INSERT INTO exhibition
VALUES ((select ArtistID from artist where ArtistID = (select ArtistID
from
artist where Firstname = 'Albert')), 78964,'Relicmilie', '2019-2-
08','2019-2-
20');
SELECT * FROM exhibition;
SELECT * FROM exhibition where (StartDate BETWEEN '2019-02-01'AND '2019-
03-27');

CREATE VIEW exhibitionEndDate AS
SELECT ArtistID,ExhibitionID,Gallery_Title, StartDate,EndDate
FROM exhibition
WHERE EndDate BETWEEN '2019-01-01'AND '2019-03-01';
```

```sql
SELECT * FROM exhibitionEndDate;
DROP VIEW exhibitionEndDate;

CREATE TABLE customer (
    CustomerID NUMBER PRIMARY KEY,
    FirstName VARCHAR2(25) NOT NULL,
    LastName VARCHAR2(25) NOT NULL,
    Address VARCHAR(50),
    City VARCHAR2(20),
    PostalCode VARCHAR2(6),
    Country VARCHAR2(30),
    Email VARCHAR(40) NOT NULL,
    PhoneNumber VARCHAR2(12));

INSERT INTO customer
VALUES (00982,'Tom','Nook', '123 Sesame Street', 'New
York','09303','USA','randomemail@yahoo.com','212-234-3344');
INSERT INTO customer
VALUES (00983,'Timmy','Nook', '124 Sesame Street', 'New
York','09303','USA','randomemail1@yahoo.com','212-456-3646');
INSERT INTO customer
VALUES (00984,'Tommy','Nook', '125 Sesame Street', 'New
York','09303','USA','randomemail2@yahoo.com','212-789-3567');
INSERT INTO customer
VALUES (00985,'Thomas','Nook', '126 Buda Street', 'New
Orleans','90210','USA','thomNook1@gmail.com','312-264-2354');
INSERT INTO customer
VALUES (00986,'Tome','Nook', '127 Buda Street', 'New
Orleans','90210','USA','tomNook2@gmail.com','312-316-6666');
INSERT INTO customer
VALUES (00987,'Timothy','Nook', '128 Buda Street', 'New
Orleans','90210','USA','timtty@gmail.com','312-387-3264');


SELECT * FROM customer WHERE Address LIKE '%Sesame%'

SELECT Email, Count(*) AS CustomersWithGmail
FROM customer
WHERE Email LIKE '%gmail%'
GROUP BY Email;


SELECT * FROM customer ORDER BY FirstName ASC;
CREATE VIEW noCustomers (total) AS SELECT count(*) from customer;
SELECT * from noCustomers;
```

```sql
DROP VIEW noCustomers;

CREATE TABLE art_order (
    OrderID NUMBER PRIMARY KEY,
    CustomerID NUMBER REFERENCES customer(CustomerID),
    ArtworkID NUMBER REFERENCES artwork(ArtworkID),
    RentalStartDate DATE NOT NULL,
    Rental_Length int,
    Delivery_Date DATE NOT NULL,
    NoPieces int,
    Order_Type VARCHAR(50));

CREATE TABLE payment (
    PaymentID NUMBER PRIMARY KEY,
    OrderID NUMBER REFERENCES art_order(OrderID),
    CustomerID NUMBER REFERENCES customer(CustomerID),
    PaymentAmount NUMBER,
    DatePayment DATE);

ALTER TABLE art_order ADD (
PaymentID NUMBER REFERENCES payment(PaymentID));

INSERT INTO art_order
VALUES (100987,(select CustomerID from customer where CustomerID
=00982),(select ArtworkID from artwork where ArtworkID=4560),'2019-4-
25',3,'2019-4-30',1,'Rental',(select PaymentID from payment where
PaymentID =200987));
INSERT INTO art_order
VALUES (100988,(select CustomerID from customer where CustomerID
=00983),(select ArtworkID from artwork where ArtworkID=6778),'2019-4-
30',1,'2019-5-05',1,'Rental',(select PaymentID from payment where
PaymentID =200988));
INSERT INTO art_order
VALUES (100989,(select CustomerID from customer where CustomerID
=00984),(select ArtworkID from artwork where ArtworkID=8674),'2019-5-
05',1,'2019-05-10',1,'Rental',(select PaymentID from payment where
PaymentID =200989));
INSERT INTO art_order
VALUES (100990,(select CustomerID from customer where CustomerID
=00985),(select ArtworkID from artwork where ArtworkID=9764),'2019-2-
20',5,'2019-5-01',1,'Rental',(select PaymentID from payment where
PaymentID
=200990));
INSERT INTO art_order
VALUES (100991,(select CustomerID from customer where CustomerID
```

```
=00986),(select ArtworkID from artwork where ArtworkID=3436),'2019-3-
18',2,'2019-6-15',1,'Rental',(select PaymentID from payment where
PaymentID
=200991));
INSERT INTO art_order
VALUES (100992,(select CustomerID from customer where CustomerID
=00987),(select ArtworkID from artwork where ArtworkID=4949),'2019-4-
20',1,'2019-10-10',1,'Rental',(select PaymentID from payment where
PaymentID =200992));

SELECT Order_Type, COUNT(*) AS NoOfRentals
FROM  art_order WHERE  OrderID IS NOT NULL
GROUP BY Order_Type;




INSERT INTO payment
VALUES (200987,(select OrderID from art_order where
OrderID=100987),(select CustomerID from customer where CustomerID
=00982),250,'2019-5-01');
INSERT INTO payment
VALUES (200988,(select OrderID from art_order where
OrderID=100988),(select CustomerID from customer where CustomerID
=00983),450,'2019-4-29');
INSERT INTO payment
VALUES (200989,(select OrderID from art_order where
OrderID=100989),(select CustomerID from customer where CustomerID
=00984),450,'2019-4-30');
INSERT INTO payment
VALUES (200990,(select OrderID from art_order where
OrderID=100990),(select
CustomerID from customer where CustomerID =00985),150,'2019-5-01');
INSERT INTO payment
VALUES (200991,(select OrderID from art_order where
OrderID=100991),(select
CustomerID from customer where CustomerID =00986),1000,'2019-4-29');
INSERT INTO payment
VALUES (200992,(select OrderID from art_order where
OrderID=100992),(select
CustomerID from customer where CustomerID =00987),1249,'2019-4-30');


UPDATE art_order
SET PaymentID = 200989
```

```sql
WHERE CustomerID = 00984;

UPDATE art_order
SET PaymentID = 200987
WHERE CustomerID = 00982;

UPDATE art_order
SET PaymentID = 200988
WHERE CustomerID = 00983;

UPDATE art_order
SET PaymentID = 200990
WHERE CustomerID = 00985;

UPDATE art_order
SET PaymentID = 200991
WHERE CustomerID = 00986;

UPDATE art_order
SET PaymentID = 200992
WHERE CustomerID = 00987;

SELECT CustomerID, Address
FROM customer
WHERE address LIKE '%Sesame%'
UNION ALL
SELECT OrderID, Order_Type FROM art_order
WHERE Order_Type='Rental';

SELECT a.nameart, a.artist, a.mediumart FROM artwork a
WHERE a.timeperiod = '1950-Present'
MINUS(SELECT a.nameart, a.artist, a.mediumart FROM artwork a
WHERE a.mediumart = 'Marble')

SELECT * FROM art_order;
SELECT * FROM payment;
SELECT * FROM payment WHERE PaymentAmount <500 and PaymentAmount > 300;
```

4. Demo Views and Simple Queries

Database Tables:

Artist:

SELECT * FROM artist ORDER BY firstname ASC;

	ARTISTID	FIRSTNAME	LASTNAME	CITY	REGION	COUNTRY	EMAIL	PHONENUMBER
1	130	Albert	Neuhuys	Istanbul	34	TR	albertsales34@gmail.com	1134373422
2	912	Anthony	Constantinous	Bydgoszcz	KP	POL	anyforsalem@gmail.com	90421563421
3	123	Dawn	Lol	Toronto	ON	CA	lol@gmail.com	6471234567
4	148	James	Lone	Toronto	ON	CA	jleartist@gmail.com	4134563421
5	783	Jean-Antoine	Watteau	Strasbourg	GES	FRA	(null)	(null)
6	754	Kendrick	Van Gogh	Barcelona	CT	ESP	(null)	(null)
7	333	Kyle	Denise	Paris	IDF	FRA	slmyartkyle@gmail.com	8896342111
8	789	Lee	Li	Brampton	ON	CA	leelol@gmail.com	4136547890
9	1	Leo	Da Vinci	Toronto	ON	CA	jleartist@gmail.com	03502381921
10	7	Leonardo	Capablanca	Perth	WA	AUS	leoartsales@gmail.com	441342634990
11	332	Louis	Picasso	Madrid	MD	ESP	(null)	(null)
12	891	Pablo	Southerland	New York	NY	US	jennysellsart@gmail.com	6784543234
13	549	Paul	Dillet	Arlon	WLX	BE	randomgall@gmail.com	8893455678
14	367	Sally	Hong	Markham	ON	CA	bluesky@yahoo.ca	6475781265
15	719	Seagle	Aivazovsky	Moscow	MOS	RUS	(null)	(null)
16	456	Smith	Jones	Scarborough	ON	CA	sje@gmail.com	4167898907
17	578	Sophia	Charlemagne	Rennes	BRE	FRA	(null)	(null)
18	432	Taylor	Octavian	Hauts-de-Seine	IDF	FRA	taywrkmadd@gmail.com	99345234428
19	900	Tyler	Willendorf	Berlin	BE	DE	germartistmai@gmail.com	8914574423

Artwork:

SELECT * FROM artwork ORDER BY Artist ASC;

	ARTWORKID	ARTISTID	ARTIST	NAMEART	EXHIBITION	DATEADDED	SIZEART	PRICE	TYPEART	MEDIUMART	TIMEPERIOD	YEARCREATED
1	6645	130	Albert Neuhuys	Reflection	Relicmilie	19-02-25	Small	8899	Sculpture	Gypsum	Realistic	1895
2	6892	912	Anthony Constantinous	Prince Augusto	Tulipshaw	19-12-19	Large	1089	Painting	Acrylic on Canvas	1900s	1949
3	4560	123	Dawn Lol	uhm	Oldendel	19-12-17	Medium	2345	Painting	Oil on Canvas	1900s	2016
4	8674	148	James Lone	Lenveyas	Leafmill	19-03-18	Large	5670	Statue	Copper	Modern	2019
5	4949	783	Jean-Antoine Watteau	Calm	Skipleaf	19-04-12	Large	4591	Painting	Oil on Canvas	Baroque	1940
6	3436	754	Kendrick Van Gogh	Silent Night	Relicmilie	19-02-25	Small	7171	Sculpture	Ivory	Medieval	1343
7	9730	333	Kyle Denise	La Ciel	Oldendel	19-10-01	Large	9080	Statue	Marble	1950-Present	1990
8	2345	789	Lee Li	Oythus	Relicmilie	19-02-08	Large	1098	Painting	Watercolor on Hot Pressed Paper	Modern	2003
9	1049	1	Leo Da Vinci	Amour	Leafmill	19-07-20	Small	9670	Painting	Oil on Canvas	Renaissance	1490
10	7369	7	Leonardo Capablanca	Belle Nuit	Oldendel	19-02-25	Medium	5010	Painting	Tempera	1950-Present	2000
11	5313	332	Louis Picasso	Outers	Relicmilie	19-02-25	Small	8899	Sculpture	Marble	Romantic	1842
12	1340	891	Pablo Southerland	Dimension	Oldendel	19-02-25	Small	9999	Sculpture	Marble	1950-Present	2007
13	9814	549	Paul Dillet	Calamity	Skipleaf	19-04-13	Large	6735	Painting	Acrylic on Canvas	1950-Present	1967
14	6778	367	Sally Hong	Rackaro	Skipleaf	19-03-18	Large	5000	Statue	Copper	Modern	2006
15	4335	719	Seagle Aivazovsky	Ganity	Oldendel	19-02-25	Small	3899	Painting	Oil on Canvas	Renaissance	1560
16	7891	456	Smith Jones	Bleu	Tulipshaw	19-02-08	Large	1356	Painting	Acrylic on Canvas	Renaissance	2019
17	9764	578	Sophia Charlemagne	Lords	Tulipshaw	19-12-19	Large	2323	Painting	Pastel	Neoclassical	1803
18	4999	432	Taylor Octavian	War and Peace	Tulipshaw	19-12-19	Medium	7000	Sculpture	Marble	1900s	1940
19	6604	900	Tyler Willendorf	Glory	Oldendel	19-02-25	Small	6899	Sculpture	Marble	1950-Present	1954

Exhibition:

SELECT * FROM exhibition;

	ARTISTID	EXHIBITIONID	GALLERY_TITLE	STARTDATE	ENDDATE
1	123	23206	Oldendel	20-01-08	20-01-05
2	456	90865	Tulipshaw	19-02-08	19-02-15
3	789	78964	Relicmilie	19-02-08	19-02-20
4	367	14532	Skipleaf	19-03-18	19-03-25
5	148	23256	Leafmill	19-03-18	19-03-25

Customer:

SELECT * FROM customer ORDER BY FirstName ASC;

	CUSTOMERID	FIRSTNAME	LASTNAME	ADDRESS	CITY	POSTALCODE	COUNTRY	EMAIL	PHONENUMBER
1	983	Timmy	Nook	124 Sesame Street	New York	09303	USA	randomemail1@yahoo.com	212-456-3646
2	982	Tom	Nook	123 Sesame Street	New York	09303	USA	randomemail@yahoo.com	212-234-3344
3	984	Tommy	Nook	125 Sesame Street	New York	09303	USA	randomemail2@yahoo.com	212-789-3567

Art Order:
SELECT * FROM art_order;

	ORDERID	CUSTOMERID	ARTID	RENTALSTARTDATE	RENTAL_LENGTH	DELIVERY_DATE	NOPIECES	ORDER_TYPE	PAYMENTID
1	100987	982	4560	19-04-25	3	19-04-30	1	Rental	200987
2	100988	983	6778	19-04-30	1	19-05-05	1	Rental	200988
3	100989	984	8674	19-05-05	1	19-05-10	1	Rental	200989

Payment:

SELECT * FROM payment;

	PAYMENTID	ORDERID	CUSTOMERID	PAYMENTAMOUNT	DATEPAYMENT
1	200987	100987	982	250	19-05-01
2	200988	100988	983	450	19-04-29
3	200989	100989	984	450	19-04-30

Queries (Advanced):

1.
```
SELECT 'Average Price of Sculptures', AVG(Price)
FROM  artwork
WHERE TypeArt = 'Sculpture';
```

	'AVERAGEPRICEOFSCULPTURES'	AVG(PRICE)
1	Average Price of Sculptures	8144.5

2.
```
SELECT a.firstname, a.lastname FROM artist a, artwork w
WHERE a.firstname||' '||a.lastname = w.artist AND a.firstname LIKE 'S%'
AND w.typeart = 'Painting'
UNION(SELECT artist.firstname, artist.lastname FROM artist

WHERE firstname LIKE 'P%');
```

	FIRSTNAME	LASTNAME
1	Pablo	Southerland
2	Paul	Dillet
3	Seagle	Aivazovsky
4	Smith	Jones
5	Sophia	Charlemagne

3.
```
SELECT *
FROM artist a RIGHT JOIN artwork b ON a.firstname||' '||a.lastname = b.artist;
```

	ARTISTID	FIRSTNAME	LASTNAME	CITY	REGION	COUNTRY	EMAIL	PHONENUMBER	ARTWORKID	ARTISTID_1	ARTIST	NAMEART	EXHIBITION	DATEADDED	SIZEART	PRICE	TYPEART	MEDIUMART	TIMEPERIOD	YEARCREATED
1	123	Dawn	Lol	Toronto	ON	CA	lol@gmail.com	6471234567	4560	123	Dawn Lol	uhm	Oldendel	19-12-17	Medium	2345	Painting	Oil on Canvas	1900s	2016
2	456	Smith	Jones	Scarborough	ON	CA	sje@gmail.com	4167898907	7891	456	Smith Jones	Bleu	Tulipshaw	19-02-08	Large	1356	Painting	Acrylic on Canvas	Renaissance	2019
3	789	Lee	Li	Brampton	ON	CA	leelol@gmail.com	4136547890	2345	789	Lee Li	Oythus	Relicmilie	19-02-08	Large	1098	Painting	Watercolor on Hot Pressed Paper	Modern	2003
4	367	Sally	Hong	Markham	ON	CA	bluesky@yahoo.ca	6475781265	6778	367	Sally Hong	Rackaro	Skipleaf	19-03-18	Large	5000	Statue	Copper	Modern	2006
5	148	James	Lone	Toronto	ON	CA	jleartist@gmail.com	4134563421	8674	148	James Lone	Lenveyas	Leafmill	19-03-18	Large	5670	Statue	Copper	Modern	2019
6	1	Leo	Da Vinci	Toronto	ON	CA	jleartist@gmail.com	03502381921	1049	1	Leo Da Vinci	Amour	Leafmill	19-07-20	Small	9670	Painting	Oil on Canvas	Renaissance	1490
7	7	Leonardo	Capablanca	Perth	WA	AUS	leoartsales@gmail.com	441342634990	7369	7	Leonardo Capablanca	Belle Nuit	Oldendel	19-02-25	Medium	5010	Painting	Tempera	1950-Present	2000
8	333	Kyle	Denise	Paris	IDF	FRA	slmyartkyle@gmail.com	8896342111	9730	333	Kyle Denise	La Ciel	Oldendel	19-10-01	Large	9080	Statue	Marble	1950-Present	1990
9	432	Taylor	Octavian	Hauts-de-Seine	IDF	FRA	taywrkmadd@gmail.com	99345234428	4999	432	Taylor Octavian	War and Peace	Tulipshaw	19-12-19	Medium	7000	Sculpture	Marble	1900s	1940
10	578	Sophia	Charlemagne	Rennes	BRE	FRA	(null)	(null)	9764	578	Sophia Charlemagne	Lords	Tulipshaw	19-12-19	Large	2323	Painting	Pastel	Neoclassical	1803
11	912	Anthony	Constantinous	Bydgoszcz	KP	POL	anyforsalem@gmail.com	90421563421	6892	912	Anthony Constantinous	Prince Augusto	Tulipshaw	19-12-19	Large	1089	Painting	Acrylic on Canvas	1900s	1949
12	754	Kendrick	Van Gogh	Barcelona	CT	ESP	(null)	(null)	3436	754	Kendrick Van Gogh	Silent Night	Relicmilie	19-02-25	Small	7171	Sculpture	Ivory	Medieval	1343
13	332	Louis	Picasso	Madrid	MD	ESP	(null)	(null)	5313	332	Louis Picasso	Outers	Relicmilie	19-02-25	Small	8899	Sculpture	Marble	Romantic	1842
14	719	Seagle	Aivazovsky	Moscow	MOS	RUS	(null)	(null)	4335	719	Seagle Aivazovsky	Ganity	Oldendel	19-02-25	Small	3899	Painting	Oil on Canvas	Renaissance	1560
15	900	Tyler	Willendorf	Berlin	BE	DE	germartistmai@gmail.com	8914574423	6604	900	Tyler Willendorf	Glory	Oldendel	19-02-25	Small	6899	Sculpture	Marble	1950-Present	1954
16	891	Pablo	Southerland	New York	NY	US	jennysellsart@gmail.com	6784543234	1340	891	Pablo Southerland	Dimension	Oldendel	19-02-25	Small	9999	Sculpture	Marble	1950-Present	2007
17	783	Jean-Antoine	Watteau	Strasbourg	GES	FRA	(null)	(null)	4949	783	Jean-Antoine Watteau	Calm	Skipleaf	19-04-12	Large	4591	Painting	Oil on Canvas	Baroque	1940
18	549	Paul	Dillet	Arlon	WLX	BE	randomgall@gmail.com	8893455678	9814	549	Paul Dillet	Calamity	Skipleaf	19-04-13	Large	6735	Painting	Acrylic on Canvas	1950-Present	1967
19	130	Albert	Neuhuys	Istanbul	34	TR	albertsales34@gmail.com	1134373422	6645	130	Albert Neuhuys	Reflection	Relicmilie	19-02-25	Small	8899	Sculpture	Gypsum	Realistic	1895

4. SELECT * FROM artist WHERE firstname LIKE 'S%';

	ARTISTID	FIRSTNAME	LASTNAME	CITY	REGION	COUNTRY	EMAIL	PHONENUMBER
1	456	Smith	Jones	Scarborough	ON	CA	sje@gmail.com	4167898907
2	367	Sally	Hong	Markham	ON	CA	bluesky@yahoo.ca	6475781265
3	578	Sophia	Charlemagne	Rennes	BRE	FRA	(null)	(null)
4	719	Seagle	Aivazovsky	Moscow	MOS	RUS	(null)	(null)

5.SELECT ArtistID,Firstname, Lastname FROM artist WHERE firstname LIKE 'L%';

	ARTISTID	FIRSTNAME	LASTNAME
1	789	Lee	Li
2	1	Leo	Da Vinci
3	7	Leonardo	Capablanca
4	332	Louis	Picasso

6. SELECT DISTINCT MediumArt FROM artwork;

	MEDIUMART
1	Marble
2	Oil on Canvas
3	Tempera
4	Gypsum
5	Copper
6	Acrylic on Canvas
7	Watercolor on Hot Pressed Paper
8	Pastel
9	Ivory

7. SELECT * FROM artwork RIGHT JOIN exhibition ON (Exhibition = Gallery Title);

	ARTWORKID	ARTIST	NAMEART	EXHIBITION	DATEADDED	SIZEART	PRICE	TYPEART	MEDIUMART	TIMEPERIOD	YEARCREATED	ARTISTID	EXHIBITIONID	GALLERY_TITLE	STARTDATE	ENDDATE
1	4560	Dawn Lol	uhm	Oldendel	19-12-17	Medium	2345	Painting	Oil on Canvas	1900s	2016	123	23206	Oldendel	20-01-08	20-01-05
2	7891	Smith Jones	Bleu	Tulipshaw	19-02-08	Large	1356	Painting	Acrylic on Canvas	Renaissance	2019	456	90865	Tulipshaw	19-02-08	19-02-15
3	2345	Lee Li	Oythus	Relicmilie	19-02-08	Large	1098	Painting	Watercolor on Hot Pressed Paper	Modern	2003	789	78964	Relicmilie	19-02-08	19-02-20
4	6778	Sally Hong	Rackaro	Skipleaf	19-03-18	Large	5000	Statue	Copper	Modern	2006	367	14532	Skipleaf	19-03-18	19-03-25
5	8674	James Lone	Lenveyas	Leafmill	19-03-18	Large	5670	Statue	Copper	Modern	2019	148	23256	Leafmill	19-03-18	19-03-25

8. SELECT * FROM artwork where MediumArt ='Copper';

	ARTWORKID	ARTIST	NAMEART	EXHIBITION	DATEADDED	SIZEART	PRICE	TYPEART	MEDIUMART	TIMEPERIOD	YEARCREATED
1	6778	Sally Hong	Rackaro	Skipleaf	19-03-18	Large	5000	Statue	Copper	Modern	2006
2	8674	James Lone	Lenveyas	Leafmill	19-03-18	Large	5670	Statue	Copper	Modern	2019

9. SELECT * FROM exhibition where (StartDate BETWEEN '2019-02-01'AND '2019-03-27');

	ARTISTID	EXHIBITIONID	GALLERY_TITLE	STARTDATE	ENDDATE
1	456	90865	Tulipshaw	19-02-08	19-02-15
2	789	78964	Relicmilie	19-02-08	19-02-20
3	367	14532	Skipleaf	19-03-18	19-03-25
4	148	23256	Leafmill	19-03-18	19-03-25
5	1	23256	Leafmill	19-03-18	19-03-25
6	432	90865	Tulipshaw	19-02-08	19-02-15
7	578	90865	Tulipshaw	19-02-08	19-02-15
8	912	90865	Tulipshaw	19-02-08	19-02-15
9	754	78964	Relicmilie	19-02-08	19-02-20
10	332	78964	Relicmilie	19-02-08	19-02-20
11	783	14532	Skipleaf	19-03-18	19-03-25
12	549	14532	Skipleaf	19-03-18	19-03-25
13	130	78964	Relicmilie	19-02-08	19-02-20

10. SELECT * FROM customer WHERE Address LIKE '%Sesame%'

	CUSTOMERID	FIRSTNAME	LASTNAME	ADDRESS	CITY	POSTALCODE	COUNTRY	EMAIL	PHONENUMBER
1	982	Tom	Nook	123 Sesame Street	New York	09303	USA	randomemail@yahoo.com	212-234-3344
2	983	Timmy	Nook	124 Sesame Street	New York	09303	USA	randomemail1@yahoo.com	212-456-3646
3	984	Tommy	Nook	125 Sesame Street	New York	09303	USA	randomemail2@yahoo.com	212-789-3567

11. SELECT * FROM payment WHERE PaymentAmount <500 and PaymentAmount > 300;

	PAYMENTID	ORDERID	CUSTOMERID	PAYMENTAMOUNT	DATEPAYMENT
1	200988	100988	983	450	19-04-29
2	200989	100989	984	450	19-04-30

12. Joining 3 Tables

```
Select w.nameart, w.artist, o.orderid, c.firstname

FROM art_order o

INNER JOIN artwork w ON o.artworkid = w.artworkid

LEFT JOIN customer c ON o.customerid = c.customerid

ORDER BY w.artist ASC;
```

	NAMEART	ARTIST	ORDERID	FIRSTNAME
1	uhm	Dawn Lol	100987	Tom
2	Lenveyas	James Lone	100989	Tommy
3	Calm	Jean-Antoine Watteau	100992	Timothy
4	Silent Night	Kendrick Van Gogh	100991	Tome
5	Rackaro	Sally Hong	100988	Timmy
6	Lords	Sophia Charlemagne	100990	Thomas

VIEWS:

1.
```
CREATE VIEW noArtists (total) AS SELECT count(*) from artist;
SELECT * from noArtists;
```

	TOTAL
1	11

2.
```
CREATE VIEW livesInToronto AS
SELECT Firstname,Lastname, City, Region, Country
FROM artist
WHERE City='Toronto';
```

	FIRSTNAME	LASTNAME	CITY	REGION	COUNTRY
1	Dawn	Lol	Toronto	ON	CA
2	James	Lone	Toronto	ON	CA
3	Leonardo	Da Vinci	Toronto	ON	CA

3.
```
CREATE VIEW artworkMediums AS
SELECT Artist,NameArt,TypeArt,MediumArt
FROM artwork
WHERE TypeArt='Painting';
SELECT * FROM artworkMediums;
```

	ARTIST	NAMEART	TYPEART	MEDIUMART
1	Dawn Lol	uhm	Painting	Oil on Canvas
2	Smith Jones	Bleu	Painting	Acrylic on Canvas
3	Lee Li	Oythus	Painting	Watercolor on Hot Pressed Paper

4.
```
CREATE VIEW exhibitionEndDate AS
SELECT ArtistID,ExhibitionID,Gallery_Title, StartDate,EndDate
FROM exhibition
WHERE EndDate BETWEEN '2019-01-01'AND '2019-03-01';
SELECT * FROM exhibitionEndDate;
```

	ARTISTID	EXHIBITIONID	GALLERY_TITLE	STARTDATE	ENDDATE
1	456	90865	Tulipshaw	19-02-08	19-02-15
2	789	78964	Relicmilie	19-02-08	19-02-20

5.
```
CREATE VIEW noCustomers (total) AS SELECT count(*) from customer;
SELECT * from noCustomers;
```

	TOTAL
1	3

5. Query Demonstration of Advanced Queries by Unix Shell

Menu Program:

```
#!/bin/bash
MainMenu()
{
while [ "$CHOICE" != "START" ]
do
        clear
        echo "=================================================================="
        echo "|                 Oracle All Inclusive Tool
|"
        echo "|                              Main Menu - Select Desired Operation(s):
|"
        echo "|       <CTRL-Z Anytime to Enter Interactive CMD Prompt>
|"
echo "------------------------------------------------------------------"
echo " $IS_SELECTEDM M)  View Manual"
echo " "
echo " $IS_SELECTED1 1)  Drop Tables"
echo " $IS_SELECTED2 2)  Create Tables"
echo " $IS_SELECTED3 3)  Populate Tables"
echo " $IS_SELECTED4 4)  Query Tables"
echo " "
echo " $IS_SELECTED4     a)  Customer and Payments"
echo " $IS_SELECTED4     b)  Artist and Artwork"
echo " $IS_SELECTED4     c)  Exhibition:Artwork and Corresponding Art in EXBID23206"
```

```
echo " $IS_SELECTED4     d)  Display Artists that have Small Sculptures"

echo " $IS_SELECTED4     e)  Average Price of Sculptures"

echo " $IS_SELECTED4     f)  Number of Rentals"

echo " $IS_SELECTED4     g)  Customers Using a Gmail Account"

echo " $IS_SELECTED4     h)  Artists with Artwork Belonging in 1950s NOT made w/ Marble"

echo " $IS_SELECTED4     i)  Customers Living on Sesame Street and Rental Orders"
echo " $IS_SELECTED4     j)  Joining First and Last Names"

echo " "

echo " $IS_SELECTEDX X)  Force/Stop/Kill Oracle DB"

echo " "

echo " $IS_SELECTEDE E)  End/Exit"

echo "Choose: "


read CHOICE


if [ "$CHOICE" == "0" ]

then

        echo "Nothing Here"

elif [ "$CHOICE" == "1" ]

then

        bash drop_tables.sh


elif [ "$CHOICE" == "2" ]

then

        bash create_tables.sh


elif [ "$CHOICE" == "3" ]
```

```
then
        bash populate_tables.sh
        Pause
elif [ "$CHOICE" == "a" ]
then
        bash q1.sh
elif [ "$CHOICE" == "b" ]
then
        bash q2.sh
elif [ "$CHOICE" == "c" ]
then
        bash q3.sh
elif [ "$CHOICE" == "d" ]
then
        bash q4.sh
elif [ "$CHOICE" == "e" ]
then
        bash q5.sh
elif [ "$CHOICE" == "f" ]
then
        bash q6.sh
  elif [ "$CHOICE" == "g" ]
then
        bash q7.sh
  elif [ "$CHOICE" == "h" ]
then
        bash q8.sh
```

```
  elif [ "$CHOICE" == "i" ]
then
        bash q9.sh
  elif [ "$CHOICE" == "j" ]
then
        bash q10.sh
elif [ "$CHOICE" == "E" ]
then
echo "Bye"
  sleep 1
        exit
else
  echo "Errpr"
  sleep 0.5
fi
done
}
#--COMMENTS BLOCK--
# Main Program
#--COMMENTS BLOCK--
ProgramStart()
{
        StartMessage
        while [ 1 ]
        do
        MainMenu
        done
```

}

ProgramStart

MENU:

```
-----------------------------------------------------------------
  M)  View Manual

  1)  Drop Tables
  2)  Create Tables
  3)  Populate Tables
  4)  Query Tables

      a)  Customer and Payments
      b)  Artist and Artwork
      c)  Exhibition:Artwork and Corresponding Art in EXBID23206
      d)  Display Artists that have Small Sculptures
      e)  Average Price of Sculptures
      f)  Number of Rentals
      g)  Customers Using a Gmail Account
      h)  Artists with Artwork Belonging in 1950s NOT made w/ Marble
      i)  Customers Living on Sesame Street and Rental Orders

  X)  Force/Stop/Kill Oracle DB

  E)  End/Exit
Choose:
```

ADVANCED QUERIES RESULTS:

a) > Customer and Payments

SELECT c.FIRSTNAME, c.LASTNAME, p.PAYMENTAMOUNT
FROM art_order o, customer c, payment p
WHERE c.CUSTOMERID = o.CUSTOMERID AND o.PAYMENTID = p.PAYMENTID
ORDER BY PAYMENTAMOUNT DESC;

```
SQL> SQL>   2    3    4
FIRSTNAME                 LASTNAME                  PAYMENTAMOUNT
------------------------- ------------------------- -------------
Timothy                   Nook                               1249
Tome                      Nook                               1000
Tommy                     Nook                                450
Tom                       Nook                                450
Timmy                     Nook                                250
Thomas                    Nook                                150

6 rows selected.
```

b) > Artist and Artwork

SELECT a.NAMEART, a.TYPEART
FROM artwork a, art_order o
WHERE o.ARTWORKID = a.ARTWORKID;

```
SQL> SQL>   2    3
NAMEART
--------------------------------------------------------------------------------
TYPEART
----------------------------------------
uhm
Painting

Rackaro
Statue

Lenveyas
Statue


NAMEART
--------------------------------------------------------------------------------
TYPEART
----------------------------------------
Lords
Painting

Silent Night
Sculpture

Calm
Painting


6 rows selected.
```

c) > Exhibition: Artwork and Corresponding Art using DISTINCT

SELECT DISTINCT a.ARTIST, a.NAMEART, a.PRICE FROM
Artwork a, Exhibition e
WHERE e.EXHIBITIONID = 23206 AND a.EXHIBITION = e.GALLERY_TITLE;

```
ARTIST
-------------------------
NAMEART
--------------------------------------------------------------------------------
     PRICE
----------

Pablo Southerland
Dimension
      9999

Tyler Willendorf
Glory

ARTIST
-------------------------
NAMEART
--------------------------------------------------------------------------------
     PRICE
----------
      6899

Kyle Denise
La Ciel
      9080

Dawn Lol

ARTIST
-------------------------
NAMEART
--------------------------------------------------------------------------------
     PRICE
----------
uhm
      2345


6 rows selected.
```

d) >Display Artists that have Small Sculptures using EXIST and DISTINCT

```
SELECT DISTINCT Artist, SizeArt,TypeArt
FROM artwork a
WHERE EXISTS
   (SELECT *
    FROM  artwork b
    WHERE a.TypeArt = 'Sculpture'
        and a.SizeArt = 'Small');
```

```
SQL> SQL>   2    3    4    5    6    7
ARTIST                    SIZEART
------------------------- --------------------
TYPEART
----------------------------------------
Tyler Willendorf          Small
Sculpture

Louis Picasso             Small
Sculpture

Kendrick Van Gogh         Small
Sculpture


ARTIST                    SIZEART
------------------------- --------------------
TYPEART
----------------------------------------
Pablo Southerland         Small
Sculpture

Albert Neuhuys            Small
Sculpture
```

e) >Average Price of Sculptures using AVG

```
SELECT 'Average Price of Sculptures', AVG(Price)
FROM  artwork
WHERE TypeArt = 'Sculpture';
```

```
SQL> SQL> SQL>   2    3
'AVERAGEPRICEOFSCULPTURES'  AVG(PRICE)
--------------------------- ----------
Average Price of Sculptures     8144.5
```

f) > Number of Rentals using COUNT and GROUP BY

SELECT Order_Type, COUNT(*) AS NoOfRentals

FROM art_order WHERE OrderID IS NOT NULL

GROUP BY Order_Type;

```
SQL> SQL>   2    3
ORDER_TYPE                                          NOOFRENTALS
--------------------------------------------------- -----------
Rental                                                        6
```

g) Customers Using a gmail account using Count and Group By
SELECT Email, Count(*) AS CustomersWithGmail

FROM customer

WHERE Email LIKE '%gmail%'

GROUP BY Email;

```
SQL> SQL>   2    3    4
EMAIL                                     CUSTOMERSWITHGMAIL
----------------------------------------- ------------------
timnook2@gmail.com                                         1
thomnook1@gmail.com                                        1
tomnook@gmail.com                                          1
```

h) > Artists with Artwork Belonging in 1950s time period not made w/ marble using SELECT & MINUS

SELECT a.nameart, a.artist, a.mediumart FROM artwork a

WHERE a.timeperiod = '1950-Present'

MINUS(SELECT a.nameart, a.artist, a.mediumart FROM artwork a

WHERE a.mediumart = 'Marble')

```
NAMEART
--------------------------------------------------------------------------------
ARTIST                    MEDIUMART
------------------------- --------------------------------------------------
Lenveyas
James Lone                Copper

Oythus
Lee Li                    Watercolor on Hot Pressed Paper

Rackaro
Sally Hong                Copper
```

i) > Customers living on Sesame Street and Rental Orders using LIKE and UNION

SELECT CustomerID, Address

FROM customer

WHERE address LIKE '%Sesame%'

UNION ALL

SELECT OrderID, Order_Type FROM art_order

WHERE Order_Type='Rental';

```
SQL> SQL>   2    3    4    5    6
CUSTOMERID ADDRESS
---------- --------------------------------------------------
       982 123 Sesame Street
       983 124 Sesame Street
       984 125 Sesame Street
    100987 Rental
    100988 Rental
    100989 Rental
    100990 Rental
    100991 Rental
    100992 Rental

9 rows selected
```

CREATING/POPULATING Database:

```
#!/bin/sh

#export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib

sqlplus64
"a84nguye/05271893@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerso
n.ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF

CREATE TABLE artist (ArtistID NUMBER PRIMARY KEY,Firstname VARCHAR2(25) NOT NULL,

Lastname VARCHAR2(25) NOT NULL,City VARCHAR2(20),Region VARCHAR2(20),

Country VARCHAR2(30) NOT NULL,Email VARCHAR(40),PhoneNumber VARCHAR2(12));


CREATE TABLE artwork (

ArtworkID NUMBER PRIMARY KEY,

Artist VARCHAR(25),

NameArt VARCHAR2(100),

Exhibition VARCHAR2(100),

DateAdded DATE NOT NULL,

SizeArt VARCHAR2(20) NOT NULL,

Price NUMBER,

TypeArt VARCHAR(40),

MediumArt VARCHAR(50),

TimePeriod VARCHAR2(12),

YearCreated NUMBER);


CREATE TABLE exhibition (

ArtistID int,

FOREIGN KEY (ArtistID) REFERENCES artist(ArtistID),

ExhibitionID int,

Gallery_Title VARCHAR(50),
```

```
StartDate DATE,
EndDate DATE);

CREATE TABLE customer (
CustomerID NUMBER PRIMARY KEY,
FirstName VARCHAR2(25) NOT NULL,
LastName VARCHAR2(25) NOT NULL,
Address VARCHAR(50),
City VARCHAR2(20),
PostalCode VARCHAR2(6),
Country VARCHAR2(30),
Email VARCHAR(40) NOT NULL,
PhoneNumber VARCHAR2(12));

CREATE TABLE art_order (
OrderID NUMBER PRIMARY KEY,
CustomerID NUMBER,
FOREIGN KEY (CustomerID) REFERENCES customer(CustomerID),
ArtworkID NUMBER REFERENCES artwork(ArtworkID),
RentalStartDate DATE NOT NULL,
Rental_Length int,
Delivery_Date DATE NOT NULL,
NoPieces int,
Order_Type VARCHAR(50));

CREATE TABLE payment (
PaymentID NUMBER PRIMARY KEY,
```

OrderID NUMBER REFERENCES art_order(OrderID),

CustomerID NUMBER REFERENCES customer(CustomerID),

PaymentAmount NUMBER,

DatePayment DATE);

ALTER TABLE art_order ADD (

PaymentID NUMBER REFERENCES payment(PaymentID));

exit;

EOF

```
SQL>   2    3
Table created.

SQL> SQL>   2    3    4    5    6    7    8    9   10   11   12
Table created.

SQL> SQL>   2    3    4    5    6    7
Table created.

SQL> SQL>   2    3    4    5    6    7    8    9   10
Table created.

SQL> SQL>   2    3    4    5    6    7    8    9   10
Table created.

SQL> SQL>   2    3    4    5    6
Table created.

SQL> SQL> SQL>   2
Table altered.
```

POPULATING TABLES:

```
#!/bin/sh

#export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib

sqlplus64
"a84nguye/05271893@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerso
n.ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF

ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY-MM-DD';


INSERT INTO artist VALUES (123, 'Dawn', 'Lol', 'Toronto', 'ON', 'CA', 'lol@gmail.com',
'6471234567');

INSERT INTO artist VALUES (456, 'Smith', 'Jones', 'Scarborough', 'ON', 'CA', 'sje@gmail.com',
'4167898907');

INSERT INTO artist VALUES (789, 'Lee', 'Li', 'Brampton', 'ON', 'CA', 'leelol@gmail.com',
'4136547890');

INSERT INTO artist VALUES (367, 'Sally', 'Hong', 'Markham', 'ON', 'CA', 'bluesky@yahoo.ca',
'6475781265');

INSERT INTO artist VALUES (148, 'James', 'Lone', 'Toronto', 'ON', 'CA', 'jleartist@gmail.com',
'4134563421');

INSERT INTO artist VALUES (001,'Leo','Da Vinci','Toronto', 'ON', 'CA', '', '');

INSERT INTO artist VALUES (007,'Leonardo','Capablanca','Perth', 'WA', 'AUS',
'leoartsales@gmail.com', '441342634990');

INSERT INTO artist VALUES (333,'Kyle','Denise','Paris', 'IDF', 'FRA', 'slmyartkyle@gmail.com',
'8896342111');

INSERT INTO artist VALUES (432,'Taylor','Octavian','Hauts-de-Seine', 'IDF', 'FRA',
'taywrkmadd@gmail.com', '99345234428');

INSERT INTO artist VALUES (578,'Sophia','Charlemagne','Rennes', 'BRE', 'FRA', '', '');

INSERT INTO artist VALUES (912,'Anthony','Constantinous','Bydgoszcz', 'KP', 'POL',
'anyforsalem@gmail.com', '90421563421');

INSERT INTO artist VALUES (754,'Kendrick','Van Gogh','Barcelona', 'CT', 'ESP', '', '');

INSERT INTO artist VALUES (332,'Louis','Picasso','Madrid', 'MD', 'ESP', '', '');
```

INSERT INTO artist VALUES (719,'Seagle','Aivazovsky','Moscow', 'MOS', 'RUS', '', '');

INSERT INTO artist VALUES (900,'Tyler','Willendorf','Berlin', 'BE', 'DE', 'germartistmai@gmail.com', '8914574423');

INSERT INTO artist VALUES (891,'Pablo','Southerland','New York', 'NY', 'US', 'jennysellsart@gmail.com', '6784543234');

INSERT INTO artist VALUES (783,'Jean-Antoine','Watteau','Strasbourg', 'GES', 'FRA', '', '');

INSERT INTO artist VALUES (549,'Paul','Dillet','Arlon', 'WLX', 'BE', 'randomgall@gmail.com', '8893455678');

INSERT INTO artist VALUES (130,'Albert','Neuhuys','Istanbul', '34', 'TR', 'albertsales34@gmail.com', '1134373422');

INSERT INTO artwork VALUES (4560,'Dawn Lol', 'uhm', 'Oldendel', '2019-12-17', 'Medium', 2345,'Painting', 'Oil on Canvas','1900s', 2016);

INSERT INTO artwork VALUES (7891,'Smith Jones', 'Bleu', 'Tulipshaw', '2019-2-08', 'Large', 1356,'Painting', 'Acrylic on Canvas','1950-Present', 2019);

INSERT INTO artwork VALUES (2345,'Lee Li', 'Oythus', 'Relicmilie', '2019-2-08', 'Large', 1098,'Painting', 'Watercolor on Hot Pressed Paper','1950-Present', 2003);

INSERT INTO artwork VALUES (6778,'Sally Hong', 'Rackaro', 'Skipleaf', '2019-3-18', 'Large', 5000,'Statue', 'Copper','1950-Present', 2006);

INSERT INTO artwork VALUES (8674,'James Lone', 'Lenveyas', 'Leafmill', '2019-3-18', 'Large', 5670,'Statue', 'Copper','1950-Present', 2019);

INSERT INTO artwork VALUES (1049,'Leo Da Vinci','Amour', 'Leafmill', '2019-7-20', 'Small', 9670,'Painting', 'Oil on Canvas','Renaissance', 1490);

INSERT INTO artwork VALUES (7369,'Leonardo Capablanca','Belle Nuit', 'Oldendel', '2019-2-25', 'Medium', 5010,'Painting', 'Tempera','1950-Present', 2000);

INSERT INTO artwork VALUES (9730,'Kyle Denise','La Ciel', 'Oldendel', '2019-10-01', 'Large', 9080,'Statue', 'Marbel','1950-Present', 1990);

INSERT INTO artwork VALUES (4999,'Taylor Octavian','War and Peace', 'Tulipshaw', '2019-12-19', 'Medium', 7000,'Sculpture', 'Marbel','1900s', 1940);

INSERT INTO artwork VALUES (9764,'Sophia Charlemagne','Lords', 'Tulipshaw', '2019-12-19', 'Large', 2323,'Painting', 'Pastel','Neoclassical', 1803);

INSERT INTO artwork VALUES (6892,'Anthony Constantinous','Prince Augusto', 'Tulipshaw', '2019-12-19', 'Large', 1089,'Painting', 'Acrylic on Canvas','1900s', 1949);

INSERT INTO artwork VALUES (3436,'Kendrick Van Gogh','Silent Night', 'Relicmilie', '2019-2-25', 'Small', 7171,'Sculpture', 'Ivory','Medieval', 1343);

INSERT INTO artwork VALUES (5313,'Louis Picasso','Outers', 'Relicmilie', '2019-2-25', 'Small', 8899,'Sculpture', 'Marbel','Romantic', 1842);

INSERT INTO artwork VALUES (4335,'Seagle Aivazovsky','Ganity', 'Oldendel', '2019-2-25', 'Small', 3899,'Painting', 'Oil on Canvas','Renaissance', 1560);

INSERT INTO artwork VALUES (6604,'Tyler Willendorf','Glory', 'Oldendel', '2019-2-25', 'Small', 6899,'Sculpture', 'Marbel','1950-Present', 1954);

INSERT INTO artwork VALUES (1340,'Pablo Southerland','Dimension', 'Oldendel', '2019-2-25', 'Small', 9999,'Sculpture', 'Marbel','1950-Present', 2007);

INSERT INTO artwork VALUES (4949,'Jean-Antoine Watteau','Calm', 'Skipleaf', '2019-4-12', 'Large', 4591,'Painting', 'Oil on Canvas','Baroque', 1940);

INSERT INTO artwork VALUES (9814,'Paul Dillet','Calamity', 'Skipleaf', '2019-4-13', 'Large', 6735,'Painting', 'Acrylic on Canvas','1950-Present', 1967);

INSERT INTO artwork VALUES (6645,'Albert Neuhuys','Reflection', 'Relicmilie', '2019-2-25', 'Small', 8899,'Sculpture', 'Gypsum','Realistic', 1895);

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Dawn')), 23206,'Oldendel', '2020-1-08','2020-1-15');

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Smith')), 90865,'Tulipshaw', '2019-2-08','2019-2-15');

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Lee')), 78964,'Relicmilie', '2019-2-08','2019-2-20');

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Sally')), 14532,'Skipleaf', '2019-3-18','2019-3-25');

```
INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'James')), 23256,'Leafmill', '2019-3-18','2019-3-25');

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Leo')), 23256,'Leafmill', '2019-3-18','2019-3-25');

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Leonardo')), 23206,'Oldendel', '2020-1-08','2020-1-15');

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Kyle')), 23206,'Oldendel', '2020-1-08','2020-1-15');

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Taylor')), 90865,'Tulipshaw', '2019-2-08','2019-2-15');

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Sophia')), 90865,'Tulipshaw', '2019-2-08','2019-2-15');

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Anthony')), 90865,'Tulipshaw', '2019-2-08','2019-2-15');

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Kendrick')), 78964,'Relicmilie', '2019-2-08','2019-2-20');

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Louis')), 78964,'Relicmilie', '2019-2-08','2019-2-20');

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Seagle')), 23206,'Oldendel', '2020-1-08','2020-1-15');
```

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Tyler')), 23206,'Oldendel', '2020-1-08','2020-1-15');

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Pablo')), 23206,'Oldendel', '2020-1-08','2020-1-15');

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Jean-Antoine')), 14532,'Skipleaf', '2019-3-18','2019-3-25');

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Paul')), 14532,'Skipleaf', '2019-3-18','2019-3-25');

INSERT INTO exhibition

VALUES ((select ArtistID from artist where ArtistID = (select ArtistID from artist where Firstname = 'Albert')), 78964,'Relicmilie', '2019-2-08','2019-2-20');

INSERT INTO customer VALUES (00982,'Tom','Nook', '123 Sesame Street', 'New York','09303','USA','randomemail@yahoo.com','212-234-3344');

INSERT INTO customer VALUES (00983,'Timmy','Nook', '124 Sesame Street', 'New York','09303','USA','randomemail1@yahoo.com','212-456-3646');

INSERT INTO customer VALUES (00984,'Tommy','Nook', '125 Sesame Street', 'New York','09303','USA','randomemail2@yahoo.com','212-789-3567');

INSERT INTO customer VALUES (00985,'Thomas','Nook', '126 Buda Street', 'New Orleans','90210','USA','thomnook1@gmail.com','312-264-2354');

INSERT INTO customer VALUES (00986,'Tome','Nook', '127 Buda Street', 'New Orleans','90210','USA','tomnook@gmail.com','312-316-6666');

INSERT INTO customer VALUES (00987,'Timothy','Nook', '128 Buda Street', 'New Orleans','90210','USA','timnook2@gmail.com','312-387-3264');

INSERT INTO art_order VALUES (100987,(select CustomerID from customer where CustomerID=00982),(select ArtworkID from artwork where ArtworkID=4560),'2019-4-

25',3,'2019-4-30',1,'Rental',(select PaymentID from payment where PaymentID=200987));

INSERT INTO art_order VALUES (100988,(select CustomerID from customer where CustomerID=00983),(select ArtworkID from artwork where ArtworkID=6778),'2019-4-

30',1,'2019-5-05',1,'Rental',(select PaymentID from payment where PaymentID=200988));

INSERT INTO art_order VALUES (100989,(select CustomerID from customer where CustomerID=00984),(select ArtworkID from artwork where ArtworkID=8674),'2019-5-

05',1,'2019-05-10',1,'Rental',(select PaymentID from payment where PaymentID=200989));

INSERT INTO art_order VALUES (100990,(select CustomerID from customer where CustomerID=00985),(select ArtworkID from artwork where ArtworkID=9764),'2019-2-

20',5,'2019-5-01',1,'Rental',(select PaymentID from payment where PaymentID=200990));

INSERT INTO art_order VALUES (100991,(select CustomerID from customer where CustomerID=00986),(select ArtworkID from artwork where ArtworkID=3436),'2019-3-

18',2,'2019-6-15',1,'Rental',(select PaymentID from payment where PaymentID=200991));

INSERT INTO art_order VALUES (100992,(select CustomerID from customer where CustomerID=00987),(select ArtworkID from artwork where ArtworkID=4949),'2019-4-

20',1,'2019-10-10',1,'Rental',(select PaymentID from payment where PaymentID =200992));

INSERT INTO payment VALUES (200987,(select OrderID from art_order where OrderID=100987),(select CustomerID from customer where CustomerID =00982),250,'2019-5-01');

INSERT INTO payment VALUES (200988,(select OrderID from art_order where OrderID=100988),(select CustomerID from customer where CustomerID =00983),450,'2019-4-29');

INSERT INTO payment VALUES (200989,(select OrderID from art_order where OrderID=100989),(select CustomerID from customer where CustomerID =00984),450,'2019-4-30');

INSERT INTO payment VALUES (200990,(select OrderID from art_order where OrderID=100990),(select CustomerID from customer where CustomerID =00985),150,'2019-5-01');

INSERT INTO payment VALUES (200991,(select OrderID from art_order where OrderID=100991),(select CustomerID from customer where CustomerID =00986),1000,'2019-4-29');

INSERT INTO payment VALUES (200992,(select OrderID from art_order where OrderID=100992),(select CustomerID from customer where CustomerID =00987),1249,'2019-4-30');

UPDATE art_order SET PaymentID = 200989 WHERE CustomerID = 00982;

UPDATE art_order SET PaymentID = 200987 WHERE CustomerID = 00983;

UPDATE art_order SET PaymentID = 200988 WHERE CustomerID = 00984;

UPDATE art_order SET PaymentID = 200990 WHERE CustomerID = 00985;

UPDATE art_order SET PaymentID = 200991 WHERE CustomerID = 00986;

UPDATE art_order SET PaymentID = 200992 WHERE CustomerID = 00987;

exit;

EOF

```
SQL> SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL> SQL>
1 row updated.

SQL>
1 row updated.

SQL>
1 row updated.

SQL>
1 row updated.

SQL>
1 row updated.

SQL>
1 row updated.
```

DROP TABLES:

```
#!/bin/sh

#export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib

sqlplus64
"a84nguye/05271893@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerso
n.ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF

DROP TABLE ART_ORDER CASCADE CONSTRAINTS;

DROP TABLE ARTIST CASCADE CONSTRAINTS;

DROP TABLE ARTWORK CASCADE CONSTRAINTS;

DROP TABLE CUSTOMER CASCADE CONSTRAINTS;

DROP TABLE EXHIBITION CASCADE CONSTRAINTS;

DROP TABLE PAYMENT CASCADE CONSTRAINTS;

exit;

EOF
```

```
SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.
```

EXITING:

```
-----------------------------------------------------------------
  M)  View Manual

  1)  Drop Tables
  2)  Create Tables
  3)  Populate Tables
  4)  Query Tables

      a)  Customer and Payments
      b)  Artist and Artwork
      c)  Exhibition:Artwork and Corresponding Art in EXBID23206
      d)  Display Artists that have Small Sculptures
      e)  Average Price of Sculptures
      f)  Number of Rentals
      g)  Customers Using a Gmail Account
      h)  Artists with Artwork Belonging in 1950s NOT made w/ Marble
      i)  Customers Living on Sesame Street and Rental Orders

  X)  Force/Stop/Kill Oracle DB

  E)  End/Exit
Choose:
E
Bye
a84nguye@europa:~/Cps510Group8$
```

QUERY OUTLINE:

```
#!/bin/sh

#export LD_LIBRARY_PATH=/usr/lib/oracle/12.1/client64/lib

sqlplus64
"a84nguye/05271893@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=oracle.scs.ryerso
n.ca)(Port=1521))(CONNECT_DATA=(SID=orcl)))" <<EOF


SELECT c.FIRSTNAME, c.LASTNAME, p.PAYMENTAMOUNT

FROM art_order o, customer c, payment p

WHERE c.CUSTOMERID = o.CUSTOMERID AND o.PAYMENTID = p.PAYMENTID

ORDER BY PAYMENTAMOUNT DESC;


SELECT a.NAMEART, a.TYPEART

FROM artwork a, art_order o

WHERE o.ARTWORKID = a.ARTWORKID;
```

```
SELECT DISTINCT a.ARTIST, a.NAMEART, a.PRICE FROM
Artwork a, Exhibition e
WHERE e.EXHIBITIONID = 23206 AND a.EXHIBITION = e.GALLERY_TITLE;

SELECT DISTINCT Artist, SizeArt,TypeArt
FROM artwork a
WHERE EXISTS
   (SELECT *
    FROM  artwork b
    WHERE a.TypeArt = 'Sculpture'
         and a.SizeArt = 'Small');

SELECT 'Average Price of Sculptures', AVG(Price)
FROM  artwork
WHERE TypeArt = 'Sculpture';

SELECT Order_Type, COUNT(*) AS NoOfRentals
FROM  art_order WHERE  OrderID IS NOT NULL
GROUP BY Order_Type;

SELECT Email, Count(*) AS CustomersWithGmail
FROM customer
WHERE Email LIKE '%gmail%'
GROUP BY Email;

SELECT a.nameart, a.artist, a.mediumart FROM artwork a
```

```
WHERE a.timeperiod = '1950-Present'
MINUS(SELECT a.nameart, a.artist, a.mediumart FROM artwork a
WHERE a.mediumart = 'Marble')

SELECT CustomerID, Address
FROM customer
WHERE address LIKE '%Sesame%'
UNION ALL
SELECT OrderID, Order_Type FROM art_order
WHERE Order_Type='Rental';

exit;
EOF
```

6. Database Normalization/ Functional Dependencies

Artist

ArtistID#	FirstName	LastName	City	Region	Country	Email	PhoneNumber

ArtistId → FirstName, LastName, City, Region, Country, Email, Phone Number

Country → Region, City

PhoneNumber → ArtistId, FirstName, LastName, City, Region, Country, Email

Email → ArtistId, FirstName, LastName, City, Region, Country, Email, PhoneNumber

Artwork

ArtworkID#	ArtistID	NameArt	Exhibition	DateAdded	SizeArt	Price	TypeArt	MediumArt	TimePeriod	YearCreated

ArtworkID → ArtistID, NameArt, Exhibition, DateAdded, SizeArt, Price, TypeArt, MediumArt, TimePeriod, YearCreated

MediumArt -> TypeArt

YearCreated → TimePeriod

Exhibition

ExhibitionID#	Gallery_Name	StartDate	EndDate

ExhibitionID → Gallery_Title, StartDate, EndDate

Customer

CustomerID#	FirstName	LastName	Address	City	PostalCode	Country	Email	PhoneNumber

CustomerID → FirstName, LastName, Address, City, PostalCode, Country, Email, PhoneNumber

Country → City, Address, PostalCode

Email → CustomerID, FirstName, LastName, Address, City, PostalCode, Country, PhoneNumber

PhoneNumber → CustomerID, FirstName, LastName, Address, City, PostalCode, Country, Email

ArtOrder

OrderID#	CustomerID	ArtworkID	RentalStartDate	RentalLength	DeliveryDate	OrderType

OrderID → CustomerID, ArtworkID, RentalStartDate, RentalLength, DeliveryDate, OrderType

RentalStartDate, RentalLength → OrderType

Payment

PaymentID#	OrderID	CustomerID	PaymentAmount	PaymentDate

PaymentID- > OrderID, CustomerID, PaymentAmount, PaymentDate

OrderID → PaymentID, CustomerID, PaymentAmount, PaymentDate

3NF/BCNF for All Tables:

\

Artist

1NF (Key is ArtistID)

ArtistID#	FirstName	LastName	City	Region	Country	Email	PhoneNumber

BCNF

Artist

ArtistId → (FirstName,LastName,Email,PhoneNumber, IDLocation)

Location

IDLocation → (City,Region,Country)

3NF

Artist

ArtistId → (FirstName,LastName, IDLocation)

ArtistID → (Email,PhoneNumber)

Location

IDLocation → (City,Region,Country)

Artwork

1NF (Key is ArtworkID)

ArtworkID#	Artist	NameArt	Exhibition	DateAdded	SizeArt	Price	TypeArt	MediumArt	TimePeriod	YearCreated

BCNF

Artwork

ArtworkID → (Artist ,NameArt, Exhibition, DateAdded, SizeArt, Price, IDType)

ArtDescription

IDType → (TypeArt, MediumArt, TimePeriod, YearCreated)

3NF

Artwork

ArtworkID → (NameArt, Exhibition, DateAdded, SizeArt, Price, IDType)

ArtworkID → (Artist)

ArtDescription

IDType → (TypeArt, MediumArt, TimePeriod, YearCreated)

Exhibition

1NF (Key is ExhibitionID)

ExhibitionID#	Gallery_Title	StartDate	EndDate

BCNF

Exhibition

ExhibitionID → (Gallery_Title)

Date

ID_Date -> (StartDate,EndDate)

3NF

Exhibition

ExhibitionID → (Gallery_Title)

ExhibitID -> GalleryTitle

ExhibitID -> (StartDate,EndDate)

Customer

1NF (Key is CustomerID)

CustomerID#	FirstName	LastName	Address	City	PostalCode	Country	Email	PhoneNumber

BCNF

CustomerID → (FirstName,LastName, Email, PhoneNumber, AddressID)

AddressID → (Address,City,PostalCode,Country)

3NF

CustomerID → (FirstName,LastName, AddressID)

CustomerID → (Email, PhoneNumber)

AddressID → (Address,City,PostalCode,Country)

ArtOrder

1NF (Key is OrderID)

OrderID#	CustomerID	ArtworkID	RentalStartDate	RentalLength	DeliveryDate	NoPieces	OrderType

BCNF

OrderID → (RentalStartDate, RentalLength, DeliveryDate, NoPieces, OrderType)

OrderID → (CustomerID, ArtworkID)

3NF

OrderID → (RentalStartDate, RentalLength, DeliveryDate, NoPieces, OrderType)
OrderID → (CustomerID, ArtworkID)

Payment

1NF (Key is PaymentID)

PaymentID#	OrderID	CustomerID	PaymentAmount	DatePayment

BCNF

PaymentID → (OrderID, CustomerID)

PaymentID → (PaymentAmount, DatePayment)

3NF

PaymentID → (OrderID, CustomerID)

PaymentID → (PaymentAmount, DatePayment)

7. 3NF Normalization Diagrams

Assignment 7: Make all tables into 3NF. Add more functional dependency (for example, by looking at the foreign keys) and make tables into lower normal forms. Use the diagram method to change them back into 3NF.

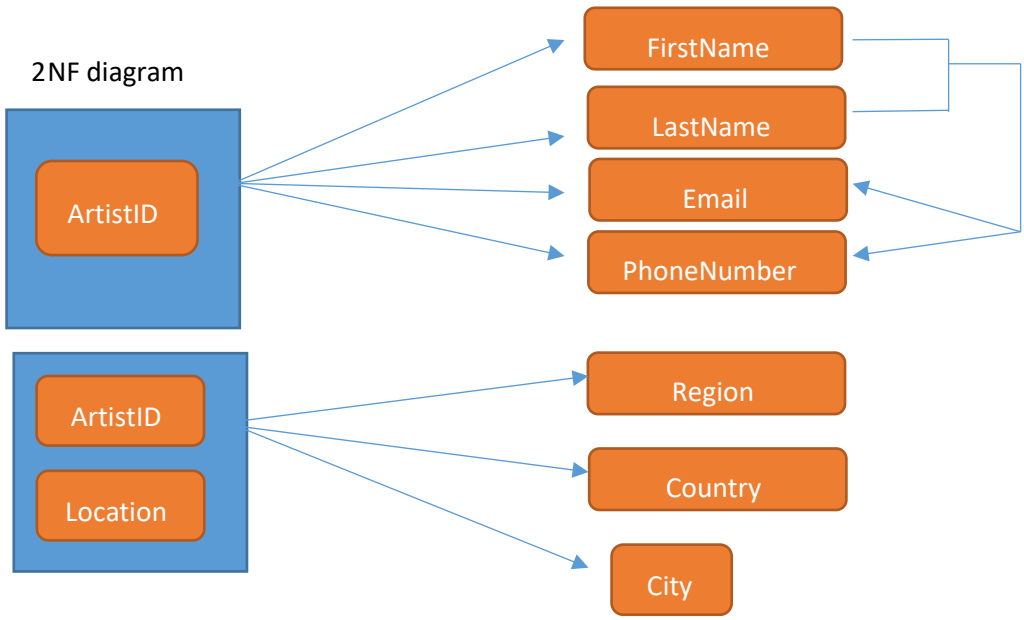
Table **Artist** = (ArtistID#, FirstName, LastName, City, Region, Country, Email, PhoneNumber)

1NF

Table **Artist** = (ArtistID#, Location#,FirstName, LastName, City, Region, Country, Email, PhoneNumber) 2NF **FD:**

ArtistId → FirstName, LastName → Email, PhoneNumber

ArtistId,Location → City, Region, Country



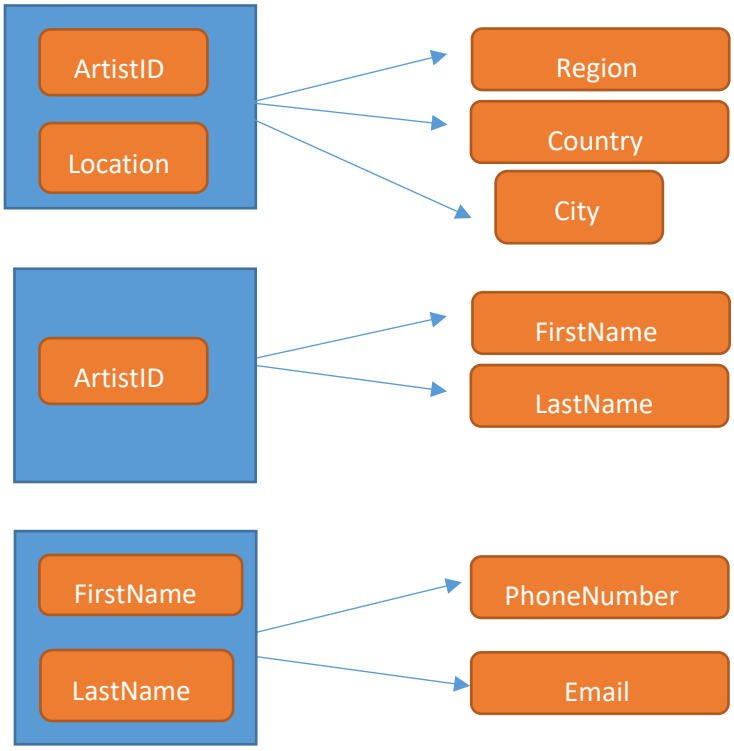
3NF diagram

Composite Key = FirstName, LastName **FD:**

AtristID, Location→ Region, Country, City

ArtistId → FirstName, LastName

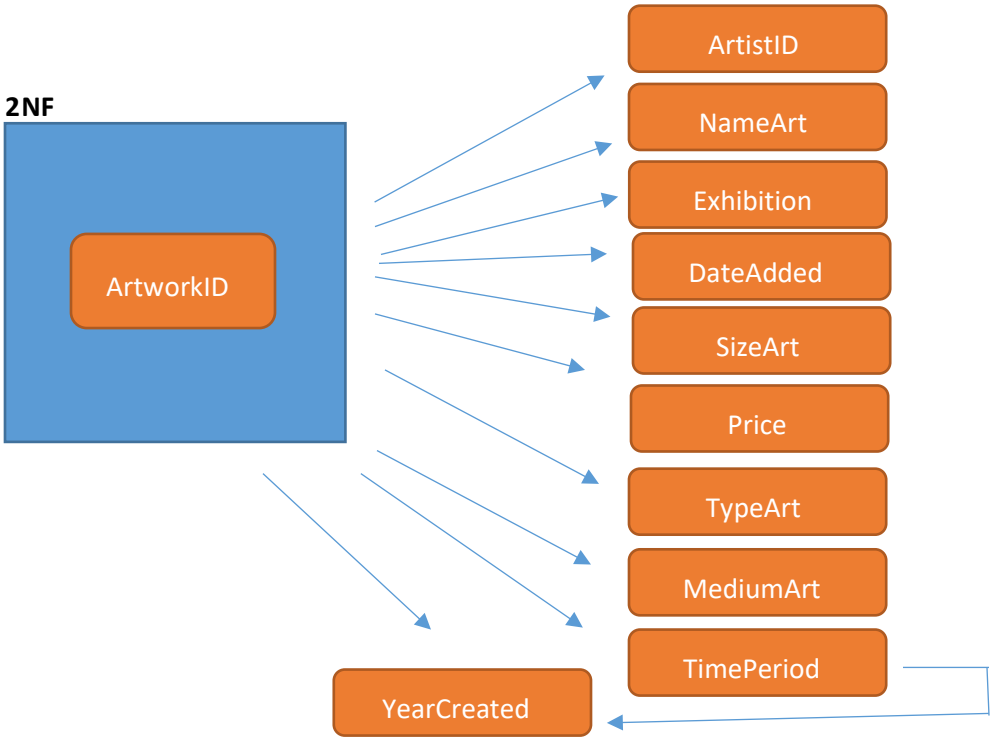
FirstName, LastName → Email, PhoneNumber



FD:

ArtworkID → ArtistID, NameArt, Exhibition, DateAdded, SizeArt, Price, TypeArt, MediumArt, TimePeriod, YearCreated

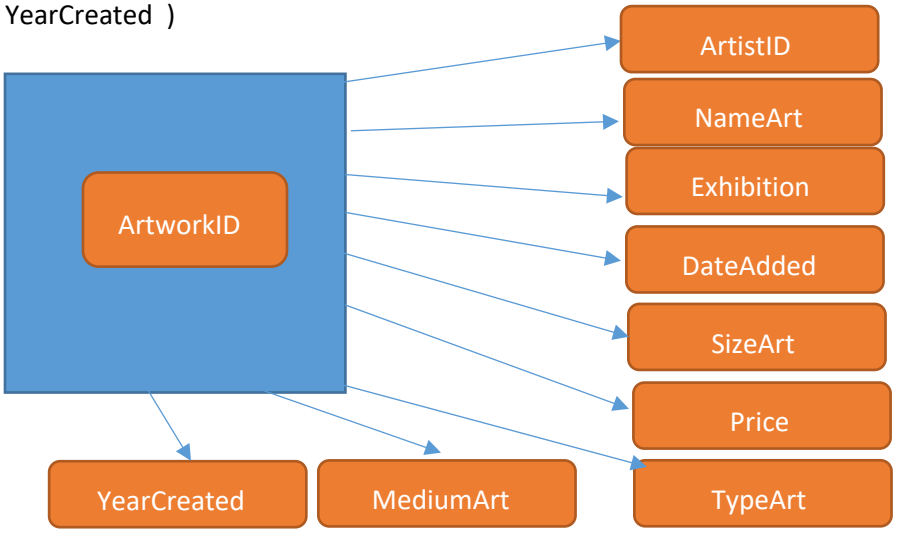
YearCreated → TimePeriod



This table is in 2NF because it exhibits a transitive dependency

3NF

ArtworkID → (ArtistID, NameArt, Exhibition, DateAdded, SizeArt, Price, TypeArt, MediumArt, YearCreated)



YearCreated → (TimePeriod)

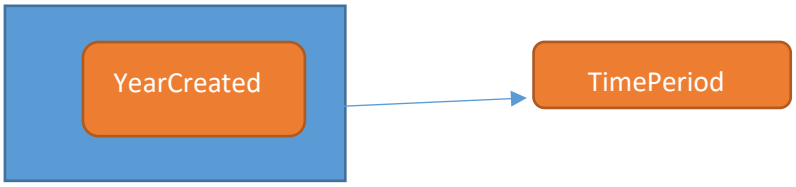


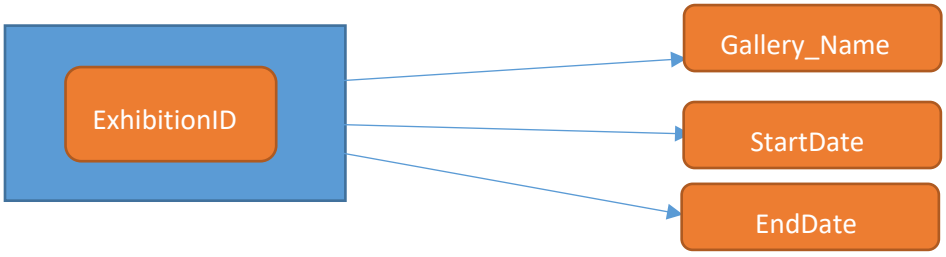
Table **Exhibition** = (<u>ExhibitionID</u>#, Gallery_Name, StartDate, EndDate)

FD:

ExhibitionID → Gallery_Name, StartDate, EndDate

2NF

ExhibitionID → Gallery_Name, StartDate, EndDate



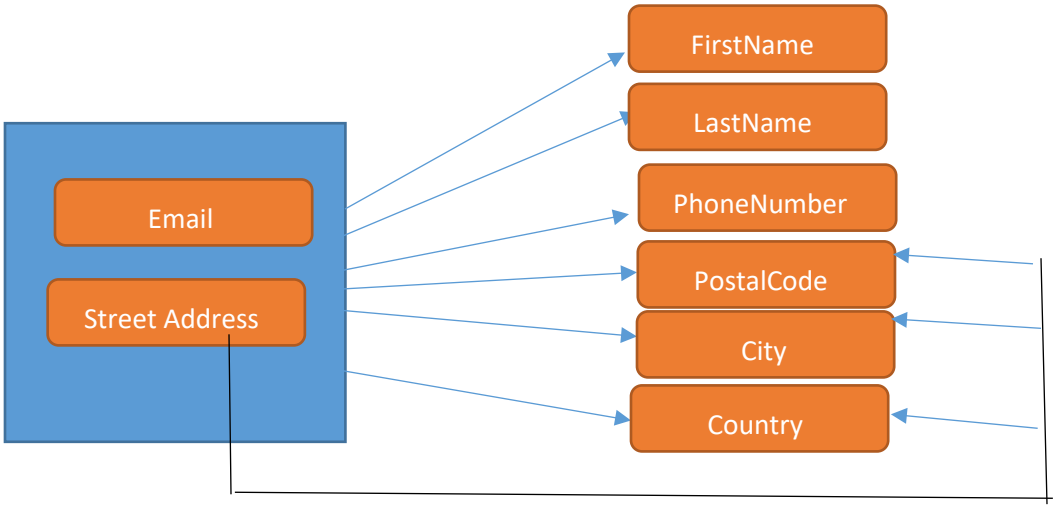
3NF

Same as 2NF, already in 3NF form

Table **Customer** = (<u>PhoneNumber, Email</u>, FirstName, LastName, StreetAddress, City, PostalCode, Country)

FD:
PhoneNumber, Email → FirstName, LastName, StreetAddress, City, PostalCode, Country StreetAddress → PostalCode, City, Country

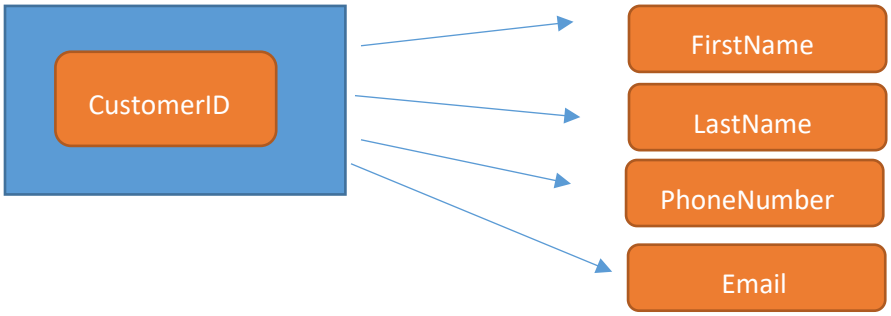
1NF Diagram



This table is 1NF because records are unique, and each cell contains a single value, but we have a non-fully functional dependency.

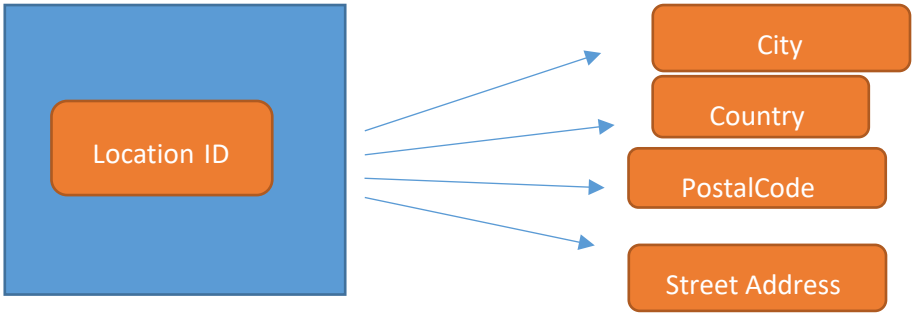
2NF diagram

CustomerID → (FirstName, LastName, PhoneNumber, Email)



CustomerID → LocationID

LocationID → City, Country, PostalCode

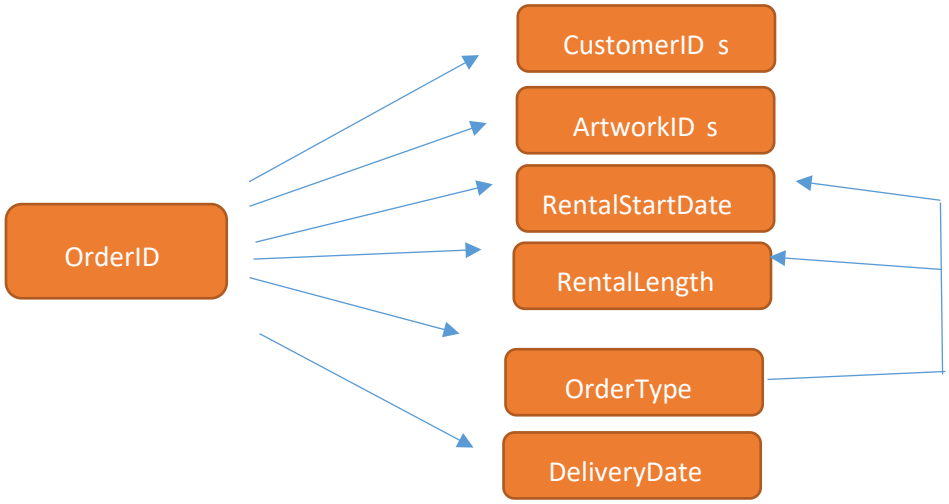


These tables are now also in 3NF.

Table **ArtOrder** = (<u>OrderID#</u>, CustomerID, ArtworkIDs, RentalStartDate, RentalLength, DeliveryDate, OrderType)

FD:
OrderID → CustomerID, ArtworkIDs, RentalStartDate, RentalLength, DeliveryDate, OrderType
RentalStartDate, RentalLength → OrderType

1NF Diagram

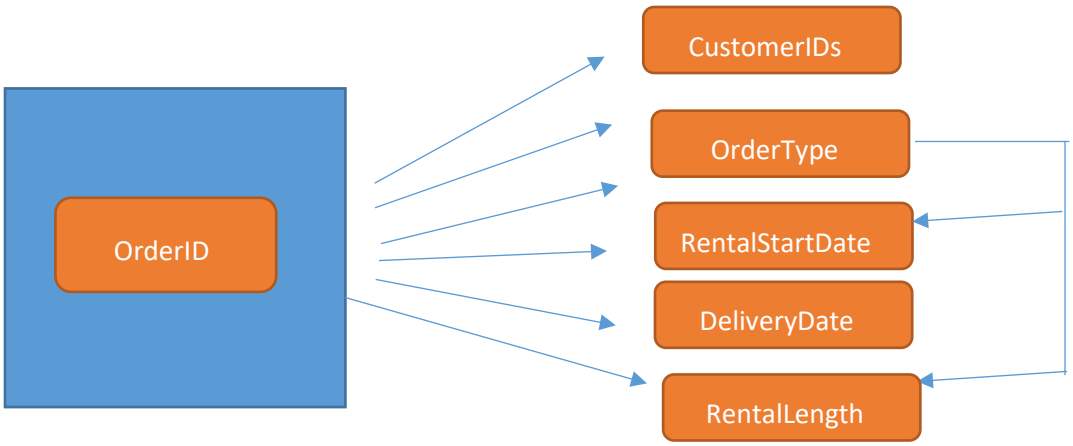


This table is 1NF because the orderID isn't necessarily unique. **2NF**

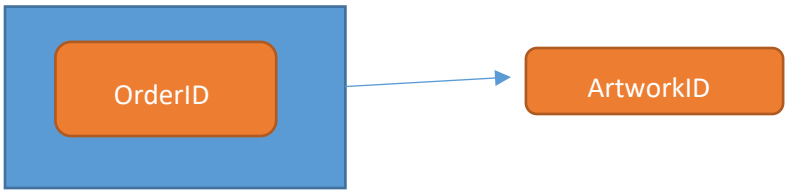
Table **ArtOrder** = (<u>OrderID#</u>, CustomerID, RentalStartDate, RentalLength, DeliveryDate, OrderType)

Table **ArtworkPurchased** = (<u>OrderID</u>, ArtworkID)

FD:
OrderID → CustomerID, RentalStartDate, RentalLength, DeliveryDate, OrderType
RentalStartDate, RentalLength → OrderType



OrderID → ArtworkID



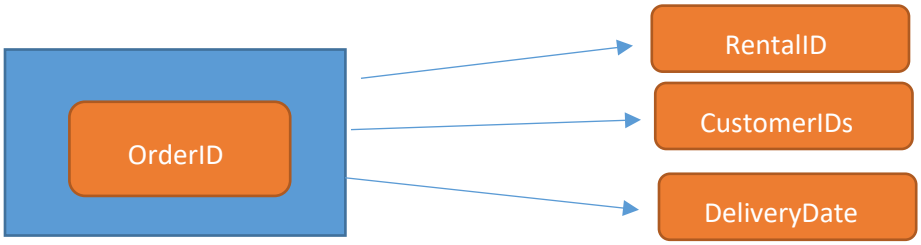
3NF

Table **ArtOrder** = (<u>OrderID#</u>, CustomerID, DeliveryDate, RentalID)

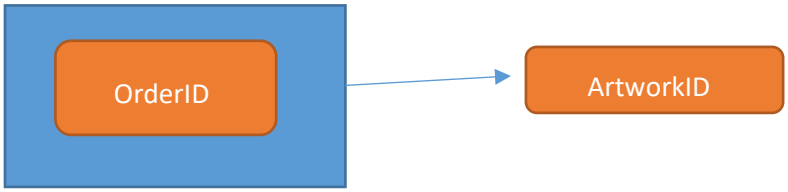
Table **ArtworkPurchased** = (<u>OrderID</u>, ArtworkID)

Table **RentalInfo** = (RentalID, RentalStartDate, RentalLength)

FD:
OrderID → CustomerID, RentalStartDate, RentalLength, DeliveryDate, OrderType
RentalStartDate, RentalLength → OrderType



OrderID → ArtworkID



RentalID → RentalStartDate, RentalLength

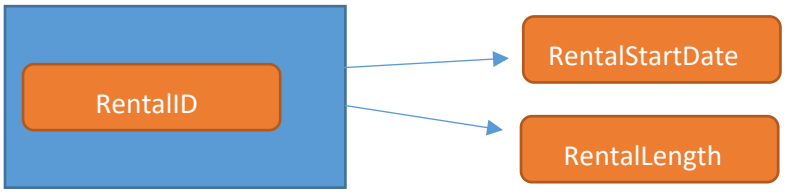


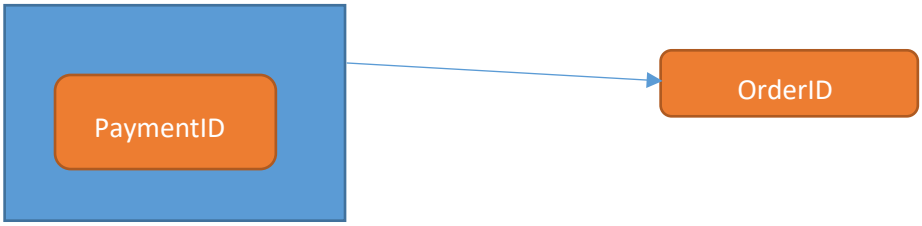
Table **Payment** = (<u>PaymentID#</u>, OrderID, CustomerID, PaymentAmount, PaymentDate)

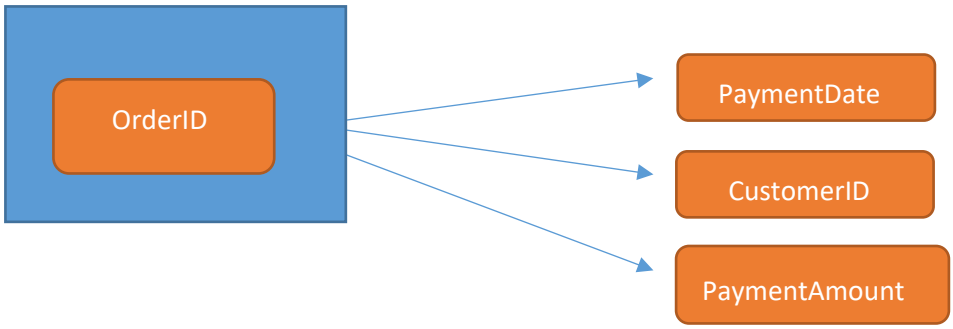
FD:
PaymentID → OrderID, CustomerID, PaymentAmount, PaymentDate

3NF

PaymentID → OrderID

OrderID → CustomerID, PaymentAmount, PaymentDate



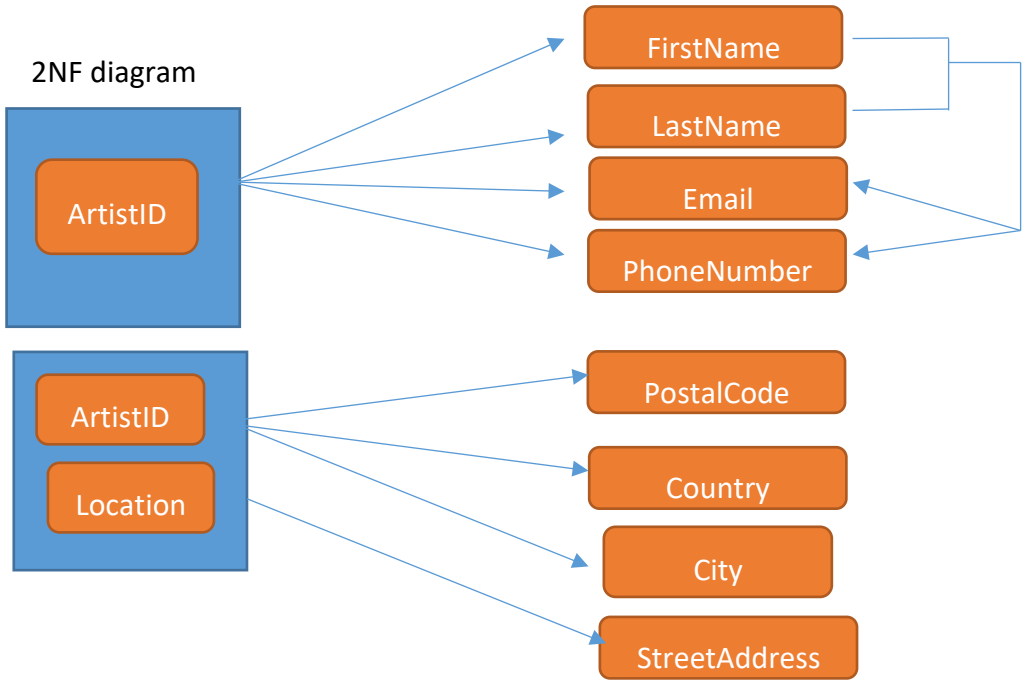


8. 3NF Normalization/ BCNF with Algorithm

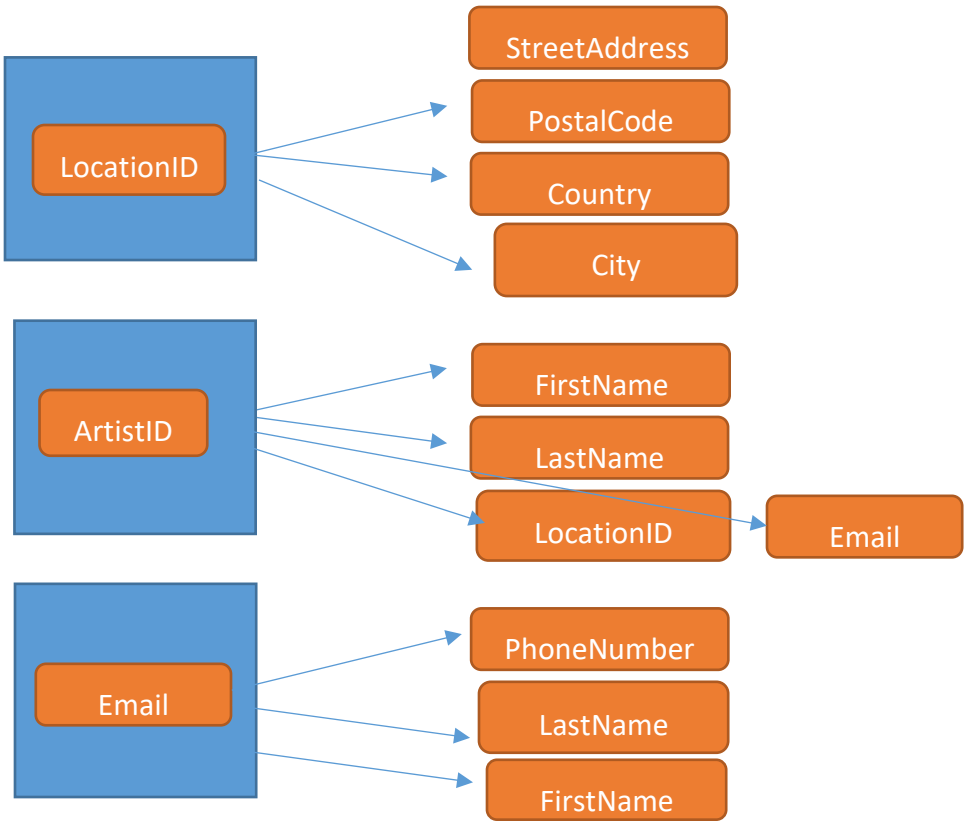
Table **Artist** = (ArtistID#, FirstName, LastName, City, PostalCode, StreetAddress, Country, Email, PhoneNumber)

1NF
Table **Artist** = (ArtistID#, Location#, FirstName, LastName, City, Region, Country, Email, PhoneNumber)

2NF
We decompose into 2 tables: Artist and ArtistLocation **FD:**
ArtistId → FirstName, LastName → Email, PhoneNumber
ArtistId, Location → City, PostalCode, StreetAddress, Country



3NF diagram We
have 3 tables:
Table **Location** = (LocationID, City, PostalCode, StreetAddress, Country)
Table **Artist** = (ArtistID#, FirstName, LastName)
Table **ArtistContact** = (FirstName, LastName, Email, PhoneNumber)



<u>BCNF Algorithm</u>

Table Location

Composite Key (to Uniquely Identify):{ Location}

R(LocationID, City, PostalCode, StreetAddress, Country)

FD:

LocationID (A) →StreetAddress (B), Country(C), City(D), PostalCode(E)

StreetAddress (B), City(D), Country(C) → LocationID (A), PostalCode(E)

PostalCode(E) → City(D), Country(C)

R(A,B,C,D,E)

A→ B, C,D,E

B, C, D → A, E

E → D, C

Checking if left side is a key:

A+ = {A,B,C,D,E} is a key

B, C, D+ = {A,B,C,D,E} is a key

E+ → {E, D, C} is not a key

Hence we break the relationship into two: R1 and R2

R1(<u>LocationID</u>, StreetAddress, PostalCode)
R2 (<u>PostalCode</u>, City, Country)

Check it satisfies BCNF: FD:

LocationID (A) →StreetAddress (B), PostalCode(E)
PostalCode(E) → City(D), Country(C)

R(A,B,C,D,E)

A→ B, C,D,E

E → D, C

Checking if left side is a key:

A+ = {A,B,E} is a key

E+ → {E, D, C} is a key

Final BCNF schema:

R1(A, B, E)

R2 (E, D, C)

Therefore, this is in BCNF form.

Dependency Preserving: Yes

Table Artist
Composite Key (to Uniquely Identify): {ArtistID}

R(ArtistID, FirstName, LastName, Email)

FD:
ArtistID (A) → FirstName(B), LastName(C), Email(D)

Email (D) → FirstName(B), LastName(C), ArtistID (A)

R(A,B,C)

A → B, C, D

D → A, B, C

Checking if left side is a key: A+
= {A, B, C, D} is a key

D+ = {A, B, C, D} is a key

Final BCNF schema is

R = (A,B,C)

Dependency Preserving: Yes

Table **ArtistContact**
Composite Key (to Uniquely Identify):{ Email }
R(FirstName, LastName, Email, PhoneNumber)
FD:

Email(C), → FirstName (A), LastName (B), PhoneNumber(D)

PhoneNumber(D) → FirstName (A), LastName (B), Email(C)

R(A,B,C,D)

C → A, B, D
D → A, B, C
Checking if left side is a key:

C+ = {A,B,C,D} is a key

D+ = {A,B,C,D} is a key

Final BCNF schema: R(A,B,C,D)

Every determinant is a candidate key.

Therefore, this is in BCNF form.

Dependency Preserving: Yes

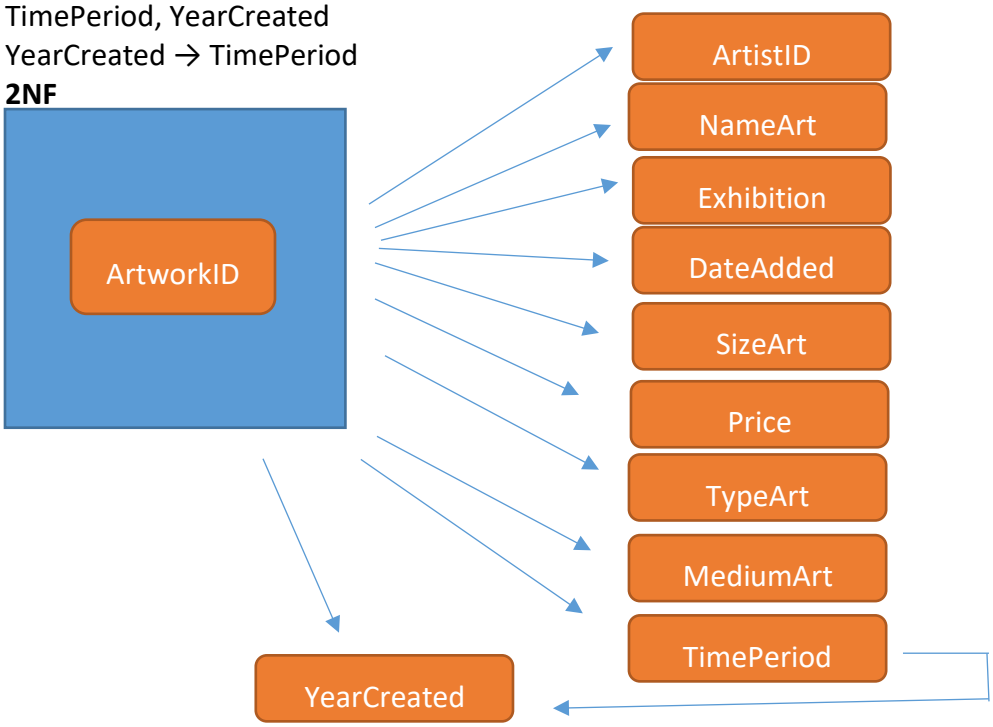
Table **Artwork**

FD:

ArtworkID → ArtistID, NameArt, Exhibition, DateAdded, SizeArt, Price, TypeArt, MediumArt, TimePeriod, YearCreated

YearCreated → TimePeriod

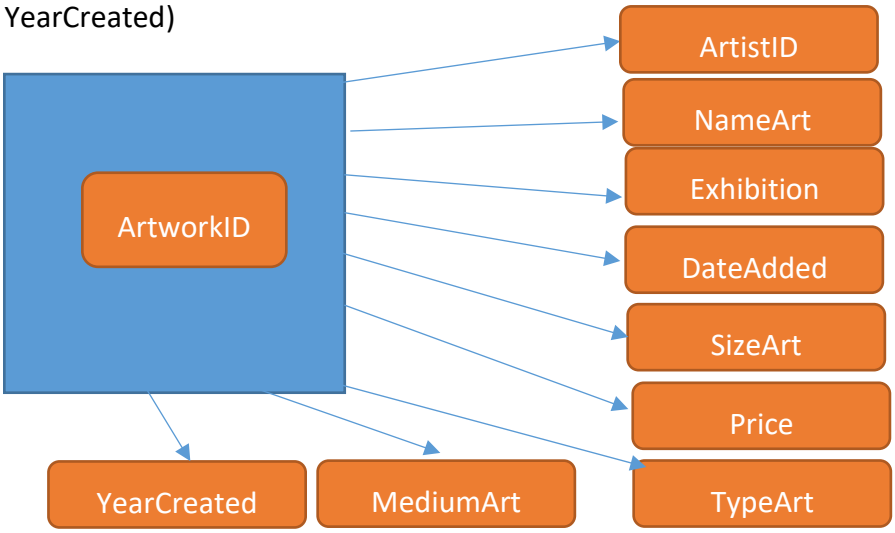
2NF



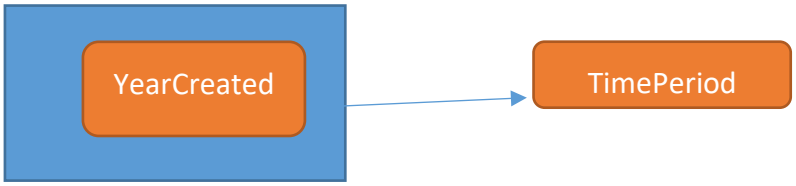
This table is in 2NF because it exhibits a transitive dependency

3NF

ArtworkID → (ArtistID, NameArt, Exhibition, DateAdded, SizeArt, Price, TypeArt, MediumArt, YearCreated)



YearCreated → (TimePeriod)



<u>BCNF Algorithm</u>

Table Artwork

Composite Key (to Uniquely Identify): {ArtworkID}

R(ArtworkID,ArtistID, NameArt, Exhibition, DateAdded, SizeArt, Price, TypeArt, MediumArt, YearCreated, TimePeriod)

FD:

ArtworkID(A)→ ArtistID(B), NameArt(C), Exhibition(D), DateAdded(E), SizeArt(F), Price(G), TypeArt(H), MediumArt(I), YearCreated(J)

R(A,B,C,D,E,F,G,H,I, J)

A→ B, C,D,E, F, G, H, I, J

Checking if left side is a key:

A+ = {A, B, C, D, E, F, G, H, I, J} is a key

Final BCNF schema: R(A,B,C,D,E,F,G,H,I, J) Therefore,
this is in BCNF form.

Dependency Preserving: Yes

Table CreationYear
Composite Key (to Uniquely Identify): { YearCreated }

R(YearCreated, TimePeriod)

FD:

YearCreated(J) →TimePeriod(K)

R(A,B)

A→ B

Checking if left side is a key:

A+ = {A, B} is a key

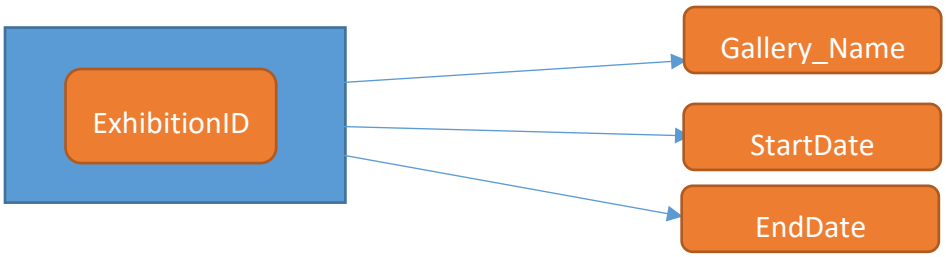
Final BCNF schema: R(A,B) Therefore,
this is in BCNF form.

Dependency Preserving: Yes

Table **Exhibition** = (ExhibitionID#, Gallery_Name, StartDate, EndDate)

FD:
ExhibitionID → Gallery_Name, StartDate, EndDate

2NF
ExhibitionID → Gallery_Name, StartDate, EndDate



3NF

Same as 2NF, already in 3NF form

<u>BCNF Algorithm</u>

Composite Key (to Uniquely Identify):{ExhibitionID}

R(ExhibitionID, Gallery_Name, StartDate, EndDate)

FD:

ExhibitionID(A) → Gallery_Name(B), StartDate(C), EndDate(D)

R(A,B,C,D)

A→ B, C,D

Checking if left side is a key:

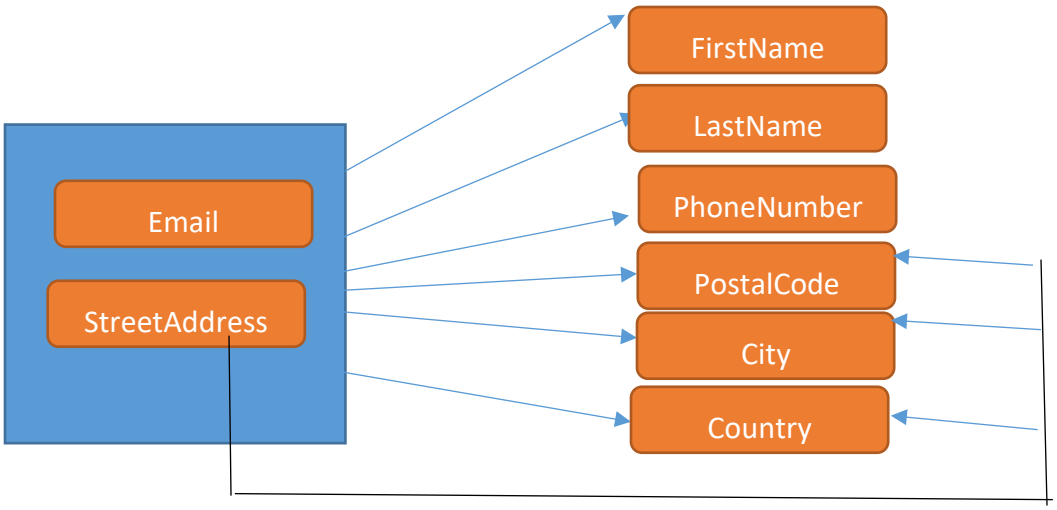
A+ = {A, B, C, D } is a key

Final BCNF schema: R(A,B,C,D) Therefore,
this is in BCNF form.

Dependency Preserving: Yes

<u>Table **Customer**</u> = (<u>PhoneNumber, Email</u>, FirstName, LastName, StreetAddress, City, PostalCode, Country) **FD:**
PhoneNumber, Email → FirstName, LastName, StreetAddress, City, PostalCode, Country
StreetAddress → PostalCode, City, Country

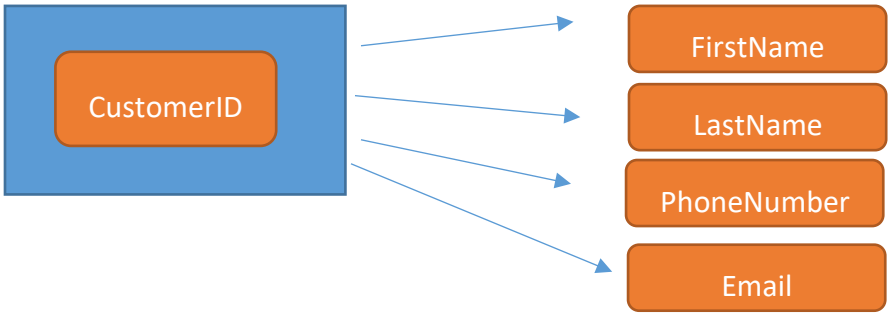
1NF Diagram



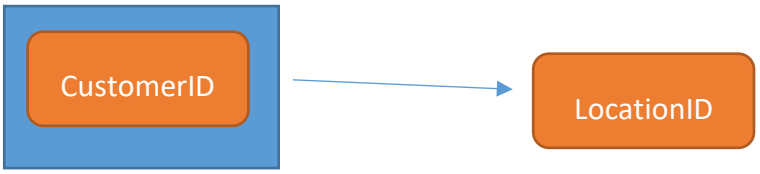
This table is 1NF because records are unique, and each cell contains a single value, but we have a non-fully functional dependency.

2NF diagram We
have 3 tables:
Table **Customer** = (CustomerID, FirstName, LastName, PhoneNumber, Email)
Table **CustomerLocation** = (CustomerID, LocationID)
Table **Location** = (LocationID, StreetAddress, City, PostalCode, Country)

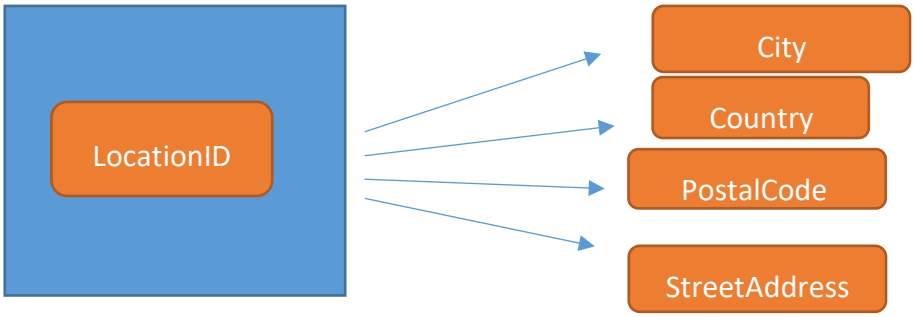
CustomerID → (FirstName, LastName, PhoneNumber, Email)



CustomerID → LocationID



LocationID → City, Country, PostalCode, StreetAddress



These tables are now also in 3NF.

<u>BCNF Algorithm</u>

Table Customer

Composite Key (to Uniquely Identify):{CustomerID}

R(CustomerID, FirstName, LastName, PhoneNumber, Email)

FD:

CustomerID(A) → FirstName (B), LastName (C), PhoneNumber (D), Email (E)

R(A,B,C,D,E)

A→ B, C, D, E

E→ A, B, C, D

D→ A, B, C, E

Checking if left side is a key:

A+ = {A, B, C, D, E} is a key

E+ = {A, B, C, D, E} is a key

D+ = {A, B, C, D, E} is a key

Every determinant is a candidate key. Therefore,
this is in BCNF form.

Final BCNF schema: R(A,B,C,D,E)

Dependency Preserving: Yes

Tabla CustomerLocation
Composite Key (to Uniquely Identify):{CustomerID}

R(CustomerID, LocationID)

FD:

CustomerID(A) → LocationID(B)

R(A,B)

A→ B

Checking if left side is a key:

A+ = {A, B} is a key

Final BCNF schema: R(A,B) Therefore,
this is in BCNF form.

Dependency Preserving: Yes

Table Location
Composite Key (to Uniquely Identify):{ LocationID}

R(LocationID, City, Country, Country, PostalCode, StreetAddress)

FD:

LocationID (A) → City(B), Country(C), PostalCode(D), StreetAddress(E)

R(A,B,C,D,E)

A→ B, C, D, E

Checking if left side is a key:

A+ = {A, B, C,D,E} is a key

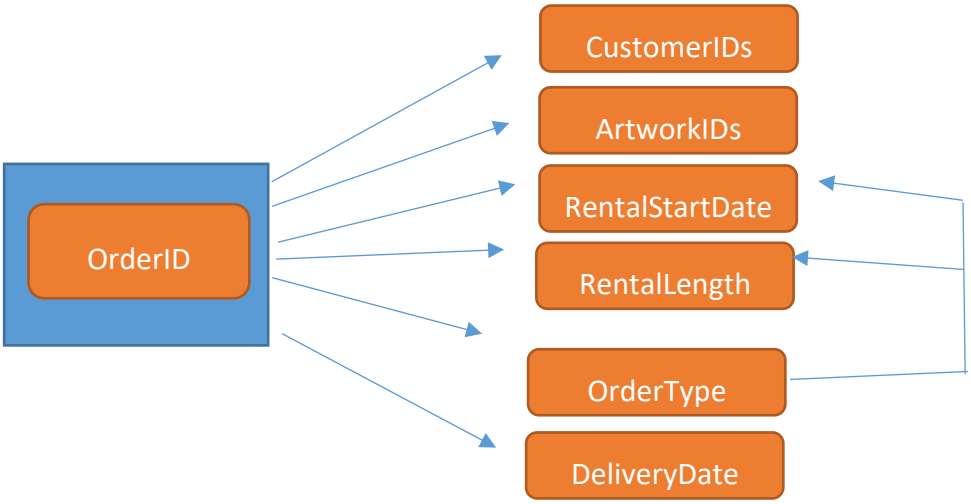
Final BCNF schema: R(A,B,C,D,E) Therefore,
this is in BCNF form.

Dependency Preserving: Yes

Table **ArtOrder** = (OrderID#, CustomerID, ArtworkIDs, RentalStartDate, RentalLength, DeliveryDate, OrderType)

FD:
OrderID → CustomerID, ArtworkIDs, RentalStartDate, RentalLength, DeliveryDate, OrderType
RentalStartDate, RentalLength → OrderType

1NF Diagram



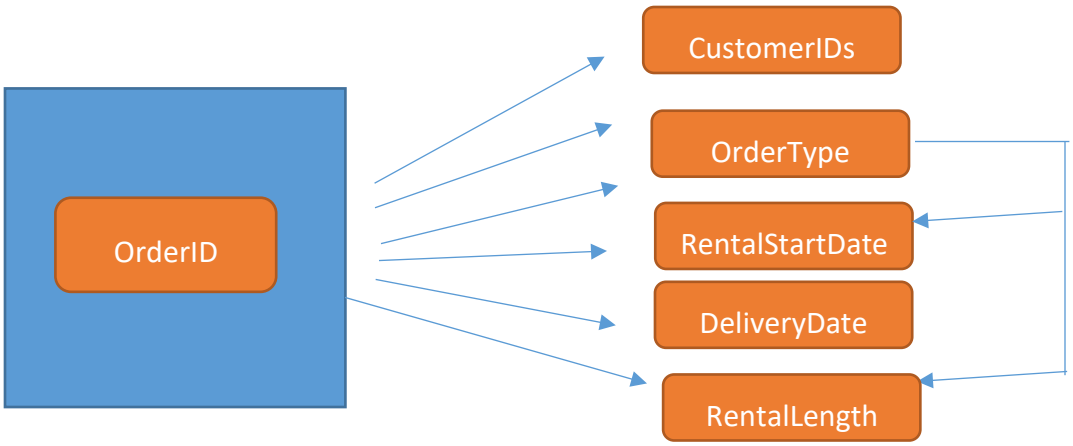
This table is 1NF because the orderID isn't necessarily unique.

2NF

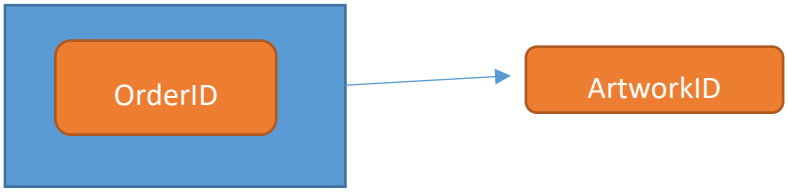
Table **ArtOrder** = (<u>OrderID#</u>, CustomerID, RentalStartDate, RentalLength, DeliveryDate, OrderType)

Table **ArtworkPurchased** = (<u>OrderID</u>, ArtworkID)

FD:
OrderID → CustomerID, RentalStartDate, RentalLength, DeliveryDate, OrderType
RentalStartDate, RentalLength → OrderType



OrderID → ArtworkID



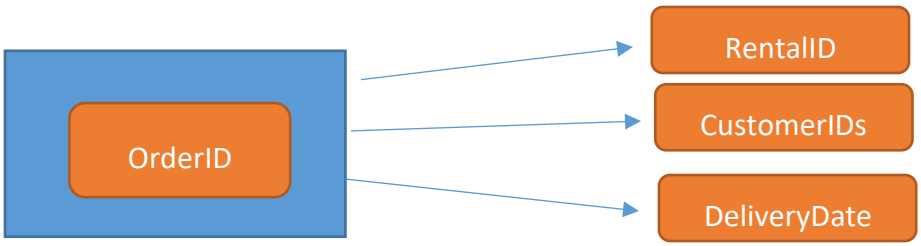
3NF

Table **ArtOrder** = (<u>OrderID#</u>, CustomerID, DeliveryDate, RentalID)

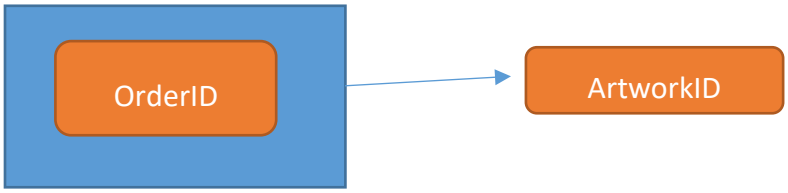
Table **ArtworkPurchased** = (<u>OrderID, ArtworkID</u>)

Table **RentalInfo** = (RentalID, RentalStartDate, RentalLength)

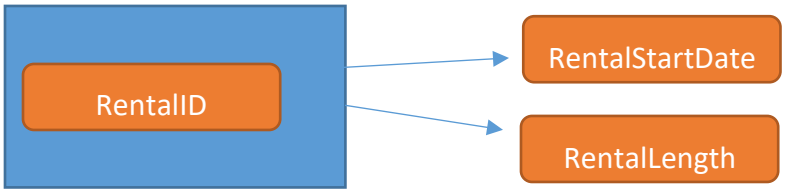
FD:
OrderID → CustomerID, RentalStartDate, RentalLength, DeliveryDate, OrderType



OrderID → ArtworkID



RentalID → RentalStartDate, RentalLength



<u>BCNF Algorithm</u>

Table ArtOrder:
Composite Key (to Uniquely Identify):{OrderID}

R(<u>OrderID#</u>, CustomerID, DeliveryDate, RentalID)

FD:

OrderID(A) → RentalID(B), CustomerID(C), DeliveryDate(D)

R(A,B,C,D)

A→ B, C,D

Checking if left side is a key:

A+ = {A, B, C, D} is a key

Final BCNF schema: R(A,B,C,D) Therefore,
this is in BCNF form.

Dependency Preserving: Yes

Table ArtworkPurchased:
Composite Key (to Uniquely Identify):{OrderID}

R(<u>OrderID#</u>, ArtworkID)

FD: OrderID(A) → ArtworkID(B)

R(A,B)

A→ B

Checking if left side is a key:

A+ = {A, B} is a key

Final BCNF schema: R(A,B) Therefore,
this is in BCNF form.

Dependency Preserving: Yes

Table RentalInfo:
Composite Key (to Uniquely Identify):{OrderID}

R(<u>RentalID, </u>RentalStartDate, RentalLength)

FD:

RentalID(A) → RentalStartDate(B), RentalLength(C)

R(A,B,C)

A→ B , C

Checking if left side is a key:

A+ = {A, B,C} is a key

Final BCNF schema: R(A,B,C) Therefore,
this is in BCNF form.

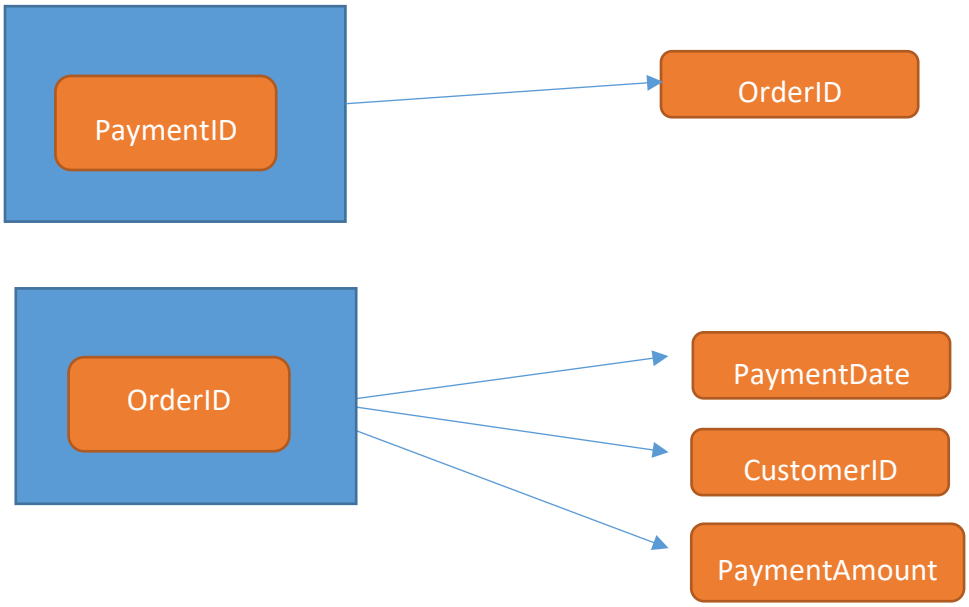
Dependency Preserving: Yes

Table **Payment** = (PaymentID#, OrderID, CustomerID, PaymentAmount, PaymentDate)

FD:
PaymentID → OrderID, CustomerID, PaymentAmount, PaymentDate

3NF
We have 2 tables**:**
Table **Payment** = (PaymentID#, OrderID)
Table **Order** = (OrderID, CustomerID, PaymentAmount, PaymentDate)

PaymentID → OrderID
OrderID → CustomerID, PaymentAmount, PaymentDate



<u>BCNF Algorithm-</u> Table **<u>Payment</u>**

Composite Key (to Uniquely Identify):{PaymentID}

R(PaymentID, OrderID)

FD:

PaymentID(A) → OrderID(B)

R(A,B)

A→ B

Checking if left side is a key:

A+ = {B} is a key

Final BCNF schema: R(A,B) Therefore,
this is in BCNF form.

Dependency Preserving: Yes

Composite Key (to Uniquely Identify):{OrderID}

R(OrderID, PaymentDate, CustomerID, PaymentAmount)

FD:

OrderID(A) → PaymentDate(B), CustomerID(C), PaymentAmount(D)

R(A,B,C,D)

A→ B,C,D

Checking if left side is a key:

A+ = {A, B, C,D} is a key

Final BCNF schema: R(A,B,C,D) Therefore,
this is in BCNF form.
Dependency Preserving: **Yes**

9. Application of Database by Web Based UI

For this assignment we decided to implement our database in apex oracle.

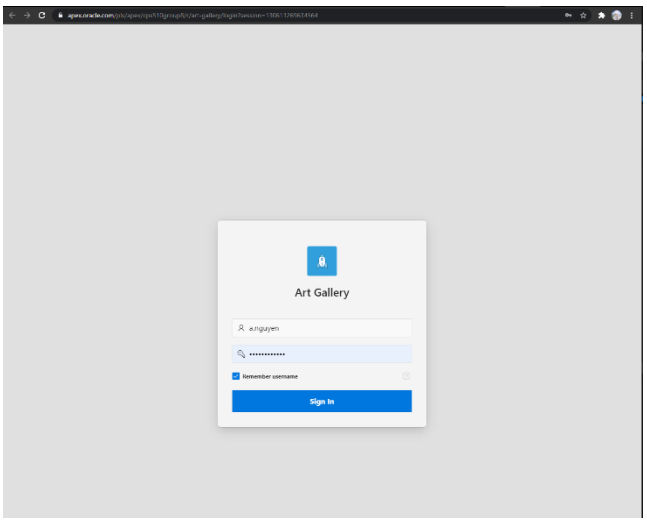
Assumptions and Limitations:

While we were working on the application, we realized that oracle APEX seemed to not allow users direct access to the database functions “create” and “drop” through the web apps. This functionality is only available in SQL workshop of APEX. We have implemented this part of the project as a bash shell script instead, which can be seen in the Create/Populate/Drop Tables section of the report.

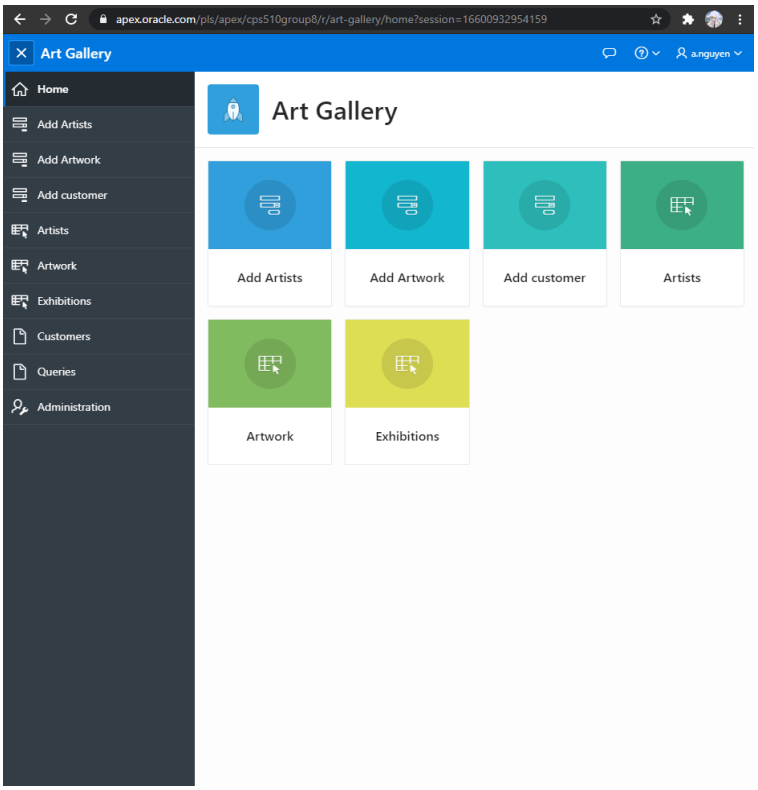
Web App: To view the app: User: a.nguyen Password: Cps510Group8

https://apex.oracle.com/pls/apex/cps510group8/r/art-gallery/login?session=710936501508639

A first look at the app (**BONUS:** menu options and navigation)



8. The app requires the user to log in. To exit the app you can simply log out.



7. The general interface of the app, as you can see there is a menu for the user to navigate through. And you can also navigate through the page via the side bar.

Search: All Text Columns Go Actions Reset

Artistid	Firstname	Lastname	Email	Phonenumber	Locationid
8	Pablo	Picasso			30
578	Sophia	Charlemagne			10
912	Anthony	Constantinous	anyforsalem@gmail.com	90421563421	11
754	Kendrick	Van Gogh			12
332	Louis	Picasso			13
900	Tyler	Willendorf	germartistmai@gmail.com	8914574423	14
891	Pablo	Southerland	jennysellsart@gmail.com	6784543234	15
783	Jean-Antoine	Watteau			16
549	Paul	Dillet	randomgall@gmail.com	8893455678	17
130	Albert	Neuhuys	albertsales34@gmail.com	1134373422	18
333	Kyle	Denise	slmyartkyle@gmail.com	8896342111	8
148	James	Lone	jleartist@gmail.com	4134563421	5
9	Alphonse	Hubert			31
7	Leonardo	Capablanca	leoartsales@gmail.com	441342634990	7
432	Taylor	Octavian	taywrkmadd@gmail.com	99345234428	9

Total 15

6. **The table of artists, with their ID, Names and Contact information as well as their locationID.**

Art Gallery a.nguyen

Search: All Text Columns Go Actions Reset

Nameart	Dateadded	Sizeart	Price	Typeart	Mediumart	Yearcreated	Exhibitionid	Timeperiod	Artist Firstname	Artist Lastname	Artworkid
Lords	12/19/2019	Large	2,323	Painting	Pastel	1,803	2	Neoclassical	Sophia	Charlemagne	9764
Prince Augusto	12/19/2019	Large	1,089	Painting	Acrylic on Canvas	1,949	2	1900s	Anthony	Constantinous	6892
Silent Night	2/25/2019	Small	7,171	Sculpture	Ivory	1,343	3	Medieval	Kendrick	Van Gogh	3436
Outers	2/25/2019	Small	8,899	Sculpture	Marble	1,842	3	Romantic	Louis	Picasso	5313
Glory	2/25/2019	Small	6,899	Sculpture	Marble	1,954	1	1950-Present	Tyler	Willendorf	6604
Dimension	2/25/2019	Small	9,999	Sculpture	Marble	2,007	1	1950-Present	Pablo	Southerland	1340
Calamity	4/13/2019	Large	6,735	Painting	Acrylic on Canvas	1,967	4	1950-Present	Paul	Dillet	9814
Generic Name	10/3/2018	Medium	1,234	Sculpture	Ceramic	2,018	1	Classic	Albert	Neuhuys	2344
Reflection	2/25/2019	Small	8,899	Sculpture	Gypsum	1,895	4	Realistic	Albert	Neuhuys	6645
Lenveyas	3/18/2019	Large	5,670	Statue	Copper	2,019	5	1950-Present	James	Lone	8674
Soup	11/25/2020	Small					1		Alphonse	Hubert	1
Belle Nuit	2/25/2019	Medium	5,010	Painting	Tempera	2,000	1	1950-Present	Leonardo	Capablanca	7369
War and Peace	12/19/2019	Medium	7,000	Sculpture	Marble	1,940	2	1900s	Taylor	Octavian	4999

Total 13

5. **The table of artworks associated with the artistID, with their respective information such as price, artist info, size, medium etc.**

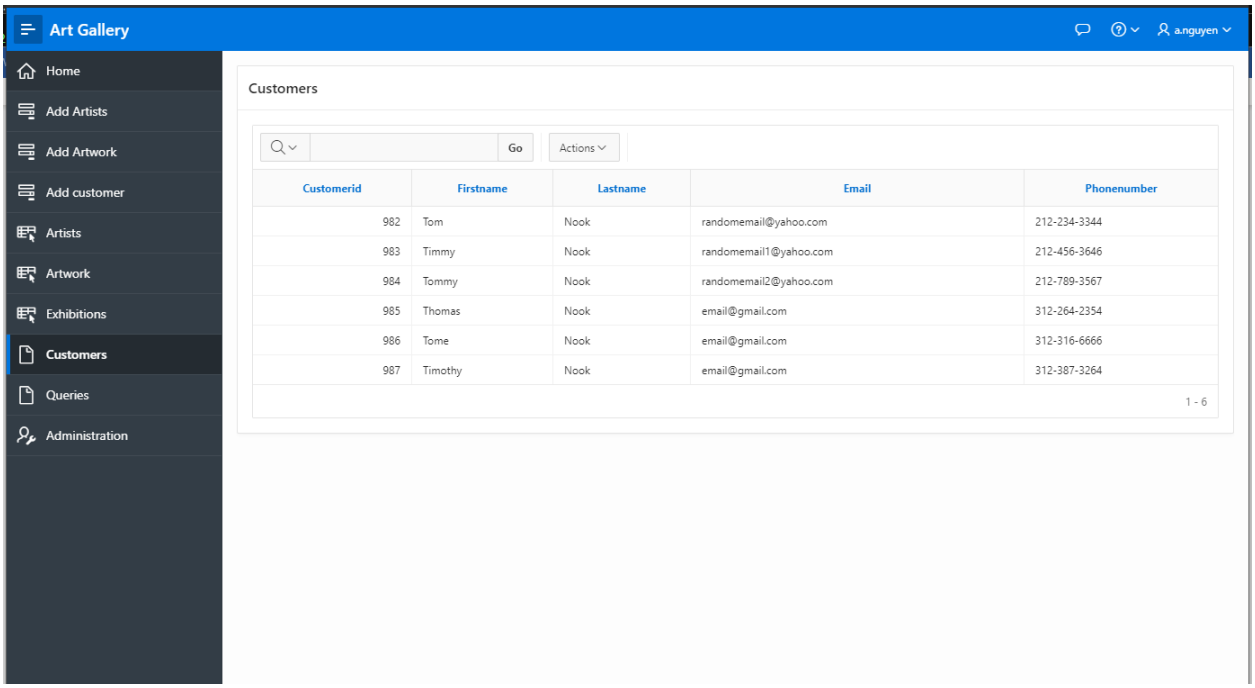
Art Gallery a.nguyen

Search: All Text Columns Go Actions Reset

Gallery Title	Startdate	Enddate
Oldendel	1/8/2020	1/15/2020
Tulipshaw	2/8/2019	2/15/2019
Relicmilie	2/8/2019	2/20/2019
Skipleaf	3/18/2019	3/25/2019
Leafmill	3/18/2019	3/25/2019

1 rows selected Total 5

4. **The table of current exhibitions on display in the gallery**

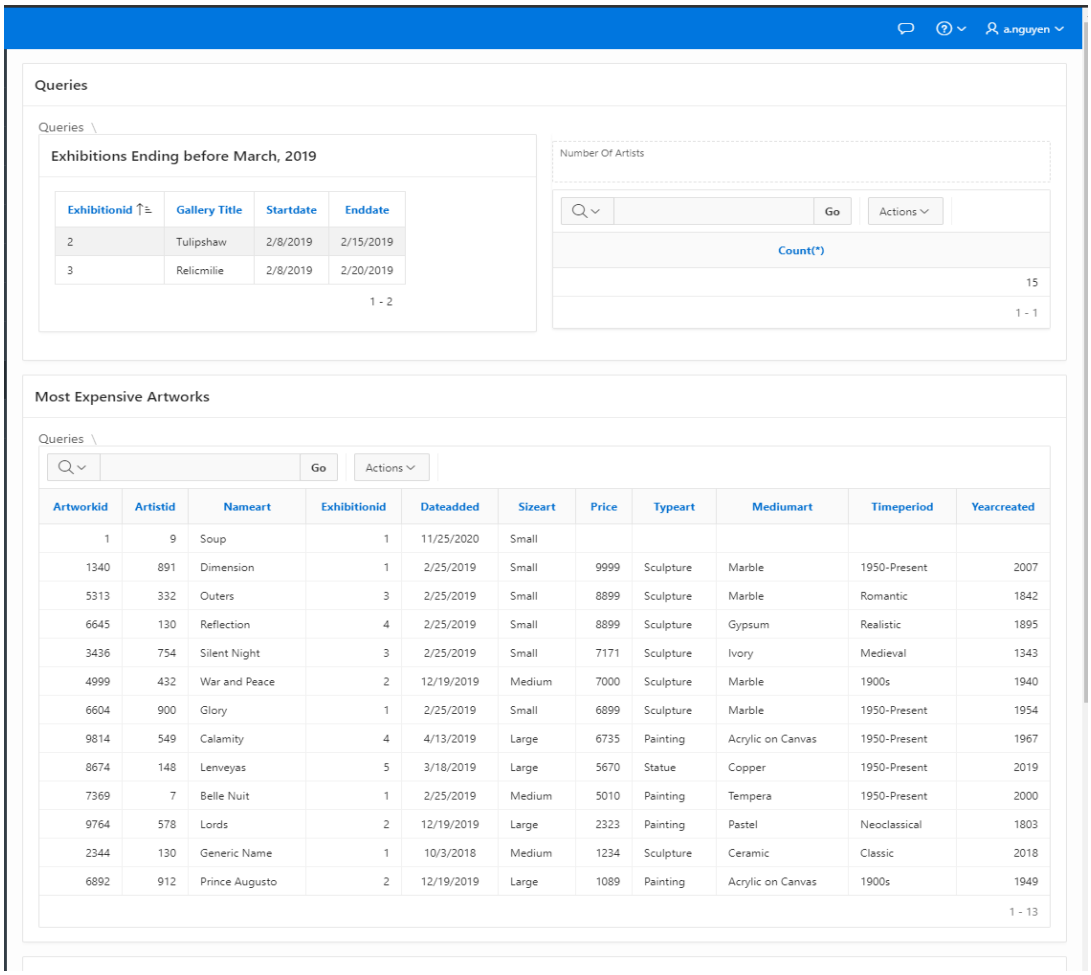


3. **The table of customers**

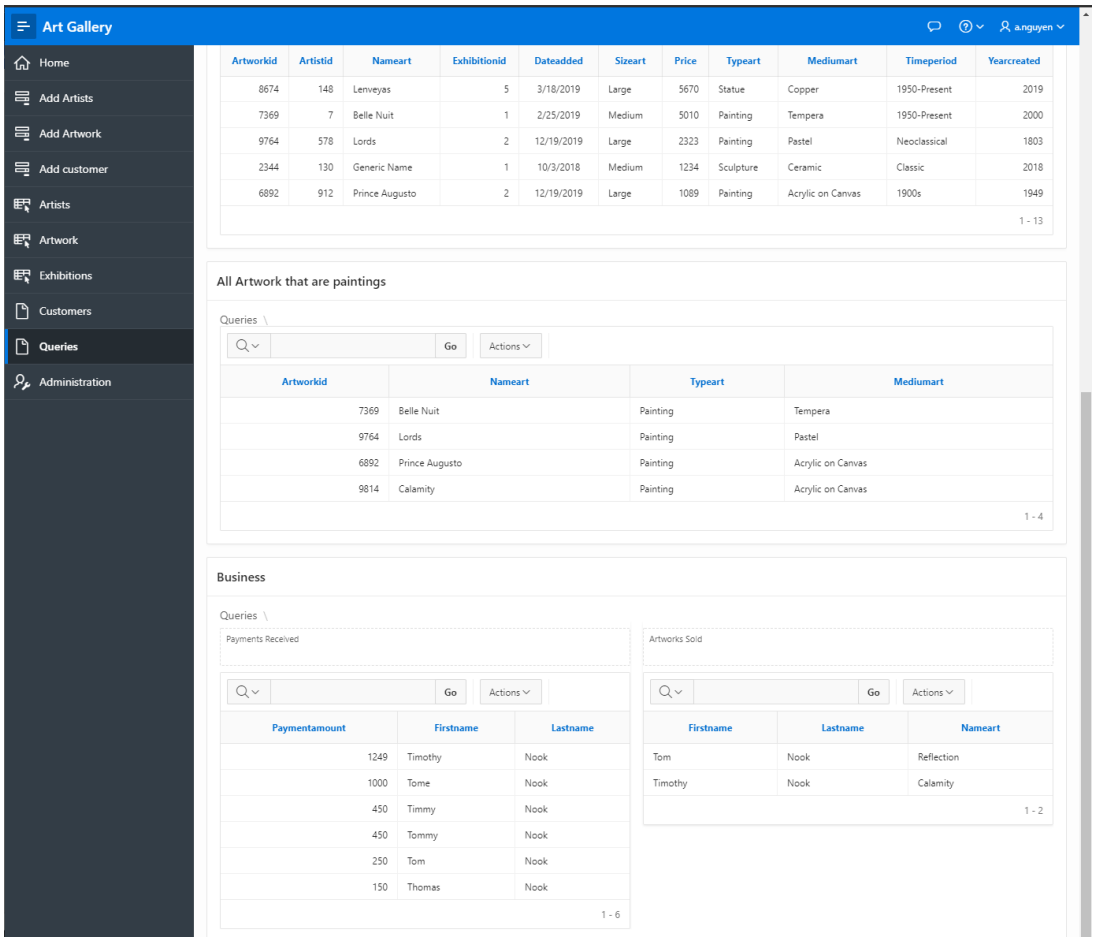
QUERIES in the Application:

Apex oracle allowed us to display queries using our database on the page through interactive reports.

The queries:

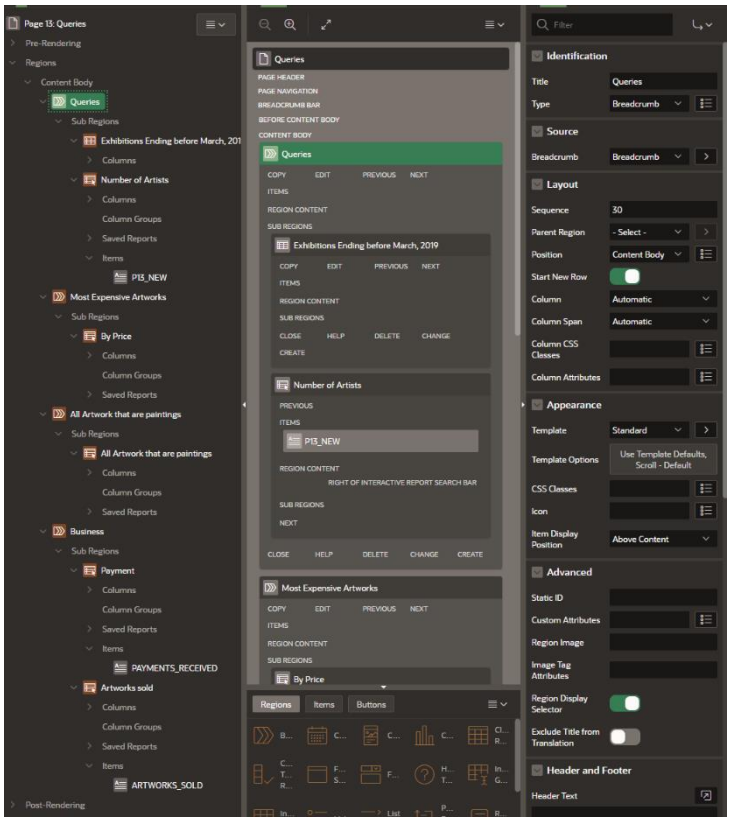


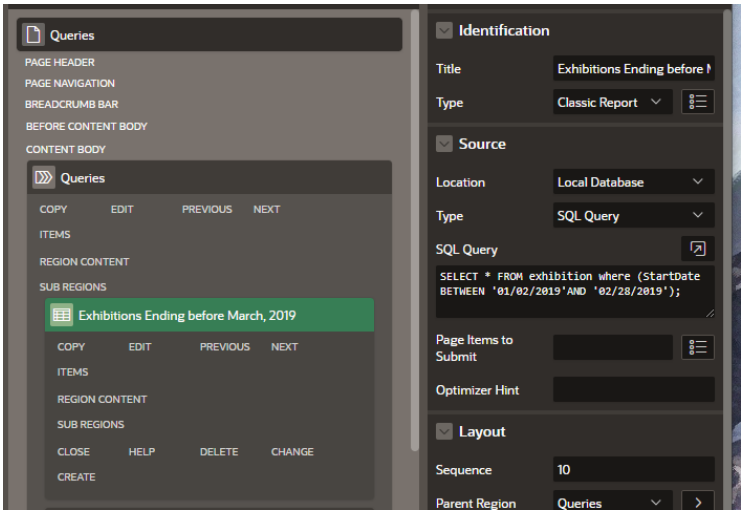
2. **Part 1: The page displays the following: Exhibitions that end before March, Number of artists in the database, and a list of artworks from most to least expensive**



1. **Part 2: The page displays the following: Artworks that are paintings, Payments received from customers, and what artworks that have been sold**

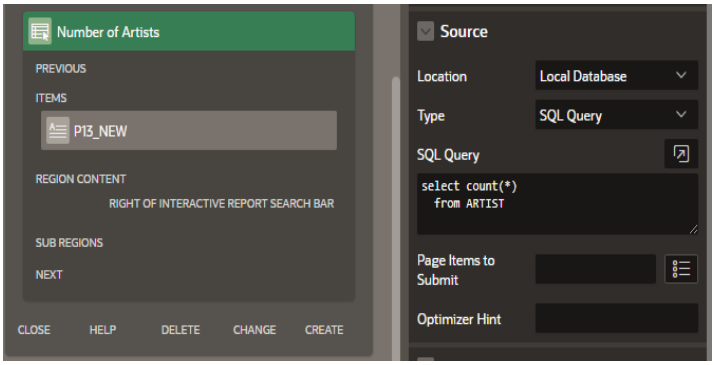
How these queries were implemented:

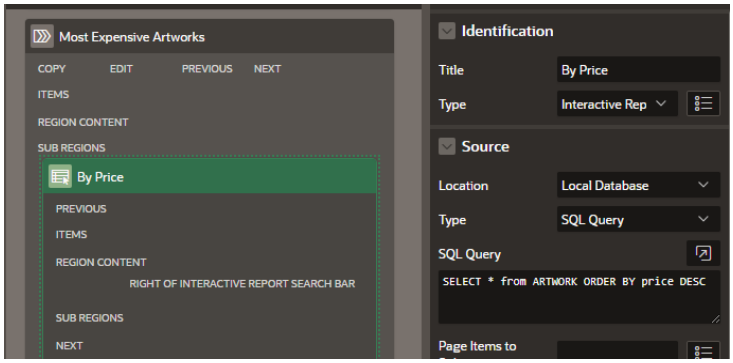


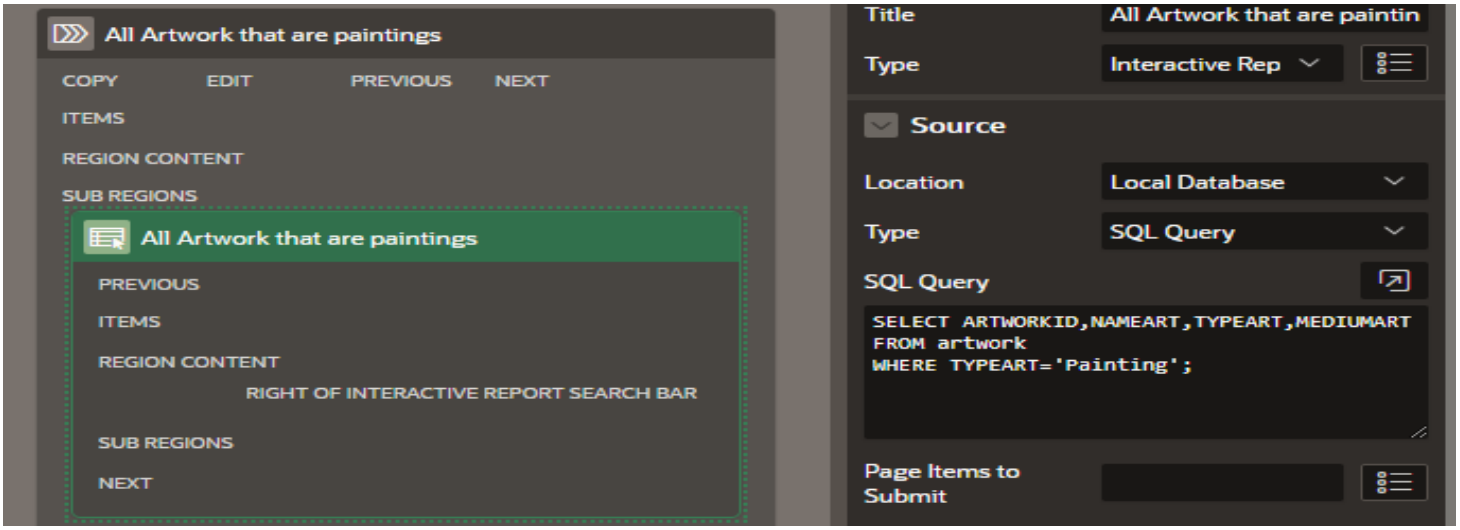


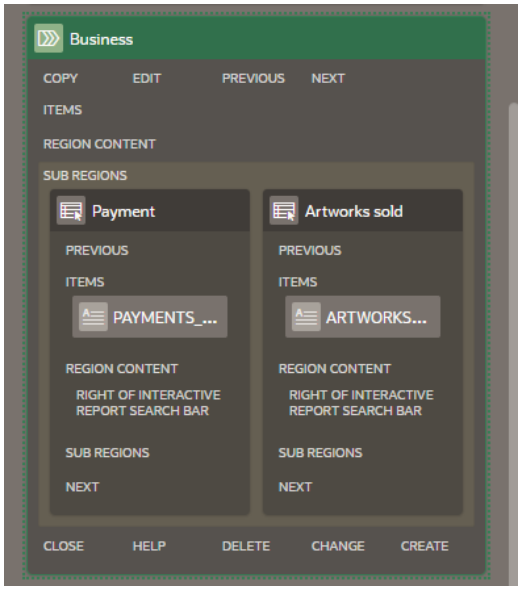
Apex oracle allowed us to edit the Queries page, where we added multiple classic/interactive reports in body of the page, which we used to display the queries as seen above.

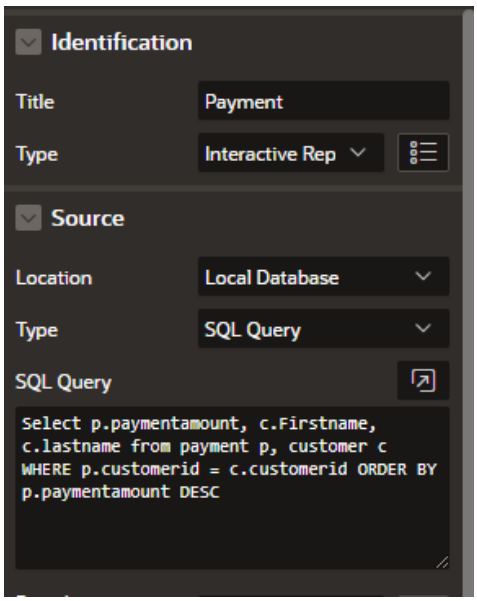
Rest of the query implementation:

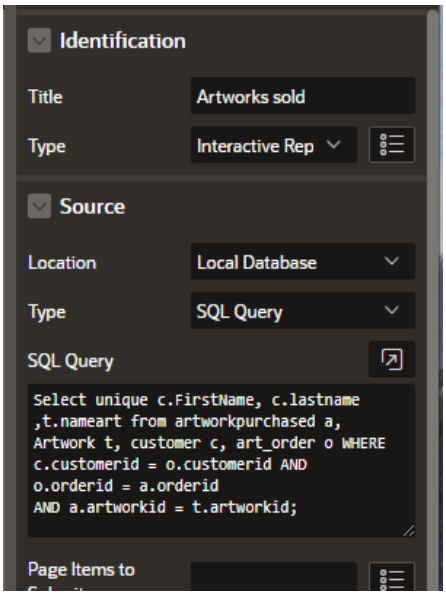




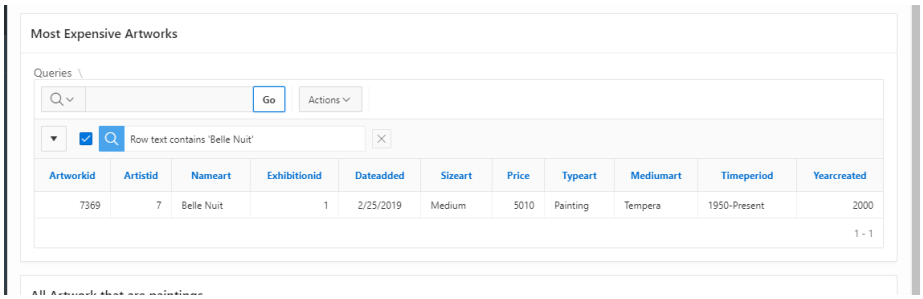




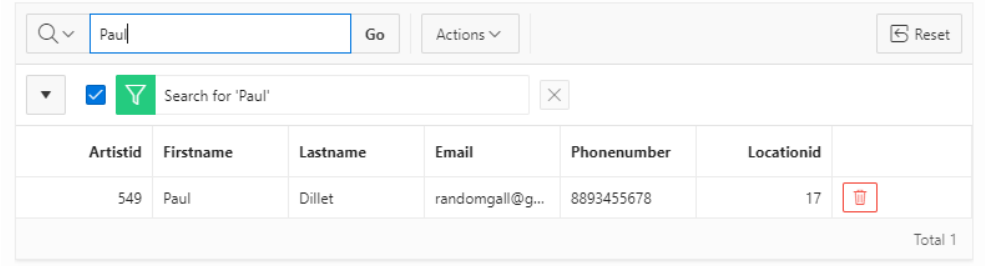


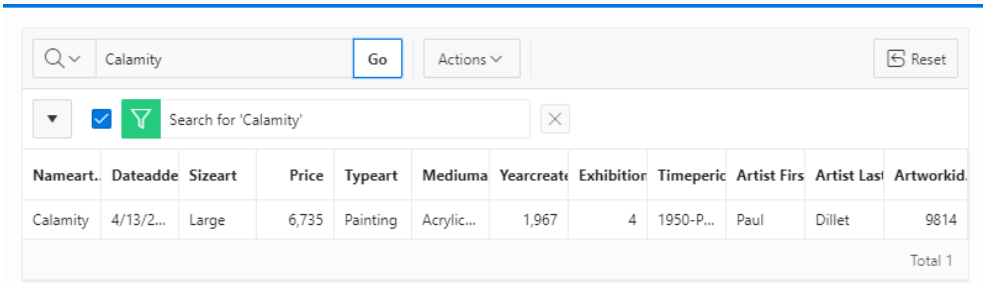


Searching for specific records:

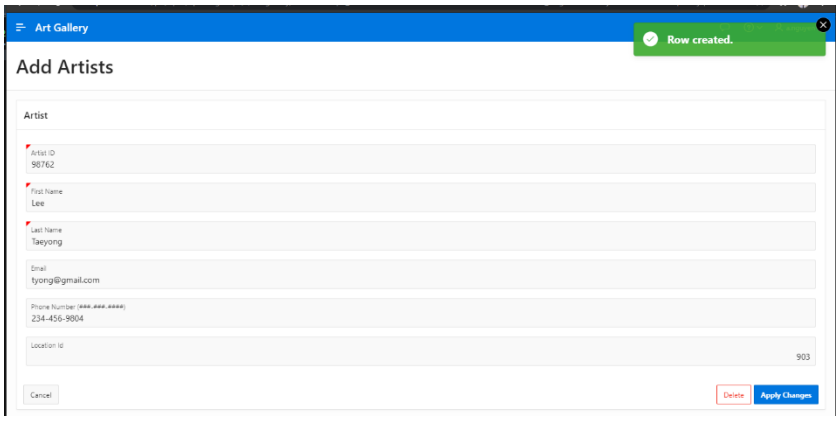


9. As seen, you can search through the query tables, and artist tables, artwork etc.





Updating/Altering Data in the Tables:

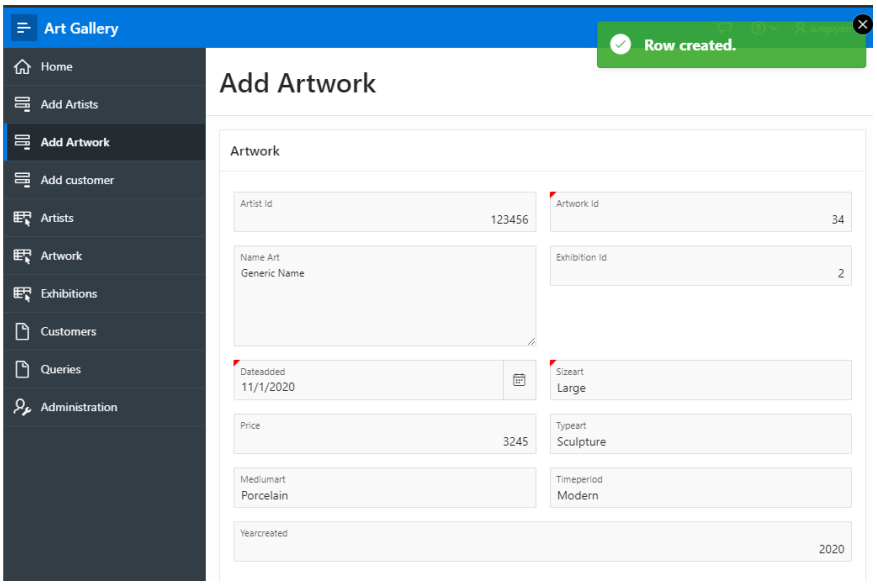


10. Adding an artist: When the data is successfully added it creates a new row in the table.

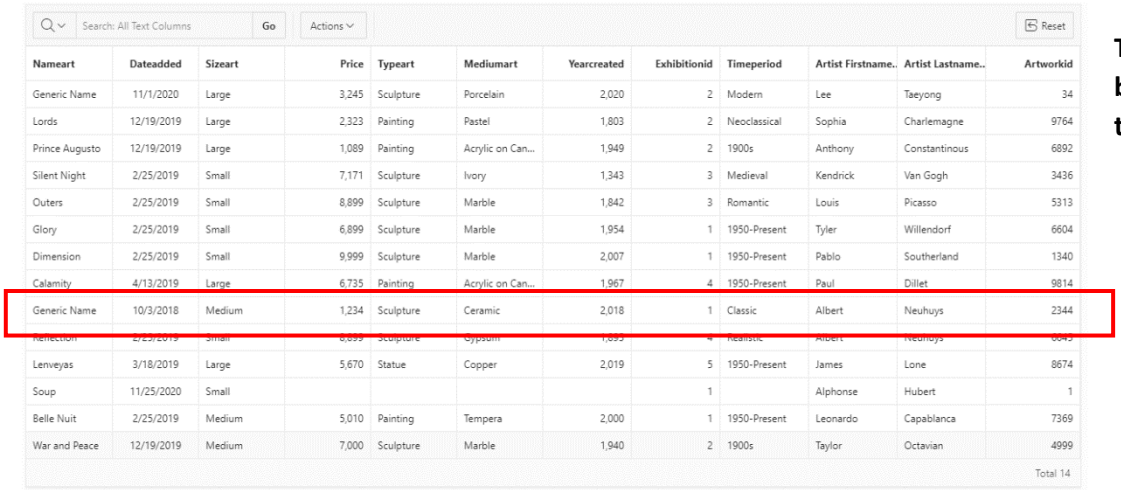
Artistid	Firstname	Lastname	Email	Phonenumber	Locationid
8	Pablo	Picasso			30
578	Sophia	Charlemagne			10
912	Anthony	Constantinous	anyforsalem@gmail.com	90421563421	11
754	Kendrick	Van Gogh			12
332	Louis	Picasso			13
900	Tyler	Willendorf	germartistmai@gmail.com	8914574423	14
891	Pablo	Southerland	jennysellsart@gmail.com	6784543234	15
783	Jean-Antoine	Watteau			16
549	Paul	Dillet	randomgall@gmail.com	8893455678	17
130	Albert	Neuhuys	albertsales34@gmail.com	1134373422	18
333	Kyle	Denise	slmyartkyle@gmail.com	8896342111	8
148	James	Lone	jleartist@gmail.com	4134563421	5
9	Alphonse	Hubert			31
7	Leonardo	Capablanca	leoartsales@gmail.com	441342634990	7
98762	Lee	Taeyong	tyong@gmail.com	234-456-9804	903
432	Taylor	Octavian	taywrkmadd@gmail.com	99345234428	9

The new artist has been successfully added to the table.

Adding artwork:



10. Adding an artwork: When the data is successfully added it creates a new row in the table. NOTE: artwork cannot be added without an existing artist. Since the artistID (foreign key) is required.

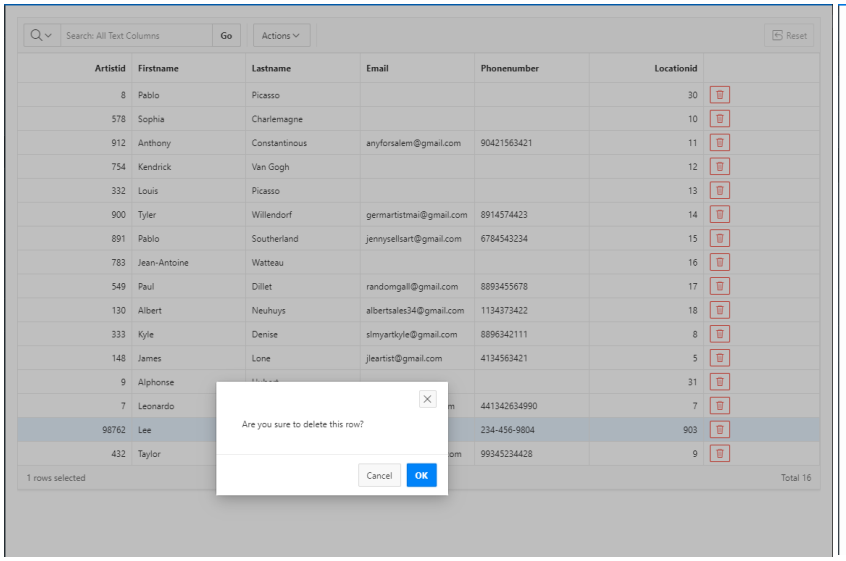


Nameart	Dateadded	Sizeart	Price	Typeart	Mediumart	Yearcreated	Exhibitionid	Timeperiod	Artist Firstname..	Artist Lastname..	Artworkid
Generic Name	11/1/2020	Large	3,245	Sculpture	Porcelain	2,020	2	Modern	Lee	Taeyong	34
Lords	12/19/2019	Large	2,323	Painting	Pastel	1,803	2	Neoclassical	Sophia	Charlemagne	9764
Prince Augusto	12/19/2019	Large	1,089	Painting	Acrylic on Can...	1,949	2	1900s	Anthony	Constantinous	6892
Silent Night	2/25/2019	Small	7,171	Sculpture	Ivory	1,343	3	Medieval	Kendrick	Van Gogh	3436
Outers	2/25/2019	Small	8,899	Sculpture	Marble	1,842	3	Romantic	Louis	Picasso	5313
Glory	2/25/2019	Small	6,899	Sculpture	Marble	1,954	1	1950-Present	Tyler	Willendorf	6604
Dimension	2/25/2019	Small	9,999	Sculpture	Marble	2,007	1	1950-Present	Pablo	Southerland	1340
Calamity	4/13/2019	Large	6,735	Painting	Acrylic on Can...	1,967	4	1950-Present	Paul	Dillet	9814
Generic Name	10/3/2018	Medium	1,234	Sculpture	Ceramic	2,018	1	Classic	Albert	Neuhuys	2344
Reflection	2/23/2019	Small	6,859	Sculpture	Gypsum	1,895	4	Realistic	Albert	Neuhuys	6845
Lenveyas	3/18/2019	Large	5,670	Statue	Copper	2,019	5	1950-Present	James	Lone	8674
Soup	11/25/2020	Small					1		Alphonse	Hubert	1
Belle Nuit	2/25/2019	Medium	5,010	Painting	Tempera	2,000	1	1950-Present	Leonardo	Capablanca	7369
War and Peace	12/19/2019	Medium	7,000	Sculpture	Marble	1,940	2	1900s	Taylor	Octavian	4999

Total 14

The new artwork has been successfully added to the table.

Deleting an artist:



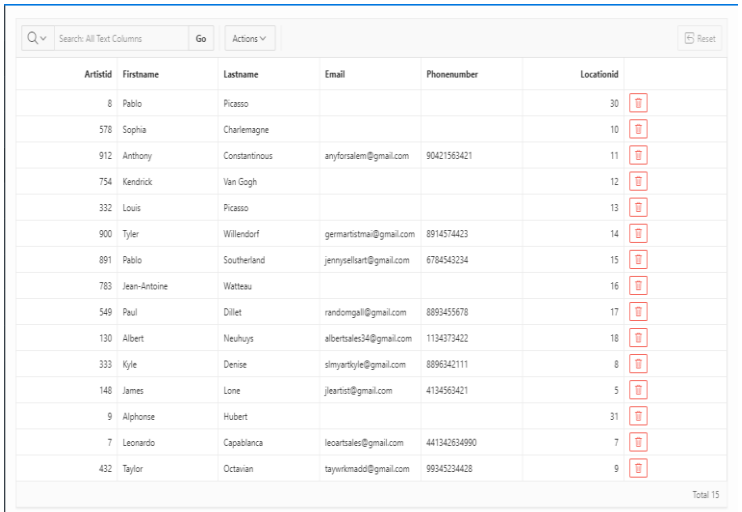
Artistid	Firstname	Lastname	Email	Phonenumber	Locationid
8	Pablo	Picasso			30
578	Sophia	Charlemagne			10
912	Anthony	Constantinous	anyforsalem@gmail.com	90421563421	11
754	Kendrick	Van Gogh			12
332	Louis	Picasso			13
900	Tyler	Willendorf	germartistmai@gmail.com	8914574423	14
891	Pablo	Southerland	jennysellsart@gmail.com	6784543234	15
783	Jean-Antoine	Watteau			16
549	Paul	Dillet	randomgall@gmail.com	8893455678	17
130	Albert	Neuhuys	albertsales34@gmail.com	1134373422	18
333	Kyle	Denise	slmyartkyle@gmail.com	8896342111	8
148	James	Lone	jleartist@gmail.com	4134563421	5
9	Alphonse				31
7	Leonardo			441342634990	7
98762	Lee			234-456-9804	903
432	Taylor			99345234428	9

Are you sure to delete this row?

Cancel OK

1 rows selected

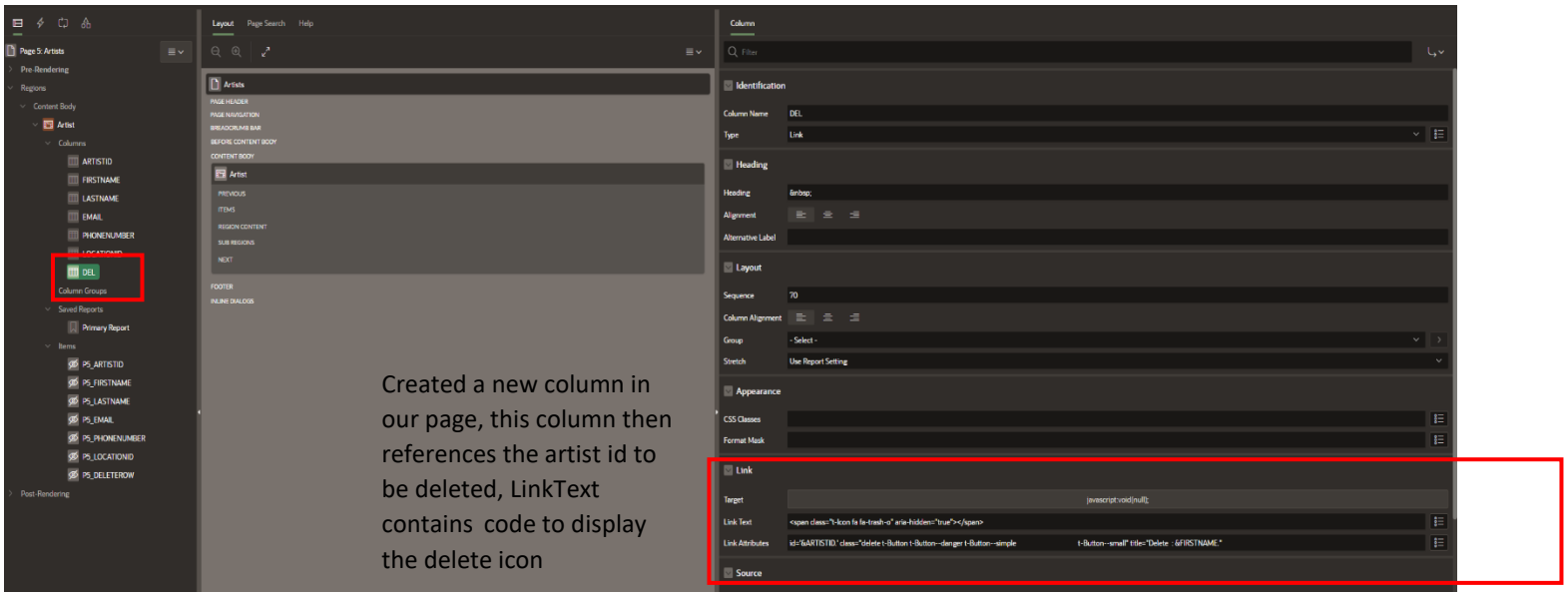
Total 16

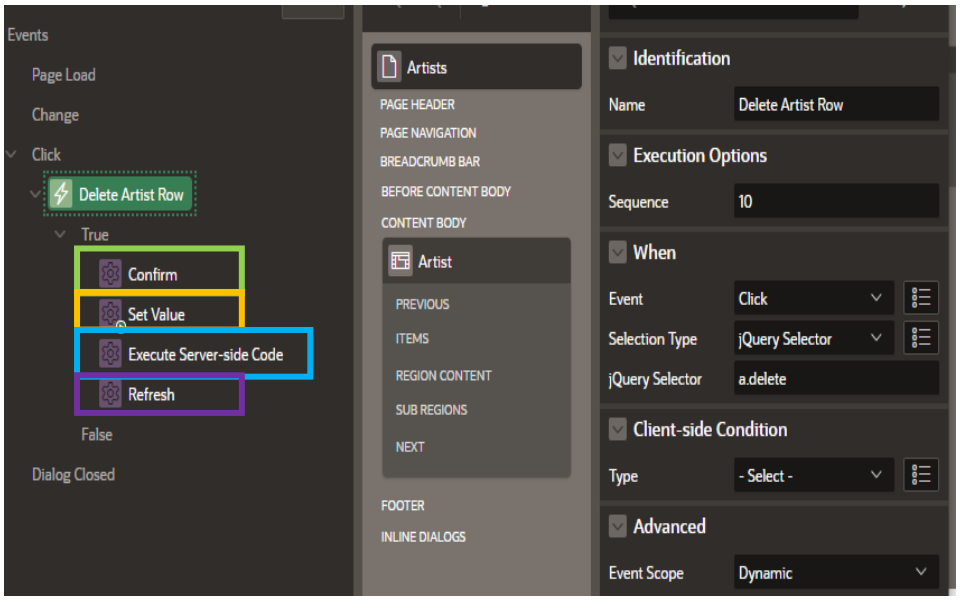


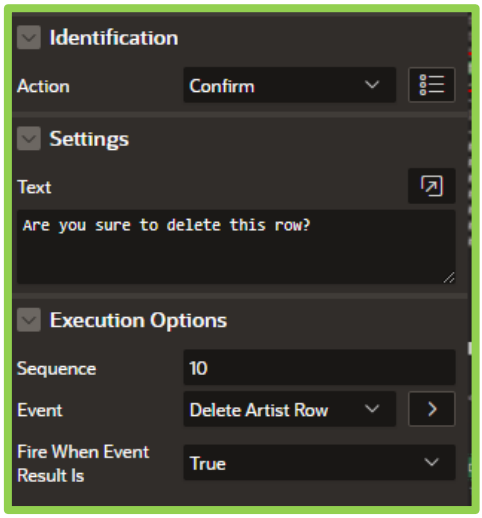
Artistid	Firstname	Lastname	Email	Phonenumber	Locationid
8	Pablo	Picasso			30
578	Sophia	Charlemagne			10
912	Anthony	Constantinous	anyforsalem@gmail.com	90421563421	11
754	Kendrick	Van Gogh			12
332	Louis	Picasso			13
900	Tyler	Willendorf	germartistmai@gmail.com	8914574423	14
891	Pablo	Southerland	jennysellsart@gmail.com	6784543234	15
783	Jean-Antoine	Watteau			16
549	Paul	Dillet	randomgall@gmail.com	8893455678	17
130	Albert	Neuhuys	albertsales34@gmail.com	1134373422	18
333	Kyle	Denise	slmyartkyle@gmail.com	8896342111	8
148	James	Lone	jleartist@gmail.com	4134563421	5
9	Alphonse	Hubert			31
7	Leonardo	Capablanca	leoartsales@gmail.com	441342634990	7
432	Taylor	Octavian	taywrkmadd@gmail.com	99345234428	9

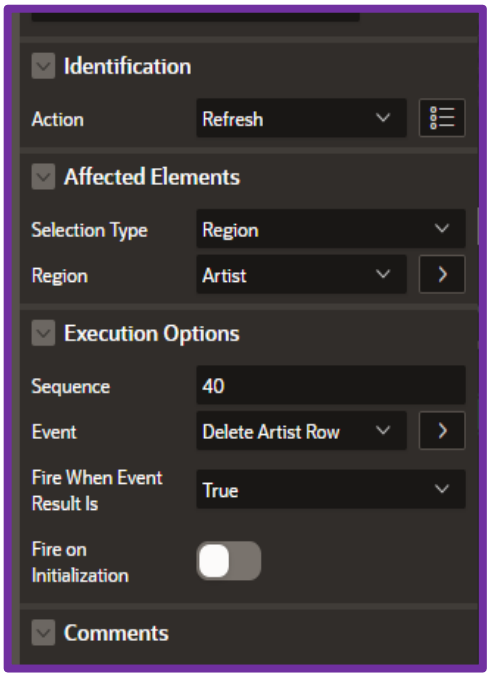
Total 15

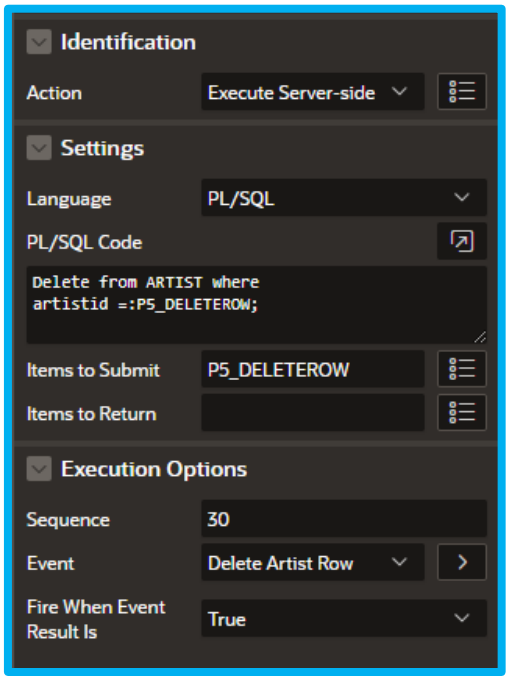
How delete was implemented:

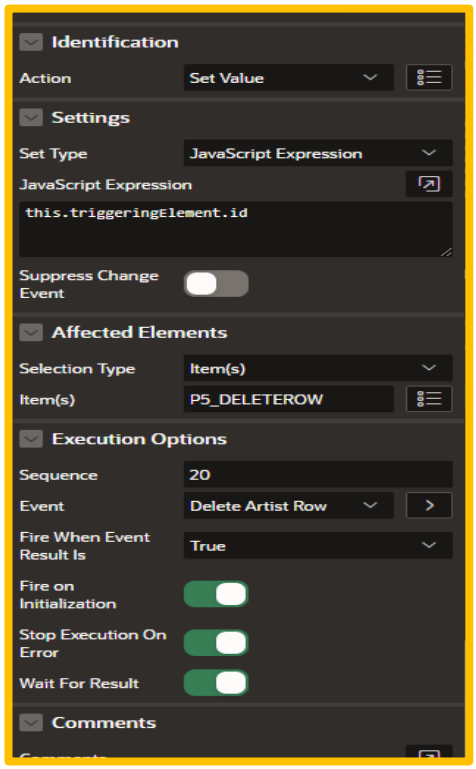












Create/Populate/Drop Tables:

NOTE: As stated above, this functionality is not available in Apex applications. We have implemented this part of the project as a bash shell script instead. Create/Populate/Drop tables files will be included in the zip file.

```
==================================================================
|                    Oracle All Inclusive Tool
|
|                       Main Menu - Select Desired Operation(s):
|
|        <CTRL-Z Anytime to Enter Interactive CMD Prompt>
|
------------------------------------------------------------------
  M)  View Manual

  1)  Drop Tables
  2)  Create Tables
  3)  Populate Tables
  4)  Query Tables

      a)  Customer and Payments
      b)  Artist and Artwork
      c)  Exhibition:Artwork and Corresponding Art in EXBID23206
      d)  Display Artists that have Small Sculptures
      e)  Average Price of Sculptures
      f)  Number of Rentals
      g)  Customers Using a Gmail Account
      h)  Artists with Artwork Belonging in 1950s NOT made w/ Marble
      i)  Customers Living on Sesame Street and Rental Orders

  X)  Force/Stop/Kill Oracle DB

  E)  End/Exit
Choose:
```

2. Menu

```
  X)  Force/Stop/Kill Oracle DB

  E)  End/Exit
Choose:
2

SQL*Plus: Release 12.1.0.2.0 Production on Tue Dec 1 20:55:13 2020

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> SQL>   2    3    4    5    6    7
Table created.

SQL> SQL>   2    3    4    5    6
Table created.

SQL> SQL>   2    3    4    5
Table created.

SQL> SQL>   2    3    4
Table created.

SQL> SQL>   2    3    4    5
Table created.

SQL> SQL>   2    3    4    5    6    7    8    9   10   11   12   13
Table created.

SQL> SQL>   2    3    4    5    6
Table created.

SQL> SQL>   2    3
Table created.

SQL> SQL>   2    3
Table created.

SQL> SQL>   2    3    4    5    6
Table created.

SQL> SQL>   2    3    4    5
Table created.

SQL> SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
==================================================================
|                    Oracle All Inclusive Tool
|
|                       Main Menu - Select Desired Operation(s):
|
|        <CTRL-Z Anytime to Enter Interactive CMD Prompt>
|
------------------------------------------------------------------
```

1. Create Tables

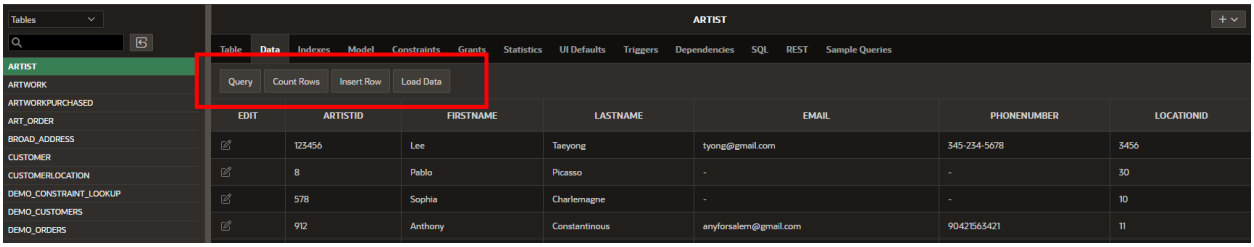
```
SQL> SQL>
1 row created.

SQL>
1 row created.

SQL> SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL> SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL>
1 row created.

SQL> SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
./menu.sh: line 54: Pause: command not found
==================================================================
```

4. Populate Tables

```
SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL>
Table dropped.

SQL> Disconnected from Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options
================================================================
```

6. Drop Tables

NOTE: Code for our Create/Populate/Drop tables files will be included in the zip file. Apex oracle also allowed us to directly submit code into the application as well as submit rows of data, a txt file containing relevant code (artgallery.txt) will be included in the zip file.



Further Notes - Assumptions and Limitations:
See documentation at:

https://docs.oracle.com/database/apex-5.1/AEUTL/managing-tables.htm#AEUTL109

and

https://docs.oracle.com/database/apex-5.1/HTMDB/managing-dynamic-actions.htm#HTMDB27020

9.2 Proving SQL Database in BCNF:

Prove that the table, ARTIST, CUSTOMER, EXHIBITION, ARTWORK, ARTWORKPURCHASED, PAYMENT is in **BCNF**

Table **Artist** = (ArtistID#, FirstName, LastName, Email, PhoneNumber, LocationID)

SQL Source Code:

```
CREATE TABLE artist (
        ArtistID NUMBER PRIMARY KEY,
        Firstname VARCHAR2(25) NOT NULL,
        Lastname VARCHAR2(25) NOT NULL,
        Email VARCHAR(40),
        PhoneNumber VARCHAR2(12),
        LocationID NUMBER UNIQUE NOT NULL);
```

From **SQL Developer** we have the following data inside ARTIST**:**

```
SELECT * FROM artist;
```

Script Output × | Query Result ×

SQL | All Rows Fetched: 19 in 0.069 seconds

	ARTISTID	FIRSTNAME	LASTNAME	EMAIL	PHONENUMBER	LOCATIONID
1	123	Dawn	Lol	lol@gmail.com	6471234567	1
2	456	Smith	Jones	sje@gmail.com	4167898907	2
3	789	Lee	Li	leelol@gmail.com	4136547890	3
4	367	Sally	Hong	bluesky@yahoo.ca	6475781265	4
5	148	James	Lone	jleartist@gmail.com	4134563421	5
6	1	Leo	Da Vinci	(null)	(null)	6
7	7	Leonardo	Capablanca	leoartsales@gmail.com	441342634990	7
8	333	Kyle	Denise	slmyartkyle@gmail.com	8896342111	8
9	432	Taylor	Octavian	taywrkmadd@gmail.com	99345234428	9
10	578	Sophia	Charlemagne	(null)	(null)	10
11	912	Anthony	Constantinous	anyforsalem@gmail.com	90421563421	11
12	754	Kendrick	Van Gogh	(null)	(null)	12
13	332	Louis	Picasso	(null)	(null)	13
14	719	Seagle	Aivazovsky	(null)	(null)	14
15	900	Tyler	Willendorf	germartistmai@gmail.com	8914574423	15
16	891	Pablo	Southerland	jennysellsart@gmail.com	6784543234	16
17	783	Jean-Antoine	Watteau	(null)	(null)	17
18	549	Paul	Dillet	randomgall@gmail.com	8893455678	18
19	130	Albert	Neuhuys	albertsales34@gmail.com	1134373422	19

Proof that the table is in BCNF:

Table Artist
Composite Key (to Uniquely Identify): {ArtistID}

R(ArtistID, FirstName, LastName, Email, PhoneNumber, LocationID)

FD:
ArtistID (A) → FirstName(B), LastName(C), Email(D), PhoneNumber(E), LocationID(F)

R(A,B,C,D,E,F)

A → B, C, D, E, F

Checking if left side is a key:

A+ = {A, B, C, D, E, F} is a key

Final BCNF schema is

R = (A,B,C,D,E,F)

Dependency Preserving: Yes

Table **Customer** = (CustomerID#, FirstName, LastName, Email, PhoneNumber)
SQL Source Code:

```
CREATE TABLE customer (
        CustomerID NUMBER PRIMARY KEY,
        FirstName VARCHAR2(25) NOT NULL,
        LastName VARCHAR2(25) NOT NULL,
        Email VARCHAR(40) NOT NULL,
        PhoneNumber VARCHAR2(12));
```

```
SELECT * FROM customer;
```

Script Output | Query Result

SQL | All Rows Fetched: 6 in 0.015 seconds

	CUSTOMERID	FIRSTNAME	LASTNAME	EMAIL	PHONENUMBER
1	982	Tom	Nook	randomemail@yahoo.com	212-234-3344
2	983	Timmy	Nook	randomemail1@yahoo.com	212-456-3646
3	984	Tommy	Nook	randomemail2@yahoo.com	212-789-3567
4	985	Thomas	Nook	email@gmail.com	312-264-2354
5	986	Tome	Nook	email@gmail.com	312-316-6666
6	987	Timothy	Nook	email@gmail.com	312-387-3264

Proof that the table is in BCNF:

Table Customer
Composite Key (to Uniquely Identify):{CustomerID}

R(CustomerID, FirstName, LastName, PhoneNumber, Email)

FD:

CustomerID(A) → FirstName (B), LastName (C), PhoneNumber (D), Email (E)

R(A,B,C,D,E)

A→ B, C, D, E

E→ A, B, C, D

D→ A, B, C, E

Checking if left side is a key:

A+ = {A, B, C, D, E} is a key

E+ = {A, B, C, D, E} is a key

D+ = {A, B, C, D, E} is a key

Every determinant is a candidate key. Therefore,
this is in BCNF form.

Final BCNF schema: R(A,B,C,D,E)

Dependency Preserving: Yes

Table **Exhibition** = (ExhibitionID#, Gallery_Name, StartDate, EndDate)

SQL Source Code:

```
CREATE TABLE exhibition (
        ExhibitionID NUMBER PRIMARY KEY,
        Gallery_Title VARCHAR(50),
        StartDate DATE,
        EndDate DATE);
```

```
SELECT * FROM exhibition;
```

Script Output | Query Result

SQL | All Rows Fetched: 5 in 0.015 seconds

	EXHIBITIONID	GALLERY_TITLE	STARTDATE	ENDDATE
1	1	Oldendel	20-01-08	20-01-15
2	2	Tulipshaw	19-02-08	19-02-15
3	3	Relicmilie	19-02-08	19-02-20
4	4	Skipleaf	19-03-18	19-03-25
5	5	Leafmill	19-03-18	19-03-25

Composite Key (to Uniquely Identify):{ExhibitionID}

R(ExhibitionID, Gallery_Name, StartDate, EndDate)

FD:

ExhibitionID(A) → Gallery_Name(B), StartDate(C), EndDate(D)

R(A,B,C,D)

A→ B, C,D

Checking if left side is a key:

A+ = {A, B, C, D } is a key

Final BCNF schema: R(A,B,C,D) Therefore,
this is in BCNF form.

Dependency Preserving: Yes

Table **Artwork** = (ArtworkID,ArtistID, NameArt, Exhibition, DateAdded, SizeArt, Price, TypeArt, MediumArt, YearCreated, TimePeriod)

SQL Source Code:

CREATE TABLE artwork (

ArtworkID NUMBER PRIMARY KEY,
ArtistID NUMBER REFERENCES artist(ArtistID),
NameArt VARCHAR2(100),
ExhibitionID NUMBER REFERENCES exhibition(ExhibitionID),
DateAdded DATE NOT NULL,
SizeArt VARCHAR2(20) NOT NULL,
Price NUMBER,
TypeArt VARCHAR(40),

MediumArt VARCHAR(50),
TimePeriod VARCHAR(15),
YearCreated NUMBER
);

Table Artwork

Composite Key (to Uniquely Identify): {ArtworkID}

R(ArtworkID,ArtistID, NameArt, Exhibition, DateAdded, SizeArt, Price, TypeArt, MediumArt, YearCreated, TimePeriod)

FD:

ArtworkID(A)→ ArtistID(B), NameArt(C), Exhibition(D), DateAdded(E), SizeArt(F), Price(G), TypeArt(H), MediumArt(I), YearCreated(J)

R(A,B,C,D,E,F,G,H,I, J)

A→ B, C,D,E, F, G, H, I, J

Checking if left side is a key:

A+ = {A, B, C, D, E, F, G, H, I, J} is a key

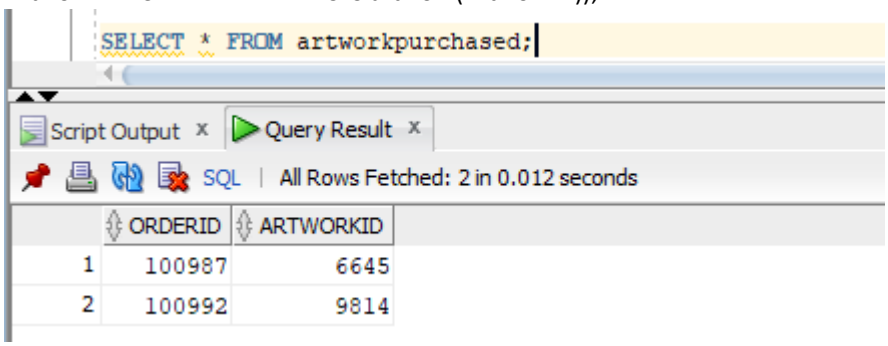
Final BCNF schema: R(A,B,C,D,E,F,G,H,I, J) Therefore, this is in BCNF form.

Dependency Preserving: Yes

Table **ArtworkPurchased** = (<u>OrderID</u>, ArtworkID)

SQL Source Code:

CREATE TABLE artworkPurchased(
OrderID NUMBER REFERENCES art_order(OrderID),
ArtworkID NUMBER REFERENCES artwork(ArtworkID));



	ORDERID	ARTWORKID
1	100987	6645
2	100992	9814

Table ArtworkPurchased:

Composite Key (to Uniquely Identify):{OrderID}

R(<u>OrderID#</u>, ArtworkID)

FD: OrderID(A) → ArtworkID(B)

R(A,B)

A→ B

Checking if left side is a key:

A+ = {A, B} is a key

Final BCNF schema: R(A,B) Therefore,
this is in BCNF form.

Dependency Preserving: Yes

Table **Payment** = (PaymentID#, OrderID, CustomerID, PaymentAmount, PaymentDate)

SQL Source Code:

```
CREATE TABLE payment (
        PaymentID NUMBER PRIMARY KEY,
        OrderID NUMBER REFERENCES art_order(OrderID) NOT NULL,
        CustomerID NUMBER REFERENCES customer(CustomerID) NOT NULL,
        PaymentAmount NUMBER NOT NULL,
        DatePayment DATE NOT NULL);
```

```
SELECT * FROM payment;
```

Script Output | Query Result

SQL | All Rows Fetched: 6 in 0.015 seconds

	PAYMENTID	ORDERID	CUSTOMERID	PAYMENTAMOUNT	DATEPAYMENT
1	200987	100987	982	250	19-05-01
2	200988	100988	983	450	19-04-29
3	200989	100989	984	450	19-04-30
4	200990	100990	985	150	19-05-01
5	200991	100991	986	1000	19-04-29
6	200992	100992	987	1249	19-04-30

BCNF Algorithm- Table Payment

Composite Key (to Uniquely Identify):{PaymentID}

R(PaymentID, OrderID)

FD:

PaymentID(A) → OrderID(B)

R(A,B)

A→ B

Checking if left side is a key:

A+ = {B} is a key

Final BCNF schema: R(A,B) Therefore,
this is in BCNF form.

Dependency Preserving: Yes

Composite Key (to Uniquely Identify):{OrderID}

R(OrderID, PaymentDate, CustomerID, PaymentAmount)

FD:

OrderID(A) → PaymentDate(B), CustomerID(C), PaymentAmount(D)

R(A,B,C,D)

A→ B,C,D

Checking if left side is a key:

A+ = {A, B, C,D} is a key

Final BCNF schema: R(A,B,C,D) Therefore,
this is in BCNF form.
Dependency Preserving: **Yes**

10. Relational Algebra Notation

1) > select all artists in table and sort them by their first name in alphabetical order
SELECT * FROM artist ORDER BY firstname ASC;

RA: σ artist

	ARTISTID	FIRSTNAME	LASTNAME	EMAIL	PHONENUMBER	LOCATIONID
1	130	Albert	Neuhuys	albertsales34@gmail.com	1134373422	19
2	912	Anthony	Constantinous	anyforsalem@gmail.com	90421563421	11
3	123	Dawn	Lol	lol@gmail.com	6471234567	1
4	148	James	Lone	jleartist@gmail.com	4134563421	5
5	783	Jean-Antoine	Watteau	(null)	(null)	17
6	754	Kendrick	Van Gogh	(null)	(null)	12
7	333	Kyle	Denise	slmyartkyle@gmail.com	8896342111	8
8	789	Lee	Li	leelol@gmail.com	4136547890	3
9	1	Leo	Da Vinci	(null)	(null)	6
10	7	Leonardo	Capablanca	leoartsales@gmail.com	441342634990	7
11	332	Louis	Picasso	(null)	(null)	13
12	891	Pablo	Southerland	jennysellsart@gmail.com	6784543234	16
13	549	Paul	Dillet	randomgall@gmail.com	8893455678	18
14	367	Sally	Hong	bluesky@yahoo.ca	6475781265	4
15	719	Seagle	Aivazovsky	(null)	(null)	14
16	456	Smith	Jones	sje@gmail.com	4167898907	2
17	578	Sophia	Charlemagne	(null)	(null)	10
18	432	Taylor	Octavian	taywrkmadd@gmail.com	99345234428	9
19	900	Tyler	Willendorf	germartistmai@gmail.com	8914574423	15

2) > list all the artists whose names start with an S
SELECT firstname * FROM artist WHERE firstname LIKE 'S%';

RA: ∏firstname (σ firstname = “S%”)(artist))

	FIRSTNAME
1	Smith
2	Sally
3	Sophia
4	Seagle

3) > select all the artists with a first name that starts with an L, showing their artistid and last name

SELECT ArtistID,Firstname, Lastname FROM artist WHERE firstname LIKE 'L%';

RA: πArtistID, firstname, lastname (σ firstname = “S%”)(artist))

SQL | All Rows Fetched: 4 in 0.011 seconds

	ARTISTID	FIRSTNAME	LASTNAME
1	789	Lee	Li
2	1	Leo	Da Vinci
3	7	Leonardo	Capablanca
4	332	Louis	Picasso

4) > select data from artwork where the medium is copper and join with the gallery title associated with it

SELECT *
FROM artwork RIGHT JOIN exhibition ON (Exhibition = Gallery_Title);
SELECT * FROM artwork where MediumArt ='Copper';

RA: σ artwork $|\bowtie|_{\text{Exhibition=Gallery_Title}}$($\sigma$ artwork = “Copper”)(artwork))

	ARTWORKID	ARTISTID	NAMEART	EXHIBITIONID	DATEADDED	SIZEART	PRICE	TYPEART	MEDIUMART	TIMEPERIOD	YEARCREATED
1	6778	367	Rackaro	4	19-03-18	Large	5000	Statue	Copper	1950-Present	2006
2	8674	148	Lenveyas	5	19-03-18	Large	5670	Statue	Copper	1950-Present	2019

5) > find the average price of all the sculptures from table artwork

SELECT 'Average Price of Sculptures', AVG(Price)

FROM artwork

WHERE TypeArt = 'Sculpture';

RA: F AVERAGE (PRICE) as Average Price of Sculptures(σ TypeArt = “Sculpture”)(artwork))

	'AVERAGEPRICEOFSCULPTURES'	AVG(PRICE)
1	Average Price of Sculptures	8144.5

6) > find the piece of artworks where the price is less than 3000

SELECT ArtworkID, NameArt, Price from artwork where price < 3000;

RA: ∏ArtworkID, NameArt, Price (σ price < 3000)(artist))

SQL | All Rows Fetched: 5 in 0.013 seconds

	ARTWORKID	NAMEART	PRICE
1	4560	uhm	2345
2	7891	Bleu	1356
3	2345	Oythus	1098
4	9764	Lords	2323
5	6892	Prince Augusto	1089

7) > SELECT addresses of customers that are NOT located on Sesame Street

SELECT LocationID, StreetAddress, PostalCode from SPECIFIC_ADDRESS WHERE NOT StreetAddress like '%Sesame%';

RA: ∏LocationID, StreetAddress, PostalCode (σ StreetAddress ≠ Sesame Street)(SPECIFIC_ADDRESS))

SQL | All Rows Fetched: 6 in 0.015 seconds

	LOCATIONID	STREETADDRESS	POSTALCODE
1	1	123 Cherry Lane	A1AB1B
2	2	456 Cherry Lane	A1AB1B
3	4	123 Main	N1N8C8
4	5	7 Elgin	LOL123
5	6	10 Champs Elysees	UM4D67
6	7	10 Downing Street	H1K8G0

8) > count the number of customers in Customer table

SELECT COUNT(CustomerID) AS Number_of_Customers from customer;

RA: σ COUNT(CustomerID)(customer))

	NUMBER_OF_CUSTOMERS
1	6

9) > count the number of artworks that are sculptures

SELECT COUNT(ArtworkID) AS Number_of_Sculptures from artwork WHERE typeArt='Sculpture';

RA: σ COUNT(ArtworkID(typeArt=’Sculpture’))(customer))

	NUMBER_OF_SCULPTURES
1	6

11. Concluding Remarks

We believe our experience with this SQL Art Gallery database was educational. The collaborative design experience introduced us to the various steps necessary to plan a successful and well implemented SQL database, all the while refining our communication and teamwork skills. We found SQL was easy to learn, and an easy to use language that allows us to store, build queries, and manipulate data. Overall, this project taught us useful skills that will be relevant in the workplace and in real life.