**PROJECT PHASE 2**

**PART 1 : -** Create statement queries

1. ‘state’ table –

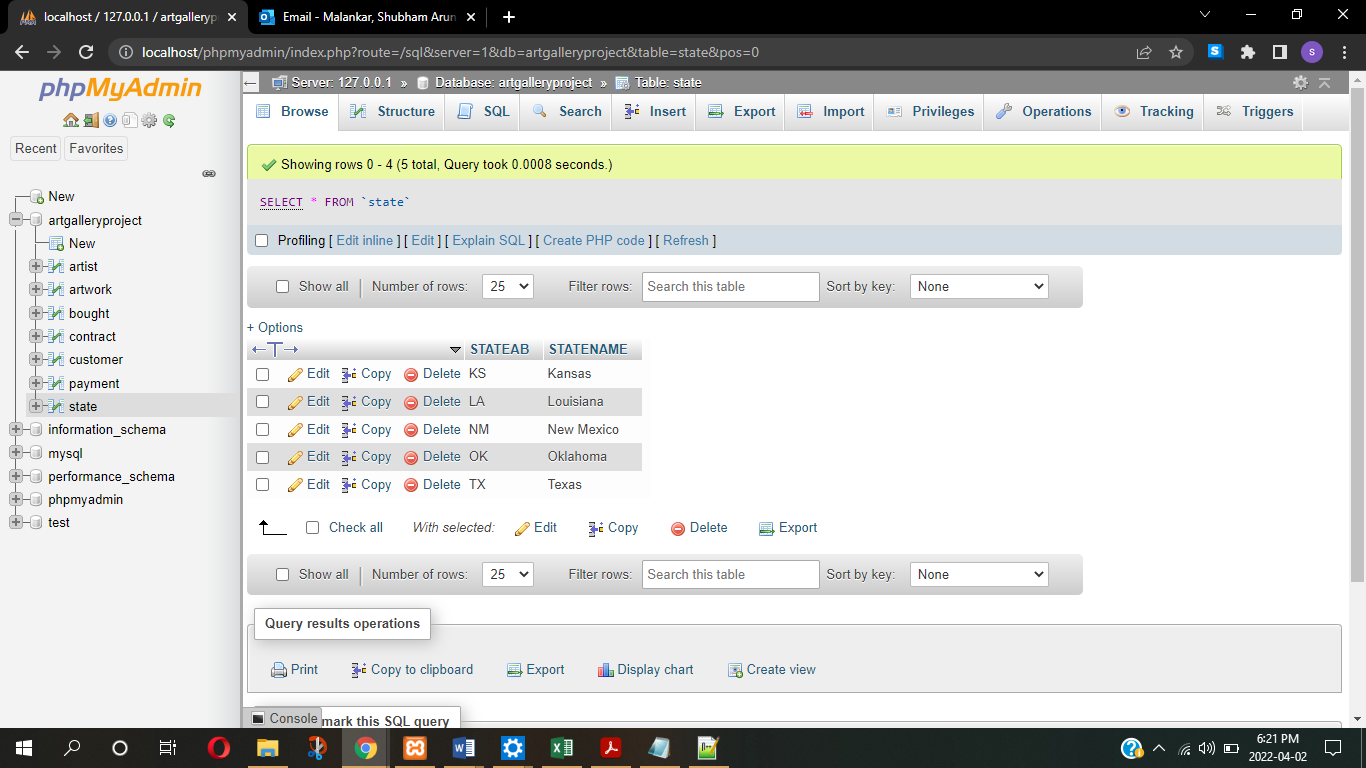
CREATE TABLE `artgalleryproject`.`state` (

`STATEAB` VARCHAR(10) NOT NULL ,

`STATENAME` TEXT NOT NULL ,

PRIMARY KEY (`STATEAB `)) ENGINE = InnoDB;





1. ‘artist’ table –

CREATE TABLE `artgalleryproject`.`artist` (

`AID` INT(50) NOT NULL ,

`NAME` TEXT NOT NULL ,

`BIRTHDATE` DATE NOT NULL ,

`DEATHDATE` DATE NULL DEFAULT NULL ,

`COMMISSION` INT(50) NOT NULL ,

`STREET` VARCHAR(500) NOT NULL ,

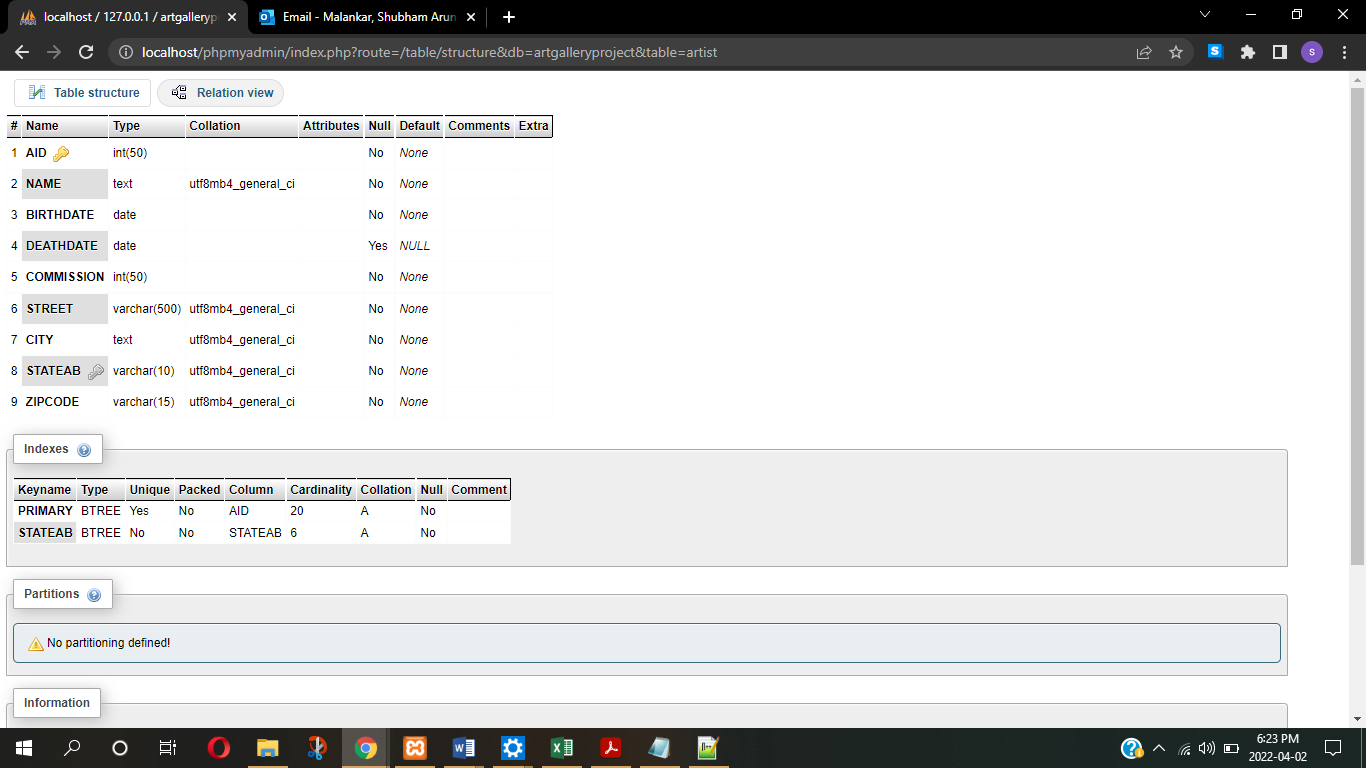
`CITY` TEXT NOT NULL ,

`STATEAB` VARCHAR(10) NOT NULL ,

`ZIPCODE` VARCHAR(15) NOT NULL ,

PRIMARY KEY (`AID`),

FOREIGN KEY(STATEAB) REFERENCES state(STATEAB)) ENGINE = InnoDB;





1. ‘customer’ table –

CREATE TABLE `artgalleryproject`.`customer` (

`CID` INT(50) NOT NULL ,

`NAME` TEXT NOT NULL ,

`STREET` VARCHAR(500) NOT NULL ,

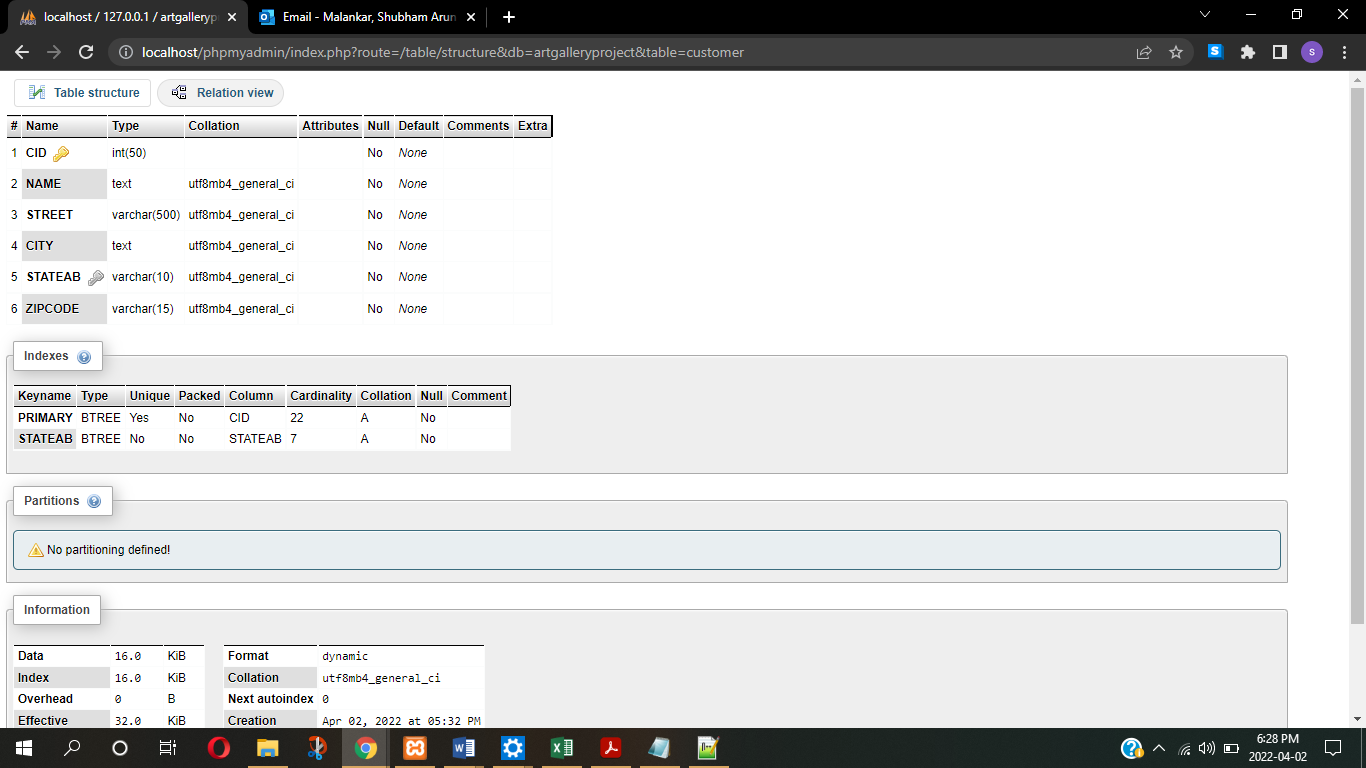
`CITY` TEXT NOT NULL ,

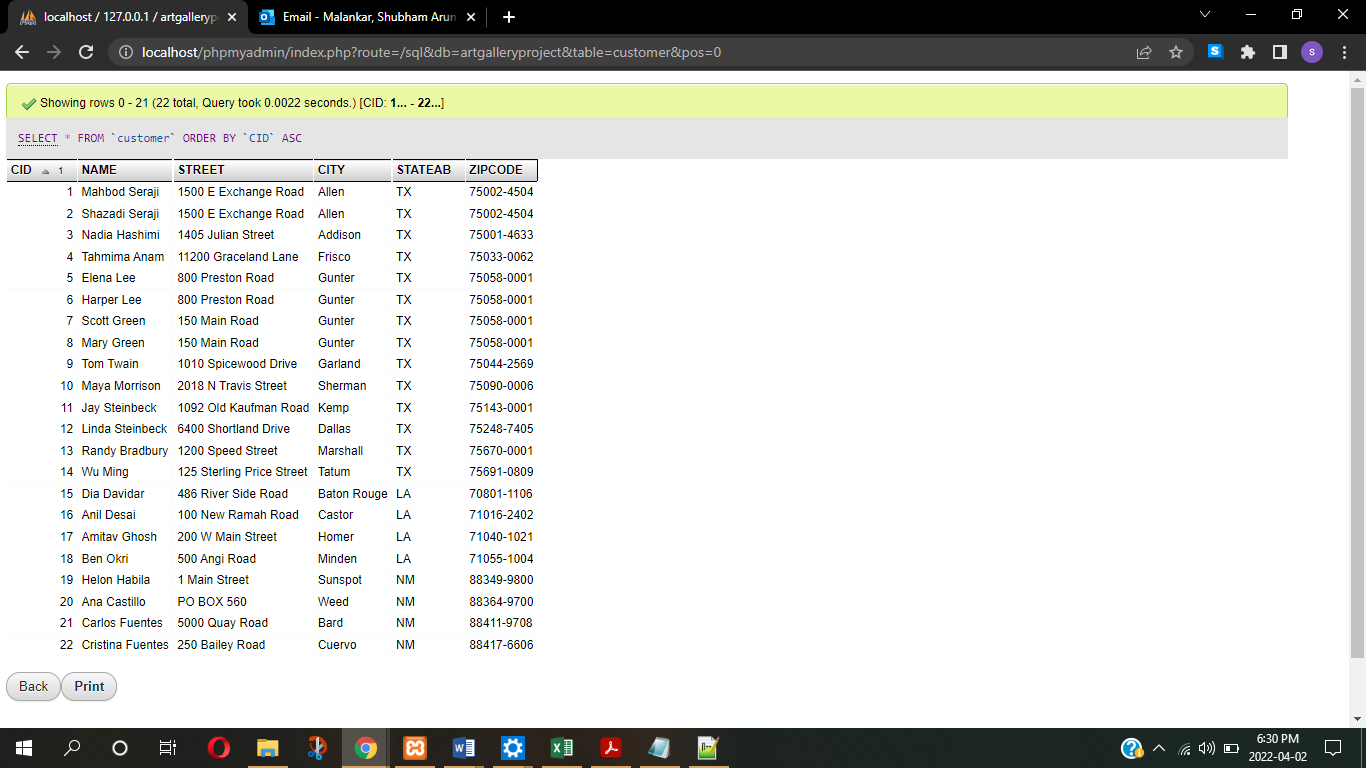
`STATEAB` VARCHAR(10) NOT NULL ,

`ZIPCODE` INT(10) NOT NULL ,

PRIMARY KEY (`CID`),

FOREIGN KEY(STATEAB) REFERENCES state(STATEAB)) ENGINE = InnoDB;





1. ‘artwork’ table –

CREATE TABLE `artgalleryproject`.`artwork` (

`AID` INT(50) NOT NULL ,

`ARTID` INT(50) NOT NULL ,

`TITLE` TEXT NOT NULL ,

`CREATIONDATE` DATE NOT NULL ,

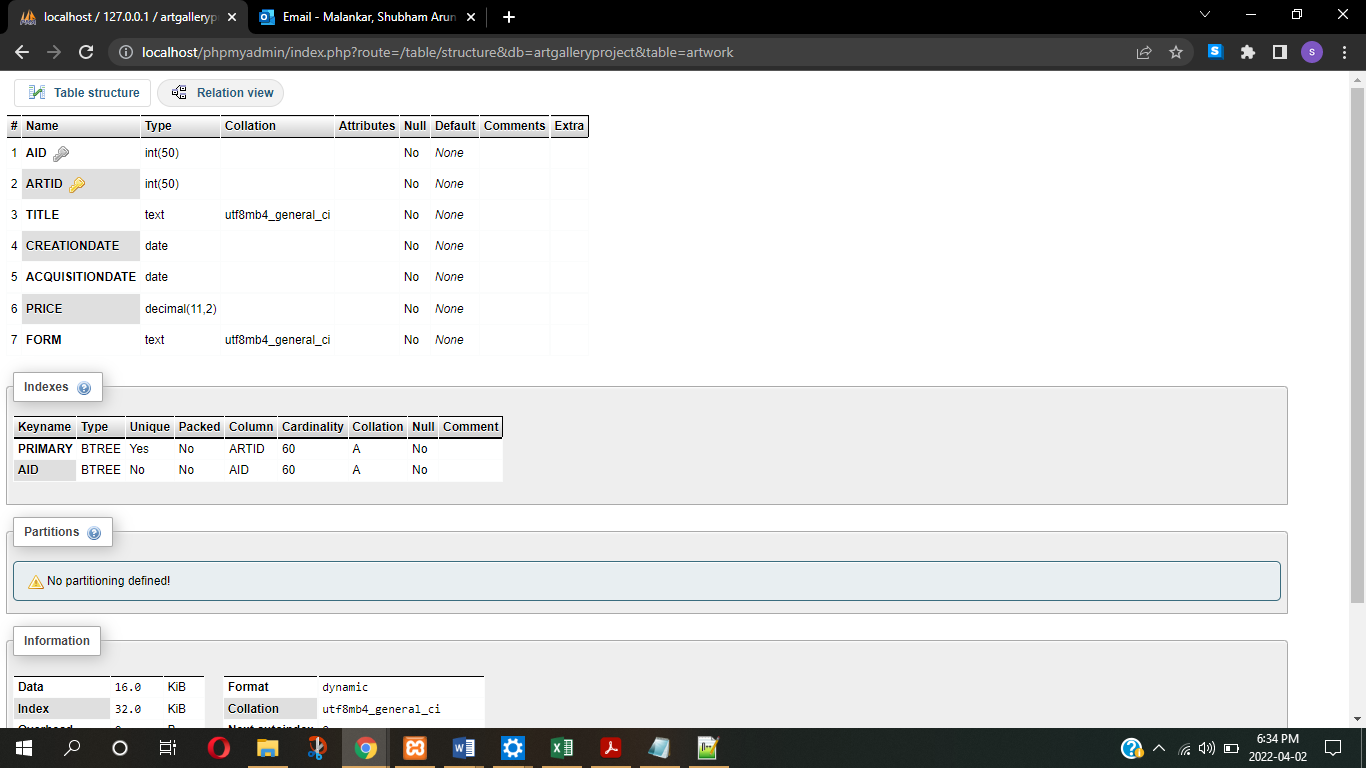
`ACQUISITIONDATE` DATE NOT NULL ,

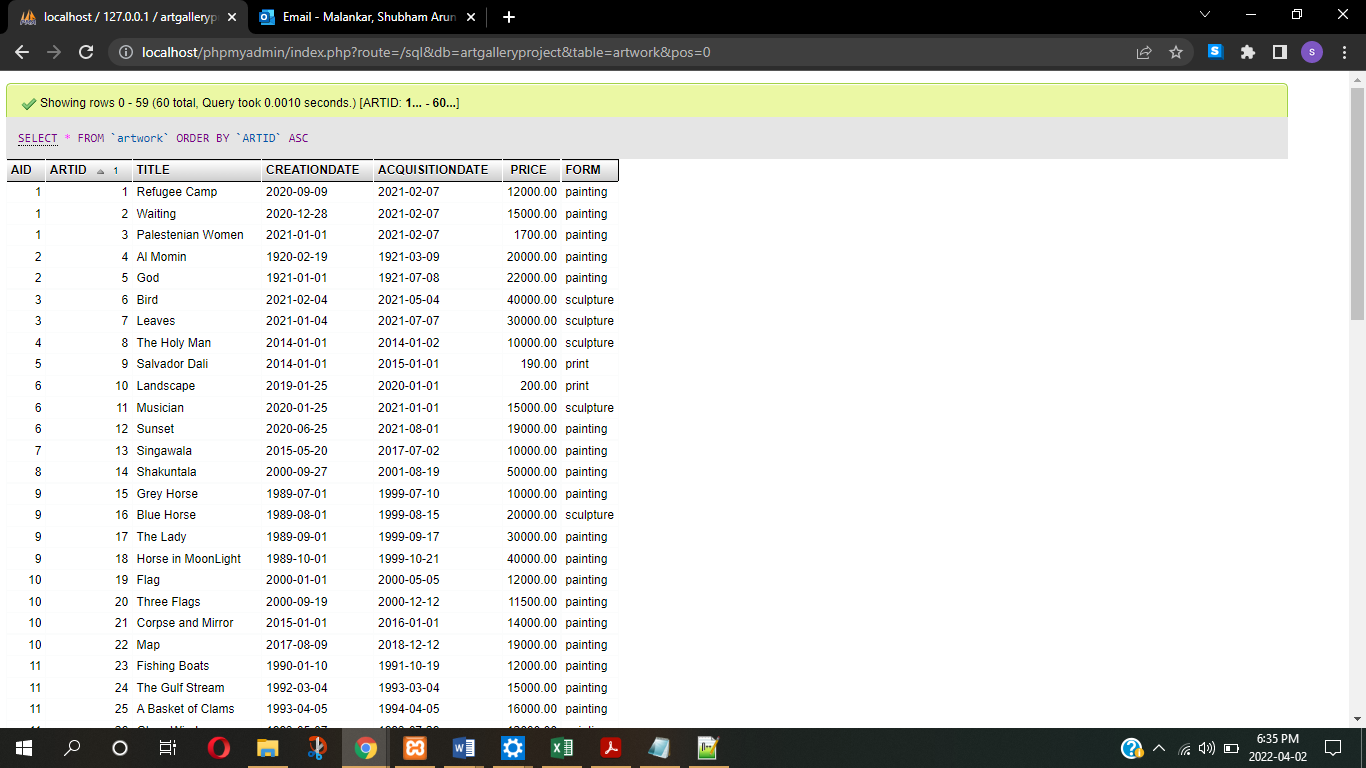
`PRICE` DECIMAL(11,2) NOT NULL ,

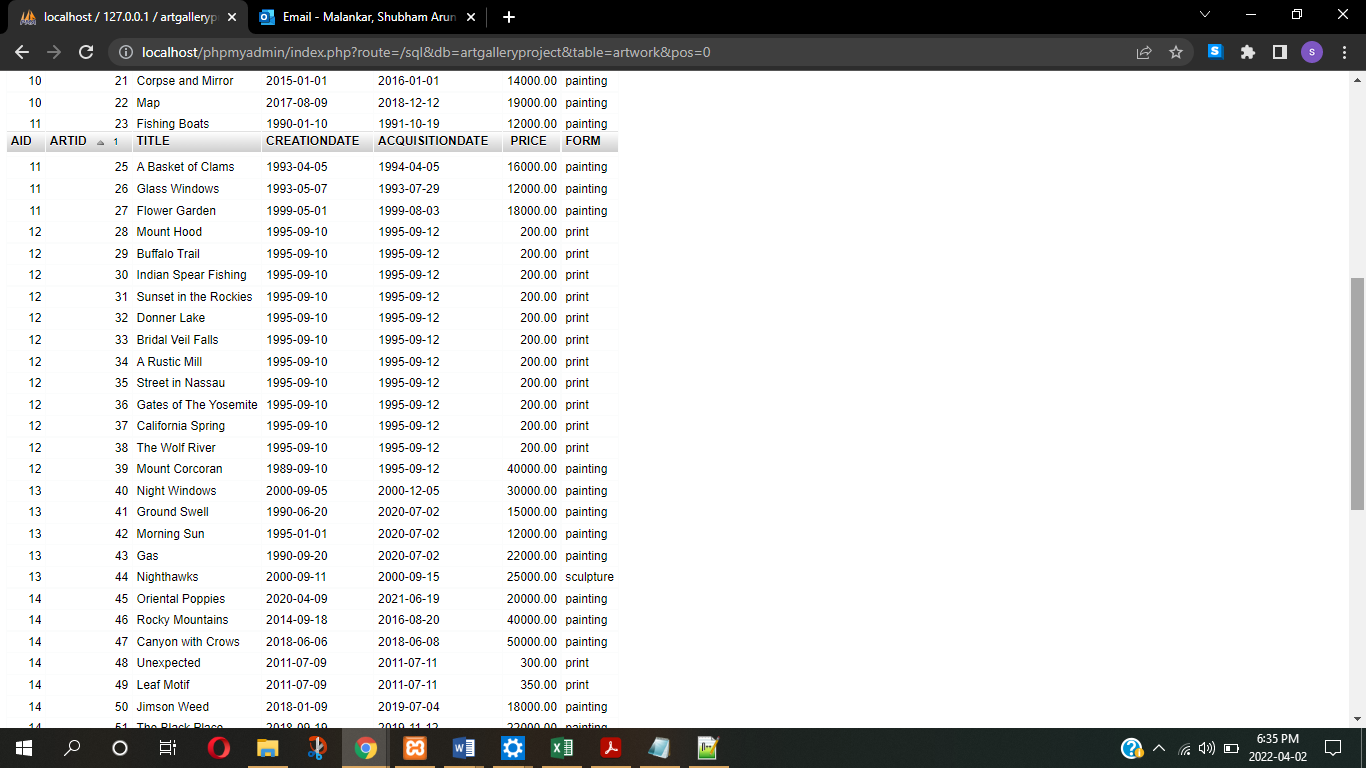
`FORM` TEXT NOT NULL ,

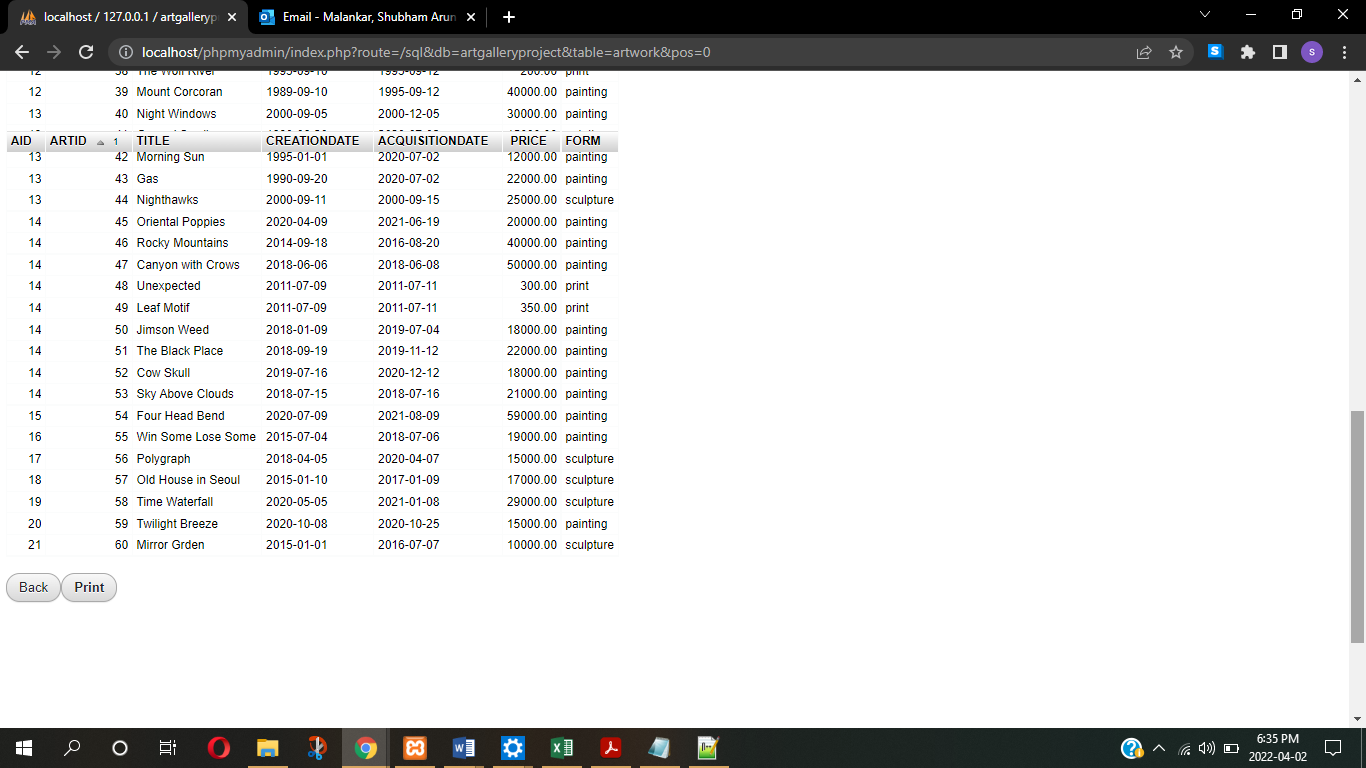
PRIMARY KEY (`ARTID`) ,

FOREIGN KEY(aid) REFERENCES artist(AID)) ENGINE = InnoDB;









1. ‘contract’ table –

CREATE TABLE `artgalleryproject`.`contract` (

`AID` INT(50) NOT NULL ,

`COID` INT(50) NOT NULL ,

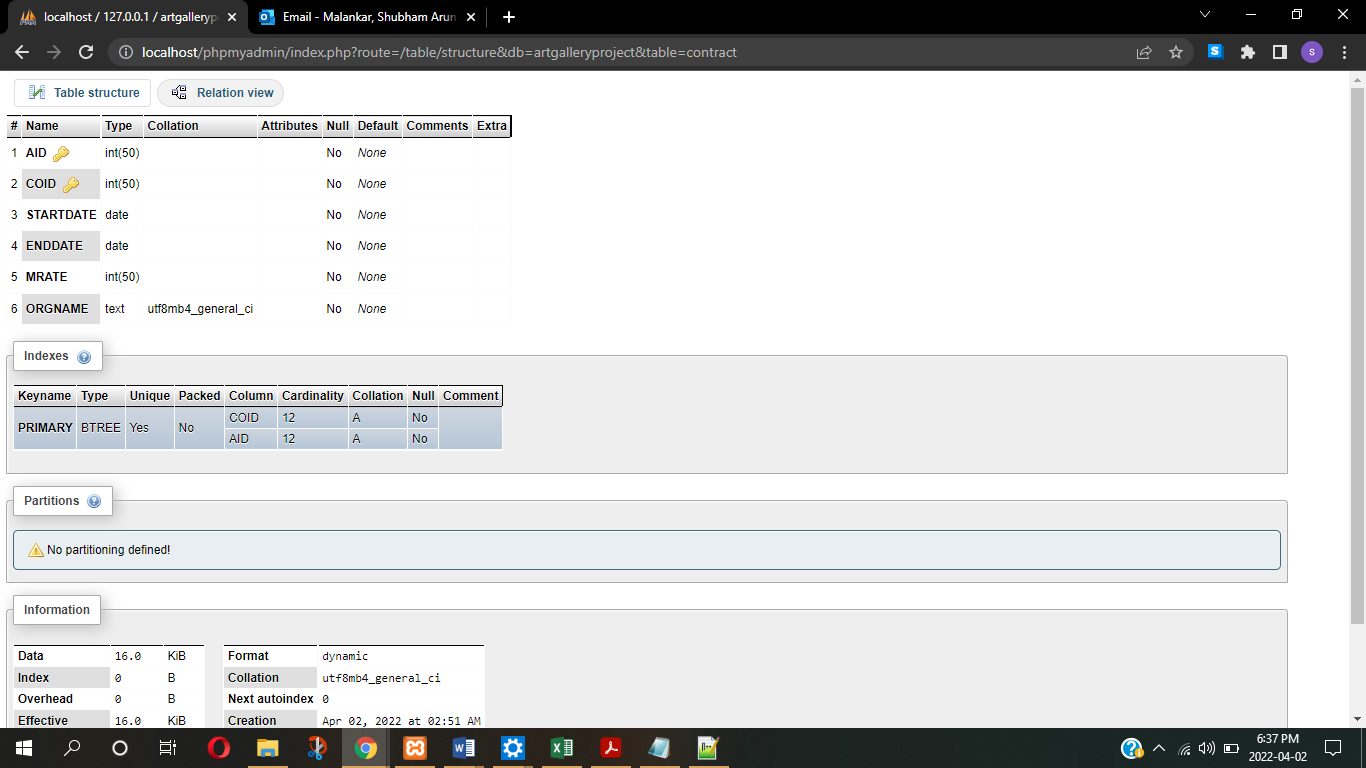
`STARTDATE` DATE NOT NULL ,

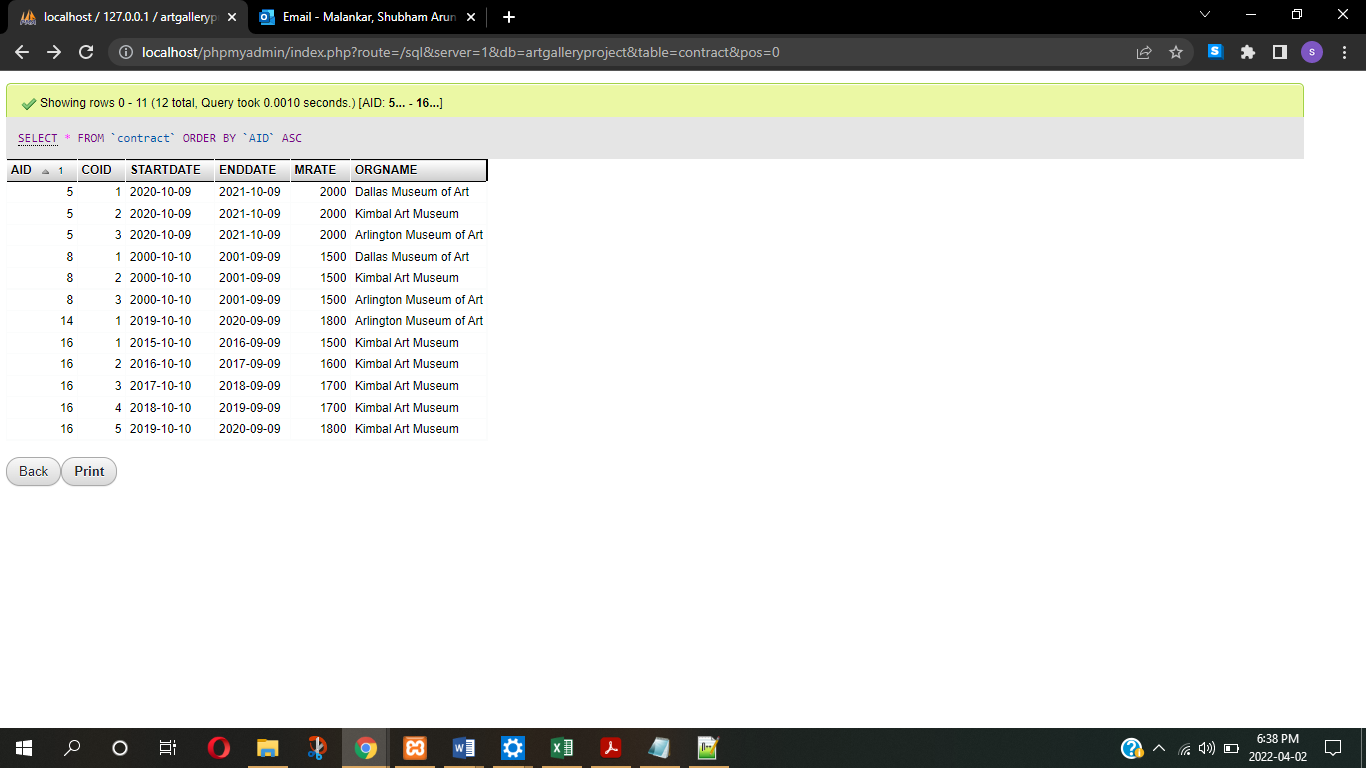
`ENDDATE` DATE NOT NULL ,

`MRATE` INT(50) NOT NULL ,

`ORGNAME` TEXT NOT NULL ,

PRIMARY KEY (`COID`, `AID`)) ENGINE = InnoDB;





1. ‘bought’ table –

CREATE TABLE `artgalleryproject`.`bought` (

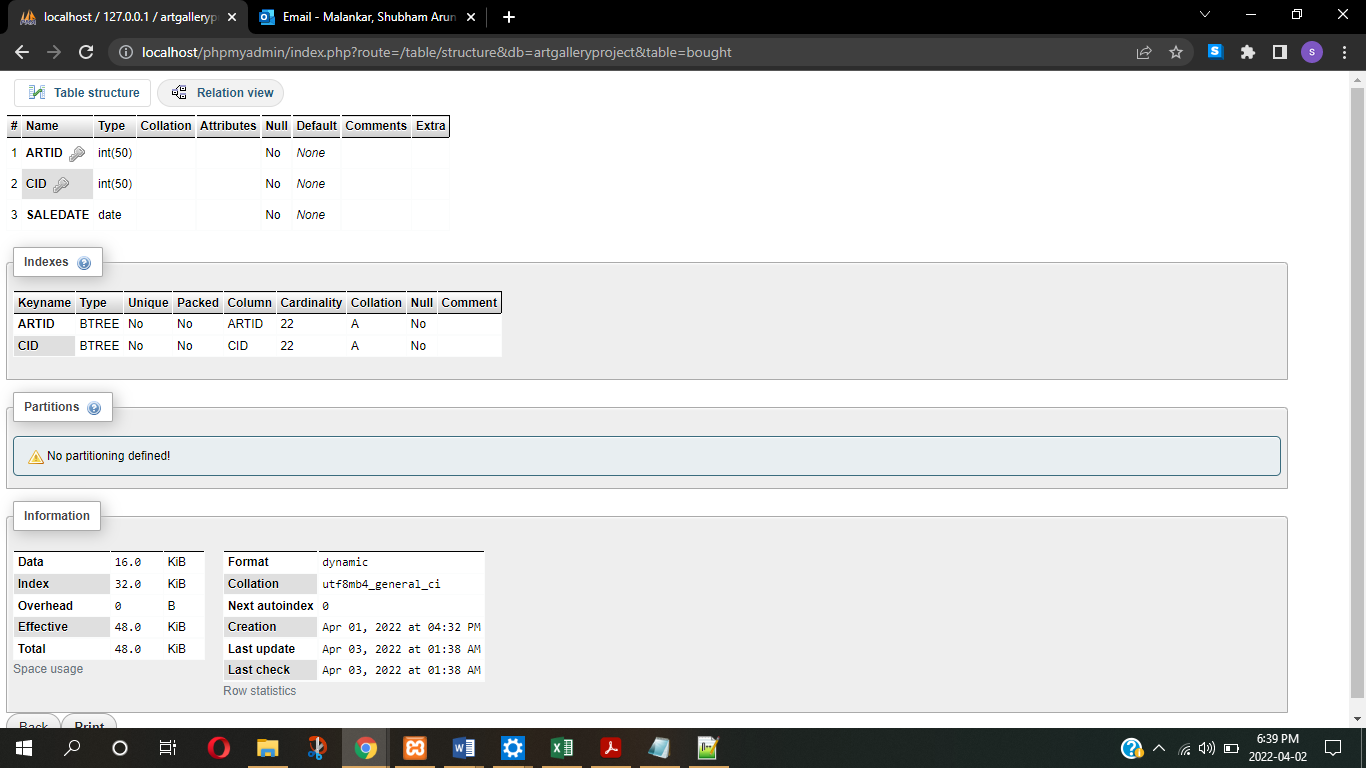
`ARTID` INT(50) NOT NULL ,

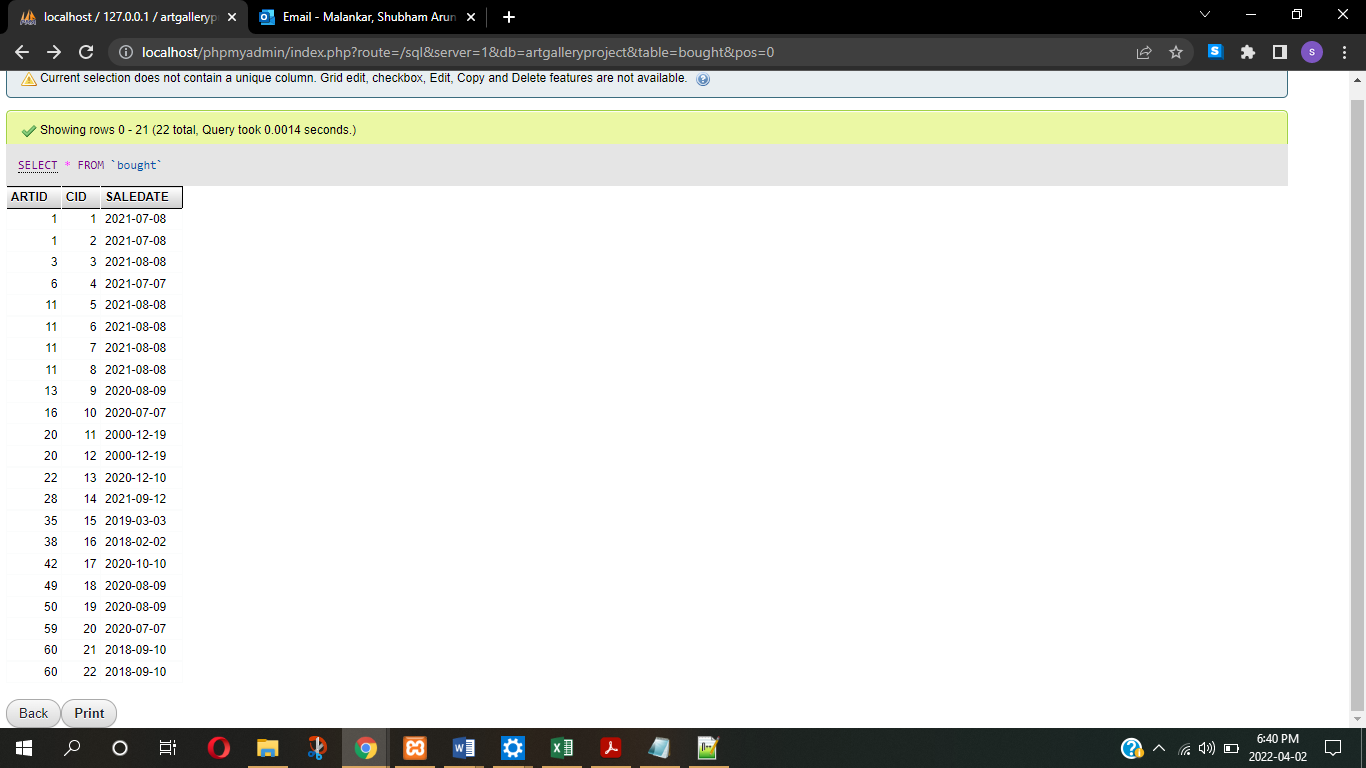
`CID` INT(50) NOT NULL ,

`SALEDATE` DATE NOT NULL ,

FOREIGN KEY(ARTID) REFERENCES artwork(ARTID),

FOREIGN KEY(CID) REFERENCES customer(CID)) ENGINE = InnoDB;





1. ‘payment’ table –

CREATE TABLE `artgalleryproject`.`payment` (

`PID` INT(50) NOT NULL ,

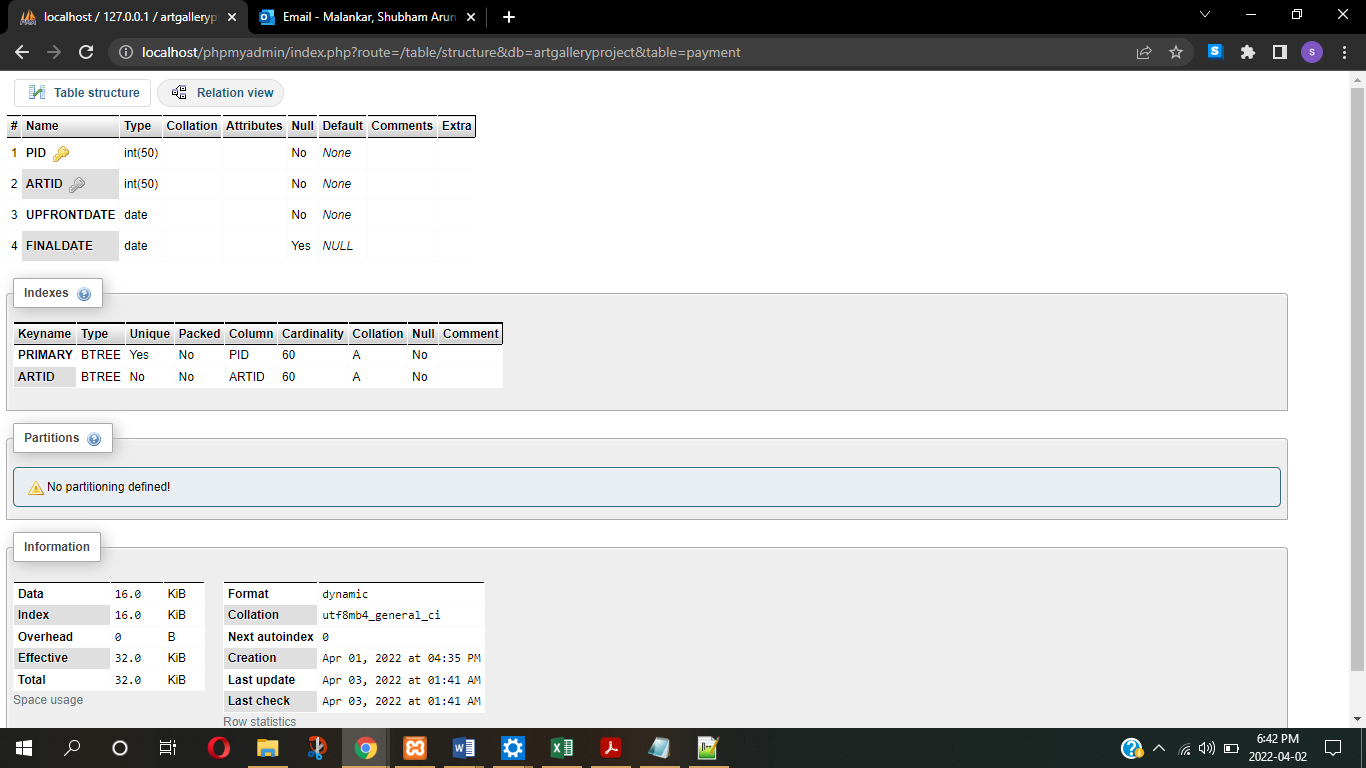
`ARTID` INT(50) NOT NULL ,

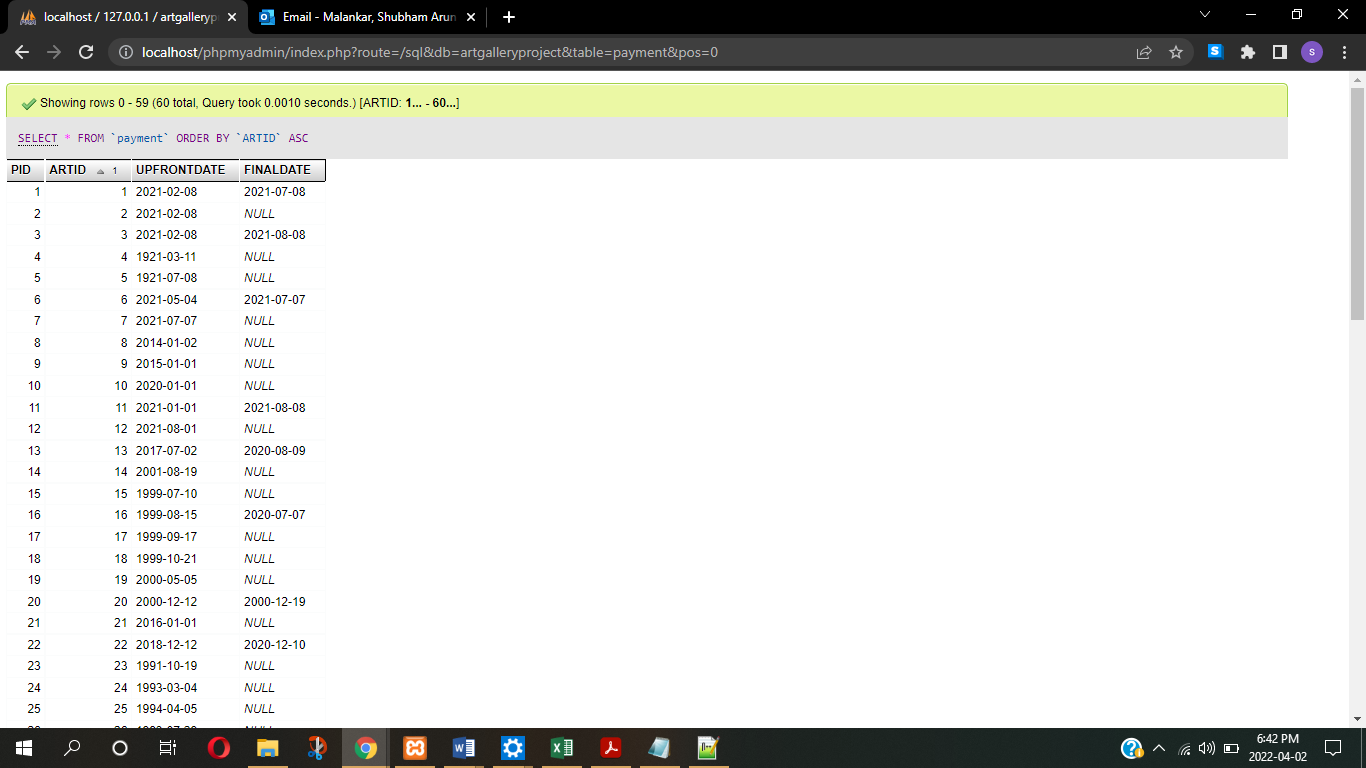
`UPFRONTDATE` DATE NOT NULL ,

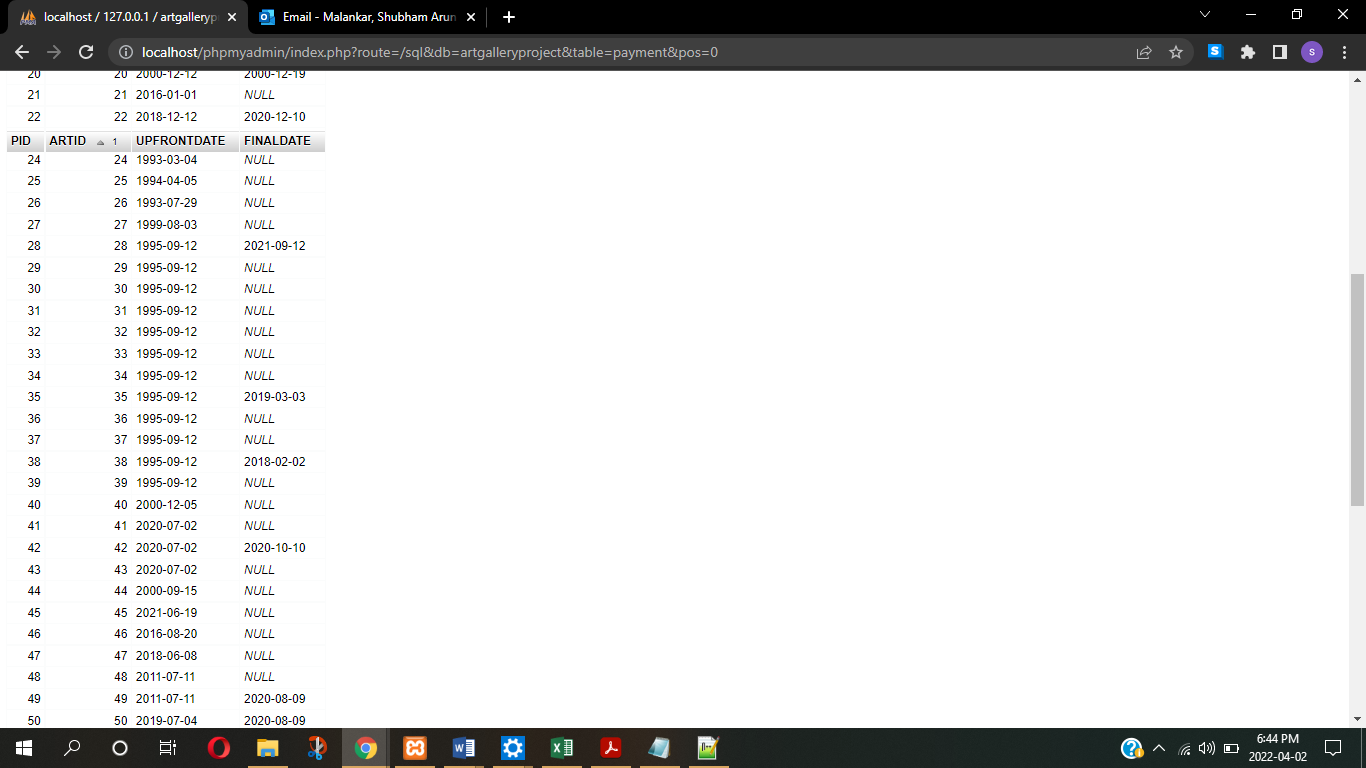
`FINALDATE` DATE NULL DEFAULT NULL ,

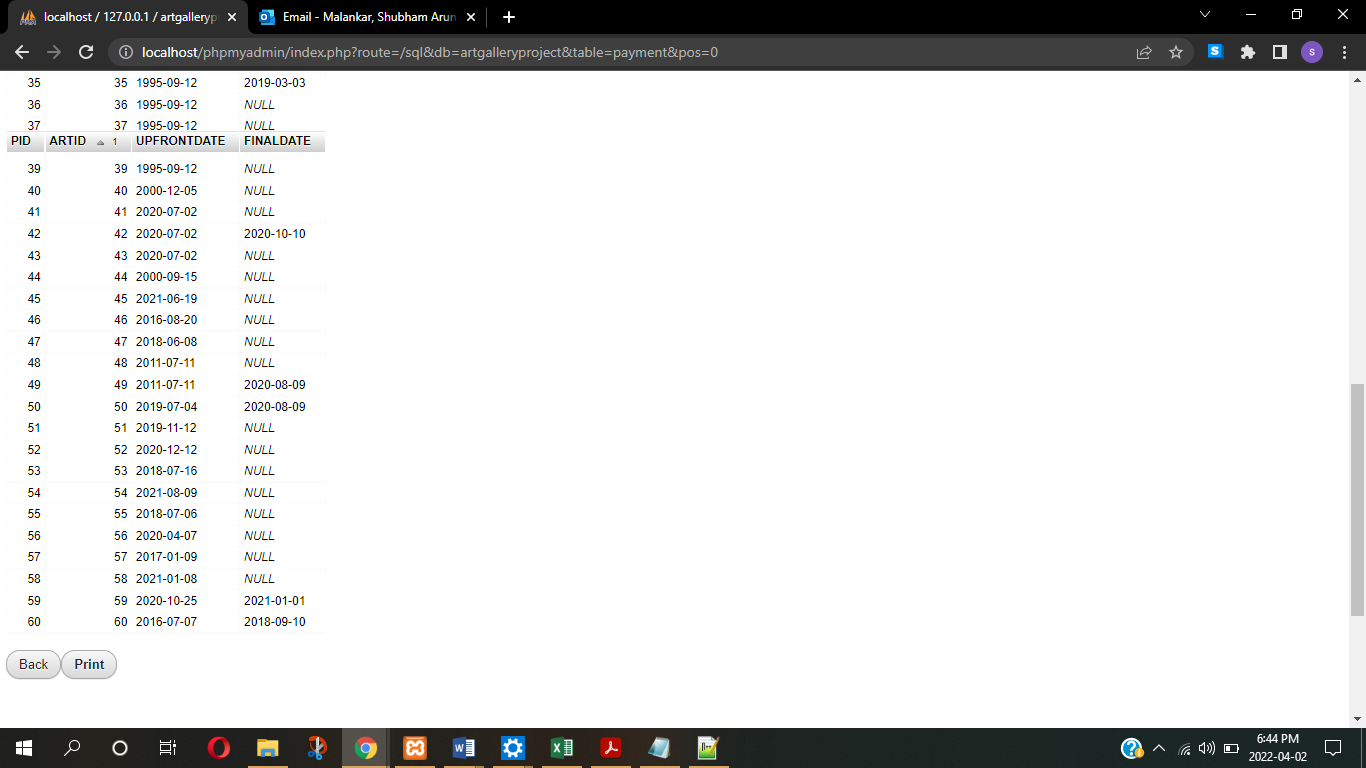
PRIMARY KEY (`PID`),

FOREIGN KEY(ARTID) REFERENCES artwork(ARTID)) ENGINE = InnoDB;





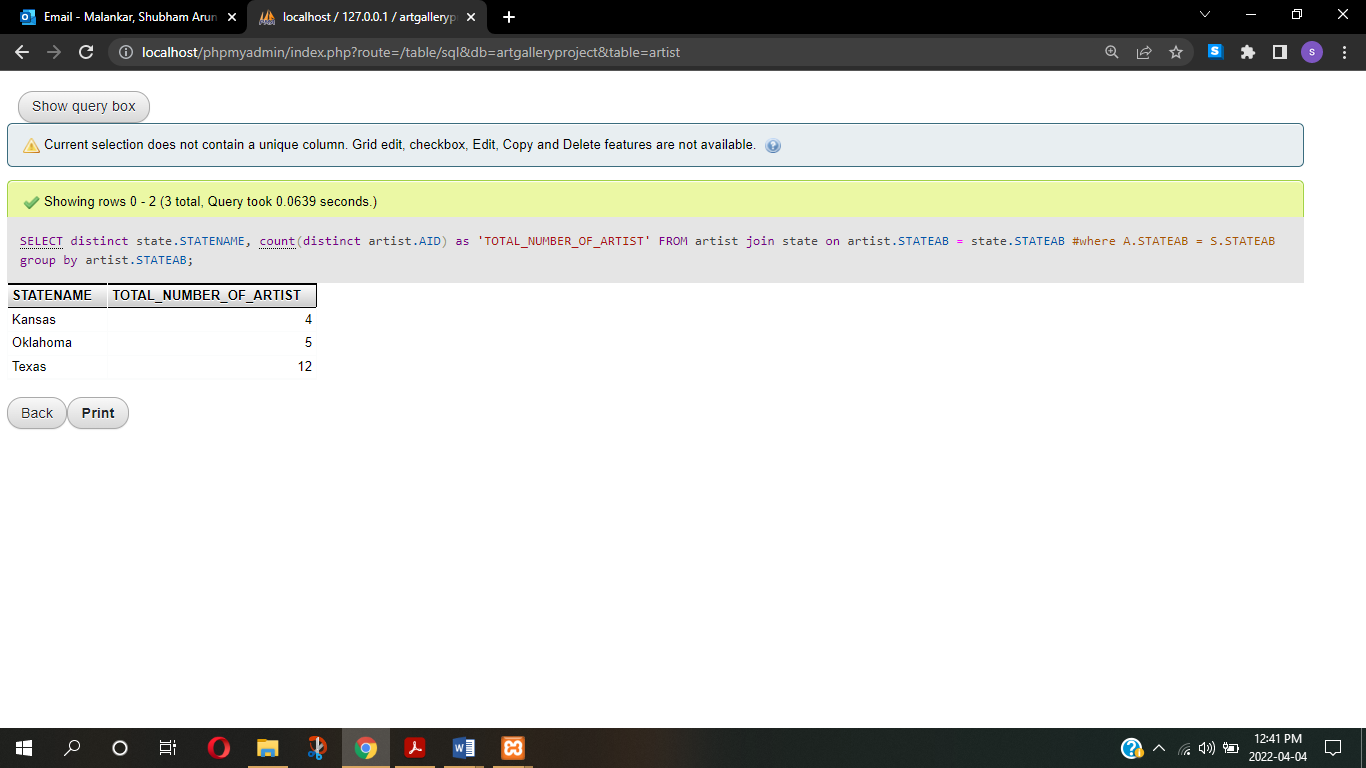




**PART 2 : -**

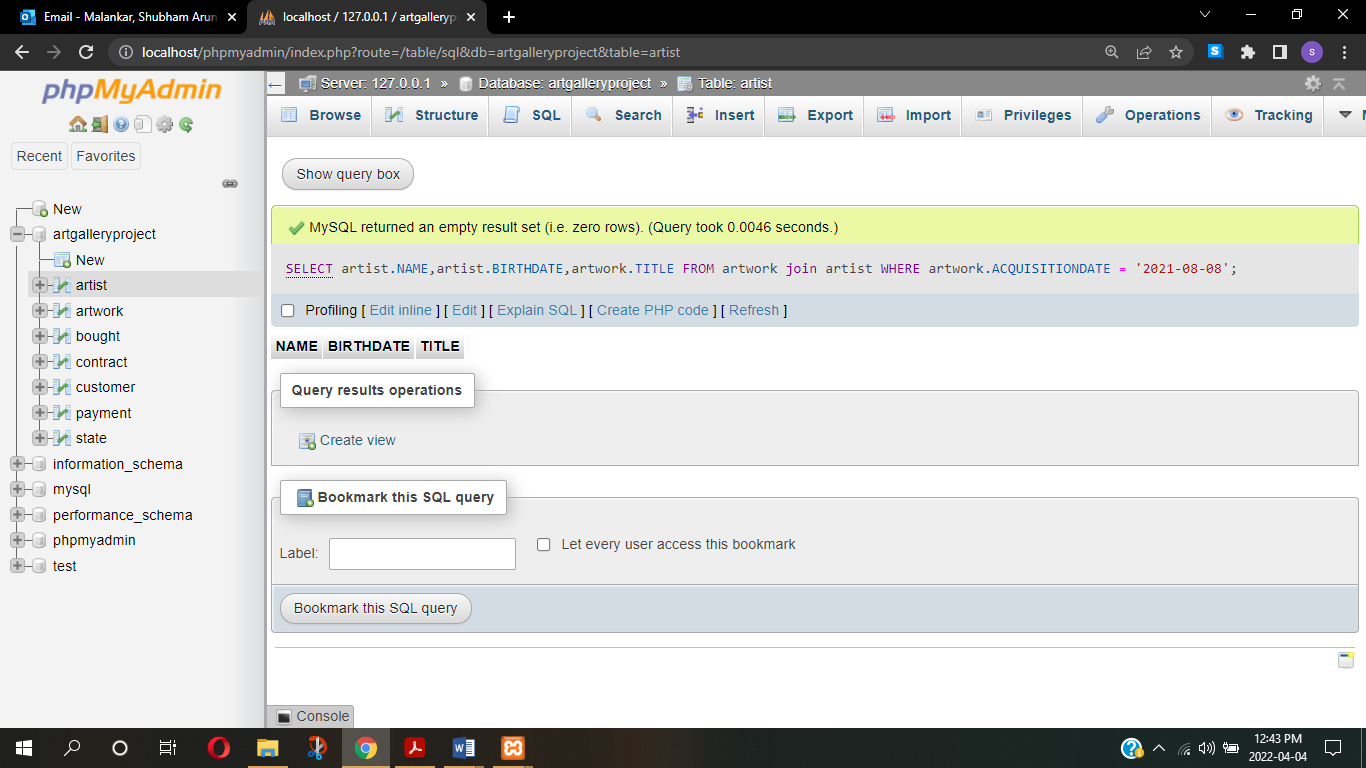
**Q1** Retrieve the total number of artists in each state along with the state name (not the state abbreviation).

**Ans –** SELECT distinct S.STATENAME, count(distinct A.AID) as 'TOTAL\_NUMBER\_OF\_ARTIST' FROM artist JOIN state on artist.STATEAB = state.STATEAB group by A.STATEAB



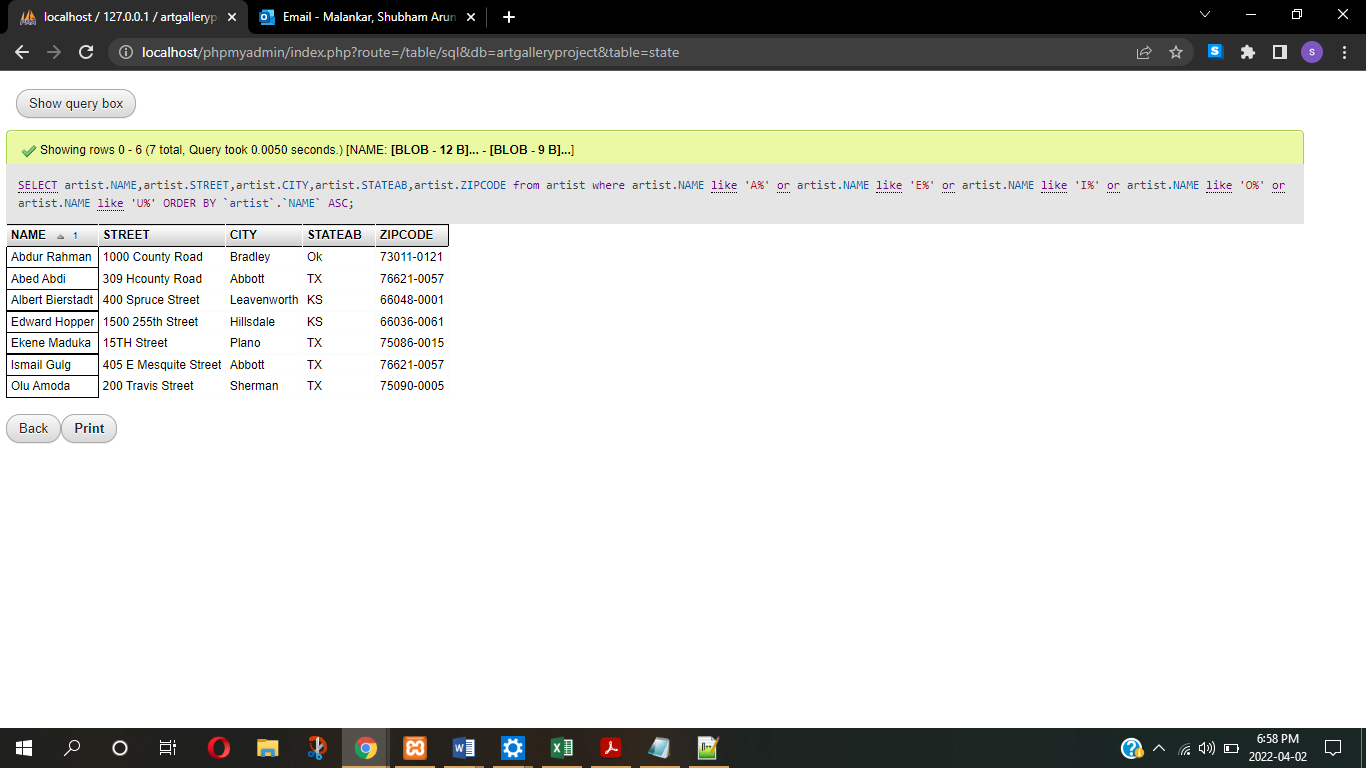
**Q2** Retrieve the artist’s name, date of birth, and the title of artwork that was sold on August 8, 2021.

**Ans –** SELECT artist.NAME, artist.BIRTHDATE, artwork.TITLE FROM artwork join artist WHERE artwork.ACQUISITIONDATE = '2021-08-08'



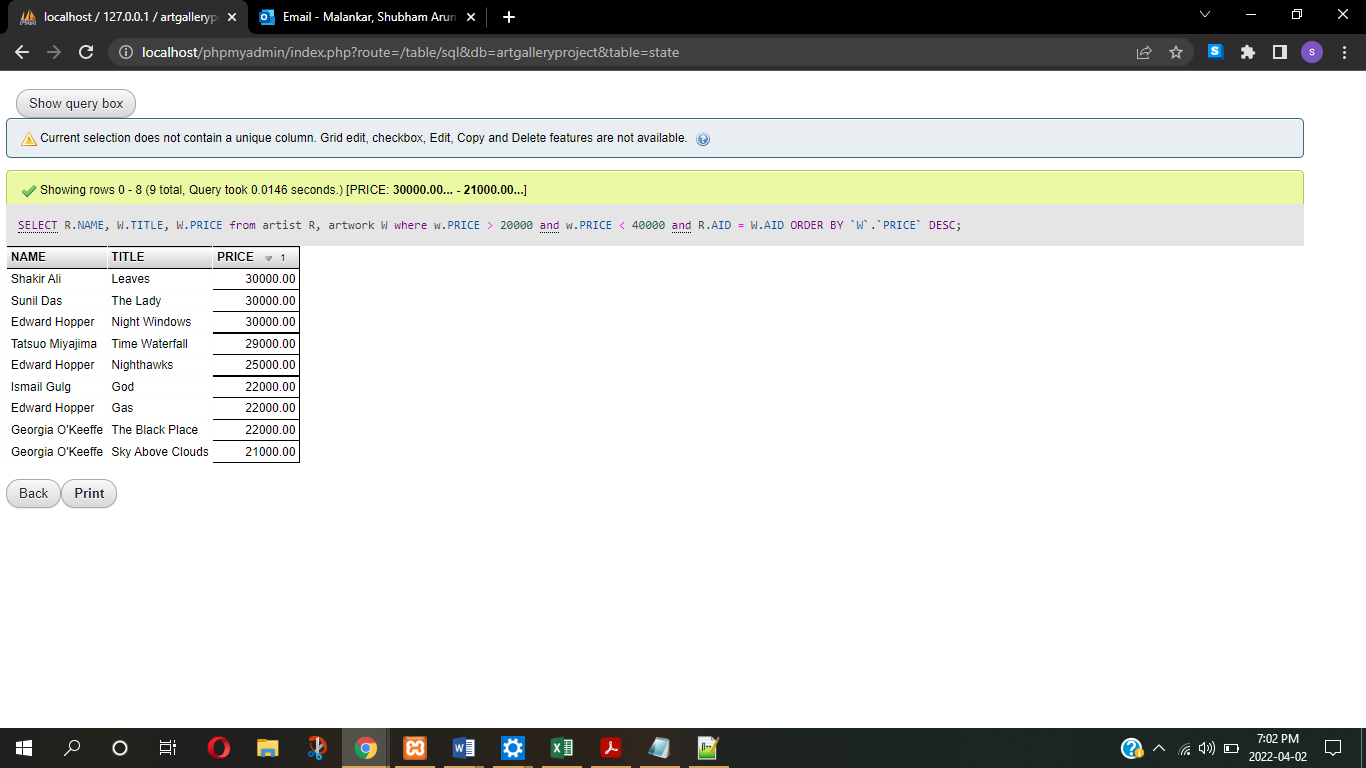
**Q3** Retrieve the names and address (Street, City, State Abbreviation, and Zipcode) of all the artists in our database whose first name start with a vowel.

**Ans –** SELECT artist.NAME,artist.STREET,artist.CITY,artist.STATEAB,artist.ZIPCODE from artist where artist.NAME like 'A%' or artist.NAME like 'E%' or artist.NAME like 'I%' or artist.NAME like 'O%' or artist.NAME like 'U%' ORDER BY `artist`.`NAME` ASC



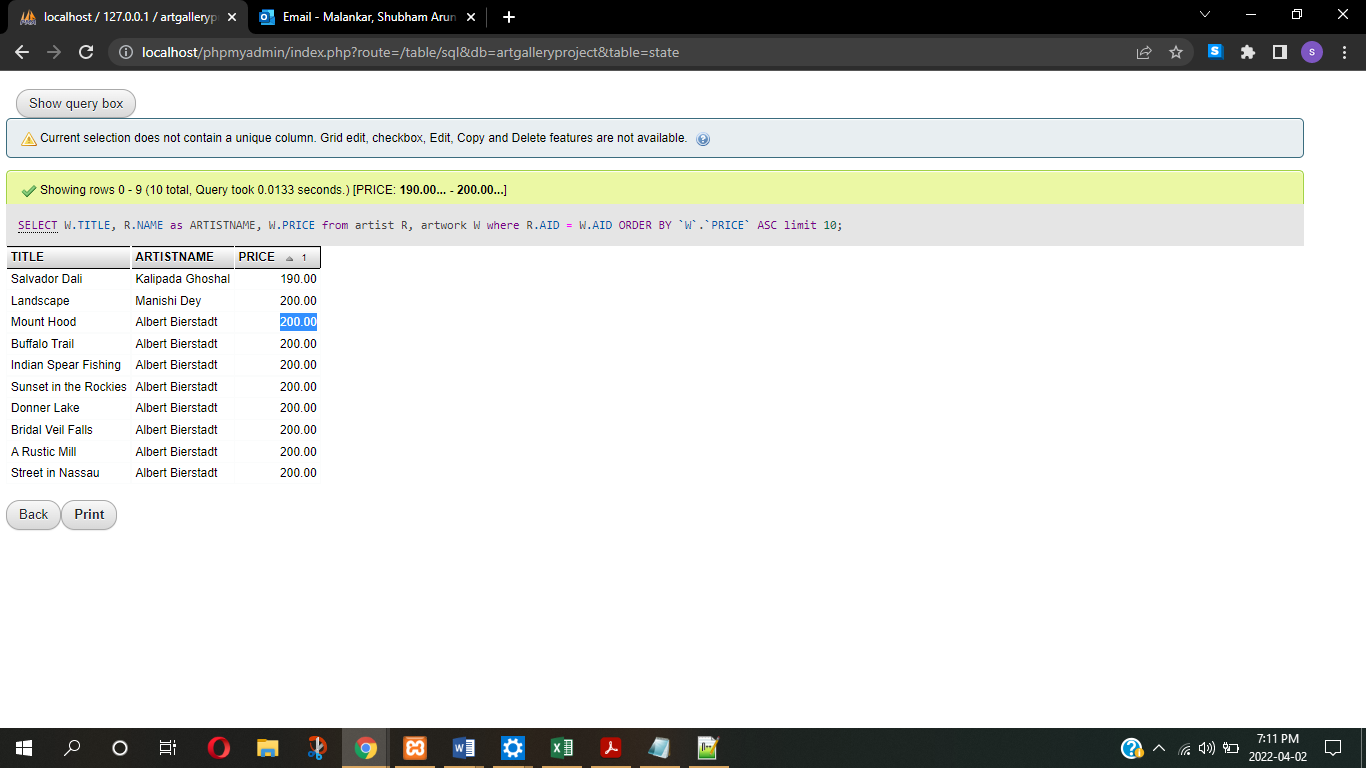
**Q4** Retrieve the names of the artists, list of their artwork (title) when the acquisition price is higher than $20,000 and less than $40,000. Order the result by the price of the artworks.

**Ans –** SELECT R.NAME, W.TITLE, W.PRICE from artist R, artwork W where w.PRICE > 20000 and w.PRICE < 40000 and R.AID = W.AID ORDER BY `W`.`PRICE` DESC



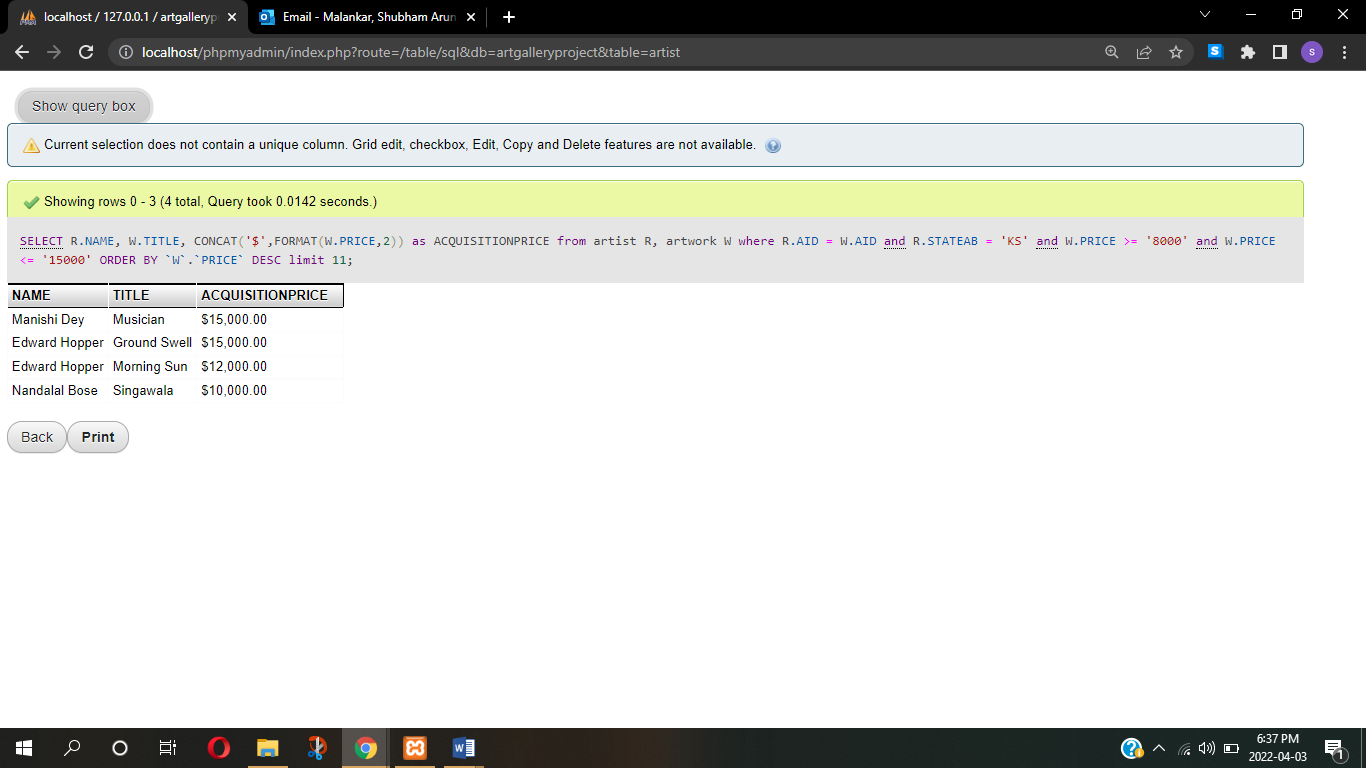
**Q5** Retrieve the titles, acquisition price, and the artists names of the 10 least expensive artworks in the gallery.

**Ans –** SELECT W.TITLE, R.NAME as ARTISTNAME, W.PRICE from artist R, artwork W where R.AID = W.AID ORDER BY `W`.`PRICE` ASC limit 10;



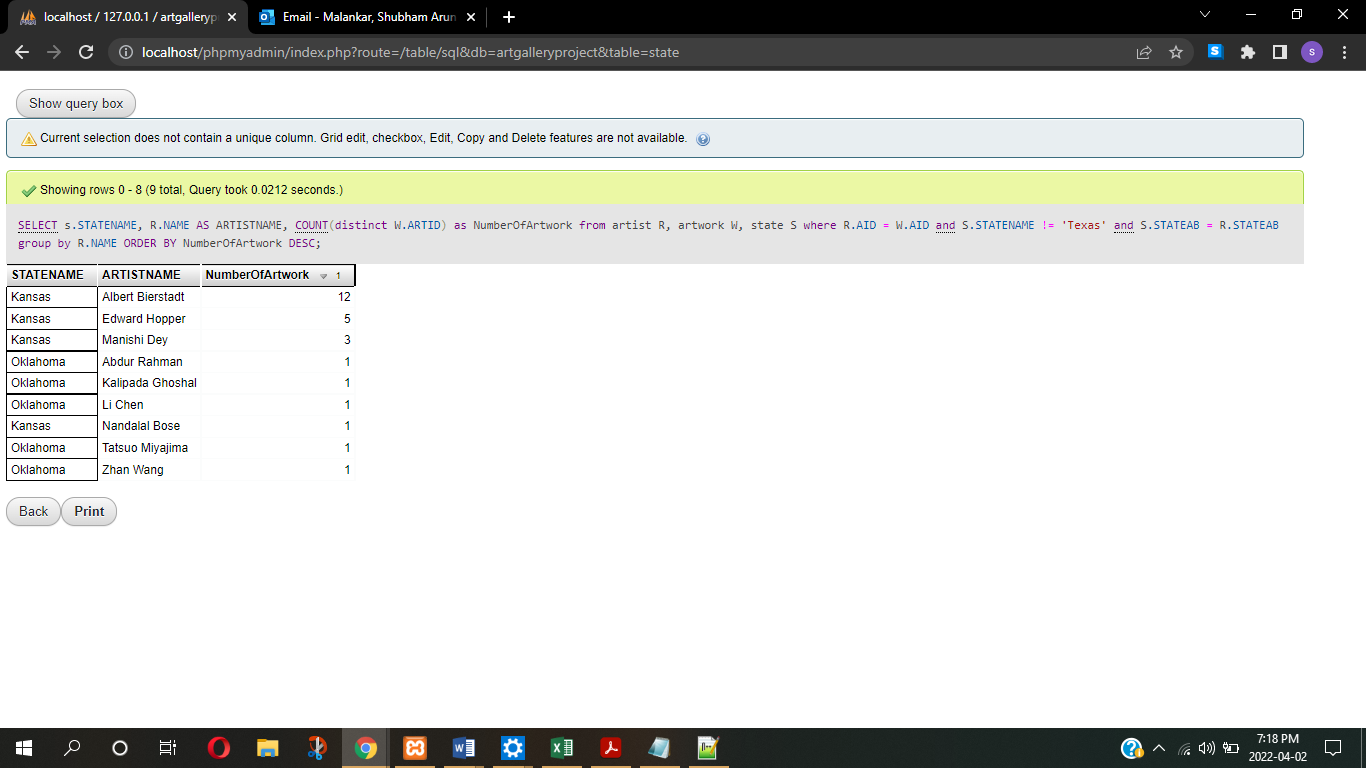
**Q6** Retrieve the artist name, artwork title, and the acquisition price for all artworks of artists who hale from Kansas and has an artwork whose acquisition price is between $8,000.00 and $15,000.00. List the result in a descending order by the price of the artwork. The price should be stated with $, appropriate commas, and up to 2 decimal places. (Example: $12,000.00)

**Ans –** SELECT R.NAME, W.TITLE, CONCAT('$',FORMAT(W.PRICE,2)) as ACQUISITIONPRICE from artist R, artwork W where R.AID = W.AID and R.STATEAB = 'KS' and W.PRICE >= '8000' and W.PRICE <= '15000' ORDER BY `W`.`PRICE` DESC;



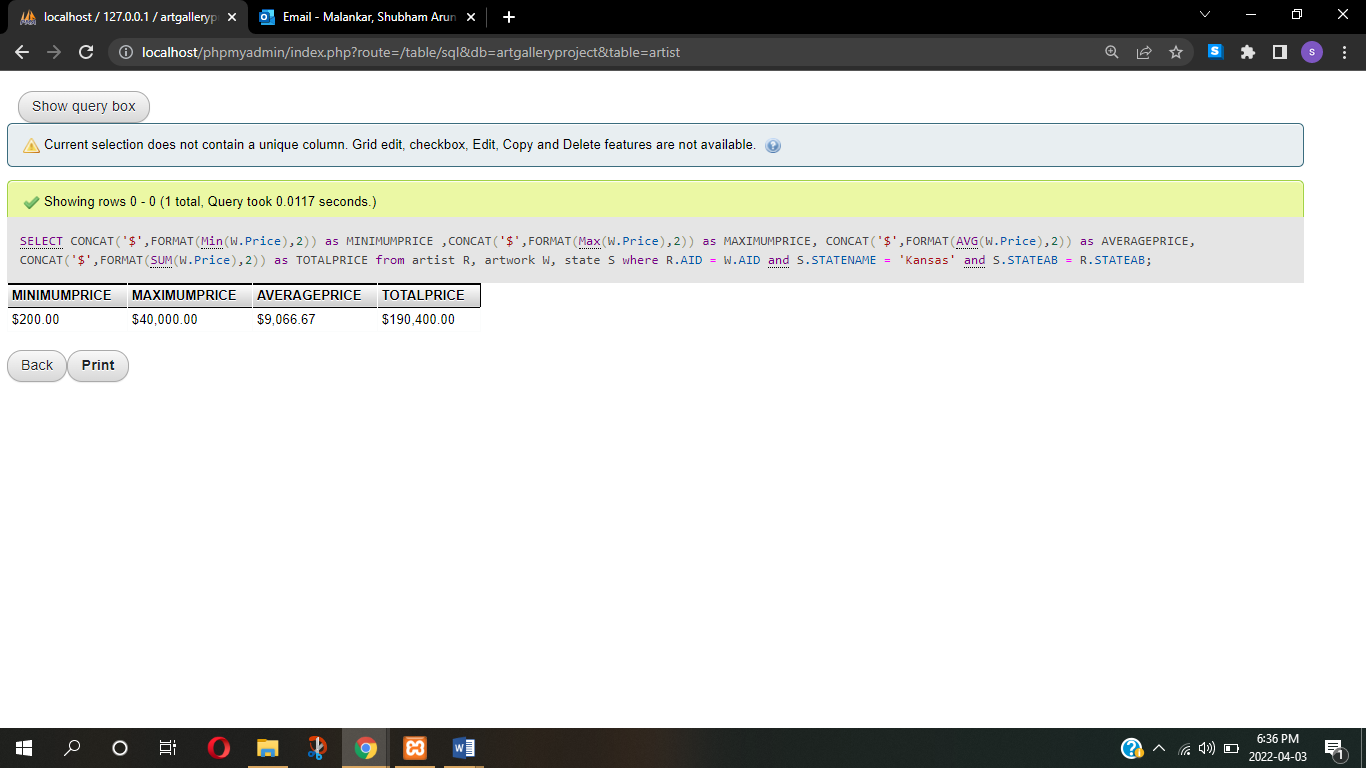
**Q7** Retrieve a list of state names, artist names from each of the states, and the total number of artworks by each artist to have been listed in the Art Gallery. Order the list by the number of their artworks. Exclude Texas artists from the list.

**Ans -** SELECT s.STATENAME, R.NAME as ARTISTNAME, COUNT(distinct W.ARTID) as NumberOfArtwork from artist R, artwork W, state S where R.AID = W.AID and S.STATENAME != 'Texas' and S.STATEAB = R.STATEAB group by R.NAME ORDER BY NumberOfArtwork DESC;



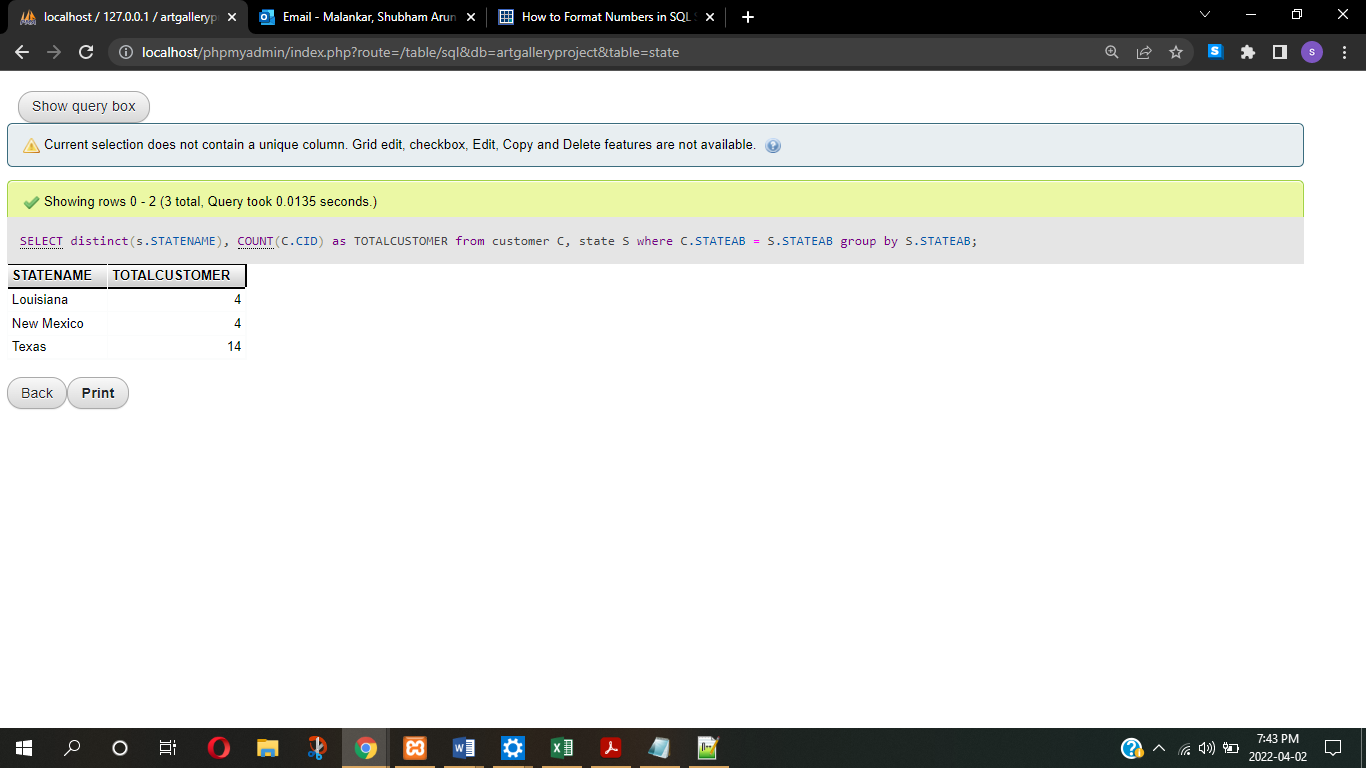
**Q8** Retrieve, the minimum, maximum, average, and total price of listed artwork of the artists from the state of Kansas. The price should be stated with $, appropriate commas, and up to 2 decimal places. (Example: $12,000.00)

**Ans -** SELECT CONCAT('$', FORMAT(Min(W.Price),2)) as MINIMUMPRICE ,CONCAT('$', FORMAT(Max(W.Price),2)) as MAXIMUMPRICE, CONCAT('$', FORMAT([AVG](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html%23function_avg)(W.Price),2)) as AVERAGEPRICE, CONCAT('$', FORMAT(SUM(W.Price),2) as TOTALPRICE from artist R, artwork W, state S where R.AID = W.AID and S.STATENAME = 'Kansas' and S.STATEAB = R.STATEAB;



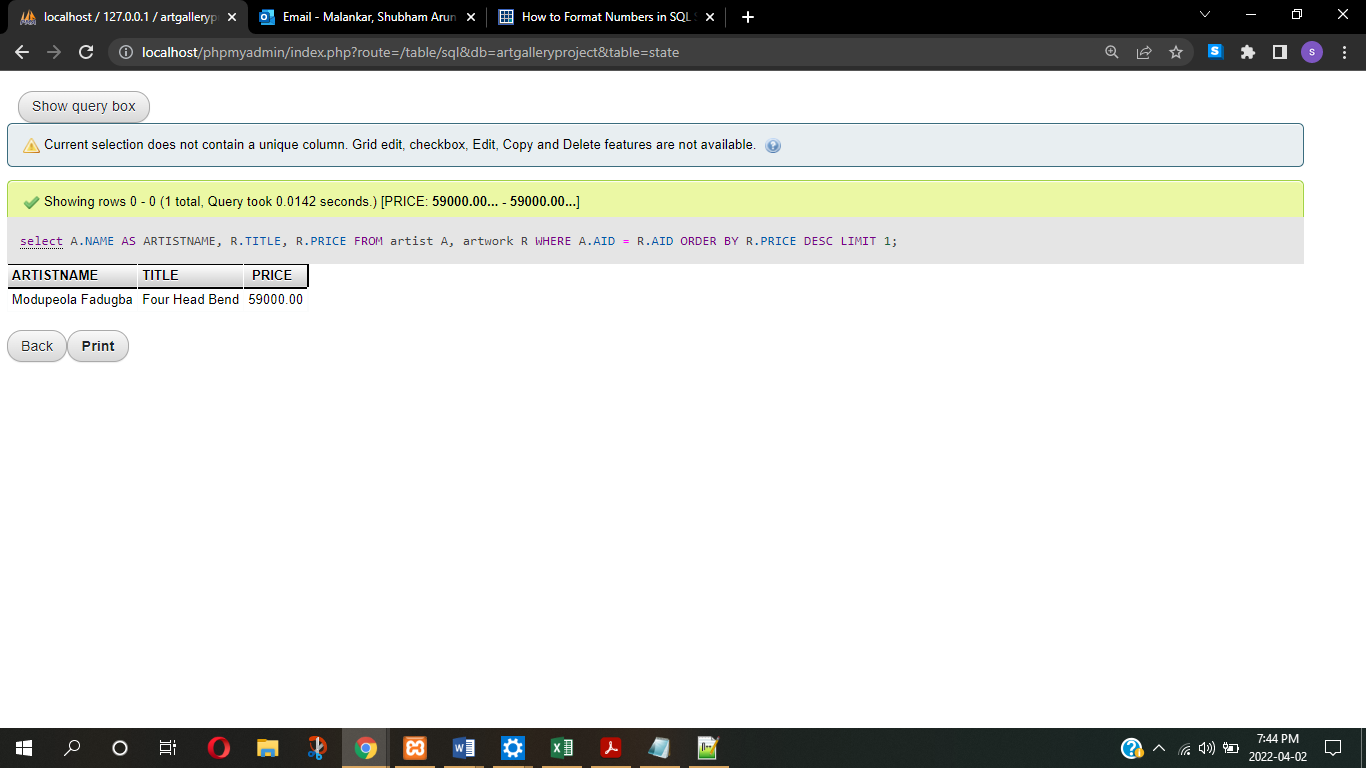
**Q9** Retrieve the states, the total number of customers from each state. (Your query will be considered incorrect if it returns the state abbreviation instead of the full state name)

**Ans –** SELECT distinct(s.STATENAME), COUNT(C.CID) as TOTALCUSTOMER from customer C, state S where C.STATEAB = S.STATEAB group by S.STATEAB;



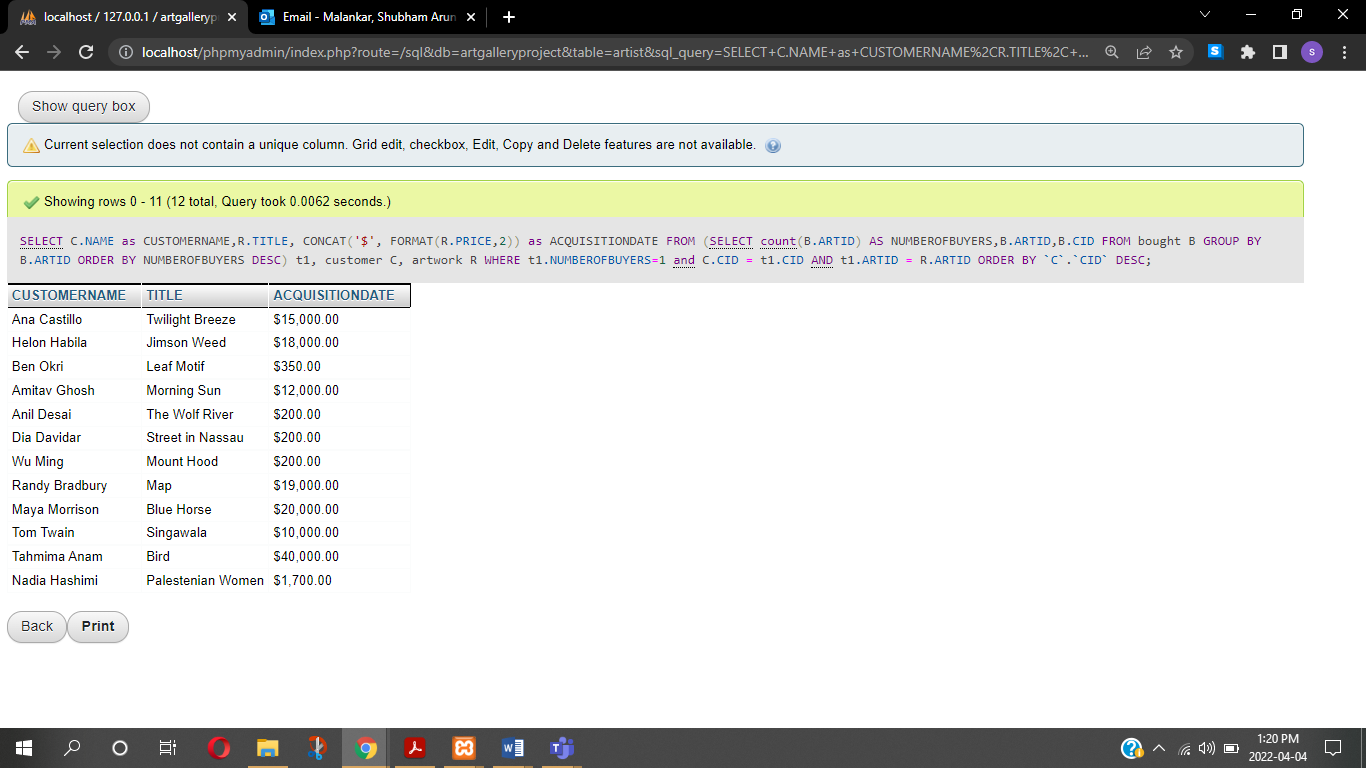
**Q10** Retrieve the artist name, artwork title and the acquisition price of the most expensive artwork acquired by the gallery so far.

**Ans –** select A.NAME as ARTISTNAME, R.TITLE, R.PRICE FROM artist A, artwork R WHERE A.AID = R.AID ORDER BY R.PRICE DESC LIMIT 1;



**Q11** Retrieve a list of records that each consists of the customer’s name, and the title of the artwork they bought, and their price for all the artwork that was bought by only one customer. How will you verify that the results of your query are correct?

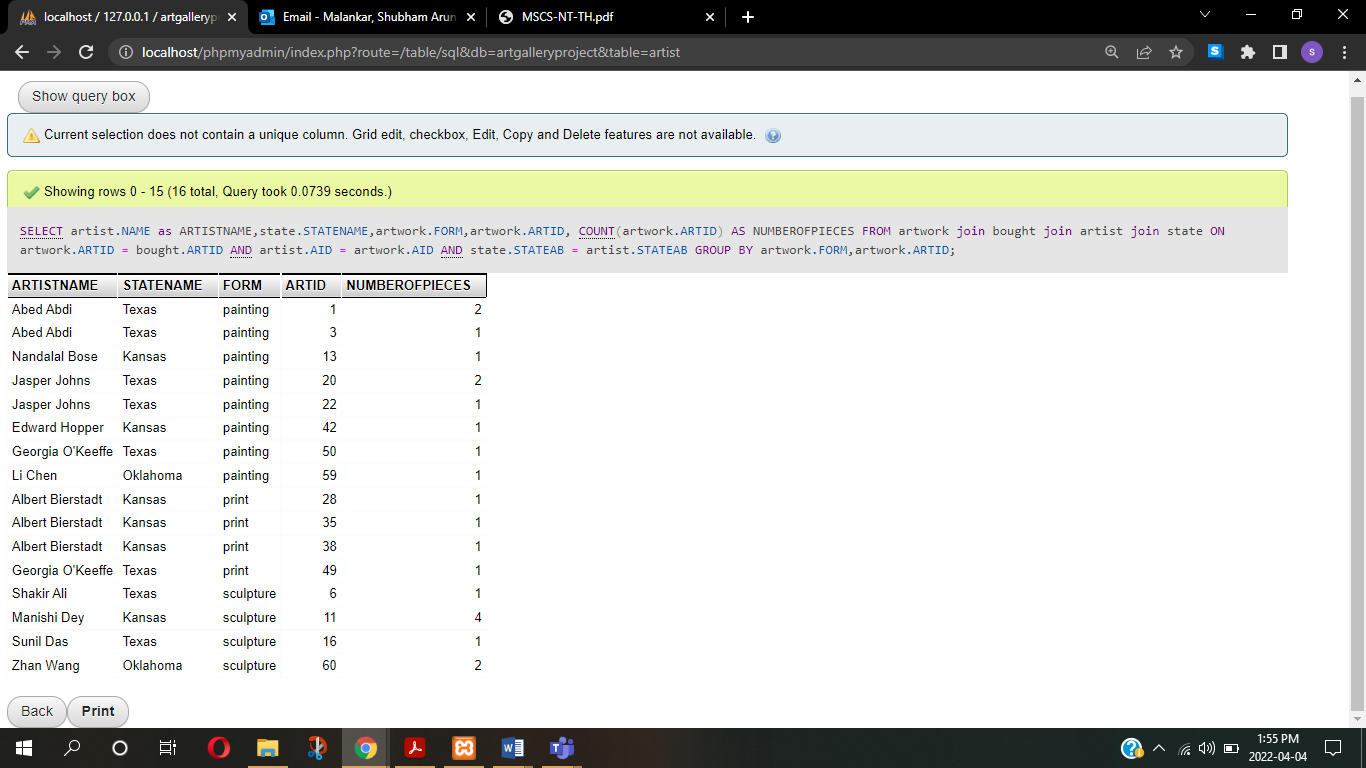
**Ans –** SELECT C.NAME as CUSTOMERNAME,R.TITLE, CONCAT('$', FORMAT(R.PRICE,2)) as ACQUISITIONDATE FROM (SELECT count(B.ARTID) AS NUMBEROFBUYERS,B.ARTID,B.CID FROM bought B GROUP BY B.ARTID ORDER BY NUMBEROFBUYERS DESC) t1, customer C, artwork R WHERE t1.NUMBEROFBUYERS=1 and C.CID = t1.CID AND t1.ARTID = R.ARTID ORDER BY `C`.`CID` DESC



Verify the result - I have sorted out the records of customer who bought only one painting using the subquery and GROUPED the customers according to it. Further we joined artwork, customer tables.

**Q12** Retrieve a list of records that is comprised of the state name, artist name, artform, and the number of pieces sold by the type of artform. Sort the list of records by the State name, artist’s name, artform, and the number of pieces sold. How can you verify that your query returns correct answers?

**Ans –** SELECT artist.NAME as ARTISTNAME,state.STATENAME,artwork.FORM,artwork.ARTID, COUNT(artwork.ARTID) AS NUMBEROFPIECES FROM artwork join bought join artist join state ON artwork.ARTID = bought.ARTID AND artist.AID = artwork.AID AND state.STATEAB = artist.STATEAB GROUP BY artwork.FORM,artwork.ARTID



Explanation – here we follow the reverse approach,

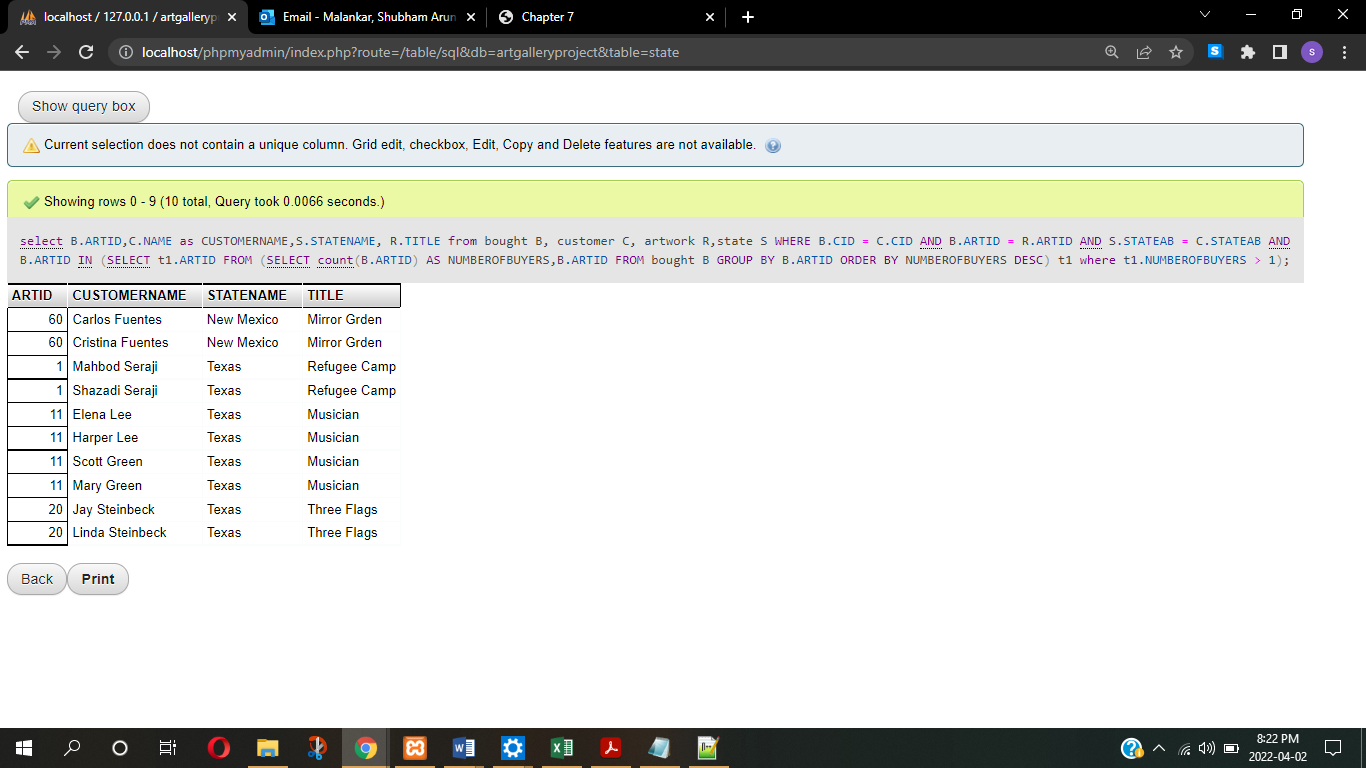
1. We first find the number pieces sold with respect to artform

2. We group them by artid.

3. once we have the artid we can easily find artist name and state name from artist and state tables respectively.

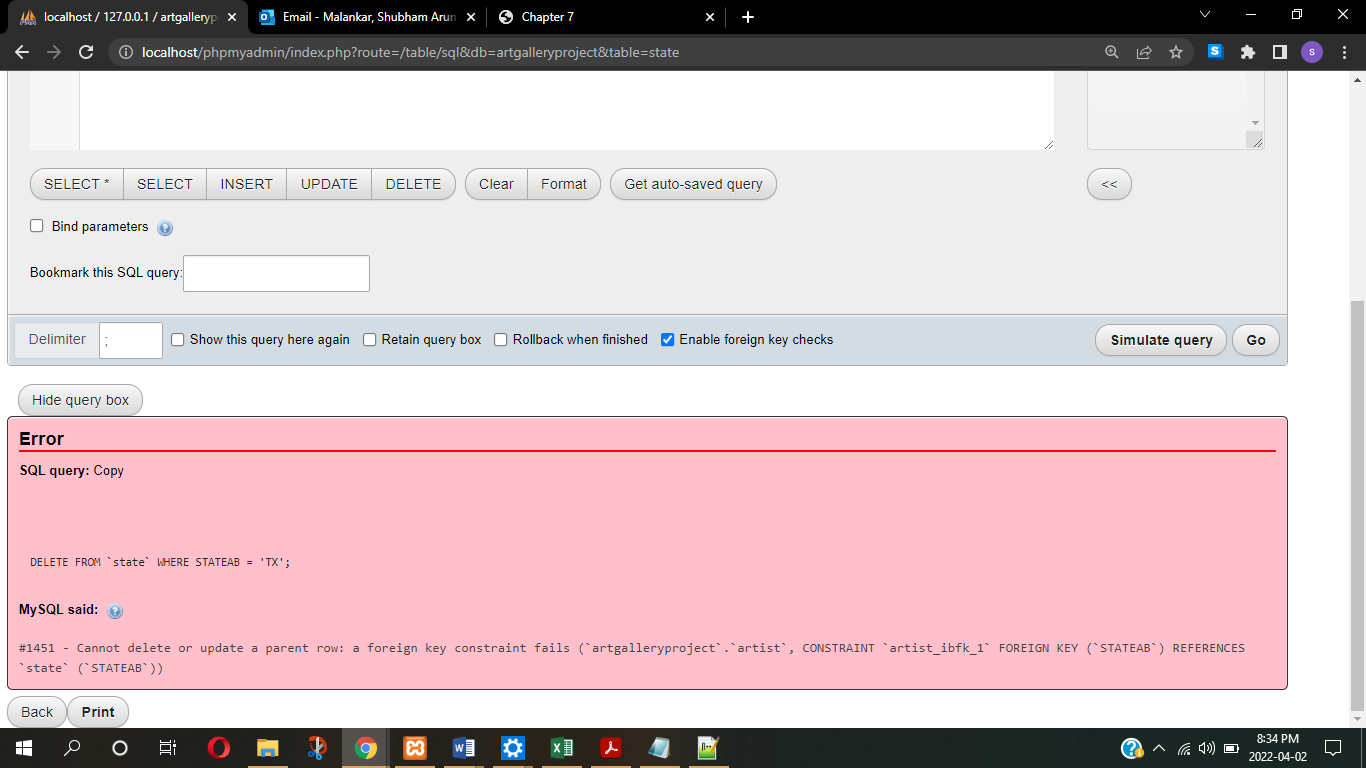
**Q13** Retrieve a list of records that each consists of the customer’s name, the state that hale from, and the title of the artwork they bought, for all the artwork that was bought by a group of two or more people.

**Ans –** SELECT B.ARTID,C.NAME as CUSTOMERNAME,S.STATENAME, R.TITLE from bought B, customer C, artwork R,state S WHERE B.CID = C.CID AND B.ARTID = R.ARTID AND S.STATEAB = C.STATEAB AND B.ARTID IN (SELECT t1.ARTID FROM (SELECT count(B.ARTID) AS NUMBEROFBUYERS,B.ARTID FROM bought B GROUP BY B.ARTID ORDER BY NUMBEROFBUYERS DESC) t1 where t1.NUMBEROFBUYERS > 1);



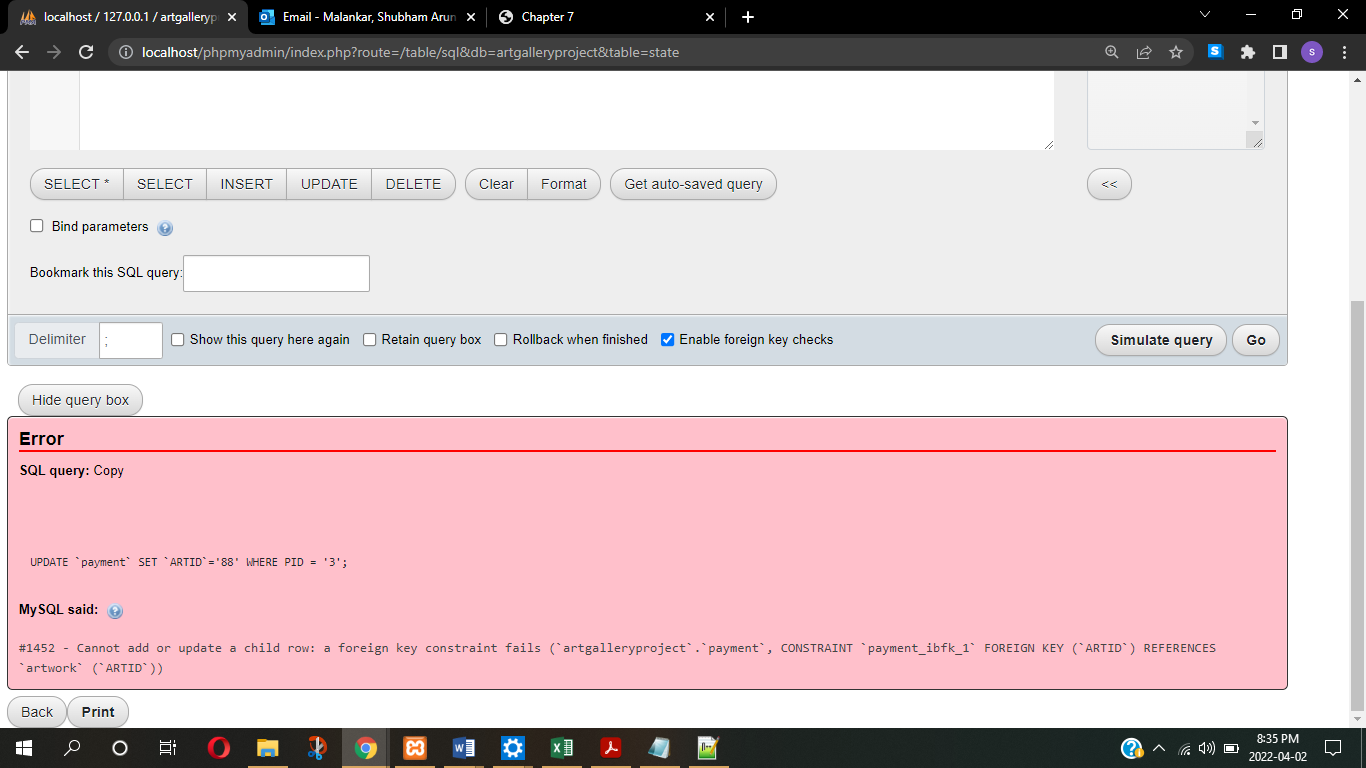
**Q14** Execute a command to delete a record that violates a referential integrity constraint. State the message produced by the DBMS.

**Ans –** DELETE FROM `state` WHERE STATEAB = 'TX'



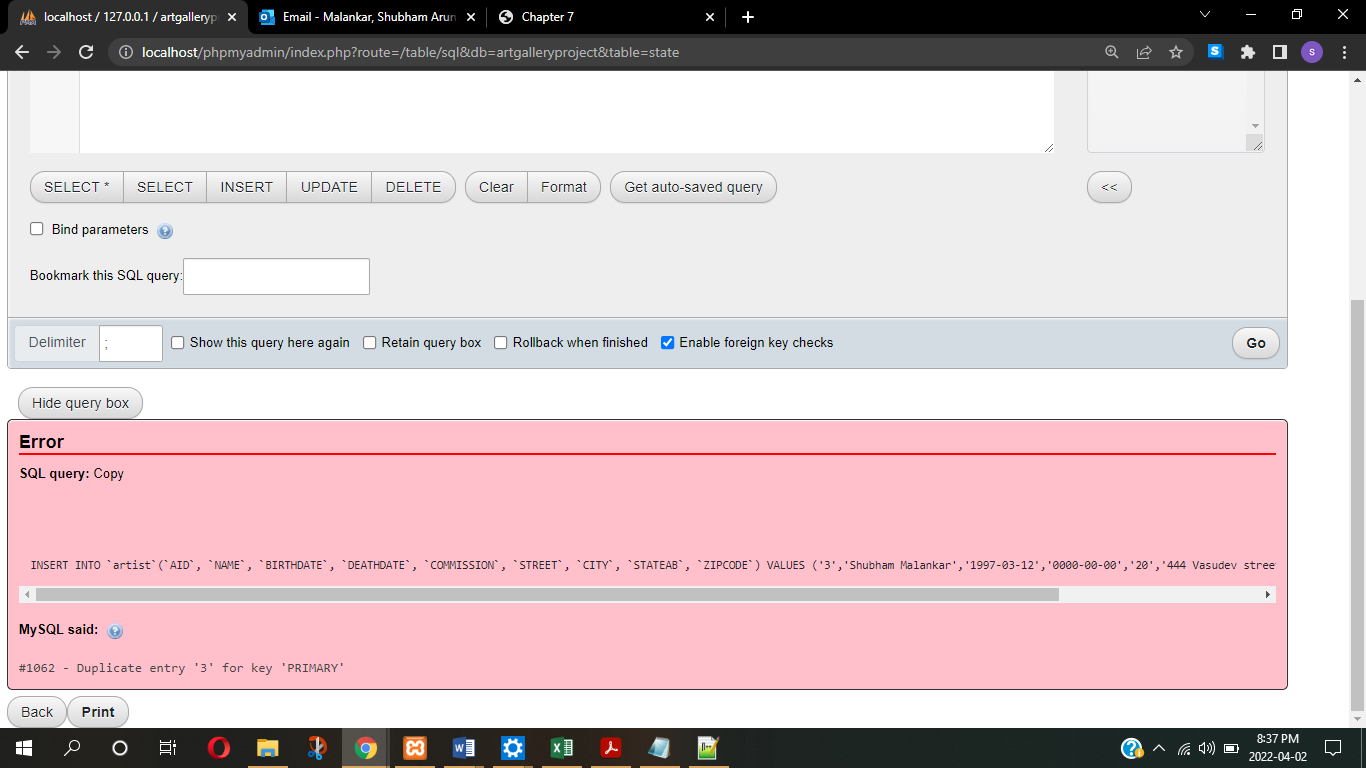
**Q15** Execute an update command for PAYMENTS table that attempts to update a record and thereby violates the foreign key constraint. State the message produced by the DBMS.

**Ans –** UPDATE `payment` SET `ARTID`='88' WHERE PID = '3'

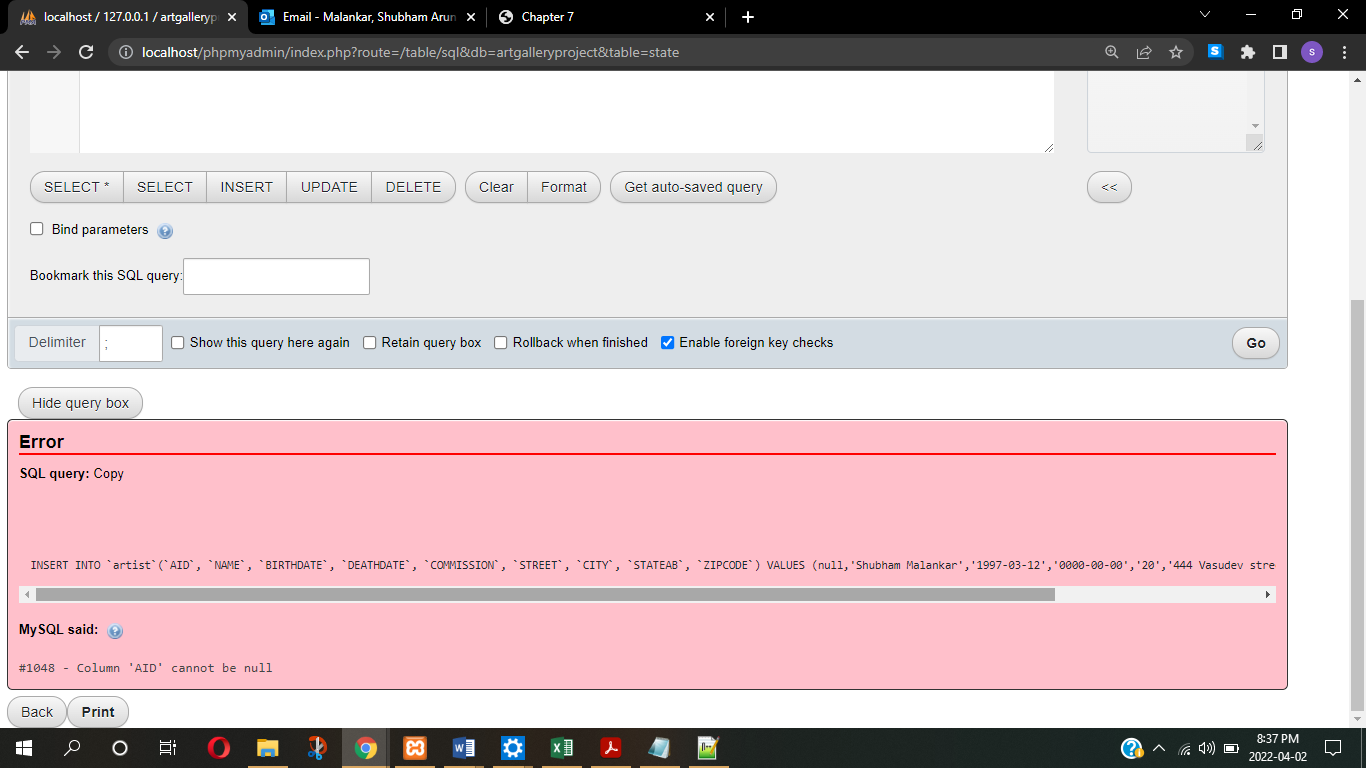


**Q16** Execute 3 insert commands for artist table that attempt to insert records, such that the records violate the explicit schema-based constraints (Key, Entity Integrity, Referential Integrity constraints). Make each of the 3 records violate a different types of integrity constraint. Include the insert statements and the error messages produced.

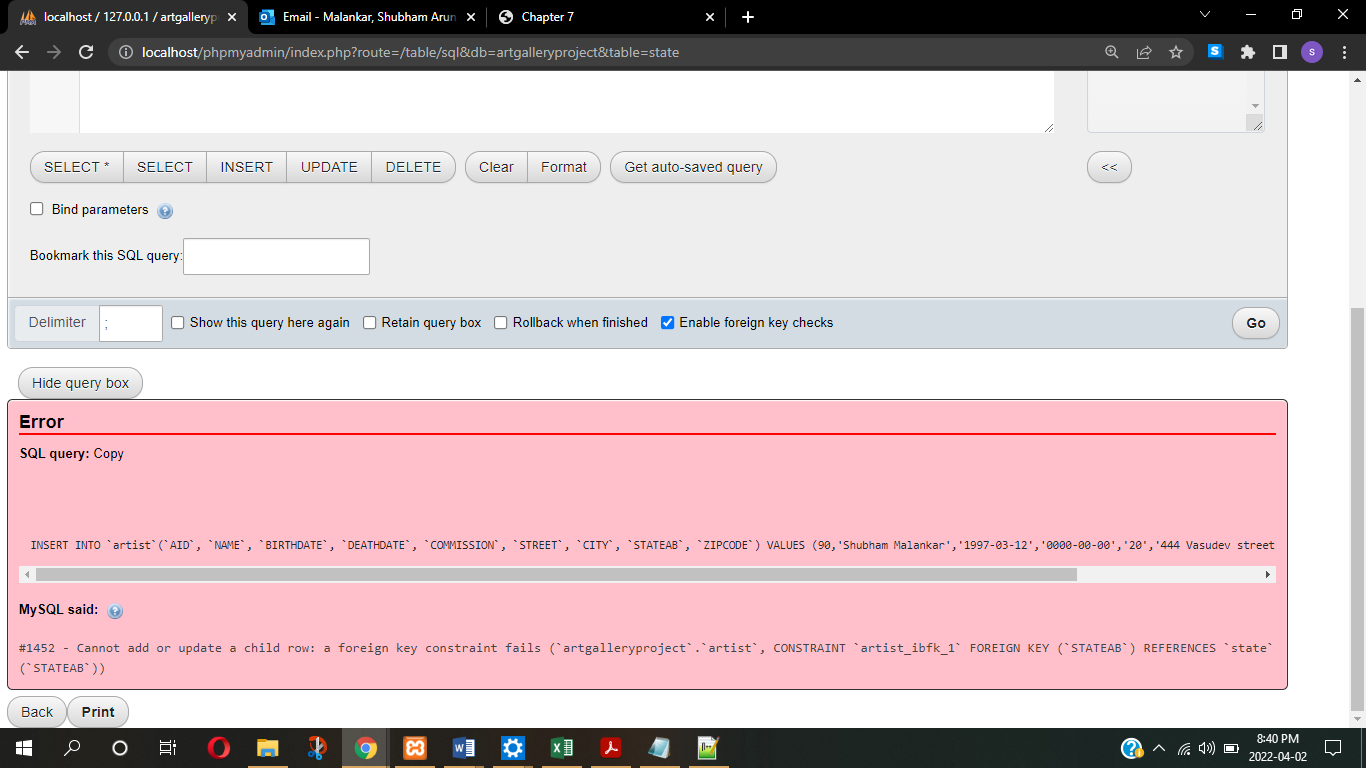
**Ans - Key integrity violation** - INSERT INTO `artist`(`AID`, `NAME`, `BIRTHDATE`, `DEATHDATE`, `COMMISSION`, `STREET`, `CITY`, `STATEAB`, `ZIPCODE`) VALUES ('3','Shubham Malankar','1997-03-12','0000-00-00','20','444 Vasudev street','Arlington','Texas','76013-1454')



**Entity integrity violation -** INSERT INTO `artist`(`AID`, `NAME`, `BIRTHDATE`, `DEATHDATE`, `COMMISSION`, `STREET`, `CITY`, `STATEAB`, `ZIPCODE`) VALUES (null,'Shubham Malankar','1997-03-12','0000-00-00','20','444 Vasudev street','Arlington','Texas','76013-1454')



**Referential integrity violation** - INSERT INTO `artist`(`AID`, `NAME`, `BIRTHDATE`, `DEATHDATE`, `COMMISSION`, `STREET`, `CITY`, `STATEAB`, `ZIPCODE`) VALUES (90,'Shubham Malankar','1997-03-12','0000-00-00','20','444 Vasudev street','Arlington','OH','76013-1454')



**References –**

1. <https://www.sqlshack.com/a-comprehensive-guide-to-the-sql-format-function/>
2. <https://www.w3schools.com/sql/func_sqlserver_concat.asp>
3. <https://www.w3schools.com/sql/sql_top.asp>
4. <https://stackoverflow.com/questions/15144393/using-sql-to-compare-counts-of-identifiers-from-two-tables>
5. <https://www.w3schools.com/sql/func_mysql_format.asp>
6. Fundamentals of Database Systems 7th edition by Elmasri and Navathe