National University of Computer and Emerging Sciences, Lahore Campus

Course:	**Database Systems (Lab)**	**Course Code:**	**CL203**
Program:	**BS (Computer Science)**	**Semester:**	**Fall 2018**
Duration:	**2 Hours**	**Total Marks:**	**40**
Paper Date:	**12-12-2018**	**Weight**	**35%**
Section:	**All**	**Page(s):**	**3**
Exam:	**Lab Final Exam**	**Reg. No**	

EXAM_NO_1

Important Instructions (Please read them before attempting the exam):

- Submit **ONLY .sql File** in this format (All parts of a question in one SQL File **named** with your **Roll Number** e.g., L13-4152). Do not zip your file.
- Copy the schema from the following path \\sandata\xeon\Ishaq\Database Systems- Fall 2018\Lab\final_schema.sql. Put the sql file in your D drive, and then **unplug the ethernet cable.**
- **Plagiarism** will result in **F grade** in lab.
- No cell phones are allowed. Sharing of **USBs** or any other items is **not allowed.**
- Submission path will be announced soon.
- **Write your roll number on this paper, and submit this paper to invigilator before leaving the lab.**

SQL Server Login Details:

- **Server: localhost or SQLExpress**
- **username: sa**
- **password: fstky2e4mdt**

Question#1 [5+5+5]

a) List the **art_id, art_name,** and **first_name** of the creator of that art piece whose total quantity sold is less than the total quantity sold of the art piece whose id is 1.

b) Create a view to list all the art pieces, along with the first and last name of their creators, which have never been sold using card payment mode.

c) List the first name and email of those creators who are trainers as well and have made an art piece of type ‘scenery’ as well as ‘portrait’ but not ‘sculpture’.

Question#2 [7]

Create a trigger such that whenever that trigger is fired as a result of insert in creators table, the trigger will insert the data itself in the creators table (the original insert statement will not insert data). However, the trigger will first check whether the first_name and last_name are different. If they are the same, data will not be inserted. Also print an appropriate message on successful as well as unsuccessful insertion.

Question#3 [3+7]

a) Create a UDF that takes **Creator_id** and returns the total quantity of art pieces of that creator sold.

b) Create a view that returns **Name and ID** of the creator whose every art piece of type 'scenery' has been sold more times than other creators who are trainers as well.

Question#4 [6+2]

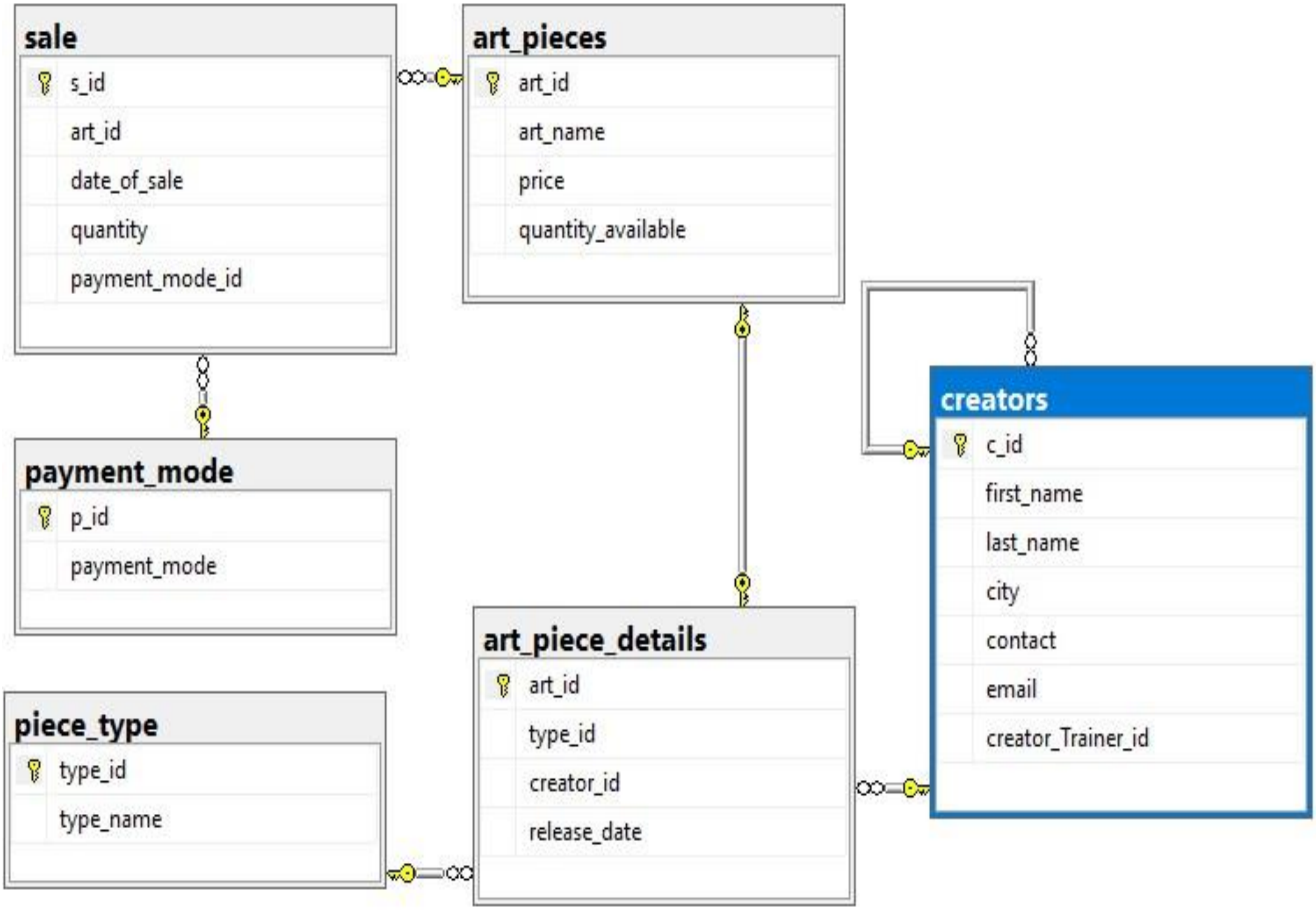
We want to add a procedure in the database that allows the customers to return the art pieces that they have bought. The amount that the customer had paid for the art pieces will be refunded. For this purpose, create a procedure ReturnSoldArtPieces which actually deletes the given s_id from sale table. But before deleting, it will perform the following tasks.

➔ The procedure checks whether the s_id is valid or not. If the s_id is valid, then deletion will be successful, and a status "Deleted Sucessfully" will be returned to the output parameter. If the id is invalid, then deletion will be unsuccessful, and a status "Sale Id Invalid" will be returned to the output parameter.

➔ System will also return the amount refunded to the customer as output parameter. Amount refunded will be calculated using price per unit and total quantity returned.

➔ The quantity of art piece which has been returned will also be restored in art_pieces table.

Also write the code to execute the procedure with s_id=1. Print the result of output parameters as well.



sale
- s_id
- art_id
- date_of_sale
- quantity
- payment_mode_id

art_pieces
- art_id
- art_name
- price
- quantity_available

creators
- c_id
- first_name
- last_name
- city
- contact
- email
- creator_Trainer_id

payment_mode
- p_id
- payment_mode

art_piece_details
- art_id
- type_id
- creator_id
- release_date

piece_type
- type_id
- type_name