Nicholas Ramella, Herat Zaveri, Harsh Patel
Ananya Dass
CS331 Database Management
05/07/2020

Deliverable 3

Goals:

The goals of the Phase 3 Deliverable are to have enough sample data to perform and illustrate all required tasks and have queries that execute. During this phase of the project we had to write SQL commands for creating tables including primary keys, secondary keys and foreign keys. We also had to write SQL commands in order to populate each table without generating errors and integrity violations. For the last part of this phase of the project we had to write SQL commands to support Photo applications, Sales applications and Reporting applications.

Changes From Deliverable 2

We did not have to change much from our relational schema created in project deliverable 2. There were small changes such as changing the names of some attributes in the influences table and removing the portraits table. Instead of having a separate table for portraits the models table directly references the photo table for the PhotoID foreign key.

Difficulties Faced

Some of the difficulties we faced while working on this phase of the project were figuring out the errors in implementing our queries and creating an application to run and display our queries. For this project we used phpmyadmin and mySQL in order to run and create our tables. There were some difficulties in learning how to use these systems and the phpmyadmin interface. When using mySQL we had to alter some of the queries we wrote as mySQL does not support all the operators of SQL such as MINUS. When creating the application that runs and displays queries we had to use HTML in order to create a web page that is easy to use. We decided that the best way to select which query to run was with a scroll bar menu. There was

some added difficulty while working on this part of the project while working remotely due to the pandemic. This made collaborating and checking each other's work much more inconvenient.

Create Table and Table Populate Commands

```
CREATE TABLE   Abstract
(PhotoID       INTEGER      NOT NULL,
Comment        VARCHAR(255),
PRIMARY KEY (PhotoID),
FOREIGN KEY (PhotoID) REFERENCES  Photo(PhotoID),

);

INSERT INTO `Abstract`(`PhotoID`, `Comment`) VALUES (111333, 'illusion');
INSERT INTO `Abstract`(`PhotoID`, `Comment`) VALUES (111444, 'Purity of Water');
INSERT INTO `Abstract`(`PhotoID`, `Comment`) VALUES (111888, 'Beautiful Colors');
INSERT INTO `Abstract`(`PhotoID`, `Comment`) VALUES (222222, 'Caveman Illusion');


CREATE TABLE   Customer
(LoginName    VARCHAR(25)       NOT NULL,
Password      VARCHAR(25)       NOT NULL,
CName         VARCHAR(25)       NOT NULL,
CType         VARCHAR(25)       NOT NULL,
BillingAddress VARCHAR(255),
Str1          VARCHAR(25),
Str2          VARCHAR(25),
City          VARCHAR(25),
State         VARCHAR(25),
Zip           INTEGER,
PRIMARY KEY (LoginName),
);


CREATE TABLE   Influences
(EPName       VARCHAR(25)       NOT NULL,
EPDate        DATE              NOT NULL,
RPName        VARCHAR(25)       NOT NULL,
RPDate        DATE              NOT NULL,
```

```
PRIMARY KEY (EPName),
PRIMARY KEY (RPName),
PRIMARY KEY (EPDate),
PRIMARY KEY (RPDate),
);


CREATE TABLE   LandScape
(PhotoID       INTEGER          NOT NULL,
Place          VARCHAR(25)      NOT NULL,
Country        VARCHAR(25)      NOT NULL,
PRIMARY KEY (PhotoID),
FOREIGN KEY (Place) REFERENCES  Location(Place),
FOREIGN KEY (PhotoID) REFERENCES  Photo(PhotoID),
FOREIGN KEY (Country) REFERENCES  Location(Country));

CREATE TABLE   Location
(Place         VARCHAR(25)      NOT NULL,
Country        VARCHAR(25)      NOT NULL,
Description    VARCHAR(255),
PRIMARY KEY (Place),
PRIMARY KEY (Country));

CREATE TABLE   Model
(MName         VARCHAR(25)      NOT NULL,
MBDate         VARCHAR(25)      NOT NULL,
MBio           VARCHAR(255),
MSex           VARCHAR(25),
PRIMARY KEY (MName),
PRIMARY KEY (MBDate));

CREATE TABLE   Models
( PhotoID      INTEGER          NOT NULL,
MName          VARCHAR(25)      NOT NULL,
MBDate         VARCHAR(25)      NOT NULL,
Agency         VARCHAR(25),
PRIMARY KEY (PhotoID),
PRIMARY KEY (MName),
PRIMARY KEY (MBDate),
```

```
FOREIGN KEY (MName) REFERENCES  Model(MName),
FOREIGN KEY (MBDate) REFERENCES  Model(MBDate),
FOREIGN KEY (PhotoID) REFERENCES  Photo(PhotoID),
);



CREATE TABLE  Photo
( PhotoID       INTEGER          NOT NULL,
Speed           INTEGER,
Film            VARCHAR(25)
F-Stop          VARCHAR(25)
Color/B & W     VARCHAR(25)
Resolution      VARCHAR(25)
Price           DECIMAL(10,2)    NOT NULL,
Date            DATE
TransID         INTEGER          NOT NULL,
PName           VARCHAR(25)      NOT NULL,
PBDate          DATE             NOT NULL,
PRIMARY KEY (PhotoID),
FOREIGN KEY (TransID) REFERENCES  Transaction(TransID),
FOREIGN KEY (PName) REFERENCES  Photographer(PName)
FOREIGN KEY (PBDate) REFERENCES  Photographer(PBDate)
);

CREATE TABLE  Photographer
( PName         VARCHAR(25)      NOT NULL,
PBDate          DATE             NOT NULL,
PBio            VARCHAR(25),
PAddress        VARCHAR(25),
Color/B & W     VARCHAR(25),
PNationality    VARCHAR(25),
PRIMARY KEY (PName),
PRIMARY KEY (PBDate),
FOREIGN KEY (PName) REFERENCES  Influences(EPName),
FOREIGN KEY (PBDate) REFERENCES  Photographer(EPDate));
```

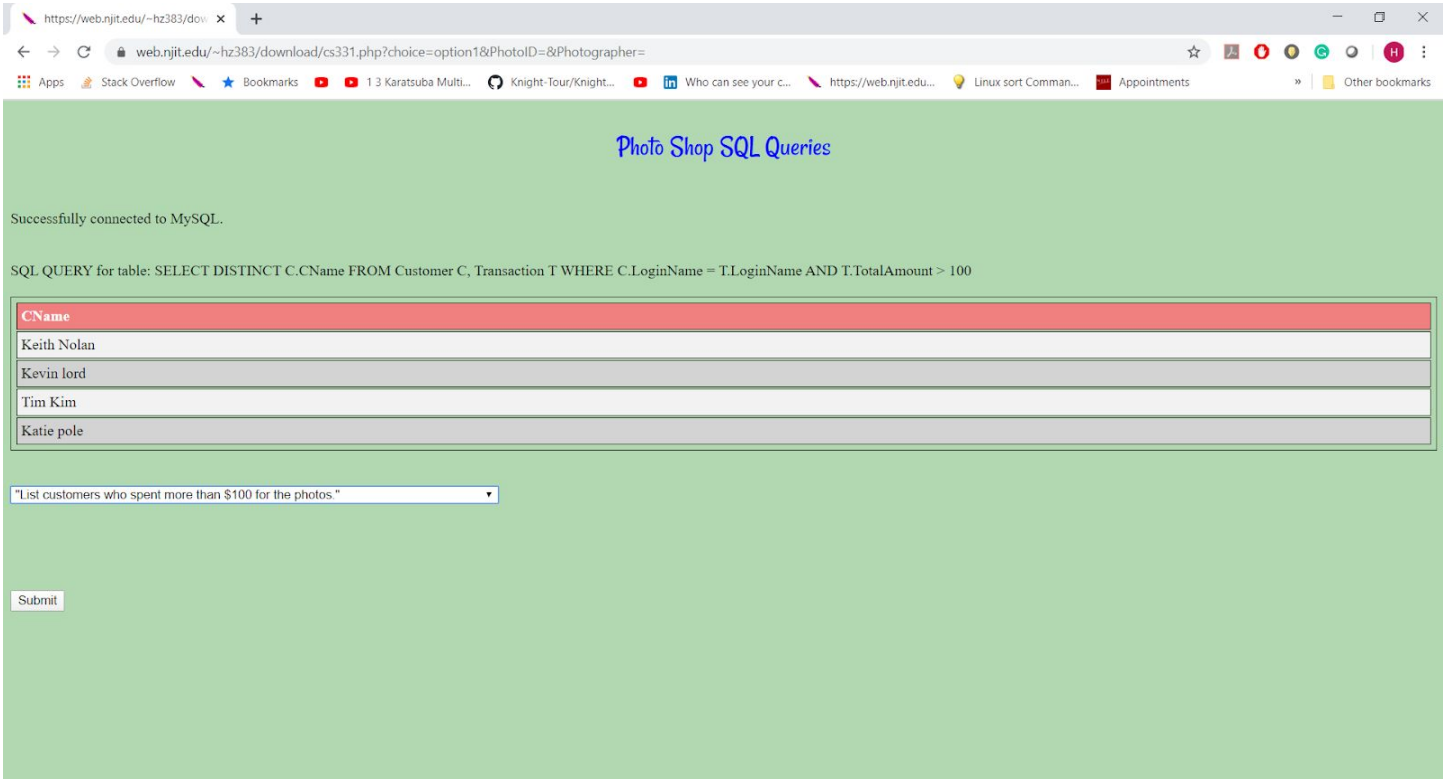
```
CREATE TABLE Transaction
(TransID        INTEGER          NOT NULL,
CardExpDate     DATE             NOT NULL,
TDate           DATE             NOT NULL,
CardNo          INTEGER          NOT NULL,
CardType        VARCHAR(25),
TotalAmount     DECIMAL(10,2)    NOT NULL,
LoginName       VARCHAR(25)      NOT NULL,

PRIMARY KEY (TransID),
FOREIGN KEY (LoginName) REFERENCES  Customer(LoginName));
```

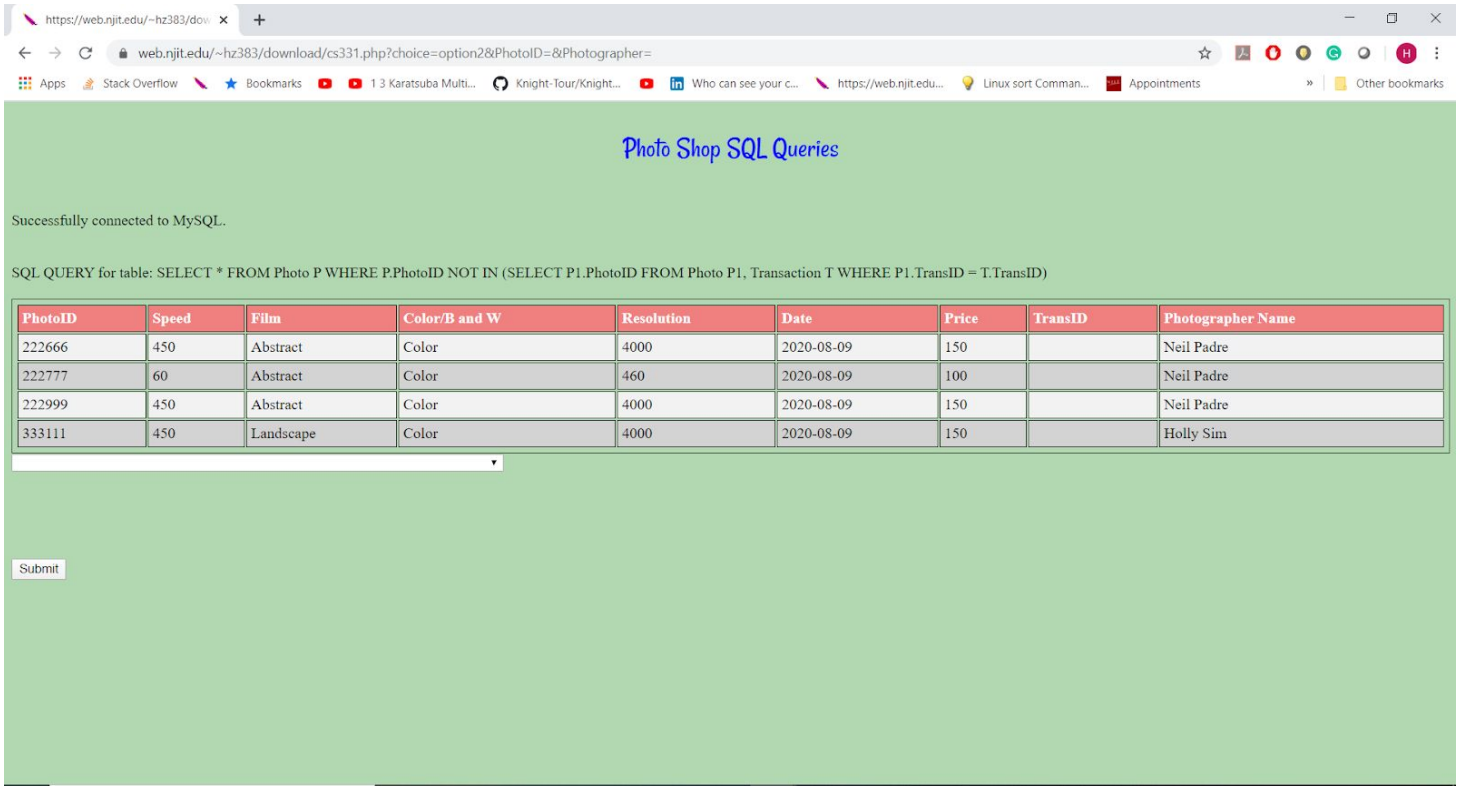
Queries

List customers who spent more than 100$ for the photos.

SELECT DISTINCT C.CName FROM Customer C, Transaction T WHERE C.LoginName = T.LoginName AND T.TotalAmount > 100



List photos which were not bought.



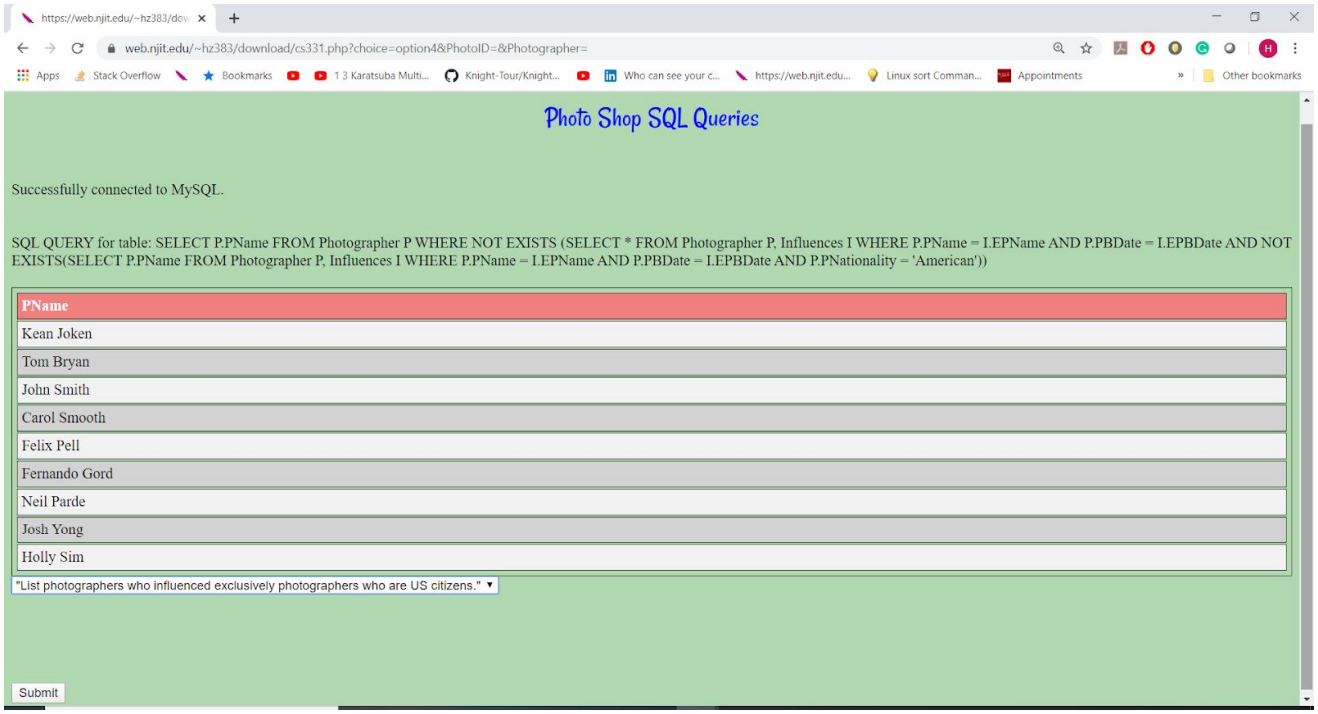
PhotoID	Speed	Film	Color/B and W	Resolution	Date	Price	TransID	Photographer Name
222666	450	Abstract	Color	4000	2020-08-09	150		Neil Padre
222777	60	Abstract	Color	460	2020-08-09	100		Neil Padre
222999	450	Abstract	Color	4000	2020-08-09	150		Neil Padre
333111	450	Landscape	Color	4000	2020-08-09	150		Holly Sim

List customers who bought all photos (portraits) in which a model X modeled.

```
SELECT C.CName FROM Customer C WHERE ((SELECT COUNT(C.LoginName)
        FROM Customer C, Transaction T, Photo P, Models M
        WHERE C.LoginName = T.LoginName AND T.TransID = P.TransID AND
P.PhotoID = M.PhotoID AND M.MName = 'Monica lewis')=(SELECT COUNT(P.PhotoID)
FROM Photo P, Models M
    WHERE P.PhotoID = M.PhotoID AND M.MName = 'Monica lewis'))
```

List photographers who influenced exclusively photographers who are US citizens.

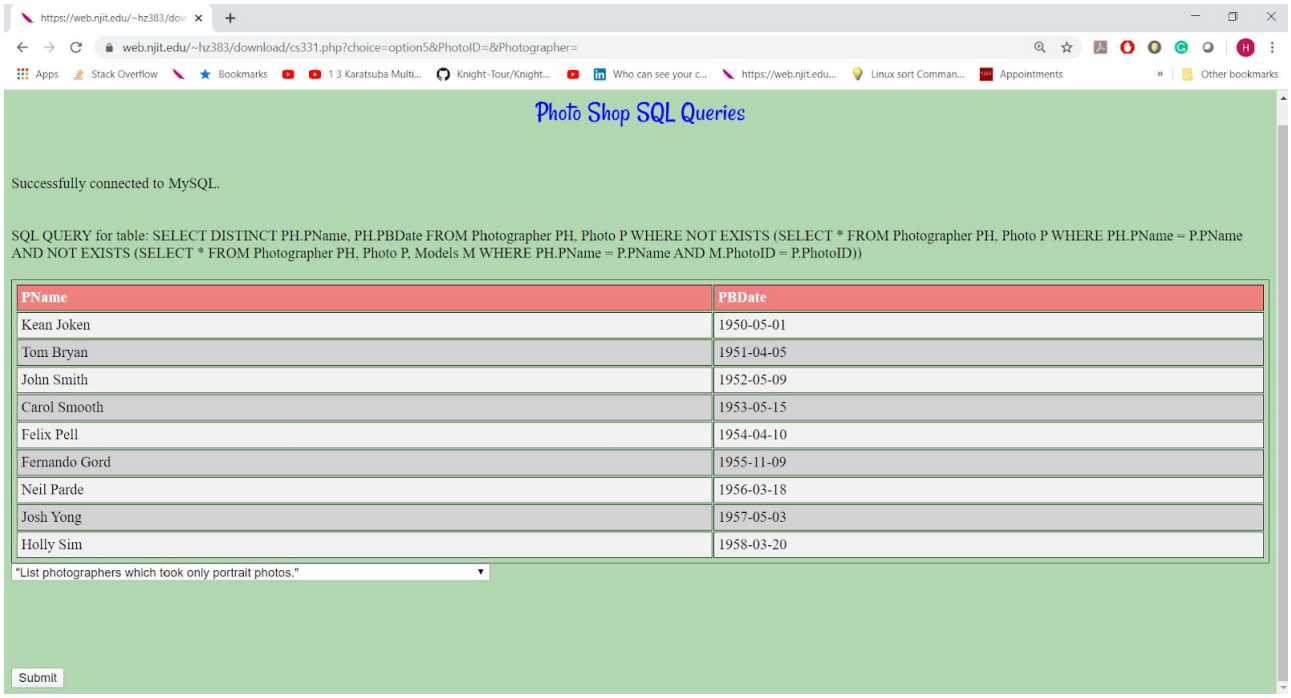
```
SELECT P.PName FROM Photographer P WHERE NOT EXISTS (SELECT * FROM
Photographer P, Influences I WHERE P.PName = I.EPName AND P.PBDate = I.EPBDate
    AND NOT EXISTS(SELECT P.PName FROM Photographer P, Influences I WHERE
P.PName = I.EPName AND P.PBDate = I.EPBDate AND P.PNationality = 'American'))
```



List photographers which took only portrait photos.

SELECT DISTINCT PH.PName, PH.PBDate FROM Photographer PH, Photo P WHERE NOT EXISTS (SELECT * FROM Photographer PH, Photo P WHERE PH.PName = P.PName AND NOT EXISTS

(SELECT * FROM Photographer PH, Photo P, Models M WHERE PH.PName = P.PName AND M.PhotoID = P.PhotoID))



List transactions (transID) which contain more than 3 photos.

SELECT P.TransID FROM Photo P, Photo P2 WHERE P.TransID = P2.TransID GROUP BY P.TransID HAVING COUNT(P.TransID)>3

List models who modeled in all photos taken by photographer Y.

SELECT DISTINCT M.MName FROM Models M WHERE NOT EXISTS (SELECT P.PName FROM Photographer PH, Photo P WHERE P.PName = PH.PName AND NOT EXISTS(SELECT P.PName FROM Photographer PH,Photo P WHERE P.PName = PH.PName AND P.PName = 'Fernando Gord'))

Rank the photographers by the total cost (sum of prices) of the photos they took.

```
SELECT P.PName, P.PBDate, SUM(P.Price)
FROM Photo P
GROUP BY P.PName, P.PBDate
ORDER BY SUM(P.Price) DESC
```

https://web.njit.edu/~hz383/dow

web.njit.edu/~hz383/download/cs331.php?choice=option8&PhotoID=&Photographer=

Apps | Stack Overflow | Bookmarks | 1 3 Karatsuba Multi... | Knight-Tour/Knight... | Who can see your c... | https://web.njit.edu... | Linux sort Comman... | Appointments | Other bookmarks

Photo Shop SQL Queries

Successfully connected to MySQL.

SQL QUERY for table: SELECT P.PName, P.PBDate, SUM(P.Price) FROM Photo P GROUP BY P.PName, P.PBDate ORDER BY SUM(P.Price) DESC

PName	PBDate	Total Price of Photos
Fernando Gord	1955-11-09	700
Neil Padre	1956-03-18	599
Holly Sim	1958-03-20	310
Neil Parde	1956-03-18	172
Tom Bryan	1951-04-05	25
Felix Pell	1954-04-10	23
Kean Joken	1950-05-01	20
Josh Yong	1957-05-03	10
John Smith	1952-05-09	5

Submit

Delete from relation Photo the photo with photoID=X.

DELETE FROM Photo WHERE PhotoID='$p_id'

Photo Shop SQL Queries

Successfully connected to MySQL.

SQL QUERY for table: DELETE FROM Photo WHERE PhotoID='222999'

After Deletion Table With Information:

PhotoID	Speed	Film	F-Stop	Color/B&W	Resolution	Price	Date	PName	PBDate
111222	250	Landscape	f-4	Blue	1080	20	2020-04-01	Kean Joken	1950-05-01
111333	250	Abstract	f-25	Color	1080	25	2000-02-01	Tom Bryan	1951-04-05
111444	50	Abstract	f-4	B and W	480	5	1997-05-12	John Smith	1952-05-09
111555	250	Portrait	f-255	Color	4000	250	2018-01-01	Fernando Gord	1955-11-09
111666	250	Landscape	f-50	Color	1080	23	2011-08-15	Felix Pell	1954-04-10
111777	60	Portrait	f-6	Color	4000	300	2000-05-25	Fernando Gord	1955-11-09
111888	50	Abstract	f-255	B and W	4000	122	2020-06-01	Neil Parde	1956-03-18
111999	266	Landscape	f-4	Color	360	10	2020-03-12	Josh Yong	1957-05-03
175687	300	Landscape	F-6	BlackWhite	90	7000	2020-05-11	Bansky	1955-11-09
222111	250	Portrait	f-50	B and W	1080	60	2002-05-19	Holly Sim	1958-03-20
222222	60	Abstract	f-50	Color	1080	50	1998-01-01	Neil Parde	1956-03-18
222333	250	Portrait	f90	Color	1080	100	2000-05-24	Holly Sim	1958-03-20
222444	250	Abstract	f50	B and W	360	199	1999-01-02	Neil Padre	1956-03-18
222555	266	Portrait	f-9	Color	1080	150	2020-01-25	Fernando Gord	1955-11-09
222666	450	Abstract	f-90	Color	4000	150	2020-08-09	Neil Padre	1956-03-18
222777	60	Abstract	f-90	Color	460	100	2020-08-09	Neil Padre	1956-03-18
333111	450	Landscape	f-90	Color	4000	150	2020-08-09	Holly Sim	1958-03-20
333444	300	Landscape	F-6	BlackWhite	90	7000	2020-05-11	Bansky	1955-11-09

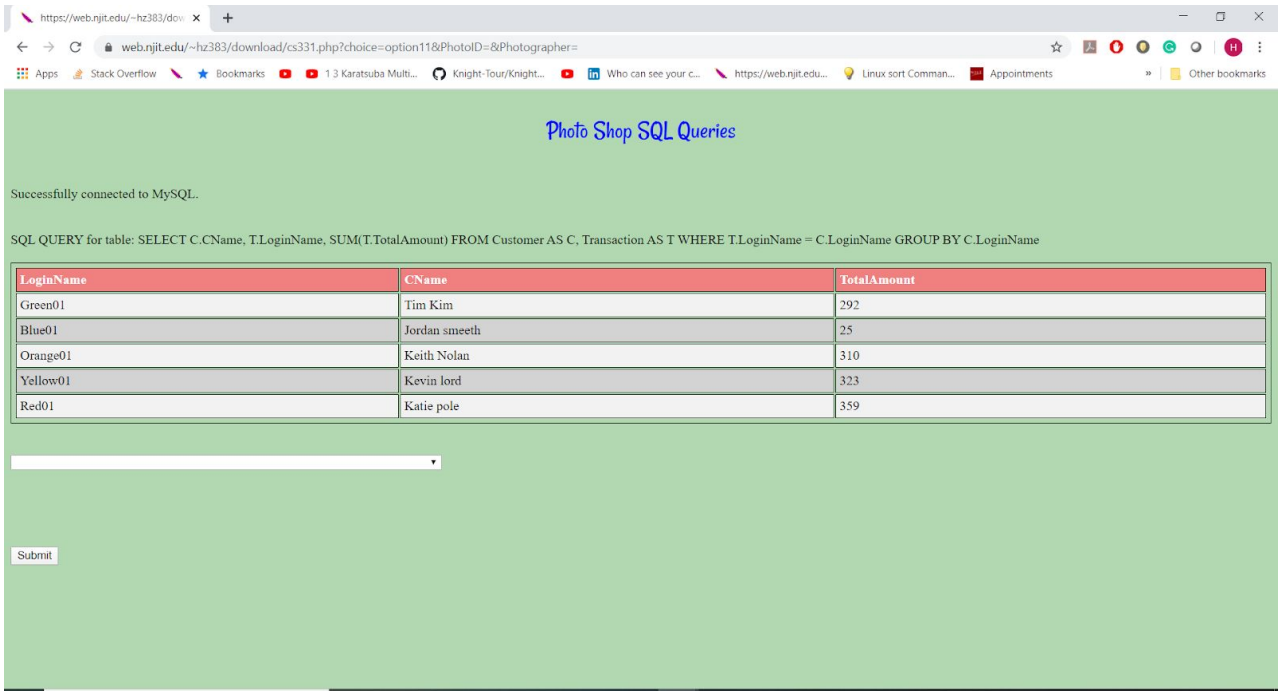
Update the photographer name of the photo with photoID=X to Y.

UPDATE Photo SET PNAME = '$pname' WHERE PhotoID = '$p_id'

PhotoID	PName	PBDate
111222	Kean Joken	1950-05-01
111333	Tom Bryan	1951-04-05
111444	John Smith	1952-05-09
111555	Fernando Gord	1955-11-09
111666	Felix Pell	1954-04-10
111777	Fernando Gord	1955-11-09
111888	Neil Parde	1956-03-18
111999	Josh Yong	1957-05-03
222111	Holly Sim	1958-03-20
222222	Neil Parde	1956-03-18
222333	Holly Sim	1958-03-20
222444	Neil Padre	1956-03-18
222555	Fernando Gord	1955-11-09
222666	Neil Padre	1956-03-18
222777	Neil Padre	1956-03-18
222999	Michael Scott	1956-03-18
333111	Holly Sim	1958-03-20

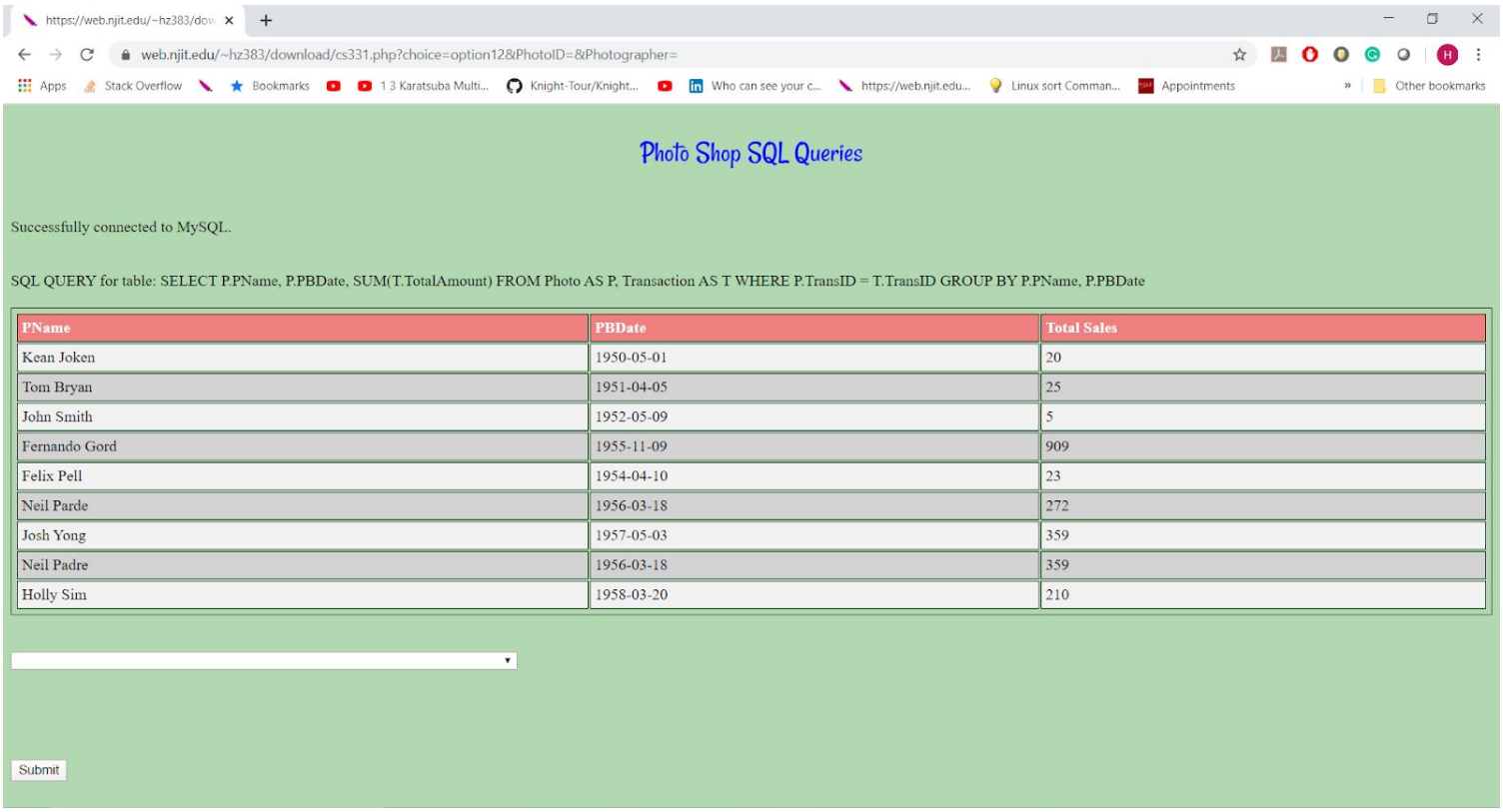
Compute total sales per customer

SELECT C.CName, T.LoginName, SUM(T.TotalAmount) FROM Customer AS C, Transaction AS T WHERE T.LoginName = C.LoginName GROUP BY C.LoginName



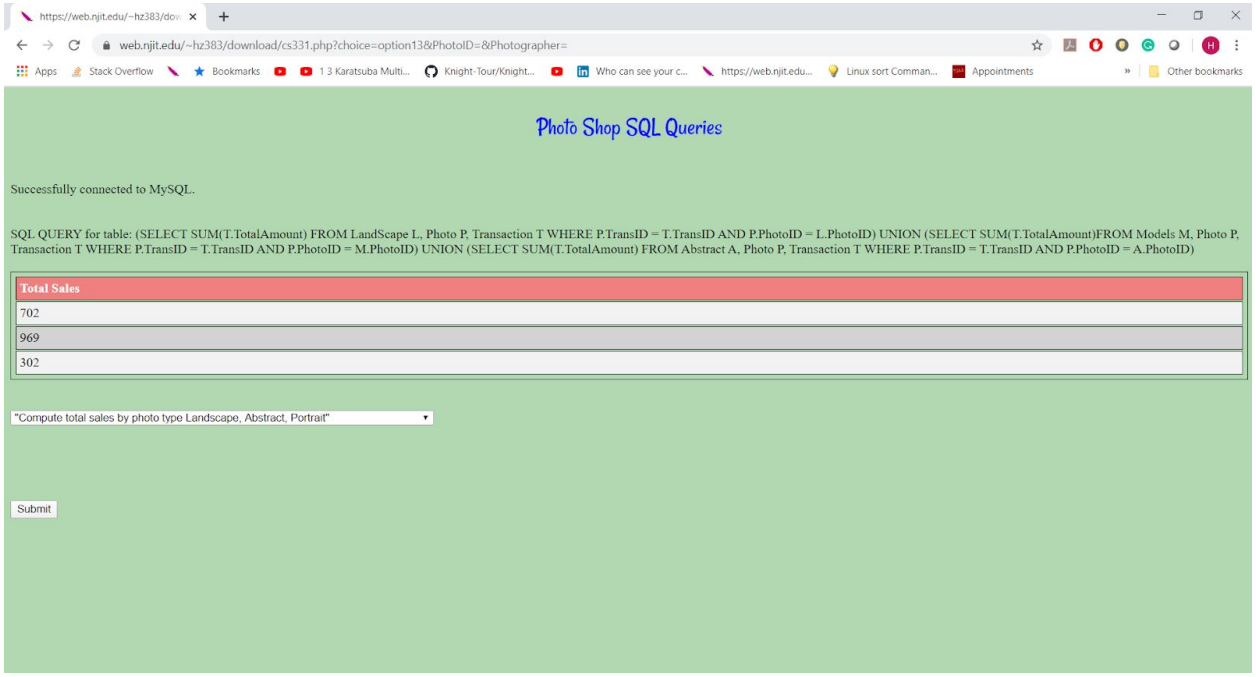
Compute total sales per photographer sorted by photographer

SELECT P.PName, P.PBDate, SUM(T.TotalAmount) FROM Photo AS P, Transaction AS T WHERE P.TransID = T.TransID GROUP BY P.PName, P.PBDate



Compute total sales by photo type (portrait, landscape etc.)

(SELECT SUM(T.TotalAmount) FROM LandScape L, Photo P, Transaction T WHERE P.TransID = T.TransID AND P.PhotoID = L.PhotoID) UNION (SELECT SUM(T.TotalAmount)FROM Models M, Photo P, Transaction T WHERE P.TransID = T.TransID AND P.PhotoID = M.PhotoID) UNION (SELECT SUM(T.TotalAmount) FROM Abstract A, Photo P, Transaction T WHERE P.TransID = T.TransID AND P.PhotoID = A.PhotoID)



Compute top n dates (in a total sales per date list)

SELECT T.TDate, SUM(T.TotalAmount) FROM Transaction AS T GROUP BY T.TDate ORDER BY SUM(T.TotalAmount) DESC