psql -h wander -U jsalvage I4f5VoiWLK8ui5

psql -h wander -U cs461\_user -d CS461Dev

and pwd is Bugofsent2

DROP TABLE Photographers CASCADE;

DROP TABLE Photographs CASCADE;

DROP TABLE PhotoSales CASCADE;

DROP TABLE Customers CASCADE;

DROP TABLE RefPhotoAttributes CASCADE;

DROP TABLE PhotoAttributes CASCADE;

DROP TABLE RefSalesCategory CASCADE;

DROP TABLE SalesCategories CASCADE;

DROP TABLE RefEquipmentUsed CASCADE;

DROP TABLE PhotoEquipment CASCADE;

DROP TABLE UpdatedPhotographs CASCADE;

CREATE TABLE Photographers

(id\_photographer SERIAL,

first\_name CHAR (20),

last\_name CHAR (20),

order\_form CHAR (30),

PRIMARY KEY (id\_photographer)

);

CREATE UNIQUE INDEX name\_index on Photographers(last\_name, first\_name);

CREATE TABLE Photographs

(

id\_photo SERIAL,

filename char(20),

title char(20),

Description char(20),

id\_photographer int,

photo\_order int,

height int,

width int,

city char (20),

state char (20),

country char(20),

date\_taken date,

PRIMARY KEY (id\_photo),

FOREIGN KEY (id\_photographer) REFERENCES Photographers(id\_photographer) ON DELETE CASCADE

);

CREATE UNIQUE INDEX filename\_index on Photographs (filename);

CREATE TABLE Customers

(

id\_customer SERIAL,

first\_name char(20),

last\_name char(20),

address1 char(50),

address2 char(50),

city char(20),

state char(20),

zip char(9),

country char(20),

PRIMARY KEY (id\_customer)

);

CREATE UNIQUE INDEX customer\_name\_index on Customers (last\_name, first\_name);

CREATE TABLE PhotoSales

(

id\_sale SERIAL,

id\_customer int,

id\_photo int,

quanity\_sold int,

id\_sale\_price float,

id\_tax int,

sale\_date date,

PRIMARY KEY (id\_sale)

);

CREATE TABLE RefPhotoAttributes

(

id\_attribute SERIAL,

attribute\_desc char(20),

PRIMARY KEY (id\_attribute));

CREATE UNIQUE INDEX attribute\_index on RefPhotoAttributes (attribute\_desc);

CREATE TABLE PhotoAttributes

(

id\_photo int,

id\_attribute int,

PRIMARY KEY (id\_photo, id\_attribute),

FOREIGN KEY (id\_photo) REFERENCES Photographs(id\_photo) ON DELETE CASCADE,

FOREIGN KEY (id\_attribute) REFERENCES RefPhotoAttributes(id\_attribute) ON DELETE CASCADE

);

CREATE TABLE RefSalesCategory

(

id\_sales\_category SERIAL,

sales\_category char(20),

PRIMARY KEY (id\_sales\_category));

CREATE UNIQUE INDEX salescat\_index on RefSalesCategory (sales\_category);

CREATE TABLE SalesCategories

(

id\_photo int,

id\_sales\_category int,

PRIMARY KEY (id\_photo, id\_sales\_category),

FOREIGN KEY (id\_photo) REFERENCES Photographs(id\_photo) ON DELETE CASCADE,

FOREIGN KEY (id\_sales\_category) REFERENCES RefSalesCategory(id\_sales\_category) ON DELETE CASCADE

);

CREATE TABLE RefEquipmentUsed

(

id\_equipment SERIAL,

equipment\_desc char(30),

PRIMARY KEY (id\_equipment));

CREATE TABLE PhotoEquipment

(

id\_photo int,

id\_equipment int,

PRIMARY KEY (id\_photo, id\_equipment),

FOREIGN KEY (id\_photo) REFERENCES Photographs(id\_photo) ON DELETE CASCADE,

FOREIGN KEY (id\_equipment) REFERENCES RefEquipmentUsed(id\_equipment) ON DELETE CASCADE

);

CREATE TABLE UpdatedPhotographs

(

filename char(20),

title char(20),

PRIMARY KEY (filename)

);

INSERT INTO Photographers (first\_name, last\_name, order\_form) VALUES ('JEFF', 'SALVAGE', 'ORDER1');

INSERT INTO Photographers (first\_name, last\_name, order\_form) VALUES ('JOHN', 'COCKTOSEN', 'ORDER2');

INSERT INTO Photographers (first\_name, last\_name, order\_form) VALUES ('BILLY JEAN', 'KING', 'ORDER1');

INSERT INTO RefEquipmentUsed (equipmentdesc) VALUES ('580X Flash II');

INSERT INTO RefEquipmentUsed (equipmentdesc) VALUES ('Canon Rebel');

INSERT INTO RefEquipmentUsed (equipmentdesc) VALUES ('300mm IS F2.8 Lens Canon');

INSERT INTO Photographs (FILENAME, TITLE, HEIGHT, WIDTH, CITY, DATE\_TAKEN, ID\_PHOTOGRAPHER) VALUES ('A','Photo 1', 10, 5, 'PHILADELPHIA', '12/12/12',1);

INSERT INTO Photographs (FILENAME, TITLE, HEIGHT, WIDTH, CITY, DATE\_TAKEN, ID\_PHOTOGRAPHER) VALUES ('B','Photo 2', 10, 5, 'NYC', '12/12/12',1);

INSERT INTO Photographs (FILENAME, TITLE, HEIGHT, WIDTH, CITY, DATE\_TAKEN, ID\_PHOTOGRAPHER) VALUES ('C','Photo 3', 1, 5, 'PHILADELPHIA', '12/1/12',1);

INSERT INTO Photographs (FILENAME, TITLE, HEIGHT, WIDTH, CITY, DATE\_TAKEN, ID\_PHOTOGRAPHER) VALUES ('D','Photo 4', 1, 5, 'PHILADELPHIA', '12/2/12',1);

INSERT INTO Photographs (FILENAME, TITLE, HEIGHT, WIDTH, CITY, DATE\_TAKEN, ID\_PHOTOGRAPHER) VALUES ('E','Photo 5', 20, 5, 'TEST LARGEST', '12/8/12',1);

INSERT INTO Photographs (FILENAME, TITLE, HEIGHT, WIDTH, CITY, DATE\_TAKEN, ID\_PHOTOGRAPHER) VALUES ('F','Photo 1', 10, 5, 'PHILADELPHIA', '12/12/12',2);

INSERT INTO Photographs (FILENAME, TITLE, HEIGHT, WIDTH, CITY, DATE\_TAKEN, ID\_PHOTOGRAPHER) VALUES ('G','Photo 2', 10, 5, 'NYC', '12/12/12',2);

INSERT INTO Photographs (FILENAME, TITLE, HEIGHT, WIDTH, CITY, DATE\_TAKEN, ID\_PHOTOGRAPHER) VALUES ('H','Photo 3', 1, 5, 'PHILADELPHIA', '12/1/12',2);

INSERT INTO Photographs (FILENAME, TITLE, HEIGHT, WIDTH, CITY, DATE\_TAKEN, ID\_PHOTOGRAPHER) VALUES ('I','Photo 4', 1, 5, 'PHILADELPHIA', '12/2/12',2);

INSERT INTO Photographs (FILENAME, TITLE, HEIGHT, WIDTH, CITY, DATE\_TAKEN, ID\_PHOTOGRAPHER) VALUES ('J','Photo 5', 20, 5, 'TEST LARGEST', '12/8/12',2);

INSERT INTO Photographs (FILENAME, TITLE, HEIGHT, WIDTH, CITY, DATE\_TAKEN, ID\_PHOTOGRAPHER) VALUES ('K','Photo 1', 10, 5, 'PHILADELPHIA', '12/4/12',3);

INSERT INTO Photographs (FILENAME, TITLE, HEIGHT, WIDTH, CITY, DATE\_TAKEN, ID\_PHOTOGRAPHER) VALUES ('L','Photo 2', 10, 5, 'PHILADELPHIA', '12/5/12',3);

INSERT INTO Photographs (FILENAME, TITLE, HEIGHT, WIDTH, CITY, DATE\_TAKEN, ID\_PHOTOGRAPHER) VALUES ('M','Photo 3', 1, 5, 'PHILADELPHIA', '12/1/12',3);

INSERT INTO Photographs (FILENAME, TITLE, HEIGHT, WIDTH, CITY, DATE\_TAKEN, ID\_PHOTOGRAPHER) VALUES ('N','Photo 4', 1, 5, 'PHILADELPHIA', '12/2/12',3);

INSERT INTO Photographs (FILENAME, TITLE, HEIGHT, WIDTH, CITY, DATE\_TAKEN, ID\_PHOTOGRAPHER) VALUES ('O','Photo 4', 1, 1, 'TEST SMALLEST', '12/20/12',3);

INSERT INTO Photographs (FILENAME, TITLE, HEIGHT, WIDTH, CITY, DATE\_TAKEN, ID\_PHOTOGRAPHER) VALUES ('P','Photo 6', 1, 10, 'TEST DUP CITY', '12/20/12',3);

INSERT INTO UpdatedPhotographs (filename, title) VALUES ('A', 'Photo 1A');

INSERT INTO UpdatedPhotographs (filename, title) VALUES ('B', 'Photo 2A');

INSERT INTO UpdatedPhotographs (filename, title) VALUES ('C', 'Photo 3A');

INSERT INTO PhotoEquipment (id\_photo, id\_equipment) VALUES (1,1);

INSERT INTO PhotoEquipment (id\_photo, id\_equipment) VALUES (1,2);

INSERT INTO PhotoEquipment (id\_photo, id\_equipment) VALUES (1,3);

INSERT INTO PhotoEquipment (id\_photo, id\_equipment) VALUES (2,1);

INSERT INTO RefPhotoAttributes (attribute\_desc) VALUES ('Dallas Cowboys');

INSERT INTO RefPhotoAttributes (attribute\_desc) VALUES ('Landscape');

INSERT INTO RefPhotoAttributes (attribute\_desc) VALUES ('Ocean');

INSERT INTO RefPhotoAttributes (attribute\_desc) VALUES ('Unassigned');

INSERT INTO PhotoAttributes (id\_photo, id\_attribute) VALUES (1,1);

INSERT INTO PhotoAttributes (id\_photo, id\_attribute) VALUES (1,2);

INSERT INTO PhotoAttributes (id\_photo, id\_attribute) VALUES (1,3);

INSERT INTO PhotoAttributes (id\_photo, id\_attribute) VALUES (3,2);

INSERT INTO PhotoSales (id\_photo, id\_sale\_price, id\_tax, sale\_date) VALUES (11,1,1,'12/12/12');

INSERT INTO PhotoSales (id\_photo, id\_sale\_price, id\_tax, sale\_date) VALUES (12,1,1,'12/12/12');

INSERT INTO PhotoSales (id\_photo, id\_sale\_price, id\_tax, sale\_date) VALUES (1,1,1,'12/12/12');

INSERT INTO RefSalesCategory (sales\_category) Values ('Cat 1');

INSERT INTO RefSalesCategory (sales\_category) Values ('Cat 2');

INSERT INTO RefSalesCategory (sales\_category) Values ('Cat 3');

INSERT INTO RefSalesCategory (sales\_category) Values ('Cat 4');

INSERT INTO SalesCategories (id\_photo, id\_sales\_category) VALUES (3,1);

INSERT INTO SalesCategories (id\_photo, id\_sales\_category) VALUES (3,4);

INSERT INTO SalesCategories (id\_photo, id\_sales\_category) VALUES (2,2);

INSERT INTO SalesCategories (id\_photo, id\_sales\_category) VALUES (11,2);

INSERT INTO SalesCategories (id\_photo, id\_sales\_category) VALUES (12,4);

LAB 1

**1) Select the photographers who sold a photo taken in the city “New York City” and the country “United States” with an attribute “Sky” and a sales category of “Cat 1”**

SELECT COUNT(Photographs.id\_photo) AS PhotoCount FROM Photographs INNER JOIN PhotoSales ON Photographs.id\_photo = PhotoSales.id\_photo INNER JOIN PhotoAttributes ON PhotoAttributes.id\_photo = Photographs.id\_photo INNER JOIN RefPhotoAttributes ON RefPhotoAttributes.id\_attribute = PhotoAttributes.id\_attribute INNER JOIN SalesCategories ON SalesCategories.id\_photo INNER JOIN RefSalesCategories ON RefSalesCategories.id\_sales\_category = SalesCategories.id\_sales\_category WHERE Photographs.city='New York City AND Photographs.country='UNITED STATES' AND RefPhotoAttributes.attribute\_desc = 'Sky' AND RefSalesCategory.category\_desc='Cat 1';

TESTING

D1: Photographs exists that have New York City as the city, but a country other than ‘UNITED STATES’, an attribute other than ‘Sky’ and a sales category other than ‘Cat 1’

D2: Photographs exists other than New York City as the city, but a country of ‘UNITED STATES’, an attribute other than ‘Sky’ and a sales category other than ‘Cat 1’

D3: Photographs exists other than New York City as the city and a country other than ‘UNITED STATES’, an attribute of ‘Sky’ and a sales category other than ‘Cat 1’

D4: Photographs exists other than New York City as the city and a country other than ‘UNITED STATES’, an attribute other than ‘Sky’ and a sales category of ‘Cat 1’

D5: Photographs exists other than New York City as the city and a country other than ‘UNITED STATES’, an attribute of ‘Sky’ and a sales category of ‘Cat 1’

D6: Photographs exists that have New York City as the city and a country other than ‘UNITED STATES’, an attribute of ‘Sky’ and a sales category of ‘Cat 1’

D7: Photographs exists that have New York City as the city and a country of ‘UNITED STATES’, an attribute of ‘Sky’ and a sales category of ‘Cat 1’

**2) Select the titles of the photographs taken in city "PHILADELPHIA" and the state "PA" in the "UNITED STATES" and bought by people who live in China. Return the count in a field called PhotoCount.**

SELECT Photographs.Title FROM Photographs INNER JOIN PhotoSales ON Photographs.id\_photo = PhotoSales.id\_photo INNER JOIN Customers ON PhotoSales.id\_customer = Customers.id\_customer WHERE Photographs.city='PHILADELPHIA' AND Photographs.state='PA' and Photographs.country='UNITED STATES' and Customers.country = 'CHINA';

**TESTING**

DS 1: the city and state matching for the photographer but not the county; a customer exists in China

DS 2: the city matching for the photographer, but not the state or country; a customer exists in China

DS 3: more than one record that matches both customer and photographer

DS 4: one record that matches both customer and photographer

DS 5 : no data that matches both customer and photographer

DS 6: more than one record that matches the photographer no records match the customer

DS 4: one record that matches the photographer, not records match the customer

**3) Select the titles of all photographs taken by Jeff Salvage that were taken outside the United States and have sold at least one print. The result set should contain (title, date\_taken). Order the results by date\_taken and then title.**

SELECT Photographs.title, Photographs.date\_taken FROM Photographs INNER JOIN PhotoSales ON Photographs.id\_photo = PhotoSales. id\_photo INNER JOIN Photographers ON Photographers.id\_photographer = Photographs.id\_photographer WHERE Photographers.first\_name = 'Jeff' and Photographers.last\_name = 'Salvage' ORDER BY Photographs.date\_taken desc, Photographs.title

D1: More than 1 photo exists from counties other than the USA on the same day for Jeff Salvage as well as some photos as the only photos taken that day that have been sold. Titles of the photos are different.

D2: Only 1 photo exists from counties other than the USA on the same day for Jeff Salvage, but there are more than one photo on different day that have been sold. Titles of the photos are different.

D3: Only 1 photo exists from counties other than the USA on the same day for Jeff Salvage, but there are more than one photo on different day that have been sold. Titles of the photos are the same.

D4: Photo only exist from the USA for Jeff Salvage. There are more than one photo on a given day and only a photo on another day that have been sold. Titles of the photos vary.

D5: Photo exist that were taken outside the USA that have been sold, but they are not taken by Jeff Salvage

D6: More than 1 photo exists from counties other than the USA on the same day for Jeff Salvage as well as some photos as the only photos taken that day that have not been sold. Some photos exist for Jeff Salvage in other countries, but they have not been sold. No other photos exist for Jeff Salvage. Titles of the photos are different.

**4) Select Title, photographer First Name, photographer Last Name of the photographs that were taken with a piece of equipment containing the word “Flash.” Order the results by Last Name, First Name, Title all in Ascending order.**

SELECT Photographs.Title, Photographers.first\_name, Photographers.last\_name FROM Photographers INNER JOIN Photographs ON Photographers.id\_photographer = Photographs.id\_photographer INNER JOIN PhotoEquipment ON PhotoEquipment.id\_photo = Photographs.id\_photo INNER JOIN RefEquipmentUsed ON RefEquipmentUsed.id\_equipment = PhotoEquipment.id\_equipment WHERE RefEquipmentUsed.equipment\_desc = '%Flash%' ORDER BY Photographers.last\_name, Photographers.first\_name, Photographs.Title;

D1: Photographs exist. Photographers exists, Flash exists in the Reference table, no photographs have the flash assigned

D2: Photographs exist. Photographers exists, Flash exists in the Reference table, one photograph has the flash assigned. Flash is in the description as ‘Flash’

D3: Photographs exist. Photographers exists, Flash exists in the Reference table, one photograph has the flash assigned. Flash is in the description as ‘ETTL Flash’

D4: Photographs exist. Photographers exists, Flash exists in the Reference table, one photograph has the flash assigned. Flash is in the description as ‘ETTL Flash V2’

D5: Photographs exist. Photographers exists, Flash exists in the Reference table, two or more photographs have the flash assigned. Flash is in the description as ‘ETTL Flash V2’

HOMEWORK #1

**1) Select the city (ies) where the customers live that bought the most photographs. If more than one city is tied for the most return the names of those cities ordered by city/state/country name.**

CREATE TABLE Customers

(

id\_customer SERIAL,

first\_name char(20),

last\_name char(20),

address1 char(50),

address2 char(50),

city char(20),

state char(20),

zip char(9),

country char(20),

PRIMARY KEY (id\_customer)

);

CREATE UNIQUE INDEX customer\_name\_index on Customers (last\_name, first\_name);

CREATE TABLE PhotoSales

(

id\_sale SERIAL,

id\_customer int,

id\_photo int,

quanity\_sold int,

id\_sale\_price float,

id\_tax int,

sale\_date date,

PRIMARY KEY (id\_sale)

);

D1: Photographs/Photographers exist, no photographs were sold.

D2: Photographs/Photographers exist, some photographs were sold to customers in different city/state/county combinations. One city/state/county has the most.

D3: Photographs/Photographers exist, some photographs were sold to customers in different city/state/county combinations. Two city/state/county has the most.

D4: Photographs/Photographers exist, some photographs were sold to customers in different city/state/county combinations. A city is repeated but has a different state and country. The combination of state and country for the customer is different, but the total photos sold is the highest for the data set.

D5: Photographs/Photographers exist, some photographs were sold to customers in different city/state/county combinations. A city is repeated but has a different state and country. The combination of state and country for the customer is different, but the total photos sold is the highest for the data set.

D6: Photographs/Photographers exist, some photographs were sold to customers in different city/state/county combinations. A city/state is repeated but has a different country. The combination of the countries for the customer is different, but the total photos sold is the highest for the data set.

D7: Photographs/Photographers exist, some photographs were sold to customers in different city/state/county combinations. A city/country is repeated but has a different state. The combination of the states for the customer is different, but the total photos sold is the highest for the data set.

**2) Select the piece of equipment used to take more photographs than any other piece of equipment. Return the name of the piece of equipment. If more than one piece is tied for the most, return both names ordered alphabetically by the equipment’s name.**

SELECT RefEquipmentUsed.equipment\_desc FROM Photographs INNER JOIN PhotoEquipment

ON Photographs.id\_Photo = PhotoEquipment.id\_Photo

INNER JOIN RefEquipmentUsed

ON PhotoEquipment.ID\_Equipment = RefEquipmentUsed.IDEquipment

GROUP BY RefEquipmentUsed.ID\_Equipment

HAVING COUNT(RefEquipmentUsed.ID\_Equipment) =

(Select COUNT(RefEquipmentUsed.ID\_Equipment) AS EquipmentCount FROM Photographs INNER JOIN PhotoEquipment

ON Photographs.id\_Photo = PhotoEquipment.id\_Photo

INNER JOIN RefEquipmentUsed

ON PhotoEquipment.ID\_Equipment = RefEquipmentUsed.IDEquipment

GROUP BY RefEquipmentUsed.ID\_Equipment

ORDER BY EquipmentCount DESC LIMIT 1)

ORDER BY RefEquipmentUsed.equipment\_desc;

D1: Photographers and photographs exists. Some photographs use equipment. Create one that is used more than any other.

D2: Photographers and photographs exists. Some photographs use equipment. Create two that is used more than any other.

D9: Photographers and photographs exists. No photographs use equipment.

D8: No data in any tables other than the reference tables

**3) Return the names of the photographers who never took a photo using a tripod**

SELECT Photographs.id\_photographer FROM Photographs

WHERE Photographs.id\_photographer NOT IN

(

Select Photographs.ID\_Photographer FROM Photographs INNER JOIN PhotoEquipment

ON Photographs.id\_Photo = PhotoEquipment.id\_Photo

INNER JOIN RefEquipmentUsed

ON PhotoEquipment.ID\_Equipment = RefEquipmentUsed.IDEquipment

WHERE RefEquipmentUsed.EquipmentDesc LIKE '%Tripod%')

D1: Photographers exists, Photographs exist, equipment used exist, but no equipment exists with the term Tripod in the desc

D2: Photographers exist, Photographs Exist, some equipment for some photos use equipment with the term Tripod in the desc

D3: Photographers exist, Photographs Exist, all equipment for some photos use equipment with the term Tripod in the desc

**4) Return the names of the person(people) who sold the most photographs without using a shutter release. In the case of a tie, order the results by the last name and first name of the photographers**

SELECT Photographs.id\_photographer, count(photographs.id\_photo) as photo\_cnt FROM Photographs INNER JOIN PhotoSales

ON Photographs.id\_photo = PhotoSales.id\_photo

WHERE Photographs.id\_photographer NOT IN

(

Select Photographs.ID\_Photo FROM Photographs INNER JOIN PhotoEquipment

ON Photographs.id\_Photo = PhotoEquipment.id\_Photo

INNER JOIN RefEquipmentUsed

ON PhotoEquipment.ID\_Equipment = RefEquipmentUsed.IDEquipment

WHERE RefEquipmentUsed.EquipmentDesc LIKE '%Shutter Release%')

GROUP BY Photographs.id\_photographer

ORDER BY photo\_cnt DESC LIMIT 1;

D1) Photographer, Photographs exist. All use a shutter release. None are sold.

D2) Photographer, Photographs exist. All use a shutter release. All are sold.

D3) Photographer, Photographs exist. All use a shutter release. Some are sold.

D4) Photographer, Photographs exist. One photographer’s photos (More than 1) use a shutter release. Some are sold.

D5) Photographer, Photographs exist. One photographer’s photos (Only 1) use a shutter release. Some are sold.

D6) Photographer, Photographs exist. One photographer’s photos (2 or more) use a shutter release. Some are sold.

**5) Return the name(s) of the photographers who took the largest and smallest photograph sold. If there is a tie, for the largest or smallest return the name of all tied for the largest/smallest. List the photographer(s) with the largest first and the smallest last. If there is a tie, order the results by LAST\_NAME, FIRST\_NAME**

SELECT photographers.last\_name, photographers.first\_name, 'Largest' as LS FROM photographs INNER JOIN

photographers on photographs.id\_photographer = photographers.id\_photographer WHERE width\*height IN

(select width\*height AS PhotoSize from photographs order by PhotoSize Desc Limit 1)

UNION

SELECT photographers.last\_name, photographers.first\_name, 'Smallest' as LS FROM photographs INNER JOIN

photographers on photographs.id\_photographer = photographers.id\_photographer WHERE width\*height IN

(select width\*height AS PhotoSize from photographs order by PhotoSize Asc Limit 1)

ORDER BY LS DESC, last\_name ASC, first\_name ASC;

D1) Photographers and photographs exist. No tie for the largest or smallest photograph.

D2) Photographers and photographs exist. A tie for the largest but not the smallest photograph exists

D3) Photographers and photographs exist. A tie for the smallest but not the largest photograph exists

D4) Photographers and photographs exist. A tie for both the largest and the smallest photograph exists

D8) No data in any tables other than the reference tables

Perhaps test sales rank?

\*9)

CREATE TABLE students

(ID\_STUDENT int,

ID\_CLASS int,

GRADE int);

SELECT ID\_STUDENT, ID\_CLASS, GRADE, RANK() OVER (PARTITION BY ID\_CLASS ORDER BY GRADE DESC) FROM Students;

Data I used had students in the same class with the same grade.

id\_student | id\_class | grade

------------+----------+-------

1 | 1 | 90

1 | 2 | 80

2 | 1 | 99

3 | 1 | 80

4 | 1 | 70

5 | 2 | 78

6 | 2 | 90

6 | 3 | 50

7 | 3 | 90

1 | 1 | 90

1 | 2 | 80

2 | 1 | 99

3 | 1 | 80

4 | 1 | 70

5 | 2 | 78

6 | 2 | 90

6 | 3 | 50

7 | 3 | 90

Kaley - (888) 280 0298