University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: ___4___

Date: ___2021/11/28_____

Group Number: ___120_____

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Hao Tian (Jack) Gong	44409431	b2z2b	haotiangong@hotmail.com
Weihao (Beren) Sun	71333785	u6x2b	sunweihao2019@163.com
Yuxiang (Felix) Fu	94074044	u6y2b	strive2p@student.ubc.ca

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

Milestone 4

Description of the Project

Our project is an amusement park management system built on the python-based framework Django and SQLite, as well as a Bootstrap 4 template developed by Colorlib (https://github.com/puikinsh/concept) for the system GUI. The database of the project models the ticket service, employment, and amusement facilities of the park. On the project website, all the tables in the database and entries for each query are listed in a navigating bar (dropdown menu), which makes it easy and intuitive to operate on it. Also, there are some basic statistics of the system, such as the total number of tourists and staff, the number of gifts that have been redeemed, and the number of rides in the park displayed at the beginning of the main page. For each query, the users are able to either decide the fields of the query specifically or complete an operation we have already initialized. The users are also able to see all the tables related to the query on a new page, and the result table after processing it. Moreover, from each query page, it is straightforward to go back to the home page by clicking the APMS (Amusement Park Management System) symbol.

The schema we use in our final project is a little bit different from what we have turned in previously. We have deleted the NOT NULL constraint from Cashier_Worksat so that Cashiers do not have to be bound to an arcade at all times. We added a few ON DELETE CASCADE specifications to ensure cashiers/machines can be deleted. Besides, we fixed some bugs in the SQL insertion. Regarding the database, we have decided to change from Oracle to SQLite because we found that it was extremely difficult to connect to the Oracle server in the Django framework. Instead, SQLite has intrinsic support from the Django framework and is easy to interact with.

SQL Queries

- **Insertion**: Add a tourist to the tourist list.

```
INSERT INTO Tourist VALUES (%s,%s,%s,%s), [ID, Name, Age, Arcadept]
```

- **Insertion**: Add a staff to the staff list.

```
INSERT INTO Staff VALUES (%s,%s), [WorkID ,Name]
```

- **Deletion**: Delete a machine from the machine list.

```
DELETE FROM Arcade WHERE Name = %s, [aName]
```

- **Deletion**: Delete an arcade from the arcade list.

```
DELETE FROM Machine WHERE MName = %s, [mName]
```

- **Update**: Update the Arcade a Cashier is working in

```
UPDATE Cashier_WorksAt
SET AName = %s
WHERE WorkID = %s", [arcade, cashierID]
```

- **Update**: Update the equipment, technician, and maintenance date for a specific ride.

```
UPDATE Ride_Maintains
SET WorkID = %s, EID = %s, TimeofInspection = date(%s)
WHERE RName = %s", [WorkID, EquipmentID, TimeofInspection, RideName]
```

- **Selection**: Select Tourist by Arcade Points

```
SELECT * FROM Tourist
WHERE ArcadePoints >= %s AND ArcadePoints <= %s, [lower_bound, upper_bound]
```

- **Selection**: Search Operator by keyword in the Qualification attribute

```
SELECT WorkID, op1.Qualification, RName FROM Operator_Operates_1 op1, Operator_Operates_2 op2
WHERE op1.Qualification = op2.Qualification
AND op1.Qualification LIKE %s, ["%" + querystring + "%"]
```

- **Projection**: Select arbitrary distinct three attributes from a table describing how tourist redeems gifts.

```
SELECT %s, %s, %s FROM Redeems r JOIN Tourist t ON t.ID = r.TID
[a1, a2, a3]
```

- **Join**: Join technician, staff list and ride maintained log to find the all names with qualifications of technicians who have maintained some ride

```
SELECT t.WorkID, t.Qualification, s.Name, rm.RName,
rm.PassengerLimit, rm.EID, rm.TimeofInspection
FROM Technician t, Staff s, Ride_Maintains rm
WHERE t.WorkID = s.WorkID AND s.WorkID=rm.WorkID
```

- **Aggregation with GROUP BY**: Find the total number of rides that tourists can play for each ticket type.

```
SELECT Type, COUNT(distinct RideName)
FROM Ticket_1 NATURAL LEFT OUTER JOIN TicketForRide
GROUP BY Type
```

- **Aggregation with HAVING:** Find the ticket type, price and count for the type sold at least 2 tickets.

```
CREATE View Ticket(TicketNo, Type, Price) AS
SELECT T1.TicketNo, T1.Type, T2.Price
FROM Ticket_1 T1, Ticket_2 T2 WHERE T1.Type=T2.Type

SELECT Type,  Price, Count(*) FROM Ticket
GROUP BY Type
HAVING Count(*)>1
```

- **Aggregation with HAVING:** Find the required points and category of the gift which requires at least 500 points for each category with at least 2 such gifts.

```
CREATE View Gift(GID, Category, PtsRequired) AS
SELECT G1.ID, G1.Category, G2. PointsRequired
FROM Gift_1 G1, Gift_2 G2
WHERE G1.Category=G2.Category

SELECT Category, PtsRequired, Count(*) FROM Gift
WHERE PtsRequired>=500
GROUP BY Category
HAVING Count(*)>2
```

- **Nested aggregation with GROUP BY:** we group tourists by the ticket type they purchased, and find groups that have arcade point averages greater than/less than the average of all tourists.

```
SELECT t1.Type, avg(t.ArcadePoints), count(*)
```

```
FROM Tourist t, TouristBuysTicket tb, Ticket_1 t1
WHERE t.ID = tb.TID AND tb.TicketNo = t1.TicketNo
GROUP BY t1.Type
HAVING avg(t.ArcadePoints) > (SELECT avg(ArcadePoints)
                              FROM Tourist)
```

(The sign can be changed based on user input)

- **Division:** Find tourists who played all the machines.

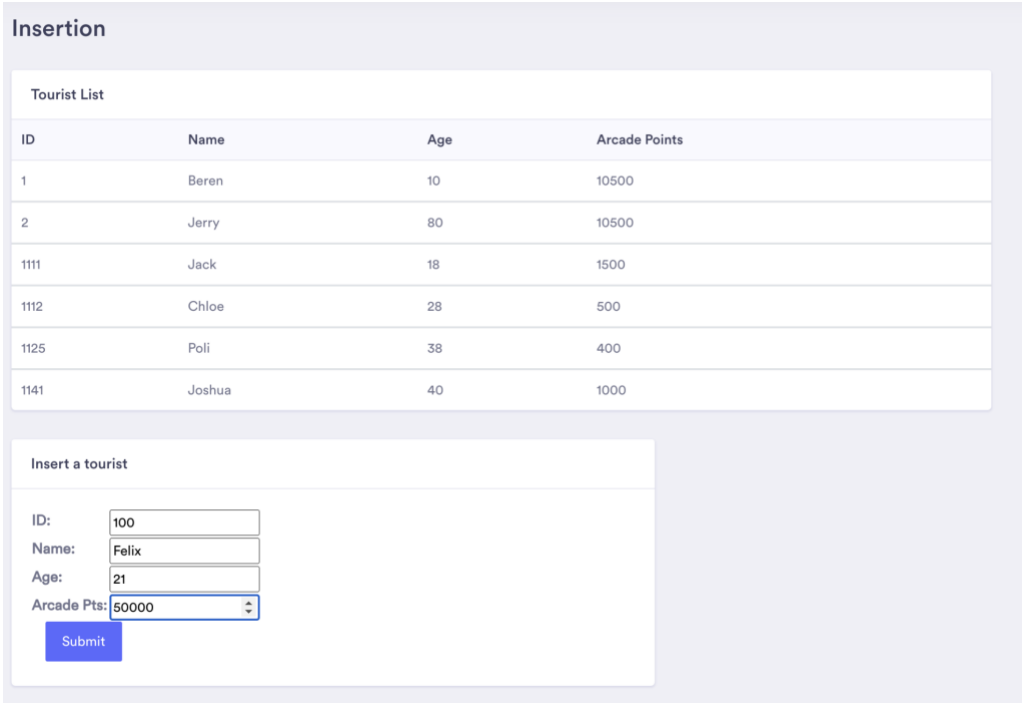
```
SELECT Name FROM Tourist T WHERE NOT EXISTS
(SELECT M.MName FROM Machine M WHERE NOT EXISTS
(SELECT TM.TID FROM TouristPlaysMachine TM
WHERE M.MName=TM.MName AND TM.TID=T.ID))
```

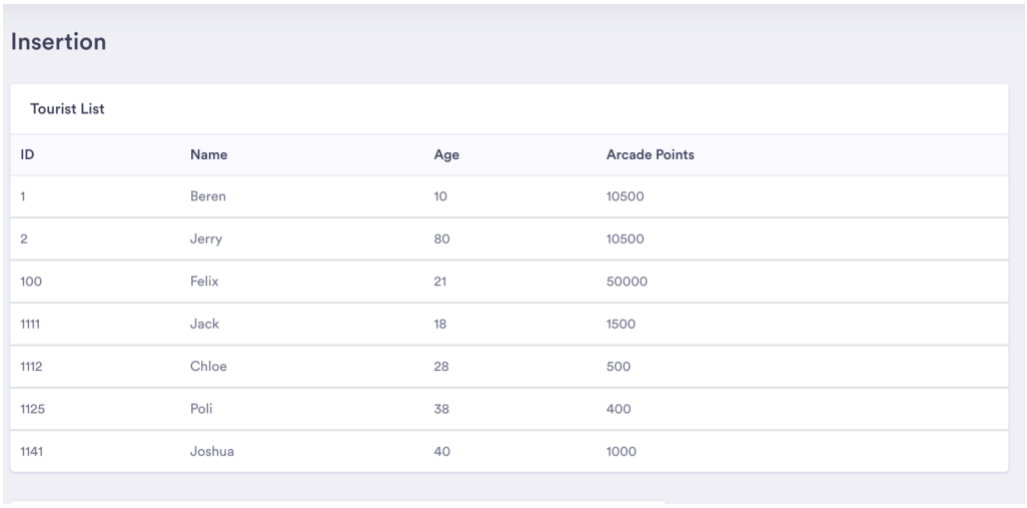
- **Division:** Find all technicians who used all the equipment.

```
SELECT Name FROM Staff S WHERE NOT EXISTS
(SELECT E.ID FROM Equipment E WHERE NOT EXISTS
(SELECT U.WID FROM Uses U WHERE U.EID=E.ID AND U.WID=S.WorkID))
```

Screenshots

- **Insertion**





Staff List

WorkID	Name
1	Bob
2	Lucy
4	Lisa
6	Susan
17	Paul
20	Lil Wayne
100	Maggie
101	Rosaline
102	Rachel
103	Peter
104	John Snow
105	Arya Stark

Insert a staff

WorkID: 200

Name: Trump

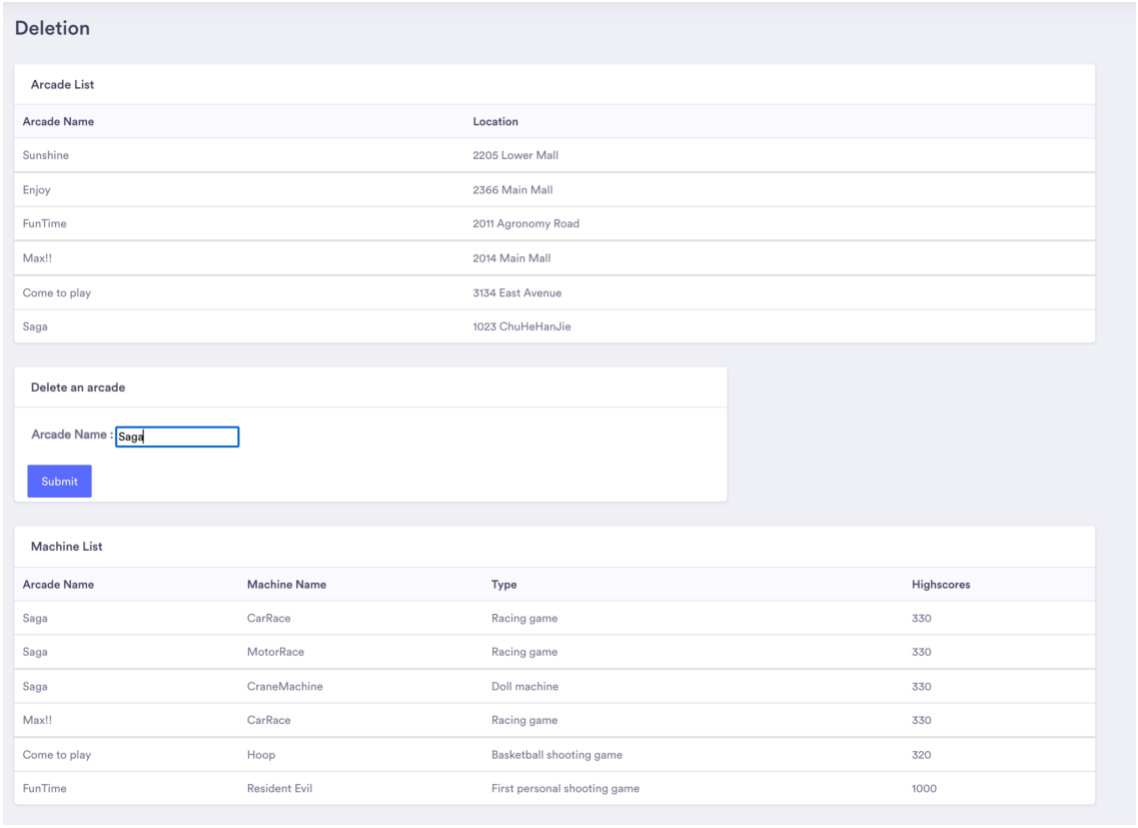
Submit

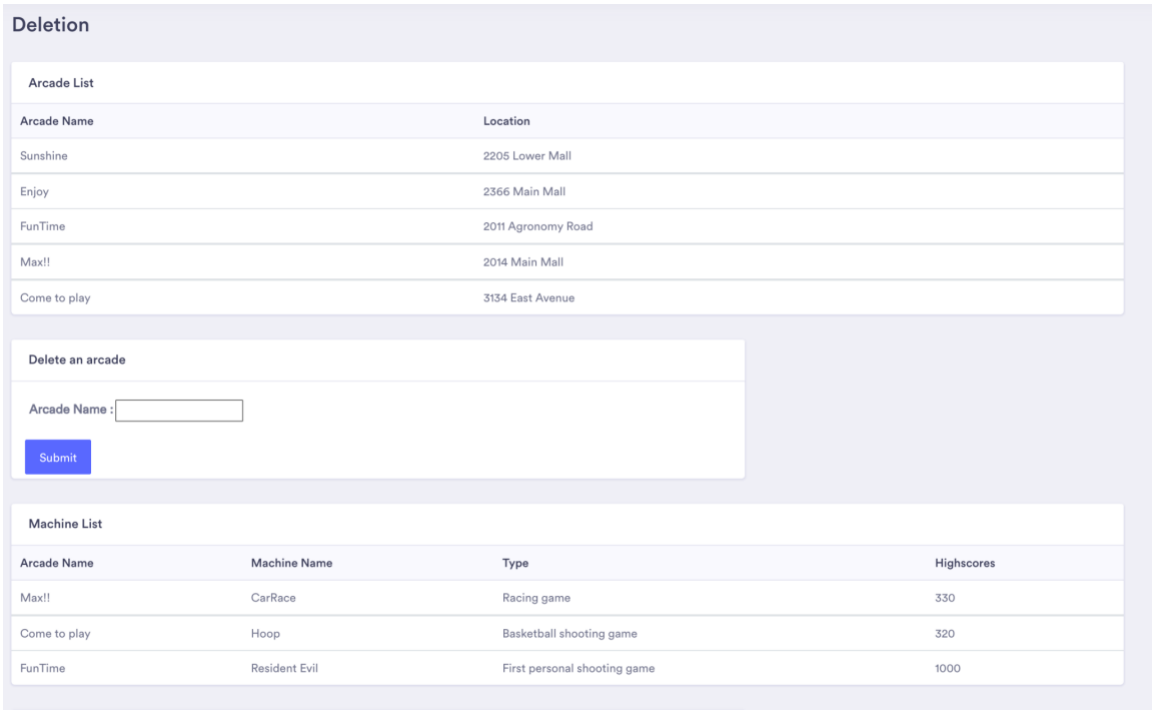
Staff List

WorkID	Name
1	Bob
2	Lucy
4	Lisa
6	Susan
17	Paul
20	Lil Wayne
100	Maggie
101	Rosaline
102	Rachel
103	Peter
104	John Snow
105	Arya Stark
200	Trump

-

- **Deletion**


Deletion
Arcade List
Arcade Name
Location
Sunshine
2205 Lower Mall
Enjoy
2366 Main Mall
FunTime
2011 Agronomy Road
Max!!
2014 Main Mall
Come to play
3134 East Avenue
Saga
1023 ChuHeHanJie
Delete an arcade
Arcade Name : Saga
Submit
Machine List
Arcade Name
Machine Name
Type
Highscores
Saga
CarRace
Racing game
330
Saga
MotorRace
Racing game
330
Saga
CraneMachine
Doll machine
330
Max!!
CarRace
Racing game
330
Come to play
Hoop
Basketball shooting game
320
FunTime
Resident Evil
First personal shooting game
1000


Deletion
Arcade List
Arcade Name
Location
Sunshine
2205 Lower Mall
Enjoy
2366 Main Mall
FunTime
2011 Agronomy Road
Max!!
2014 Main Mall
Come to play
3134 East Avenue
Delete an arcade
Arcade Name :
Submit
Machine List
Arcade Name
Machine Name
Type
Highscores
Max!!
CarRace
Racing game
330
Come to play
Hoop
Basketball shooting game
320
FunTime
Resident Evil
First personal shooting game
1000

Machine List

Arcade Name	Machine Name	Type	Highscores
Max!!	CarRace	Racing game	330
Come to play	Hoop	Basketball shooting game	320
FunTime	Resident Evil	First personal shooting game	1000

Delete a machine

Machine Name : Hoop

Submit

Machine List

Arcade Name	Machine Name	Type	Highscores
Max!!	CarRace	Racing game	330
FunTime	Resident Evil	First personal shooting game	1000

- **Update**

Cashier Work Assignments

Work ID	Arcade Name
1	FunTime
2	FunTime
4	Come to play
6	FunTime
17	Max!!
20	Saga

Cashier Work Assignments

Work ID	Arcade Name
1	FunTime
2	FunTime
4	Come to play
6	FunTime
17	Max!!
20	Saga

Ride Maintainance Log

Ride Name	Passenger Limit	Work ID	Equipment ID	Time of Inspection
Roller-coaster	40	102	11	Jan. 1, 2019
Carousel	40	104	13	Sept. 28, 2021
Flume ride	50	104	11	Nov. 28, 2021
Ferris wheel	50	101	10	Oct. 1, 2019
Haunted Mansion	15	104	13	Jan. 1, 2019
Jungle cruise	15	105	13	Sept. 21, 2019

Update Maintainance Log

Ride Name: Flume ride
Work ID: 105
Equipment ID: 10
Time of Inspection: 2020-11-28
Submit

Ride Maintainance Log

Ride Name	Passenger Limit	Work ID	Equipment ID	Time of Inspection
Roller-coaster	40	102	11	Jan. 1, 2019
Carousel	40	104	13	Sept. 28, 2021
Flume ride	50	105	10	Nov. 28, 2020
Ferris wheel	50	101	10	Oct. 1, 2019
Haunted Mansion	15	104	13	Jan. 1, 2019
Jungle cruise	15	105	13	Sept. 21, 2019

- Selection

Selection

Tourist List

ID	Name	Age	Arcade Points
1	Beren	10	10500
2	Jerry	80	10500
1111	Jack	18	1500
1112	Chloe	28	500
1125	Poli	38	400
1141	Joshua	40	1000

Select Tourist by Arcade Points

Arcade points >= 500
Arcade points <= 2000
Submit

Results:

ID	Name	Age	Arcade Points
1111	Jack	18	1500
1112	Chloe	28	500
1141	Joshua	40	1000

Operator List

Work ID	Qualifications	Assigned Ride
1	Qualified to operate the Carousel	Carousel
2	Qualified to operate the Carousel	Carousel
4	Qualified to operate the Ferris wheel	Ferris wheel
6	Qualified to operate the Jungle cruise	Jungle cruise
17	Qualified to operate the Roller-coaster	Roller-coaster
20	Qualified to operate the Flume ride	Flume ride

Search Operator by Qualification

Search for operators that has qualification: Roller

Submit

Results:

WorkID	Qualification	Assigned Ride
17	Qualified to operate the Roller-coaster	Roller-coaster

- **Projection**

Tourist table joins Redeems table

Gift ID	Tourist ID	Tourist Name	Age	Arcade Points
3	1111	Jack	18	1500
5	1112	Chloe	28	500
6	1125	Poli	38	400
8	1141	Joshua	40	1000
20	1	Beren	10	10500
25	2	Jerry	80	10500

Projection (any three distinct attributes)

Select the attribute you are interested in: Gift ID

Select the attribute you are interested in: Tourist Name

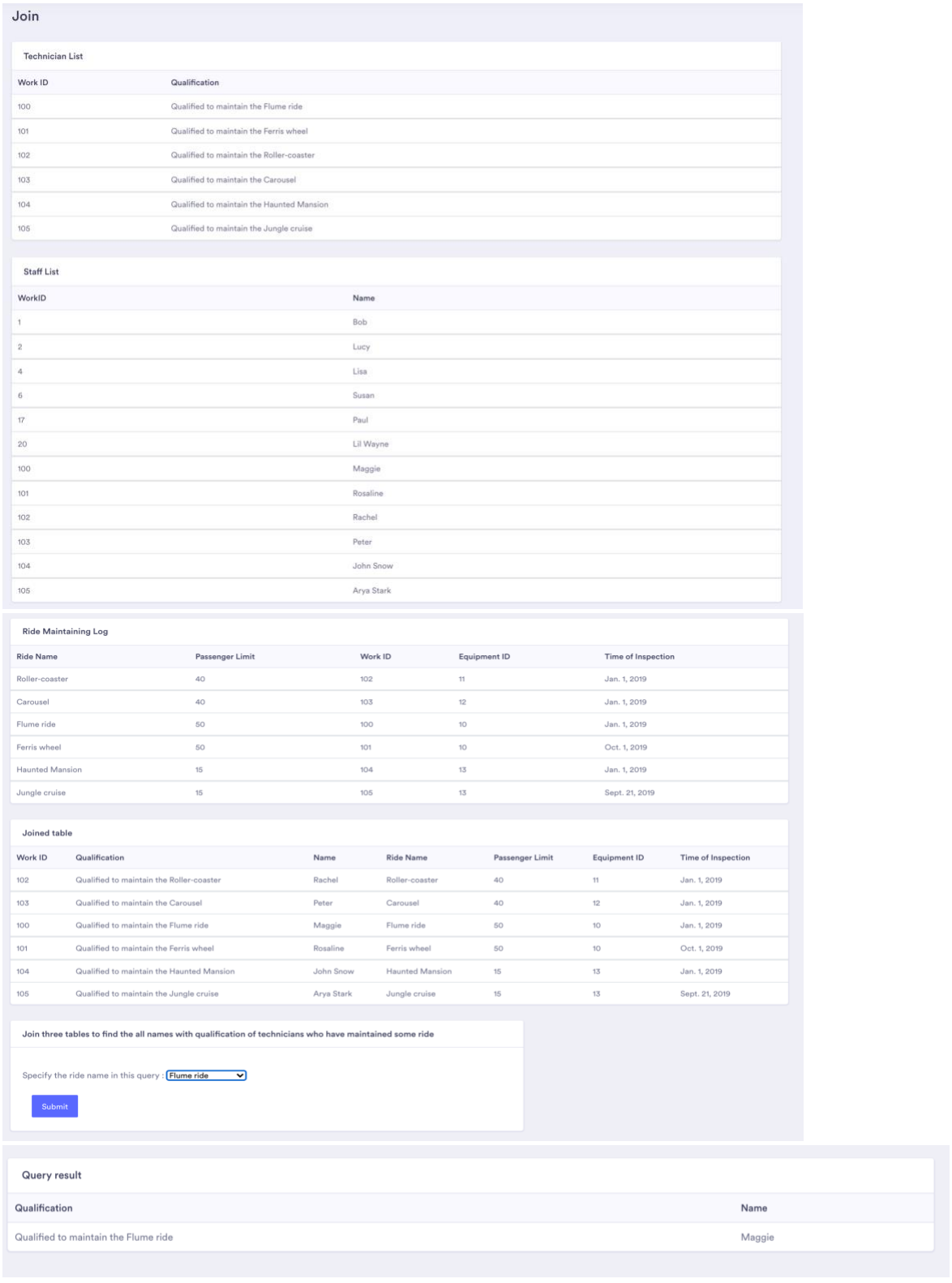
Select the attribute you are interested in: Arcade Points

Submit

Result

Gift ID	Tourist Name	Arcade Points
3	Jack	1500
5	Chloe	500
6	Poli	400
8	Joshua	1000
20	Beren	10500
25	Jerry	10500

- **Join**



Join

Technician List

Work ID	Qualification
100	Qualified to maintain the Flume ride
101	Qualified to maintain the Ferris wheel
102	Qualified to maintain the Roller-coaster
103	Qualified to maintain the Carousel
104	Qualified to maintain the Haunted Mansion
105	Qualified to maintain the Jungle cruise

Staff List

WorkID	Name
1	Bob
2	Lucy
4	Lisa
6	Susan
17	Paul
20	Lil Wayne
100	Maggie
101	Rosaline
102	Rachel
103	Peter
104	John Snow
105	Arya Stark

Ride Maintaining Log

Ride Name	Passenger Limit	Work ID	Equipment ID	Time of Inspection
Roller-coaster	40	102	11	Jan. 1, 2019
Carousel	40	103	12	Jan. 1, 2019
Flume ride	50	100	10	Jan. 1, 2019
Ferris wheel	50	101	10	Oct. 1, 2019
Haunted Mansion	15	104	13	Jan. 1, 2019
Jungle cruise	15	105	13	Sept. 21, 2019

Joined table

Work ID	Qualification	Name	Ride Name	Passenger Limit	Equipment ID	Time of Inspection
102	Qualified to maintain the Roller-coaster	Rachel	Roller-coaster	40	11	Jan. 1, 2019
103	Qualified to maintain the Carousel	Peter	Carousel	40	12	Jan. 1, 2019
100	Qualified to maintain the Flume ride	Maggie	Flume ride	50	10	Jan. 1, 2019
101	Qualified to maintain the Ferris wheel	Rosaline	Ferris wheel	50	10	Oct. 1, 2019
104	Qualified to maintain the Haunted Mansion	John Snow	Haunted Mansion	15	13	Jan. 1, 2019
105	Qualified to maintain the Jungle cruise	Arya Stark	Jungle cruise	15	13	Sept. 21, 2019

Join three tables to find the all names with qualification of technicians who have maintained some ride

Specify the ride name in this query : Flume ride

Submit

Query result

Qualification	Name
Qualified to maintain the Flume ride	Maggie

- **Aggregation with Group By**

Aggregation with Group By

Ticket for ride joins ticket list (left natural join)

Ticket Number	Type of Ticket	Ride Name
12	Ferris wheel only	Ferris wheel
13	Ferris wheel only	None
14	Ferris wheel only	None
100	Combo_1	None
123	Adolescent	Ferris wheel
123	Adolescent	Flume ride
123	Adolescent	Haunted Mansion
123	Adolescent	Jungle cruise
123	Adolescent	Roller-coaster
124	Adult	Ferris wheel
124	Adult	Flume ride
124	Adult	Haunted Mansion
124	Adult	Jungle cruise
124	Adult	Roller-coaster
125	Combo_1	Carousel
125	Combo_1	Ferris wheel
126	Combo_1	Carousel
126	Combo_1	Ferris wheel
200	Combo_2	Haunted Mansion
200	Combo_2	Jungle cruise
200	Combo_2	Roller-coaster
201	Combo_2	Haunted Mansion
200	Combo_2	Haunted Mansion
200	Combo_2	Jungle cruise
200	Combo_2	Roller-coaster
201	Combo_2	Haunted Mansion
201	Combo_2	Jungle cruise
201	Combo_2	Roller-coaster
202	Combo_2	Haunted Mansion
202	Combo_2	Jungle cruise
202	Combo_2	Roller-coaster
213	Senior	None

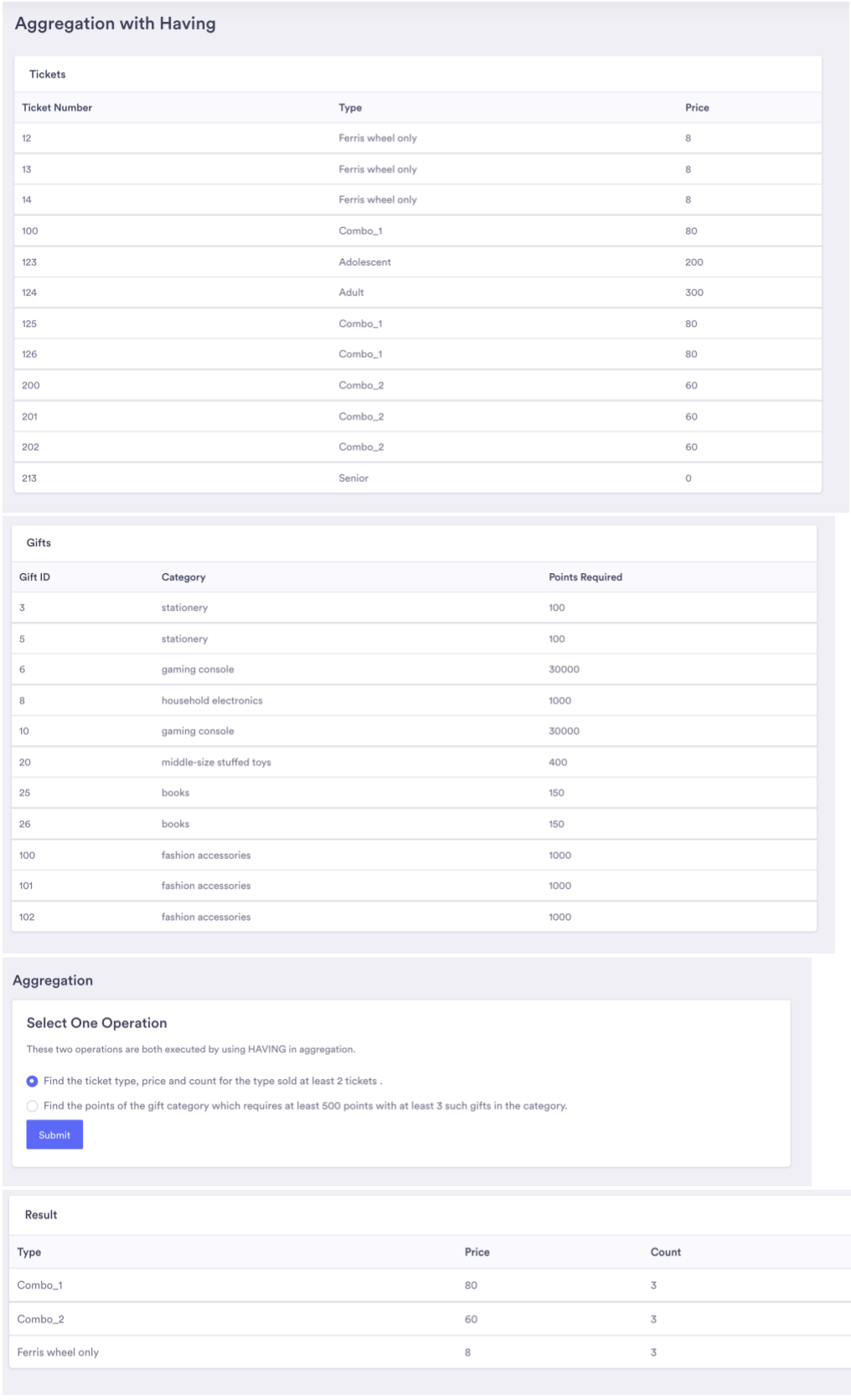
Query: Find the total number of rides that tourists can play for each ticket type.

Submit

Result

Type of ticket	Count
Adolescent	5
Adult	5
Combo_1	2
Combo_2	3
Ferris wheel only	1
Senior	0

- **Aggregation with Having**



Aggregation with Having

Tickets

Ticket Number	Type	Price
12	Ferris wheel only	8
13	Ferris wheel only	8
14	Ferris wheel only	8
100	Combo_1	80
123	Adolescent	200
124	Adult	300
125	Combo_1	80
126	Combo_1	80
200	Combo_2	60
201	Combo_2	60
202	Combo_2	60
213	Senior	0

Gifts

Gift ID	Category	Points Required
3	stationery	100
5	stationery	100
6	gaming console	30000
8	household electronics	1000
10	gaming console	30000
20	middle-size stuffed toys	400
25	books	150
26	books	150
100	fashion accessories	1000
101	fashion accessories	1000
102	fashion accessories	1000

Aggregation

Select One Operation

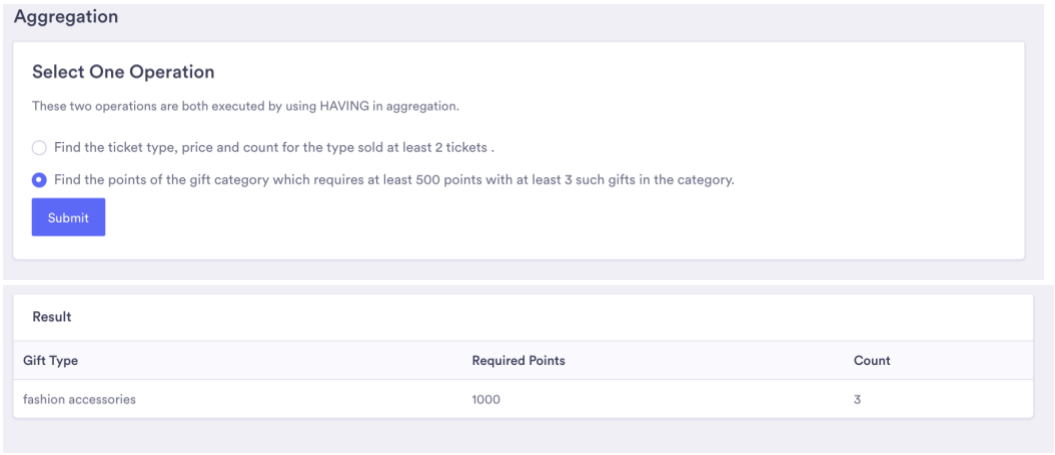
These two operations are both executed by using HAVING in aggregation.

- (•) Find the ticket type, price and count for the type sold at least 2 tickets .
- ( ) Find the points of the gift category which requires at least 500 points with at least 3 such gifts in the category.

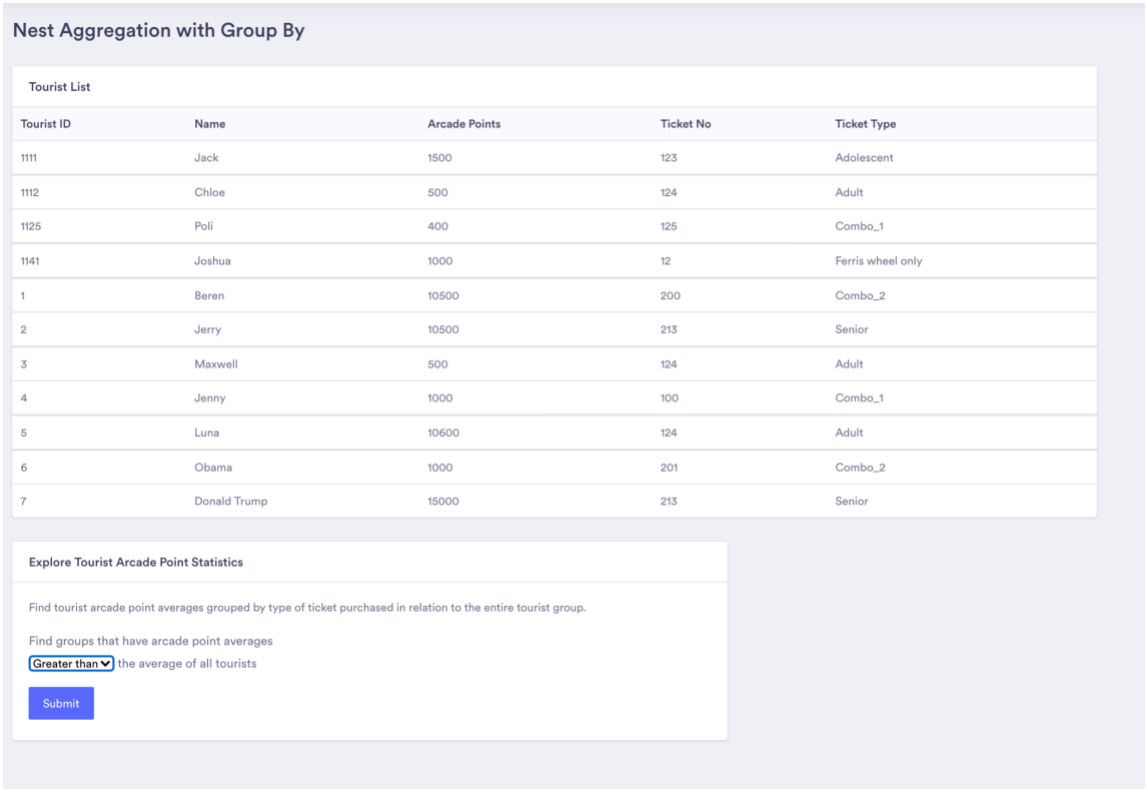
Submit

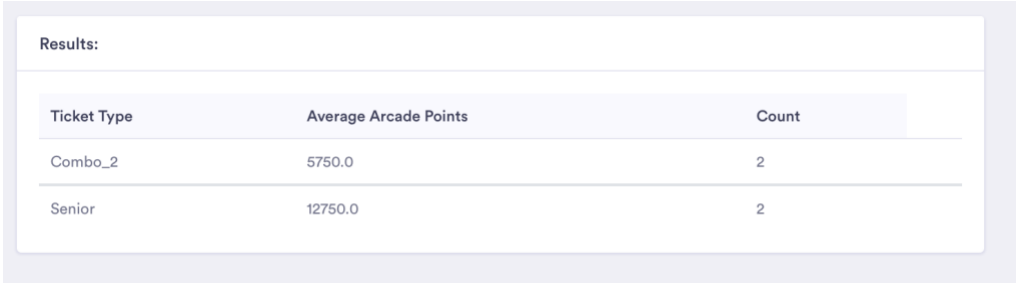
Result

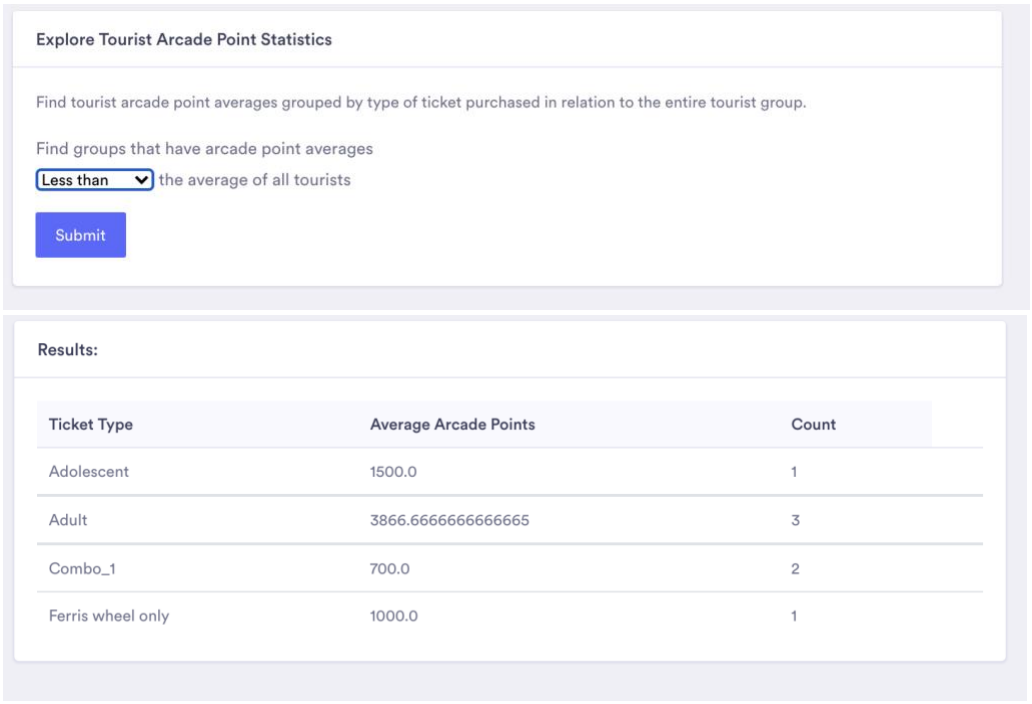
Type	Price	Count
Combo_1	80	3
Combo_2	60	3
Ferris wheel only	8	3



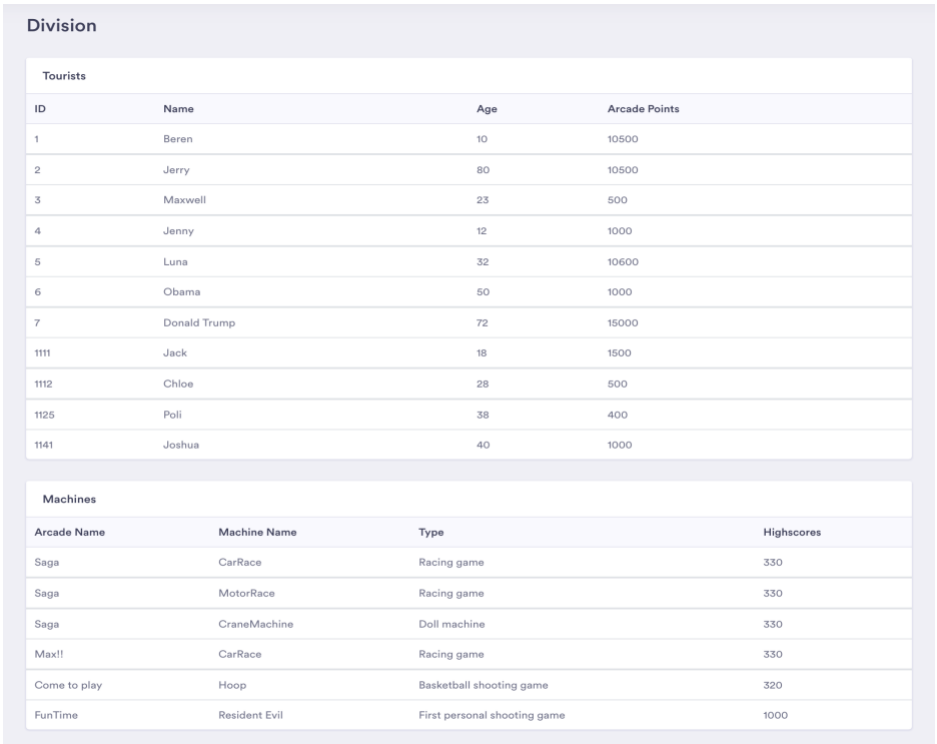
- **Nested Aggregation with Group By**






Explore Tourist Arcade Point Statistics
Find tourist arcade point averages grouped by type of ticket purchased in relation to the entire tourist group.
Find groups that have arcade point averages
Less than
the average of all tourists
Submit
Results:
Ticket Type
Average Arcade Points
Count
Adolescent
1500.0
1
Adult
3866.6666666666665
3
Combo_1
700.0
2
Ferris wheel only
1000.0
1

- **Division**


Division
Tourists
ID
Name
Age
Arcade Points
1
Beren
10
10500
2
Jerry
80
10500
3
Maxwell
23
500
4
Jenny
12
1000
5
Luna
32
10600
6
Obama
50
1000
7
Donald Trump
72
15000
1111
Jack
18
1500
1112
Chloe
28
500
1125
Poli
38
400
1141
Joshua
40
1000
Machines
Arcade Name
Machine Name
Type
Highscores
Saga
CarRace
Racing game
330
Saga
MotorRace
Racing game
330
Saga
CraneMachine
Doll machine
330
Max!!
CarRace
Racing game
330
Come to play
Hoop
Basketball shooting game
320
FunTime
Resident Evil
First personal shooting game
1000

Tourist Plays Machine

Tourist ID	Arcade Name	Machine Name	Points Earned
1111	Saga	CarRace	14
1111	Max!!	CarRace	15
1112	Saga	CraneMachine	10
2	Saga	CarRace	20
1125	Come to play	Hoop	10
1141	FunTime	Resident Evil	8
1111	Come to play	Hoop	8
1111	Saga	MotorRace	10
1111	Saga	CraneMachine	13
1111	FunTime	Resident Evil	20
1112	Max!!	CarRace	15
1112	Saga	MotorRace	10
1112	Come to play	Hoop	30
1112	FunTime	Resident Evil	15
1112	Saga	CarRace	10

Equipments

ID
10
11
12
13
14
15

Technicians

Work ID	Technician Name	Equipment ID
100	Maggie	10
101	Rosaline	10
102	Rachel	11
103	Peter	12
104	John Snow	13
105	Arya Stark	14
100	Maggie	11
100	Maggie	12
100	Maggie	13
100	Maggie	14
100	Maggie	15

Division

Select One Operation

These two operations are both executed by divisions.

- (•) Find tourists who played all the machines regardless in which arcade.
- ( ) Find technicians who used all the equipments.

Submit

Result

Name
Jack
Chloe

Division

Select One Operation

These two operations are both executed by divisions.

- ○ Find tourists who played all the machines regardless in which arcade.
- ◉ Find technicians who used all the equipments.

Submit

Result

Name
Maggie