University of British Columbia, Vancouver

Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 4
Date: June 14, 2023
Group Number: 2

| Name | Student Number | CS Alias | Preferred Email Address |
|---|---|---|---|
| Beth Koschel | 28150423 | g9n3c@ugrad.cs.ubc.ca | bethanylkoschel@gmail.com |
| Harbir Bajwa | 20972261 | p8i2e@ugrad.cs.ubc.ca | harbirbajwa@hotmail.com |
| Apram Ahuja | 14367403 | v0w2d@ugrad.cs.ubc.ca | apram235ahuja@gmail.com |

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

## 2. Create Tables and Populate Data

See file named ‘create-zoo.sql’ in the Github repo to view script that was used to create all the tables and data in the database.

```
drop table Zoo1 CASCADE CONSTRAINTS;

drop table Zoo2 CASCADE CONSTRAINTS;

drop table ZooManager CASCADE CONSTRAINTS;

drop table Order1 CASCADE CONSTRAINTS;

drop table Ticket1 CASCADE CONSTRAINTS;

drop table Ticket2 CASCADE CONSTRAINTS;

drop table Animal1 CASCADE CONSTRAINTS;

drop table FeatureAnimal1 CASCADE CONSTRAINTS;

drop table ResidentAnimal1 CASCADE CONSTRAINTS;

drop table Habitat1 CASCADE CONSTRAINTS;

drop table Habitat2 CASCADE CONSTRAINTS;

drop table Exhibits CASCADE CONSTRAINTS;

drop table TemporaryVisit CASCADE CONSTRAINTS;

drop table FoodVendor1 CASCADE CONSTRAINTS;

drop table FoodVendor2 CASCADE CONSTRAINTS;


CREATE TABLE Zoo1 (

zName CHAR(32) PRIMARY KEY,

city CHAR(32)

);


INSERT ALL

INTO Zoo1 (zName, city) VALUES ('Toronto Zoo', 'Toronto')

INTO Zoo1 (zName, city) VALUES ('Greater Vancouver Zoo',
'Vancouver')
INTO Zoo1 (zName, city) VALUES ('Winnipeg Zoo', 'Winnipeg')
INTO Zoo1 (zName, city) VALUES ('Victoria Bug Zoo', 'Victoria')
INTO Zoo1 (zName, city) VALUES ('Edmonton Valley Zoo', 'Edmonton')
SELECT 1 FROM DUAL;

CREATE TABLE Zoo2 (
zooID INTEGER PRIMARY KEY,
zName CHAR(32),
province CHAR(32),
employeeID INTEGER NOT NULL,
UNIQUE (employeeID)
);

INSERT ALL
INTO Zoo2 (zooID, zName, province, employeeID) VALUES (1, 'Toronto
Zoo', 'ON', 101)
INTO Zoo2 (zooID, zName, province, employeeID) VALUES (2, 'Greater
Vancouver Zoo', 'BC', 102)
INTO Zoo2 (zooID, zName, province, employeeID) VALUES (3, 'Winnipeg
Zoo', 'MB', 103)
INTO Zoo2 (zooID, zName, province, employeeID) VALUES (4, 'Victoria
Bug Zoo', 'BC', 104)
INTO Zoo2 (zooID, zName, province, employeeID) VALUES (5, 'Edmonton
Valley Zoo', 'AB', 105)
INTO Zoo2 (zooID, zName, province, employeeID) VALUES (6, 'Montreal
Biodome', 'QC', 106)
SELECT 1 FROM DUAL;

CREATE TABLE ZooManager (
eName CHAR(32),
employeeID INTEGER,
joinDate DATE,
zooID INTEGER NOT NULL,
UNIQUE (zooID),
PRIMARY KEY (employeeID),
FOREIGN KEY (zooID)
REFERENCES Zoo2(zooID)
ON DELETE CASCADE
);

INSERT ALL
INTO ZooManager (eName, employeeID, joinDate, zooID) VALUES ('John
Smith', 101, DATE '2022-01-15', 1)
INTO ZooManager (eName, employeeID, joinDate, zooID) VALUES ('Emily
Johnson', 102, DATE '2021-06-28', 2)
INTO ZooManager (eName, employeeID, joinDate, zooID) VALUES
('Michael Williams', 103, DATE '2023-03-10', 3)
INTO ZooManager (eName, employeeID, joinDate, zooID) VALUES ('Sophia
Brown', 104, DATE '2022-09-05', 4)
INTO ZooManager (eName, employeeID, joinDate, zooID) VALUES ('Daniel
Davis', 105, DATE '2021-12-01', 5)
INTO ZooManager (eName, employeeID, joinDate, zooID) VALUES ('Olivia
Wilson', 106, DATE '2023-04-18', 6)
SELECT 1 FROM DUAL;

ALTER TABLE Zoo2
ADD FOREIGN KEY (employeeID)
REFERENCES ZooManager(employeeID)
ON DELETE CASCADE;

CREATE TABLE Order1 (
orderNumber INTEGER,
orderDate DATE,
paymentMethod CHAR(6),
zooID INTEGER NOT NULL,
PRIMARY KEY (orderNumber),
FOREIGN KEY (zooID)
REFERENCES Zoo2(zooID)
ON DELETE CASCADE
);
```

```
INSERT ALL
INTO Order1 (orderNumber, orderDate, paymentMethod, zooID) VALUES
(101, DATE '2023-01-15', 'CC', 1)
INTO Order1 (orderNumber, orderDate, paymentMethod, zooID) VALUES
(102, DATE '2023-02-20', 'PayPal', 2)
INTO Order1 (orderNumber, orderDate, paymentMethod, zooID) VALUES
(103, DATE '2023-03-10', 'CC', 1)
INTO Order1 (orderNumber, orderDate, paymentMethod, zooID) VALUES
(104, DATE '2023-04-05', 'Cash', 3)
INTO Order1 (orderNumber, orderDate, paymentMethod, zooID) VALUES
(105, DATE '2023-05-18', 'PayPal', 2)
INTO Order1 (orderNumber, orderDate, paymentMethod, zooID) VALUES
(106, DATE '2023-06-01', 'CC', 4)
SELECT 1 FROM DUAL;


CREATE TABLE Ticket1 (
ticketType CHAR(7) PRIMARY KEY,
price REAL
);


INSERT ALL
INTO Ticket1 (ticketType, price) VALUES ('Adult', 20.99)
INTO Ticket1 (ticketType, price) VALUES ('Child', 10.99)
INTO Ticket1 (ticketType, price) VALUES ('Senior', 15.99)
INTO Ticket1 (ticketType, price) VALUES ('Student', 17.99)
INTO Ticket1 (ticketType, price) VALUES ('Family', 45.99)
INTO Ticket1 (ticketType, price) VALUES ('Group', 12.99)
SELECT 1 FROM DUAL;


CREATE TABLE Ticket2 (
ticketID INTEGER,
orderNumber INTEGER,
ticketType CHAR(7),
PRIMARY KEY (ticketID, orderNumber),
FOREIGN KEY (orderNumber)
REFERENCES Order1(orderNumber)
ON DELETE CASCADE
);

INSERT ALL
INTO Ticket2 (ticketID, orderNumber, ticketType) VALUES (1001, 101,
'Adult')
INTO Ticket2 (ticketID, orderNumber, ticketType) VALUES (1002, 102,
'Child')
INTO Ticket2 (ticketID, orderNumber, ticketType) VALUES (1003, 103,
'Adult')
INTO Ticket2 (ticketID, orderNumber, ticketType) VALUES (1004, 104,
'Senior')
INTO Ticket2 (ticketID, orderNumber, ticketType) VALUES (1005, 105,
'Child')
INTO Ticket2 (ticketID, orderNumber, ticketType) VALUES (1006, 106,
'Student')
SELECT 1 FROM DUAL;

CREATE TABLE Exhibits (
exhibitID INTEGER,
exhibitTitle CHAR(32),
visitorCapacity INTEGER,
popularityRating INTEGER,
zooID INTEGER NOT NULL,
PRIMARY KEY (exhibitID),
FOREIGN KEY (zooID)
REFERENCES Zoo2(zooID)
ON DELETE CASCADE
);

INSERT ALL
INTO Exhibits (exhibitID, exhibitTitle, visitorCapacity,
popularityRating, zooID)
VALUES (1, 'Big Cat', 100, 9, 1)
```

```
INTO Exhibits (exhibitID, exhibitTitle, visitorCapacity,
popularityRating, zooID)
VALUES (2, 'Rainforest Adventure', 150, 8, 2)
INTO Exhibits (exhibitID, exhibitTitle, visitorCapacity,
popularityRating, zooID)
VALUES (3, 'Grassland Safari', 80, 7, 1)
INTO Exhibits (exhibitID, exhibitTitle, visitorCapacity,
popularityRating, zooID)
VALUES (4, 'Desert Oasis', 50, 6, 3)
INTO Exhibits (exhibitID, exhibitTitle, visitorCapacity,
popularityRating, zooID)
VALUES (5, 'Tropical Paradise', 120, 9, 2)
INTO Exhibits (exhibitID, exhibitTitle, visitorCapacity,
popularityRating, zooID)
VALUES (6, 'Savanna Safari', 90, 7, 1)
SELECT 1 FROM DUAL;



CREATE TABLE Habitat1 (
habitatType CHAR(32),
humidity REAL,
temperature REAL,
PRIMARY KEY (habitatType)
);

INSERT ALL
INTO Habitat1 (habitatType, humidity, temperature)
VALUES ('Savanna', 60.5, 30.2)
INTO Habitat1 (habitatType, humidity, temperature)
VALUES ('Rainforest', 80.2, 25.8)
INTO Habitat1 (habitatType, humidity, temperature)
VALUES ('Desert', 20.8, 40.6)
INTO Habitat1 (habitatType, humidity, temperature)
VALUES ('Grassland', 55.1, 28.5)
```

```
INTO Habitat1 (habitatType, humidity, temperature)
VALUES ('Arctic', 30.7, -10.2)
INTO Habitat1 (habitatType, humidity, temperature)
VALUES ('Woodland', 65.3, 24.9)
SELECT 1 FROM DUAL;

CREATE TABLE Habitat2 (
habitatID INTEGER,
exhibitID INTEGER,
habitatType CHAR(32),
vegetation CHAR(32),
PRIMARY KEY (habitatID),
FOREIGN KEY (exhibitID)
REFERENCES Exhibits(exhibitID)
ON DELETE CASCADE
);

INSERT ALL
INTO Habitat2 (habitatID, exhibitID, habitatType, vegetation)
VALUES (1, 1, 'Savanna', 'Grass')
INTO Habitat2 (habitatID, exhibitID, habitatType, vegetation)
VALUES (2, 2, 'Rainforest', 'Trees')
INTO Habitat2 (habitatID, exhibitID, habitatType, vegetation)
VALUES (3, 3, 'Grassland', 'Grass')
INTO Habitat2 (habitatID, exhibitID, habitatType, vegetation)
VALUES (4, 4, 'Desert', 'Cacti')
INTO Habitat2 (habitatID, exhibitID, habitatType, vegetation)
VALUES (5, 2, 'Rainforest', 'Vines')
INTO Habitat2 (habitatID, exhibitID, habitatType, vegetation)
VALUES (6, 1, 'Savanna', 'Trees')
SELECT 1 FROM DUAL;
```

```
CREATE TABLE Animal1 (
species CHAR(32) PRIMARY KEY,
conservationStatus CHAR(4)
);


INSERT ALL
INTO Animal1 (species, conservationStatus) VALUES ('Acinonyx
jubatus', 'LC')
INTO Animal1 (species, conservationStatus) VALUES ('Panthera
pardus', 'NT')
INTO Animal1 (species, conservationStatus) VALUES ('Antilocapra
americana', 'LC')
INTO Animal1 (species, conservationStatus) VALUES ('Lamprotornis
superbus', 'LC')
INTO Animal1 (species, conservationStatus) VALUES ('Ateles
belzebuth', 'VU')
INTO Animal1 (species, conservationStatus) VALUES ('Gazella dorcas',
'LC')
INTO Animal1 (species, conservationStatus) VALUES ('Giraffa
camelopardalis', 'VU')
INTO Animal1 (species, conservationStatus) VALUES ('Panthera
tigris', 'EN')
INTO Animal1 (species, conservationStatus) VALUES ('Bison bison',
'LC')
INTO Animal1 (species, conservationStatus) VALUES ('Ploceus
cucullatus', 'LC')
INTO Animal1 (species, conservationStatus) VALUES ('Alouatta
caraya', 'LC')
INTO Animal1 (species, conservationStatus) VALUES ('Oryx gazella',
'LC')
SELECT 1 FROM DUAL;


CREATE TABLE ResidentAnimal1 (
animalID INTEGER,
habitatID INTEGER NOT NULL,
species CHAR(32),
sex CHAR(2),
age INTEGER,
healthStatus CHAR(10),
monthsAtZoo INTEGER,
favouriteFood CHAR(32),
PRIMARY KEY (animalID),
FOREIGN KEY (habitatID)
REFERENCES Habitat2(habitatID)
ON DELETE CASCADE
);

INSERT ALL
INTO ResidentAnimal1 (animalID, habitatID, species, sex, age,
healthStatus, monthsAtZoo, favouriteFood) VALUES (19, 1, 'Giraffa
camelopardalis', 'F', 9, 'Good', 18, 'Leaves')
INTO ResidentAnimal1 (animalID, habitatID, species, sex, age,
healthStatus, monthsAtZoo, favouriteFood) VALUES (20, 2, 'Panthera
tigris', 'M', 6, 'Excellent', 12, 'Meat')
INTO ResidentAnimal1 (animalID, habitatID, species, sex, age,
healthStatus, monthsAtZoo, favouriteFood) VALUES (21, 3, 'Bison
bison', 'M', 4, 'Good', 24, 'Grass')
INTO ResidentAnimal1 (animalID, habitatID, species, sex, age,
healthStatus, monthsAtZoo, favouriteFood) VALUES (22, 4, 'Ploceus
cucullatus', 'F', 3, 'Good', 9, 'Seeds')
INTO ResidentAnimal1 (animalID, habitatID, species, sex, age,
healthStatus, monthsAtZoo, favouriteFood) VALUES (23, 5, 'Alouatta
caraya', 'M', 5, 'Fair', 15, 'Fruits')
INTO ResidentAnimal1 (animalID, habitatID, species, sex, age,
healthStatus, monthsAtZoo, favouriteFood) VALUES (24, 6, 'Oryx
gazella', 'F', 7, 'Good', 6, 'Grass')
SELECT 1 FROM DUAL;

CREATE TABLE FeatureAnimal1 (
animalID INTEGER,
habitatID INTEGER NOT NULL,
species CHAR(32),
sex CHAR(2),
age INTEGER,
healthStatus CHAR(10),
originLocation CHAR(32),
PRIMARY KEY (animalID),
FOREIGN KEY (habitatID)
REFERENCES Habitat2(habitatID)
ON DELETE CASCADE
);

INSERT ALL
INTO FeatureAnimal1 (animalID, habitatID, species, sex, age,
healthStatus, originLocation) VALUES (13, 1, 'Acinonyx jubatus',
'M', 5, 'Good', 'Africa')
INTO FeatureAnimal1 (animalID, habitatID, species, sex, age,
healthStatus, originLocation) VALUES (14, 2, 'Panthera pardus', 'F',
7, 'Excellent', 'Asia')
INTO FeatureAnimal1 (animalID, habitatID, species, sex, age,
healthStatus, originLocation) VALUES (15, 3, 'Antilocapra
americana', 'M', 3, 'Good', 'North America')
INTO FeatureAnimal1 (animalID, habitatID, species, sex, age,
healthStatus, originLocation) VALUES (16, 4, 'Lamprotornis
superbus', 'F', 4, 'Good', 'Africa')
INTO FeatureAnimal1 (animalID, habitatID, species, sex, age,
healthStatus, originLocation) VALUES (17, 5, 'Ateles belzebuth',
'M', 6, 'Fair', 'South America')
INTO FeatureAnimal1 (animalID, habitatID, species, sex, age,
healthStatus, originLocation) VALUES (18, 6, 'Gazella dorcas', 'F',
2, 'Good', 'Africa')
SELECT 1 FROM DUAL;

CREATE TABLE TemporaryVisit (
startDate DATE,
endDate DATE,
animalID INTEGER,
employeeID INTEGER,
PRIMARY KEY (startDate, endDate, animalID),
FOREIGN KEY (employeeID)
REFERENCES ZooManager(employeeID)
ON DELETE CASCADE,
FOREIGN KEY (animalID)

REFERENCES FeatureAnimal1(animalID)

ON DELETE CASCADE

);


INSERT ALL

INTO TemporaryVisit (startDate, endDate, animalID, EmployeeID)
VALUES (DATE '2023-01-15', DATE '2023-01-20', 13, 101)
INTO TemporaryVisit (startDate, endDate, animalID, EmployeeID)
VALUES (DATE '2023-02-05', DATE '2023-02-12', 14, 102)
INTO TemporaryVisit (startDate, endDate, animalID, EmployeeID)
VALUES (DATE '2023-03-10', DATE '2023-03-15', 15, 103)
INTO TemporaryVisit (startDate, endDate, animalID, EmployeeID)
VALUES (DATE '2023-04-01', DATE '2023-04-07', 16, 104)
INTO TemporaryVisit (startDate, endDate, animalID, EmployeeID)
VALUES (DATE '2023-05-18', DATE '2023-05-25', 17, 105)
INTO TemporaryVisit (startDate, endDate, animalID, EmployeeID)
VALUES (DATE '2023-06-10', DATE '2023-06-17', 18, 106)
SELECT 1 FROM DUAL;


CREATE TABLE FoodVendor1 (

vName CHAR(32) PRIMARY KEY,

foodType CHAR(32)

);


INSERT ALL

INTO FoodVendor1 (vName, foodType)

VALUES ('Japadog', 'Hot Dogs')

INTO FoodVendor1 (vName, foodType)

VALUES ('Rain or Shine', 'Ice-Cream')

INTO FoodVendor1 (vName, foodType)

VALUES ('Sushi Spot', 'Japanese')

INTO FoodVendor1 (vName, foodType)

VALUES ('Taco Truck', 'Mexican')

INTO FoodVendor1 (vName, foodType)
```

```
VALUES ('Pizza Palace', 'Italian')

INTO FoodVendor1 (vName, foodType)

VALUES ('Crepes Corner', 'French')

SELECT 1 FROM DUAL;


CREATE TABLE FoodVendor2 (

vendorID INTEGER,

vName CHAR(32),

rating REAL,

zooID INTEGER NOT NULL,

PRIMARY KEY (vendorID),

FOREIGN KEY (zooID)

REFERENCES Zoo2(zooID)

ON DELETE CASCADE

);


INSERT ALL

INTO FoodVendor2 (vendorID, vName, rating, zooID)

VALUES (1, 'Japadog', 4.2, 1)

INTO FoodVendor2 (vendorID, vName, rating, zooID)

VALUES (2, 'Rain or Shine', 4.5, 2)

INTO FoodVendor2 (vendorID, vName, rating, zooID)

VALUES (3, 'Sushi Spot', 4.1, 1)

INTO FoodVendor2 (vendorID, vName, rating, zooID)

VALUES (4, 'Taco Truck', 4.3, 3)

INTO FoodVendor2 (vendorID, vName, rating, zooID)

VALUES (5, 'Pizza Palace', 4.4, 2)

INTO FoodVendor2 (vendorID, vName, rating, zooID)

VALUES (6, 'Crepes Corner', 4.0, 1)

SELECT 1 FROM DUAL;
```

Attach copy of schema and screenshots that show what data is present in each relation after SQL initialization script is run

SCHEMA AS PER MILESTONE 2:

a.

- Zoo (zooID: INTEGER, city: CHAR(32), zName: CHAR(32), province: CHAR(4), **EmployeeID**: INTEGER)
- ZooManager (eName: CHAR(32), employeeID: INTEGER, joinDate: DATE, **zooID**: INTEGER)
- Order (orderNumber: INTEGER, orderDate: DATE, paymentMethod: CHAR(4), **zooID**: INTEGER)
- Ticket (ticketID: INTEGER, **orderNumber**: INTEGER, ticketType: CHAR(4), price:REAL)
- FeatureAnimal (animalID: INTEGER, **habitatID**: INTEGER, species: CHAR(32), sex: CHAR(2), age: INTEGER, healthStatus: CHAR(4), conservationStatus: CHAR(4), originLocation: CHAR(32))
- ResidentAnimal (animalID: INTEGER, **habitatID**: INTEGER, species: CHAR(32), sex: CHAR(2), age: INTEGER, healthStatus: CHAR(4), conservationStatus: CHAR(4), monthsAtZoo: INTEGER, favouriteFood: CHAR(32))
- TemporaryVisit (startDate: DATE, endDate: DATE, **animalID**: INTEGER, **EmployeeID**: INTEGER)
    - (Combined TemporaryVisit, Makes, and the whole aggregation into one table)
- Exhibits (exhibitID: INTEGER, exhibitTitle: CHAR(32), visitorCapacity: INTEGER, popularityRating: INTEGER, **zooID**: INTEGER)
- Habitat (habitatID: INTEGER, **exhibitID**: INTEGER, habitatType: CHAR(32), humidity: REAL, temperature: REAL, vegetation: CHAR(32))
- FoodVendor (vendorID: INTEGER, vName: CHAR(32), rating: REAL, foodType: CHAR(32), **zooID**: INTEGER)

b.

**Zoo**
Primary key: zooID
Candidate key: zooID
Foreign key: employeeID NOT NULL, UNIQUE

**ZooManager**
Primary key: employeeID
Candidate key: employeeID

Foreign key: zooID NOT NULL, UNIQUE

**Order**
Primary key: orderNumber
Candidate key: orderNumber
Foreign key: zooID NOT NULL

**Ticket**
Primary key: ticketID, orderNumber
Candidate key: ticketID, orderNumber
Foreign key: orderNumber

**FeatureAnimal**
Primary key: animalID
Candidate key: animalID
Foreign key: habitatID NOT NULL

**ResidentAnimal**
Primary key: animalID
Candidate key: animalID
Foreign key: habitatID NOT NULL

**TemporaryVisit**
Primary key: {animalID, startDate, endDate}
Candidate key: {animalID, startDate, endDate}
Foreign key: animalID, employeeID

**Exhibits**
Primary key: exhibitID
Candidate key: exhibitID
Foreign key: zooID NOT NULL

**Habitat**
Primary key: habitatID
Candidate key: habitatID
Foreign key: exhibitID

**FoodVendor**
Primary key: vendorID
Candidate key: vendorID
Foreign key: zooID NOT NULL

## 3. Project Summary

The project involves developing a Zoo Animal Management System that focuses on the relationships between various entities. The system will capture and model the interactions between zoos, zoo managers, orders, animals, exhibits, habitats, and food vendors. It will provide functionality to manage exhibits, track animal visits, analyze visitor demographics, monitor revenue from ticket sales. The database will enable efficient zoo operations and aid decision-making processes related to exhibit popularity, animal selection, and vendor management.

### What the project accomplished

In this project we created an app that links to a database for managing a zoo. First we decided on an entity relation diagram to capture the relations we wanted in our zoo application. We then used that diagram to decompose and create a schema for each relation. Finally, we completed the implementation of this app. We created a SQL database involving all tables in our schema and included some sample entries. We also created a back-end in PHP to connect to the SQL server and process requests. To improve the user friendliness, we also have a front-end created with HTML/CSS. This front end makes our app easier for the user to access and use our database to manage zoo operations as described in the project summary.

### Note any differences in the final schema from the original one (if there are any differences explain why they occurred)

There were no differences in the schema from Milestone 2.

Queries

```
-- INSERT
-- "Based on all the information user inputs which adheres to the bounds in php these SQL statements insert information in Animal1 and FeatureAnimal1"
INSERT
INTO Animal1
VALUES (
    :bind5,
    :bind8
    );
INSERT
INTO FeatureAnimal1
VALUES (
    :bind1,
    :bind6,
    :bind5,
    :bind3,
    :bind2,
    :bind4,
    :bind7
    );

-- UPDATE
-- "Select an AnimalID whose attributes are to be updated"
SELECT *
FROM FeatureAnimal1
WHERE animalID =  ($animalID);
-- "Then through multiple text boxes user can update the attributes they wish to be updated"
UPDATE FeatureAnimal1
SET Age=:bind2, Sex=:bind3, HealthStatus=:bind4, Species=:bind5, HabitatID=:bind6, OriginLocation=:bind7
    WHERE AnimalID=:bind1;


-- COUNT
-- "Counts number of tuples in FeatureAnimal Table"
SELECT Count(*) FROM FeatureAnimal1;


-- VIEW
-- "Returns all the attributes in Animal1 and FeatureAnimal1"
SELECT *
FROM Animal1, FeatureAnimal1
```

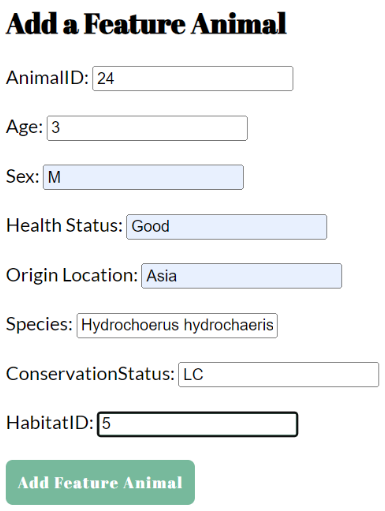
```
WHERE Animal1.species = FeatureAnimal1.species;


-- DELETE
-- "Deletes all the information from Habitat2 and Habitat1 (due to cascade) with specified HabitatID"
DELETE
FROM Habitat2
WHERE habitatID =($habitatID);

-- SELECTION
-- "Based on the table selected this query would find the value from Attribute specified in fromParam which satisfies the condition in WHERE clause "
SELECT ($_GET['selectParam']) FROM ($_GET['fromParam']) WHERE
($_GET['whereParam1']);


-- PROJECTION
-- "Returns values from the specified attributes in specified table"
SELECT ($_GET['selectAtt'])
FROM ($_GET['fromTab']);

-- JOIN
-- "Joins ResidentAnimal1, Habitat2 and Exhibits based on a condition in whereParam"
SELECT *
FROM ResidentAnimal1 r, Habitat2 h, Exhibits e
WHERE r.habitatID=h.habitatID
    AND h.exhibitID=e.exhibitID
    AND ($_GET['whereParam']);


-- HAVING
-- Returns vName and foodType Having higher rating than Average Rating of every FoodVendors
SELECT fv1.vName, fv1.foodType
FROM FoodVendor1 fv1
WHERE fv1.vName
    IN (SELECT fv2.vName
        FROM FoodVendor2 fv2
        GROUP BY fv2.vName
        HAVING AVG(fv2.rating) > (SELECT AVG(rating)
                                  FROM FoodVendor2));
```

```
-- Aggregated GROUP BY
-- Returns Average Vistor Capacity and Habitat type where popularityRating is higher than 6
SELECT h2.habitatType, AVG(e.visitorCapacity) as averageCapacity
FROM Habitat2 h2, Exhibits e
WHERE e.popularityRating > 6
    AND h2.exhibitID = e.exhibitID
    GROUP BY h2.habitatType;



-- Nested Aggregated GROUP BY
-- Returns Average Age of FeatureAnimal1 where Popularity Rating is higher than Average in their species
SELECT a.species, AVG(a.age) AS averageAge,e.popularityRating
FROM Exhibits e, FeatureAnimal1 a, Habitat2 h
WHERE h.exhibitID = e.exhibitID
    AND a.habitatID = h.habitatID
    GROUP BY a.species,e.popularityRating
    HAVING  e.popularityRating >= AVG(e.popularityRating);

-- DIVISION
-- Returns Species of ResidentAnimal1 which are presented in Exhibits i.e., associated with them.
SELECT species
FROM ResidentAnimal1
WHERE NOT EXISTS (SELECT habitatID
                    FROM ResidentAnimal1 a1
                    WHERE NOT EXISTS (SELECT h.habitatID
                                        FROM Habitat2 h, Exhibits e
                                        WHERE h.exhibitID = e.exhibitID
                                            AND a1.habitatID = h.habitatID));
```

## Screenshots

INSERT - inserting a feature animal (Tables: Animal1 and FeatureAnimal1)

Input:



### Add a Feature Animal

AnimalID: 24

Age: 3

Sex: M

Health Status: Good

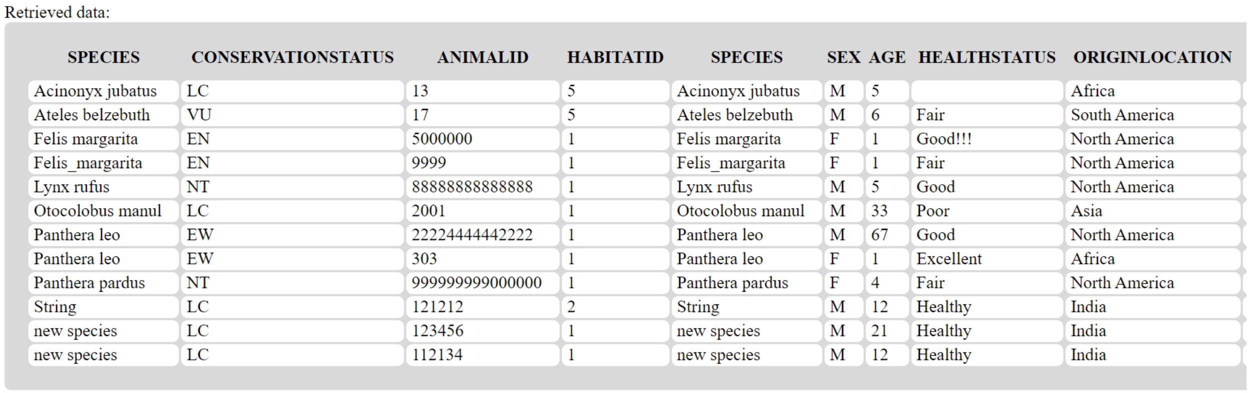
Origin Location: Asia

Species: Hydrochoerus hydrochaeris

ConservationStatus: LC

HabitatID: 5

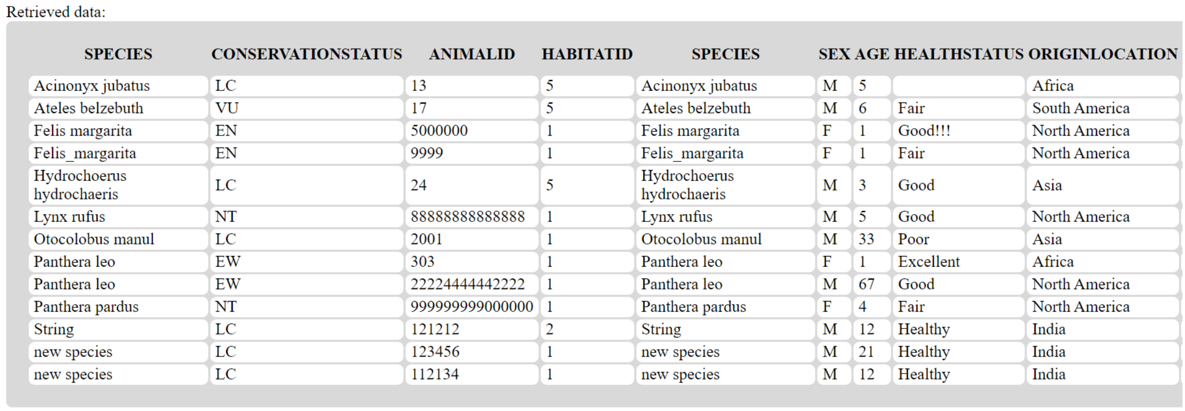
Before



Retrieved data:

| SPECIES | CONSERVATIONSTATUS | ANIMALID | HABITATID | SPECIES | SEX | AGE | HEALTHSTATUS | ORIGINLOCATION |
|---|---|---|---|---|---|---|---|---|
| Acinonyx jubatus | LC | 13 | 5 | Acinonyx jubatus | M | 5 |  | Africa |
| Ateles belzebuth | VU | 17 | 5 | Ateles belzebuth | M | 6 | Fair | South America |
| Felis margarita | EN | 5000000 | 1 | Felis margarita | F | 1 | Good!!! | North America |
| Felis_margarita | EN | 9999 | 1 | Felis_margarita | F | 1 | Fair | North America |
| Lynx rufus | NT | 88888888888888 | 1 | Lynx rufus | M | 5 | Good | North America |
| Otocolobus manul | LC | 2001 | 1 | Otocolobus manul | M | 33 | Poor | Asia |
| Panthera leo | EW | 22224444442222 | 1 | Panthera leo | M | 67 | Good | North America |
| Panthera leo | EW | 303 | 1 | Panthera leo | F | 1 | Excellent | Africa |
| Panthera pardus | NT | 999999999000000 | 1 | Panthera pardus | F | 4 | Fair | North America |
| String | LC | 121212 | 2 | String | M | 12 | Healthy | India |
| new species | LC | 123456 | 1 | new species | M | 21 | Healthy | India |
| new species | LC | 112134 | 1 | new species | M | 12 | Healthy | India |

After

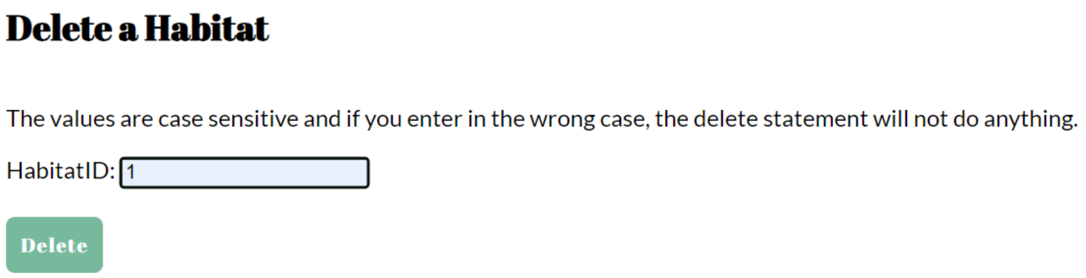


Retrieved data:

| SPECIES | CONSERVATIONSTATUS | ANIMALID | HABITATID | SPECIES | SEX | AGE | HEALTHSTATUS | ORIGINLOCATION |
|---|---|---|---|---|---|---|---|---|
| Acinonyx jubatus | LC | 13 | 5 | Acinonyx jubatus | M | 5 |  | Africa |
| Ateles belzebuth | VU | 17 | 5 | Ateles belzebuth | M | 6 | Fair | South America |
| Felis margarita | EN | 5000000 | 1 | Felis margarita | F | 1 | Good!!! | North America |
| Felis_margarita | EN | 9999 | 1 | Felis_margarita | F | 1 | Fair | North America |
| Hydrochoerus hydrochaeris | LC | 24 | 5 | Hydrochoerus hydrochaeris | M | 3 | Good | Asia |
| Lynx rufus | NT | 88888888888888 | 1 | Lynx rufus | M | 5 | Good | North America |
| Otocolobus manul | LC | 2001 | 1 | Otocolobus manul | M | 33 | Poor | Asia |
| Panthera leo | EW | 303 | 1 | Panthera leo | F | 1 | Excellent | Africa |
| Panthera leo | EW | 22224444442222 | 1 | Panthera leo | M | 67 | Good | North America |
| Panthera pardus | NT | 999999999000000 | 1 | Panthera pardus | F | 4 | Fair | North America |
| String | LC | 121212 | 2 | String | M | 12 | Healthy | India |
| new species | LC | 123456 | 1 | new species | M | 21 | Healthy | India |
| new species | LC | 112134 | 1 | new species | M | 12 | Healthy | India |

## DELETE

Input:

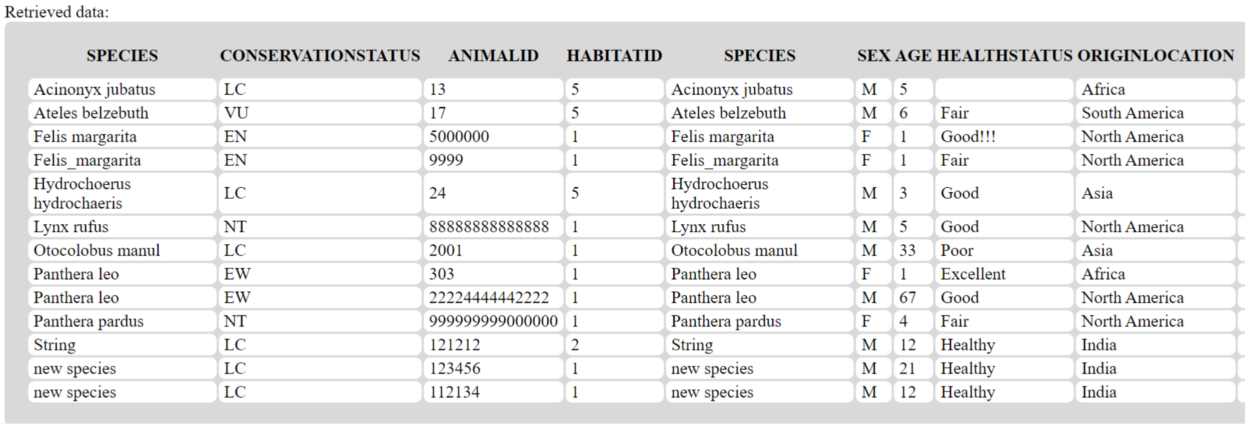


# Delete a Habitat

The values are case sensitive and if you enter in the wrong case, the delete statement will not do anything.

HabitatID: 1

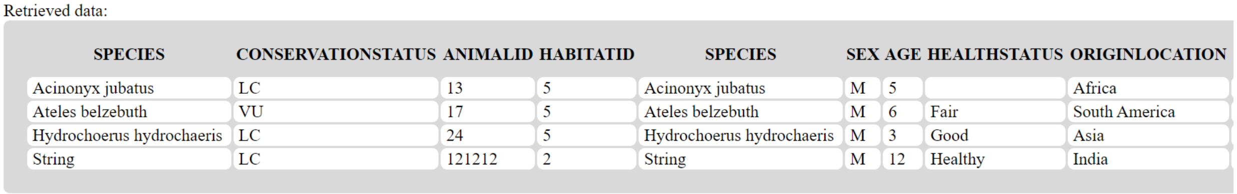
Before:



Retrieved data:

| SPECIES | CONSERVATIONSTATUS | ANIMALID | HABITATID | SPECIES | SEX | AGE | HEALTHSTATUS | ORIGINLOCATION |
|---|---|---|---|---|---|---|---|---|
| Acinonyx jubatus | LC | 13 | 5 | Acinonyx jubatus | M | 5 | | Africa |
| Ateles belzebuth | VU | 17 | 5 | Ateles belzebuth | M | 6 | Fair | South America |
| Felis margarita | EN | 5000000 | 1 | Felis margarita | F | 1 | Good!!! | North America |
| Felis_margarita | EN | 9999 | 1 | Felis_margarita | F | 1 | Fair | North America |
| Hydrochoerus hydrochaeris | LC | 24 | 5 | Hydrochoerus hydrochaeris | M | 3 | Good | Asia |
| Lynx rufus | NT | 88888888888888 | 1 | Lynx rufus | M | 5 | Good | North America |
| Otocolobus manul | LC | 2001 | 1 | Otocolobus manul | M | 33 | Poor | Asia |
| Panthera leo | EW | 303 | 1 | Panthera leo | F | 1 | Excellent | Africa |
| Panthera leo | EW | 22224444442222 | 1 | Panthera leo | M | 67 | Good | North America |
| Panthera pardus | NT | 999999999000000 | 1 | Panthera pardus | F | 4 | Fair | North America |
| String | LC | 121212 | 2 | String | M | 12 | Healthy | India |
| new species | LC | 123456 | 1 | new species | M | 21 | Healthy | India |
| new species | LC | 112134 | 1 | new species | M | 12 | Healthy | India |

Result:

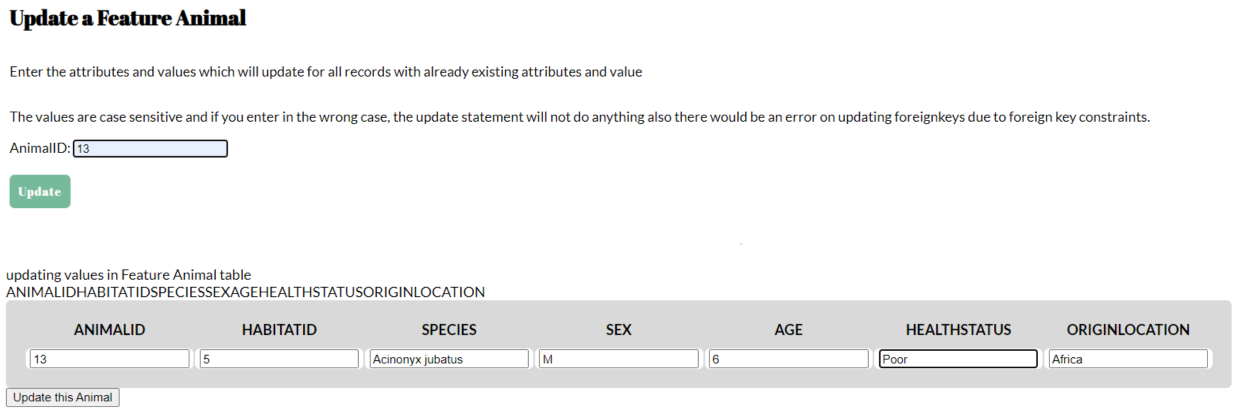


Retrieved data:

| SPECIES | CONSERVATIONSTATUS | ANIMALID | HABITATID | SPECIES | SEX | AGE | HEALTHSTATUS | ORIGINLOCATION |
|---|---|---|---|---|---|---|---|---|
| Acinonyx jubatus | LC | 13 | 5 | Acinonyx jubatus | M | 5 | | Africa |
| Ateles belzebuth | VU | 17 | 5 | Ateles belzebuth | M | 6 | Fair | South America |
| Hydrochoerus hydrochaeris | LC | 24 | 5 | Hydrochoerus hydrochaeris | M | 3 | Good | Asia |
| String | LC | 121212 | 2 | String | M | 12 | Healthy | India |

# UPDATE

## Input:



### Update a Feature Animal

Enter the attributes and values which will update for all records with already existing attributes and value

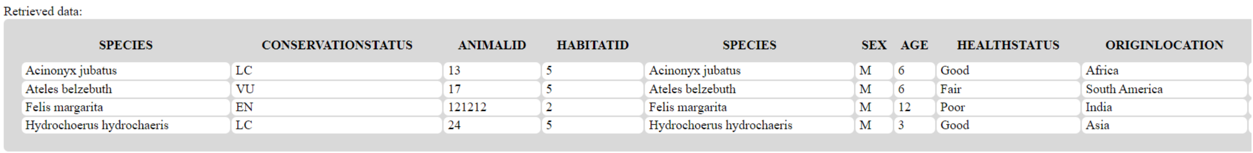
The values are case sensitive and if you enter in the wrong case, the update statement will not do anything also there would be an error on updating foreignkeys due to foreign key constraints.

AnimalID: 13

updating values in Feature Animal table
ANIMALIDHABITATIDSPECIESSEXAGEHEALTHSTATUSORIGINLOCATION

| ANIMALID | HABITATID | SPECIES | SEX | AGE | HEALTHSTATUS | ORIGINLOCATION |
|---|---|---|---|---|---|---|
| 13 | 5 | Acinonyx jubatus | M | 6 | Poor | Africa |

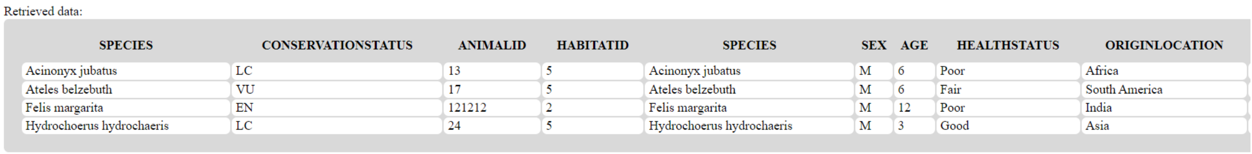
## Before:



Retrieved data:

| SPECIES | CONSERVATIONSTATUS | ANIMALID | HABITATID | SPECIES | SEX | AGE | HEALTHSTATUS | ORIGINLOCATION |
|---|---|---|---|---|---|---|---|---|
| Acinonyx jubatus | LC | 13 | 5 | Acinonyx jubatus | M | 6 | Good | Africa |
| Ateles belzebuth | VU | 17 | 5 | Ateles belzebuth | M | 6 | Fair | South America |
| Felis margarita | EN | 121212 | 2 | Felis margarita | M | 12 | Poor | India |
| Hydrochoerus hydrochaeris | LC | 24 | 5 | Hydrochoerus hydrochaeris | M | 3 | Good | Asia |

## Result

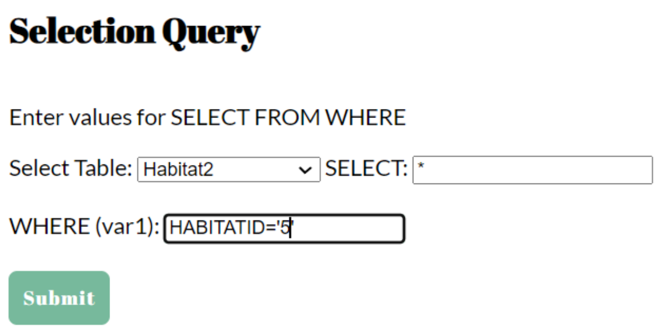


Retrieved data:

| SPECIES | CONSERVATIONSTATUS | ANIMALID | HABITATID | SPECIES | SEX | AGE | HEALTHSTATUS | ORIGINLOCATION |
|---|---|---|---|---|---|---|---|---|
| Acinonyx jubatus | LC | 13 | 5 | Acinonyx jubatus | M | 6 | Poor | Africa |
| Ateles belzebuth | VU | 17 | 5 | Ateles belzebuth | M | 6 | Fair | South America |
| Felis margarita | EN | 121212 | 2 | Felis margarita | M | 12 | Poor | India |
| Hydrochoerus hydrochaeris | LC | 24 | 5 | Hydrochoerus hydrochaeris | M | 3 | Good | Asia |

## SELECTION

Input:



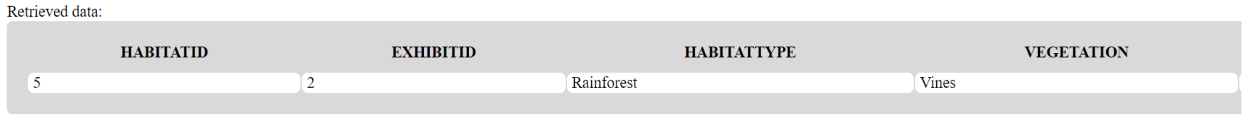
### Selection Query

Enter values for SELECT FROM WHERE

Select Table: Habitat2 SELECT: *

WHERE (var1): HABITATID='5'

Result:

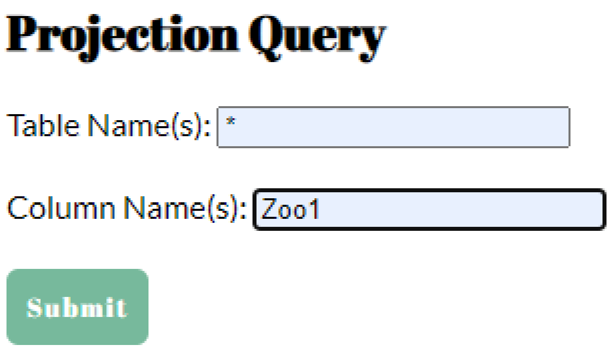


Retrieved data:

| HABITATID | EXHIBITID | HABITATTYPE | VEGETATION |
|---|---|---|---|
| 5 | 2 | Rainforest | Vines |

**PROJECTION**

Input:

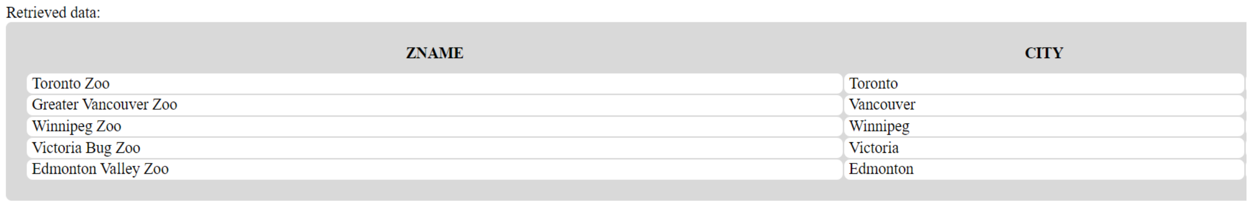


Projection Query

Table Name(s): *

Column Name(s): Zoo1

Result:

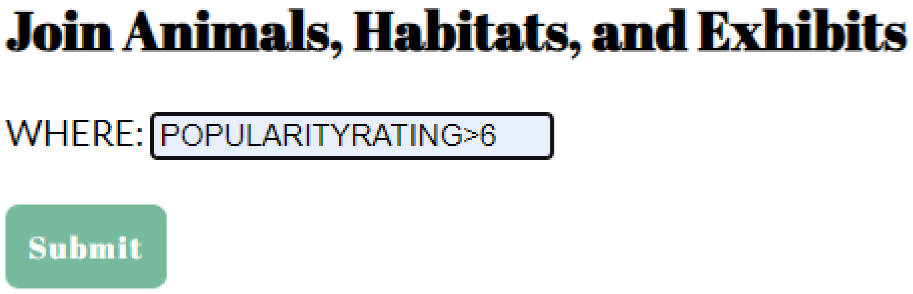


Retrieved data:

| ZNAME | CITY |
|---|---|
| Toronto Zoo | Toronto |
| Greater Vancouver Zoo | Vancouver |
| Winnipeg Zoo | Winnipeg |
| Victoria Bug Zoo | Victoria |
| Edmonton Valley Zoo | Edmonton |

**JOIN**

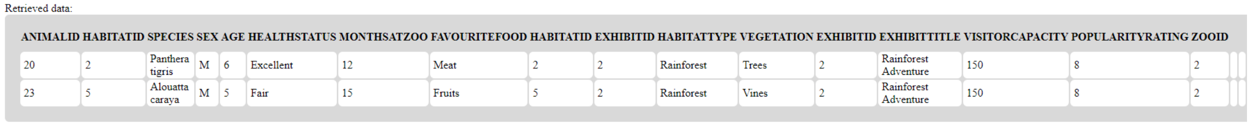
Input:



# Join Animals, Habitats, and Exhibits

WHERE: POPULARITYRATING>6

Result:

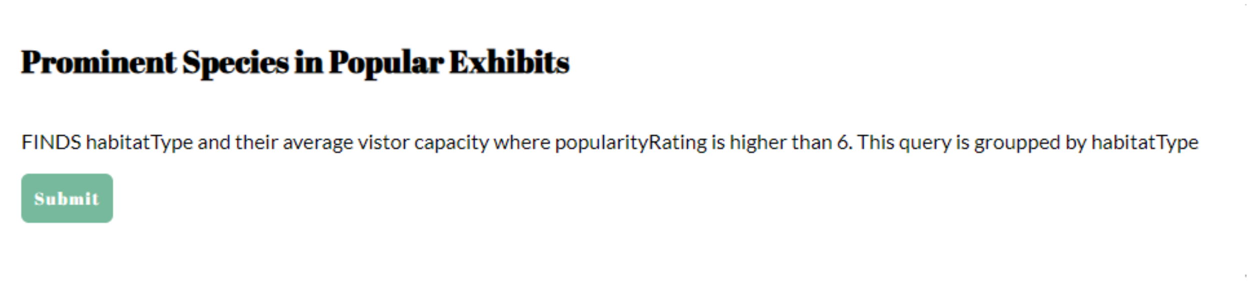


Retrieved data:

| ANIMALID | HABITATID | SPECIES | SEX | AGE | HEALTHSTATUS | MONTHSATZOO | FAVOURITEFOOD | HABITATID | EXHIBITID | HABITATTYPE | VEGETATION | EXHIBITID | EXHIBITTITLE | VISITORCAPACITY | POPULARITYRATING | ZOOID | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 20 | 2 | Panthera tigris | M | 6 | Excellent | 12 | Meat | 2 | 2 | Rainforest | Trees | 2 | Rainforest Adventure | 150 | 8 | 2 | | |
| 23 | 5 | Alouatta caraya | M | 5 | Fair | 15 | Fruits | 5 | 2 | Rainforest | Vines | 2 | Rainforest Adventure | 150 | 8 | 2 | | |

## AGGREGATION WITH GROUP BY

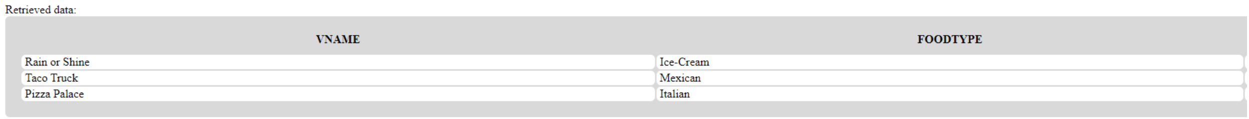
Input:



Prominent Species in Popular Exhibits

FINDS habitatType and their average vistor capacity where popularityRating is higher than 6. This query is groupped by habitatType

Result:

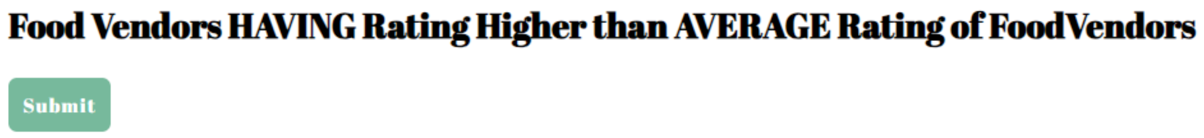


Retrieved data:

| VNAME | FOODTYPE |
|---|---|
| Rain or Shine | Ice-Cream |
| Taco Truck | Mexican |
| Pizza Palace | Italian |

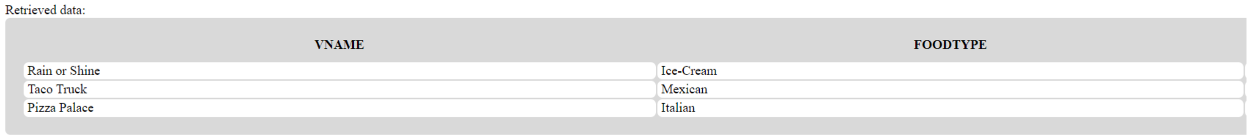
## AGGREGATION WITH HAVING

Input:



Food Vendors HAVING Rating Higher than AVERAGE Rating of FoodVendors

Result:

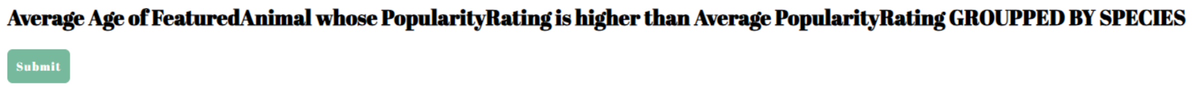


Retrieved data:

| VNAME | FOODTYPE |
|---|---|
| Rain or Shine | Ice-Cream |
| Taco Truck | Mexican |
| Pizza Palace | Italian |

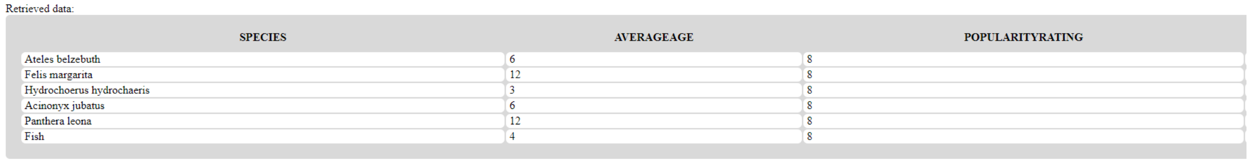
## AGGREGATION WITH NESTED GROUP BY

Input:



**Average Age of FeaturedAnimal whose PopularityRating is higher than Average PopularityRating GROUPPED BY SPECIES**

Result:

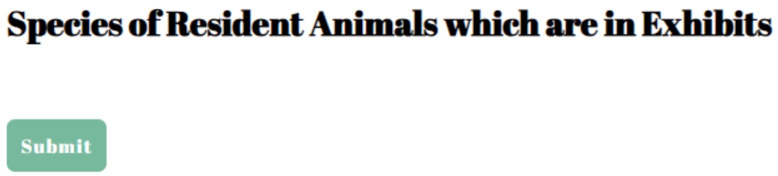


Retrieved data:

| SPECIES | AVERAGEAGE | POPULARITYRATING |
|---|---|---|
| Ateles belzebuth | 6 | 8 |
| Felis margarita | 12 | 8 |
| Hydrochoerus hydrochaeris | 3 | 8 |
| Acinonyx jubatus | 6 | 8 |
| Panthera leona | 12 | 8 |
| Fish | 4 | 8 |

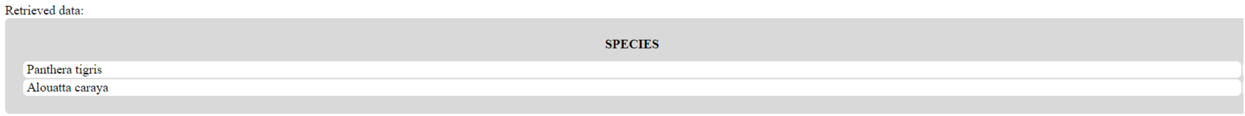
**Division**

Input:



Species of Resident Animals which are in Exhibits

Result:



Retrieved data:

| SPECIES |
|---|
| Panthera tigris |
| Alouatta caraya |

4.Include any extra information into a README file

See Github Repo for README.txt file.