

CPSC 304 Project Cover Page

Milestone #: 4

Date: December 01, 2023

Group Number: 65

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Sharon Marfatia	79155529	c8c4d	ssmubc@students.cs.ubc.ca
Chris Jung	14231609	b4i0j	eunhocj@students.cs.ubc.ca
Ayan Qadir	90759622	z7e0k	aqadir01@students.cs.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

CPSC 304 Project Milestone 4: Aquarium Database

Project Description

A brief (~2-3 sentences) summary of your project. Many of your TAs are managing multiple projects so this will help them remember details about your project.

Our database is designed to comprehensively model the many aspects of an aquarium exhibition, capturing both its internal and external components. Our database models core entities of an aquarium which are water tanks, devices within the tanks/maintenance equipment, marine animals, aquatic plants, the supply chain of curated animal diet, and the individuals responsible for operations and maintenance. Our project aims to address and represent the intricacies of the aquarium domain, and as an example of a real-life situation, the database can be applied to efficiently manage large-scale.

A short description of the final project, and what it accomplished.

Our final project allows users to interact with a user-friendly GUI application that enables the query operations listed on the criteria. It immediately updates the Oracle Database when changes are made by the user on the GUI application. With a combination of various panels, buttons and selection options for various operations our final project is a robust implementation of an Aquarium Management system that holds the potential for real-world applications in professional settings.

A brief (~2-3 sentences) description of any changes you have made to your schema from previous milestones.

There was one minor change made in one of the relationships from our previous design. The relation MAINTAINS has a total participation constraint rather than partial. In our database project, each time a water tank or aquarist is inserted, they have to be assigned to an aquarist or water tank. This is visible in our INSERT methods for each relation.

Oracle Database does not support ENUM so we modified ENUM statements to instead use CONSTRAINT CHECK and IN operations. Here is an example:

```
CREATE TABLE Exhibit (
    id INTEGER PRIMARY KEY,
    exhibit_name VARCHAR(255) NOT NULL,
    exhibit_status VARCHAR(255) NOT NULL,
    CONSTRAINT status_check CHECK (exhibit_status IN ('OPEN', 'CLOSED', 'UNDER MAINTENANCE', 'COMING SOON'))
);
COMMIT;

INSERT INTO Exhibit VALUES (17, 'Octopus Exhibit', 'OPEN');
COMMIT;
INSERT INTO Exhibit VALUES (18, 'Shark Exhibit', 'OPEN');
COMMIT;
INSERT INTO Exhibit VALUES (19, 'Dolphin Exhibit', 'OPEN');
COMMIT;
INSERT INTO Exhibit VALUES (20, 'Turtle Exhibit', 'OPEN');
COMMIT;
INSERT INTO Exhibit VALUES (21, 'Squid Exhibit', 'OPEN');
COMMIT;
```

Replaced names for attributes that were reserved words in Oracle. Here is an example for Equipment where the original attribute name was “function” and now is “equipment_function”.

```
CREATE TABLE Equipment (
    item_id INTEGER PRIMARY KEY,
    equipment_function VARCHAR(255) NOT NULL,
    weight DECIMAL(10,2) NOT NULL,
    equipment_size VARCHAR(255) NOT NULL,
    date_installed DATE NOT NULL,
    FOREIGN KEY (item_id) REFERENCES ItemQuantity(id) ON DELETE CASCADE
);
```

Repository Link: https://github.students.cs.ubc.ca/CPSC304-2023W-T1/project_b4i0j_c8c4d_z7e0k.git

SQL Script:

Our SQL Script, “Create_Tables.sql” is provided on the project repository on GitHub.

Here is a citation for information we studied while working on the script:

<https://stackoverflow.com/questions/58726718/set-variable-value>

Queries: INSERT Operation

The relation we will be using to demonstrate the INSERT operation relationship between the superclass: STAFF and the subclasses: AQUARIST, CUSTODIAN and VETERINARIAN

For this specific demonstration, we will be using STAFF and AQUARIST to demonstrate this behavior.

Database State of STAFF and AQUARIST (before Query)

STAFF

SALARY	ID	STAFF_NAME	DATEHIRED
1000.5	100	Sam	2023-10-15
1000.5	101	Anna	2023-10-15
145000.8	102	Kevin	2023-10-16
145000.8	103	John	2023-10-16
145000.8	104	Mohammed	2023-10-17
145000.8	105	James	2023-10-17
145000.8	106	Wataru	2023-10-18
100000.8	107	Michael	2022-10-05
1000.5	108	Kim	2022-10-30
1000.5	109	Danny	2022-09-15
100000.8	110	Rachel	2022-09-16
100000.8	111	Baam	2021-08-16
100000.8	112	Megumi	2021-07-16
100000.8	113	Oshimhen	2021-04-16
1000.5	114	Messi	2021-04-16

AQUARIST

	ID	DIVING_LEVEL	WATER_TANK_ID
1	107	100.00	1
2	110	100.00	2
3	111	100.00	3
4	112	100.00	4
5	113	100.00	5

Query performed for INSERT operation: INSERT INTO STAFF WHERE id = 115;

Code reference: Please refer to line 2002 insertStaff(id, salary, staffName, dateHired) in AquariumManagementDB.java within the src directory

Staff

ID :

115

Salary :

100000.00

Staff Name :

Kabir

Date Hired :

2023-12-01

Search

Add


Update

Delete

GUI Component of Operation (INSERT)

Database State of STAFF and AQUARIST (after Query)

Insert Data

 **Staff(ID: 115) was inserted successfully**

OK

Aquarium Manager

SALARY	ID	STAFF_NAME	DATEHIRED
100000	115	Kabir	2023-12-01
1000.5	100	Sam	2023-10-15
1000.5	101	Anna	2023-10-15
145000.8	102	Kevin	2023-10-16
145000.8	103	John	2023-10-16
145000.8	104	Mohammed	2023-10-17
145000.8	105	James	2023-10-17
145000.8	106	Wataru	2023-10-18
100000.8	107	Michael	2022-10-05
1000.5	108	Kim	2022-10-30
1000.5	109	Danny	2022-09-15
100000.8	110	Rachel	2022-09-16
100000.8	111	Baam	2021-08-16
100000.8	112	Megumi	2021-07-16
100000.8	113	Oshimhen	2021-04-16
1000.5	114	Messi	2021-04-16

Now we INSERT this same new employee, into the AQUARIST table.

Aquarist


ID : 115

Diving Level : 210

Water Tank ID : 4

Search Add Update Delete

Insert Data

 **Aquarist(ID: 115) was inserted successfully**

OK

The new Employee (ID: 115) is now also in the AQUARIST table:

	ID	DIVING_LEVEL	WATER_TANK_ID
1	115	210.00	4
2	107	100.00	1
3	110	100.00	2
4	111	100.00	3
5	112	100.00	4
6	113	100.00	5

Queries: DELETE Operation

The relation we will be using to demonstrate the DELETE ON CASCADE functionality in Oracle is the ISA relationship between the superclass: STAFF and the subclasses: AQUARIST, CUSTODIAN and VETERINARIAN

For this specific demonstration, we will be using STAFF and CUSTODIAN to demonstrate this behavior.

Database State of STAFF and CUSTODIAN (before Query)

STAFF

SALARY	ID	STAFF_NAME	DATEHIRED
1000.5	100	Sam	2023-10-15
1000.5	101	Anna	2023-10-15
145000.8	102	Kevin	2023-10-16
145000.8	103	John	2023-10-16
145000.8	104	Mohammed	2023-10-17
145000.8	105	James	2023-10-17
145000.8	106	Wataru	2023-10-18
100000.8	107	Michael	2022-10-05
1000.5	108	Kim	2022-10-30
1000.5	109	Danny	2022-09-15
100000.8	110	Rachel	2022-09-16
100000.8	111	Baam	2021-08-16
100000.8	112	Megumi	2021-07-16
100000.8	113	Oshimhen	2021-04-16
1000.5	114	Messi	2021-04-16

CUSTODIAN

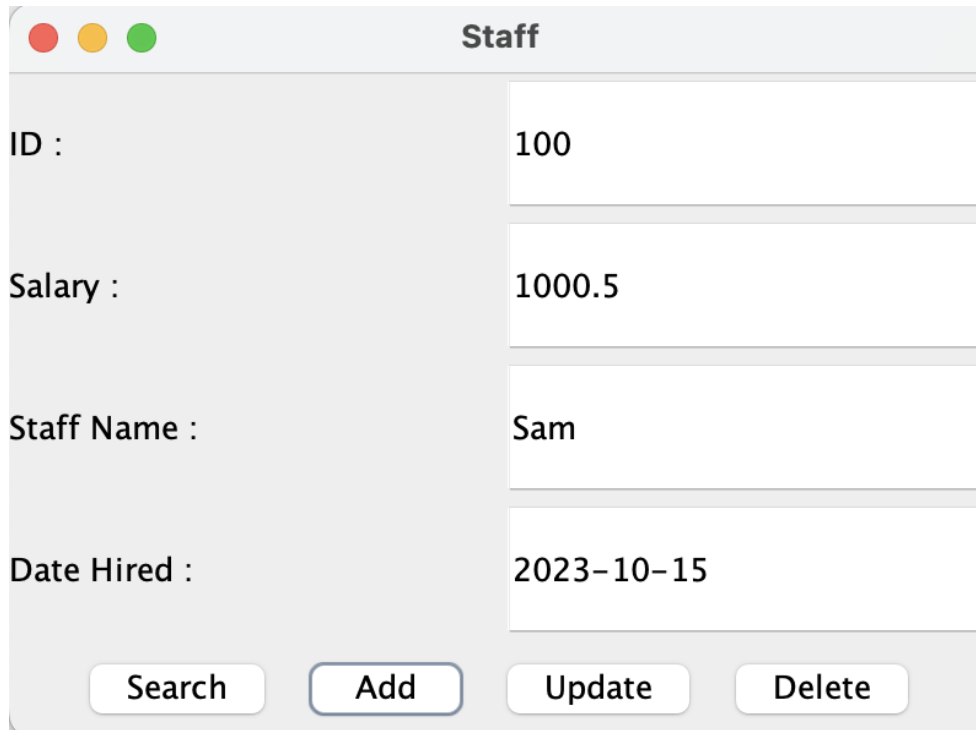
Query performed for DELETE operation: DELETE FROM STAFF WHERE id = 100;

EXHIBIT_ID	ID
17	100
18	101
19	108
20	109
21	114

Code reference: Please refer to the function deleteStaff(int id) at line 2237 in AquariumManagementDB.java within the src directory

GUI Component of Operation (DELETE)

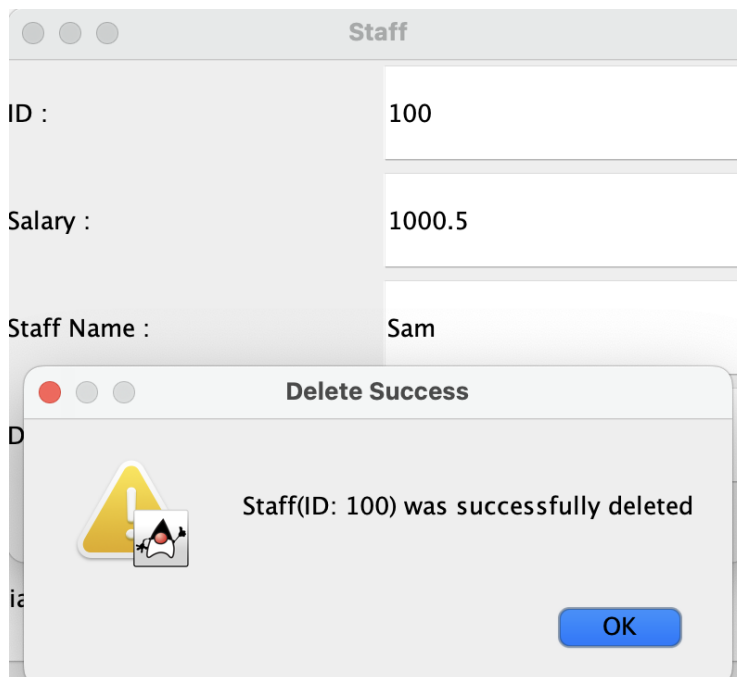
We will type out the information for the Staff object we want to delete in the following form by clicking Manage Staff



A macOS-style window titled "Staff" with a light gray background and three colored window control buttons (red, yellow, green) in the top-left corner. The form contains four labeled text input fields on the left and their corresponding values in text boxes on the right. The labels are "ID :", "Salary :", "Staff Name :", and "Date Hired :". The values are "100", "1000.5", "Sam", and "2023-10-15" respectively. At the bottom of the window, there are four rounded rectangular buttons: "Search", "Add", "Update", and "Delete". The "Add" button is highlighted with a blue border.

ID :	100
Salary :	1000.5
Staff Name :	Sam
Date Hired :	2023-10-15

Buttons: Search, Add, Update, Delete



University of British Columbia, Vancouver

Department of Computer Science

The tuple was successfully delete from Staff

Database State of STAFF and CUSTODIAN (after Query)

SALARY	ID	STAFF_NAME	DATEHIRED
1000.5	101	Anna	2023-10-15
145000.8	102	Kevin	2023-10-16
145000.8	103	John	2023-10-16
145000.8	104	Mohammed	2023-10-17
145000.8	105	James	2023-10-17
145000.8	106	Wataru	2023-10-18
100000.8	107	Michael	2022-10-05
1000.5	108	Kim	2022-10-30
1000.5	109	Danny	2022-09-15
100000.8	110	Rachel	2022-09-16
100000.8	111	Baam	2021-08-16
100000.8	112	Megumi	2021-07-16
100000.8	113	Oshimhen	2021-04-16
1000.5	114	Messi	2021-04-16

STAFF

EXHIBIT_ID	ID
18	101
19	108
20	109
21	114

CUSTODIAN

The tuple with id = 100 from STAFF and CUSTODIAN was deleted as observed in the above screenshots demonstrating the ON DELETE CASCADE behavior

Queries: UPDATE Operation

The relation we will be using to demonstrate the UPDATE operation relationship between the superclass: STAFF and the subclasses: AQUARIST, CUSTODIAN and VETERINARIAN

For this specific demonstration, we will be using STAFF and CUSTODIAN to demonstrate this behavior.

Database State of STAFF and CUSTODIAN (before Query)

STAFF

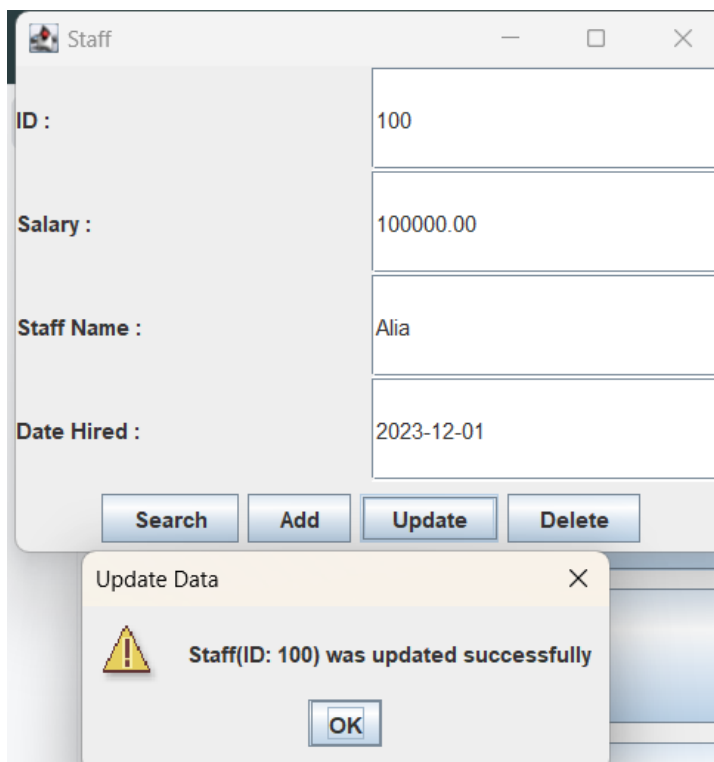
SALARY	ID	STAFF_NAME	DATEHIRED
1000.5	100	Sam	2023-10-15
1000.5	101	Anna	2023-10-15
145000.8	102	Kevin	2023-10-16
145000.8	103	John	2023-10-16
145000.8	104	Mohammed	2023-10-17
145000.8	105	James	2023-10-17
145000.8	106	Wataru	2023-10-18
100000.8	107	Michael	2022-10-05
1000.5	108	Kim	2022-10-30
1000.5	109	Danny	2022-09-15
100000.8	110	Rachel	2022-09-16
100000.8	111	Baam	2021-08-16
100000.8	112	Megumi	2021-07-16
100000.8	113	Oshimhen	2021-04-16
1000.5	114	Messi	2021-04-16

CUSTODIAN

ID	EXHIBIT_ID
100	17
101	18
108	19
109	20
114	21

Query performed for **UPDATE** operation: UPDATE STAFF WHERE id = 100;

Code reference: Please refer to line 2140 updateStaff(id, salary, staffName, dateHired) in AquariumManagementDB.java within the src directory



GUI Component of Operation (UPDATE)

Database State of STAFF and CUSTODIAN (after Query)

University of British Columbia, Vancouver
Department of Computer Science

SALARY	ID	STAFF_NAME	DATEHIRED
100000	100	Alia	2023-12-01
1000.5	101	Anna	2023-10-15
145000.8	102	Kevin	2023-10-16
145000.8	103	John	2023-10-16
145000.8	104	Mohammed	2023-10-17
145000.8	105	James	2023-10-17
145000.8	106	Wataru	2023-10-18
100000.8	107	Michael	2022-10-05
1000.5	108	Kim	2022-10-30
1000.5	109	Danny	2022-09-15
100000.8	110	Rachel	2022-09-16
100000.8	111	Baam	2021-08-16
100000.8	112	Megumi	2021-07-16
100000.8	113	Oshimhen	2021-04-16
1000.5	114	Messi	2021-04-16

Now we can UPDATE this same employee in the CUSTODIAN table.

Custodian

D :

100

Exhibit ID :

19


Search

Add

Update

Delete

Update Data



Custodian(ID: 100) was updated successfully

OK

The Employee (ID: 100) in the CUSTODIAN table:

ID	EXHIBIT_ID
100	19
101	18
108	19
109	20
114	21

Queries: Selection

The relation we will be using to demonstrate the PROJECTION functionality in Oracle is the relation WATERTANKLOGISTICS, WATERTANKPH and

AQUARIST_MAINTAINS_WATERTANK. This performs a double JOIN on these three relations on water_tank_id and temperature.

For this specific demonstration, we will passing this conditional statement: ID > 3 AND LIGHTINGLEVEL = 'Medium' OR PH > 6.5

Code reference: Please refer to the function selectWaterTank(JSONArray waterTankConditions) at line 1244 in AquariumManagementDB.java within the src directory

GUI demonstration of Selection

Aquarium Manager						
LIGHTINGLEVEL	TEMPERATURE	PH	EXHIBIT_ID	ID	AQUARIST_ID	WATER_TANK_LOGISTI...
MEDIUM	27.0	7.0	18	6	113	Tortoise Tank
MEDIUM	27.5	7.1	18	1	111	Shark Tank
MEDIUM	27.8	6.8	17	2	110	Octopus Tank
LOW	27.8	6.8	20	5	113	Turtle Tank
MEDIUM	28.0	6.7	19	3	111	Dolphin Tank
MEDIUM	27.9	6.7	21	4	112	Squid Tank

[Back to Home](#) [Show All Tank](#) [Advanced Search](#)

Under ListTanks, all the tuples from the WaterTank (WATERTANKLOGISTICS, WATERTANKPH and AQUARIST_MAINTAINS_WATERTANK) are displayed

Advanced Search

Name	Condition	Comparison	Value
ID	AND	>	3
Name	Not U...		
Volume(l)	Not U...		
Temperature(...	Not U...		
Lighting Level	AND	=	Medium
In Exhibit(ID)	Not U...		
Ph	OR	>	6.5
Aquarist(ID)	Not U...		

Submit

Close

The advanced search selection menu in the GUI allows the user to compose their WHERE clause. In our database design, ANDs are grouped before ORs.

Query performed for SELECTION operation: SELECT wl.ID, wl.WATER_TANK_LOGISTICS_NAME, wl.VOLUME, wl.TEMPERATURE, wp.PH, wl.LIGHTINGLEVEL, wl.EXHIBIT_ID, m.AQUARIST_ID FROM WATERTANKLOGISTICS wl JOIN WATERTANKPH wp ON wl.TEMPERATURE = wp.TEMPERATURE JOIN AQUARIST_MAINTAIN_WATERTANK m ON m.WATER_TANK_ID = wl.ID WHERE ID > 3 AND LIGHTINGLEVEL = 'MEDIUM' OR PH > 6.5;

Aquarium Manager							
ID	Name	Volume(l)	Temperature(°C)	Lighting Level	In Exhibit(ID)	Ph	Aquarist(ID)
1	Shark Tank	1000.45	27.5	MEDIUM	18	7.0	107
2	Octopus Tank	1500.45	27.8	MEDIUM	17	6.8	110
3	Dolphin Tank	1000.45	28.0	MEDIUM	19	6.7	111
4	Squid Tank	1000.45	27.9	MEDIUM	21	6.7	112
5	Turtle Tank	500.5	27.8	LOW	20	6.8	113
5	Turtle Tank	500.5	27.8	LOW	20	6.8	114
1	Shark Tank	1000.45	27.5	MEDIUM	18	7.0	115
1	Shark Tank	1000.45	27.5	MEDIUM	18	7.0	116
2	Octopus Tank	1500.45	27.8	MEDIUM	17	6.8	117
3	Dolphin Tank	1000.45	28.0	MEDIUM	19	6.7	118
1	Shark Tank	1000.45	27.5	MEDIUM	18	7.0	119
4	Squid Tank	1000.45	27.9	MEDIUM	21	6.7	120
4	Squid Tank	1000.45	27.9	MEDIUM	21	6.7	121

This query displays the result of the SELECT query that passes all these conditions

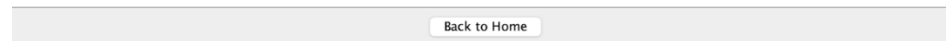
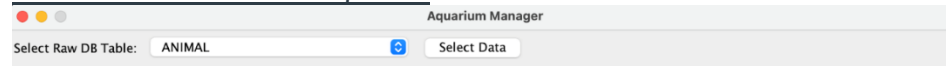
Queries: Projection

The relation we will be using to demonstrate the PROJECTION functionality in Oracle is the relation ANIMAL.

For this specific demonstration, we will be selecting the four attributes: ANIMAL_ID, NAME, SPECIES, WATER_TANK_ID

Code reference: Please refer to the function `getRawData(JSONObject relationObj)` at line 60 (this will change when we clean up the code) in `AquariumManagementDB.java` within the `src` directory

GUI demonstration of Projection



This is the GUI view before selecting any attributes from the relation ANIMAL.

The query that is being performed: `SELECT ID, NAME, SPECIES, WATER_TANK_ID FROM ANIMAL;`

We select the relevant fields from the checkbox menu

Select fields to display

ID	<input checked="" type="checkbox"/>
ANIMAL_NAME	<input checked="" type="checkbox"/>
SPECIES	<input checked="" type="checkbox"/>
AGE	<input type="checkbox"/>
LIVINGTEMP	<input type="checkbox"/>
WATER_TANK_ID	<input checked="" type="checkbox"/>
VETERINARIAN_ID	<input type="checkbox"/>

View Data **Cancel**

The GUI now displays the selected attributes of each tuple in the relation ANIMAL.

Aquarium Manager

Select Raw DB Table: **ANIMAL** **Select Data**

WATER_TANK_ID	SPECIES	ANIMAL_NAME	ID
5	Humphead wrasse	wrasse	36
5	Fin Whale	Phinius	37
5	Beluga Whale	Benjamin	38
2	Octopus vulgaris	Common Octopus	32
3	Orcinus orca	Orca	33
4	Vampyroteuthis infernalis	Vampire Squid	34
5	Aldabrachelys gigantea	Aldabra giant tortoise	35

Queries: Join

The relation we will be using to demonstrate the JOIN functionality in Oracle is the relation WATERTANKLOGISTICS, WATERTANKPH and AQUARIST_MAINTAINS_WATERTANK. This performs a double JOIN on these three relations on water_tank_id and temperature.

For this specific demonstration, we will passing this conditional statement: LIVING_TEMP > 27.5

Code reference: Please refer to the function selectWaterTank(JSONArray waterTankConditions) at line 1244 (this will change when we clean up the code) in AquariumManagementDB.java within the src directory

Aquarium Manager						
LIGHTINGLEVEL	TEMPERATURE	PH	EXHIBIT_ID	ID	AQUARIST_ID	WATER_TANK_LOGISTI...
MEDIUM	27.0	7.0	18	6	113	Tortoise Tank
MEDIUM	27.5	7.1	18	1	111	Shark Tank
MEDIUM	27.8	6.8	17	2	110	Octopus Tank
LOW	27.8	6.8	20	5	113	Turtle Tank
MEDIUM	28.0	6.7	19	3	111	Dolphin Tank
MEDIUM	27.9	6.7	21	4	112	Squid Tank

[Back to Home](#)[Show All Tank](#)[Advanced Search](#)

Query performed for JOIN operation: SELECT wl.ID, wl.WATER_TANK_LOGISTICS_NAME, wl.VOLUME, wl.TEMPERATURE, wp.PH, wl.LIGHTINGLEVEL, wl.EXHIBIT_ID, m.AQUARIST_ID FROM WATERTANKLOGISTICS wl JOIN WATERTANKPH wp ON wl.TEMPERATURE = wp.TEMPERATURE JOIN AQUARIST_MAINTAIN_WATERTANK m ON m.WATER_TANK_ID = wl.ID WHERE ID > 3 AND LIGHTINGLEVEL = 'MEDIUM' AND PH > 6.5;

Advanced Search

Name	Condition	Comparison	Value
ID	Not U...		
Name	Not U...		
Volume(l)	Not U...		
Temperature(...)	Not U...		
Lighting Level	Not U...		
In Exhibit(ID)	Not U...		
Ph	OR	>	6.7
Aquarist(ID)	Not U...		

Submit

Close

We are choosing pH > 6.7 as the condition for our WHERE clause. According to our backend, the OR or AND clause gets dropped if we only pass one condition.

ID	Name	Volume(l)	Temperature(°C)	Lighting Level	In Exhibit(ID)	Ph	Aquarist(ID)
1	Shark Tank	1000.45	27.5	MEDIUM	18	7.0	107
2	Octopus Tank	1500.45	27.8	MEDIUM	17	6.8	110
5	Turtle Tank	500.5	27.8	LOW	20	6.8	113
5	Turtle Tank	500.5	27.8	LOW	20	6.8	114
1	Shark Tank	1000.45	27.5	MEDIUM	18	7.0	115
1	Shark Tank	1000.45	27.5	MEDIUM	18	7.0	116
2	Octopus Tank	1500.45	27.8	MEDIUM	17	6.8	117
1	Shark Tank	1000.45	27.5	MEDIUM	18	7.0	119

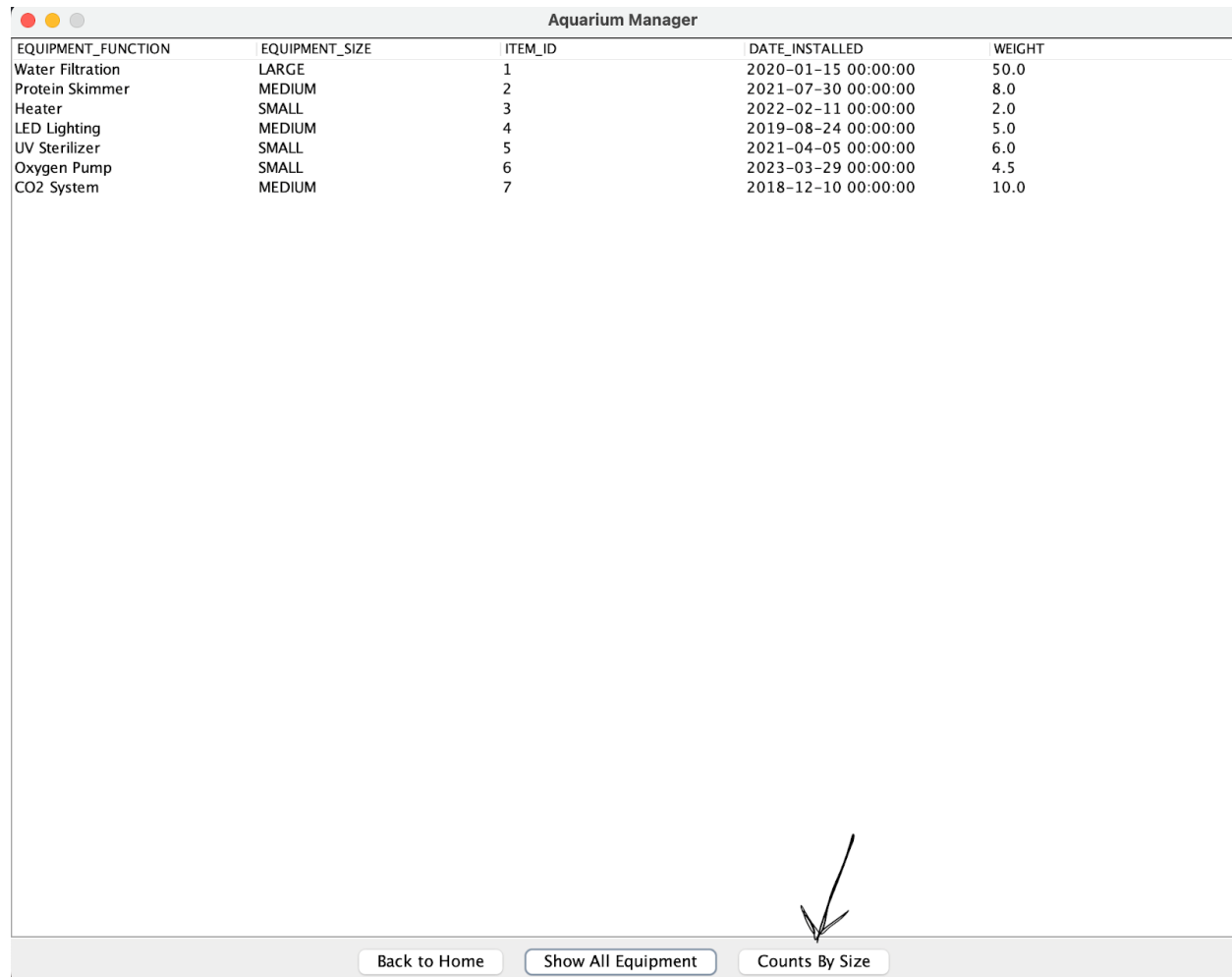
This is the result of the query after performing the JOIN query with the WHERE clause.

Queries: Aggregation with Group By

The relation we will be using to demonstrate the Aggregation with Group By functionality in Oracle is the relation EQUIPMENT.

For this specific demonstration, we will be counting equipment in each equipment size grouping by equipment size

Code reference: Please refer to the function `groupByEquipmentSize()` at line 930 in `AquariumManagementDB.java` within the `src` directory



EQUIPMENT_FUNCTION	EQUIPMENT_SIZE	ITEM_ID	DATE_INSTALLED	WEIGHT
Water Filtration	LARGE	1	2020-01-15 00:00:00	50.0
Protein Skimmer	MEDIUM	2	2021-07-30 00:00:00	8.0
Heater	SMALL	3	2022-02-11 00:00:00	2.0
LED Lighting	MEDIUM	4	2019-08-24 00:00:00	5.0
UV Sterilizer	SMALL	5	2021-04-05 00:00:00	6.0
Oxygen Pump	SMALL	6	2023-03-29 00:00:00	4.5
CO2 System	MEDIUM	7	2018-12-10 00:00:00	10.0

Back to Home Show All Equipment Counts By Size

We start off by clicking on List Equipment on the home panel. It shows all the Equipment tuples in our relation EQUIPMENT. We can process the query by clicking on Count By Sizes at the bottom panel

Query performed for Aggregation with GROUP BY: `SELECT EQUIPMENT_SIZE, COUNT(*)
FROM EQUIPMENT GROUP BY EQUIPMENT_SIZE;`

The result of the query shows the count of each equipment size in the relation EQUIPMENT.

Aquarium Manager	
EQUIPMENT_SIZE	EquipmentCount
LARGE	1
SMALL	3
MEDIUM	3

Queries: Aggregation with Having

The relation we will be using to demonstrate the HAVING functionality in Oracle is the relation GROWN_IN_PLANT.

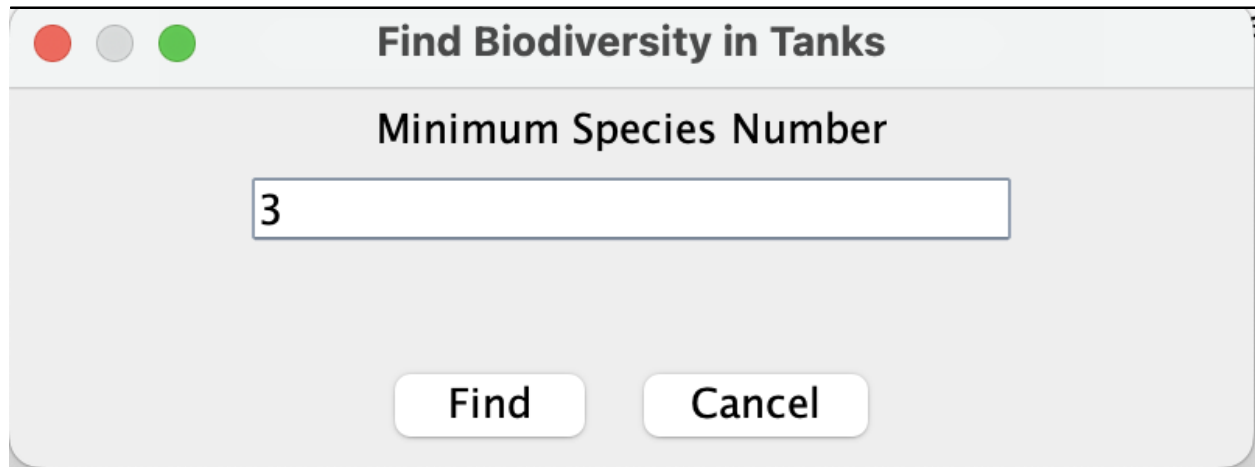
For this specific demonstration, we will passing this conditional statement: `min_species > 3`

Code reference: Please refer to the function `getPlantDiversityInTanksAboveThreshold(int speciesThreshold)` at line 2569 (this will change when we clean up the code) in `AquariumManagementDB.java` within the `src` directory

Aquarium Manager				
WATER_TANK_ID	LIVING_TEMP	LIVING_LIGHT	SPECIES	PLANT_ID
1	22.5	100.0	Water Lily	101
2	18.0	80.0	Seaweed	102
3	25.0	70.0	Mangrove	103
4	26.0	90.0	Coral	104
5	24.0	60.0	Anubias	105
3	24.5	70.0	Duckweed	106
1	25.0	66.0	Water lettuce	107
2	25.5	65.0	Watercress	108
2	25.3	67.0	Coontail	109
4	27.0	80.0	American Lotus	110
4	26.0	90.0	Chara	111
5	22.0	85.0	WaterThymes	112
1	22.5	60.0	Canna	113
2	26.5	65.0	Northern Blue Flag	114
2	27.3	67.0	Coontail	115
3	24.5	85.0	Mosquito ferns	116
3	26.0	73.0	Parrots leather	117
5	22.0	82.0	Bladderwot	118
1	21.0	64.0	Potamogeton	119
2	25.5	87.0	Water lily	120

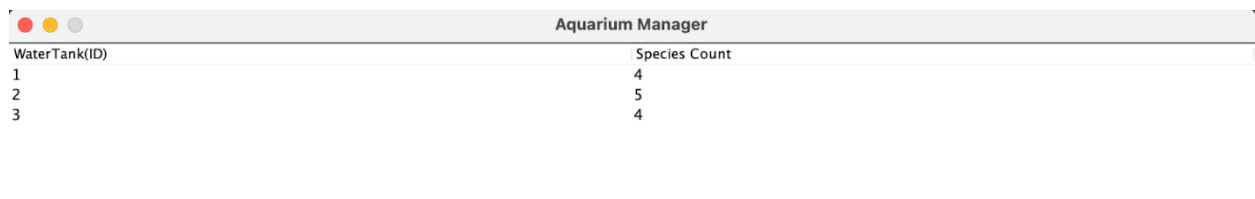
Back to Home
Show All Plants
Find Biodiversity in Tanks

We click on List Plants that shows all the tuples in the GROWN_IN_PLANT relation. We can process the query by clicking on Find Biodiversity in Tanks at the bottom panel.



We choose our minimum distinct species to be 3.

Query to be processed for HAVING: SELECT WATER_TANK_ID, COUNT(DISTINCT SPECIES) AS Species_Count FROM GROWN_IN_PLANT GROUP BY WATER_TANK_ID HAVING COUNT(DISTINCT SPECIES) > 3;



WaterTank(ID)	Species Count
1	4
2	5
3	4

This is the result of our query after processing the HAVING query.

Queries: Nested Aggregation with Group By

The relation we will be using to demonstrate the Nested Aggregation with Group By functionality in Oracle is the relation ANIMAL.

For this specific demonstration, we will be calculating the average age of animal species that are above a living temperature threshold

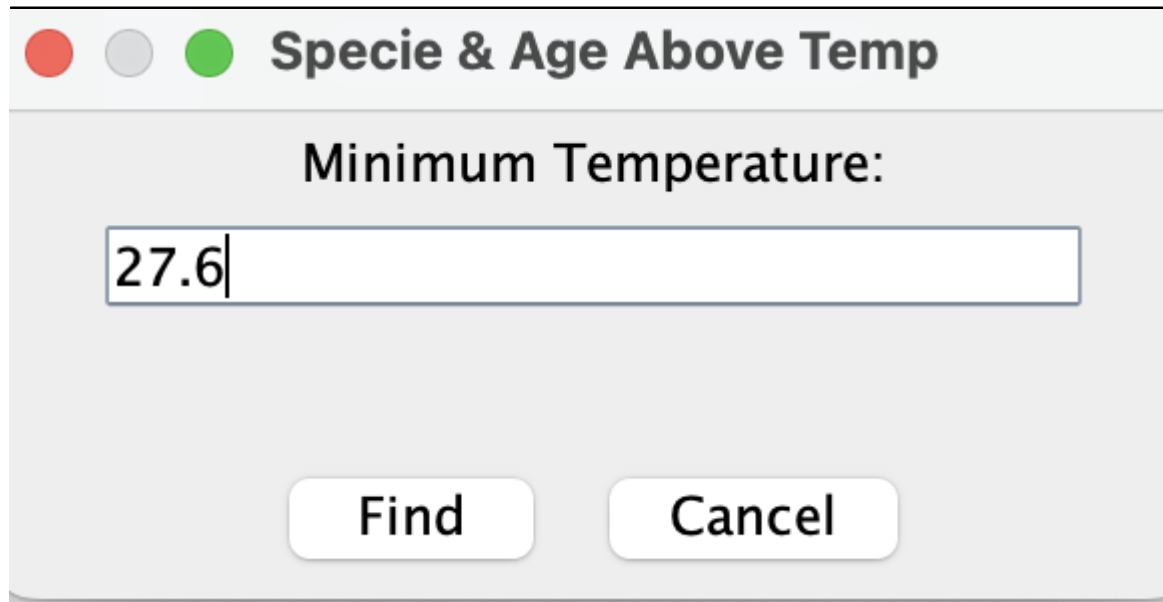
Code reference: Please refer to the function `groupByAnimalSpeciesAndAverageAgeAboveLivingTemp(double temperatureThreshold)` at line 1586 (this will change when we clean up the code) in `AquariumManagementDB.java` within the `src` directory

Aquarium Manager						
WATER_TANK_ID	VETERINARIAN_ID	SPECIES	ANIMAL_NAME	ID	LIVINGTEMP	AGE
1	102	Carcharodon carcharias	Great White Shark	31	27.5	6
2	102	Octopus vulgaris	Common Octopus	32	27.5	7
3	103	Orcinus orca	Orca	33	27.5	5
4	104	Vampyroteuthis infern...	Vampire Squid	34	27.5	3
5	105	Aldabrachelys gigantea	Aldabra giant tortoise	35	27.5	1
5	105	Humphead wrasse	wrasse	36	27.8	1
5	105	Fin Whale	Phinius	37	27.8	4
5	105	Beluga Whale	Benjamin	38	27.7	3
5	106	Beluga Whale	John	39	27.7	4
5	106	Beluga Whale	Alexander	40	27.6	8
5	106	Fin Whale	Benjamin	41	27.7	3
5	106	Fin Whale	Benjamin	42	27.7	3
5	106	Humphead wrasse	Sam	43	27.5	1
5	106	Humphead wrasse	Michael	44	27.5	1

Back to Home
Show All Animals
Find Expert Vet
Species & Age Above Temp

The GUI displays all the tuples in the Animal relation before running our query.

Query performed for Aggregation By Group By: "SELECT a.SPECIES, AVG(a.AGE) FROM ANIMAL a GROUP BY a.SPECIES HAVING AVG(a.AGE) > (SELECT AVG(a2.AGE) FROM ANIMAL a2 WHERE a2.LIVINGTEMP > 27.6);



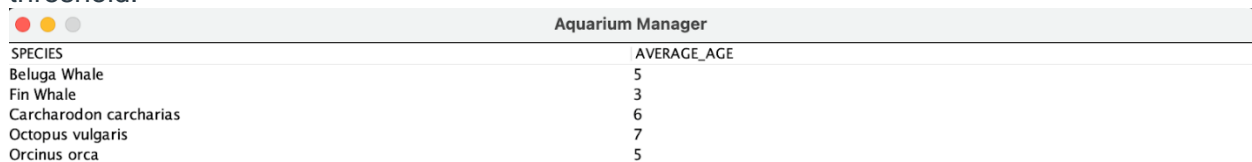
Specie & Age Above Temp

Minimum Temperature:

27.6

Find Cancel

After clicking the Species & Age Above Temp button, we can input a minimum temperature threshold.



SPECIES	AVERAGE_AGE
Beluga Whale	5
Fin Whale	3
Carcharodon carcharias	6
Octopus vulgaris	7
Orcinus orca	5

The result of the query is the average age of all animal species that are above the temperature threshold.

Queries: Division

The relation we will be using to demonstrate the DIVISION operation are ANIMAL and VETERINARIAN

Show Tank List (on GUI) we then click on the “Find Expert Vet”:

University of British Columbia, Vancouver
Department of Computer Science

Aquarium Manager						
WATER_TANK_ID	VETERINARIAN_ID	SPECIES	ANIMAL_NAME	ID	LIVINGTEMP	AGE
1	103	Carcharodon	Bull Shark	39	27.7	5
1	102	Carcharodon	Great White Shark	31	27.5	6
2	102	Octopus vulgaris	Common Octopus	32	27.5	7
3	103	Orcinus orca	Orca	33	27.5	5
4	104	Vampyroteuthis infernalis	Vampire Squid	34	27.5	3
5	105	Aldabrachelys gigantea	Aldabra giant tortoise	35	27.5	1
5	105	Humphead wrasse	wrasse	36	27.8	1
5	105	Fin Whale	Phinius	37	27.8	4
5	105	Beluga Whale	Benjamin	38	27.7	3

Back to HomeShow All AnimalsFind Expert VetSpecies & Age Above Temp

Query performed for DIVISION operation:

```
String sql = SELECT v.ID FROM VETERINARIAN v
WHERE NOT EXISTS
  SELECT a.ID
  FROM ANIMAL a
  WHERE a.SPECIES = 'Carcharodon'
  AND NOT EXISTS (
    SELECT *
    FROM ANIMAL a2 WHERE a2.VETERINARIAN_ID = v.ID AND a2.SPECIES =
'Carcharodon'));
```

Code reference: Please refer to line 2607 getAnimalExpertVets(String species) in AquariumManagementDB.java within the src directory

University of British Columbia, Vancouver
Department of Computer Science

The screenshot shows the 'Aquarium Manager' application window. At the top, there is a table with 7 columns: WATER_TANK_ID, VETERINARIAN_ID, SPECIES, ANIMAL_NAME, ID, LIVINGTEMP, and AGE. The table contains 10 rows of data. Below the table, there is a large grey rectangular area. In the center of this area is a 'Find Expert Vet' dialog box. The dialog box has a title bar with a close button. Inside, there is a label 'Species Name:' followed by a text input field containing the text 'Carcharodon'. Below the input field are two buttons: 'Find' and 'Cancel'. At the bottom of the application window, there is a horizontal bar with four buttons: 'Back to Home', 'Show All Animals', 'Find Expert Vet', and 'Species & Age Above Temp'.

WATER_TANK_ID	VETERINARIAN_ID	SPECIES	ANIMAL_NAME	ID	LIVINGTEMP	AGE
1	103	Carcharodon	Bull Shark	39	27.7	5
1	102	Carcharodon	Great White Shark	31	27.5	6
2	102	Octopus vulgaris	Common Octopus	32	27.5	7
3	103	Orcinus orca	Orca	33	27.5	5
4	104	Vampyroteuthis infernalis	Vampire Squid	34	27.5	3
5	105	Aldabrachelys gigantea	Aldabra giant tortoise	35	27.5	1
5	105	Humphead wrasse	wrasse	36	27.8	1
5	105	Fin Whale	Phinius	37	27.8	4
5	105	Beluga Whale	Benjamin	38	27.7	3

Find Expert Vet

Species Name:

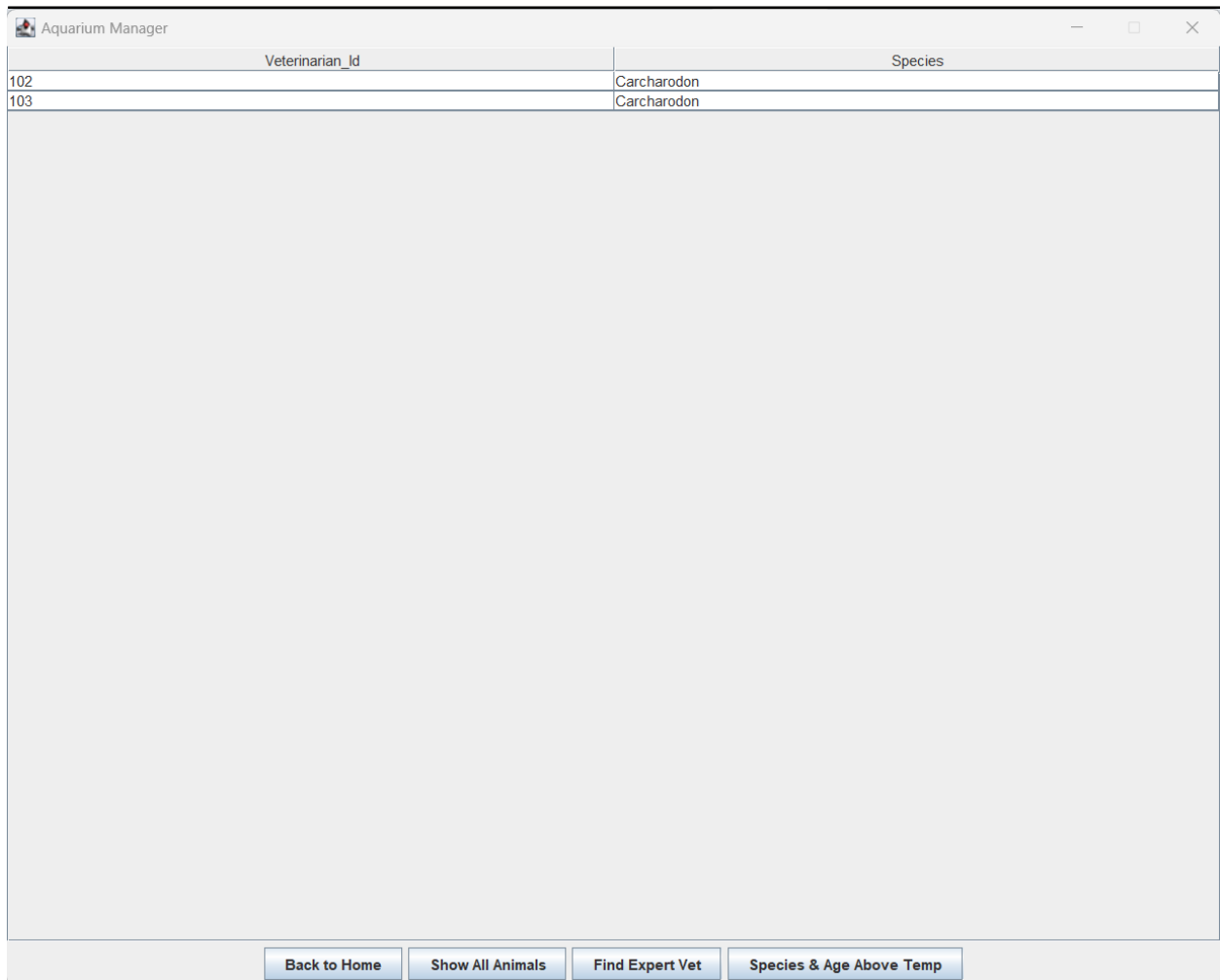
Find Cancel

Back to Home Show All Animals Find Expert Vet Species & Age Above Temp

GUI Component of Operation (DIVISION)

Application State (after Query)

It displays all the Veterinarian IDs that work with the inputted species and the corresponding species in a separate column.



Veterinarian_Id	Species
102	Carcharodon
103	Carcharodon

Back to Home Show All Animals Find Expert Vet Species & Age Above Temp

We can also test on a Species that does not exist in the Database:

University of British Columbia, Vancouver
Department of Computer Science

Aquarium Manager

WATER_TANK_ID	VETERINARIAN_ID	SPECIES	ANIMAL_NAME	ID	LIVINGTEMP	AGE
1	103	Carcharodon	Bull Shark	39	27.7	5
1	102	Carcharodon	Great White Shark	31	27.5	6
2	102	Octopus vulgaris	Common Octopus	32	27.5	7
3	103	Orcinus orca	Orca	33	27.5	5
4	104	Vampyroteuthis infernalis	Vampire Squid	34	27.5	3
5	105	Aldabrachelys gigantea	Aldabra giant tortoise	35	27.5	1
5	105	Humphead wrasse	wrasse	36	27.8	1
5	105	Fin Whale	Phinius	37	27.8	4
5	105	Beluga Whale	Benjamin	38	27.7	3

Find Expert Vet

Species Name:

Random

Find Cancel

Back to Home Show All Animals Find Expert Vet Species & Age Above Temp

It will then return:

Aquarium Manager

WATER_TANK_ID	VETERINARIAN_ID	SPECIES	ANIMAL_NAME	ID	LIVINGTEMP	AGE
1	103	Carcharodon	Bull Shark	39	27.7	5
1	102	Carcharodon	Great White Shark	31	27.5	6
2	102	Octopus vulgaris	Common Octopus	32	27.5	7
3	103	Orcinus orca	Orca	33	27.5	5
4	104	Vampyroteuthis infernalis	Vampire Squid	34	27.5	3
5	105	Aldabrachelys gigantea	Aldabra giant tortoise	35	27.5	1
5	105	Humphead wrasse	wrasse	36	27.8	1
5	105	Fin Whale	Phinius	37	27.8	4
5	105	Beluga Whale	Benjamin	38	27.7	3

Result

No experts found for Random

OK

Thank you.