

# Relational Algebra

In this assignment you will be writing *relational algebra* (not SQL) queries to select various sets of data. Attached is a schema of a fish database.

Fish- General fish categories. The TankType FK references the type of tank this fish should be kept in.

TankTypes- The types of tanks fish can be in (salt water, reef, brackish, freshwater etc)

FoodTypes- Types of food that exist in the fish universe (alge, bloodworms, flakes etc).

FishDiet- This is a many to many table describing the types of food different types of fish can eat. The pair indicates that a fish is able to eat that kind of food.

TankInstance- This is a specific tank that exists in the world. The TankType FK reference the type of tank this tank is.

FishInstance- This is a specific fish that exists in the real world.

## Questions (Each question has 20 points)

1. Select the **CommonName** and **FoodType Name** of all pairs of **Fish** and **FoodTypes** that are compatible. This means a single **Fish** may appear several times (once for each kind of food it can eat).
2. Select the *CommonName* and *ScientificName* of all **Fish** in **TankInstances** of *TankSize* 60.
3. Select all of the **FoodType Names** that can be used to feed at least one **FishInstance** in the **TankInstance** with *TankInstanceId* of 44.
4. Select all **FishInstance** *FishInstanceId* which are in a tank that is not compatible for them. In other words, if the **Fish** *TankType* is not the same as the *TankType* of the **TankInstance** a **FishInstance** is in, it is in an incompatible tank.
5. Convert the following into relational algebra:

```
SELECT Player.id, Team.name, City.name FROM Player
INNER JOIN Team ON Player.team_id = Team.id
INNER JOIN City ON Team.city_id = City.id
WHERE Player.score = 200;
```