

Assignment3

July 30, 2016

- 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).

$\pi_{Fish.CommonName, FoodTypes.Name}((Fish \bowtie_{Fish.FishId=FishDiet.Fish} FishDiet) \bowtie_{FishDiet.Food=FoodTypes.FoodId} FoodTypes)$

- 2) Select the CommonName and ScientificName of all **Fish** in **TankInstances** of TankSize 60.

$\pi_{Fish.CommonName, Fish.ScientificName}(\sigma_{TankSize=60}((Fish \bowtie_{Fish.FishId=FishInstance.FishType} FishInstance) \bowtie_{FishInstance.TankId=TankInstances.TankId} TankInstances))$

- 3) Select all of the **FoodType** Names that can be used to feed at least one **FishInstance** in the **TankInstance** with TankInstanceId of 44.

$\pi_{FoodTypes.Name}(\sigma_{TankInstance.TankInstanceId=44} \text{ and } count(FishInstance.FishInstanceId)>0(((FoodTypes \bowtie_{FoodTypes.FoodId=FishInstance.FoodId} FishInstance) \bowtie_{FishInstance.TankId=TankInstance.TankId} TankInstance)))$

- 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.

$\pi_{FishInstance.FishInstanceId}(\sigma_{Fish.TankType \neq TankInstance.TankType}(((FishInstance \bowtie_{FishInstance.FishId=Fish.FishId} Fish) \bowtie_{FishInstance.TankId=TankInstance.TankId} TankInstance) \bowtie_{FishInstance.FoodId=FoodTypes.FoodId} FoodTypes))$

- 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;

$\pi_{Player.id, Team.name, City.name}(\sigma_{Player.score=200}((Player \bowtie_{Player.team_id=Team.id} Team) \bowtie_{Team.city_id=City.id} City))$