

# Anforderungsanalyse

## 1. Aquarium

Anf 1.1: Ein Aquarium wird durch eine eindeutige Aquariumsnummer identifiziert.

Anf 1.2: Ein Aquarium erhält zusätzlich Daten zu Füllvolumen und Standort.

Anf 1.3: Ein Aquarium kann Beziehungen mit Tieren, Pflanzen, Messpunkte und Wasserwechseln haben.

## 2. Tier

Anf 2.1: Ein Tier wird durch eine eindeutige Tiernummer identifiziert. Aufgrund unterschiedlicher Farbmorphen, Zuchtlinien und nicht vollständiger Klassifizierung kann der lat. Name nicht zur alleinigen Identifizierung verwendet werden. Die Verwendung einer Kombination aus lat. Name und Spezifizierung wäre möglich, aber aufgrund der resultierenden Länge des Schlüssels, der leicht mehr als 30 Zeichen erreichen kann, wird diese Idee verworfen.

Anf 2.2: Ein Tier erhält zusätzlich Daten:

- Lat. Name
- Zuchtform
- Deutscher Name
- Temperaturbereich
- Maximale Länge
- Bereich im Aquarium
- Verhalten
- Haltung als
- Min. Anzahl
- Preisbereich
- Futtermittel

## 3. Pflanze

Anf 3.1: Eine Pflanze wird durch eine eindeutige Pflanzennummer identifiziert. Aufgrund unterschiedlicher Farbmorphen, Zuchtlinien und nicht vollständiger Klassifizierung kann der lat. Name nicht zur alleinigen Identifizierung verwendet werden. Die Verwendung einer Kombination aus lat. Name und Spezifizierung wäre möglich, aber aufgrund der resultierenden Länge des Schlüssels, der leicht mehr als 30 Zeichen erreichen kann, wird diese Idee verworfen.

Anf 3.2: Eine Pflanze erhält zusätzlich Daten:

- Lat. Name
- Zuchtform
- Deutscher Name
- Temperaturbereich
- Wuchshöhe
- Verwendung im Aquarium

#### 4. Messpunkt

Anf 4.1: Ein Messpunkt wird durch einen Zeitstempel identifiziert. Nur in Verbindung mit dem Aquarium in dem dieser Messwert gemessen wurde, entsteht aber eine eindeutige Identifikation, da zu jedem Zeitpunkt pro Aquarium nur ein Messwert gemessen werden kann.

Anf 4.2: Ein Messpunkt erhält zusätzlich einen Messwert und eine Messmenge.

#### 5. Wasserwechsel

Anf 5.1: Ein Wasserwechsel wird durch einen Zeitstempel identifiziert. Nur in Verbindung mit dem Aquarium in dem dieser Wasserwechsel durchgeführt wurde, entsteht aber eine eindeutige Identifikation, da zu jedem Zeitpunkt pro Aquarium nur ein Wasserwechsel durchgeführt werden kann.

Anf 5.2: Ein Wasserwechsel erhält zusätzlich die gewechselte Wassermenge.

#### 6. Beziehungen

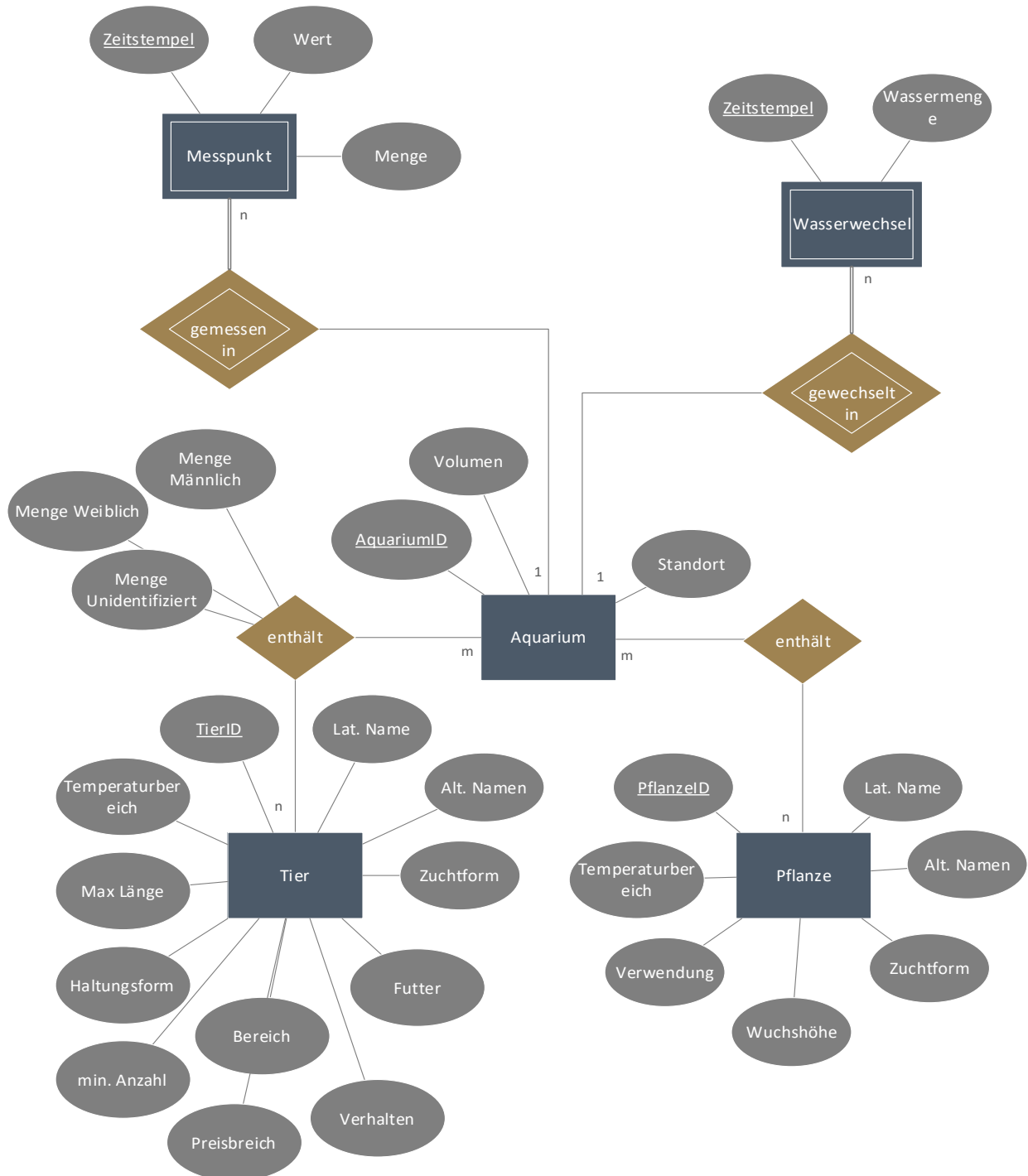
Anf 6.1: Ein Aquarium kann eine n zu m Beziehung mit Tieren haben, um abzubilden, dass eine Art von Tier in einem Aquarium gehalten wird. Zusätzliches Attribut dieser Beziehung ist eine Anzahl mit Unterscheidung zwischen männlich, weiblich und nicht identifiziert.

Anf 6.2: Ein Aquarium kann eine n zu m Beziehung mit Pflanzen haben, um abzubilden, dass eine Art von Pflanze in einem Aquarium gehalten wird.

Anf 6.3: Messwerte und Wasserwechsel werden mittels einer 1 zu n Beziehung je einem Aquarium zugeordnet.

## Konzeptioneller Entwurf

### Aquarienvverwaltung



## Logischer Entwurf

### Entitäten

Aquarium (AquariumID, Volumen, Standort)

{AquariumID} -> {Volumen, Standort}

Eine Abhängigkeit der Art {Standort} -> {AquariumID} ist nicht möglich, da der Standort nur bis auf den Raum beschrieben wird.

Tier (ID, Lat. Name, Alt. Namen, Zuchtform, Futter, Verhalten, Bereich, Temperaturbereich, max. Länge, Haltungsform, min. Anzahl, Preisbereich)

{TierID} -> {Lat. Name, Alt. Namen, ...}

{Lat. Name, Zuchtform} -> {TierID, Alt. Namen, ...}

Pflanze (ID, Lat. Name, Alt. Namen, Zuchtform, Wuchshöhe, Temperaturbereich, Verwendung)

{PflanzeID} -> {Lat. Name, Alt. Namen, ...}

{Lat. Name, Zuchtform} -> {PflanzeID, Alt. Namen, ...}

Messpunkt (↑AquariumID, Zeitstempel, Wert, Menge)

{AquariumID, Zeitstempel} -> {Wert, Menge}

Eine weitere Abhängigkeit ist nicht möglich, da {Zeitstempel, Wert, Menge} auf kein Aquarium schließen lässt.

Wasserwechsel (↑AquariumID, Zeitstempel, Wassermenge)

{AquariumID, Zeitstempel} -> {Wassermenge}

Eine weitere Abhängigkeit ist nicht möglich, da {Zeitstempel, Wassermenge} auf kein Aquarium schließen lässt.

## Beziehungen

Aquarium\_enhält\_Tier (↑Aquarium AquariumID, ↑Tier TierID, Menge Männlich, Menge Weiblich, Menge Unidentifiziert)

{↑Aquarium AquariumID, ↑Tier TierID} -> {Menge Männlich, Menge Weiblich, Menge Unidentifiziert}

Aquarium\_enhält\_Pflanze(↑Aquarium AquariumID, ↑Pflanze PflanzeID)

Hier entstehen keine funktionellen Abhängigkeiten

Da es sich bei Messpunkt und Wasserwechsel um schwache Entitäten, werden die Beziehungen Messpunkt\_gemessenIn\_Aquarium und Wasserwechsel\_gewechseltIn\_Aquarium nicht abgebildet. Diese Beziehung wird bereits bei den jeweiligen Entitäten berücksichtigt.

## Anwendung der Normalformen:

### 1. Normalform:

Ein Lateinischer Name besteht grundsätzlich immer aus einem Gattungsnamen und einem Artzusatz. Diese Spaltenaufteilung wird implementiert.

- Tier (ID, Gattungsname, Artzusatz, Alt. Namen, Zuchtform, Futter, Verhalten, Bereich, Temperaturbereich, max. Länge, Haltungsform, min. Anzahl, Preisbereich)
- Pflanze (ID, Gattungsname, Artzusatz, Alt. Namen, Zuchtform, Wuchshöhe, Temperaturbereich, Verwendung)

### 2. Normalform:

Wie in der Begründung der Verwendung einer TierID und PflanzeID bereits erwähnt, kann ein Tier oder eine Pflanze auch durch den Lateinischen Namen und eventuelle Zuchtformen identifiziert

werden. Dies führt zu einer partiellen Abhängigkeit. Zur Beseitigung dieser werden die Tabellen Tier und Pflanze aufgeteilt.

- Tier (ID, Gattungsname, Artzusatz, Zuchtform)
- Pflanze (ID, Gattungsname, Artzusatz, Zuchtform)

Zur Speicherung der Daten werden neue Tabellen erstellt:

- Tierdaten (ID, Alt. Namen, Futter, Verhalten, Bereich, min. Temperatur, max. Temperatur, max. Länge, Haltungsform, min. Anzahl, Preisbereich)
- Pflanzendaten (ID, Alt. Namen, Wuchshöhe, Temperaturbereich, Verwendung)

### 3. Normalform:

Die 3. Normalform ist überall erfüllt, da kein Attribut vom Primärschlüssel transitiv abhängig ist.

#### BCNF:

Die BCNF ist erfüllt, da nur in den Tabellen Tier und Pflanze mehr als ein Kandidatenschlüssel existiert und hier aber gilt: {ID} -> {Gattungsname, Artzusatz, Zuchtform} und {Gattungsname, Artzusatz, Zuchtform} -> {ID}. Da es sonst keine funktionalen Abhängigkeiten gibt, gibt es auch keine transitive Abhängigkeit zwischen Primattributen.

### Anomalien

#### Update

Werden Daten zu demselben Schlüssel mehrfach gespeichert, können diese bei einer Änderung nicht zeitgleich verändert werden. Es tritt also eine Anomalie auf, bei der zu einem Zeitpunkt z.B. für eine AquariumID zwei unterschiedliche Standorte gespeichert sind.

#### Insert

Wird ein Eintrag in die Datenbank aufgenommen, kann es zu dieser Anomalie kommen, wenn der Primärschlüssel oder andere Attribute keinen Wert erhalten.

#### Delete

Diese Anomalie tritt auf, wenn durch das Löschen eines Eintrages mehr Information verloren geht, als gewollt. Wenn z.B. durch das löschen des einzigen Wasserwechsels eines Aquarium auch die AquariumID mit Volumen und Standort verloren ginge.

### Integrität

#### Entitätsintegrität

Jeder Primärschlüssel darf in jeder Tabelle nur einen Eintrag haben. Es werden für einige surrogate Keys eingesetzt, da es teilweise keinen Schlüsselkandidaten gibt (zwei gleiche Aquarien ohne Inhalt im selben Raum) oder dieser zu komplex ist Tier mit {Gattungsname, Artzusatz, Zuchtform}.

#### Referentielle Integrität

Für Relationen werden nur die Primärschlüssel benutzt. Dieser wird als Fremdschlüssel verwendet und kann null sein, es sei denn, er wird in Beziehungen wie Messpunkt\_gemessenIn\_Aquarium als Teil eines Primärschlüssels verwendet.

Die Implementierung auf railway.app bleibt derweilen online.

RestAPI laut Dokumentation:

<https://aquariumverwaltungdae22-production.up.railway.app/swagger-ui.html>

Zugang Datenbank:

url: postgresql://postgres:FGbk2LJP3hpWyuCeh0z6@containers-us-west-70.railway.app:6735/railway

Oder für pgAdmin:

Hostname: containers-us-west-70.railway.app

Port: 6735

pwd: FGbk2LJP3hpWyuCeh0z6

user: postgres

SourceCode auf Github:

[https://github.com/gregor-hass/aquariumverwaltung\\_dae22](https://github.com/gregor-hass/aquariumverwaltung_dae22)

### ***1\_create.sql***

*/\* Als DBMS wird PostgreSQL verwendet. Es entspricht den Anforderungen an relationelle Datenbanken und wurde auch in der VO verwendet.\*/*

*/\* Erstellen der Tabellen laut Logischem Entwurf \*/*

DROP SCHEMA IF EXISTS AV CASCADE;

CREATE SCHEMA AV;

SET SEARCH\_PATH TO AV;

*/\*-----\*/*

*/\* Aquarien \*/*

```
CREATE TABLE aquariums (  
    aquarium_id SERIAL PRIMARY KEY,  
    aquarium_vol decimal(7,3),  
    aquarium_room VARCHAR(30)  
);
```

*/\*-----\*/*

*/\* Tier \*/*

```
CREATE TABLE animals (  
    animal_id SERIAL PRIMARY KEY,  
    genus VARCHAR(30),  
    species VARCHAR(30),  
    morph VARCHAR(30)  
);
```

*/\* Tierdaten \*/*

```
CREATE TABLE animaldata (  
    animal_id INTEGER NOT NULL,  
    common_name VARCHAR(100),  
    feed VARCHAR(100),
```

```

behaviour VARCHAR(100),
zone VARCHAR(30),
temperature_max INTEGER,
temperature_min INTEGER,
CHECK (temperature_min < temperature_max),
length_max decimal(4,1),
husbandry VARCHAR(100),
min_group INTEGER,
cost INTEGER,
FOREIGN KEY (animal_id) REFERENCES animals (animal_id) ON DELETE CASCADE

);

```

```

/*-----*/

```

```

/* Pflanze */

```

```

CREATE TABLE plants (
    plant_id SERIAL PRIMARY KEY,
    genus VARCHAR(30),
    species VARCHAR(30),
    morph VARCHAR(30)
);

```

```

/* Pflanzendaten */

```

```

CREATE TABLE plantdata (
    plant_id INTEGER NOT NULL,
    common_name VARCHAR(50),
    height decimal(4,1),
    temperature_max INTEGER,
    temperature_min INTEGER,
    CHECK (temperature_min <= temperature_max),
    usage VARCHAR(30),
    FOREIGN KEY (plant_id) REFERENCES plants (plant_id) ON DELETE CASCADE
);

```



```
);
```

```
/*-----*/
```

```
/* Messpunkt */
```

```
CREATE TABLE measurements(  
    aquarium_id INTEGER NOT NULL,  
    time_stamp timestamp NOT NULL default current_timestamp,  
    val VARCHAR,  
    amount decimal(4,2),  
    FOREIGN KEY (aquarium_id) REFERENCES aquariums (aquarium_id) ON DELETE CASCADE  
);
```

```
/*-----*/
```

```
/* Wasserwechsel */
```

```
CREATE TABLE waterchanges(  
    aquarium_id INTEGER NOT NULL,  
    time_stamp timestamp NOT NULL default current_timestamp,  
    amount decimal(5,1),  
    FOREIGN KEY (aquarium_id) REFERENCES aquariums (aquarium_id) ON DELETE CASCADE  
  
);
```

```
/*-----*/
```

```
/* Aquarium_enthält_Tier */
```

```
CREATE TABLE aquarium_contains_animal(  
    aquarium_id INTEGER NOT NULL,  
    animal_id INTEGER NOT NULL,  
    amount_male INTEGER,  
    amount_female INTEGER,  
    amount_unknown INTEGER,
```

```

FOREIGN KEY (aquarium_id) REFERENCES aquariums (aquarium_id) ON DELETE CASCADE,
FOREIGN KEY (animal_id) REFERENCES animals (animal_id) ON DELETE CASCADE
);

/*-----*/
/* Aquarium_enthält_Pflanze */
CREATE TABLE aquarium_contains_plant(
    aquarium_id INTEGER NOT NULL,
    plant_id INTEGER NOT NULL,
    FOREIGN KEY (aquarium_id) REFERENCES aquariums (aquarium_id) ON DELETE CASCADE,
    FOREIGN KEY (plant_id) REFERENCES plants (plant_id) ON DELETE CASCADE
);

/*-----*/
/* Views */
CREATE VIEW allanimaldata AS
SELECT * FROM animals NATURAL JOIN animaldata;
CREATE VIEW allplantdata AS
SELECT * FROM plants NATURAL JOIN plantdata;

```

## ***2\_populate.sql***

```
SET SEARCH_PATH TO AV;
```

```
/* Erstellen von Testdaten */
```

```
BEGIN TRANSACTION;
```

```
INSERT INTO aquariums VALUES
```

```
(1,60,'Wohnzimmer');
```

```
INSERT INTO aquariums VALUES
```

```
(2,120,'Wohnzimmer');
```

```
INSERT INTO aquariums VALUES
```

```
(3,4,'Schlafzimmer');
```

```
INSERT INTO aquariums VALUES
```

```
(4,12,'Eingang');
```

```
/*-----*/
```

```
INSERT INTO animals(animal_id,genus, species,morph) VALUES
```

```
(1,'Betta','Imbellis', 'wild');
```

```
INSERT INTO animals(animal_id,genus, species,morph) VALUES
```

```
(2,'Betta','Imbellis', 'Red Band');
```

```
INSERT INTO animals(animal_id,genus, species) VALUES
```

```
(3,'Danio','Margaritatus');
```

```
INSERT INTO animals(animal_id,genus, species) VALUES
```

```
(4,'Otocinclus', 'Affinis');
```

```
INSERT INTO animals(animal_id,genus, species) VALUES
```

```
(5,'Corydoras', 'Panda');
```

INSERT INTO animals(animal\_id,genus, species,morph) VALUES

(6,'Corydoras', 'Panda', 'long fin');

INSERT INTO animals(animal\_id,genus, species) VALUES

(7,'Corydoras','Sterbai');

/\*-----\*/

INSERT INTO animaldata VALUES

((SELECT animal\_id from animals WHERE genus = 'Betta' and species = 'Imbellis' and morph = 'wild'),'Friedlicher Kampffisch','Pelets, Flockenfutter, Proteinreich', 'friedlicher Einzelgänger', 'Oberflächennah', 32,24,4,'Einzel oder Paarweise, genug Unterschlupf bieten', 1,12);

INSERT INTO animaldata VALUES

((SELECT animal\_id from animals WHERE genus = 'Betta' and species = 'Imbellis' and morph = 'Red Band'),'Friedlicher Kampffisch','Pelets, Flockenfutter, Proteinreich', 'friedlicher Einzelgänger', 'Oberflächennah', 32,24,4,'Einzel oder Paarweise, genug Unterschlupf bieten', 1,12);

INSERT INTO animaldata VALUES

((SELECT animal\_id from animals WHERE genus = 'Danio' and species = 'Margaritatus'), 'Perlhuhnbarbling, Celestial Pearl Danio, CPD, Galaxy Rasboras', 'schwebendes Futter, fressen nicht von Boden oder Oberfläche', 'leicht scheue Gruppen- oder Schwarmtiere', 'Mitte-Unten',24,20, 2.5, 'Gruppe', 6,7);

INSERT INTO animaldata VALUES

((SELECT animal\_id from animals WHERE genus = 'Otocinclus' and species = 'Affinis'), 'Ohrgitter Harnischwels', 'Aufwuchsfresser, Algentabs oder Spirulinabelag auf Wurzeln/Stein', 'Saugwels, vermehrt auf Glasflächen oder glattem Holz/Steinen', 'Überall wo Aufwuchs wächst', 27,20,5, 'Gruppe', 6,5);

INSERT INTO animaldata VALUES

((SELECT animal\_id from animals WHERE genus = 'Corydoras' and species = 'Panda' and morph IS NULL), 'Panda Panzerwelse', 'Grundelnde Restefresser, rote Wurmähnliche Nahrung bevorzugt', 'quirlicher Gruppenfisch', 'Boden, manchmal Mitte', 27,22,5,'Gruppe', 6,6);

/\*-----\*/

INSERT INTO plants(genus, species, morph) VALUES

```
('Echinodorus','Grisebachii', 'Amazonicus');
```

```
INSERT INTO plants(genus, species, morph) VALUES
```

```
('Rotala','sp.', 'Pearl');
```

```
INSERT INTO plants(genus, species, morph) VALUES
```

```
('Microsorium','Pteropus', 'Trident');
```

```
INSERT INTO plants(genus, species, morph) VALUES
```

```
('Microsorium','Pteropus', 'Windeløv');
```

```
INSERT INTO plants(genus, species, morph) VALUES
```

```
('Bucephalandra','sp.', 'Serimbu Brown');
```

```
INSERT INTO plants(genus, species) VALUES
```

```
('Utricularia','Graminifolia');
```

```
INSERT INTO plantdata VALUES
```

```
((SELECT plant_id from plants WHERE genus = 'Echinodorus' and morph =  
'Amazonicus'),'Amazonas-Schwertpflanze',30,28,22,'Mitte - Hintergrund');
```

```
INSERT INTO plantdata VALUES
```

```
((SELECT plant_id from plants WHERE genus = 'Rotala' and morph = 'Pearl'),NULL,25,30,18,'Mitte -  
Hintergrund');
```

```
INSERT INTO plantdata(plant_id, usage) VALUES
```

```
((SELECT plant_id from plants WHERE genus = 'Bucephalandra' and morph = 'Serimbu  
Brown'),'Aufsitzer');
```

```
INSERT INTO plantdata(plant_id, usage) VALUES
```

```
((SELECT plant_id from plants WHERE genus = 'Utricularia' and species =  
'Graminifolia'),'Bodendecker');
```

INSERT INTO aquarium\_contains\_animal VALUES

(1,1,1,1,0);

INSERT INTO aquarium\_contains\_animal VALUES

(1,5,3,8,0);

INSERT INTO aquarium\_contains\_animal VALUES

(2,4,0,0,10);

INSERT INTO aquarium\_contains\_animal VALUES

(2,3,5,10,1);

INSERT INTO aquarium\_contains\_animal VALUES

(2,7,8,7,0);

INSERT INTO aquarium\_contains\_animal VALUES

(3,3,0,0,12);

INSERT INTO aquarium\_contains\_plant VALUES

(1,(SELECT plant\_id from plants WHERE genus = 'Echinodorus' and morph = 'Amazonicus'));

INSERT INTO aquarium\_contains\_plant VALUES

(1,(SELECT plant\_id from plants WHERE genus = 'Utricularia' and species = 'Graminifolia'));

INSERT INTO aquarium\_contains\_plant VALUES

(2,(SELECT plant\_id from plants WHERE genus = 'Bucephalandra' and morph = 'Serimbu Brown'));

INSERT INTO aquarium\_contains\_plant VALUES

(3,(SELECT plant\_id from plants WHERE genus = 'Microsorium' and species = 'Pteropus' and morph = 'Trident'));

INSERT INTO measurements(aquarium\_id, val, amount) VALUES

(1, 'PO4', 0.2),

(1, 'NO3', 0.8),

(1, 'NO2', 0.5),

(1, 'CO2', 0.2);

INSERT INTO measurements(aquarium\_id, val, amount) VALUES

```
(2, 'PH', 6.8),  
(2, 'NO3', 0.8),  
(2, 'KH', 6),  
(2, 'GH', 8);
```

```
INSERT INTO measurements(aquarium_id, val, amount) VALUES
```

```
(3, 'PO4', 0.2),  
(3, 'NO3', 0.8),  
(3, 'NO2', 0.5),  
(3, 'Fe', 0.2);
```

```
INSERT INTO waterchanges(aquarium_id, amount) VALUES
```

```
(1, 20),  
(2, 30),  
(3, 1),  
(4, 3);
```

```
COMMIT;
```

### ***3\_delete.sql***

```
SET SEARCH_PATH TO AV;
```

```
/* Löscht alle Tiere der Gattung Corydoras*/
```

```
DELETE FROM animals WHERE Genus = 'Corydoras';
```

```
/* Löscht die spezifische Pflanze Microsorium Pteropus 'Trident' nicht aber den morph 'Windeløv'*/
```

```
DELETE FROM plants WHERE Genus = 'Microsorium' and species = 'Pteropus' and morph = 'Trident';
```

```
/* Löscht das Aquarium mit der id 2 */
```

```
DELETE FROM aquariums WHERE aquarium_id = 2;
```



#### ***4\_delete.sql***

```
SET SEARCH_PATH TO av;
```

```
-- drop the views
```

```
DROP VIEW IF EXISTS allanimaldata;
```

```
DROP VIEW IF EXISTS allplantdata;
```

```
-- drop the tables one by one (and cascade the delete)
```

```
DROP TABLE IF EXISTS aquariums CASCADE;
```

```
DROP TABLE IF EXISTS animals CASCADE;
```

```
DROP TABLE IF EXISTS animaldata CASCADE;
```

```
DROP TABLE IF EXISTS plants CASCADE;
```

```
DROP TABLE IF EXISTS plantdata CASCADE;
```

```
DROP TABLE IF EXISTS measurements CASCADE;
```

```
DROP TABLE IF EXISTS waterchanges CASCADE;
```

```
DROP TABLE IF EXISTS aquarium_contains_plant CASCADE;
```

```
DROP TABLE IF EXISTS aquarium_contains_animal CASCADE;
```

```
-- drop the whole schema
```

```
DROP SCHEMA IF EXISTS av CASCADE;
```

### **5\_grants.sql**

```
SET SEARCH_PATH TO AV;
```

```
CREATE ROLE viewgroup;
```

```
/* Gruppe die Daten über Tiere und Pflanzen abrufen darf, aber keine Daten ändern. z.B. für  
Öffentlichen Zugriff auf Informationen oder digitale Infotafeln. Es werden nur select Rechte gewährt,  
da dieser User keine Änderungen vornehmen darf*/
```

```
CREATE ROLE sciencegroup;
```

```
/* Gruppe die wissenschaftliche Messdaten erfassen darf. Bekommt auch Lesenden Zugriff auf die  
Aquarenliste, um die passende ID finden zu können.*/
```

```
GRANT CONNECT ON DATABASE railway to viewgroup;
```

```
GRANT USAGE ON SCHEMA av to viewgroup;
```

```
GRANT SELECT ON
```

```
av.aquariums,av.animals,av.animaldata,av.aquarium_contains_animal,av.plants,av.plantdata,av.aqu  
arium_contains_plant,av.allanimaldata,av.allplantdata TO viewgroup;
```

```
GRANT CONNECT ON DATABASE railway to sciencegroup;
```

```
GRANT USAGE ON SCHEMA av to sciencegroup;
```

```
GRANT SELECT ON av.aquariums TO sciencegroup;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE ON av.measurements TO sciencegroup;
```

```
CREATE USER testuser WITH PASSWORD 'testpassword' IN ROLE viewgroup;
```

```
CREATE USER scienceuser WITH PASSWORD 'testpassword' IN ROLE sciencegroup;
```

```
/* -----
```

```
/* test grants with user testuser*/
```

```
select * from av.animals
```

```
select * from av.allanimaldata
```

```
/* outputs the TABLE*/
```

```
INSERT INTO plants(genus, species) VALUES
```

```
('Testplant','Graminifolia');
```

```
/*
```

```
ERROR: permission denied for table plants
```

```
SQL state: 42501
```

```
*/
```

```
select * from av.measurements
```

```
/*
```

```
ERROR: permission denied for table measurements
```

```
SQL state: 42501
```

```
*/
```

```
/* -----
```

```
/* test grants with user scienceuser*/
```

```
select * from aquariums
```

```
/* outputs the TABLE*/
```

```
INSERT INTO aquariums VALUES
```

```
(1,60,'Test');
```

```
/*
```

```
ERROR: permission denied for table aquariums
```

```
SQL state: 42501
```

```
*/
```

```
INSERT INTO measurements(aquarium_id, val, amount) VALUES
```

```
(1, 'PO4', 0.2),
```

```
/* add the measurement*/
```

```
select * from measurements
```

```
/* outputs the TABLE*/
```

```
/* drop user */
```

```
REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA av FROM viewgroup;
```

```
REVOKE ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA av FROM viewgroup;
```

```
REVOKE ALL PRIVILEGES ON ALL FUNCTIONS IN SCHEMA av FROM viewgroup;
```

```
REVOKE ALL PRIVILEGES ON SCHEMA av FROM viewgroup;
```

```
REVOKE ALL PRIVILEGES ON DATABASE railway FROM viewgroup;
```

```
DROP USER viewgroup;
```

## **6\_select.sql**

```
SET SEARCH_PATH TO AV;
```

```
/* Ausgabe der Tiere in einem bestimmten Aquarium */
```

```
SELECT * FROM aquarium_contains_animal NATURAL JOIN animals WHERE aquarium_id = 1;
```

```
/* Ausgabe der Tiere in einem bestimmten Aquarium */
```

```
SELECT * FROM aquarium_contains_plant NATURAL JOIN plants WHERE aquarium_id = 1;
```

```
/* Ausgabe des Wissenschaftlichen Namen aller Tiere und Pflanzen in einem bestimmten Aquarium */
```

```
SELECT genus, species, morph FROM (aquarium_contains_animal NATURAL JOIN animals) NATURAL  
FULL OUTER JOIN (aquarium_contains_plant NATURAL JOIN plants) WHERE aquarium_id=1;
```

```
/* Ausgabe aller Tiere die in keinem Aquarium gehalten werden */
```

```
SELECT * FROM animals WHERE animal_id IN (SELECT animal_id FROM animals EXCEPT SELECT  
animal_id FROM aquarium_contains_animal)
```

```
/* Views */
```

```
/* drop views (duplicate to 3_delete.sql) */
```

```
DROP VIEW allanimaldata;
```

```
DROP VIEW allplantdata;
```

```
/* Erstellen einer View fuer jeweils alle Tier- und Pflanzenarten (duplicate to 1_create.sql)*/
```

```
CREATE VIEW allanimaldata AS
```

```
SELECT * FROM animals NATURAL JOIN animaldata;
```

```
CREATE VIEW allplantdata AS
```

```
SELECT * FROM plants NATURAL JOIN plantdata;
```

```
/* Anzeigen der Daten über Views */
```

```
select * from allanimaldata
```

```
select * from allplantdata
```

## **7\_rest\_requests.sql**

### **# REST Client Plugin**

\* see here:

<https://marketplace.visualstudio.com/items?itemName=humao.rest-client>

\* Should replace the postman collection in the future

\* Check also the **\*\*swagger ui documentation\*\*** for more infos about possible requests:

\* ``<https://dae5-raindrops-ws2022-test-1-production.up.railway.app/swagger-ui.html>``

### **# Variables**

\* Definition with @ and usage with {{...}}

@url = <https://aquariumverwaltungdae22-production.up.railway.app/>

### **# Info - Controller**

GET {{url}}/info HTTP/1.1

### **# Aquarium Controller**

GET {{url}}/newAquarium/testroom/56 HTTP/1.1

answer:

new 56.0l Aquarium saved for testroom

### **## Get All Aquariums**

GET {{url}}/aquarium HTTP/1.1

answer:

```
[
  {
    "aquariumId": 1,
    "volume": 60,
    "room": "Wohnzimmer"
  },
  ...
  {
    "aquariumId": 5,
    "volume": 56,
    "room": "testroom"
  }
]
```

### **# Animal Controller**

### **## Add new animals**

all 3 ways to add an animal:

POST {{url}}/newAnimal/TestGenusREST1/TestSpeciesREST1 HTTP/1.1

answer:

new Animal 8 (TestGenusREST1 TestSpeciesREST1) saved

-----

POST {{url}}/newAnimal/TestGenusREST2/TestSpeciesREST2/TestMorphREST2  
HTTP/1.1

answer:

new Animal 9 (TestGenusREST2 TestSpeciesREST2 TestMorphREST2) saved

-----

POST {{url}}/newAnimalMorph/TestGenus3/TestMorph3 HTTP/1.1

answer:

new Animal 10 (TestGenus3 sp. TestMorph3) saved

### **## Get All Animals**

GET {{url}}/animals HTTP/1.1

answer:

```
[
  {
    "animalId": 1,
    "genus": "Betta",
    "species": "Imbellis",
    "morph": "wild"
  },
  ...
  {
    "animalId": 9,
    "genus": "TestGenusREST2",
    "species": "TestSpeciesREST2",
    "morph": "TestMorphREST2"
  },
  {
    "animalId": 10,
    "genus": "TestGenus3",
    "species": "sp.",
    "morph": "TestMorph3"
  }
]
```

### **# Waterchange Controller**

#### **## Add new waterchanges**

POST {{url}}/newWaterchange/1/50 HTTP/1.1

answer:  
new Waterchange at 2023-03-31 08:39:29.312605 (Aquarium 1: 50.01 )  
saved

**## Get Waterchanges for aquarium\_id**  
POST {{url}}/waterchanges/1 HTTP/1.1

answer:  
[  
 {  
 "aquarium\_id": 1,  
 "time": "2023-03-30T14:29:07.713+0000",  
 "amount": 20  
 },  
 {  
 "aquarium\_id": 1,  
 "time": "2023-03-31T08:39:29.312+0000",  
 "amount": 50  
 }  
]

**# Waterchange Controller**  
**## Add new waterchanges**

POST {{url}}/newMeasurement/1/NO2/0.3 HTTP/1.1

answer:  
new Measurement at 2023-03-31 09:13:45.960769 (Aquarium 1: 0.3 NO2 )  
saved

**## Get Waterchanges for aquarium\_id**  
POST {{url}}/measurements/1 HTTP/1.1

answer:  
[  
 {  
 "aquarium\_id": 1,  
 "time": "2023-03-30T14:29:07.713+0000",  
 "amount": 0.2,  
 "type": "P04"  
 },  
 ...  
 {  
 "aquarium\_id": 1,  
 "time": "2023-03-30T14:29:07.713+0000",



```
    "amount": 0.2,  
    "type": "CO2"  
  },  
  {  
    "aquarium_id": 1,  
    "time": "2023-03-31T09:13:45.960+0000",  
    "amount": 0.3,  
    "type": "NO2"  
  }  
]
```