

Ausarbeitung des Datenbankpraktikums für die Miniwelt „Tierwelt“

Zielstellung: Erstellen einer Datenbank für die Reptilien-AG des Emmy-Noether-Gymnasiums

Inhalt

| | |
|------------------------------------------------------------------|----|
| doc | 4 |
| tierwelt.txt | 4 |
| tierwelt.pdf | 4 |
| modell | 4 |
| tierwelt-Welt.txt | 4 |
| tierwelt-ERModell.svg [zusätzlich tierwelt-ERModell.dia] | 5 |
| tierwelt-SemBed.txt | 6 |
| tierwelt-FAen.txt | 6 |
| tierwelt-RelMod.txt | 7 |
| tierwelt-Trafo.txt | 8 |
| tierwelt-Daten.txt | 9 |
| tierwelt-Daten. xlsx | 9 |
| tierwelt-Habitattypen.csv | 9 |
| tierwelt-Terrarien.csv | 9 |
| tierwelt-Unterarten.csv | 9 |
| tierwelt-Tiere.csv | 9 |
| tierwelt-Verbrauchsmittel.csv | 9 |
| tierwelt-Leuchtmittel.csv | 9 |
| tierwelt-Bodenbelag.csv | 9 |
| tierwelt-Pflanzen.csv | 9 |
| tierwelt-Wasserbecken.csv | 9 |
| tierwelt-Deko.csv | 9 |
| tierwelt-Arbeitsdiensttypen.csv | 9 |
| tierwelt-Arbeitsdienste.csv | 9 |
| tierwelt-Verbrauch.csv | 10 |
| tierwelt-Futter.csv | 10 |
| tierwelt-Tierpflege.csv | 10 |
| tierwelt-Terrariumpflege.csv | 10 |
| tierwelt-Lagerpflege.csv | 10 |
| tierwelt-Anfragen.txt | 10 |
| relational | 10 |
| DES create-tierwelt.sql | 10 |
| DES insert-tierwelt.sql | 10 |
| DES tierwelt SQL-Abfragen.sql | 10 |
| DES tierwelt RA-Abfragen.ra | 11 |
| sql | 11 |
| create-tierwelt.sql | 11 |
| drop-tierwelt.sql | 11 |
| grant-tierwelt.sql | 11 |

| | |
|--------------------------------------------------|----|
| insert-tierwelt.sql | 11 |
| insert-tierwelt_dienstestrang.sql | 11 |
| insert-tierwelt_tierstrang.sql..... | 11 |
| insert-tierwelt_verbrauchsmittelstrang.sql | 11 |
| install-tierwelt.sql..... | 11 |
| uninstall-tierwelt.sql..... | 11 |
| update.sql | 11 |
| Tierwelt-query.sql..... | 11 |
| Tierwelt-query.ans..... | 11 |
| did | 12 |
| tierwelt-allgemein_did.txt..... | 12 |
| tierwelt-aufgaben_lsg_did.txt | 12 |
| view | 15 |
| Tierwelt-view.txt..... | 15 |
| classFormular.php | 15 |
| classOverview.php..... | 15 |
| classTasks.php | 15 |
| index.php | 16 |
| jQuery_valid.php..... | 16 |
| showChange.php | 16 |
| showFormular.php | 16 |
| showOverview.php..... | 16 |
| showTasks.php | 16 |

doc

[tierwelt.txt](#)

[im entsprechenden Ordner zu finden]

[tierwelt.pdf](#)

[im entsprechenden Ordner zu finden]

modell

[tierwelt-Welt.txt](#)

Beschreibung der Miniwelt „Tierwelt“

Bereits seit einigen Jahren können Besucher:innen des Emmy-Noether-Gymnasiums in den Räumlichkeiten zahlreiche Reptilien bewundern. Verantwortlich hierfür ist die Reptilien-AG aus Schüler:innen, eine Person im freiwilligen ökologischen Jahr und eine betreuenden Lehrkraft.

Die Beteiligten achten darauf, dass die Tiere in den unterschiedlichen Terrarien ein auf ihre Art angepasstes Habitat vorfinden. Die Unterschiede in der Ernährung und Versorgung der Tiere, aber auch die Pflege und Instandhaltung der unterschiedlichen Lebensbereiche ist herausfordernd und interessant zugleich.

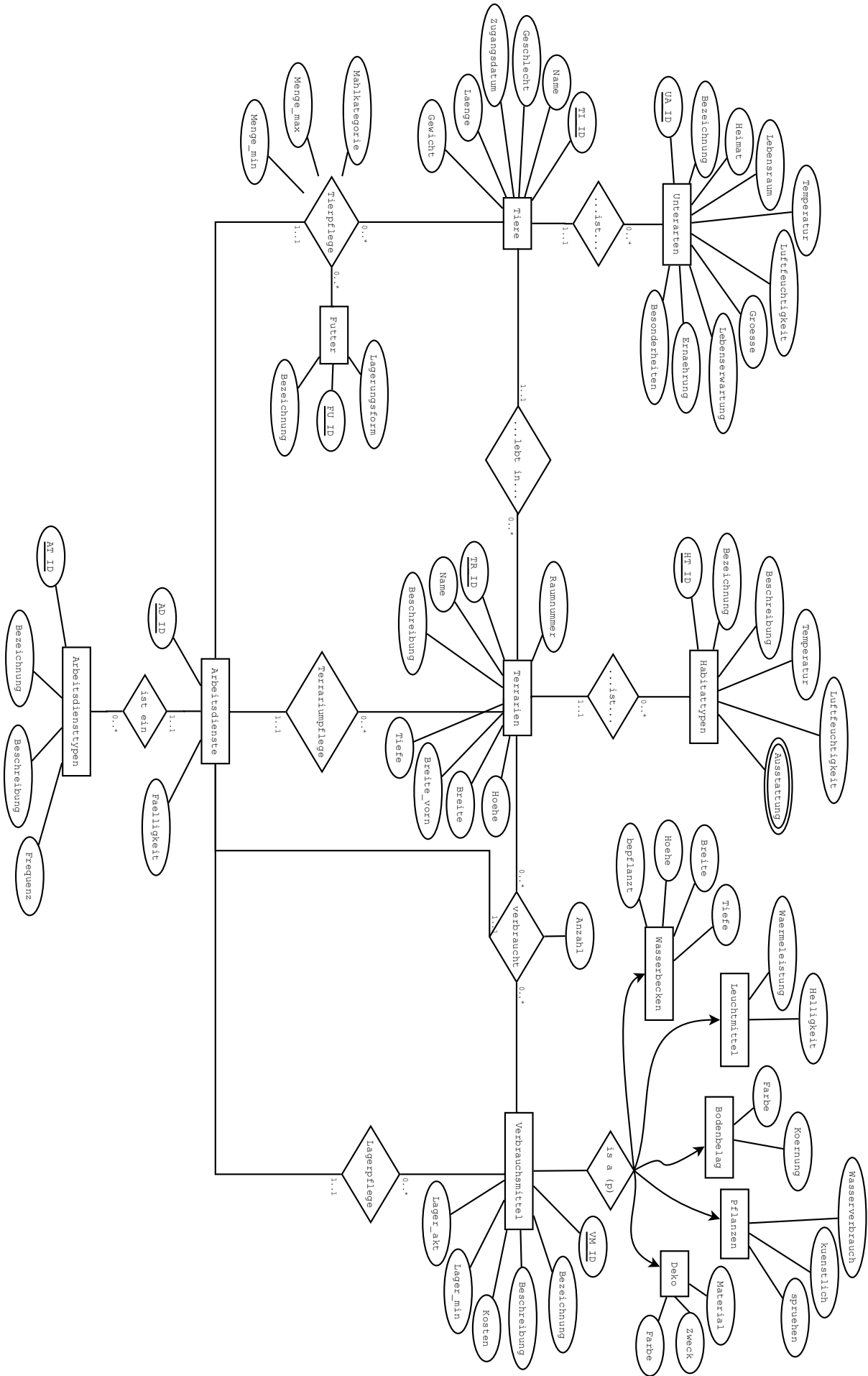
Für die Zusammenarbeit ist es besonders wichtig, die unterschiedlichen Dienste und Verantwortlichkeiten – Tierpflege, Terrariumpflege und Verbrauchsmittelpflege - transparent und so abwechslungsreich wie möglich zu gestalten, wobei Stellvertretersysteme für eine Absicherung im Krankheitsfall sorgen sollen.

Genauere Beschreibung der Dienste

Im Bereich der Tierpflege ist insbesondere darauf zu achten, dass die Tiere individuell abgestimmtes Futter in unterschiedlichen Frequenzen bekommen.

Terrariumpflege bedeutet vor allem, darauf zu achten, dass die Bepflanzung, Bewässerung und Temperatureinstellung den Anforderungen entsprechen.

Verbrauchsmittelpflege heißt, die Bevorratung von Futter und Ersatzteilen zu überprüfen und zu gewährleisten.



tierwelt-SemBed.txt

Semantische Bezüge bzw. Integritätsbedingungen, die sich nicht im Modell wiederfinden

theoretisch unbeschränkte Anzahl der Fütterungen etc

Es gibt keine Kontrolle darüber, wie viele Fütterungsdienste zu einem Tier erstellt werden. Ähnlich verhält es sich mit anderen Dienstarten.

Anzeige des Fälligkeitsdatums bei Diensten, die mehrmals pro Woche ausgeführt werden müssen

Bei solchen Diensten wird lediglich das aktuellste Datum in der Woche als Fälligkeitsdatum angezeigt. [Bei täglichen Diensten also auch nur ein Wochentag.]

tierwelt-FAen.txt

Übersicht über die funktionalen Abhängigkeiten

Relation Unterarten

UA_ID → {Bezeichnung, Heimat, Lebensraum, Temperatur, Luftfeuchtigkeit, Groesse, Lebenserwartung, Ernaehrung, Besonderheiten}

sowie alle trivialen Kombinationen, in denen UA_ID enthalten ist

Relation Tiere

TI_ID → {Name, UA_ID, Geschlecht, Zugangsdatum, Laenge, Gewicht, TR_ID}

sowie alle trivialen Kombinationen, in denen TI_ID enthalten ist

{Name, Zugangsdatum} → {TI_ID, UA_ID, Geschlecht, Laenge, Gewicht, TR_ID}

Relation Tierpflege

AD_ID → {TI_ID, FU_ID, Menge_min, Menge_max, Mahlkatgorie}

sowie alle trivialen Kombinationen, in denen AD_ID enthalten ist

Relation Habitattypen

HT_ID → {Bezeichnung, Beschreibung, Temperatur, Luftfeuchtigkeit, Ausstattung}

sowie alle trivialen Kombinationen, in denen HT_ID enthalten ist

Relation Terrarien

TR_ID → {Name, Beschreibung, Raumnummer, HT_ID, Hoehe, Tiefe, Breite, Breite_vorn}

sowie alle trivialen Kombinationen, in denen TR_ID enthalten ist

{Name, Raumnummer} → {TR_ID, Beschreibung, HT_ID, Hoehe, Tiefe, Breite, Breite_vorn}

Relation Terrariumpflege

AD_ID → TR_ID

sowie alle trivialen Kombinationen, in denen AD_ID enthalten ist

Relation Arbeitsdienste

AD_ID → {AT_ID, Faelligkeit}

sowie alle trivialen Kombinationen, in denen AD_ID enthalten ist

Relation Arbeitsdiensttypen

AT_ID → {Bezeichnung, Beschreibung, Frequenz}

sowie alle trivialen Kombinationen, in denen AT_ID enthalten ist

Relation Wasserbecken

VM_ID → {bepflanzt, Breite, Hoehe, Tiefe}

sowie alle trivialen Kombinationen, in denen VM_ID enthalten ist

Relation Deko

VM_ID → {Material, Farbe, Zweck}

sowie alle trivialen Kombinationen, in denen VM_ID enthalten ist

Relation Pflanzen

VM_ID → {Wasserverbrauch, kuenstlich, spruehen}

sowie alle trivialen Kombinationen, in denen VM_ID enthalten ist

Relation Leuchtmittel

VM_ID → {Waermeleistung, Helligkeit}

sowie alle trivialen Kombinationen, in denen VM_ID enthalten ist

Relation Bodenbelag

VM_ID → {Farbe, Koernung}

sowie alle trivialen Kombinationen, in denen VM_ID enthalten ist

Relation Verbrauchsmittel

VM_ID → {Bezeichnung, Beschreibung, Lager_min, Lager_akt, Kosten}

sowie alle trivialen Kombinationen, in denen VM_ID enthalten ist

Relation Verbrauch

{TR_ID, VM_ID} → {Anzahl, AD_ID}

AD_ID → {TR_ID, Anzahl, VM_ID}

sowie alle trivialen Kombinationen, in denen AD_ID enthalten ist

Relation Lagerpflege

AD_ID → VM_ID

Relation Futter

FU_ID → {Bezeichnung, Lagerungsform}

[tierwelt-RelMod.txt](#)

Relationenmodell der Miniwelt

Habitattypen (HT_ID#, Bezeichnung *varchar*, Beschreibung *varchar*, Temperatur *int*, Luftfeuchtigkeit *int*, Ausstattung *varchar*)

Terrarien (TR_ID#, Name *varchar*, Beschreibung *varchar*, Raumnummer *varchar*, Breite *int*, Breite_vorn *int*, Tiefe *int*, Hoehe *int*, HT_ID)

Unterarten (UA_ID#, Bezeichnung *varchar*, Heimat *varchar*, Lebensraum *varchar*, Temperatur *varchar*, Luftfeuchtigkeit *varchar*, Groesse *varchar*, Lebenserwartung *varchar*, Ernaehrung *varchar*, Besonderheiten *varchar*)

Tiere (TI_ID#, Name *varchar*, Geschlecht *char*, Laenge *int*, Gewicht *int*, Zugangsdatum *date*, UA_ID#, TR_ID#)

Verbrauchsmittel (VM_ID#, Bezeichnung *varchar*, Beschreibung *varchar*, Lager_min *int*, Lager_akt *int*, Kosten *float*)

Leuchtmittel (VM_ID#, Helligkeit *int*, Waermeleistung *int*)

Bodenbelag (VM_ID#, Farbe *varchar*, Koernung *varchar*)

Pflanzen (VM_ID#, Wasserverbrauch *int*, kuenstlich *bool*, spruehen *bool*)

Wasserbecken (VM_ID#, Hoehe *int*, Breite *int*, Tiefe *int*, bepflanzt *bool*)

Deko (VM_ID#, Material *varchar*, Farbe *varchar*, Zweck *varchar*)

Arbeitsdiensttypen (AT_ID#, Bezeichnung varchar, Beschreibung varchar, Frequenz int)

Arbeitsdienste (AD_ID#, AT_ID#, Faelligkeit date)

Verbrauch (TR_ID#, VM_ID#, AD_ID#, Anzahl int)

Futter (FU_ID#, Bezeichnung varchar, Lagerungsform varchar)

Tierpflege (TI_ID#, FU_ID#, AD_ID#, Menge_min int, Menge_max int, Mahlkatgorie int)

Terrariumpflege (AD_ID#, TR_ID#)

Lagerpflege (VM_ID#, AD_ID#)

[tierwelt-Trafo.txt](#)

Transformation (Abbildung) von Entity-Relationship-Modells in das relationale Modell

1. Regel:

Jeder Entitätstyp wird als Tabelle (Relation) dargestellt. Der Schlüssel des Entitätstyps wird Schlüssel der Tabelle.

- ➔ Regel wurde angewendet auf folgende Entitätstypen der Tierwelt-DB:
"Unterarten", "Tiere", "Futter", "Habitattypen", "Terrarien", "Arbeitsdiensttypen",
"Arbeitsdienste", "Wasserbecken", "Leuchtmittel", "Bodenbelag", "Pflanzen", "Deko" und
"Verbrauchsmittel"

2. Regel:

Jede $n : m$ - Beziehung wird durch eine eigene Tabelle dargestellt.

- ➔ Regel wurde NICHT angewendet auf Relationen der Tierwelt-DB

3. Regel:

Jede 1:n und 1:1 - Beziehung, die eigene Attribute hat, wird durch eine eigene Tabelle dargestellt.

- ➔ Regel wurde angewendet auf folgende Relationen der Tierwelt-DB:
"Verbrauch"

4. Regel:

Hat eine 1:1 oder 1:n - Beziehung keine eigenen Attribute dann gilt:

Ist eine Entität aus E2 zwingendes Mitglied (d. h. genau 1) einer 1 : n - Beziehung mit einer Entität aus E1, dann erhält E2 den primären Schlüssel von E1 als Attribut (Fremdschlüssel).

Sofern eine Entität aus E2 freies Mitglied (d. h. 0,1 - konditionell) der 1 : n - Beziehung mit einer Entität aus E1 ist, wird diese Beziehung gewöhnlich in einer eigenen Relation dargestellt.

- ➔ Regel wurde angewendet auf folgende Relationen der Tierwelt-DB:

- „Tier“ [1..1] ... ist ... [0..*] „Unterart“: jedes Tier ist zwingendes Mitglied genau einer Unterart: Tiere erhalten primären Schlüssel von Unterart (ua_id) als Fremdschlüssel
- „Tier“ [1..1] ... lebt in ... [0..*] „Terrarium“: jedes Tier ist zwingendes Mitglied genau eines Terrariums: Tiere erhalten primären Schlüssel von Terrarien (tr_id) als Fremdschlüssel
- „Terrarium“ [1..1] ... ist vom ... [0..*] „Habitattypen“: jedes Terrarium ist zwingendes Mitglied genau eines Habitattypen: Terrarien erhalten primären Schlüssel von Habitattypen (ht_id) als Fremdschlüssel
- „Arbeitsdienst“ [1..1] ... ist ein ... [0..*] „Arbeitsdiensttypen“: jeder Arbeitsdienst ist zwingendes Mitglied genau eines Arbeitsdiensttypen: Arbeitsdienst erhält primären Schlüssel von Arbeitsdiensttypen (at_id) als Fremdschlüssel
- IS-A-Relationen (disjunkt partiell): „Wasserbecken“ ... is a (p) ... „Verbrauchsmittel“, „Leuchtmittel“ ... is a (p) ... „Verbrauchsmittel“, „Bodenbelag“ ... is a (p) ... „Verbrauchsmittel“, „Pflanzen“ ... is a (p) ... „Verbrauchsmittel“, „Deko“ ... is a (p) ... „Verbrauchsmittel“: jedes Verbrauchsmittel ist (genau) ein Wasserbecken, Leuchtmittel, Bodenbelag, Pflanze oder Dekorationsartikel: primärer Schlüssel von Verbrauchsmittel (vm_id) wird für alle IS-A-Relationen / -Tabellen als eigener primärer Schlüssel verwendet

- IS-A-Relationen (disjunkt total): "Lagerpflege" ... is a (t) ... „Arbeitsdienst“, "Terrariumpflege" ... is a (t) ... „Arbeitsdienst“, "Tierpflege" ... is a (t) ... „Arbeitsdienst“: jeder Arbeitsdienst muss genau einem Pflegedienst zugeordnet werden: primärer Schlüssel von Arbeitsdienst (ad_id) wird für alle Pflege-Tabellen als eigener primärer Schlüssel verwendet

[tierwelt-Daten.txt](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Daten.xlsx](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Habitattypen.csv](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Terrarien.csv](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Unterarten.csv](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Tiere.csv](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Verbrauchsmittel.csv](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Leuchtmittel.csv](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Bodenbelag.csv](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Pflanzen.csv](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Wasserbecken.csv](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Deko.csv](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Arbeitsdiensttypen.csv](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Arbeitsdienste.csv](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Verbrauch.csv](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Futter.csv](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Tierpflege.csv](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Terrariumpflege.csv](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Lagerpflege.csv](#)

[im entsprechenden Ordner zu finden]

[tierwelt-Anfragen.txt](#)

Alle Anfragen nummeriert

- (1) Zeige alle Reptilienarten (vollständig) an!
- (2) Zeige die Namen aller Tiere an!
- (3) Zeige die Name aller weiblichen Tiere an!
- (4) Zeige alle Tiere mit ihren Informationen zur jeweiligen Tierart an!
- (5) Zeige die Namen aller Tiere an, die zu den „Bartagame“ gehören!
- (6) Zeige die Namen aller Tiere an, welche in einem Terrarium des Typs „Wüste“ leben!
- (7) Zeige an, wann „Exo“ zunächst gefüttert werden muss!
- (8) Zeige die ID, Namen und Beschreibungen aller Arbeitsdienste an, welche diese Woche fällig werden!
- (9) Zeige an, wann bei „Exo“ das nächste mal das Wasser gewechselt werden muss!
- (10) Zeige alle Terrarien an, für welche es mindestens zwei aktuelle Arbeitsdienste gibt!

relational

[DES create-tierwelt.sql](#)

[im entsprechenden Ordner zu finden]

[DES insert-tierwelt.sql](#)

[im entsprechenden Ordner zu finden]

[DES tierwelt SQL-Abfragen.sql](#)

[im entsprechenden Ordner zu finden]

[DES tierwelt RA-Abfragen.ra](#)

[im entsprechenden Ordner zu finden]

sql

[create-tierwelt.sql](#)

[im entsprechenden Ordner zu finden]

[drop-tierwelt.sql](#)

[im entsprechenden Ordner zu finden]

[grant-tierwelt.sql](#)

[im entsprechenden Ordner zu finden]

[insert-tierwelt.sql](#)

[im entsprechenden Ordner zu finden]

[insert-tierwelt_dienstestrang.sql](#)

[im entsprechenden Ordner zu finden]

[insert-tierwelt_tierstrang.sql](#)

[im entsprechenden Ordner zu finden]

[insert-tierwelt_verbrauchsmittelstrang.sql](#)

[im entsprechenden Ordner zu finden]

[install-tierwelt.sql](#)

[im entsprechenden Ordner zu finden]

[uninstall-tierwelt.sql](#)

[im entsprechenden Ordner zu finden]

[update.sql](#)

[im entsprechenden Ordner zu finden]

[Tierwelt-query.sql](#)

[im entsprechenden Ordner zu finden]

[Tierwelt-query.ans](#)

[im entsprechenden Ordner zu finden]

Didaktischer Kommentar zum Umgang mit dem Datenbankpaket

Lebensweltbezug

Arbeitsgemeinschaften gehören zum normalen Schulleben dazu. Ihre Mitglieder rekrutieren sich im Allgemeinen aus unterschiedlichen Klassen und Klassenstufen, sodass für die Kommunikation wichtiger Informationen die kurzen Wege innerhalb einer Lerngruppe oder eines Klassenraums wegfallen.

- ➔ Was wichtig ist, sollte digital gespeichert, aktualisiert und kommuniziert werden. Hierfür bietet sich die Benutzung einer Datenbank mit externem Zugriff an.

Besonders motivierend könnte es sein, mithilfe des vorliegenden Pakets - oder Teilen davon – als Vorlage eine eigene Datenbank für eine AG an der eigenen Schule zu entwerfen bzw. anzupassen.

Methodisches

Offensichtlich ist es möglich, der Lerngruppe oder Teilgruppen den Bau einer Datenbank zur vorliegenden Miniwelt als Projektaufgabe zu stellen. Zu Bedenken ist hierbei allerdings, dass sämtliche Materialien [vermutlich bis zur Unbrauchbarkeit] anpassungsbedürftig würden, ließe man die Schüler:innen mit abweichenden ER-Modellen und Bezeichnungen weiterarbeiten.

- ➔ Nur ein schrittweises, weniger offenes Vorgehen mit einer Regelmäßigen Rückkehr zu Ankerpunkten bzgl. der Nomenklatur und Struktur der vorliegenden Datenbank erhält die volle Nutzbarkeit des Pakets.
- ➔ Falls absehbar ist, dass die Lerngruppe bei einer solchen Vorgehensweise auf Motivationsprobleme stieße, sollte ein offener Modus gewählt werden, in dem eine Datenbank zu einer ähnlichen Miniwelt [schuleigene AG] entworfen wird, während das vorliegende Paket dazu genutzt wird, Beispiellösungen für universale Problemstellungen zu liefern, die besprochen und ggf. gemeinsam angepasst werden können.

Aufgaben mit Lösungen und kurzem methodisch-didaktischen Kommentar

(11)Zeige alle Reptilienarten (vollständig) an!

RA: Unterarten

DRC: {UA_ID, Bezeichnung, Heimat, Lebensraum, Temperatur, Luftfeuchtigkeit, Groesse, Lebenserwartung, Ernaehrung, Besonderheiten | Unterarten (UA_ID, Bezeichnung, Heimat, Lebensraum, Temperatur, Luftfeuchtigkeit, Groesse, Lebenserwartung, Ernaehrung, Besonderheiten)}

SQL: SELECT * FROM Unterarten;

- ➔ **Didaktischer Hintergrund:** Erlernen der generellen Formulierung für Abfragen einzelner Tabellen

(12)Zeige die Namen aller Tiere an!

RA: $\pi_{\text{Name}}(\text{Tiere})$

DRC: {Name | Tiere (Name)}

SQL: SELECT Name FROM Tiere;

- ➔ **Didaktischer Hintergrund:** Einführung von Projektionen (dem Anzeigen einzelner, erwünschter Spalten) von Abfragen einzelner Tabellen

(13) **Zeige die Name aller weiblichen Tiere an!**

RA: $\pi_{\text{Name}}(\sigma_{\text{Geschlecht} = 'W'}(\text{Tiere}))$

DRC: {Name | Tiere (Name, Geschlecht) \wedge Geschlecht = 'W' }

SQL: SELECT Name FROM Tiere WHERE Geschlecht = 'W' ;

- ➔ **Didaktischer Hintergrund:** Einführung von Selektionen (dem Anzeigen einzelner Zeilen, die bestimmte Bedingungen erfüllen) von Abfragen einzelner Tabellen

(14) **Zeige alle Tiere mit ihren Informationen zur jeweiligen Tierart an!**

RA: Tiere \bowtie Unterarten

DRC: {TI_ID, Name, Geschlecht, Zugangsdatum, Laenge, Gewicht, TR_ID, UA_ID, Bezeichnung, Heimat, Lebensraum, Temperatur, Luftfeuchtigkeit, Groesse, Lebenserwartung, Ernaehrung, Besonderheiten | Tiere (TI_ID, Name, UA_ID, Geschlecht, Zugangsdatum, Laenge, Gewicht, TR_ID) \wedge Unterarten (UA_ID, Bezeichnung, Heimat, Lebensraum, Temperatur, Luftfeuchtigkeit, Groesse, Lebenserwartung, Ernaehrung, Besonderheiten)}

SQL: SELECT * FROM Tiere NATURAL JOIN Unterarten;

- ➔ **Didaktischer Hintergrund:** Einführung von Natural Joins (der inhaltlichen Kombination) für Abfragen mehrerer Tabellen

(15) **Zeige die Namen aller Tiere an, die zu den „Bartagame“ gehören!**

RA: $\pi_{\text{Name}}(\text{Tiere} \bowtie (\sigma_{\text{Bezeichnung} = 'Bartagame'}(\text{Unterarten})))$

DRC: {Name | Tiere (Name, UA_ID) \wedge Unterarten (UA_ID, Bezeichnung) \wedge Bezeichnung = 'Bartagame' }

SQL: SELECT Name FROM Tiere NATURAL JOIN
(SELECT * FROM Unterarten WHERE Bezeichnung = 'Bartagame') AS xyz;

- ➔ **Didaktischer Hintergrund:** Kombination erlernter Operationen, hier Natural Join, Projektion und Selektion

(16) **Zeige die Namen aller Tiere an, welche in einem Terrarium des Typs „Wüste“ leben!**

RA: $\pi_{\text{Name}}(\text{Tiere} \bowtie \text{Terrarien} \bowtie (\sigma_{\text{Bezeichnung} = 'Wüste'}(\text{Habitattypen})))$

DRC: {Name | Tiere (Name, TR_ID) \wedge Terrarien (TR_ID, HT_ID) \wedge Habitattypen (HT_ID, Bezeichnung) \wedge Bezeichnung = 'Wüste' }

SQL: SELECT Name FROM Tiere NATURAL JOIN Terrarien NATURAL JOIN
(SELECT * FROM Habitattypen WHERE Bezeichnung = 'Wüste') AS xyz;

- ➔ **Didaktischer Hintergrund:** Verkettung von Natural Joins für komplexere Abfragen

(17)Zeige an, wann „Exo“ zunächst gefüttert werden muss!

RA: $\pi_{\text{Faelligkeit}}(\text{Tierpflege} \bowtie \text{Arbeitsdienste} \bowtie (\sigma_{\text{Bezeichnung} = \text{'Fütterung*'}}(\text{Arbeitsdiensttypen})) \bowtie (\sigma_{\text{Name} = \text{'Exo'}}(\text{Tiere})))$

DRC: {Faelligkeit | Arbeitsdienste (Faelligkeit,AD_ID,AT_ID) \wedge Arbeitsdiensttypen(AT_ID,Bezeichnung) \wedge Tierpflege(AD_ID,TI_ID) \wedge Tiere(TI_ID,Name) \wedge Bezeichnung = 'Fütterung*' \wedge Name = 'Exo'}

SQL: SELECT Faelligkeit FROM Tierpflege NATURAL JOIN Arbeitsdienste NATURAL JOIN (SELECT AT_ID FROM Arbeitsdiensttypen WHERE Bezeichnung LIKE 'Fütterung%') AS x NATURAL JOIN (SELECT TI_ID FROM Tiere WHERE Name = 'Exo') AS y;

[Anmerkung: PSQL liefert ab einer gewissen Anzahl von Attributen in den Ergebnismenge keine Antworten mehr. Daher wurden die Unterabfragen bei SQL auf die wesentlichen Attribute beschränkt, um eine funktionierende Abfrage zu erhalten.]

➔ **Didaktischer Hintergrund:** Weitere Verkomplizierung der inhaltlichen Komplexität; Einführung des „Like“-Operators

(18)Zeige die ID, Namen und Beschreibungen aller Arbeitsdienste an, welche diese Woche fällig werden!

RA: $\pi_{\text{AD_ID, Bezeichnung, Beschreibung}}(\sigma_{\text{Faelligkeit(Woche) = Heute(Woche)}}(\text{Arbeitsdienste}) \bowtie \text{Arbeitsdiensttypen})$

DRC: {AD_ID,Bezeichnung, Beschreibung | Arbeitsdiensttypen(Bezeichnung,Beschreibung,AT_ID) \wedge Arbeitsdienste(AT_ID,Faelligkeit) \wedge Faelligkeit(Woche) = Heute(Woche)}

SQL: SELECT AD_ID, Bezeichnung, Beschreibung FROM Arbeitsdiensttypen NATURAL JOIN (SELECT * FROM Arbeitsdienste WHERE DATE_PART('week', Faelligkeit) = DATE_PART('week', CURRENT_DATE)) AS xyz;

➔ **Didaktischer Hintergrund:** Arbeiten mit Datumsformaten

(19)Zeige an, wann bei „Exo“ das nächste mal das Wasser gewechselt werden muss!

RA: $\pi_{\text{Faelligkeit}}(\sigma_{\text{Name} = \text{'Exo'}}(\text{Tiere}) \bowtie \text{Terrarien} \bowtie \text{Verbrauch} \bowtie \text{Arbeitsdienste} \bowtie (\sigma_{\text{Bezeichnung} = \text{'Frischwasser'}}(\text{Arbeitsdiensttypen}))$

DRC: {Faelligkeit | Arbeitsdienste(Faelligkeit,AT_ID,AD_ID) \wedge Arbeitsdiensttypen(Bezeichnung,AT_ID) \wedge Verbrauch(AD_ID,TR_ID) \wedge Terrarien(TR_ID) \wedge Tiere(Name,TR_ID) \wedge Name = 'Exo' \wedge Bezeichnung = 'Frischwasser'}

SQL: SELECT Faelligkeit FROM (SELECT TR_ID FROM Tiere WHERE Name = 'Exo') AS x NATURAL JOIN Terrarien NATURAL JOIN Verbrauch NATURAL JOIN Arbeitsdienste NATURAL JOIN (SELECT AT_ID FROM Arbeitsdiensttypen WHERE Bezeichnung = 'Frischwasser') AS y;

[Anmerkung: PSQL liefert ab einer gewissen Anzahl von Attributen in den Ergebnismenge keine Antworten mehr. Daher wurden die Unterabfragen bei SQL auf die wesentlichen Attribute beschränkt, um eine funktionierende Abfrage zu erhalten.]

- ➔ **Didaktischer Hintergrund:** Weitere Vertiefung und Anwendung bisher geübter Abfragen und komplexer werdender Zusammenhänge

(20) **Zeige alle Terrarien an, für welche es mindestens zwei aktuelle Arbeitsdienste gibt!**

RA: $Q1 = \pi_{TR_ID, AD_ID}(\text{Verbrauch}) \cup \pi_{TR_ID, AD_ID}(\text{Terrariumpflege})$

$Q2 = \rho_{TR_ID2, A_ID2}(Q1)$

$Q3 = Q1 \bowtie_{TR_ID = TR_ID2 \wedge AD_ID \neq AD_ID2} Q2$

$\text{Terrarien} \bowtie \pi_{TR_ID}(Q3)$

DRC: $Q1 = \{TR_ID, AD_ID \mid \text{Verbrauch}(TR_ID, AD_ID) \vee \text{Arbeitsdienste}(TR_ID, AD_ID)\}$

$Q2 = \{TR_ID, AD_ID \mid Q1\{TR_ID, AD_ID\} \wedge Q1\{TR_ID, AD_ID2\} \wedge AD_ID \neq AD_ID2\}$

$\{TR_ID, Name, Beschreibung, Raumnummer, HT_ID, Hoehe, Tiefe, Breite, Breite_vorn \mid$

$\text{Terrarien}(TR_ID, Name, Beschreibung, Raumnummer, HT_ID, Hoehe, Tiefe, Breite, Breite_vorn)$

$\wedge Q2(TR_ID)\}$

SQL: `SELECT * FROM Terrarien NATURAL JOIN (SELECT TR_ID, COUNT(*) FROM`

`(SELECT TR_ID, AD_ID FROM Verbrauch`

`UNION SELECT TR_ID, AD_ID FROM Terrariumpflege) AS x`

`GROUP BY TR_ID`

`HAVING COUNT(*) > 1) as y;`

- ➔ **Didaktischer Hintergrund:** Einführung von Group By, Having und Count(); Nutzen von Umbenennung und Mehrfachnutzung von Zwischenergebnissen

view

[Tierwelt-view.txt](#)

Beschreibung und Motivation des Views

Verwendung von php

Wir haben uns für die Implementierung mit php entschieden, da go an den berliner Schulen noch nicht verbreitet genug ist.

Anwendungen

Es ist möglich,

- sich einen Überblick über die Tabellen der Datenbank zu machen.
- sich über die ausstehenden Dienste innerhalb eines wählbaren Datumsbereichs zu informieren [geordnet nach den Dienstbereichen Tierpflege, Terrariumpflege und Lagerpflege].
- einen Dienst als erledigt zu kennzeichnen.
- einen Dienst zu verschieben.
- neue Datenbankeinträge zu machen.

[classFormular.php](#)

[im Unterordner tierwelt-htdocs zu finden]

[classOverview.php](#)

[im Unterordner tierwelt-htdocs zu finden]

[classTasks.php](#)

[im Unterordner tierwelt-htdocs zu finden]

[index.php](#)

[im Unterordner tierwelt-htdocs zu finden]

[jQuery_valid.php](#)

[im Unterordner tierwelt-htdocs zu finden]

[showChange.php](#)

[im Unterordner tierwelt-htdocs zu finden]

[showFormular.php](#)

[im Unterordner tierwelt-htdocs zu finden]

[showOverview.php](#)

[im Unterordner tierwelt-htdocs zu finden]

[showTasks.php](#)

[im Unterordner tierwelt-htdocs zu finden]
