

F. Unterrichtsbezogenes Datenbankpraktikum

Die im unterrichtsbezogenen Datenbankpraktikum (UDBP) zu erbringende Leistung stellt den Bewertungsteil des zweiten Semesters der Vorlesung zu Datenbanksystemen dar. Er teilt sich auf in eine schriftliche Dokumentation der Praktikumsergebnisse sowie eine kurze (max. 30 Minuten) Vorstellung der Miniwelt, einiger SQL-Anfragen und des programmierpraktischen Teils des Datenbankpraktikums zusammen, die im Verhältnis 3 zu 1 in die Bewertung eingehen. Das Praktikum wird in kleinen Gruppen durchgeführt (2-4 Personen), idealerweise zu dritt.

Aufgabenstellung:	01.03.2021
Praktikumsbeginn:	15.03.2021
Abgabe der Ausarbeitung:	07.06.2021
Präsentation:	14.06.2021

Eine Abgabe nach dem angegebenen Abgabetermin führt zur Bewertung *nicht bestanden*! Es muss zudem eine Zuordnung einzelner Teile der schriftlichen Ausarbeitung zu den Gruppenmitgliedern möglich und die Vorstellung möglichst gleichmäßig auf alle Gruppenmitglieder verteilt sein.



Damit die Orientierung in fremdem Material später für alle Gruppenteilnehmer schnell und einfach möglich ist, ist es wichtig, dass Sie sich beim Zusammenstellen der Dokumente an eine einheitliche Ordnerstruktur für Ihre Ausarbeitung halten (siehe Seite F-5). Sinnvoll ist die Erstellung eines Git-Archivs.

F.1. Einordnung

DBS-Miniwelten modellieren den Kern eines größeren realen Anwendungssystems, das wegen seiner Komplexität aus informatischer Sicht nicht im Unterricht zugänglich ist¹⁰⁵. Die Modellierung der Miniwelt kann als ein Beispiel und Leitlinie bei der Einführung und Erarbeitung von fachlichen Konzepten des Themenkreises „Datenbanksysteme“ im Unterricht dienen.

Die Erfahrung lehrt, dass der Informatiklehrer viele schöne Projektideen aus reinem Zeitmangel liegen lassen muss. Die Praktikumsaufgabe setzt hier an, indem sie die zeittressenden Vorbereitungsaufgaben für die Durchführung eines Projekts erledigt. Die Umsetzung in den Unterricht wird durch die Herstellung einer guten und vollständigen Dokumentation erleichtert. Eine konkrete Umsetzung in Unterrichtsplanung (schülergerecht aufbereitetes Material, Arbeitsblätter, Tests, ...) gehört nicht mehr zur Aufgabenstellung.

Modellierung

Die Kunst bei der Erfindung einer geeigneten Miniwelt liegt unter anderem darin,

- Welten zu finden, die dem Erfahrungsschatz der Schülerinnen und Schüler nahestehen oder zumindest mit geringem Aufwand erfasst werden können,

¹⁰⁵ Was nicht heißt, dass einige Kollegen nicht doch auf den Gedanken kommen, das „große reale System“ aus Anwendersicht im Unterricht zu behandeln. Das ist möglich, kostet viel Zeit, kann erstaunliche Einsichten erzeugen, der Nutzen für die Realisierung tiefergehender informatischer Lernziele ist jedoch begrenzt.

- eine knappe, präzise textuelle Beschreibung der Miniwelt zu entwerfen,
- klar zu machen, welcher Detaillierungsgrad erreicht werden soll,
- zu definieren, welche Ecken der Miniwelt nur lose oder gar nicht spezifiziert werden,
- eine Modellierbarkeit im ER-Modell mit 5-7 Entitäten zu gewährleisten,
- eine kleine, aber 'artenreiche' Population der Miniwelt zu generieren
- und eine spezielle Sicht auf die Daten zu motivieren, die modular mit Go oder PHP realisierbar ist.

F.2. Teilaufgaben

- (a) Präzise textuelle Beschreibung des zu modellierenden Weltausschnitts (ohne Vorgriff auf Entwurfsentscheidungen)
 - (b) Erstellung eines ER-Modells mit vollständiger Attributierung, Schlüsselangabe und Komplexitäten
 - (c) Angabe der verwendeten Transformationsregeln
 - (d) Transformation des ER-Modells in eine Menge von Relationen (Schlüssel, Fremdschlüssel, abstrakte Wertebereiche für Attribute)
 - (e) Angabe von statischen und dynamischen Integritätsbedingungen
 - (f) Angabe von funktionalen Abhängigkeiten
 - (g) Konkrete Umsetzung des Relationenentwurfs in einen Datenentwurf für die Implementierung in SQL (postgresql),
 - (i) Umsetzung des textuellen Relationenentwurfs in DDL und DML
 - (ii) begründete Wahl besonderer Attributdatentypen
 - (iii) Umsetzung der Integritätsbedingungen, wo es möglich ist
 - (h) Entwurf einer Beispielpopulation der Tabellen: realitätsnah, variationsreich, aber klein
 - (i) Konstruktion einer Aufgabensequenz von einfachen bis zu schwierigen Anfragen in
 - (i) Relationenalgebra,
 - (ii) TRC und DRC (optional),
 - (iii) Datalog (optional)
 - (iv) und SQL.
- Eine Reihe von Aufgabenstellungen werden nur in SQL sinnvoll sein (Aggregatfunktionen, Gruppierung etc.), so dass für die anderen Anfragesprachen eine kleine(!) Auswahl getroffen werden kann.
- (j) Erstellung von Musterlösungen zu diesen Anfragen
 - (k) Dokumentation der Anfragergebnisse
 - (l) ggf. Alternativlösungen
 - (m) Auswahl und Implementierung einer Sicht auf die erstellten Daten in Form einer interaktiven Anwendung (in Go oder webbasiert in PHP), dazu
 - (i) Begründung und Motivation der Sicht
 - (ii) Beschreibung der interaktiven Basismöglichkeiten
 - (iii) Beschreibung optionaler Möglichkeiten
 - (iv) Erstellung einer datenbankgestützten, interaktiven Anwendung in Go oder PHP auf der Basis der erstellten Daten

Anforderung an die Miniwelten

Die Konstruktion eigener Miniwelten nach den bereits genannten Kriterien ist ausdrücklich erwünscht, sollte aber bitte **vor** Beginn der Arbeiten mit der Projektleitung abgesprochen werden. Bei der verbalen Formulierung kann man sich hinsichtlich des Umfangs und der Detailgenauigkeit an den vielen Beispielen der Vorlesung leiten lassen. Es sei noch einmal darauf hingewiesen, dass die Miniwelt der Erfahrungswelt der Schülerinnen und Schüler möglichst nahe stehen sollte.

Anmerkungen zur Bearbeitung

Es geht *nicht* um ein möglichst realitätsnahes Modell, vielmehr soll von zu vielen Details abgesehen werden, um ein möglichst übersichtliches (aber auch nicht *zu* kleines) Modell zu erhalten. Der Schwerpunkt der Entwicklungsarbeit und ein großer zeitlicher Aufwand geht in die Konstruktion der Aufgabensequenz (Anfragen, Lösungen, Ergebnisse). Angestrebt ist eine graduell ansteigende Schwierigkeit der Aufgaben mit breiter Vielfalt in den Lösungsmethoden. Schön wäre, wenn die meisten Anfragen jeweils genau einen neuen Aspekt gegenüber den vorherigen Anfragen demonstrierten. Besonders wichtig ist die Entwicklungsarbeit zum SQL-Teil: die Ausarbeitung soll möglichst viele Konzepte von SQL demonstrieren.

Auch der programmiertechnische Teil (GO oder PHP) des UDBP soll explizit nicht jedes mögliche in der Realität vorkommende Detail abbilden bzw. fordern. Es können allerdings punktuelle Verfeinerungen als Zusatzaufgabe vergeben werden. Dieser Teil liefert für den Unterricht die Möglichkeit, weitere binnendifferenzierte Aufgaben zu stellen. Wichtig ist bei der Konzeption eine möglichst modulare Struktur der Anwendungsschicht zu erreichen, so dass Schülerinnen und Schüler in jedem Fall zu einem fertigen Produkt kommen können. Wichtig im Web-Zusammenhang ist eine saubere Trennung von Design und Struktur (sollten CSS-Dateien verwendet werden), der datenbankunabhängige Zugriff mittels PDO sowie eine übersichtliche und gut dokumentierte Implementierung.

F.3. Schriftliche Ausarbeitung

Die Ausarbeitung des Berichts sollte folgende Punkte umfassen:¹⁰⁶

Modellierung

- (a) Knappe, präzise textuelle Beschreibung der Miniwelt in einfachen Aussagesätzen. Dokument: <db>.txt, eine halbe bis eine Seite Text, ASCII-Format.
- (b) Das Entity-Relationship-Modell unter Angabe aller Attribute und der Beziehungskomplexitäten. Achten Sie darauf, dass unter den Attributen mindestens ein numerisches ist, so dass Aggregierungsanfragen möglich werden.
Dokument: klassische Notation, übersichtliche Handzeichnung, ggf. Scan davon in Form einer Datei <db>-ERModell.<xxx> (mit xxx = tex, png, pdf, eps) beispielsweise.

¹⁰⁶ Alle Textdateien dabei bitte in UTF-8-Kodierung.

- (c) Verbale Angabe von semantischen Bezügen, die sich im Modell nicht wiederfinden, durch Prädikate (Constraints, Integritätsbedingungen). Dokument: knapp formulierte Textprädikate in der Sprache des Modells (`<db>-Semantik.txt`).
- (d) Eine Angabe der im Modell vorkommenden funktionalen Abhängigkeiten in einer Datei `<db>-FAen.txt`.
- (e) Ein Relationenmodell dazu in dritter Normalform mit Angabe von Schlüsseln, Fremdschlüsseln und Notation von Integritätsbedingungen als logische Ausdrücke in einer Datei `<db>-RelMod.txt`.
- (f) Verbale Kurzbeschreibung des Transformationswegs unter Angabe der benutzten Transformationsregeln (`<db>-Trafo.txt`).
- (g) Eine kleine, gut überlegte Sammlung von weltnahen Beispieldaten zu allen Relationen in der Datenbank `<db>`. Die verwendeten Namen sollen plausibel, realitätsnah und paradigmatisch für eine größere Datenmenge sein. Die Auflistung der Daten sollte auf einer DIN-A4-Seite möglich sein (Format: `<db>-Daten.txt` und `<db>-Daten.csv` in Tabellenform).
- (h) Zur Konstruktion der Anfragen: Anordnung nach steigendem Schwierigkeitsgrad. Jede Anfrage soll möglichst eine neue Komplexitätsstufe, Methode oder Implementierungstechnik demonstrieren. Insbesondere sollen auch Anfragen mit Negation, All- oder Existenzaussagen, Aggregatfunktionen etc. vertreten sein. *Dies ist der schwierigste und kreativste Teil.*
Mit anderen Worten: der Konstruktionsprozess der Anfragen orientiert sich weniger an der Semantik des Miniweltmodells, sondern viel mehr an der Demonstration der jeweiligen Sprachmittel.
Der Entwicklungs- und Ausleseprozess der Anfragen wird eng mit den Implementierungen gekoppelt sein, mindestens 20 Anfragen sollten es werden.
Dokumentationsformat: `<db>-query<nn>.txt`, eine zusammenfassende Textdatei mit allen Anfragen in fortlaufender Nummerierung.

Relationenalgebra mit DES

Im Einzelnen sind gesucht:

- (a) Gegebenenfalls eine Datei `<db>-schema.ra` mit dem Relationenschema zu `<db>` sowie `<db>-<rel>.ra`-Dateien mit den Daten für die einzelnen Relationen des Schemas, falls nicht die ODBC-Verbindung zu PostgreSQL genutzt werden kann.
- (b) Die Implementierung der Anfragen in einer kommentierten ra-Datei `<db>-query.ra`. Nicht alle Anfragen werden im SQL-fernen Kontext sinnvoll sein (solche mit Aggregatfunktionen z. B.), sie können weggelassen werden. Die Aufgabenstellung steht jeweils als Kommentar über der auswertbaren Anfrage.
- (c) Das Protokoll aller Anfrageergebnisse in einer Datei `<db>-query.ans`. Das geht z. B. mit einer einfachen Funktion, die alle Anfragen sequentiell auswertet und dabei die Ausgabe mit `&>` in eine Datei umlenkt.

Implementierung im Datenbanksystem (PostgreSQL)

Im Einzelnen sind gesucht:

- (a) *Der DDL-Teil*: eine dokumentierte Datei `Create-<db>.sql` zur Anlage der Tabellenstruktur einer Datenbank (oder eines Schemas) `<db>` in PostgreSQL.
- (b) Eine dokumentierte Datei `Drop-<db>.sql` zur Beseitigung der Datenbank `<db>`.
- (c) *Die Daten*: zwei Dateien `Insert-<db>-data.sql` und `Delete-<db>-data.sql`, die die Probedaten mit generischen SQL-Befehlen (`INSERT`, `DELETE`) in `<db>` einfügen bzw. aus ihr löschen. Bitte hierzu keine PostgreSQL-spezifischen Verfahren wählen.
- (d) *Die Anfragen*: Die Implementierung aller Anfragen (numeriert!) als `<db>-query.sql`. Die Aufgabe steht jeweils als Kommentar über der ausführbaren Anfrage. Oft sind Alternativ-Implementierungen möglich, deren Varianten in einer Datei mit dem Namen `<db>-query-alt.sql` angegeben werden können. Ein begleitender didaktisch-methodischer Kommentar zu den Anfragen und den Lösungsvarianten ist in der zusammenfassenden Ausarbeitung erwünscht.
- (e) Die Protokollierung der Anfrageergebnisse in den Dateien `<db>-query<nn>.ans` mittels `psql -o` oder mittels des Schalters `o <datei>`.

Didaktisch-Methodisches

Würdigen Sie unter den Aspekten Schwierigkeitsgrad, Umfang, Zeitbedarf und Bedeutsamkeit im Kontext des Informatikunterrichts ihre bis hierher geleistete Arbeit. Welche Teile sind unverzichtbar, welche ggf. fakultativ. Haben Sie Ideen zur methodischen Umsetzung? Was wären in einer Weiterführung sinnvolle Fragestellungen? Die Antworten gehen in eine Datei `<db>-did.txt`

Ausarbeitung

Erwünscht ist eine zusammenfassende Darstellung Ihrer Praktikumsausarbeitung als PDF-Dokument. Alle Texte, Aufgaben und Lösungen sollen überdies in einer *logischen Struktur* bereitgestellt werden, die es leicht macht, für Unterrichtszwecke geeignete Teile zu übernehmen. Mit dieser Struktur ergibt sich auch ein Arbeitsplan für die Lösung der Praktikumsaufgabe.

Die textliche Ausarbeitung steht in den Dokumenten `<db>.txt` (formatiertes ASCII, UTF8), ggf. \LaTeX -Format `<db>.tex` (`<db>.odt` oder `<db>.doc` geht auch) und einer Druckfassung `<db>.pdf`.

Stellen Sie alle praktikumsergebnisse in einer Archiv-Datei `<db>.tgz` mit folgender Binnenstruktur zusammen:

```
<db>
|-- doc                                (A) Zusammenfassende Ausarbeitung
|   |-- <db>.txt                        ASCII-Text
|   |-- <db>.tex                        optional :-)
|   |-- <db>.odt                        ggf.
|   |-- <db>.doc[x]                     ggf.
|   '-- <db>.pdf                        in jedem Fall
|
|-- modell                             (B) Modelldokumente
|   |-- <db>-Welt.txt
|   |-- <db>-ERModell.png               (oder .tex, .pdf, .eps, .jpg, .svg)
|   |-- <db>-Semantik.txt
|   |-- <db>-FAen.txt
|   '-- <db>-RelMod.txt
```

```
| |-- <db>-Trafo.txt
| |-- <db>-Daten.txt          Formatierte Tabellen ASCII
| |-- <db>-<rel>.csv          TAB- oder ';' -getrennte Daten (ggf.)
| |-- <db>-query<nn>.txt      Text zu Anfrage <nn>
| '-- <db>-Anfragen.txt       Alle Anfragen nummeriert
|
|-- relational                (C) RALG/TRC/DRC/DL (DES)
| |-- <db>-query.ra           Alle RALG-Anfragen mit Kommentar
| |-- <db>-query.trc          Alle Tupelkalkül-Anfragen mit Kommentar
| |-- <db>-query.drc          Alle Domänenkalkül-Anfragen mit Kommentar
| |-- <db>-query.dl           Alle Datalog-Anfragen mit Kommentar
| |-- <db>-query-ra.ans       Alle RALG-Anfrageergebnisse
| |-- <db>-query-trc.ans      Alle Tupelkalkül-Anfrageergebnisse
| |-- <db>-query-drc.ans      Alle Domänenkalkül-Anfrageergebnisse
| '-- <db>-query-dl.ans       Alle Datalog-Anfrageergebnisse
|
|-- sql                       (D) SQL (PostgreSQL)
| |-- Create-<db>.sql
| |-- Drop-<db>.sql
| |-- Insert-<db>-data.sql
| |-- Delete-<db>-data.sql
| |-- <db>-query.sql          Alle Anfragen mit Kommentar
| |-- <db>-query-alt.sql      ggf. Varianten
| |-- <db>-query.ans          Anfrageergebnisse
| '-- <db>-query-alt.ans      Anfrageergebnisse der Varianten
|
|-- did                       (E) Didaktisch-Methodisches
| |-- <db>-did.txt            Didaktisch-methodische Hinweise
| '-- ...                     Weiteres Material (Arbeitsblätter, ...)
|
'-- view                      (F) Implementierung eines Views in Go/PHP (Auswahl)
| |-- <db>-view.txt           Beschreibung und Motivation des Views
| |-- <db>-basics.txt         Basismöglichkeiten der Anwendung
| |-- <db>-extended.txt       erweiterte Möglichkeiten
| |-- <db>-htdocs             Webserver-Verzeichnisinhalt (nur PHP)
| |-- index.html/php          Startdatei im Webserververzeichnis
| |-- conn-inc.php            Include-Datei für Datenverbindung (ggf.)
| |-- <db>.css                CSS-Datei (ggf.)
| '-- ...
|-- <db>-go                   Go-Dateien (nur Go)
| |-- module                  verwendete externe Go-Module (ggf.)
| |-- <mod1>
| '-- ...
'-- <db>.go                   Hauptprogramm-Datei
```

Präsentation

Bei der Präsentation wird eine kurze Einführung in die modellierte Miniwelt erwartet, wobei Entitäten und Beziehungen mit ihren Komplexitäten anhand eines ER-Diagramms vorzustellen sind. Die Relevanz des Weltausschnittes für SuS muss deutlich werden. Anschließend ist eine kleine Auswahl an Anfragen an das Relationenalgebra- bzw. SQL-System vorzuführen sowie der programmiertechnische Teil vorzustellen. Dabei soll insbesondere die Realitätsnähe des gewählten Views erläutert werden. Die Präsentation soll eine Dauer von 30 Minuten nicht überschreiten.

F.4. Bewertung des UDBP

Kandidat A: _____

Kandidat C: _____

Kandidat B: _____

Kandidat D: _____

Ermittlung des Gesamtergebnisses

Jeder Kandidat muss erkennbar an **mindestens einem Bewertungskriterium der Präsentation** (Darstellung der Miniwelt, Anfragen in SQL und im Relationenmodell, Anwendungsprogramm) und an **mindestens zwei Bewertungskriterien der schriftlichen Ausarbeitung** (Qualität der Miniwelt, Modellierung, Entwurf und Population, Güte der Anfragen (SQL/RA) und Anwendungsprogramm) mitgewirkt haben. Für diese Kriterien erfolgt eine individuelle Bewertung. Zusammen mit der Bewertung der allgemeinen Aspekte ergeben sich daraus die individuellen Punktsommen und daraus die Anteile a_P für die Präsentation und a_A für die schriftliche Ausarbeitung. Diese Anteile ergeben im Verhältnis 1 zu 3 den Gesamtanteil a .

$$a = \frac{1}{4} \cdot a_P + \frac{3}{4} \cdot a_A$$

Die Zuordnung des Gesamtanteils a zur Note erfolgt nach dem in der gymnasialen Oberstufe üblichen Schlüssel. Bei $a < 45\%$ gilt das Datenbankpraktikum als nicht bestanden.

Bedeutung und Bewertung der Merkmalsausprägungen

Symbol	textliche Beschreibung	Punkte
++	sehr ausgeprägt	5
+	ausgeprägt	4
+ / o	im Allgemeinen ausgeprägt	3

Symbol	textliche Beschreibung	Punkte
o / -	teilweise ausgeprägt	2
-	kaum ausgeprägt	1
--	nicht vorhanden	0

Individuelle Gesamtergebnisse

	A	B	C	D
erreichter Anteil a_P für die Präsentation				
erreichter Anteil a_A für die schriftliche Ausarbeitung				
erreichter Anteil a für das Datenbankpraktikum				
erreichte Note				

Bewertungsraster

Präsentation (25%)		++	+	+/-	o/-	-	--
Darstellung der Miniwelt (Klarheit der Darstellung, Relevanz des Weltausschnittes, Übersichtlichkeit des ER-Modells, Darstellung der Transformation ins Relationenmodell)	A						
	B						
	C						
	D						
Anfragen in SQL und im Relationenmodell (Korrektheit, Vielfalt und Progression der Auswahl, Darstellung von alternativen Lösungen)	A						
	B						
	C						
	D						
Anwendungsprogramm (PHP oder Go) (Korrektheit der Ausführung, Erläuterung der Funktion, Realitätsnähe des realisierten Views)	A						
	B						
	C						
	D						
Allgemeine Aspekte (Einhalten der zeitlichen Vorgaben, Gleichverteilung der Aufgaben, Qualität der Präsentation)							
schriftliche Ausarbeitung (75%)		++	+	+/-	o/-	-	--
Qualität der Miniwelt (Relevanz, Beschreibung, angemessene Komplexität)	A						
	B						
	C						
	D						
Modellierung (ER-Modell, Transformation ins RA-Modell, Begründung der Entwurfsentscheidungen)	A						
	B						
	C						
	D						
Entwurf und Population (DDL-Qualität, Berücksichtigung von Integritätsbedingungen, Güte der Population)	A						
	B						
	C						
	D						
Güte der Anfragen (SQL/RA) (Progression, Vielfalt und Motivation, Alternativen, Korrektheit und Vollständigkeit, Ausnutzen von SQL-Spezifika)	A						
	B						
	C						
	D						
Anwendungsprogramm (Lauffähigkeit, Fehlertoleranz, Übersichtlichkeit und Wiederverwendbarkeit des Codes, Benutzerfreundlichkeit, Modularisierung für Unterrichtszwecke)	A						
	B						
	C						
	D						
Allgemeine Aspekte (Vollständigkeit und Übersichtlichkeit der Dokumentation, Einhalten der vorgegebenen Binnenstruktur, Erleichterung der Fremdnutzung des Projektes)							