Klausur zur LV 36 669a: Daten effizient speichern und verarbeiten

Allgemeine Hinweise:

1. **Bearbeitungszeit: 150 Minuten**, maximal erreichbare **Punktzahl: 94**. Die jeweils erreichbare Punktzahl ist bei jeder Frage angegeben. Bitte teilen Sie Ihre Arbeitszeit entsprechend ein.

2. **Geben Sie alle Ihre Lösungen gesammelt in einer pdf-Datei ab.**

3. Denken Sie daran, Ihre **Daten (Vor-/Nachname, Matrikelnummer, Studienfächer, Fachsemester)** in die Abgabedatei einzutragen, **bevor** Sie mit der Bearbeitung beginnen.

4. Vergessen Sie nicht, Ihrer Abgabe eine **Eidesstattliche Erklärung** hinzuzufügen (Template siehe Angabenordner)!

5. Zugelassene Hilfsmittel: **Vorlesungsskript/Internet, Gruppenarbeit ist NICHT ZULÄSSIG**

6. Geben Sie keine mehrdeutigen (oder mehrere) Lösungen an. In solchen Fällen wird stets die Lösung mit der geringeren Punktzahl gewertet. Eine richtige und eine falsche Lösung ergeben also 0 Punkte.

.

Aufgabe 1 - Schema-on-Write vs. Schema-on-Read (2 Punkte)

Welche Situation könnte eine:n technische:n Entscheider:in dazu veranlassen, auf Schema-on-Read zu setzen und nicht auf Schema-on-Write? Und welche Umstände oder Anforderungen könnten zu einer gegenteiligen Entscheidung führen?

Aufgabe 2 - Datenmodellierung und relationales Schema (34 Punkte)

Aufgrund mangelnder Digitalisierung hinkt ein Land bei der Impfung seiner Bevölkerung gegen die Krankheit Covid-19 hinterher. Dieses Land beauftragt nun ein Unternehmen, in dem Sie arbeiten, eine Software für das Impfmanagement zu entwickeln. Als Informationswissenschaftler:in besitzen Sie profundes Wissen im Bereich der Datenmodellierung und sind damit beauftragt im Rahmen dieses Projektes ein Datenmodell für das zu implementierende Impfmanagementsystem auszuarbeiten. Um Anforderungen zu erheben, führen Sie ein Interview mit dem/der Gesundheitsminister:in des auftraggebenden Landes. Nach der Aufzeichnung des Gesprächs, liegt folgendes Transkript vor:

Gesundheitsminister:in: "Vielen Dank für Ihre Hilfe. Wir sind mit der Logistik total überfordert. Wir müssen einfach anerkennen, dass analoges Arbeiten nicht mehr möglich ist. Wir wollen das Impfmanagement von jetzt an online abwickeln. Können Sie uns da weiterhelfen?"

DB-Spezialist:in: "Sehr gerne. Welche Daten sollen denn gespeichert werden?"

Gesundheitsminister:in: "Zentral sind natürlich die Daten der Bürger:innen, die sich für eine Impfung registrieren wollen. Wir müssen mindestens Ihre Namen, die Anschrift und die Telefonnummer abspeichern."

DB-Spezialist:in: "Ok gut, das ist schon mal ein Anfang. Was sollen Bürger:innen bei der Registrierung im Impfportal noch angeben müssen? Welche Informationen sind für Sie relevant?"

Gesundheitsminister:in: "Entscheidend ist natürlich, wie alt sie sind, ob bestimmte Vorerkrankungen vorliegen oder ob sie in systemrelevanten Berufsgruppen arbeiten. Je nachdem werden sie einer bestimmten Impfgruppe zugeordnet. Also jede:r Bürger:in wird dann genau einer Impfgruppe zugewiesen. Diese Impfgruppen haben dann unterschiedliche Prioritätslevel. Außerdem wollen wir für jede Impfgruppe eine Beschreibung abspeichern, sodass wir nachvollziehen können, welche Personen wir welcher Gruppe zuordnen. Die Priorisierung bzw. Gruppenzugehörigkeit kann sich übrigens im Nachhinein noch ändern."

DB-Spezialist:in: "Sehr gut, das sind schon mal Informationen, mit denen man arbeiten kann. Was jetzt noch wichtig ist, ist der Ablauf der Impfungen. Wie wird das logistisch organisiert? Wir benötigen Infos darüber, da das Auswirkungen darauf hat, in welcher Beziehung die zu speichernden Daten zueinander stehen."

Gesundheitsminister:in: "Klar, das kann ich Ihnen noch erklären. Jede:r Bürger:in wird in einem Impfzentrum von einem Arzt/einer Ärztin mit einem bestimmten Impfstoff zu einem vorher vereinbarten Termin geimpft. Soweit zur allgemeinen Logistik. Nun zum Detail. Die bisher zugelassenen Impfstoffe erfordern zwei Impfungen, bis sich der vollständige Schutz entfaltet. Jedoch werden vermutlich auch

bald Impfstoffe zugelassen, bei denen eine Impfung ausreicht. Das bedeutet, dass Bürger:innen maximal zwei Impfungen bekommen können. Was den Impfstoff betrifft, so müssen wir in unserem Impfmanagementsystem auch festhalten, wie der Name des Impfstoffes lautet. Das ist wichtig wenn es um den zweiten Impftermin geht. Außerdem soll gespeichert werden, wie viele Impfungen bei einem bestimmten Impfstoff nötig sind, bis dieser die volle Wirkung entfalten kann. Bei den Impfstoffen unterscheiden wir drei Arten. mRNA-Impfstoffe, Vektorimpfstoffe und Totimpfstoffe. Bei den mRNA-Impfstoffen wollen wir zusätzlich wissen, welcher Spezialkühlschrank für die Kühlung nötig ist. Bei den Vektorimpfstoffen wollen wir noch festhalten, welche Virusart verwendet wird und beim Totimpfstoff ist der sogenannte *Grad der Aufreinigung* relevant."

DB-Spezialist:in: "Super, das sind schon eine Menge Informationen. Damit werde ich jetzt mal ein erstes ER-Diagramm entwerfen und dann nochmal auf Sie zurückkommen, falls ich weiteren Input benötige."

Gesundheitsminister:in: "Dankeschön!"

1. Erstellen Sie basierend auf diesem Transkript ein EER-Diagramm und ergänzen Sie im Text nicht erwähnte Attribute (pro Entitätstyp mind. zwei neben dem PK/eventuelle Subtypes sind ausgenommen) und Beziehungen sinnvoll. Geben Sie für jede Beziehung sinnvolle Kardinalitäten an. Dabei muss stets die minimale und maximale Kardinalität ersichtlich sein. Verwenden Sie zur Modellierung die im Kurs erlernte Crow's Foot-Notation. (22 Punkte)
2. Reflektieren Sie Ihre Lösung und begründen Sie Designentscheidungen, die dem Transkript nicht explizit zu entnehmen sind. Diskutieren Sie auch die Rolle von schwachen Entitäten in Ihrem Diagramm. Warum haben Sie sie (nicht) verwendet? (6 Punkte)
3. Überführen Sie Ihr Diagramm in ein relationales Schema, das in 3NF vorliegt. Verwenden Sie folgende Notation: (7 Punkte)

   Name_der_Relation (<u>Primärschlüsselattribut</u>, Attribut 1, Attribut 2, <u>Fremdschlüsselattribut</u>)
   oder alternativ:
   Name_der_Relation (Primärschlüsselattribut [PK], Attribut 1, Attribut 2, Fremdschlüsselattribut [FK])

Aufgabe 3 - SQL-Statements (15 Punkte)

In der angehängten Datei *AdventureWorks2008_db_diagram.pdf* sehen Sie ein Schema der *AdventureWorks*-Datenbank. Erstellen Sie mit Hilfe dieses Schemas SQL-Statements. Primär- und Fremdschlüssel werden in der Datei entsprechend als PK und FK bezeichnet. Weitere Notationen, wie U1 (= Unique Index) sind für die Bearbeitung der Aufgabe nicht von Bedeutung.

1. Geben Sie Vor- und Nachname aller im System gespeicherten Personen zurück.

2. Geben Sie Vor- und Nachname aller im System gespeicherten Kunden zurück.

3. Geben Sie für jede:n Angestellte:n (spezifiziert durch Vorname, Nachname und Geburtsdatum) an, wann seine/ihre Schicht beginnt und endet.

4. Geben Sie Vor-, Nachname und Bestelldatum der Kunden aus, die das Produkt "Chair" bestellt haben. (Hinweis: Gehen Sie davon aus, dass BusinessEntityID in SalesOrderHeader ein FK auf CustomerID in Customer ist)

5. Geben Sie Listenpreis und Name des teuersten Produkts aus.

6. Geben Sie die Anzahl der Angestellten aus, die zwischen dem 01.01.1960 und dem 30.03.1960 (beides inklusive) geboren wurden.

7. Geben Sie für jede Produktklasse den Durchschnittslistenpreis aus.

Aufgabe 4 - ACID und DBMS-Technologie (18 Punkte)

Rufen Sie sich den gesamten Ablauf in Erinnerung, wie - ggf. auch sehr komplexe - SQL-Queries abgearbeitet werden, beginnend vom Empfang der Statements, über die mehrstufige Abarbeitung innerhalb des DBMS, bis zur Rückmeldung der Antwort.

4.1 Gesamtbild ACID

Beschreiben Sie, welche Teile der ACID-Garantien wo im Ablauf relevant sind. Welche Probleme könnten dabei entstehen und welche Teile von ACID versichern dann jeweils dort dem Benutzer, dass das Problem eben nicht auftritt? Nennen Sie jeweils auch die verschiedenen Vorgehensweisen oder Verfahren, die das DBMS in seinem Repertoire hat, um die aufgezeigten Probleme zu lösen / zu vermeiden. (12 Punkte)

4.2 Details und Trade-Offs

Wir haben zwei Möglichkeiten kennengelernt, wie ein DBMS verhindern kann, dass sich parallel ausgeführte SQL-Anfragen gegenseitig stören: Locks und Multiversioning (3+3 Punkte).

a) Beschreiben Sie beide Vorgehensweisen.

Der erreichbare Grad an Parallelität hängt von vielen Detail-Abwägungen ab:

b) Locks: Das DBMS kann die bestehenden Daten eher grob- oder eher feingranular mit Locks versehen - wieso ist das ein trade-off?
c) Multiversioning: Wenn das Verhältnis von erzwungenen Rollbacks zu erfolgreichen Commits schlechter wird, ist das DBMS zwar beschäftigt, es werden aber kaum noch SQL-Anfragen erfolgreich beendet. Was könnte das DBMS dagegen unternehmen?

Aufgabe 5 - XML-Schema (8 Punkte)

Im Anhang befindet sich die Datei *plant_catalog.xml.* Erstellen Sie zu dieser Datei eine XML-Schema-Datei, die folgendes leistet:

- Legen Sie überall möglichst präzise Typen fest. Begründen Sie alle Ihre Entscheidungen.
- Wären präzisere als die von Ihnen gewählten Datentypen denkbar und wie könnten diese beschaffen sein?
- Legen Sie die Verschachtelungsstruktur des Dokuments bzw. der Elemente fest. Begründen Sie alle Ihre Entscheidungen.

Aufgabe 6 - XQuery (6 Punkte)

Im Anhang befindet sich die Datei *plant_catalog.xml.* Erstellen Sie folgende XQuery-Abfragen:

1. Geben Sie für jede Pflanze deren botanische Bezeichnung in einem Wurzelknoten `<botanical_names></botanical_names>` mit folgender Struktur aus:
```
<botanical_names>
      <botanical>Sanguinaria canadensis</botanical>
      <botanical>Aquilegia canadensis</botanical>
      ...
</botanical_names>
```
2. Geben Sie den allgemeinen Namen (common) aller Pflanzen aus, die weniger als 5$ kosten. Sammeln Sie Ihre Ergebnisse im Wurzelknoten `<cheap_plants></cheap_plants>` mit folgender Struktur:
```
<cheap_plants>
      <common>Bloodroot</common>
      <common>Hepatica</common>
      ...
</cheap_plants>
```
3. Gruppieren Sie die Pflanzen danach, wie viel Licht sie benötigen. Das Ergebnis Ihres XQuery-Ausdrucks soll folgende Struktur aufweisen:
```
<light_type>
      <mostly_shady>
            <common>Bloodroot</common>
            <common>Columbine</common
      </mostly_shady>
      <mostly_sunny>
            ...
      </mostly_sunny>
      ...

</light_type>
```

Aufgabe 7 - Markup-Sprachen (11 Punkte)

7.1 Datentypen

Nehmen Sie für diese Aufgabe an, dass JSON-Schema noch nicht erfunden wurde. Vergleichen Sie "JSON (ohne JSON-Schema)" mit "XML & XML-Schema" bzgl. der Unterstützung von Datentypen:

a) Welche Möglichkeiten werden für den Benutzer jeweils vordefiniert?
Können eigene Datentypen definiert werden? (3 Punkte)

b) Welche Vor- und Nachteile erwachsen aus den beiden Ansätzen? (2 Punkte)

7.2 Paradigmen

Wir haben drei verschiedene Paradigmen kennengelernt, die eine I/O-Bibliothek für Markupsprachen dem Benutzer zum Lesen und Schreiben von Markup-Dokumenten anbieten kann: Data Binding (z.B. Java Jackson), Streaming (z.B. SAX-Parser) oder das manuelle Bearbeiten der Baumstruktur (z.B. DOM/JDOM).

a) Beschreiben Sie kurz das Grundprinzip aller drei Vorgehensweisen. (3 Punkte)

b) Finden Sie für jeden der drei Ansätze je ein Anwendungsbeispiel und begründen Sie in allen drei Fällen, warum der jeweilige Ansatz dort die beste Wahl wäre. (3 Punkte)