

Fachprüfung 43052 Datenbanken 1
(Wintersemester 2019/20)
21.02.2020

Hinweise

- Prüfen Sie Ihre Klausur auf Vollständigkeit.
- Lassen Sie die Klausur zusammengeheftet.
- Bei der Bearbeitung der Klausur dürfen **keine** Hilfsmittel verwendet werden.
- Der Klausurtext enthält ausreichend Platz zur Lösung der Aufgaben. Sie können auch die Rückseiten der Blätter für Ihre Lösungen nutzen. Sollten Sie trotzdem zusätzliches Papier benötigen, wenden Sie sich an die Klausuraufsicht. Die Nutzung eigenen Papiers ist nicht gestattet.
- Sollte Ihre Lösung nicht unmittelbar unter oder neben der Aufgabenstellung stehen, machen Sie bitte einen entsprechenden Hinweis. Streichen Sie diejenigen Teile der von Ihnen geschriebenen Texte deutlich durch, die nicht in die Bewertung eingehen sollen. Von mehreren angegebenen Lösungen ist die zu Bewertende eindeutig zu kennzeichnen.
- Bei Ihren Lösungen werden außer der korrekten Problemlösung auch die verständliche Formulierung, ausreichende Dokumentation und eine übersichtliche Schreibweise bewertet.
- Bitte schreiben Sie deutlich.

Aufgabe	Max. Punktzahl	Erreichte Punkte
1. Warming up	9	
2. Wissensfragen	15	
3. SQL (Grundlagen)	10	
4. SQL (Abfragen)	25	
5. SQL (Datendefinition und Datenintegrität)	10	
6. SQL (Benutzersichten)	11	
7. Datenbankprogrammierung	20	
Bonuspunkte	10	
Gesamtpunktzahl	100+10	

Viel Erfolg!

Aufgabe 1 Warming Up (9 P)

Beantworten Sie die folgenden Fragen.

- a) Was liefert die Ausführung des folgenden SQL-Statements? Markieren Sie die beiden richtigen Aussagen.

```
SELECT Nachname FROM Kunde
MINUS
SELECT Nachname FROM Kunde WHERE Ort='Bochum'
ORDER BY Nachname
```

- A Die Nachnamen werden in aufsteigender Reihenfolge sortiert angezeigt.
 - B In der ORDER BY Klausel fehlt die Angabe der Sortierreihenfolge.
 - C Es fehlen Klammern, damit sich das ORDER BY auf alle Tupel auswirkt.
 - D Es gibt eine Fehlermeldung, da Tupel nicht eindeutig identifizierbar sind.
 - E Die Abfrage liefert bei MySQL eine Fehlermeldung.
- b) In der Tabelle Kunde sind die Kunden Meitner, Einstein und Curie gespeichert. Was liefert die folgende SQL-Anfrage als Ergebnis? Begründen Sie Ihre Antwort.

```
SELECT Nachname FROM Kunde
WHERE 'M%' LIKE Nachname
```

- c) Eine Integritätsbedingung lautet, dass das Geburtsdatum (Attribut GebDat) in der Vergangenheit liegen muss. Wie kann diese Integritätsbedingung in den angegebenen DBMS umgesetzt werden? Begründen Sie Ihre Antwort.
- a. Access

 - b. Oracle

Aufgabe 2: Wissensfragen (15 P)

a) Erklären Sie kurz das **Transaktionskonzept** und nennen Sie jeweils die Begriffe, die das Wort „**ACID**“ bilden.

b) Erläutern Sie bitte durch 1-2 Sätze die folgenden Codd'sche Regeln.

a. Datenintegration

b. Flexible Operationen

c. Logische Datenunabhängigkeit

- c) Erläutern Sie bitte, wie genau eine **natürliche Verbundoperation** (natural join) durchgeführt wird. Können dabei Null-Werte entstehen und – falls ja – an welchen Stellen?

Aufgabe 3 SQL Grundlagen (10 P)

Es sollen die Kunden gelistet werden, die den Artikel 4812 mehr als 10-mal in ihrem Warenkorb haben. Geben Sie nur die Kundennummern und die Anzahl aus.

a) Geben Sie die zugehörige Anfrage in der relationalen Algebra an.

b) Geben Sie die zugehörige SQL-Anfrage an.

c) Erläutern Sie, was die folgende SQL-Anweisung bewirkt. Beachten Sie die Schemata der jeweiligen Tabellen.

```
INSERT INTO Bestellposition (Bestellnummer, Position, Artikelnummer, Anzahl)
SELECT 5927, ROWNUM, Artikelnummer, Anzahl
FROM Warenkorb WHERE Kundennummer=7562
```

Aufgabe 4 SQL (Abfragen) (25 P)

- a) Listen Sie die Bestellnummer zusammen mit der Anzahl der zugehörigen Bestellpositionen. Gelistet werden sollen nur offene Bestellungen mit mehr als 3 Bestellpositionen.
Nutzen Sie **nur** Verbundoperationen, d.h. keine Unterabfragen und keine Mengenoperationen.
- b) Listen Sie die offenen Bestellungen, die nur Artikel mit einem Preis größer als 100 [Euro] beinhalten. Listen Sie die zugehörigen Bestell- und Kundennummern. Nutzen Sie **nur** Unterabfragen (Subqueries) und Auswahloperatoren, d.h. keine Verbundoperationen und keine Mengenoperationen und keine Aggregatfunktionen.

- c) Listen Sie die Artikel, die nicht bestellt wurden oder die keinen Lagerplatz zugewiesen bekommen haben. Listen Sie die Artikelnummer, die Artikelbezeichnung sowie die jeweilige Ausnahme („keine Bestellung vorhanden“, „kein Lagerplatz vorhanden“). Sortieren Sie die Ergebnismenge aufsteigend nach dem Artikelnamen. Nutzen Sie nur Verbundoperationen und Mengenoperationen, d.h. keine Unterabfragen.

	ARTIKELNUMMER	ARTIKELNAME	AUSNAHME
1	4816	Anatomie-Atlas	keine Bestellung vorhanden
2	4820	Datenbank-Skript	keine Bestellung vorhanden
3	4899	Harry Potter Band 20	kein Lagerplatz vorhanden
4	4899	Harry Potter Band 20	keine Bestellung vorhanden

Aufgabe 5 SQL-DDL und Datenintegrität (10 P)

Gegeben ist folgendes unvollständiges Installationsskript für eine zusätzliche Tabelle **Probe-Abo** zu dem Buchhandlungsschema im Anhang, das Informationen für ein Probe-Abonnement beinhaltet, bei dem eine bestimmte Anzahl Artikel einer festen Kategorie pro Zeitraum geliefert werden.

Gegeben ist folgendes Relationsschema für das Probe-Abo:

ProbeAbo (Abonummer, *Kategorie*, *Kundennummer*, Beginn, Anzahl)

Realisieren Sie die folgenden Einschränkungen / Integritätsbedingungen:

- Abonummer soll Primärschlüssel sein.
- Kategorie muss einem bereits existierenden Kategorie-Kürzel entsprechen
- Kundennummer muss zu einem bereits bekannten Kunden gehören.
- Wird eine Kategorie aus unserer Datenbank gelöscht, so sollen auch alle Probe-Abos zu dieser Kategorie gelöscht werden.
- Der Beginn des Probe-Abos muss ein gültiges Datum sein.
- Die Anzahl der abonnierten Artikel kann nur erhöht werden (z.B. von einem pro Monat auf zwei pro Monat), aber nicht verringert werden.

Hinweis: Geben Sie im Falle von dynamischen Integritätsbedingungen lediglich die entsprechende Lösungsidee an. Eine Implementierung von dynamischen Integritätsbedingungen ist hier nicht nötig.

```
CREATE TABLE `ProbeAbo` (  
    Abonummer int(11),  
    KategorieKRZ CHAR (3Byte),  
    Kundennummer int(11),  
    Beginn date,  
    Anzahl int (2),
```

```
) ;
```


Aufgabe 7 Datenbankprogrammierung (20 P)

Gegeben ist die nicht-direkt änderbare Benutzersicht:

```
CREATE VIEW artikelView
```

```
AS
```

```
SELECT artikelnummer, artikelname, preis, Sum(Lagerbestand) Lagerbestand
```

```
FROM Artikel a JOIN Lager l ON a.artikelnummer=l.anummer
```

```
GROUP BY artikelnummer, artikelname, preis
```

- a) Begründen Sie, weshalb diese Sicht nicht änderbar ist.
- b) Schreiben Sie ein Datenbankprogramm, durch welches Änderungen des Artikelnamens und des Artikelpreises auf dieser Benutzersicht möglich werden.

- c) Schreiben Sie eine Funktion „getLagerbestand“ in Oracle, welche für einen Artikel mit der Artikelnummer an den zugehörigen Lagerbestand (über alle Standorte hinweg) zurückliefert. Wenn kein Lagerplatz zu dem Artikel vorhanden ist, soll eine entsprechende Fehlermeldung geworfen werden. Achten Sie auf eine strukturierte Fehlerbehandlung.

Buchhandlung-Schema:

- Kunde (Kundennummer, Nachname, Vorname, Anrede, Geburtsdatum, Ort)
- Warenkorb (Kundennummer, Artikelnummer, Anzahl)
- Artikel (Artikelnummer, Artikelname, Autor, Preis, Ausgabe, Kategoriekrz)
- Artikelkategorie (Kategoriekrz, Bezeichnung, Oberkategorie, Oberkategoriekrz)
- Lager (Lagernummer, Standort, ANummer, Lagerbestand)
- Bestellung (Bestellnummer, Datum, Kundennummer, Bestellstatus, Versandstatus)
- Bestellposition (Bestellnummer, Position, Artikelnummer, Anzahl, Preis, Reduktion)
- Bestellstatus: offen, bestellt, abgeschlossen, storniert
- Versandstatus: offen, vorbereitet, versandt, retourne

