

Aufgabe 1

Gib das Ergebnis folgender SQL-Anweisungen unter Verwendung der Beispieldaten auf Seite 5 an.

a) <pre>SELECT B.TITEL, BE.EXEMPLAR_NR FROM BUCH B JOIN BUCH_EXEMPLAR BE ON BE.ISBN = B.ISBN WHERE LENGTH (B.TITEL) < 5 ORDER BY B.TITEL;</pre>	Ergebnis HIER eintragen
b) <pre>SELECT BENUTZER_NR, NACHNAME, VORNAME FROM BENUTZER B WHERE NOT EXISTS (SELECT (*) FROM AUSLEIH_VORGANG AV WHERE AV.BENUTZER_NR = B.BENUTZER_NR);</pre>	Ergebnis HIER eintragen
c) <pre>SELECT B.TITEL, COUNT(*) ANZAHL FROM BUCH B JOIN BUCH_EXEMPLAR BE ON BE.ISBN = B.ISBN JOIN AUSLEIH_VORGANG AV ON AV.EXEMPLAR_NR = BE.EXEMPLAR_NR GROUP BY B.TITEL HAVING COUNT(*) < 4 ORDER BY B.TITEL;</pre>	Ergebnis HIER eintragen

Aufgabe 2

Gib SQL Anweisungen zur Durchführung folgender Aktionen (in der u. g. Reihenfolge) für die Beispieldatenbank auf Seite 5 an.

Aufnahme eines Benutzers mit BENUTZER_NR = 12

Aufnahme eines neuen Buchs mit ISBN = 99

Aufnahme von 3 Exemplaren zum Buch mit ISBN = 99

Ausleihe des Buchs mit ISBN = 99 durch den Benutzer mit BENUTZER_NR = 12 am 01.03.2013

Rückgabe des Buchs mit ISBN = 99 am 31.03.2013

Erneute Ausleihe des Buchs mit ISBN = 99 durch den Benutzer mit BENUTZER_NR = 12 am 04.04.2013

Änderung des Standorts des Buchs mit ISBN = 99 auf den Wert 'B'

Löschen des Buchs mit ISBN = 99 unter Beachtung der Datenkonsistenz

Aufgabe 3

Gegeben sei die Beispieldatenbank auf Seite 5. Ergänze PRIMARY KEY und FOREIGN KEY Constraints durch geeignete ALTER TABLE Anweisungen.

Bitte HIER eintragen!

ALTER TABLE BENUTZER

ALTER TABLE BUCH

ALTER TABLE BUCH_EXEMPLAR

ALTER TABLE AUSLEIH_VORGANG

Aufgabe 4

Kreuze in der folgenden Tabelle die richtigen Antworten an. Für falsche Antworten gibt es Minuspunkte.

Aussage	richtig	falsch
MS Access ist eine Datenbank.		
SQL ist eine Programmiersprache.		
Durch die Anweisung DROP kann ein Datenbankschema geändert werden.		
UPDATE dient zur Erweiterung eines Datenbankschemas.		
Jeder Join lässt sich aus Kreuzprodukt und Selektion zusammensetzen.		
Mit Hilfe eines Joins kann man keine Duplikate erzeugen.		
Die Anweisung CREATE TABLE D AS SELECT...kann zur Erzeugung eines Tabellenduplikats verwendet werden.		
Die Bedeutung der Bezeichnung "Relationship" ist im ERM identisch mit der Bezeichnung "Assoziation".		
Integritätsbedingungen müssen sich immer auf die Tabelle beziehen, in der sie deklariert werden.		
Beginn und Ende einer Transaktion werden vom Datenbanksystem festgelegt.		
VIEWS kann man in jeder Beziehung genau so behandeln wie Tabellen.		
Gegeben sei folgende View: CREATE VIEW SUMME AS SELECT SUM(GEHALT) S FROM MITARBEITER; Die Anweisung UPDATE SUMME SET S = 123; ändert die Werte in der VIEW.		
Referentielle Integrität bedeutet, dass in der Datenbank nur Verweise auf ganzzahlige Schlüssel zulässig sind.		
Eine SQL-Anweisung darf höchstens eine GROUP BY Klausel enthalten.		
Die Daten einer Relation sind in der Datenbank stets aufsteigend nach dem Primärschlüssel sortiert.		
Magnetplatten sind ungeeignet zur persistenten Speicherung von Daten.		
Eine Abteilung, die keine Mitarbeiter hat, ist ein typisches Beispiel für eine schwache Entität.		
Folgende Gegenstände kann man als Entitäten modellieren: Hund, Katze, Maus.		
Folgende Gegenstände kann man nicht als Entitäten modellieren: Farbe, Temperatur, Gewicht.		
Die Mengenoperation „Vereinigung“ entspricht einem JOIN in SQL.		
Die Anweisung KOMM-MIT dient dazu, Änderungen in einer Datenbank festzuschreiben.		
B-Bäume können ausschließlich numerische Werte als Schlüssel aufnehmen.		
Die Größe eines B-Baum-Knotens sollte sich an den physischen Organisationseinheiten des verwendeten Speichermediums orientieren.		
Der Begriff "B-Baum" steht für "Binärbaum".		
B-Bäume wachsen nur, wenn die Wurzel geteilt werden muss.		
Die Anfrage SELECT * FROM A JOIN B ON A.a=B.b; führt zum gleichen Ergebnis wie die Anfrage SELECT * FROM A JOIN B ON B.b = A.a;		
GROUP BY PNR veranlasst außer der Gruppierung nach PNR auch gleichzeitig die Sortierung der Daten nach PNR.		
Wenn man Daten in einer VIEW ändert wirkt sich das sofort auf die Daten der Basistabelle aus.		

Beispieldatenbank - Datenblatt -

Gegeben sei folgende Datenbank für die Verwaltung der Ausleihvorgänge von Büchern durch die Benutzer einer Bibliothek.

```
CREATE TABLE BUCH (
    ISBN                INTEGER,                /* IDENTIFIZIERT EINEN BUCHTITEL */
    BUCH_TITEL          VARCHAR2(40),
    AUTOR               VARCHAR2(25),
    VERLAG              VARCHAR2(25),
    ERSCHEINUNGS_JAHR   INTEGER);

CREATE TABLE BUCH_EXEMPLAR (
    ISBN                INTEGER,                /* VERWEIS AUF EIN BUCH          */
    EXEMPLAR_NR         INTEGER,                /* DURCHNUMMERIERUNG PRO BUCH    */
    STANDORT            VARCHAR2(20) NULL); /* STANDORT IN DER BIBLIOTHEK    */

CREATE TABLE BENUTZER (
    BENUTZER_NR         INTEGER,                /* IDENTIFIZIERT EINEN BENUTZER */
    NACHNAME            VARCHAR2(30),
    VORNAME            VARCHAR2(20));

CREATE TABLE AUSLEIH_VORGANG(
    AV_NR               INTEGER,                /* IDENTIFIZIERT AUSLEIHVORGANG */
    ISBN                INTEGER,
    EXEMPLAR_NR         INTEGER,
    BENUTZER_NR         INTEGER,
    AUSGABE_DATUM       DATE,                /* TATSÄCHLICHES AUSGABEDATUM   */
    RUECKGABE_DATUM     DATE);                /* TATSÄCHLICHES RÜCKGABEDATUM  */
```

BENUTZER

BENUTZER_NR	NACHNAME	VORNAME
1	Rumpel	Stilzchen
2	Klabauter	Mann
3	König	Frosch
4	Rot	Käppchen

BUCH

ISBN	BUCH_TITEL	AUTOR	VERLAG	ERSCHEINUNGS_JAHR
1	Alles umsonst	Senger	HSKA	2000
3	Nie wieder DB I	Senger	HSKA	2002
4	DB I	Senger	BNN	1870
2	NoSql	Senger	HSKA	2012

BUCH_EXEMPLAR

ISBN	EXEMPLAR_NR	STANDORT
1	1 1	
1	2 1	
2	1 99	
1	3 44	
3	2 22	
3	3 22	
4	1 3	

AUSLEIH_VORGANG

AV_NR	ISBN	EXEMPLAR_NR	BENUTZER_NR	AUSGABE_DATUM	RUECKGABE_DATUM
1	1	1	1	06.06.13 00:00	(null)
2	1	2	1	06.06.13 00:00	06.06.13 00:00
3	2	1	3	01.02.10 00:00	28.02.10 00:00
4	2	1	4	15.05.12 00:00	17.05.12 00:00
5	3	2	2	07.08.12 00:00	18.08.12 00:00
6	3	2	2	19.06.13 00:00	(null)
7	4	1	2	01.05.13 00:00	(null)