

Klausur

Grundlagen der Wirtschaftsinformatik

Sommersemester 2022

Die Probeklausur besteht aus 4 Aufgabentypen und ist schriftlich auf Papier zu lösen. Sie umfasst insgesamt 10 Seiten. Stellen Sie sicher, dass Sie zunächst alle Blätter ihrer Klausur mit ihrem Namen und ihrer Matrikelnummer versehen (oben rechts). Sofern Sie noch zusätzliche Seiten benötigen, können Sie die natürlich bekommen. Auch diese müssen dann mit Name und Matrikelnummer versehen werden.

Blätter, die keine Matrikelnummer enthalten, werden nicht korrigiert.

Hilfsmittel: Für die Bearbeitung der Klausur sind als Hilfsmittel sämtliche Vorlesungsunterlagen (Folien, Skripte, Übungsblätter) sowie eigene Notizen erlaubt. Zusätzlich werden für einige der Aufgaben Software-Tools über das WWW zur Verfügung gestellt, die zur Entwicklung der Lösungen benutzt werden können. Das Ergebnis muss jedoch auf Papier übertragen werden.

In jeder Aufgabe können maximal 25 Punkte erreicht werden, d.h. in der gesamten Klausur sind maximal 100 Punkte erreichbar. Zum Bestehen der Klausur müssen mindestens 50 Punkte erreicht werden.

Aufgabe	Erreichte Punkte
1	
2	
3	
4	
Σ	

Aufgabe 1 Prozessmodellierung mit BPMN (25 Punkte)

Gegeben ist der nachfolgend geschilderte Anwendungsfall. Erstellen Sie ein BPMN Modell für die angegebene Prozessbeschreibung. Wenn Sie möchten, können Sie für die Modellierung das Ihnen bekannte Online-Tool unter

<https://datascience.hs-bochum.de/bpmn/>

nutzen. Die zu entwickelnde Prozessbeschreibung muss jedoch als Lösung auf Papier übertragen werden.

Anwendungsfall:

Für ihre Musik-Event und Konzert-Agentur planen Sie regelmäßig Konzerte und Auftritte verschiedener Künstler. Über das Jahr hinweg wird eine Liste mit potentiellen Künstlern gepflegt, die für Konzerte gebucht werden sollen. Um Fehlplanungen im Prozess auszuschließen, soll der Prozess zunächst als BPMN modelliert und später in einem Software-System programmiert werden.

Der Ablauf für die Konzertplanung läuft dann immer auf die folgende Weise ab:

Zunächst wird eine Anfrage an das Management eines Künstlers geschickt, ob an einer Buchung Interesse besteht. Falls der Künstler/das Management kein Interesse oder keine Zeit für ein Event hat, ist der Prozess beendet.

Hat der Künstler an einem Engagement Interesse, wird nach einer passenden Halle gesucht. Falls keine Halle gefunden wurde, wird der Prozess mit einer Absage beendet. Ist eine Halle gefunden, wird ein Vertrag an das Management des Künstlers und den Betreiber der Halle geschickt. Wenn sowohl der Künstler als auch der Hallenbetreiber den Vertrag bestätigen/unterschreiben, ist der Prozess erfolgreich beendet. Andernfalls endet der Prozess ebenfalls mit einer Absage.

Hinweis: Es ist nur die Veranstaltungsplanung für einen Künstler zu modellieren, also nicht auch noch die zusätzliche Verwaltung der Liste von Künstlern.

Aufgabe 2 ER Modellierung (25 Punkte)

Als Event-Manager für Konzertveranstaltungen benötigen Sie eine Datenbank für Konzerte, Festivals oder andere Veranstaltungen.

Es müssen Künstler gespeichert werden, für die Name und Adresse hinterlegt werden müssen. Jeder Künstler wird von genau einem Manager betreut, ein Manager kann dabei allerdings mehrere Künstler betreuen. Für Manager müssen der Name und die Telefonnummer gespeichert werden.

Zusätzlich müssen die Veranstaltungsorte wie Hallen, Clubs oder ähnliches gespeichert werden. Für jeden Veranstaltungsort sind der Name, die Adresse und die maximale Anzahl an Zuschauern zu speichern.

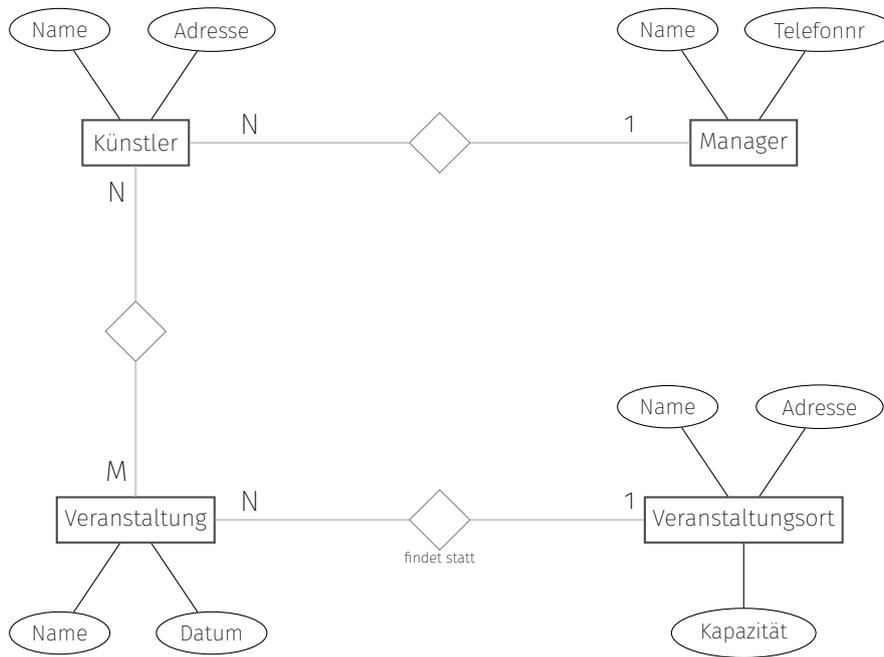
Im Kern soll ihr Datenmodell natürlich auch noch die Veranstaltungen speichern. Eine Veranstaltung findet zu einem gegebenen Datum an einem Veranstaltungsort statt. Bei jeder Veranstaltung treten einer oder mehrere Künstler auf.

Ihre Aufgabe:

Erstellen Sie ein ER-Diagramm (Chen-Notation) zu diesem Anwendungsfall. Nutzen Sie für die Erstellung des ER-Diagramms diese und/oder ggf. die folgende Seite.

Name: _____
Matrikel-Nr:

--	--	--	--	--	--	--	--	--	--

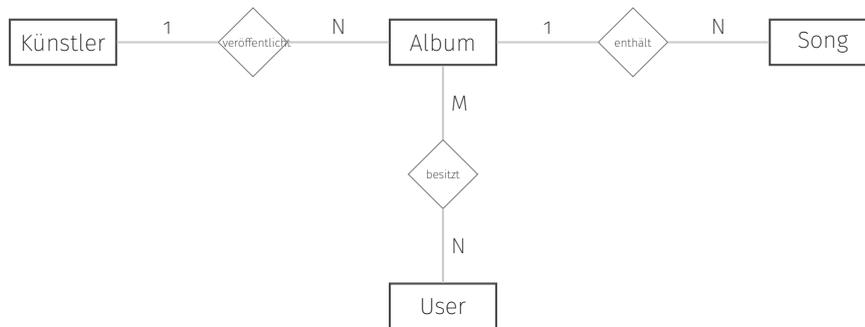


Aufgabe 3 Ableitung von Datenbanktabellen (25 Punkte)

Neben Ihrer Eventagentur betreiben Sie auch eine Musik-Plattform, bei der Künstler Ihre Songs veröffentlichen können, die dann von zahlenden Benutzern online gekauft werden können.

Jeder Künstler veröffentlicht dort ein oder mehr Alben, die jeweils aus einem oder mehreren Songs bestehen. Benutzer können immer nur ganze Alben kaufen. Für jeden Künstler wird sein Name gespeichert, für Alben der Name des Albums und das Erscheinungsjahr. Für die Songs wird der Titel und die Länge in Sekunden gespeichert. Beim Kauf eines Albums wird neben der Zuordnung von Kunde und Album auch noch das Kaufdatum mit gespeichert.

Das folgende ER-Diagramm zeigt die Beziehungen der einzelnen Entitäten zueinander.



Aufgabe: Leiten Sie aus dem obigen ER-Diagramm die entsprechenden Datenbanktabellen ab. Jede Tabelle soll dabei in der folgenden Form dargestellt werden:

Name der Tabelle			
Leer PK FK	Attributname1	Datentyp1	Leer nn
Leer PK FK	Attributname2	Datentyp2	Leer nn
...

Zeichnen Sie die Tabellen zu dem ER-Diagramm in der obigen Form auf dieser und ggf. der nachfolgenden Seite:

Mögliche Lösung:

Album			
PK	Id	int	nn
FK	Künstler	int	nn
	Name	String	nn
	Erscheinungsjahr	int	nn

Song			
PK	Id	int	nn
FK	Album	int	nn
	Titel	String	nn
	Länge	int	nn

Künstler			
PK	Id	int	nn
	Name	String	nn

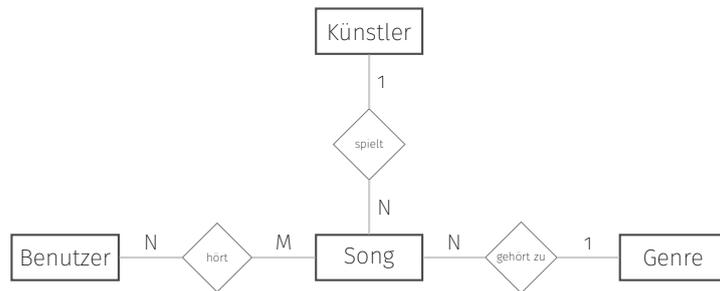
User			
PK	Id	int	nn
	Name	String	nn

AlbumUser			
PK, FK	Album	int	nn
PK, PK	User	int	nn
	Datum	date	nn

Aufgabe 4 SQL Abfragen (25 Punkte)

Ihre Musikplattform ist auch mit einem Streaming-Dienst ausgestattet. Dazu existiert eine Datenbank, die neben den Songs, Künstlern und Benutzern auch die Informationen darüber, welcher User welchen Song abgespielt hat speichert. In der Tabelle *Benutzer-Song* wird dazu jedesmal die Benutzer ID, Song ID und das Datum eingetragen, wenn ein Benutzer einen Song abspielt.

Für jeden Song wird zusätzlich auch das Genre gespeichert und für jeden Benutzer eine Reihe weiterer Merkmale (Alter, Geschlecht, etc.). Das folgende ER-Diagramm zeigt die Struktur der Datenbank:



Aus dem ER-Diagramm ergeben sich die unten abgebildeten Tabellen, die mit Beispieldaten gefüllt sind. Sie können über die URL

https://datascience.hs-bochum.de/musik_db

auf eine Datenbank mit diesem Schema zugreifen, die weitere Beispieldaten enthält.

Benutzer

id	alter	geschlecht
1	27	m
2	29	w
3	19	m
4	47	w
5	58	w
6	63	m
7	59	w
8	51	w
9	35	m

Musiker

id	name
1	Die Happy
2	Metallica
3	Eric Clapton
4	Milow
5	Ed Sheeran
6	The Weeknd
7	Vincent weiss
8	Clueso

Genre

id	name
1	Pop
2	Klassik
3	Rock
4	Electronic
5	Blues
6	Heavy Metal

BenutzerSong

benutzer	song	datum
1	1	2022-04-04
2	1	2022-04-08
7	7	2022-04-03
5	9	2022-04-17
7	5	2022-04-17
6	4	2022-04-18
3	5	2022-04-06
2	5	2022-04-19
3	5	2022-04-05
2	6	2022-04-17

Song

id	titel	genre	musiker
1	Supersonic Speed	3	1
2	Enter Sandman	6	2
3	Layla	5	3
4	Ayo Technology	1	4
5	Afterglow	1	5
6	Wer wenn nicht wir	1	7
7	Blinding Lights	1	6
8	Bad Habits	1	5
9	Neuanfang	1	8
10	Flugmodus	1	8

Aufgabe: Geben Sie für die folgenden Fragestellungen jeweils einen SQL-Befehl an, der die entsprechende Ausgabe produziert. Zu jeder Fragestellung ist jeweils die Liste der Spalten angegeben, die in der Ergebnistabelle enthalten sein sollen.

Hinweis: Die Ergebnistabellen ihrer Befehle sollen die gleichen Spaltenbezeichnungen haben, wie in der jeweiligen Aufgabe angegeben. Benutzen Sie dafür die Benennung der Spalten mit **AS**, z.B.

```
SELECT count(id) AS Anzahl FROM Benutzer
```

Schreiben Sie den SQL-Befehl für die jeweilige Fragestellung in die zugehörige Box.

1. Ermitteln Sie die Anzahl der Benutzer in der Altersgruppe bis 25 Jahre.
(Spalten: **Anzahl**)

```
SELECT count(id) FROM Benutzer WHERE Alter <= 25
```

2. Geben Sie die Anzahl der verfügbaren Songs nach Genres an. Es soll in der **Genre** Spalte jeweils der Name des Genres angezeigt werden.
(Spalten: **Genre** | **Anzahl**)

```
SELECT Genre.name AS Genre, count(Song.Id) AS Anzahl  
FROM Song  
LEFT JOIN Genre ON Song.genre = Genre.id  
GROUP BY Genre.id
```

3. Geben Sie jeden Song (ID) und die Häufigkeit aus, die er abgespielt wurde (Tabelle BenutzerSong). Das Ergebnis soll absteigend nach Abspielhäufigkeit sortiert werden.
(Spalten: **SongID** | **Anzahl**)

```
SELECT benutzersong.song AS SongID, count(*) AS Anzahl  
FROM BenutzerSong  
GROUP BY song  
ORDER BY count(*) DESC
```

4. Geben Sie die verschiedenen Liedernummern (Song IDs) der abgespielten Streams an, die von Benutzern in der Altersgruppe von 18 bis (einschließlich) 25 gehört werden.

(Spalten: **SongID**)

```
SELECT DISTINCT song AS SongID FROM BenutzerSong
LEFT JOIN Benutzer ON BenutzerSong.benutzer = Benutzer.id
WHERE benutzer.alter >= 18 AND benutzer.alter <= 25
```

5. Ermitteln Sie das Durchschnittsalter der männlichen und weiblichen Benutzer.

(Spalten: **Geschlecht** | **Durchschnittsalter**)

```
SELECT geschlecht, AVG(alter) AS Durchschnittsalter
FROM Benutzer
GROUP BY geschlecht
```

6. Hören eher Frauen oder Männer die meisten Musiktitel? Geben Sie einen Befehl an der die Anzahl der verschiedenen Songs je Geschlecht angibt.

(Spalten: **Geschlecht** | **Anzahl**)

```
SELECT geschlecht, count(song) FROM BenutzerSong
LEFT JOIN Benutzer ON BenutzerSong.benutzer = Benutzer.id
GROUP BY geschlecht
```