

☒ DES3UEG1: Glock Name Elias Leonhardsberger Aufwand in h 5
☐ DES3UEG2: Werth Punkte _____ Kurzzeichen Tutorin _____

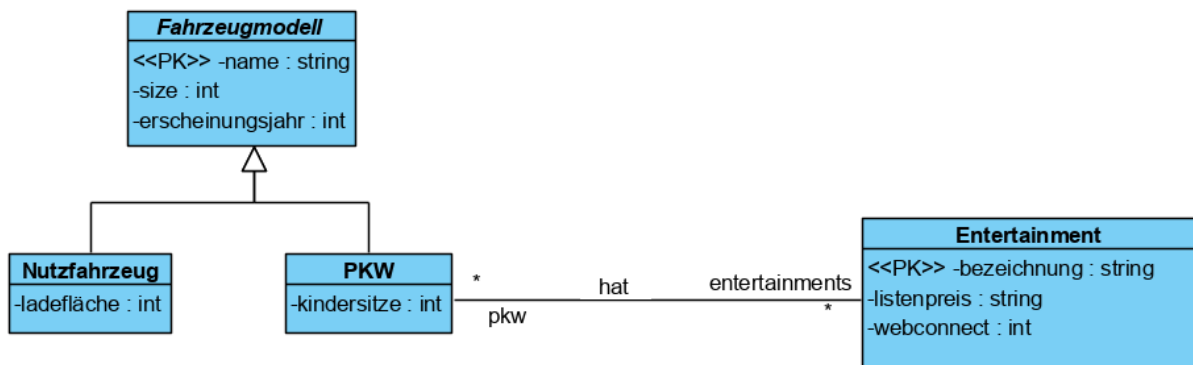
1. Abbildung Generalisierung

(4 Punkte – je 2 Pkte.)

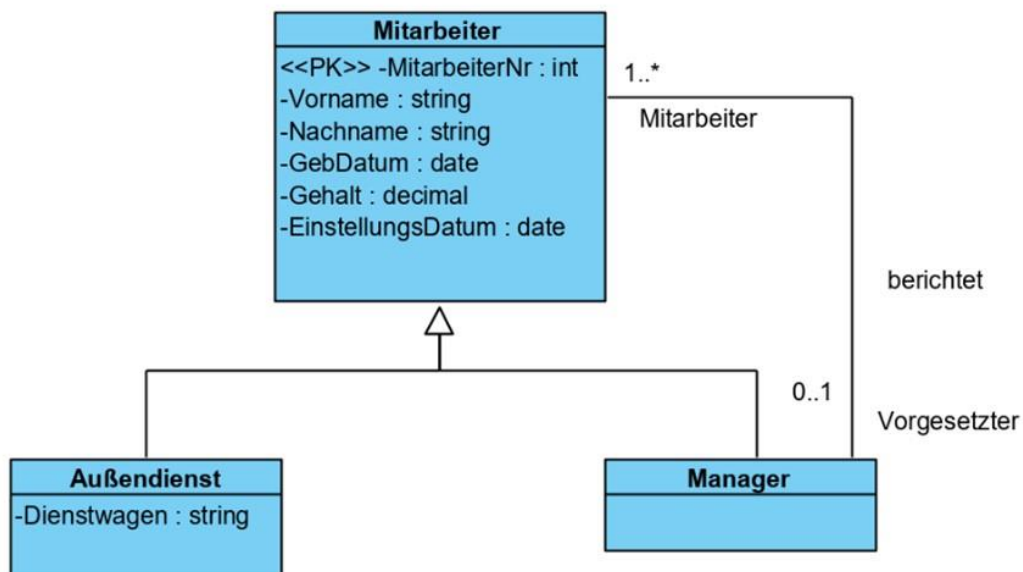
Erstellen Sie für die angegebenen UML-Klassen-Diagramme jeweils ein **Relationenmodell**, um die Attribute in Tabellen in der Datenbank abzubilden, zB Tabelle 1 (PKAttr1, Attr2). Wählen Sie jeweils ein geeignetes Abbildungsmodell für die Generalisierungsbeziehung.

Begründen Sie Ihre Wahl: D.h. nennen Sie das gewählte Abbildungsmodell und geben Sie dessen Vor- und Nachteile an. Nehmen Sie dabei Bezug auf die Eigenschaften der Generalisierungsbeziehung, d.h. ob diese **vollständig/unvollständig** und **überlappend/disjunkt** ist. Geben Sie auch an, warum sich die anderen Modelle nicht bzw. schlechter eignen.

- a) Ein Fahrzeughersteller verwaltet die Fahrzeuge von zwei Sparten Nutzfahrzeuge und PKWs. Die Sparten werden getrennt verwaltet, ein Modell wird nicht in beiden Sparten hergestellt.



- b) In einem Unternehmen werden Mitarbeiter verwaltet, Mitarbeiter im Außendienst erhalten ein Fahrzeug zur Verfügung gestellt; im Außendienst gibt es ebenso Manager.



2. SQL-Wiederholung (Sakila-DB)

(5 Punkte)

1. Geben Sie eine Liste mit Filmtitel aus, deren Namen an vierter Stelle ein 'A' enthält, geben Sie die Titel so aus, dass jeweils der erste Buchstabe eines Wortes mit einem Großbuchstaben beginnt (zB Atlantis Cause). (67 Zeilen, 0,5 Punkte)
2. Geben Sie alle Inventar-Ids aus, die noch nie verliehen wurden. Verwenden Sie einen Mengenoperator. (1 Zeile, 0,5 Punkte)
3. Geben Sie die Anzahl der Verleihvorgänge zwischen 1.1.2015 und 31.12.2015 aus (Start des Verleihvorgangs, *rental_date*). (1 Zeile, 1 Punkt)
4. Geben Sie alle Filialen mit ID (*store_id*) und Stadt (*city*) gemeinsam mit den Nachnamen der zugeordneten Angestellten aus (JOIN). Listen Sie auch Filialen auf, die keine:n Angestellte:n zugewiesen haben und verwenden Sie den Platzhalter 'no staff'. (9 Zeilen, 1 Punkt)
5. Geben Sie die Namen aller Schauspieler aus, die in Filmen mitspielen, die nach 2006 erschienen sind und in denen auch die Schauspieler mit den IDs 10, 20 oder 30 vorkommen. (36 Zeilen, 1 Punkt)
6. Geben Sie für den Kunden mit der ID 250 alle Verleihvorgänge mit dem Start des Verleihvorgangs und dem bezahlten Betrag aus. Geben Sie das Datum in zwei Sprachen aus (**Format** 'Mi, 11. Oktober 2023' bzw. 'Wed, 11st October 2023' aus, verwenden Sie dazu die Funktion *to_char* inkl. NLS-Parameter und recherchieren Sie bei Bedarf in der Oracle-Dokumentation. (1 Punkt)

3. Sakila-Statistik

(3 Punkte)

Erstellen Sie die folgenden Statistikberichte für die Sakila-Geschäftsführung: Nehmen Sie die Nummer, Vor und Nachname des Managers und die Größe des Inventars (Anzahl der Filme) für jeden Store auf, die folgende Bedingungen erfüllt:

1. Keine oder mehr als 1 Angestellte/r
2. Höchste Anzahl von Filmen
3. Niedrigste Anzahl von Filmen

Hinweis:

Beachten Sie, dass es Stores ohne Mitarbeiter und ohne Inventar geben kann bzw. könnte. Prüfen Sie die Korrektheit Ihrer Abfragen, indem Sie einen neuen Store einfügen (z.B. *address_id* 223, *last_update* jetzt, *manager_staff_id* 1, *store_id* 7)

Machen Sie durchgeführte Änderungen nach Ausführung der Teststatements wieder rückgängig!

4. Aggregate und Gruppierungen (Sakila-Datenbank)

(5 Punkte)

1. Ermitteln Sie die Titel jener Filme, die zwar in der Filmdatenbank existieren, allerdings in keinem Geschäft angeboten werden. (42 Zeilen, 1 Punkt)
2. Geben Sie pro Kunden-Name (Vorname, Nachname) die Anzahl geliehener Filme an. Sortieren Sie nach der Anzahl aufsteigend. (599 Zeilen, 1 Punkt)
3. Ermitteln Sie alle Filme, die die längsten in ihrer Sprache sind und geben Sie Titel, Dauer (*length*) sowie Sprache (Name) aus. Im Ergebnis sollen nur jene Filme angezeigt werden, bei denen eine Originalsprache eingetragen ist. (11 Zeilen, 1 Punkt)
4. Geben Sie in Absteigender Reihenfolge an, welcher Kunde (Vorname, Nachname, ID) bereits vier oder mehr Horror-Filme ausgeliehen hat. (34 Zeilen, 1 Punkt)

5. Geben Sie den besten Kunden (gemessen am Umsatz, d.h. wie viel er gesamt bezahlt hat) pro Store an, sowie den Umsatz und die Store-ID des Kunden). Bedenken Sie dabei den Fall, dass Personen gleich heißen könnten. (6 Zeilen, 1 Punkt)

5. GROUP BY mit GROUPING SETS / ROLLUP (Sakila-Datenbank) (3 Punkte)

Erstellen Sie eine Filmabfrage und ermitteln Sie die Anzahl der Filme und die Summe der Längen in den Kategorien ‚Comedy‘ und ‚Music‘

- je Rating
- je Kategorie und Rating
- „Comedy“ und „Music“ gesamt

Entwickeln Sie drei Varianten:

- (1) einmal mit der Verwendung des GROUP BY ROLLUP Operators (1 Punkt) und
- (2) zum Vergleich dazu mit GROUPING SETS (1 Punkt) und
- (3) komplett ohne ROLLUP und SETS (Hinweis: 3 Selects mit UNION) (1 Punkt).

Vergleichen Sie die Zugriffspläne und kommentieren Sie diese.

Hinweis: SET AUTOTRACE ON erlaubt in SQL*PLUS die Ausgabe des Zugriffsplan sowie einiger Statistiken eines SQL-Statement und bietet eine einfache Möglichkeit zum Vergleich von Abfragen.

5. GROUP BY mit GROUPING SETS / CUBE (Sakila-Datenbank) (4 Punkte)

1. Erstellen Sie mit GROUPING SETS eine Abfrage, um folgende Gruppierungen anzuzeigen: (1 Punkt)

- manager_staff_id, store_id, staff_id
- manager_staff_id, store_id
- store_id, staff_id

Die Abfrage soll die Summe der Erlöse für jede dieser Gruppen berechnen (s. Lösungsauszug).

MANAGER_STAFF_ID	STORE_ID	STAFF_ID	SUM...
1	1	1	44694,83
2	2	2	...
3	3	3	...
...
5	5	5	6662,75
...
1	1	(NULL)	44694,83
(NULL)	1	1	44694,83
...
...	5	6	6454,64

2. Erstellen Sie einen Bericht über verliehene Filme, der pro Land, Jahr und Kategorie die Summe des Umsatzes und die Anzahl der Bezahlvorgänge (vgl. payment) zusammenfasst. Berücksichtigen Sie nur die Kategorien 'Family', 'Children' und 'Travel'. Erstellen Sie ein SQL-Statement und verwenden Sie GROUPING SETS. Sortieren Sie die Ausgabe nach Ländern. (1,5 Punkte)

Jahr	Land	Kategorie	Anzahl	Summe
NULL	Australia	Children	153	1128,88
	...			
2013	Australia	NULL	5	41,16
	...			
NULL	NULL	NULL	2878	21871,51

3. Erstellen Sie eine Abfrage, um für alle Manager, die in den Stores angestellt sind, folgendes anzuzeigen (1 Punkte):

- Manager-Id
- Store und Gesamterlös für jeden Manager
- Gesamterlöse aller Manager
- Kreuztabellenwerte für die Anzeige des Gesamterlöses für jeden Standort
- Gesamterlös, unabhängig vom Standort

(Auszug aus der Lösung; Bitte vervollständigen Sie die Tabelle in Ihrer Abgabe)

MANAGER-ID	STORE	Gesamterlös
NULL	NULL	115657,58
NULL	1	44694,83
...
...
3	NULL	6620,07
3	3	6620,07
...

4. Prüfen Sie die Ausgabe der obigen Aufgabe. Schreiben Sie mit der GROUPING-Funktion eine Abfrage, um festzustellen, ob die Nullwerte in den Spalten, die den GROUP BY Ausdrücken entsprechen, von der CUBE-Operation verursacht werden. (0,5 Punkte)

DES3UE Übung 2

Elias Leonhardsberger

20. November 2024, Hagenberg

Inhaltsverzeichnis

1	Abbildung Generalisierung	6
1.1	a) Fahrzeuge	6
1.1.1	Relationenmodell	6
1.2	b) Mitarbeiter	6
1.2.1	Relationenmodell	6
2	SQL-Wiederholung	6

1 Abbildung Generalisierung

Die Groß/Kleinschreibung wurde aus der Angabe übernommen.

1.1 a) Fahrzeuge

Für die Vollständigkeit der Generalisierung kann man nur Annahmen treffen, da die Angabe nicht ausführlich genug ist. Ich nehme daher an das die Generalisierung *vollständig* ist. Aus der Angabe liest man auch heraus das sie *disjunkt* ist.

Da es eine *n to m* Beziehung auf eine der Subklassen gibt entscheide ich mich für das Partitionierungsmodell. Dieses bietet mir die Möglichkeit die Relation nur auf die Subklasse via Foreign Key zu definieren. Weiters ist sie die einzige Abbildung die die Normalformen einhält. Performanzweise könnte man über eine Einrelationenabbildung streiten, da man aber die Beziehung nicht richtig abbilden kann fällt diese weg.

1.1.1 Relationenmodell

Fahrzeugmodell(name, size, erscheinungsjahr)
 Nutzfahrzeug(name : FK(*Fahrzeugmodell*), ladefläche)
 PKW(name : FK(*Fahrzeugmodell*), kindersitze)
 Entertainment(bezeichnung, listenpreis, webconnect)
PKW__Entertainment(bezeichnung : FK(*Entertainment*), name : FK(*PKW*))

1.2 b) Mitarbeiter

Dieses Beispiel ist eindeutig *unvollständig* und *überlappend*.

Da es wieder eine Beziehung auf eine der Subklassen gibt entscheide ich mich wieder für das Partitionierungsmodell. Die Einrelationenabbildung wäre wieder möglich und ist, da ein Manager keine Attribute hat, auch sinnvoll, aber man kann nicht über das Model garantieren, dass nur Manager Vorgesetzte sein können.

1.2.1 Relationenmodell

Mitarbeiter(MitarbeiterNr, Vorname, Nachname,
 GebDatum, Gehalt, EinstellungsDatum
 Vorgesetzter : FK(*Manager*))

 Manager(MitarbeiterNr : FK(*Mitarbeiter*))
Aussendienst(MitarbeiterNr : FK(*Mitarbeiter*), Dienstwagen)

2 SQL-Wiederholung