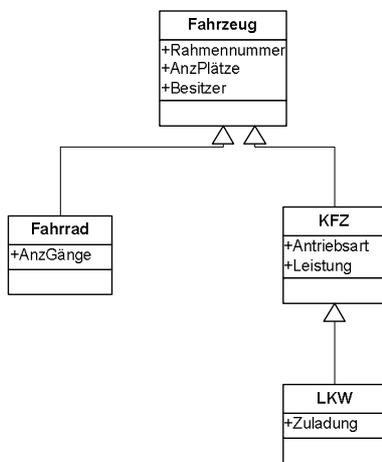


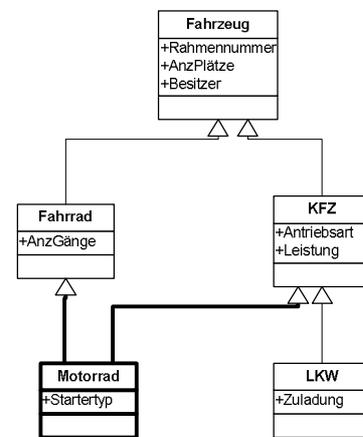
UML

1. Setzen sie das obige, Ihnen bekannte E/R-Diagramm in das entsprechende UML-Klassendiagramm um (zunächst noch ohne Ergänzungen)
2. Ergänzen sie die Klasse Motorrad, die sowohl von Fahrrad als von KFZ erbt, und als weiteres Attribut „Startertyp“

Lösung zu 1

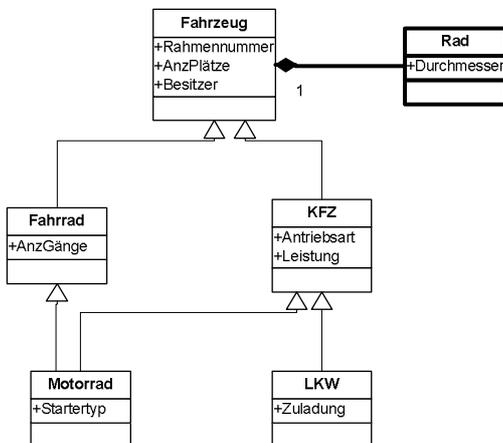


Lösung zu 2

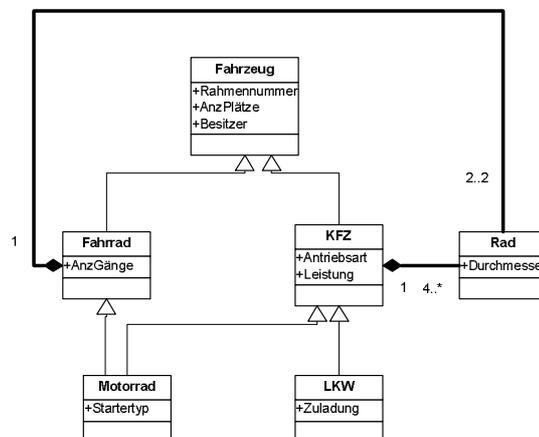


3. a) Ergänzen sie eine Klasse Rad, welche das Attribut Durchmesser hat. Diese Klasse ist Teil der Komposition von Fahrzeug
- b) Ergänzen sie die Multiplizitäten dieser Komposition. Denken sie daran, dass ein Fahrrad genau zwei Räder hat, ein KFZ 4 oder mehr Räder.

Lösung zu 3a)



Lösung zu 3b)



4. Setzen sie das Ergebnis von 1.) in ein relationales Schema um. Verwenden Sie dazu die Strategien 2 und 4 aus der Vorlesung.
- a) Was verwenden Sie am besten als Schlüssel?

Nach klassischer Modellierung (siehe auch Blatt 4/Aufgabe3) wäre Rahmennummer ein guter Schlüssel, da bei allen Fahrzeugen (auch den spezialisierten) vorkommt und einzigartig ist. Alternativ könnten Identifierschlüssel (z.B. fortlaufende Nummern) verwendet werden, da dies mehr der Semantik von OO-Systemen entspricht. Um die Umsetzung vergleichbar mit Blatt 4/Aufgabe 3 zu halten, wird im folgenden Rahmennummer verwendet.

Umsetzung nach 2. Strategie

Fahrzeug	(<u>Rahmennr</u> , AnzPlätze, Besitzer)
Fahrrad	(<u>Rahmennr</u> , AnzPlätze, Besitzer, AnzGänge)
KFZ	(<u>Rahmennr</u> , AnzPlätze, Besitzer, Antriebsart, Leistung)
LKW	(<u>Rahmennr</u> , AnzPlätze, Besitzer, Antriebsart, Leistung Zuladung)

Einträge für Fahrrad erfolgen in Fahrrad und Fahrzeug, für LKW in LKW, KFZ und Fahrzeug.

Umsetzung nach 4. Strategie:

Fahrrad	(<u>Rahmennr</u> , AnzPlätze, Besitzer, AnzGänge)
KFZ	(<u>Rahmennr</u> , AnzPlätze, Besitzer, Antriebsart, Leistung)
LKW	(<u>Rahmennr</u> , AnzPlätze, Besitzer, Antriebsart, Leistung Zuladung)

```
CREATE VIEW FAHRZEUG AS
SELECT Rahmennr, AnzPlätze, Besitzer FROM Fahrrad UNION
SELECT Rahmennr, AnzPlätze, Besitzer FROM KFZ UNION
SELECT Rahmennr, AnzPlätze, Besitzer FROM LKW
```

- b) Wie müssen die ON UPDATE und ON DELETE-Regeln aussehen?

Umsetzung nach 4:

Keine Regeln notwendig, da die Daten jeder Instanz in genau einer Tabelle gespeichert werden.

Umsetzung nach 2:

Semantisches Problem: Was ist Primärschlüssel, was ist Fremdschlüssel?

Wenn ein Fahrrad gelöscht wird, kann dieses entweder

- i. als Fahrrad gelöscht werden (automatisches Löschen bei Fahrzeug) oder
- ii. als Fahrzeug gelöscht werden (automatisches Löschen bei Fahrrad)

Praktisch geht i) nicht, da sich ein Fremdschlüssel immer auf genau eine Tabelle beziehen muss.

Deshalb ist nur ii) machbar. Somit gilt
CREATE TABLE Fahrrad (
Rahmennummer INTEGER NOT NULL PRIMARY KEY REFERENCES
Fahrzeug ...
Die weiteren Tabellen werden nach demselben Muster angelegt.

Für ON UPDATE gelten die gleichen Einschränkungen, womit vermutlich
NO ACTION die beste Wahl.

c) Kann man alle in der Vorlesung geforderten Integritätsbedingungen damit erfüllen?

Nein, da

1. sich die Beziehungen in beide Richtungen nicht darstellen lassen
2. sich ON UPDATE und ON CASCADE-Regeln stets auf Schlüsselattribute beziehen, nach der Vorlesung aber gelten muss:
Wenn Fahrrad.Rahmennummer = Fahrzeug.Rahmennummer, dann
Fahrrad.AnzPlätze = Fahrzeug.AnzPlätze und
Fahrrad.Besitzer = Fahrzeug.Besitzer.
Ähnliches gilt bei den anderen Tabellen.

Diese Integritätsbedingungen können nur auf Anwendungsseite oder mit Triggern sichergestellt werden.

5. Setzen sie die Klasse Motorrad entsprechend der Strategien 1 und 2. um. Wie müssen jeweils die Integritätsbedingungen aussehen?

1. Strategie:

Motorrad (Rahmennummer, Startertyp)

Alle vererbten Attribute werden in den Tabellen der Vorfahren gehalten.

2. Strategie:

Motorrad (Rahmennr, AnzPlätze, Besitzer, AnzGänge, Antriebsart, Leistung)

Alle vererbten Attribute werden in Motorrad und (mehreren) Tabellen der Vorfahren gehalten.

Bei den Lösch- und Updateregeln treten die gleichen Probleme wie bei 4b) auf, weiterhin muss man aus den zwei Vorfahren Fahrrad und KFZ einen aussuchen, auf den sich der Fremdschlüssel bezieht.

Bei der 2. Strategie kommt noch die Konsistenzhaltung der replizierten Attribute hinzu. Hier gilt, dass die Attribute bei beiden Vorfahren und deren Vorfahr gleich sein müssen, z.B.

Motorrad.Rahmennummer = Fahrrad.Rahmennummer = KFZ.Rahmennummer
Fahrzeug.Rahmennummer.=>

Motorrad.Besitzer = Fahrrad. Besitzer = KFZ. Besitzer Fahrzeug. Besitzer
usw.

6. Setzen Sie die Ergänzungen von 3b) in das relationale Schema um. Verwenden sie dazu Strategie 1 und 2 aus der Vorlesung. Gibt es eine Möglichkeit, die „genau zwei Räder“ beim Fahrrad auszudrücken?

Strategie 1: Schwache Entität auf der Seite des zusammengesetzten Objekts
Rad (Radnummer, Durchmesser)
Fahrrad (Rahmennummer, AnzGänge, Rad1, Rad2)
KFZ: nicht möglich, da die Anzahl der Räder nicht fest ist
Mit dieser Modellierung ist das „genau zwei Räder“ des Fahrrads ausgedrückt

Strategie 2: Schlüsseleinbettung bei Rad
Rad (Radnummer, Rahmennummer, Durchmesser)
Fahrrad, KFZ unverändert. Hier kann das „genau zwei Räder“ beim Fahrrad nicht ausgedrückt werden.

7. Wie ändert sich die Umsetzung, wenn Räder nicht als Komposition, sondern als Aggregation dargestellt wird? (Denken Sie dabei an ein Reserveräder, die in größeren Mengen für mehrere Autos vorgehalten werden!)

Bei der 1. Strategie bleibt die Umsetzung gleich. Es wäre zu überlegen, ob nicht die 3. Strategie (Fahrzeug als starke Entität) besser ist, da der Verlust eines Rades nicht zu Vernichtung aller Fahrzeuge führen sollte.

Mit der 2. Strategie ist eine Umsetzung direkt nicht mehr möglich, da mehrere Rahmennummern nicht für ein Rad eingebettet werden können. Möglich ist jedoch die Umsetzung mit einer separaten Tabelle
FahrzeugRad (Rahmennummer, Radnummer)

Klausurtermin: Die Finalklausur findet statt am 13.2.2004 von 16 Uhr c.t. bis 18 Uhr in Raum PH HS 1 (PH 2501). Details auf der Webseite zur Übung.