**Paketversanddienst-Dokumentation**

**Überlegung zur Modellierung**

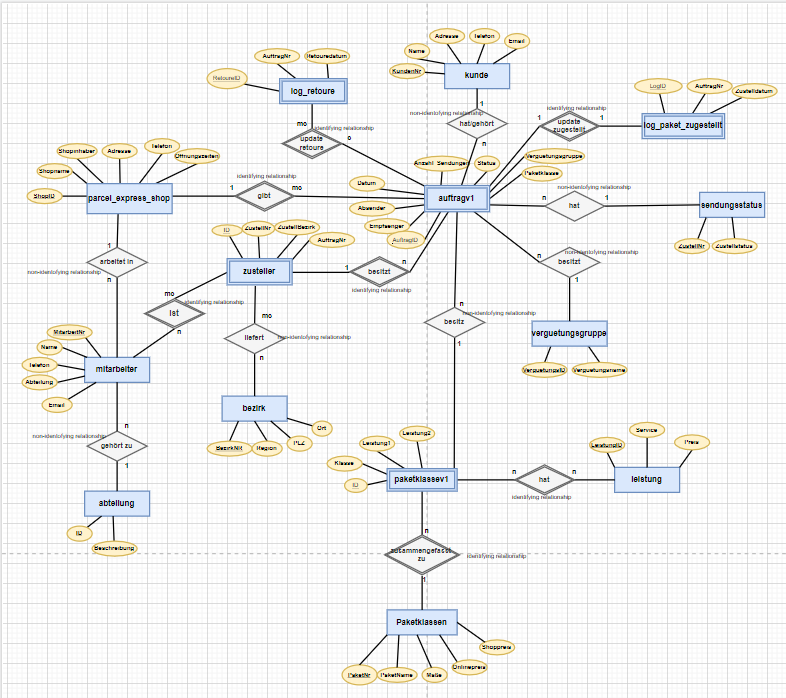
Folglich wird die Modellierung der Aufgabe 3 des zweiten Testats, der Paketversanddienst, beschrieben und durch eine Datenbank modelliert.

Grundlegende Überlegungen waren wie folgende:

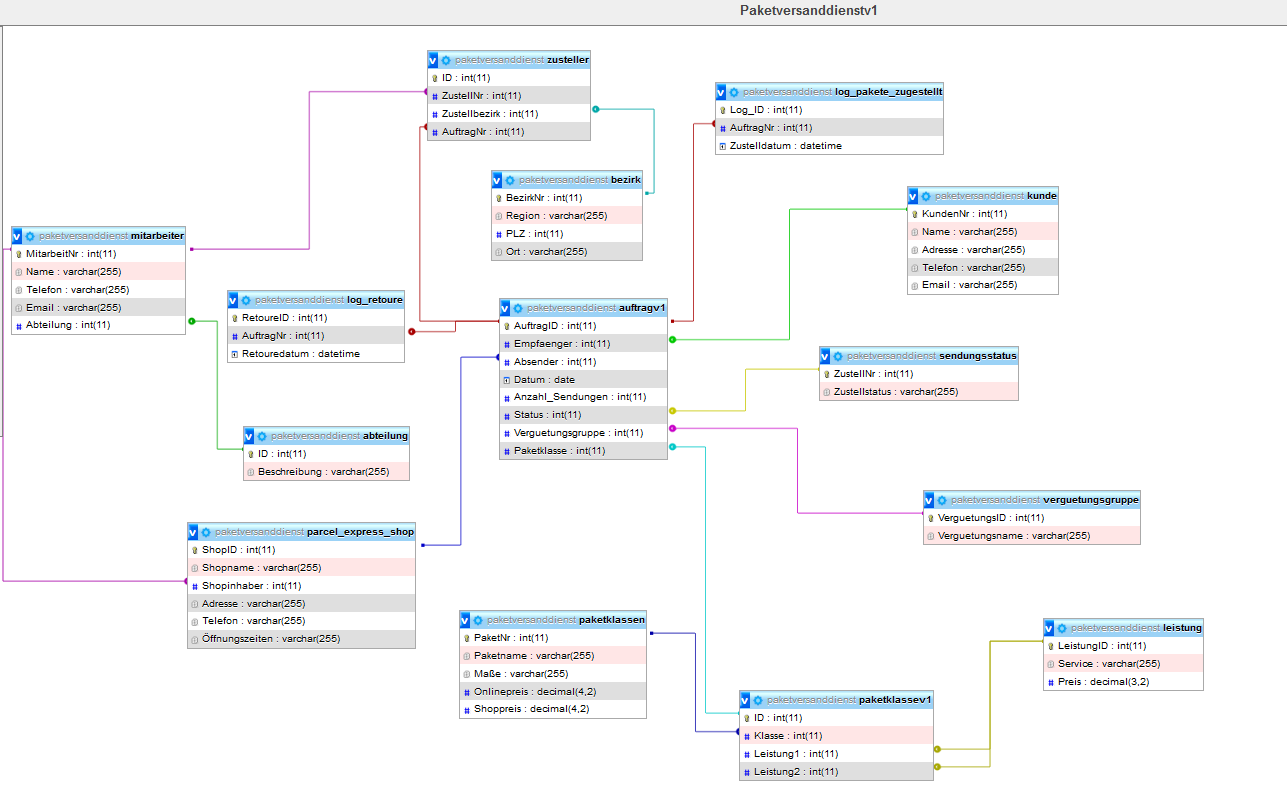
* Es existieren n-viele beispielsweise Aufträge, Kunden, Zusteller, Vergütungsgruppen,etc.
* All diese erhalten eine ID
* All diese werden mit Stammdaten (=Eigenschaften) befüllt
* Es existieren Verknüpfungen und Abhängigkeiten zwischen Ihnen

Nun soll eine grobe Skizzierung eines ER-Diagramms (=Entity Relationship) folgen, das anhand der Skizzierung in eine MYSQL-Datenbank realisiert wird.

*Skizze ERD:*



**phpMyAdmin Designer Ansicht**

****

Der Designer aus phpMyAdmin verdeutlicht das Beziehungsschema der jeweiligen Relationen. Hierbei ist zu beachten, dass die Beziehungsrichtung folglich vom großen „halbkreis“ ausgeht. Diese stellen die Verknüpfung vom Fremdschlüssel dar.  
Ein Beispiel zudem.   
Die Tabelle mitarbeiter besitzt einen Primary Key „MitarbeitNr“ und die Tabelle zusteller besitzt unteranderem den Foreign Key „ZustellNr“. Von ZustellNr wird auf MitarbeitNr referenziert, somit ist die Beziehung deren kenntlich gemacht worden.

**Beispiel Szenarien**

Was passiert, wenn ein Mitarbeiter, der Zusteller oder Shopinhaber ist, gelöscht wird?

Nun, in diesem Fall steht in der parcel\_express\_shop-Tabelle in der Spalte Shopinhaber ein NULL solange kein neuer für diese Position ernannt wurde. Genauso geschieht dasselbe in der Zusteller-Tabelle. Wobei das Problem ist, dass jeder Zusteller einem Auftrag zugeordnet wird, d.h., dass wenn der eine Zusteller entfernt wird für beispielsweise den Auftrag mit der ID 2, so bleibt der Auftrag noch bestehen, jedoch ohne Zusteller. Daher muss manuell schnellstmöglich für den Auftrag ohne einen Zusteller ein Neuer hinzugefügt werden. Ansonsten würde der Auftrag nie ausführbar werden und den Status „verloren gegangen?“ erhalten, da die Spanne einer Lieferung innerhalb von zwei Wochen erfolgen muss.

Jede Paketklasse besitzt genau zwei Leistungen, die mit den Spalten Leistung1 und Leistung2 realisiert wurde. Was passiert, aber wenn eine Paketklasse nur noch eine Leistung besitzen soll oder sogar mehr als zwei?

In diesem Fall wird die Leistung einer der beiden Spalten gelöscht und erhält den Wert NULL. Falls eine dritte oder vierte Leistung derselben Paketklasse, zum Beispiel bei einem speziellen Angebot, hinzugefügt werden soll, muss eine neue Spalte hinzugefügt werden, die auf die Leistung-Tabelle referenziert.

Ein letztes Szenario könnte sein, dass ein Auftrag als Absender oder Empfänger den Wert NULL hat. Doch was bedeutet das?

Das bedeutet, dass für den Auftrag noch nicht entschieden wurde welchem Shop dieser Auftrag gehören soll. Jedoch sollte man sich schnell einig werden, da das Bestelldatum existent bleibt und nach 14 Tagen der Auftrag den Status verloren gegangen? erhält. Doch sofern ein Auftrag einen Absender, sprich Shop hat, kann kein Shop einfach gelöscht werden. Das heißt, falls ein Shop gelöscht werden müsste, weil die Umsätze nicht stimmen, dann muss jeder Auftrag, der zu dem Shop gehört erledigt worden sein. Ansonsten geht es nicht. Das Problem jedoch ist, dass die Aufträge nach der Zustellung oder Retoure nicht verschwinden, sondern einen extra Eintrag in den Tabellen „log\_pakete\_zugestellt“ und „log\_retoure“ erhalten mit dem entsprechenden Datum. Dies könnte man anhand eines Stored Procedure lösen, indem man sagt, der Shop mit keinen Aufträgen und weniger als XXX Umsätzen pro Jahr muss geschlossen werden. Doch diese Lösung habe ich in meiner Implementierung nicht berücksichtigt, da ich davon ausgehe, dass kein Shop gelöscht werden müsste, sondern höchstens die Mitarbeiter entlassen werden.

Nun zum Empfänger, der Empfänger ist der jeweilige Kunde, der die Bestellung abgegeben hat und sofern es einen Kunden gibt, der einen Auftrag abgegeben hat, kann dieser in der Kunden-Tabelle nicht gelöscht werden. Die Straße, der Name, etc. vom Kunden kann geändert werden, jedoch manuell.

Das sind ein paar Beispiel Szenarien, die in der Datenbank auftreten können mitsamt ihrem Lösungsweg. Jedoch ist die Datenbank nicht perfekt umgesetzt worden, sodass solche Szenarien einen hohen Aufwand und viel Konzentration mit sich bringen.

**Umsetzung des Modells**

Mithilfe der erarbeiteten Grundlagen, die von Nöten für die Realisierung sind, ist eine Struktur erkennbar, dass durch phpMyAdmin in die Datenbank eingearbeitet wurde.

Letztlich folgen die Tabellen und deren Einzelheiten, wobei die dick gedruckten, die Tabellen sind, dessen Entitäten einen eindeutigen Identifikator erhalten -> z.B. eine ID

**Parcel Express Shop**

die Tabelle enthält,

* eine Shop-ID (Primary Key)
* einen Shopnamen
* einen Shopinhaber (Foreign Key)
* eine Adresse
* eine Telefonnummer
* eine Öffnungszeit
* Shop-ID, falls mehrere Shops denselben Namen haben, um sie eindeutig identifizieren zu können
* Shop-Inhaber, referenziert zu der Tabelle „Miterbeiter“, denn ein Mitarbeiter kann Zusteller, Shop-Inhaber oder vor Ort als Kassierer tätig sein

**Mitarbeiter**

die Tabelle enthält,

* eine MitarbeitNr (Primary Key)
* einen Namen
* eine Telefonnummer
* eine E-Mail
* eine Abteilung
* MitarbeitNr ist quasi die Nummer um Mitarbeiter z.B. mit denselben Namen, Abteilungen,etc zu unterscheiden

**Paketklassev1**

die Tabelle enthält,

* ID (Primary Key) -> zur Identifizierung
* eine Klasse (Foreign Key)
* eine Leistung1 (Foreign Key)
* eine Leistung2 (Foreign Key)
* Um die Leistungen zu bestimmen existiert eine weitere Tabelle „**Leistung**“ mit den jeweiligen Service-Angeboten und deren Preise
* Paketklassev1 referenziert auf „**Paketklassen**“. Diese sind die möglichen Varianten

**Paketklassen**

Die Tabelle enthält,

* eine PaketNr (Primary Key)
* einen Paketname
* eine Maße
* einen Onlinepreis
* einen Shoppreis
* Diese Tabelle gibt Auskunft, welche Paket Varianten möglich ist.

**Leistungen**

die Tabelle enthält,

* eine LeistungID (Primary Key)
* ein Service
* einen Preis -> Was kosten die jeweiligen Service
* Die Tabelle „**Leistungen**“ wie oben erwähnt, ist für die Service-Angebote mit deren Preisen zuständig.

**Auftragv1**

die Tabelle enthält,

* eine AuftragID (Primary Key)
* einen Empfänger (Foreign Key) -> an wen wird der Auftrag gesendet
* einen Absender (Foreign Key) -> von welchem Shop geht der Auftrag aus
* ein Datum -> ist das Bestelldatum
* eine Anzahl der Sendungen -> wie viele Sendungen hat der Auftrag
* ein Status (Foreign Key) -> der momentane Sendungsstatus
* eine Vergütungsgruppe (Foreign Key) -> welche Kategorie wird im Auftrag versendet
* eine Paketklasse (Foreign Key) -> Klasse mit Preisen inklusive Leistungspreise
* AuftragID ist der PrimaryKey und sorgt dafür, dass wenn ein Absender mehrere Pakete absendet diese auch anhand der Nummer zu identifizieren sind
* Absender referenziert auf die ShopID der „**Parcek Express Shop**“ Tabelle um festzustellen von welchem Shop die Pakete versendet werden
* Empfaenger referenziert auf die KundenNr in der Tabele „**Kunde**“ um den Kunden, der das Paket bekommt zu identifizieren
* Status referenziert auf die ZustellNr in der „**Sendungsstatus**“ Tabelle um den momentanigen Status des Aufrgas zu bekommen
* Verguetungsgruppe referenziert auf die Tabelle „**Verguetungsgruppen**“, um festzustellen welcher Auftrag welche Kategorie bzw. Gruppe bedient
* Paketklasse referenziert auf die Tabelle „**preisklassev1“** um die Preise mitsamt den Leistungen zu ermitteln. (Jeder Auftrag kann nur eine Paketklasse haben!-> vereinfachte Idee)

**Kunde**

die Tabelle enthält,

* eine KundenNr (Primary Key) -> Identifikator der Kunden
* einen Namen
* eine Adresse
* eine Telefonnummer
* eine E-Mail
* Diese Tabelle ist eine zufällig gewählte Kundenliste mit der zusätzlichen KundenNr, der auf die **Auftragstabelle** referenziert, um den Empfänger festzustellen

**Bezirk**

Die Tabelle enthält,

* eine BezirkNr (Primary Key) -> Identifikator der Bezirke
* eine Region
* eine Postleitzahl
* einen Ort
* Die Tabelle beinhaltet alle Bezirke, die die Shops liefern. In dem Falle sind es einige aus Hessen und zwei aus Nordrhein-Westfalen (es sollte eine kleine Paketversanddienst Netzwerk darstellen, also nicht vergleichbar mit der DHL, etc!)
* Jeder zusteller ist einem speziellen Gebiet zugeordnet, jedoch kann es bei unmengen an Lieferungen dazu kommen, dass zwei Zusteller einen Bezirk abdecken müssen, daher das Referenzieren von ZustellBezirk aus der **Zustellertabelle** mit dieser hier.

**Vergütungsgruppe**

Die Tabelle enthält,

* eine VergütungsID (Primary Key) -> Identifikator der Vergütungen
* eine Vergütungsname
* Gibt mithilfe einer ID an welche Gruppen existieren, die die Kunden erwerben können.

Grundsätzlich haben alle Shops alle Vergütungsgruppen auf Lager.

**Sendungsstatus**

Die Tabelle enthält,

* Eine ZustellNr (Primary Key) -> Identifikator vom Status der Lieferung
* Eine Zustellstatus
* Gibt die Auswahl der Statusliste insgesamt an.

**Zusteller**

die Tabelle enthält,

* ID (Primary Key) -> zur Identifizierung
* eine ZustellNr (Foreign Key)
* einen Zustellbezirk (Foreign Key)
* eine AuftragsNr (Foreign Key)
* Ist der Lieferant des jeweiligen Shops, das auf die **Mitarbeiter,- Bezirk-** und **Auftragv1tabelle** referenziert.

**Abteilung**

Die Tabelle enthält,

* ID (Primary Key)
* Beschreibung
* Die Tabelle dient dazu die Abteilungen der Mitarbeiter zugeben.

**Log\_Pakete\_zugestellt**

Die Tabelle enthält,

* eine LogID (Primary Key)
* eine AuftragNr (Foreign Key)
* einen Datum -> Zustelldatum
* Wenn ein Paket den Status in „zugestellt“ ändert, wird dieser in der Tabelle mit seiner ID, seiner Auftragnummer und Zustelldatum dokumentiert.

**Log\_retoure**

Die Tabelle enthält,

* eine Retoure-ID (Primary Key)
* eine AuftragsNr (Foreign Key)
* ein Retouredatum
* Wenn ein Paket den Status „Retoure“ erhält wird dieser in der Tabelle mitsamt einer ID, dem Auftragsnummer und dem Retouredatum dokumentiert

**SQL – Abfragen**

1. [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) mitarbeiter.Name FROM auftragv1, mitarbeiter JOIN zusteller ON mitarbeiter.MitarbeitNr = zusteller.ZustellNr WHERE auftragv1.AuftragID = 1 [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) auftragv1.AuftragID = zusteller.AuftragNr GROUP BY mitarbeiter.Name

* Filtert den Namen des Mitarbeiters, der für die Auftrag-Nummer XY zuständig ist.

1. SELECT auftragv1.AuftragID, sendungsstatus.Zustellstatus

FROM auftragv1, sendungsstatus

WHERE auftragv1.AuftragID = 1 AND sendungsstatus.ZustellNr = auftragv1.Status

GROUP BY auftragv1.AuftragID;

* Sendungsstatus eines bestimmten Auftrags.

1. SELECT paketklassen.Paketname, paketklassen.Onlinepreis, paketklassen.Shoppreis

FROM paketklassev1, paketklassen

WHERE paketklassev1.Klasse = paketklassen.PaketNr

GROUP BY paketklassev1.Klasse;

* Angabe der Paketklassen mit den Preisen.

1. SELECT mitarbeiter.Name

FROM zusteller, bezirk, mitarbeiter

WHERE zusteller.ZustellNr = mitarbeiter.MitarbeitNr AND zusteller.Zustellbezirk = bezirk.BezirkNr AND bezirk.PLZ = 34117  
GROUP BY mitarbeiter.Name;

* Filtert den Zusteller bei Namen für den Bezirk bzw PLZ XY.

1. SELECT mitarbeiter.Shopinhaber

FROM parcel\_express\_shop, mitarbeiter

WHERE parcel\_express\_shop.Shopinhaber = mitarbeiter.MitarbeitNr;

* Alle Shopinhaber der Mitarbeitertabelle.

1. SELECT parcel\_express\_shop.Shopinhaber AS ID, mitarbeiter.Name, parcel\_express\_shop.Shopname

FROM parcel\_express\_shop, mitarbeiter

WHERE parcel\_express\_shop.ShopID = 2 AND parcel\_express\_shop.Shopinhaber = mitarbeiter.MitarbeitNr;

* Shopinhaber einer bestimmten Filiale XY mit der ID in der Mitarbeitertabelle.



1. Select PaketNr, Paketname, (COUNT(LT1.Service)+COUNT(LT2.Service)) AS Anzahl\_Leistungen, (paketklassen.Onlinepreis+(LT1.Preis+LT2.Preis)) AS Gesamtpreis\_Online, (paketklassen.Shoppreis+(LT1.Preis+LT2.Preis)) AS Gesamtpreis\_Shop  
   FROM paketklassen   
   JOIN paketklassev1 ON paketklassen.PaketNr=paketklassev1.Klasse   
   JOIN leistung LT1 ON LT1.LeistungID=paketklassev1.Leistung1   
   JOIN leistung LT2 ON LT2.LeistungID=paketklassev1.Leistung2 AND LT1.LeistungID<>LT2.LeistungID   
   WHERE paketklassen.PaketNr= '1'

* Ausgabe der Gesamtpreise einer Paketklasse inklusive den Serviceanzahl für Online -und Shoppreis. 

1. SELECT kunde.Name, kunde.Adresse

FROM kunde, auftragv1

WHERE kunde.KundenNr = auftragv1.Empfaenger AND auftragv1.AuftragID = 1;

* Gibt den Kundennamen und seine Adresse an für seinen Auftrag.

1. SELECT kunde.Name, kunde.Adresse, SUM(auftragv1.Anzahl\_Sendungen)AS Anzahl\_Pakete

FROM kunde, auftragv1

WHERE kunde.KundenNr = auftragv1.Empfaenger AND auftragv1.AuftragID = 3;

* Anzahl der Pakete bzw Sendungen eines Kunden für eine bestimmten Auftrag.

1. SELECT COUNT(auftragv1.Status) AS Auftrag\_der\_Retoure

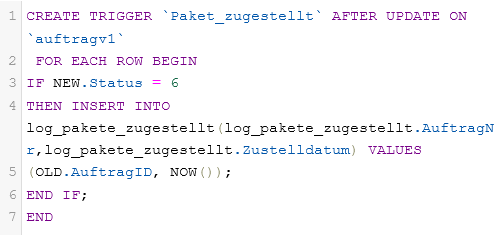
FROM auftragv1, sendungsstatus

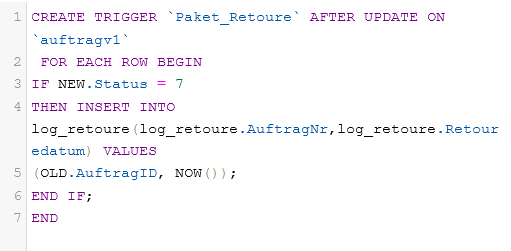
WHERE sendungsstatus.Zustellstatus = 'Retoure' AND auftragv1.Status = sendungsstatus.ZustellNr;

* Gesamtzahl aller Retouren aller Shops.

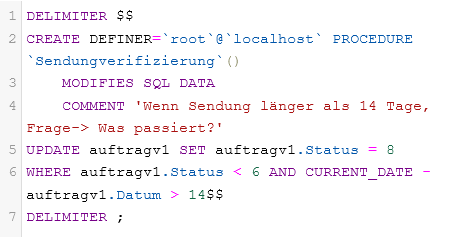
1. [SELECT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/select.html) log\_retoure.Retouredatum AS Datum FROM log\_retoure [LEFT](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/string-functions.html#function_left) JOIN auftragv1 ON log\_retoure.AuftragNr = auftragv1.AuftragID [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) auftragv1.Status = 'Retoure' [AND](http://localhost/phpmyadmin/url.php?url=https://dev.mysql.com/doc/refman/5.5/en/logical-operators.html#operator_and) log\_retoure.RetoureID = 1

* Ausgabe vom Datum der Retoure.

**Trigger**

* Der Trigger dokumentiert alle Änderungen der Aufträge in den Status zugestellt (=6) mitsamt ihrer AuftragsID sowie das Zustelldatum, das in der Tabelle log\_pakete\_zugestellt gespeichert wird.
* Analog die Änderungen der Aufträge in den Status retoure(=7) mitsamt ihrer AuftragsID sowie Retoredatum, das in der Tabelle log\_retoure gespeichert wird.

**Stored Procedure**



Schaut in der Tabelle Auftrag, ob ein Auftrag länger als 2 Wochen dauert und nicht zugestellt oder als Retoure betitelt wurde, wird sich gefragt ob sie verloren gegangen ist.

**Bemerkungen und Kritik an der Umsetzung**

Primär galt unser Vorhaben der sinnvollen und schlüssigen Modellierung der Aufgabenstelle. Die Kriterien für die Umsetzung der Modellierung waren zumal die Zwecke der Normalisierung einzuhalten sowie die referenzielle Integrität zu beachten. Letzteres sorgt für die Funktionalität der Abläufe einer solchen imaginären Paketversanddienst, wenn beispielsweise ein Auftrag erfüllt und somit das Paket zugestellt wurde oder ein Mitarbeiter gekündigt wurde. All dies wurde mithilfe von Fremdschlüsselbeziehungen abgesichert.

Die Funktionen von Trigger und Stored Procedure wurden sinnvoll der Realität entsprechend in die Datenbank integriert. Ohne diese beiden Integrationen würde die Datenbank im Wesentlichen funktionieren, jedoch würde die realitätsnahe Umsetzung fehlen. Sie unterschreiben eine Besonderheit des Modells.

Ein grundsätzlicher Kritikpunkt des Modells besteht darin, dass die Kunden keine Rechnungsposition über ihre Bestellungen erfahren, sowie keine Benachrichtigung per E-Mail über ihre Sendung erhalten, das heutzutage in der Realität gang und gäbe geworden ist. Ebenso ist ein wesentlicher Kritikpunkt, dass die Bezirke, die geliefert werden, enorm beschränkt ist. Ein realitätskonformer Paketversanddienst würde im ganzen Land liefern.

Ein weiterer Kritikpunkt liefert die Tabelle „Verguetungsgruppe“. Dieser ist begrenzt an den möglichen Gruppen, die von den Kunden erworben werden können. Jedoch will ich an der Stelle anmerken, dass die Firma bzw. die Dienstleistung eine neugeründete und Kleine darstellen soll. Sie ist keineswegs vergleichbar mit der DHL oder ähnlichem.

Genug mit der Kritik an der Modellierung. Die positiven Aspekte des Modells sind die realitätsnahe Modellierung, die Vielfältigkeit sowie die Funktionsfähigkeit im realen Gebrauch. Die Tabellen „log\_pakete\_zugestellt“ und „log\_retoure“ bieten eine super Übersicht über die bereits fertigen Aufträge, sowie die Aufträge die als Retoure gelten. Ohne eine solche Dokumentation würde ein Chaos eintreffen.  
Genauso bietet der Procedure eine gute Rückmeldung über Aufträge, die nach zwei Wochen noch immer nicht den Status „zugestellt“ oder „retoure“ erhalten haben, denn so wird sich gefragt was mit dieser Lieferung passiert ist.  
Durch die ausgedachten SQL-Befehle lässt sich schließen, dass diese Modellierung in der Realität eine Anwendung finden würde.

Abschließend möchte ich erwähnen, dass für die Datenbank einige weitere Verbesserungen gemacht werden können. Anhand der Kritikpunkte wird verdeutlicht, dass nicht alles zu 100% funktioniert, jedoch als Grundgerüst eine sehr gute Schablone dient.