

Projektaufgabe E-Bike-Shops

**Teil der Prüfungsleistung im Modul
Datenbanken (EDV-Nr. 113222)
Sommersemester 2022**

Projektgruppe <Gruppennummer> mit den Gruppenteilnehmer:

<Belge>, <Sinem>	<41956>	<Medieninformatik>
<Gerhardt>, <Gabriela>	<41938>	<Mobile Medien>
<Knoblauch>, <Teje>	<42330>	<Mobile Medien>
<Seifert>, <Fiona>	<41936>	<Mobile Medien>

Hiermit bestätigen die oben genannten Gruppenteilnehmer, dass die Projektaufgabe E-Bike-Shop ohne Unterstützung dritter und unter Nennung aller Quellen und eingesetzten Hilfsmittel erstellt wurde. Es ist uns bekannt, dass ein Verstoß gem. § 17 Abs. 5 Satz 1 zum Nichtbestehen der kompletten Prüfungsleistung im Modul Datenbanken (EDV-Nr. 113222) führt.

Stuttgart, den

<Sinem Belge>

<Gabriela Gerhardt>

<Teje Knoblauch>

<Fiona Seifert>

Inhalt

1. Entwurfsentscheidung	3
2. Szenario	3
3. ER-Diagramme	4
3.1 Physisches ER-Diagramm	4
3.2 Logisches ER -Diagramm	5
4. Systembeschreibung	6
5. SQL Skripte	9
-- DROP Constraints und Tabellen	9
-- Anlegen der Tabelle	9
-- Einfügen von CONSTRAINTS	10
-- UPDATE TRIGGER	11
-- INSERT TRIGGER	11
-- Insert Befehle	11
-- löst Update Trigger aus	12
Bewertungsschema	13

1. Entwurfsentscheidung

Der Web-shop 'ME-Bikes' stellt personalisierte E-Bikes her.

Der Nutzer kann zwischen:

- Verschiedenen Modellen
- E-Bike Akkus
- Farben
- Schriftzügen
- Reifen

wählen.

2. Szenario

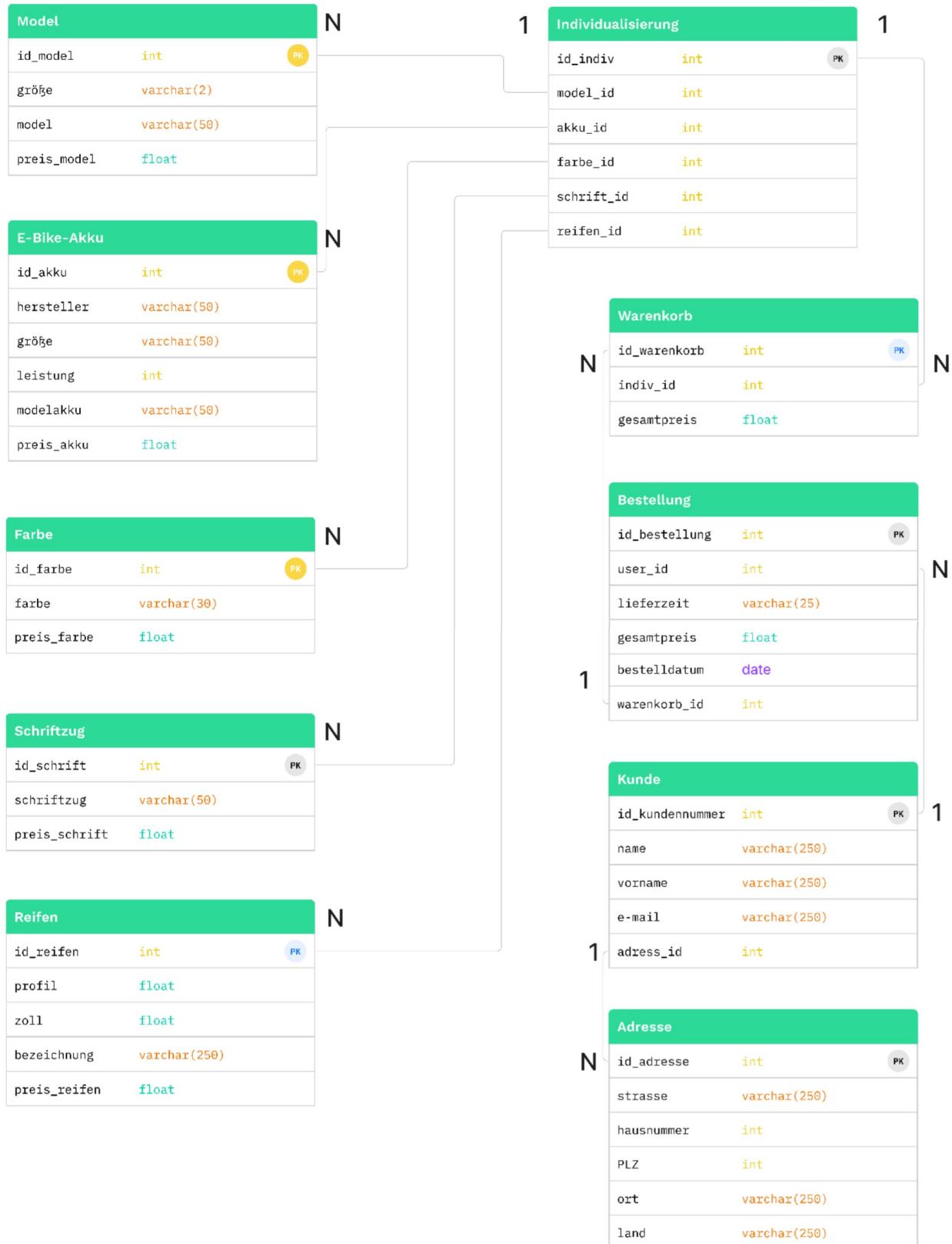
Der Shop 'ME-Bikes' hat sich nach vermehrten Schwierigkeiten durch Corona und vielfach sinkender Kundenzahlen dazu entschieden auf einen Webshop umzusteigen.

Da die Verkaufszahlen allerdings schon längere Zeit eher sinken als steigen, ist der Geschäftsführer auf die Idee gekommen, ausschließlich personalisierte E-Bikes zu verkaufen.

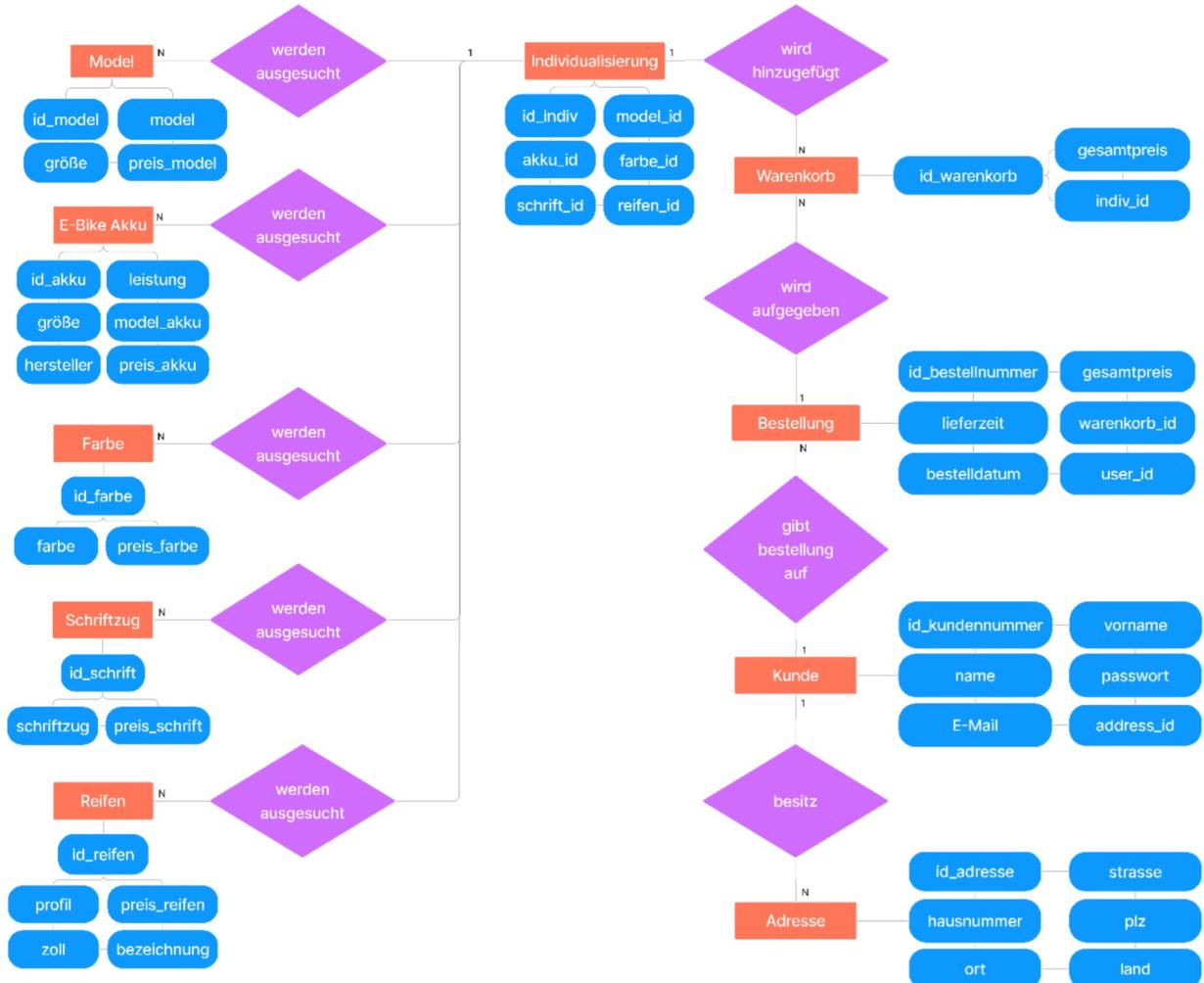
Durch den wachsenden Trend von personalisierten Gegenständen, hofft der Geschäftsführer auf steigende Umsätze.

3. ER-Diagramme

3.1 Physisches ER-Diagramm



3.2 Logisches ER -Diagramm



4. Systembeschreibung

Das ER-Modell kann in zwei Blöcke unterteilt werden, welche die Hauptfunktionen des E-Bike Web-shops beschreiben. Auf diese zwei Blöcke wird im Folgenden darauf eingegangen.

Block 1: Sein individuelles E-Bike zusammenstellen

Der Kunde xy hat von seinem Freund mitbekommen, dass der Webshop ME-Bikes personalisierte E-Bikes herstellt. Da er sich schon seit längerem ein E-Bike zulegen möchte, beschließt er, dass dies die perfekte Möglichkeit ist, da die Idee ihm richtig gefällt.

- Er öffnet die Seite des Webshops und kann sich als erstes zwischen verschiedenen Modellen und deren Größe entscheiden. Bezuglich der Größe kann man wählen zwischen 'S', 'M', 'L' und 'XL'. Dies haben wir mit einem Check Constraint umgesetzt.
- Als nächstes muss er sich einen Akku für sein E-Bike aussuchen. Hierbei spielen verschiedene Faktoren wie: Leistung, Größe, Hersteller, Model und dem Preis eine Rolle.
- Danach kann er sich für eine Farbe entscheiden und ihm wird direkt angezeigt wie viel Aufpreis für die Farbe entstehen würde.
- Er kann sich für sein E-Bike auch einen personalisierten Schriftzug überlegen. Zuerst kann er die Anzahl der Buchstaben aussuchen um den Aufpreis zu erfahren, da der Aufpreis pro Buchstabe berechnet wird. Bei dem Abschluss der Bestellung kann er den gewünschten Schriftzug dann als Kommentar einfügen.
- Letztendlich kann er sich noch die passenden Reifen zu seinem E-Bike aussuchen die z.B. verschiedene Profile, Größen, Preise und Bezeichnungen haben.

Um die fertige Version des E-Bikes mit seinen ausgewählten Modifikationen visualisieren zu können, haben wir die Tabelle 'Individualisierung' erstellt.

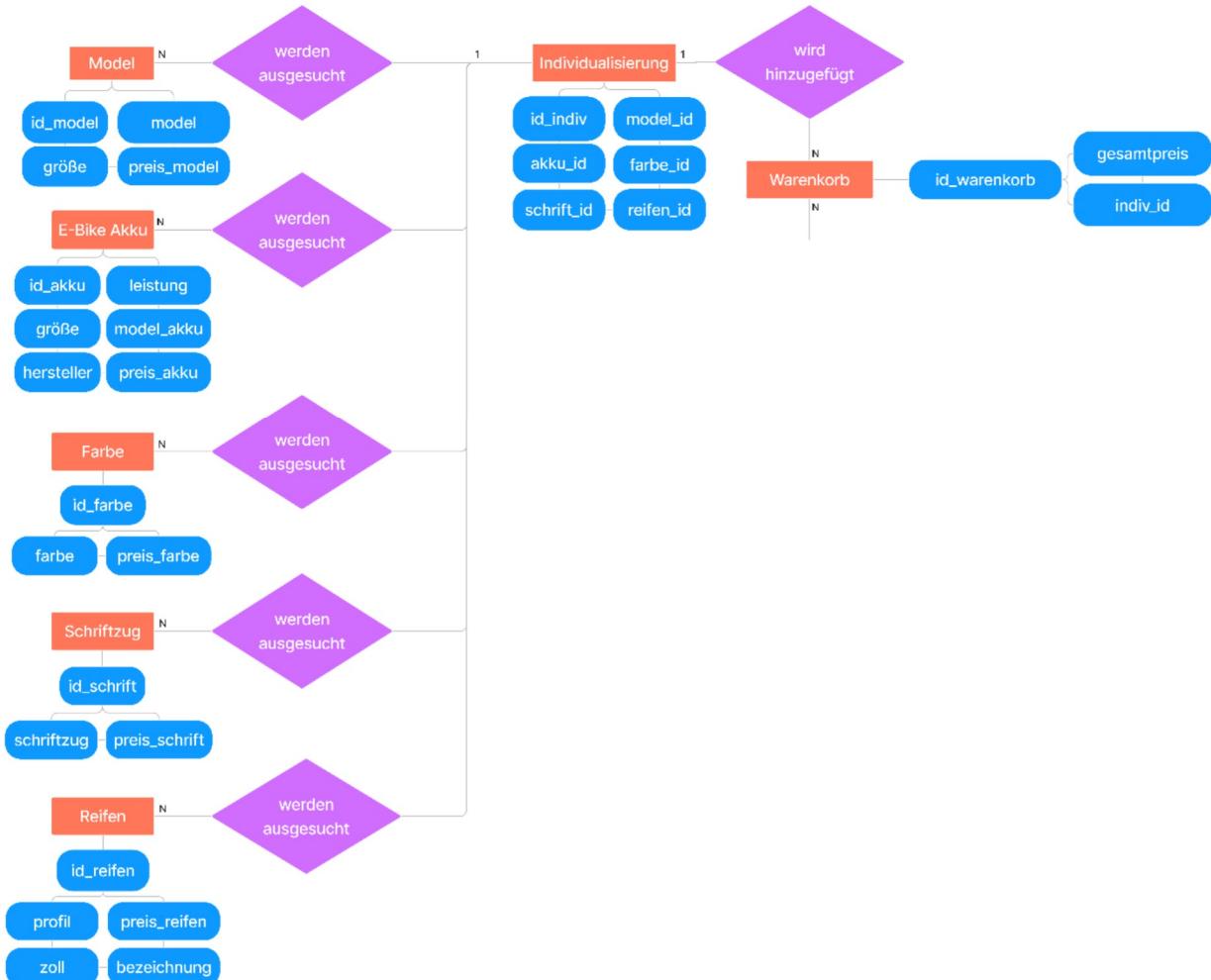
Die Tabelle fasst die ID's der einzelnen Individualisierungen (Modell, E-Bike Akku, Farbe, Schriftzug und Reifen).

Betrachten wir die Tabelle genauer haben wir zwei Defaults. Einmal für die Schrift: wenn jemand keinen Schriftzug möchte, muss er nichts angeben und dann wird automatisch kein Schriftzug und somit kein Aufpreis verlangt.

Den zweiten Default haben wir bei der Farbe. Wenn der Kunde keine Farbe wählt, wird automatisch der Standard 'Silber' ausgewählt.

Ursprünglich wollten wir den Gesamtpreis automatisch berechnen lassen, dies konnten wir allerdings trotz intensiver Recherche leider nicht umsetzen. Deswegen haben wir uns dazu entschlossen, dass dieser dann eben händisch ausgerechnet werden muss.

Gefällt dem Kunden das Endergebnis kann er es dann in den Warenkorb tun.



Zweiter Block: Kundenkonto und Bestellung

Nachdem der Kunde sein personalisiertes E-Bike in den Warenkorb getan hat und sich entscheiden hat es zu kaufen muss er sich einen Account bei uns erstellen.

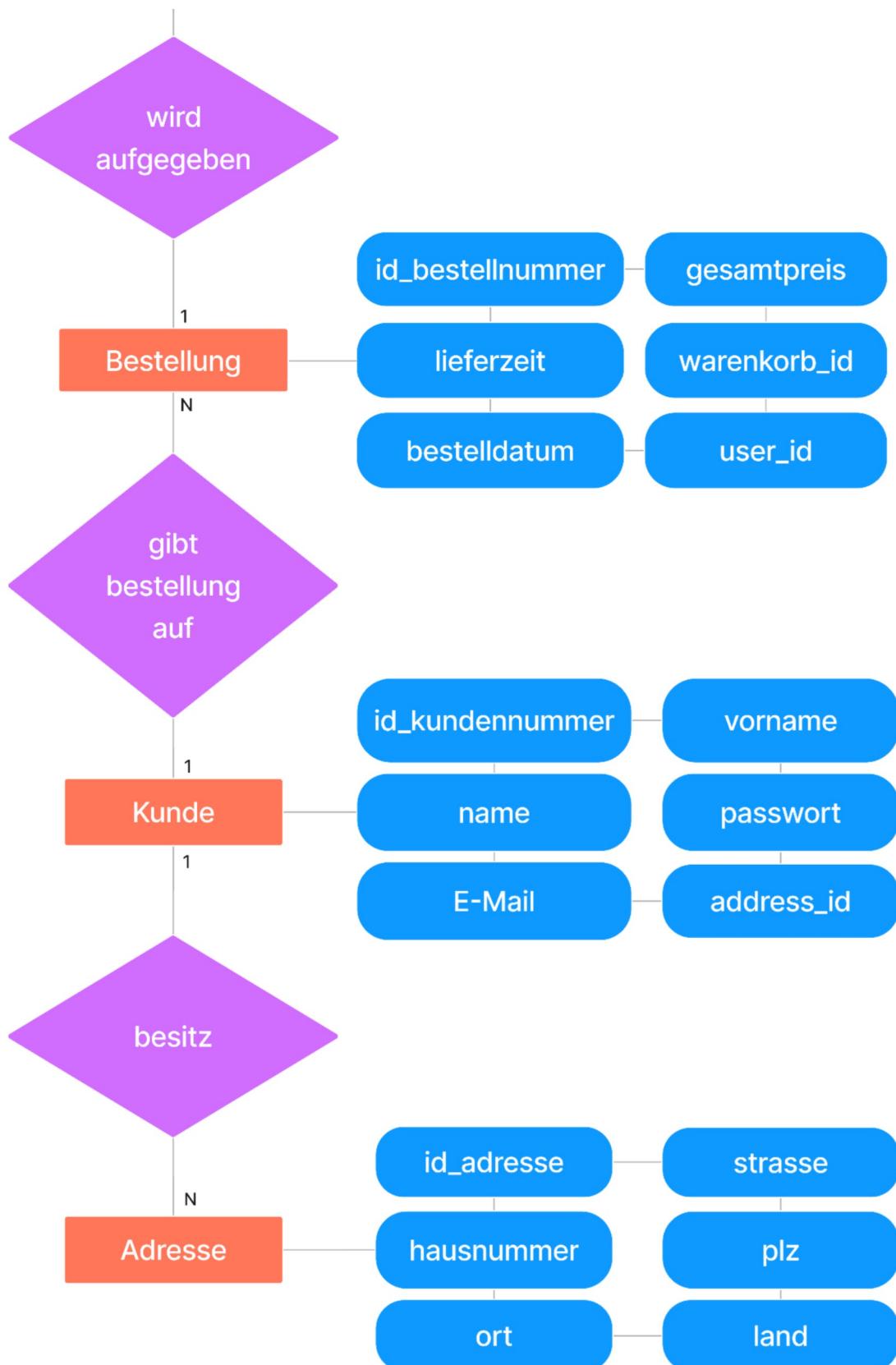
Hierbei muss er folgende Daten angeben:

Name, Vorname und E-Mail. Das Erstelltdatum wird über einen Insert-Trigger hinzugefügt. Die Adresse wird über eine externe Tabelle 'Adresse' abgespeichert, um einen einfacheren Zugang für die anderen Tabellen zu ermöglichen.

Nach dem dieser Anmelde Prozess abgeschlossen ist, kann der Kunde seine Bestellung aufgeben. In der Bestellung steht die Bestellnummer drinnen, der Kunde, der es bestellt hat, der Warenkorb inhalt, Bestelldatum und Gesamtpreis.

Die Bestellung inkludiert ebenfalls die Lieferzeit damit der Kunde weiß, wann sein 'ME-Bike' voraussichtlich ankommt.

Bei der Bezahlung des ME-Bikes gibt es für jede erste Bestellung einen Rabattcode, den man dann als E-Mail bekommt und den bei der ersten Bestellung einlösen kann. Dies haben wir mit einem Update- Trigger umgesetzt.



5. SQL Skripte

```
-- DROP Constraints und Tabellen
ALTER TABLE individualisierung DROP CONSTRAINT model_fk;
ALTER TABLE individualisierung DROP CONSTRAINT schrift_fk;
ALTER TABLE individualisierung DROP CONSTRAINT akku_fk;
ALTER TABLE individualisierung DROP CONSTRAINT farbe_fk;
ALTER TABLE individualisierung DROP CONSTRAINT reifen_fk;
ALTER TABLE warenkorb DROP CONSTRAINT indiv_fk;
ALTER TABLE kunde DROP CONSTRAINT adresse_fk;
ALTER TABLE bestellung DROP CONSTRAINT warenkorb_fk;
ALTER TABLE bestellung DROP CONSTRAINT user_fk;

ALTER TABLE model DROP CONSTRAINT model_key;
ALTER TABLE ebikeakku DROP CONSTRAINT akku_key;
ALTER TABLE farbe DROP CONSTRAINT farbe_key;
ALTER TABLE schriftzug DROP CONSTRAINT schrift_key;
ALTER TABLE reifen DROP CONSTRAINT reifen_key;
ALTER TABLE individualisierung DROP CONSTRAINT indiv_key;
ALTER TABLE warenkorb DROP CONSTRAINT warenkorb_key;
ALTER TABLE adresse DROP CONSTRAINT adresse_key;
ALTER TABLE kunde DROP CONSTRAINT kunde_key;
ALTER TABLE bestellung DROP CONSTRAINT bestellung_key;

ALTER TABLE model DROP CONSTRAINT model_Check;
ALTER TABLE adresse DROP CONSTRAINT adresse_Check;

drop table bestellung;
drop table kunde;
drop table adresse;
drop table warenkorb;
drop table individualisierung;
drop table reifen;
drop table schriftzug;
drop table farbe;
drop table ebikeakku;
drop table model;

-- Anlegen der Tabelle
create table model (
    id_model INT,
    groesse VARCHAR(2) NOT NULL,
    model VARCHAR(50) NOT NULL,
    preis_model FLOAT NOT NULL
);

create table ebikeakku (
    id_akku INT,
    hersteller VARCHAR(50) DEFAULT 'N.N',
    groesse VARCHAR(100) NOT NULL,
    leistung INT NOT NULL,
    modelakku VARCHAR(50) NOT NULL,
    preis_akku FLOAT NOT NULL
);

create table farbe (
    id_farbe INT,
    farbe VARCHAR(30),
    preis_farbe FLOAT NOT NULL
);

create table schriftzug (
    id_schrift INT,
```

```

schriftzug VARCHAR(50),
preis_schrift FLOAT
);

create table reifen (
id_reifen INT,
profil FLOAT,
zoll FLOAT,
bezeichnung varchar(250),
preis_reifen FLOAT NOT NULL
);

create table individualisierung (
id_indiv INT,
model_id INT NOT NULL,
schrift_id INT DEFAULT '1',
akku_id INT NOT NULL,
farbe_id INT DEFAULT '1',
reifen_id INT NOT NULL,
gesamtpreis INT NOT NULL
);

create table warenkorb (
id_warenkorb INT,
indiv_id INT NOT NULL,
gesamtpreis float not null
);

create table adresse (
id_adresse int,
strasse varchar(250) not null,
hausnummer int not null,
plz int not null,
ort VARCHAR(250) not null,
land VARCHAR(250) DEFAULT 'Deutschland' not null
);

create table Kunde (
id_kundennummer int,
name varchar(250) not null,
vorname varchar(250) not null,
email varchar(250) not null,
adress_id int NOT NULL,
erstell_datum DATE not null
);

create table bestellung (
id_bestellnummer int,
user_id int not null,
lieferzeit varchar(25) not null,
warenkorb_id int not null,
bestelldatum date not null,
gesamtpreis float not null,
rabatt_code varChar(25)
);

-- Einfügen von CONSTRAINTS
ALTER TABLE model ADD CONSTRAINT model_Check CHECK (groesse IN
('S','M','L','XL'));
ALTER TABLE adresse ADD CONSTRAINT adresse_Check CHECK (land IN ('Deutschland',
'Sweden', 'Finnland', 'England', 'Österreich'));

ALTER TABLE model ADD CONSTRAINT model_key PRIMARY KEY (id_model);
ALTER TABLE ebikeakku ADD CONSTRAINT akku_key PRIMARY KEY (id_akku);

```

```

ALTER TABLE farbe ADD CONSTRAINT farbe_key PRIMARY KEY (id_farbe);
ALTER TABLE schriftzug ADD CONSTRAINT schrift_key PRIMARY KEY (id_schrift);
ALTER TABLE reifen ADD CONSTRAINT reifen_key PRIMARY KEY (id_reifen);
ALTER TABLE individualisierung ADD CONSTRAINT indiv_key PRIMARY KEY (id_indiv);
ALTER TABLE warenkorb ADD CONSTRAINT warenkorb_key PRIMARY KEY (id_warenkorb);
ALTER TABLE adresse ADD CONSTRAINT adresse_key PRIMARY KEY (id_adresse);
ALTER TABLE kunde ADD CONSTRAINT kunde_key PRIMARY KEY (id_kundennummer);
ALTER TABLE bestellung ADD CONSTRAINT bestellung_key PRIMARY KEY
(id_bestellnummer);

ALTER TABLE individualisierung ADD CONSTRAINT model_fk FOREIGN KEY (model_id)
REFERENCES model(id_model);
ALTER TABLE individualisierung ADD CONSTRAINT schrift_fk FOREIGN KEY
(schrift_id) REFERENCES schriftzug(id_schrift);
ALTER TABLE individualisierung ADD CONSTRAINT akku_fk FOREIGN KEY (akku_id)
REFERENCES ebikeakku(id_akku);
ALTER TABLE individualisierung ADD CONSTRAINT farbe_fk FOREIGN KEY (farbe_id)
REFERENCES farbe(id_farbe);
ALTER TABLE individualisierung ADD CONSTRAINT reifen_fk FOREIGN KEY (reifen_id)
REFERENCES reifen(id_reifen);
ALTER TABLE warenkorb ADD CONSTRAINT indiv_fk FOREIGN KEY (indiv_id) REFERENCES
individualisierung(id_indiv);
ALTER TABLE kunde ADD CONSTRAINT adresse_fk FOREIGN KEY (adresse_id) REFERENCES
adresse(id_adresse);
ALTER TABLE bestellung ADD CONSTRAINT warenkorb_fk FOREIGN KEY (warenkorb_id)
REFERENCES warenkorb(id_warenkorb);
ALTER TABLE bestellung ADD CONSTRAINT user_fk FOREIGN KEY (user_id) REFERENCES
kunde(id_kundennummer);

-- UPDATE TRIGGER
CREATE OR REPLACE TRIGGER Update_Trigger_Two BEFORE UPDATE ON bestellung
REFERENCING NEW AS NEW OLD AS OLD
FOR EACH ROW
BEGIN
:NEW.gesamtpreis := :OLD.gesamtpreis - 149.99;
END;
/

-- INSERT TRIGGER
CREATE OR REPLACE TRIGGER Insert_Trigger_Two BEFORE INSERT ON kunde
REFERENCING NEW AS NEW OLD AS OLD
FOR EACH ROW
BEGIN
:NEW.erstell_datum := SYSDATE;
END;
/

-- Insert Befehle
INSERT INTO model VALUES (1, 'S', 'citybike', 500.00);
INSERT INTO model VALUES (2, 'M', 'citybike', 999.98);
INSERT INTO model VALUES (3, 'L', 'citybike', 1299.99);
INSERT INTO model VALUES (4, 'L', 'mountainbike', 2350.00);
INSERT INTO model VALUES (5, 'S', 'speedbike', 1865.99);

INSERT INTO ebikeakku VALUES (1, 'Bosch', '325 mm x 92 mm x 90 mm', '500',
'PowerPack', 729.00);
INSERT INTO ebikeakku VALUES (2, 'AkkuBikes', '273 mm x 106 mm x 153 mm', '450',
'PowerPack', 289.00);
INSERT INTO ebikeakku VALUES (3, 'Yamaha', '372 x 98,2 x 98,4mm', '745',
'PowerPack', 749.95);

INSERT INTO farbe VALUES (1, 'silber', 0.00);
INSERT INTO farbe VALUES (2, 'rot', 5.99);

```

```

INSERT INTO farbe VALUES (3, 'schwarz', 5.99);
INSERT INTO farbe VALUES (4, 'schwarz + matt', 15.00);
INSERT INTO farbe VALUES (5, 'pink', 5.99);
INSERT INTO farbe VALUES (6, 'blau', 5.99);
INSERT INTO farbe VALUES (7, 'silber + metallic', 15.00);

INSERT INTO schriftzug VALUES (1, 'Kein Schriftzug', 0.00);
INSERT INTO schriftzug VALUES (2, '5', 10.99);
INSERT INTO schriftzug VALUES (3, '6', 11.99);
INSERT INTO schriftzug VALUES (4, '7', 12.99);
INSERT INTO schriftzug VALUES (5, '8', 13.99);
INSERT INTO schriftzug VALUES (6, '9', 14.99);
INSERT INTO schriftzug VALUES (7, '10', 15.99);

INSERT INTO reifen VALUES (1, 1.60, 28.0, 'Performance Reflex', 20.99);
INSERT INTO reifen VALUES (2, 1.65, 29.0, 'Double Safety System Breaker',
35.89);
INSERT INTO reifen VALUES (3, 1.55, 2.2, 'Energizer +', 54.99);

INSERT INTO adresse VALUES (1, 'New York Ring', 25, 71686, 'Remseck am Neckar',
'Deutschland');
INSERT INTO adresse (id_adresse, strasse, hausnummer, plz, ort)
VALUES (2, 'Floschweg', 15, 71665, 'Vaihingen an der Enz');
INSERT INTO adresse VALUES (3, 'Viksvägen', 24, 45795, 'Kämpersvik', 'Sweden');

INSERT INTO individualisierung VALUES (1, 5, 6, 1, 2, 1, 2636.96);
INSERT INTO individualisierung VALUES (2, 3, 2, 2, 1, 2, 1635.87);
INSERT INTO individualisierung VALUES (3, 4, 7, 1, 3, 3, 3155.97);
INSERT INTO individualisierung VALUES (4, 3, 7, 2, 3, 3, 1665.96);

INSERT INTO warenkorb VALUES (1, 1, 2636.96);
INSERT INTO warenkorb VALUES (2, 3, 3155.97);
INSERT INTO warenkorb VALUES (3, 2, 1635.87);

INSERT INTO kunde (id_kundennummer, name, vorname, email ,adress_id) VALUES
(130940, 'Muster', 'Annalena', 'annalena.muster@gmail.com', 1);
INSERT INTO kunde (id_kundennummer, name, vorname, email ,adress_id) VALUES
(2242, 'Depp', 'Johnny', 'Johnny.Depp@gmail.com', 2);
INSERT INTO kunde (id_kundennummer, name, vorname, email ,adress_id) VALUES
(35673, 'Sky', 'Luke', 'SkyLuke@web.com', 3);

INSERT INTO bestellung (id_bestellnummer, user_id, lieferzeit, warenkorb_id,
bestelldatum, gesamtpreis) VALUES (13674, 130940, '3-4 Wochen', 3, '02.06.2022',
1635.87);
INSERT INTO bestellung (id_bestellnummer, user_id, lieferzeit, warenkorb_id,
bestelldatum, gesamtpreis) VALUES (13784, 2242, '3-4 Wochen', 2, '03.06.2022',
3155.97);
INSERT INTO bestellung (id_bestellnummer, user_id, lieferzeit, warenkorb_id,
bestelldatum, gesamtpreis) VALUES (13567, 35673, '4-5 Wochen', 1, '03.06.2022',
2636.96);

-- löst Update Trigger aus
UPDATE bestellung SET rabatt_code = '123NEW' WHERE warenkorb_id = 3;

```

Bewertungsschema (bitte als letzte Seite der Ausarbeitung einbinden):

Kriterium	max.	ist	Kommentar
Originalität	10		
Relative Qualität	15		
Spezifikation / Systembeschreibung	15		
ER-Modell inkl. formaler Korrektheit im Sinn der Normalisierung	10		
Sinnvolle Nutzung von Normierungsdaten	5		
Lauffähigkeit	5		
Restartfähigkeit	5		
Vollständigkeit der Testdaten	5		
Umfang der Umsetzung des Datenmodells	-	-	
Anzahl der Tabellen	5		
Vollständige Nutzung der Datentypen	5		
Sinnvolle Nutzung von Randbedingungen	5		
Sinnvolle Nutzung von Indexierung	5		
Insert-Trigger	5		
Update-Trigger	5		
Summe			