

## Klausur Datenbanksysteme, Lösungen

### Aufgaben (Punkte ohne Gewähr)

#### Aufgabe 1: Entity-Relationship-Diagramm

(33)

Eine Autoverleihfirma will eine Datenbank verwenden. Erstellen Sie ein ER-Diagramm, für diese Datenbank. Achten sie auf darauf, ...

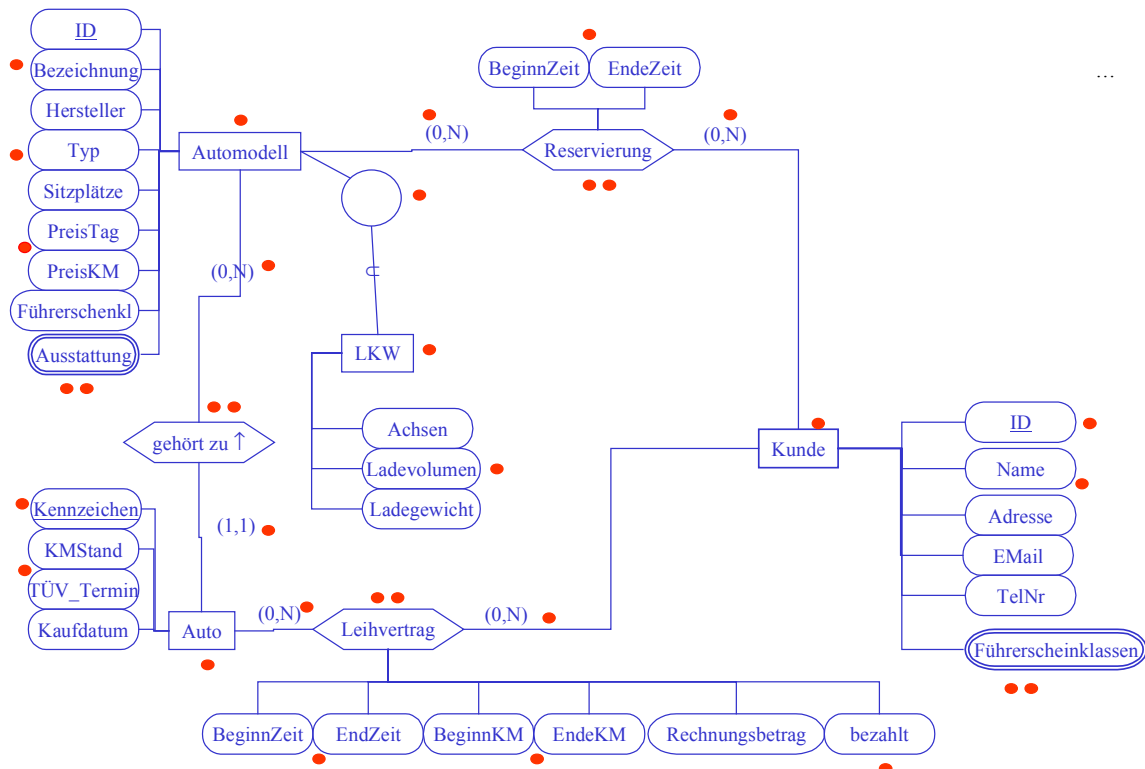
- dass Sie ein ER-Diagramm erstellen sollen und nicht ein Diagramm einer relationalen Tabellenstruktur, d.h. M:N-Relationen sind noch nicht in Verbindungstabellen umgewandelt, es gibt mehrwertige Attribute, und eine Strukturierung in Unterentitäten ist möglich.
- dass Sie Anomalien und Redundanzen vermeiden.

Folgende Informationen sollen aus der Datenbank effizient abgefragt werden können (Informationen, die hier nicht genannt sind, brauchen auch nicht ins ER-Diagramm, selbst wenn Sie sie für eine Datenbank einer Autoverleihfirma für sinnvoll erachten.):

- Suche nach allen Kunden, z. B. um Werbematerial verschicken zu können.
- Suche nach allen Automodellen des Verleihers; als Suchbedingung kann angegeben werden der Hersteller, der Typ (z. B. Van, LKW, Kombi, Cabrio), die Anzahl der Sitzplätze den Preis pro Tag, den Preis pro Kilometer oder Ausstattungsdetails (z. B. Klimaanlage, Navigationsystem, ESP). Dabei kann ein Automodell beliebig viele Ausstattungsdetails haben.. Bei LKWs ist zusätzlich die Zahl der Achsen, das Ladevolumen und das Ladegewicht relevant.
- Suche nach allen Autos des Verleihers. Der Verleiher will dabei als Suchbedingungen eventuell angeben das Kennzeichen, den Kilometerstand, das Datum des letzten TÜV-Termins, das Kaufdatum oder das Automodell.
- Auch eine Suche nach einer Kombination von Eigenschaften von Automodellen und Autoexemplaren soll möglich sein, z. B. die Suche nach den Kilometerständen und TÜV-Terminen aller Cabrios.
- Kunden können sich online ein Automodell (nicht ein konkretes Auto) reservieren. Dazu müssen Sie den Zeitraum angeben, in dem sie ein Fahrzeug leihen wollen, z. B. vom 14.7.02, 9:00 bis 16.7.02, 17:00.
- Bei jeder Reservierung soll im Datenbanksystem automatisch geprüft werden können, ob die nötige Führerscheinklasse für das Automodell in den Führerscheinklassen des Kunden enthalten ist.
- Während die Reservierung noch unverbindlich bleibt und sich nur auf ein Automodell bezieht, wird beim Ausleihen ein konkreter Leihvertrag geschlossen, der sich dann auch auf ein konkretes Auto bezieht.
- Für jeden Ausleihvorgang wird der Kilometerstand vor Beginn und nach Ende der Ausleihe eingetragen sowie der tatsächliche Beginn des Leihvertrags und Datum/Zeit der Rückgabe. Aus diesen Daten kann dann der Rechnungsbetrag ermittelt werden. Wurde die Rechnung bezahlt, so soll dies ebenfalls in der Datenbank gespeichert werden.

#### Abgabe ER-Diagramm

Geben Sie das Blatt mit dem Entity-Relationship-Diagramm ab. Sie erhalten ein Blatt mit der Tabellenstruktur einer relationalen Datenbank als Grundlage für die weiteren Aufgaben. Mehrdeutige Relationen sind bereits in Verbindungstabellen umgewandelt (grau unterlegt), Primärschlüssel sind unterstrichen und der Bezug durch Fremdschlüssel mit Pfeilen dargestellt. Die Spaltennamen sind selbst erklärend.

**Aufgabe 2: SQL-Abfrage**

(4)

Geben Sie eine SQL-Abfrage an, die Bezeichnungen und Typen aller Automodelle ermittelt, die weniger als 50 € pro Tag und weniger als 0,20 € pro km kosten.

```
SELECT Bezeichnung, Typ
FROM Automodell
WHERE PreisTag < 50 AND PreisKM < 0.2;
```

**Aufgabe 3: SQL-Abfrage**

(9)

Geben Sie eine SQL-Abfrage an, mit der alle Kunden mit Namen ausgegeben werden, die schon einmal ein Cabrio ausgeliehen haben, sortiert nach dem Anfangsdatum der Ausleihe.

```
SELECT Kunde.Name
FROM Kunde, Leihvertrag, Auto, Automodell
WHERE Kunde.ID = Leihvertrag.Kunde_ID AND
      Auto.Kennzeichen = Leihvertrag.Auto_Kennz AND
      Auto.Modell_ID = Automodell.ID AND
      Automodell.Typ = 'Cabrio'
ORDER BY BeginnZeit;
```

**Aufgabe 4: SQL-Abfrage**

(8)

Geben Sie eine SQL-Abfrage an, die für jedes Automodell die Anzahl der bei der Leihfirma vorhandenen Autos (Exemplare) ermittelt.

```
SELECT Bezeichnung, COUNT(Kennzeichen)
FROM Auto, Automodell
WHERE Auto.Modell_ID = Automodell.ID
GROUP BY Bezeichnung;
```

**Aufgabe 5: SQL-Anweisungen****(12)**

Die Leihfirma hat am 10. Juli 2002 einen neuen PKW angeschafft. Es handelt sich um einen VW Golf Variant (Kombi) mit dem Kennzeichen RV UT-421, 5 Sitze mit der Zusatzausstattung: Klimaanlage, Anhängerkupplung. Der Kilometerstand ist 120 km, das Auto war als Neuwagen natürlich noch nicht beim TÜV. Es ist das erste Auto dieses Modells bei der Leihfirma. Geben Sie SQL-Anweisungen an, mit denen diese Informationen in die Datenbank eingetragen werden können.

```
INSERT INTO Automodell VALUES
(100, 'Golf Variant', 'Kombi', 'VW', 5, 3, 30.00, 0.15) •;
-- Die ID 100 sowie die Preise sind frei erfunden.

INSERT INTO Auto VALUES
('RV UT-421', 120, '10.07.2002', NULL •, 100 •);

INSERT INTO Ausstattung VALUES
(100 •, 'Klimaanlage' •);

INSERT INTO, Ausstattung VALUES
(100 •, 'Anhängerkupplung' •);
```

**Aufgabe 6:****(14)**

Bei der Abfrage

```
SELECT Auto.Kennzeichen, Kunde.Name
FROM Auto, Leihvertrag, Kunde
WHERE Leihvertrag.Auto_Kennz = Auto.Kennzeichen AND
      Leihvertrag.Kunde_ID = Kunde.ID;
```

soll die optimale Reihenfolge der Abarbeitung durch den Abfrageoptimierer des Datenbankmanagementsystems ermittelt werden. Berechnen Sie dazu einen Schätzwert für die Anzahl der Datenzugriffe für die beiden unten angegebenen Reihenfolgen. Gehen Sie dazu von 100 Autos, 1000 Kunden und 10000 Leihverträgen aus. Für die Primärschlüssel Kennzeichen bei Auto und ID bei Kunde gibt es jeweils einen Index, für die Tabelle Leihvertrag gibt es keinen Index.

**Erklären Sie kurz Ihre Formel, z. B.**

$(100 + 1000) * 10000^2 = 1,1 * 10^{11}$  Für alle Autos und alle Kunden  
suche alle Verträge und für jeden davon noch mal alle Verträge

Reihenfolge 1: „Auto – Leihvertrag – Kunde“

$100 * (10000 + 100 * \lceil \lg(1000) \rceil) = 1100000$  Datensätze • Für jedes Auto •  
durchsuche alle Leihverträge •  
davon für alle passenden (ca. 100) •  
suche den zugehörigen Kunden (lg wegen Index) •

Reihenfolge 2: „Leihvertrag – Auto – Kunde“

$10000 * (\lceil \lg(100) \rceil + \lceil \lg(1000) \rceil) = 17000$  Datensätze • Für jeden Leihvertrag •  
suche das passende Auto (lg wegen Index) •  
und den zugehörigen Kunden (lg wegen Index) •

**Aufgabe 7: Trigger****(20)**

Erstellen Sie einen Datenbank-Trigger, der beim Eintrag einer neuen Reservierung überprüft, ob die benötigte Führerscheinklasse für das reservierte Automodell in den Führerscheinklassen des Kunden enthalten ist. Wenn dies nicht der Fall ist, soll eine Fehlermeldung ausgegeben werden und die Aktion abgebrochen werden.

Hinweis:

Falls Sie die genaue PL/SQL-Syntax einer Funktion oder eines Konstrukts nicht kennen, verwenden Sie C-Syntax oder beschreiben die Operation, die Sie durchführen wollen, in einem Kommentar.

```
CREATE OR REPLACE TRIGGER Führerscheinklasse_korrekt
BEFORE INSERT AFTER • //nichtzutreffendes streichen
INSERT OR DELETE • //nichtzutreffendes streichen

ON Reservierung •• //Tabellenname ergänzen

FOR EACH ROW FOR EACH STATEMENT • //nichtzutreffendes streichen

DECLARE
FKlasse_Automodell INTEGER; •
FKlasse_passt INTEGER;

BEGIN
SELECT Führerscheinklasse • INTO FKlasse_Automodell •
FROM Automodell •
WHERE Automodell.ID = :new.Modell_ID; ••
-- Hole die Führerscheinklasse des reservierten Automodells
-- und speichere sie in der Variablen FKlasse_Automodell

SELECT COUNT(Kunde_ID) • INTO FKlasse_passt •
FROM Führerscheinklassen •
WHERE Führerscheinklassen.Klasse = FKlasse_Automodell • AND
      Führerscheinklassen.Kunde_ID = :new.Kunde_ID; ••
-- Suche in der Tabelle Führerscheinklassen nach den Führerschein-
-- klassen des Kunden, der reserviert, und zähle, wie oft die
-- benötigte Führerscheinklasse darin ist;
-- mögliche Ergebnisse 0 oder 1

IF FKlasse_passt = 0 •
THEN RAISE_APPLICATION_ERROR(-20100, 'Benötigte Führerscheinklasse
                                     für Automodell fehlt'); ••

END IF;

END;
/
```

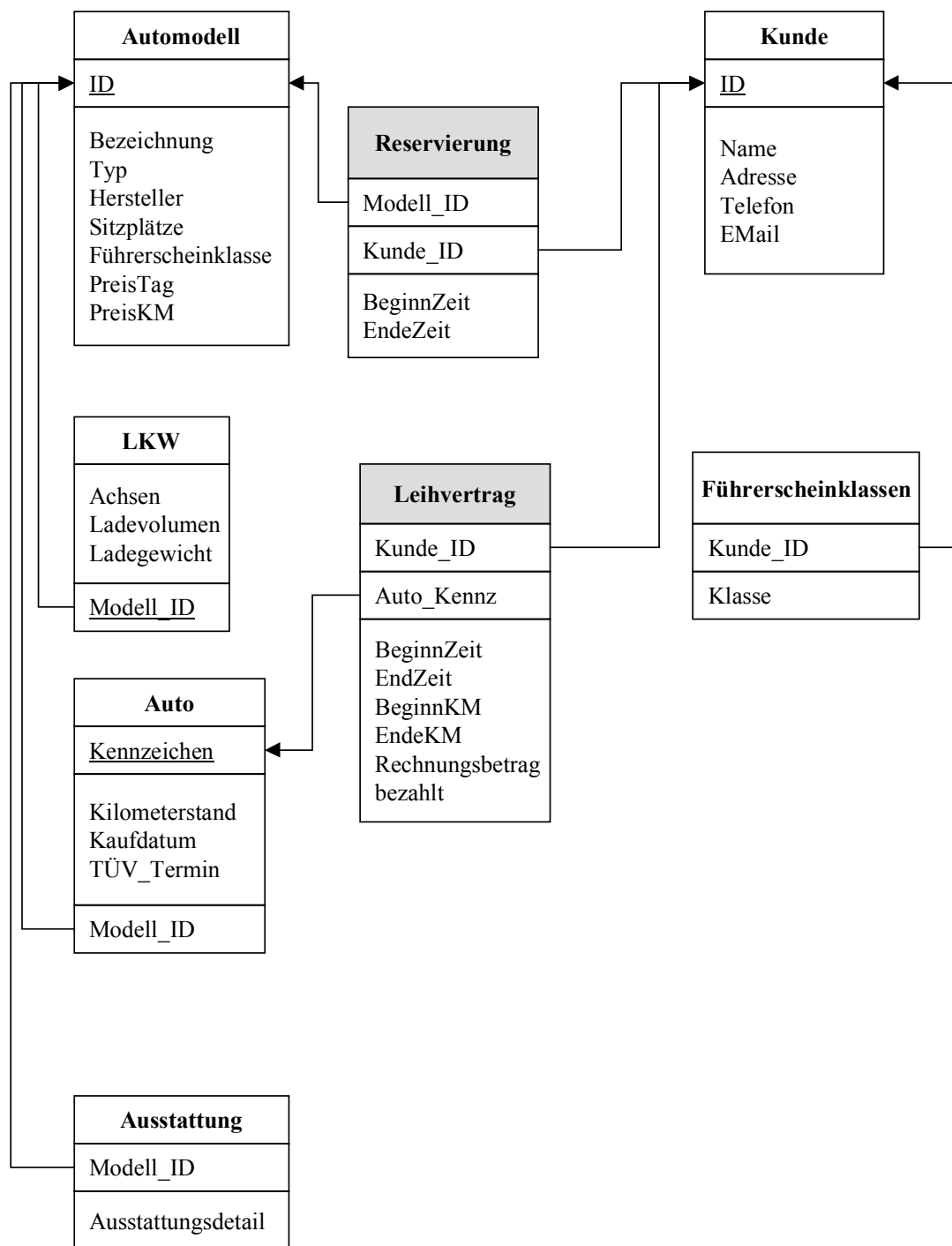


Abbildung 1: Tabellenstruktur der Datenbank für die Autoleihfirma