Technische Universität München
Fakultät für Informatik
Prof. R. Bayer, Ph.D.,
Dipl.-Inform. Michael Bauer
Dr. Gabi Höfling

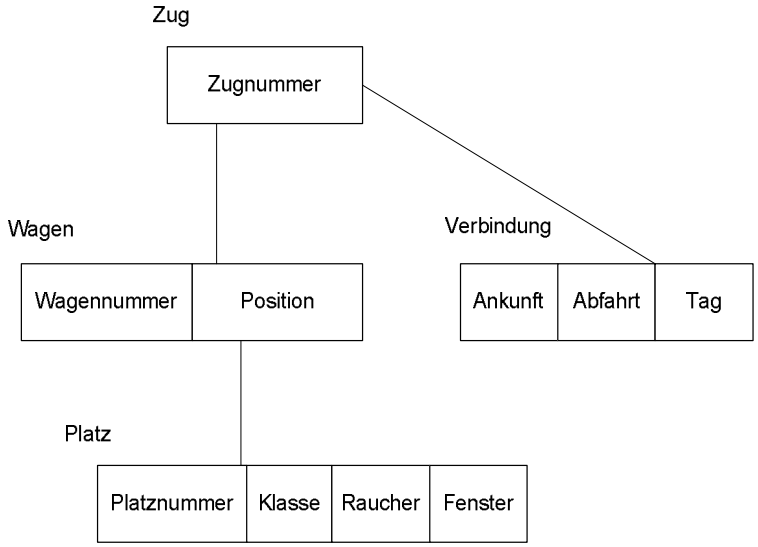
WS 2003/04,
Datenbanksysteme I,
Übungsblatt 11 –Lösungsvorschlag

2.2.2004

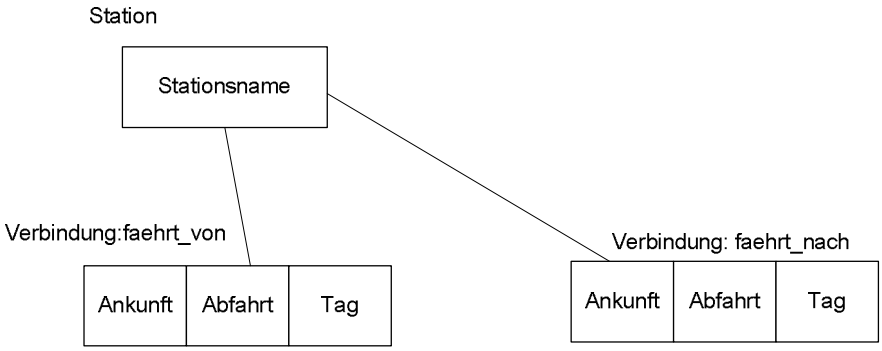
Hierarchische Datenbanken

1. Umsetzung E/R-Schema auf strikt hierarchisches Schema

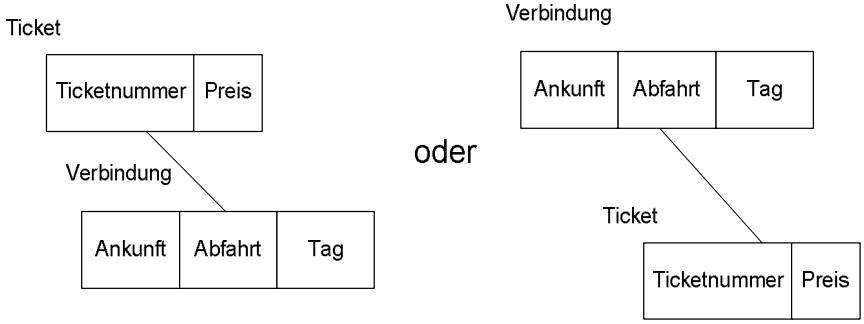
a) Beginnen Sie die Umsetzung bei der Entität Zug und setzen Sie alle Entitäten und Relationen um, die davon über 1-N-Beziehungen erreichbar sind.



b) Erstellen Sie nun ein weiteres Stück der Umsetzung ausgehend von Station

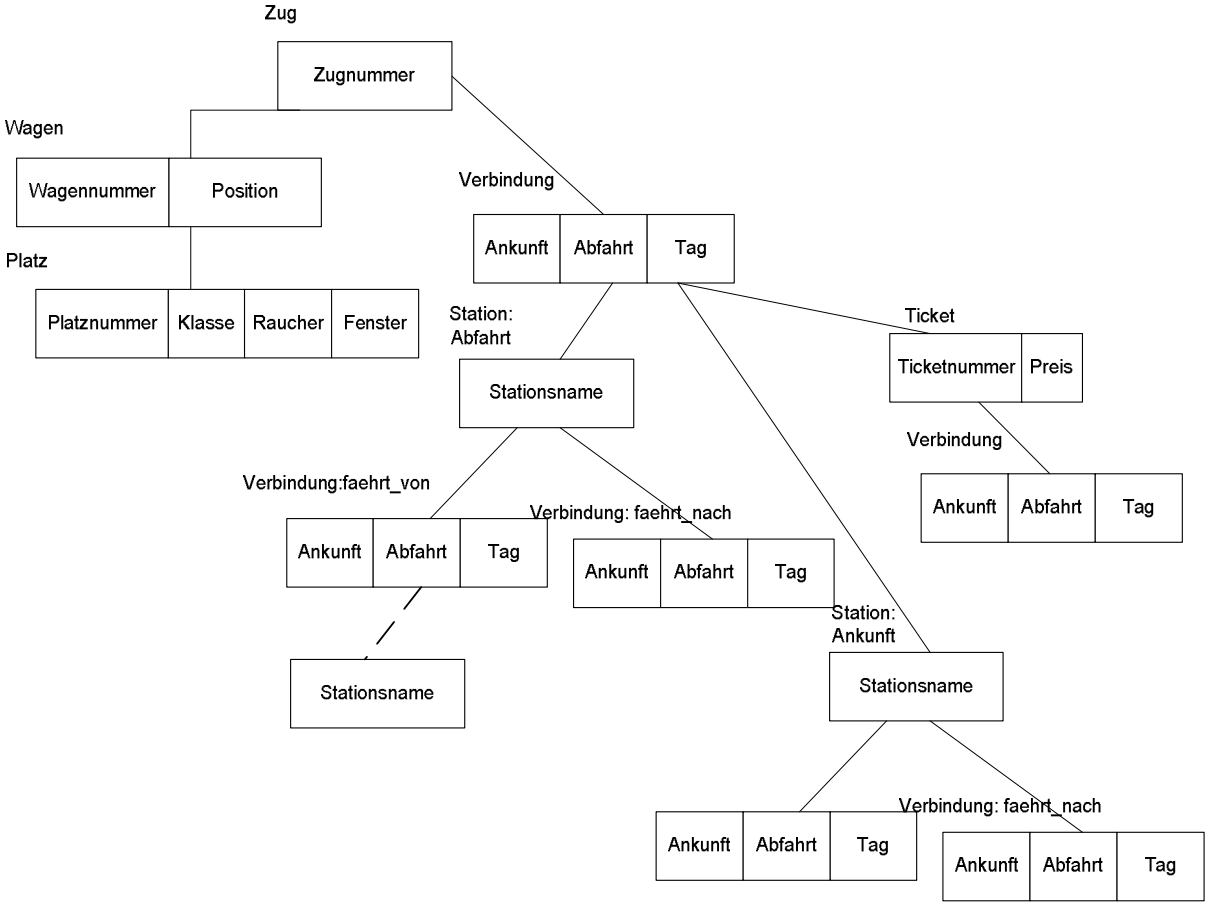


c) Setzen Sie nun die Beziehung zwischen Verbindung und Ticket um



d) Wie sieht nun das Gesamtschema aus? Wählen Sie dazu eine Gesamtwurzel aus!

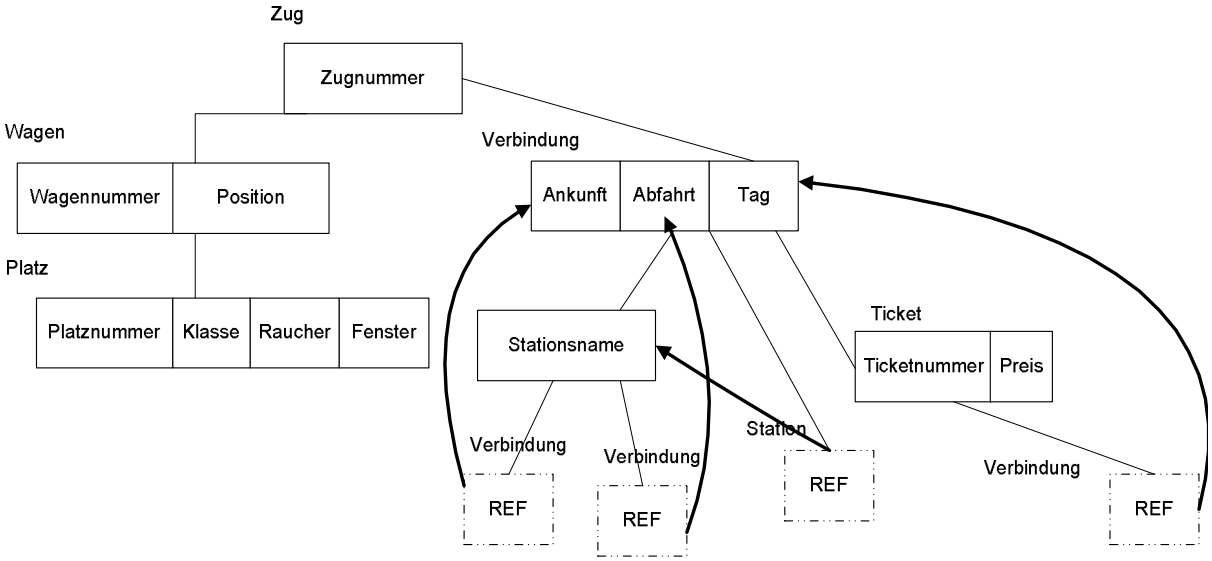
Eine mögliche Variante wählt Zug als Gesamtwurzel (Vergleiche Kursbuch der Bahn)



Die Kombination führt zu einer großen Menge an Redundanz, da viele Datensätze durch die Beziehungen mehrfach auftauchen.

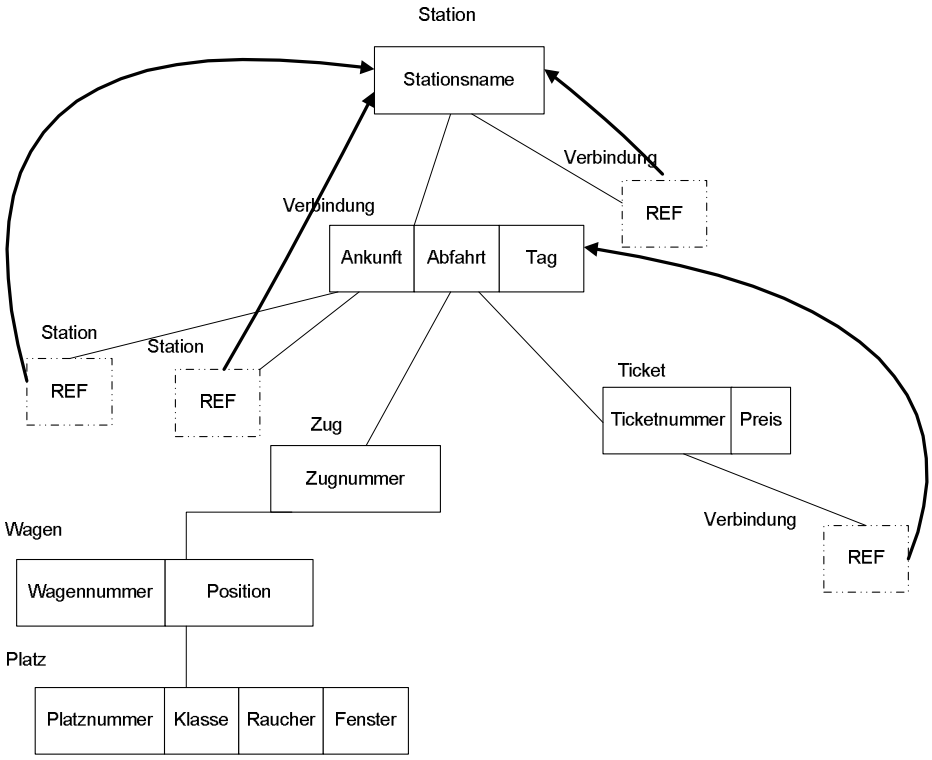
2. Umsetzung mit Referenzen

a) Identifizieren Sie redundante Einträge in 1d) und ersetzen Sie sie durch Referenzen

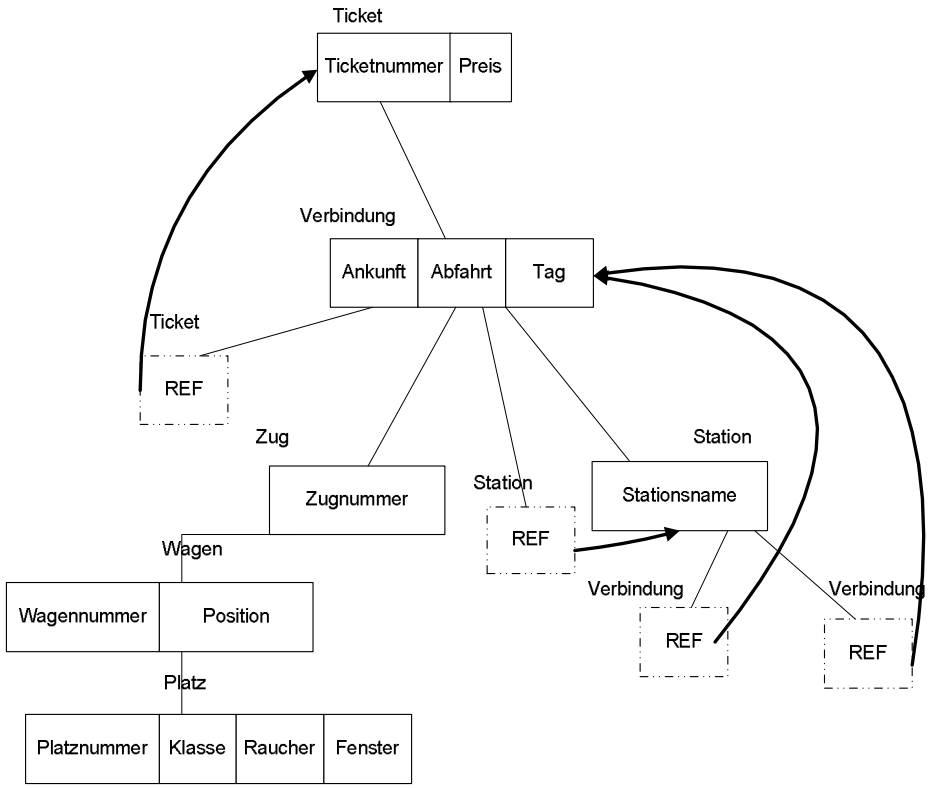


b) Wie ändert sich die Gesamtstruktur, wenn Sie eine andere Wurzel wählen?

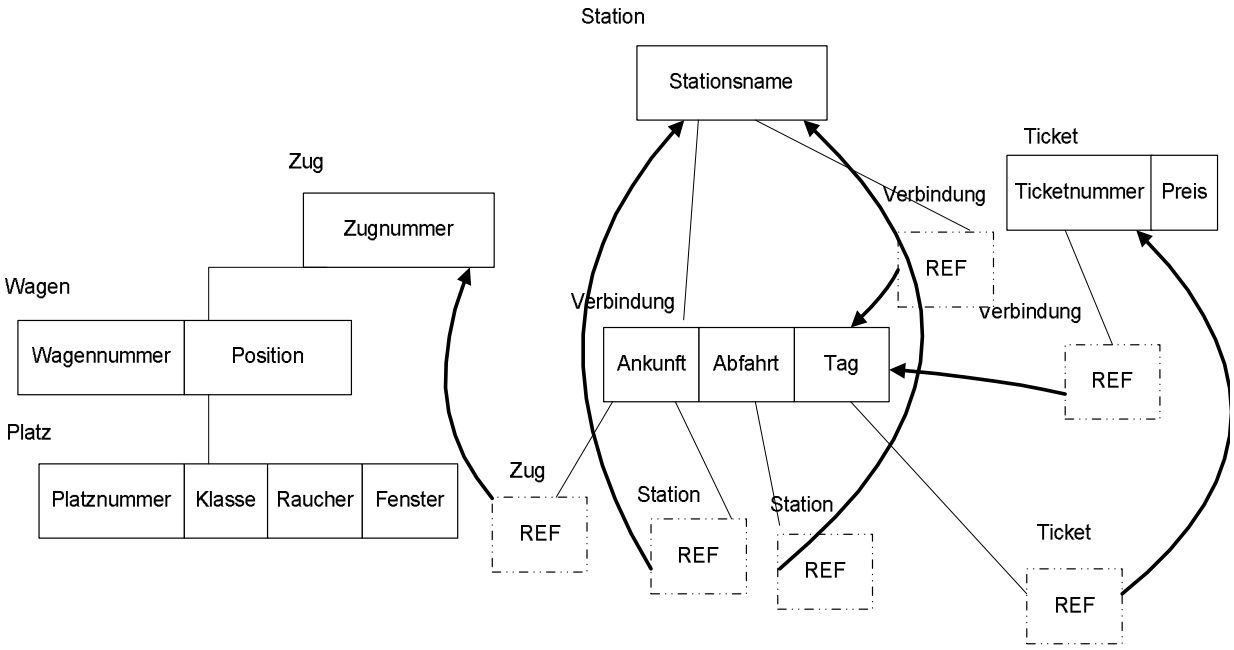
Station als Wurzel



Ticket als Wurzel



c) Wie sieht das Modell mit mehreren unabhängigen Wurzeln aus?



3) Abfragen auf dem hierarchischen Datenbanken

Nehmen sie Zug als Wurzel an.

a) Ermitteln Sie alle Wagen, die an vierter Position in einem Zug sind.

Nach Notation der Vorlesung & Java-ähnlicher Syntax:

```
Vector ergebnis;

GU ZUG
While(DBSTATUS = 0) {
        Wagen w = GNP WAGEN (Position = 4)
        If (DBSTATUS = 0) {
                Ergebnis.add(w)
        }
        GN ZUG
}
```

Idee:

- Schleife über alle Züge
- Suche einen Wagen mit Position 4 im Zug (es gibt maximal einen!)
- Wenn gefunden, ins Ergebnis
- Wenn nicht, weiter mit nächstem Zug

b) Ermitteln Sie alle Züge, die Raucherplätze am Fenster in der ersten Klasse haben

```
Vector ergebnis;
Boolean found = false;

Zug z = GU ZUG
WHILE (DBSTATUS = 0) {
        found = false;
        GNP WAGEN
        WHILE (DBSTATUS =0 && !found) {
                GNP Platz (Raucher=True, Klasse=1, Fenster=True)
                        IF (DBSTATUS = 0)  {
                                Found == true;
                        }
                GNP WAGEN
        }
        if (found) {
                ergebnis.add(z)
        }
        z = GN Zug
}
```

Idee:

- Schleife über alle Züge
- Schleife über alle Wagen – Abbruch, sobald ein Wagen Plätze nach den Kriterien enthält
- Suche nach einem Platz, der die Kriterien erfüllt
- Wenn gefunden, Ende der Wagenschleife, ins Ergebnis einfügen
- Wenn nicht, weiter mit nächstem Wagen

Zum Vergleich in SQL:

```
SELECT z.Zugnummer
FROM Zug z, Wagen w, Platz p
WHERE z.Zugnummer = w.Zugnummer
AND w.Wagennummer = p.Wagennummer
AND p.Raucher = 't'
AND p.Fenster ='t'
AND p.Klasse = 1
```