Datenbankentwurf

Stefan Maihack Dipl. Ing. (FH)
Datum: 15.03.2020

Datenbankentwurf

Entwurfsaufgabe

- Datenhaltung für **mehrere** Anwendungssysteme und **mehrere** Jahre
- Anforderungen an den Entwurf:

  → Anwendungsdaten jeder Anwendung sollen aus Daten der Datenbank ableitbar sein. „Möglichst effizient“

  → Nur „vernünftige“ (wirklich benötigte) Daten sollen gespeichert werden.

  → Nicht-redundante Speicherung

  → Beschreiben die Abläufe und Funktion, die modelliert werden soll.

Datenbankentwurf

Entwurfsprozess

- Abfolge von Entwurfsdokumenten

  → Von abstrakter Beschreibung

  → bis tatsächlichen Realisierung in einem DBMS

  → Verschiedene Beschreibungsformalismen (ER, Relationsmodell, SQL DDL usw.)

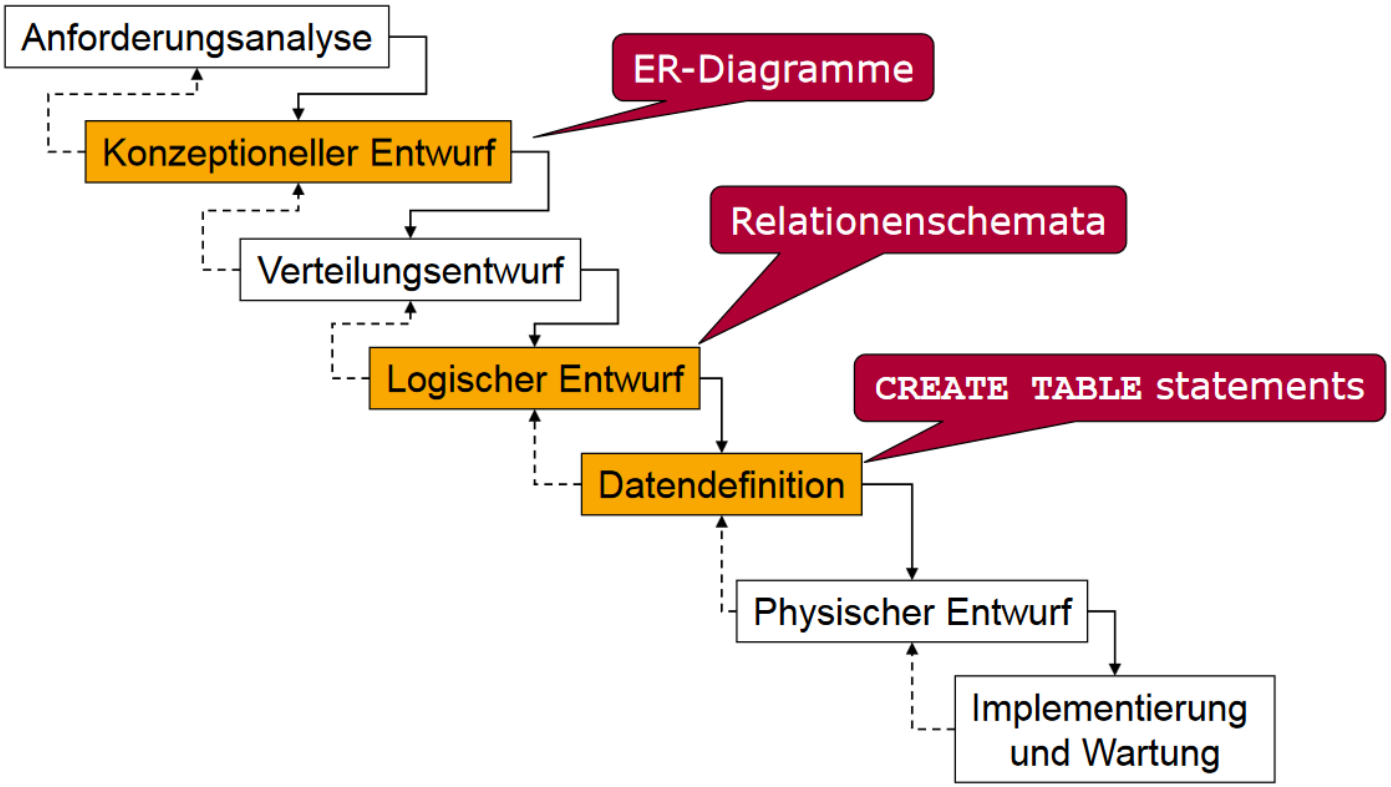
- In jedem Schritt

  → Informationserhaltung

  → Konsistenzerhaltung

Datenbankentwurf

Datenbankentwurf



Datenbankentwurf

Konzeptioneller Entwurf

- Erste formale Beschreibung des Fachproblems
  → UoD: Universe of Discourse (Diskurswelt)

- Sprachmittel: semantisches Datenmodell
  → ER

- Vorgehensweise
  → Modellierung von Sichten z.B. für verschiedene Fachabteilungen
  → Analyse der vorliegenden Sichten in Bezug auf Konflikte (Namenskonflikte, Typkonflikte, Bedingungskonflikte, Strukturkonflikte)
  → Integration der Sichten in ein Gesamtschema

- Ergebnis
  → konzeptionelles Gesamtschema, z.B. ER-Diagramm

Datenbankentwurf

Datenanalyse & Beschaffung

- Schritt 1: ER-Modell der Datenbank erstellen
- Schritt 2: Informationssichtung & Informationssammlung

  → Sammlung des Informationsbedarfs z.B. in http://www.overpass-turbo.eu.
- Ergebnis

  → OSM-Rohdaten als XML-Datei
- Schritt 3: Rohdaten in Tabellenform umwandeln

  → OSM-Rohdaten in Excel importieren und in Tabellenform umwandeln
- „Klassischer“ DB-Entwurf

  → nur Datenanalyse und Folgeschritte
- Funktionsentwurf

  → Abfragen entwerfen

Datenbankentwurf

Regeln

- Arten von Regeln

  → allgemein gebräuchliche Regeln

  → Unternehmungsregeln

  → gesetzliche Vorschriften

  → Regeln des gesunden Menschenverstandes

- Niemals vergessen:

  → Das naheliegende zu hinterfragen

  → undurchsichtige und komplizierte Sachverhalte klären

Datenbankentwurf

Begriffe: Objekte, Eigenschaften etc.

- Objekte (Tabellen) sind die „Subjekte“ oder „Dinge“ des Systems, das modelliert werden soll. Z.B. Laden mit einem Bestellsystem hat folgende Objekte:

  → Produkte
  → Produktkategorien
  → Kunden
  → Käufe
  → Rechnungen
  → Kreditkarten
  → Banken

  ➔ Darauf achten, dass es sich hierbei **nicht** nur um Greifbares handelt.

- Eigenschaften (Spalten) der Objekte untersuchen (z.B. Das Objekte „Rechnung“ kann die folgenden Eigenschaften besitzen:

  → Rechnungsnummer
  → Ausstellungsdatum
  → Kundenname und –adresse
  → etc.

Datenbankentwurf

Das Gesamtbild

- Objekte, Abläufe und Regeln bilden das Gesamtbild.
- Objekte werden in Tabellen überführt (z.B. Objekte „Rechnung“)

Rechnungs-nummer	Ausstellungs-datum	Wert	Kundenname	Etc.

- Festlegung der Datentypen (Integer, Character, Float)

Datenbankentwurf

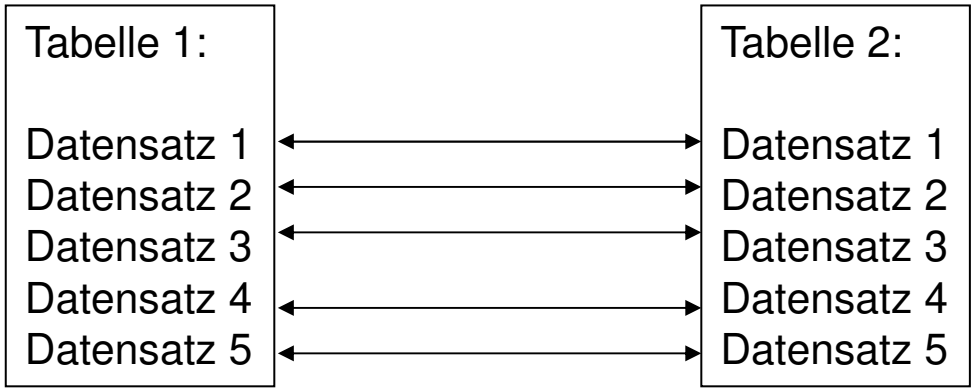
Beziehungen modellieren

- Sehr schwierig
- Falsch definierte Beziehungen erschweren später Dinge zu ordnen
- Normalisierung der Tabellen mit einbeziehen

Datenbankentwurf

Relationship-Typen (1 : 1)

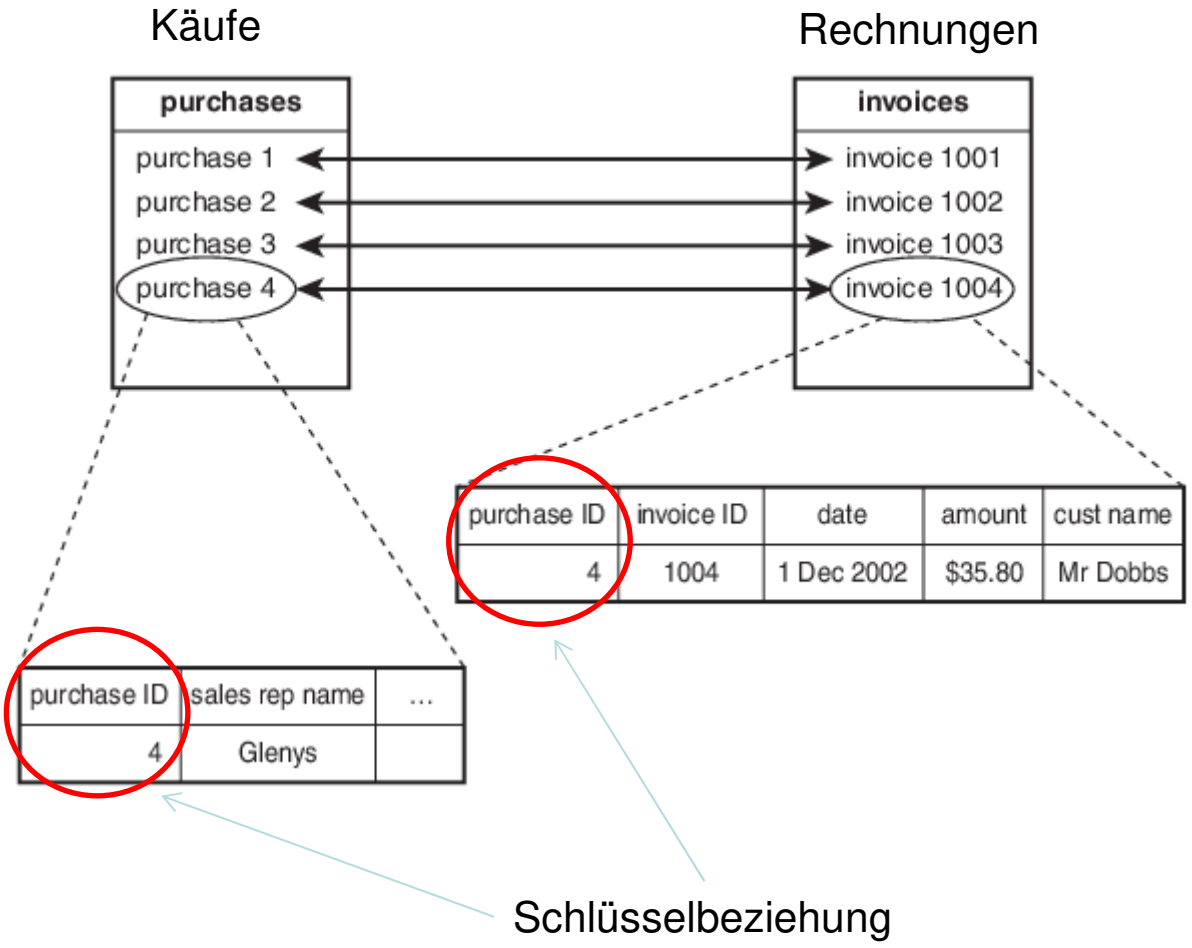
- Die 1 : 1 Beziehung (einfachste Beziehung):
  Wenn Entities zweier Entity-Typen so miteinander in Beziehung stehen, dass sie sich gegenseitig erfordern, dann spricht man von einer Eins – zu – Eins Beziehung.
- Mit anderen Worten: Jeder Datensatz aus einer Tabelle ist genau ein Datensatz aus einer anderen Tabelle zugeordnet.



- Das bedeutet auch, dass das Hinzufügen eines Datensatzes in der einen Tabelle, auch das Hinzufügen eines anderen Datensatzes in der anderen Tabelle erfordert.

Datenbankentwurf

Relationship-Typen (1 : 1)

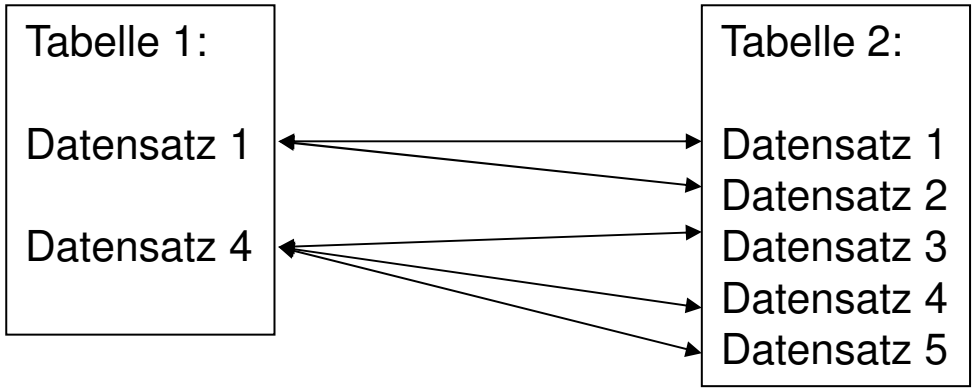


Datenbankentwurf

Relationship-Typen (1 : n)

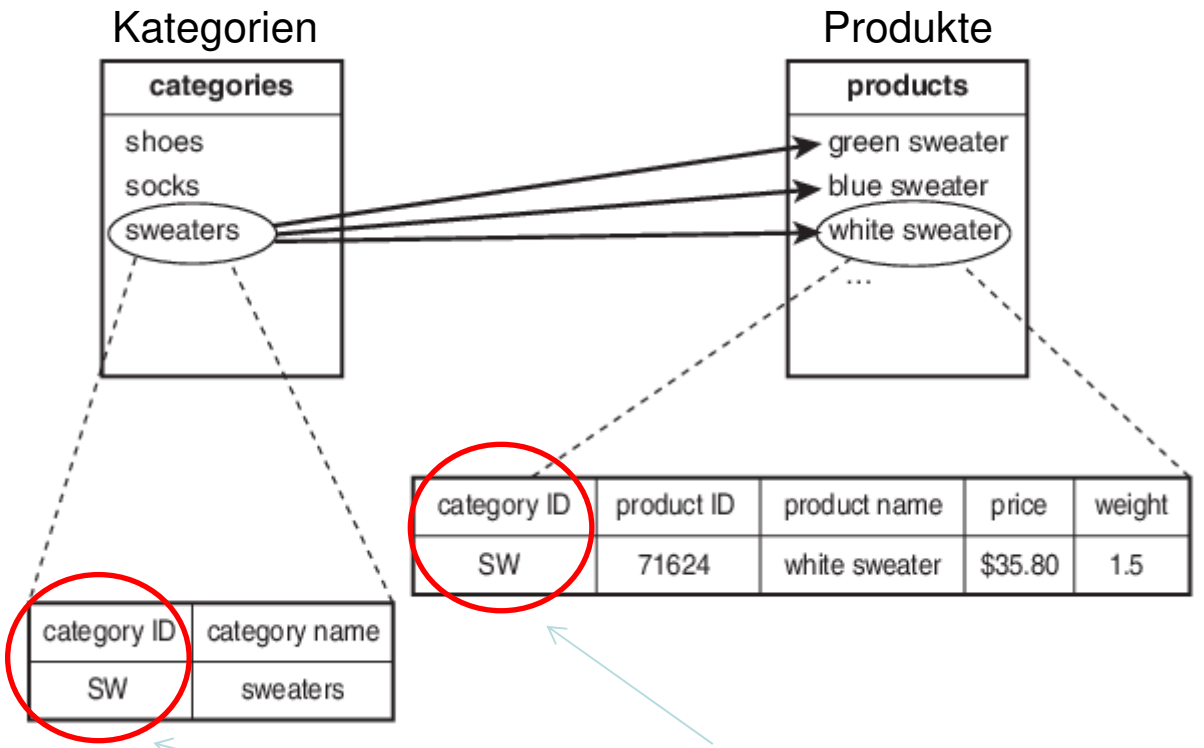
- Steht jede Entity eines Entity-Typs mit unbestimmt vielen Entities eines anderen Entity-Typs in Beziehung, so spricht man von einer Eine-zu-Vielen Beziehung.
- In der umgekehrten Richtung ist dieser Beziehungstyp immer eindeutig.
- Beispiel:
  In einer Firma hat jeder Angestellte höchstens einen Vorgesetzten, aber mehrere (oder keine) Mitarbeiter.

  ➔ Beziehungstyp: Angestellter_trägt_Personalverantwortung (Ein Beziehungstyp auf die Entity selbst = rekursive Beziehung)



Datenbankentwurf

Relationship-Typen (1 : n)



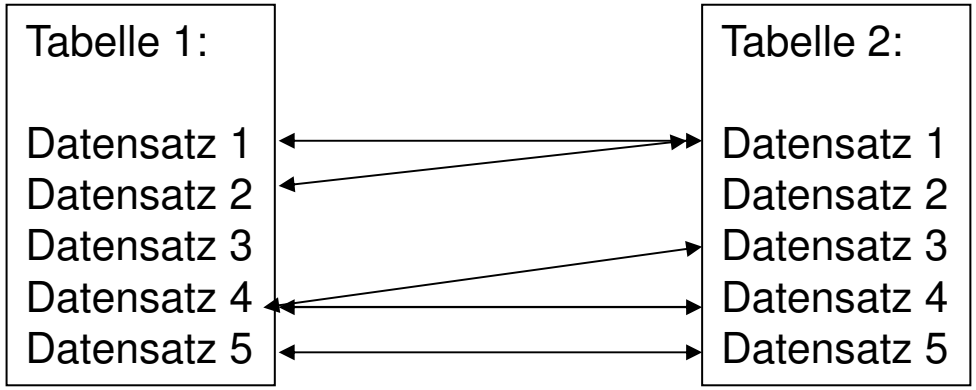
Fremdschlüssel (dient zum verknüpfen der Tabellen)

Primärschlüssel (Identifiziert jedes Produkt eindeutig)

Datenbankentwurf

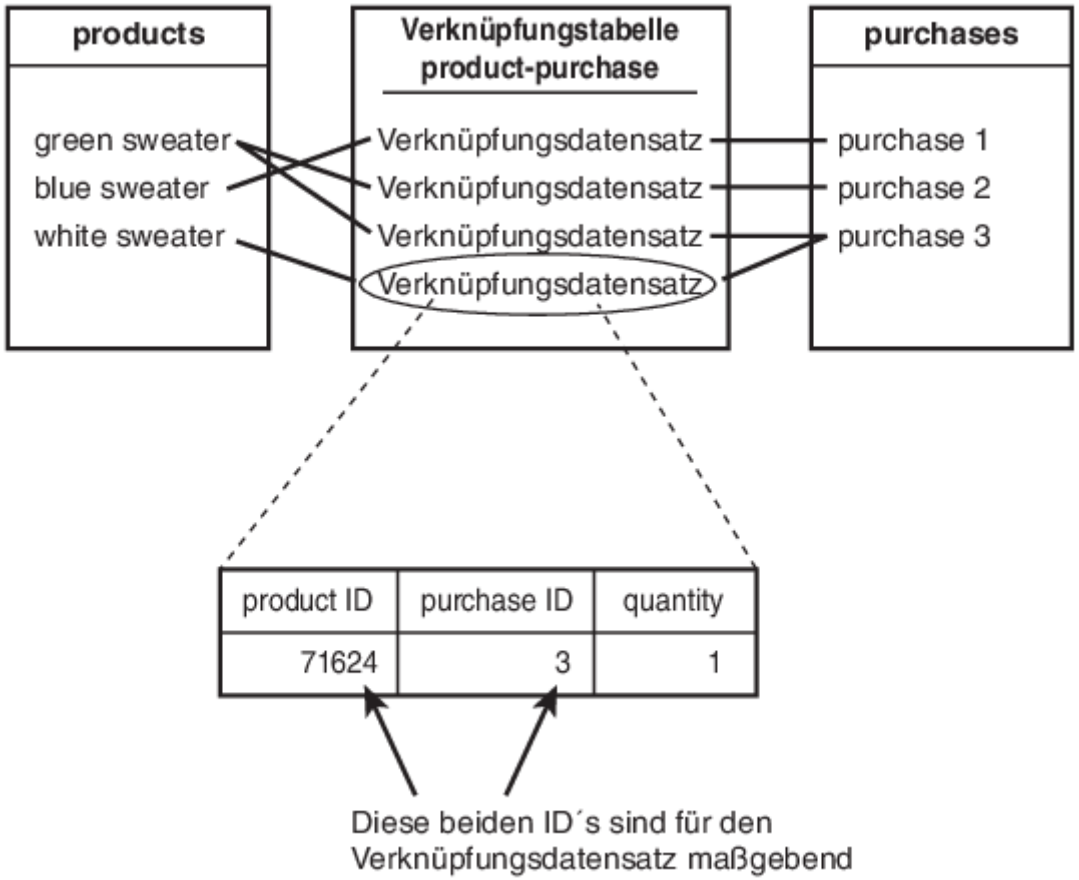
Relationship-Typen (n : m)

- Stehen die Entities eines Entity-Typs mit mehreren Entities eines Entity-Typs in Beziehung und umgekehrt, dann handelt es sich um eine Viele-zu-Vielen Beziehung.
- Häufig in der Praxis anzutreffen und erfordert wegen der Komplexität besondere Aufmerksamkeit.
- In Datenbanken lässt sich dieser Beziehungstyp nur mit einer Zwischentabelle vernünftig verwalten. Dadurch werden n : m Beziehungen in 1 : n Beziehungen aufgelöst.
- Beispiel:
  In einer Firma stehen die Entities „Angestellte“ in einer n : m Beziehung zu den Entities „Projekt“.
  Jeder Angestellte arbeitet an einem oder mehreren Projekten. Andererseits können an einem Projekt auch mehrere Angestellte arbeiten.



Datenbankentwurf

Relationship-Typen (n : m)



Das Entity – Relationship-Modell (ER-Modell)

- P. Chen spezifizierte 1976 ein Modell – das Entity-Relationship-Modell (ER-Modell). Dieses Modell wird häufig dazu genutzt, um Informationsstrukturen der realen Welt auf ein Schema abzubilden, das dann in die Datenstrukturen transformiert wird.
- Das ER-Modell ist damit ein Hilfsmittel für DB-Entwickler, die Datenstrukturen zu erkennen, sie zu gruppieren und miteinander in Beziehung zu setzten.
- Unter Verwendung dieser graphischen Hilfe werden letztlich Tabellenstrukturen gebildet.
- Selbst wenn Datenstrukturen sehr einfach aufgebaut sind, sollte auf das ER-Modell nicht verzichtet werden.

  ➔ Zeit, die einmal in das Modell investiert wurde, wird später wieder gewonnen, da logische Fehler bereits im Keim erstickt wurden.

  ➔ Außerdem ist das ER-Modell eine gute Datenbankdokumentation

Das Entity – Relationship-Modell (Begriff: Entity)

- Entity:
  Ein Entity ist ein strukturiertes Datenobjekt, das Eigenschaften besitzt, die durch Attribute beschrieben werden.

  Jedes Attribut besitzt einen definierten Wertebereich, der im Sinne des Modells nicht mehrstellig sein soll (Bedingung auch für die 1. NF) – keine Listen oder Mengen.

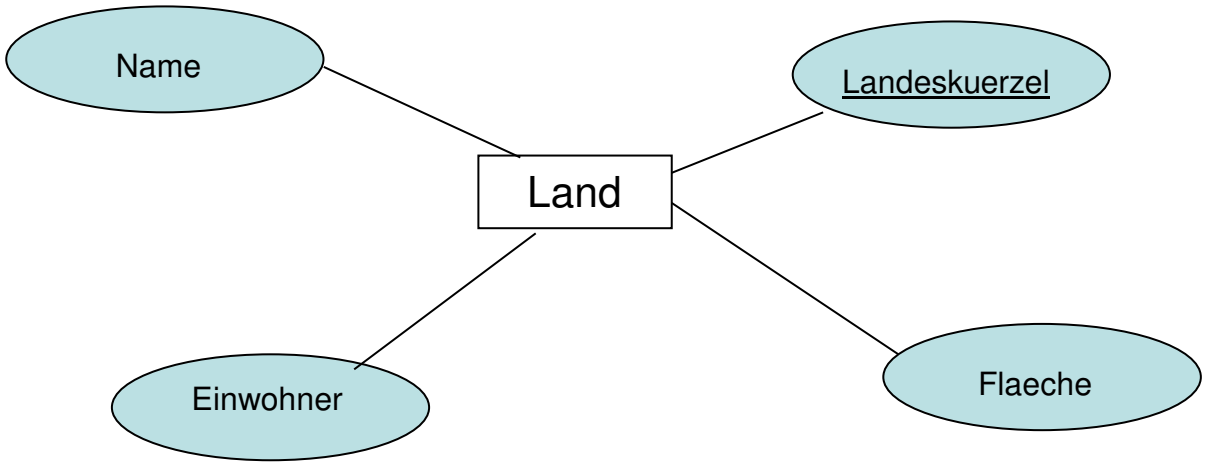
- Beispiel: In einer Firma kann ein Angestellter als Entity angesehen werden. Attribute dieser Entity sind Personalnummer, Name usw.

  Entity ➔ Tabelle

  Attribute ➔ Spalten einer Tabelle

Das Entity – Relationship-Modell (Begriff: Entity-Typen)

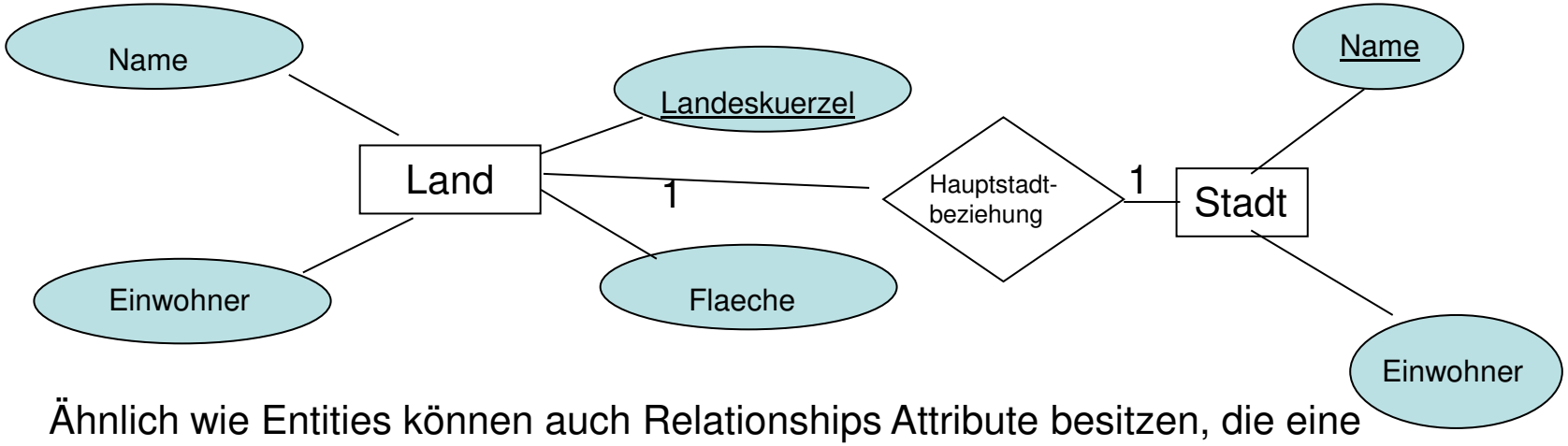
- Die Vereinigung aller Entities mit gleichen charakteristischen Eigenschaften (Attributen) wird als Entity-Type bezeichnet.
- Ein oder mehrere Attribute eines Entities müssen als Primärschlüssel definiert sein.
- Der Primärschlüssel dient dazu, alle Entities eines Entity-Typen eindeutig voneinander zu unterscheiden.



- Entitytypen werden im ER-Modell als Rechteck dargestellt.
- Attribute des Entitytyps werden als Ovale mit einem Anstrich zu dem Rechteck dargestellt.

Das Entity – Relationship-Modell (Begriff: Relationen)

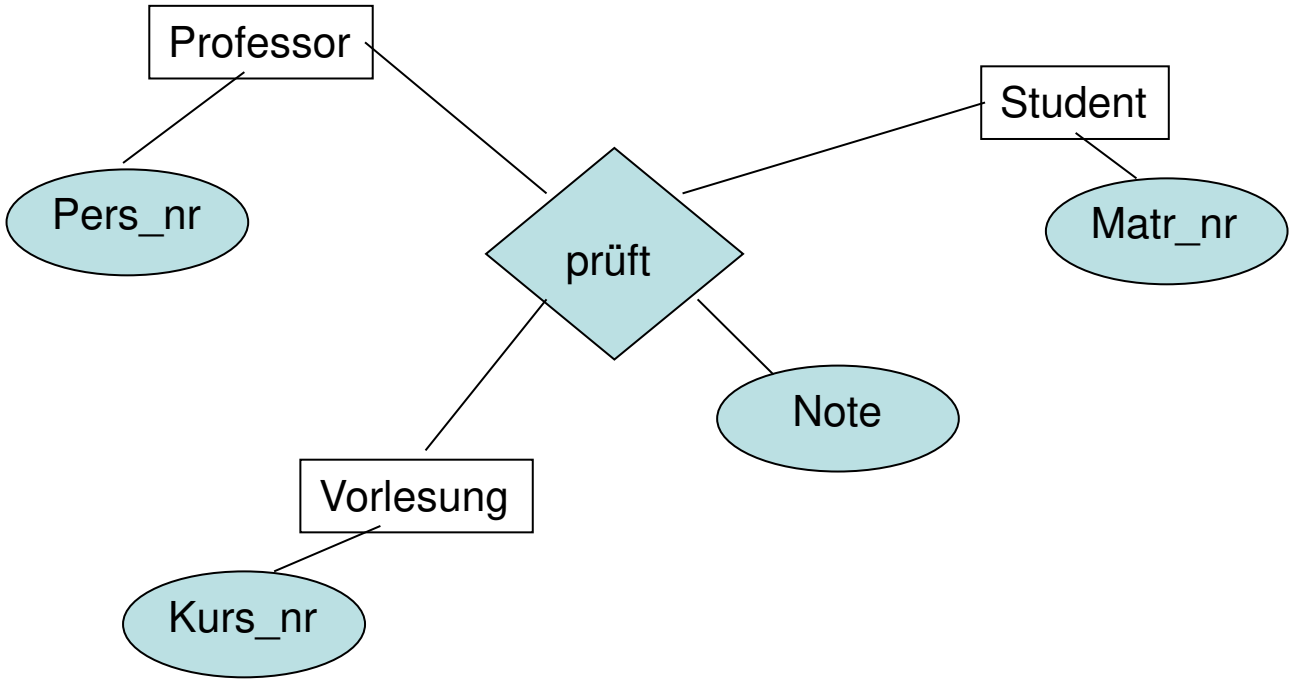
- Relationen sind Bezeichnungen zwischen Entities (Beschrieben werden diese durch sogenannte Relationships) Relationships sind keine real existierenden Objekte. Eine Relationship muss mindestens 2 Entities zu einander in Beziehung setzen.
- Z.B. Setzt man die Entities „Angestellter“ und „Projekt“ mit einander in Beziehung, erhält man die Beziehung Projekt_wird_bearbeitet von.
- Diese Beziehung ist ein Zustand, aber kein existierendes Objekt.



- Ähnlich wie Entities können auch Relationships Attribute besitzen, die eine Beziehung in spezielle Eigenschaften beschreiben. Die Attributsstrukturen und die beteiligten Entity-Typen einer Relationship werden in einem Relationship-Typ festgelegt. Relationship-Typen werden im ER-Modell als Raute dargestellt.

Das Entity – Relationship-Modell
Relationship-Typen

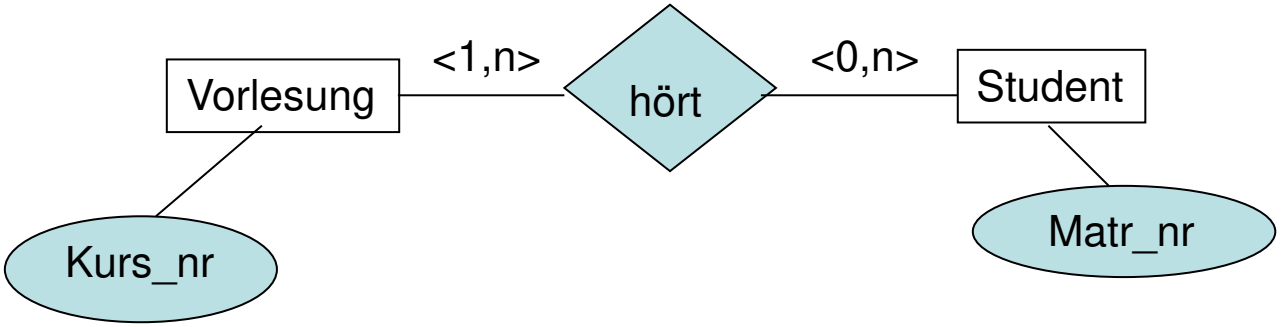
- Beziehungstypen können auch zwischen mehr als zwei Objekttypen bestehen und weiterhin ebenfalls Attribute besitzen.



- Ein Professor prüft Studenten über den Inhalt einer Vorlesung und verpasst ihm dabei eine Note

Das Entity – Relationship-Modell Kardinalitäten

- Die Art eines Beziehungstyps wird durch die Kardinalität exakter beschrieben.
- Jeder Gegenstandstyp innerhalb eines Beziehungstyps hat eine Minimale- und Maximale-Kardinalität.
- Kardinalitäten geben die minimale- sowie die maximale Anzahl von Beziehungsausprägungen an, die die Entität im Rahmen eines Beziehungstyps haben kann.
- Beispiel: Ein Student kann keine oder eine beliebige Anzahl von Vorlesungen hören (Minimal: 0; Maximal: n)

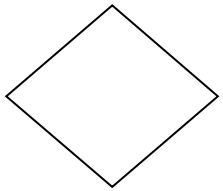


- Eine Vorlesung wird von minimal einem Studenten gehört, i. a. sind es aber n Studenten, die in einem Semester eine Vorlesung hören (Minimal: 1; Maximal:n)

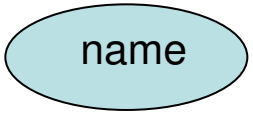
Das Entity – Relationship-Modell Notationen

- Entitytype

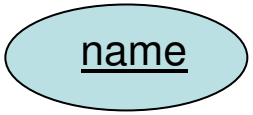
- Relationstype



- Attribut



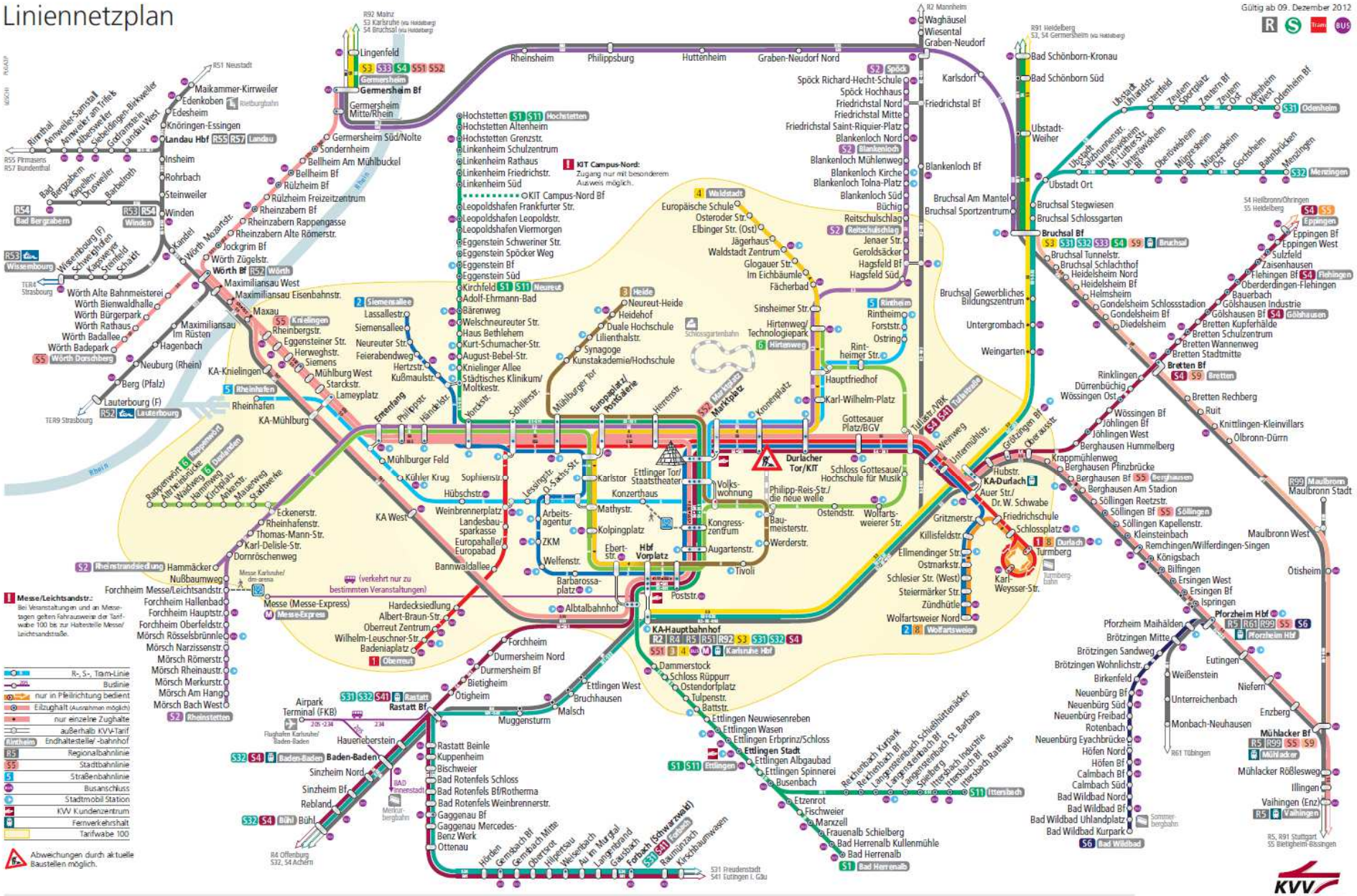
- Schlüsselattribut



- Kardinalität <1,n>; <0,n>; <3, 6> ect.

Liniennetzplan

Gültig ab 09. Dezember 2012



AVG Albtal-Verkehrs-Gesellschaft mbH
Tullastraße 71, 76131 Karlsruhe
S1 S11 S31 S32 S4 S41 S5 S51 S52 S6 S9

DB Regio AG - Regionalverkehr Württemberg
Presselstraße 17, 70191 Stuttgart
R5 R91

DB Regio AG - Regionalverkehr Südbaden
Bismarckallee 7a, 79098 Freiburg
R4

DB Regio AG - DB ZugBus Alb-Bodensee GmbH
Bahnhof 1, 72160 Horb am Neckar
R61 R9

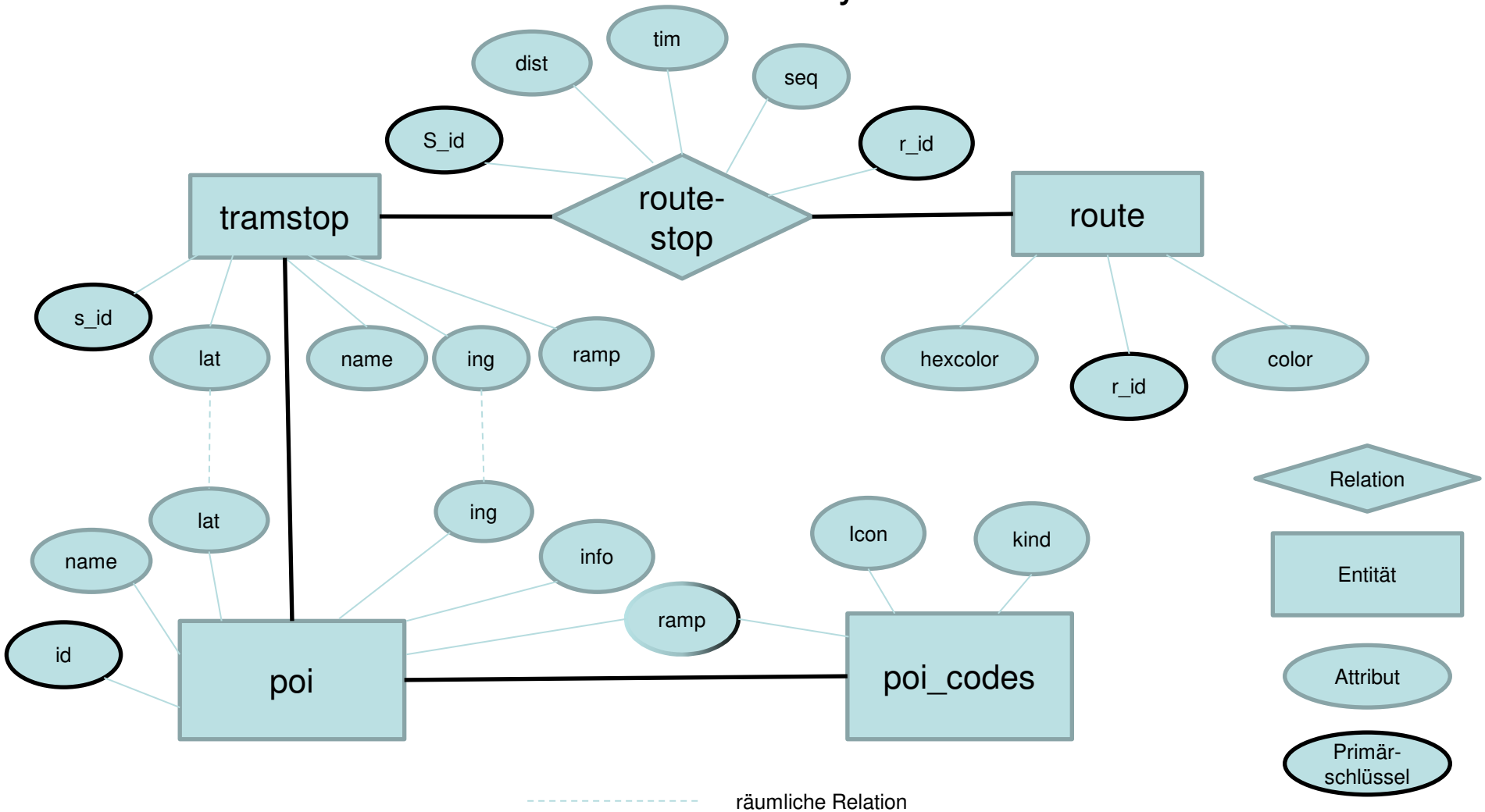
DB Regio AG - Regio RheinNeckar
Am Victoria-Turm 2, 68163 Mannheim
R2 R51 R52 R53 R54 R55 R57 R92
S3 S4 S5 S33

VBK Verkehrsbetriebe Karlsruhe GmbH
Tullastraße 71, 76131 Karlsruhe
1 2 3 4 5 6 8
S5

KVV Bewegt alle.

ER-Modell Beispiel

Das Stadtinformationssystem Brüssel



Wichtige SQL-Kommandos
Tabellen mit Daten füllen „INSERT“

- Der INSERT-Befehl eignet sich zum Einfügen neuer Datensätze in bereits bestehender Tabellen
  Für das INSERT-Kommando gibt es drei verschiedene Arten:

- 1. INSERT-Kommando (SQL-Norm)

  INSERT INTO <Tabellenname> [(Spalte1, Spalte2, …)] VALUES (Wert1, Wert2, …);

- 2. INSERT-Kommando mit SET
  Das INSERT-Kommando mit SET ist übersichtlicher, jedoch nicht in der SQL-Norm vereinbart.

  INSERT INTO <Tabellenname> SET Spalte1=Wert1 [Spalte2=Wert2, …];

- INSERT-Kommando mit SELECT-Anweisung
  Der Vorteil besteht darin, dass Sie mehrere Datensätze aus einer Tabelle auswählen, um diese in einer anderen zu speichern.

  INSERT INTO <Tabellename> [(Spalte1, …)] SELECT …

Wichtige SQL-Kommandos
Tabellen mit Daten füllen „REPLACE“

- REPLACE hat die gleiche Syntax wie INSERT (alle 3 Varianten davon). Der Unterschied besteht darin, dass bestehende Datensätze ersetzt werden können.
- Bei einer REPLACE-Anweisung überprüft MySQL einen doppelten Datensatz, indem die Schlüsselattribute verwendet werden, die als UNIQUE definiert sind. Im Normalfall der Primärschlüssel.
- Besonders gut geeignet, zur Synchronisation von 2 Tabellen. Beim INSERT müssen die Datensätze vorher gelöscht werden.

- Beispiel:

  REPLACE INTO Strassen VALUES (strassenname, anzahl_hausnummern, plz);

Wichtige SQL-Kommandos
Tabellen abfragen mit „SELECT“

- Mit der SELECT-Anweisung können Daten in Tabellen abgefragt werden.

  SELECT [<Spalte1>, <Spalte2>, …] | [*] FROM <Tabellename> WHERE <Bedingung>
- Hinter <Bedingung> verbergen sich eine oder mehrere Bedingungen, die einen Wahrheitswert liefern (TRUE oder FALSE).

  Beispiel:

  SELECT strassen, plz FROM strassen_tabelle WHERE strassen=‚Spöckerstraße‘;

- ACHTUNG: Groß- und Kleinschreibung wird unterschieden, wenn der Zeichenkettentyp den Zusatz BINARY hat oder es sich um einen BLOB handelt.

Wichtige SQL-Kommandos
Tabellen abfragen mit „SELECT“

- Mit dem Operator LIKE fragen Sie ähnliche Zeichenketten ab.
- Beim Vergleich steht das %-Zeichen für eine beliebige Anzahl passender Zeichen, das Zeichen _(Unterstrich) verwenden Sie für ein passendes Zeichen.

- Beispiel:

  SELECT * FROM strassen_tabelle WHERE strassen_name LIKE ‚%ck%‘;

- Die Umkehrung von LIKE ist NOT LIKE

Übungen - MySQL

- Aufgabe:

  1. Erstellen Sie ein ER-Modell ihrer Studienarbeit

  2. Extrahieren sie die Daten aus OpenStreetMap

  3. Erzeugen sie analog zu den MySQL-Tabellen Excel-Tabellen und befüllen Sie diese mit Daten.

  4. Erstellen Sie die Datenbank mittels SQL-Script in MySQL.

  5. Füllen Sie eine Tabelle davon mit dem SQL-Kommando INSERT.

Aufgabe 3 - MySQL

- ~~Exportieren Sie die Exceltabelle in eine CSV-Datei und importieren Sie dann die Daten mit Hilfe des SQL-Befehls LOAD DATA~~

~~LOAD DATA [LOCAL] INFILE ‚dateiname' INTO TABLE <Tabellename> [Optionen]~~

~~Für [Optionen] setzen Sie ein:~~

~~FIELDS TERMINATED BY ‚;'~~
~~[OPTIONALLY] ENCLOSED BY ‚\"~~
~~ESCAPE BY '\\'~~
~~LINES TERMINATED BY ‚\n'~~

~~Verwenden Sie LOCAL, um anzugeben, dass die Datei im Verzeichnissystem des Client gesucht wird.~~