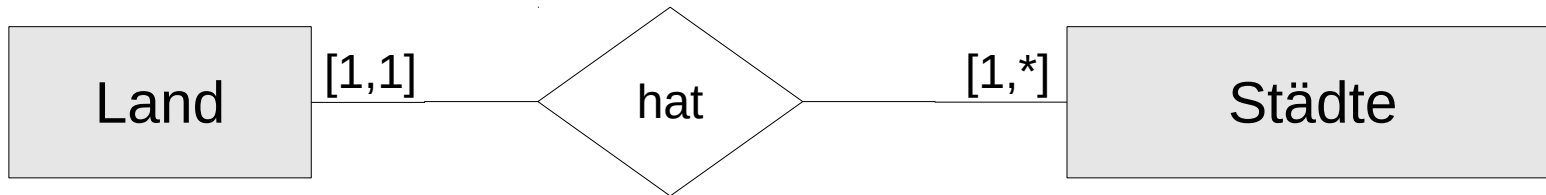


S(tructured)Q(uey)L(anguage) Beziehungen zwischen Tabellen



Beziehungen zwischen Tabellen

- Verknüpfung zwischen zwei Tabellen.
- Darstellung von Beziehungen zwischen Entitäten.
- Verweis auf ein Schlüsselfeld in einer anderen Tabelle.

Primärschlüssel (Primary key)

- Der Primärschlüssel wird bei der Neuanlage eines Datensatzes gesetzt.
- Der Primärschlüssel kommt nur einmal in einer Tabelle vor. Der Schlüsselwert identifiziert einen Datensatz, eine Zeile, in einer Tabelle.
- Der Primärschlüssel wird während der Lebensphase eines Datensatzes nicht verändert.

... setzen

```
CREATE TABLE IF NOT EXISTS tblOrt (  
  id BIGINT NOT NULL,  
  ort VARCHAR(200) NOT NULL,  
  
  PRIMARY KEY (id),  
  UNIQUE (id, ort)  
)
```

Spaltenbedingung

```
id BIGINT NOT NULL UNIQUE PRIMARY KEY
```

- Der Befehl `PRIMARY KEY` am Ende einer Zeile kennzeichnet ein Primärschlüssel-Feld in einer Tabelle.
- Das Schlüsselfeld ist einmalig (`UNIQUE`).
- Das Schlüsselfeld darf nicht leer sein (`NOT NULL`).

Tabellenbedingung

```
PRIMARY KEY (id)
```

- Der Befehl `PRIMARY KEY` kennzeichnet ein Primärschlüsselfeld in einer Tabelle.
- In den runden Klammern folgt eine Auflistung von Feldern, die als Schlüsselfelder genutzt werden.
- In diesem Beispiel besteht der Primärschlüssel nur aus einem Datenfeld.

Hinweise zu Bezeichnern

- Der Feldname oder der Tabellename beginnt mit einem Buchstaben.
- Ein Bezeichner besteht aus den lateinischen Groß- und Kleinbuchstaben und den Ziffern.
- Ein Bezeichner enthält als Sonderzeichen nur den Unterstrich.
- Ein Bezeichner sollte nicht länger als 18 Zeichen sein. Die maximale Länge ist abhängig vom verwendeten System.
- SQL-Befehle können nicht als Bezeichner genutzt werden.

Auswahl eines Bezeichners

- Der Name `ID` oder `id` wird häufig bei Feldern genutzt, deren Schlüsselwert automatisiert gesetzt wird.
- Namen eines künstlichen Attributs des Objekts in der realen Welt. Zum Beispiel Artikelnummer, Bestellnummer, KFZ-Nummer etc..
- Namen in der Form `[tabelleName]_ID` oder `[tabelleName]_id` können genutzt werden.

Weitere Hinweise

- Der Speicherplatz für einen Primärschlüssel sollte so klein wie möglich und so groß wie nötig sein.
- Um so kleiner der Speicherbedarf eines Primärschlüsselwertes ist, um so kleiner ist der Index. Mit Hilfe eines Index wird die Suche nach Werten in einer Tabelle beschleunigt.

Automatische Berechnung des Schlüssels

```
CREATE TABLE IF NOT EXISTS tblAdresstyp(  
    ID SERIAL UNIQUE PRIMARY KEY,  
    adresstyp VARCHAR(200) NOT NULL  
)
```

- Der „Datentyp“, SERIAL ist ein Alias für folgende Arbeitsschritte: Erzeuge eine Sequenz. Berechne einen neuen Wert für die Sequenz.
- Ein Feld SERIAL darf nicht leer sein. Die Bedingung NOT NULL ist in dem Datentyp SERIAL enthalten.

Erzeugung und Nutzung einer Sequenz

```
CREATE SEQUENCE adresstyp_id_seq
    INCREMENT BY 1 START WITH 1;

CREATE TABLE IF NOT EXISTS tblAdresstyp(
    ID INTEGER UNIQUE
        DEFAULT NEXTVAL('adresstyp_id_seq')
    PRIMARY KEY,
    adresstyp VARCHAR(200) NOT NULL
)
```

Erläuterung

- `CREATE SEQUENCE` erzeugt eine neue Sequenz. Die Sequenz hat einen eindeutigen Namen. Die neue Sequenz startet mit dem Wert eins und wird bei jedem Aufruf um eins inkrementiert.
- Mit Hilfe der Funktion `nextval()` wird der aktuelle Wert des Zählers geholt und entsprechend der Angaben in der Sequenz verändert. Der neue Zählerstand wird in das Datenfeld geschrieben.
- Hinweis: Tabellen, die auf `_seq` enden, enthalten Informationen zu einer Sequenz.

Zusammengesetzter Primärschlüssel

```
CREATE TABLE IF NOT EXISTS tblPerson(  
  
  anrede varchar(10) NOT NULL DEFAULT 'UNBEKANNT',  
  vorname varchar(100),  
  nachname varchar(200) NOT NULL,  
  geburtsdatum DATE NOT NULL  
  aufgenommenAm Date NOT NULL DEFAULT current_Date,  
  
  CONSTRAINT KEY_PERSON  
    PRIMARY KEY (geburtsdatum, aufgenommenAm, nachname)  
);
```

Erläuterung

```
PRIMARY KEY (geburtsdatum, aufgenommenAm, nachname)
```

- Der zusammengesetzte Schlüssel wird als Tabellenbedingung definiert.
- Dem Befehl `PRIMARY KEY` folgen runde Klammern. Die runden Klammern begrenzen eine Liste von Feldern. Die Felder werden in der Liste durch ein Komma getrennt.
- Der zusammengesetzte Schlüssel ist einmalig.

Fremdschlüssel (Foreign key)

- Platzhalter für Datensätze, die in einer anderen Tabelle abgelegt sind.
- Verweis auf einen Primärschlüssel in einer anderen Tabelle.
- Ein Fremdschlüssel kann beliebig oft in einer Tabelle vorkommen.
- Primär- und Fremdschlüssel müssen vom gleichen Feldtyp sein.

... setzen

```
CREATE TABLE IF NOT EXISTS personWohnt (  
  id SERIAL,  
  personnr BIGINT,  
  strasse VARCHAR(200),  
  hausnummer VARCHAR(10),  
  postleitzahl VARCHAR(5) NOT NULL,  
  wohnort BIGINT NOT NULL,  
  adresstyp INT DEFAULT 0,  
  
  CONSTRAINT KEY_PERSONWOHNT PRIMARY KEY(id),  
  CONSTRAINT FKEY_PERSON FOREIGN KEY(personnr)  
    REFERENCES tblPerson(personnr),  
  CONSTRAINT FKEY_ORT FOREIGN KEY(wohnort)  
    REFERENCES tblOrt(id),  
  CONSTRAINT FKEY_ADRESSTYP FOREIGN KEY(adresstyp)  
    REFERENCES tblAdresstyp(id)  
  
);
```


Spaltenbedingung

```
personnr BIGINT REFERENCES tblPerson(personnr),
```

- Der Befehl `REFERENCES` definiert am Ende einer Felddefinition den Verweis auf ein Primärschlüssel-feld.
- Ein Verweis wird folgendermaßen definiert: `tabelle(feld)`.

Hinweise zum Namen

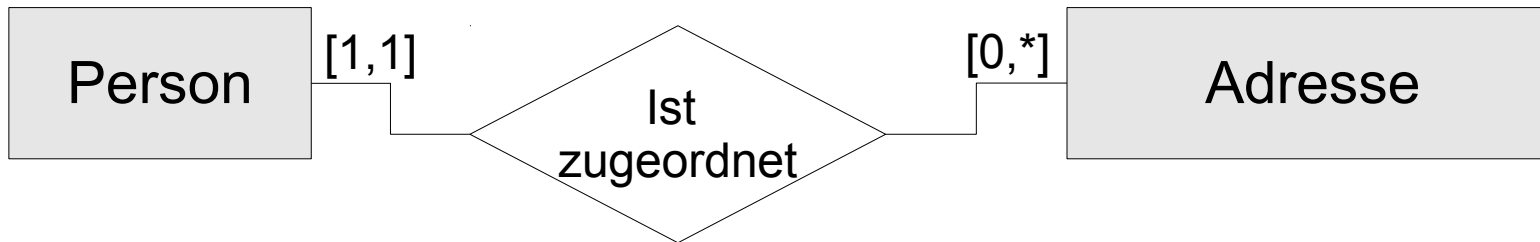
- Häufig wird der Name aus dem Feld und der Tabelle, auf die verwiesen wird, genutzt.
- Das Muster `[verweisTabelle]_ID` oder `[verweisTabelle]_id` kann genutzt werden.
- Namen eines künstlichen Attributs des Objekts in der realen Welt. Zum Beispiel Kundennummer, Bestellnummer, KFZ-Nummer etc.

Tabellenbedingung

```
CONSTRAINT FKEY_PERSON  
FOREIGN KEY (personnr)  
REFERENCES tblPerson (personnr)
```

- Der Befehl `FOREIGN KEY` kennzeichnet ein Fremdschlüssel-Feld in einer Tabelle. In den runden Klammern folgt eine Auflistung von Feldern, die als Fremdschlüssel genutzt werden.
- Der Befehl `REFERENCES` definiert den Verweis zu einem Feld in einer anderen Tabelle.
- Mit Hilfe des Befehls `CONSTRAINT` kann der Bedingung ein Namen gegeben werden.

1 : n - Beziehung



- Zu einem Datensatz in einer Tabelle existieren beliebig viele Datensätze in einer anderen Tabelle.
- Der Primärschlüssel in der Tabelle der „1-Seite“ wird als Fremdschlüssel in der Tabelle der „n-Seite“ übernommen.
- Häufigste Beziehung.

... definieren

Master-Tabelle

```
CREATE TABLE IF NOT EXISTS tblPerson(  
personnr BIGSERIAL UNIQUE,
```

1

```
CONSTRAINT KEY_PERSON PRIMARY KEY(personnr)  
);
```

```
CREATE TABLE IF NOT EXISTS personWohnt(  
id SERIAL,  
personnr BIGINT,
```

[0,*]

```
CONSTRAINT KEY_PERSONWOHNT PRIMARY KEY(id),  
CONSTRAINT FKEY_PERSON FOREIGN KEY(personnr)  
REFERENCES tblPerson(personnr)
```

Detail-Tabelle

```
);
```

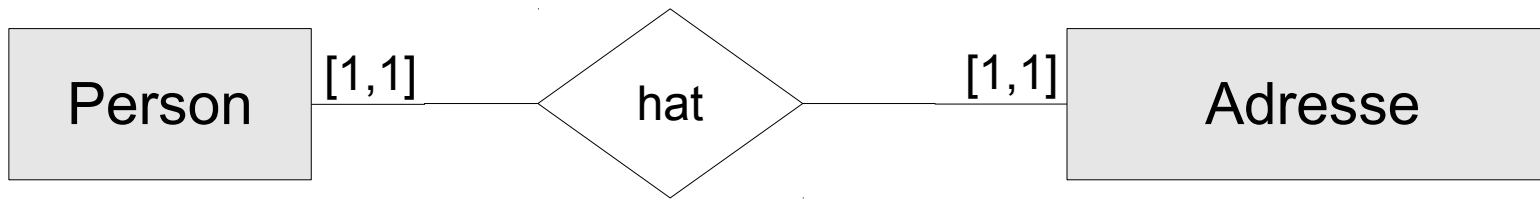
Master-Tabelle

- Darstellung der Kardinalität „1“ in einer 1 : n – Relation.
- Jeder Datensatz in der Master-Tabelle wird durch einen Primärschlüssel eindeutig identifiziert.
- „Wurzel“ einer Ordnerstruktur. Der Ordner kann weitere Unterordner enthalten.

Detail-Tabelle

- Darstellung der Kardinalität n in einer $1 : n$ – Relation.
- Details zu einem Datensatz in einer anderen Tabelle.
- Jeder Datensatz wird wie in der Master-Tabelle durch ein Primärschlüssel eindeutig identifiziert.
- Mindestens ein Datenfeld in der Detail-Tabelle enthält einen Verweis auf einen Datensatz in der dazugehörigen Master-Tabelle.
- Unterordner, die aber auch selber wieder andere Ordner enthalten können.

1 : 1 - Beziehung



- Beziehungen mit dem Verb „ist“ oder „hat“ bilden eine 1 : 1 – Beziehung ab.
- Diese Beziehungsart kann in einer Tabelle dargestellt werden.
- Es existiert höchstens ein Datensatz in der Detail-Tabelle.

... definieren

Master-Tabelle

```
CREATE TABLE IF NOT EXISTS tblPerson(  
personnr BIGSERIAL UNIQUE,
```

1

```
CONSTRAINT KEY_PERSON PRIMARY KEY(personnr)  
);
```

```
CREATE TABLE IF NOT EXISTS personWohnt(  
id SERIAL,  
personnr BIGINT,
```

[1,1]

```
CONSTRAINT KEY_PERSONWOHNT PRIMARY KEY(id),  
CONSTRAINT FKEY_PERSON FOREIGN KEY(personnr)  
REFERENCES tblPerson(personnr)  
UNIQUE (id, personnr)
```

```
);
```

Detail-Tabelle

Master-Tabelle

- Darstellung der Kardinalität „1“ in einer 1 : n – Relation.
- Jeder Datensatz in der Master-Tabelle wird durch einen Primärschlüssel eindeutig identifiziert.
- „Wurzel“ einer Ordnerstruktur. Der Ordner kann weitere Unterordner enthalten.

Detail-Tabelle

- Darstellung der Kardinalität 1 in einer 1 : 1 – Relation.
- Details zu einem Datensatz in einer anderen Tabelle.
- Jeder Datensatz wird wie in der Master-Tabelle durch ein Primärschlüssel eindeutig identifiziert.
- Mindestens ein Datenfeld in der Detail-Tabelle enthält einen Verweis in einer Master-Tabelle.
- Auf jeden Datensatz in der Master-Tabelle kann exakt einmal verwiesen werden. Der Fremdschlüssel ist einmalig (`UNIQUE`).
- Unterordner, die aber auch selber wieder andere Ordner enthalten können.

Referentielle Integrität

- Regeln beim Löschen und Aktualisieren von Datensätzen, denen Details zugeordnet sind.
- Sind die Beziehungen zwischen den Tabellen korrekt?
- Ist das Gegenstück zu einem Fremdschlüssel in der angegebenen Master-Tabelle vorhanden?
- Die Einstellungsmöglichkeiten zur referentiellen Integrität werden nicht von allen Datenbank-Managementsystemen unterstützt.

Definition

- Es gibt zu jedem Datensatz in der Detailtabelle einen passenden Datensatz in der Master-Tabelle.
- Datensätze in einer Master-Tabelle, die mit mindestens einem Datensatz in der Detailtabelle verbunden sind, können nicht gelöscht werden.

Aktualisierungsoption „Keine Aktion“

```
FOREIGN KEY (personnr) REFERENCES tblPerson (personnr)
```

- Standardeinstellungen bei der Definition eines Fremdschlüssels.
- Falls kein passender Primärschlüssel in der Master-Tabelle existiert, wird ein Fehler gemeldet.
- Solange der Primärschlüssel existiert, darf dieser nicht verändert werden. Jede Änderung des Primärschlüssel wird als Fehler gemacht und abgewiesen.

Weitere Möglichkeiten

```
FOREIGN KEY (personnr) REFERENCES tblPerson (personnr)  
ON UPDATE CASCADE,
```

```
FOREIGN KEY (personnr) REFERENCES tblPerson (personnr)  
ON UPDATE SET NULL,
```

```
FOREIGN KEY (personnr) REFERENCES tblPerson (personnr)  
ON UPDATE SET DEFAULT,
```

```
FOREIGN KEY (personnr) REFERENCES tblPerson (personnr)  
ON UPDATE RESTRICT
```

Erläuterung

- `ON UPDATE CASCADE`. Alle, zu dem geänderten Primärschlüssel, gehörenden Fremdschlüssel werden automatisiert angepasst.
- `ON UPDATE SET NULL`. Alle, zu dem geänderten Primärschlüssel, gehörenden Fremdschlüssel werden automatisch auf Null gesetzt. Der Fremdschlüssel in der Detail-Tabelle ist leer.
- `ON UPDATE SET DEFAULT`. Alle, zu dem geänderten Primärschlüssel, gehörenden Fremdschlüssel werden automatisch auf den, in der Felddefinition angegebenen Standardwert gesetzt.
- `ON UPDATE SET RESTRICT`. Entspricht der normalen Integritätsregel, wird aber sofort bei der Änderung des Datenfeldes ausgeführt.

Löschoption „Keine Aktion“

```
FOREIGN KEY (personnr) REFERENCES tblPerson (personnr)
```

- Standardeinstellung bei der Definition eines Fremdschlüssels.
- Ein Primärschlüssel, zu dem mindestens ein Fremdschlüssel existiert, kann nicht gelöscht werden.
- Falls kein passender Primärschlüssel in der Master-Tabelle existiert, wird ein Fehler gemeldet.

Weitere Möglichkeiten

```
FOREIGN KEY (personnr) REFERENCES tblPerson (personnr)  
ON DELETE CASCADE,
```

```
FOREIGN KEY (personnr) REFERENCES tblPerson (personnr)  
ON DELETE SET NULL,
```

```
FOREIGN KEY (personnr) REFERENCES tblPerson (personnr)  
ON DELETE SET DEFAULT,
```

```
FOREIGN KEY (personnr) REFERENCES tblPerson (personnr)  
ON DELETE RESTRICT
```

Erläuterung

- `ON DELETE CASCADE`. Alle, zu dem gelöschten Primärschlüssel, gehörenden Fremdschlüssel werden automatisiert gelöscht.
- `ON DELETE SET NULL`. Alle, zu dem gelöschten Primärschlüssel, gehörenden Fremdschlüssel werden automatisch auf Null gesetzt. Der Fremdschlüssel in der Detail-Tabelle ist leer.
- `ON DELETE SET DEFAULT`. Alle, zu dem gelöschten Primärschlüssel, gehörenden Fremdschlüssel werden automatisch auf den angegebenen Standardwert gesetzt.
- `ON DELETE SET RESTRICT`. Entspricht der normalen Integritätsregel, wird aber sofort bei der Löschung des Inhaltes eines Datenfeldes ausgeführt.