**Primärschlüssel:**

Bei einem Primärschlüssel handelt es sich um die Spalte (oder Spalten) einer Tabelle, mit der oder mit denen sich die Datensätze (Zeilen) dieser Tabelle **eindeutig identifizieren** lassen. Ein Primärschlüsselkann aus mehreren Elementen bestehen.

Jeder Primärschlüssel darf in einer Tabelle nur einmal vorkommen. Zudem sollte ein Primärschlüssel minimal sein.

Ein Bild, das Text, Screenshot, Zahl, Schrift enthält.

Automatisch generierte Beschreibung

**Fremdschlüssel:**

Ein Fremdschlüssel ist **ein Feld in einer Tabelle, das auf den Primärschlüssel einer anderen Tabelle verweist**. Über die Schlüssel werden Tabellen miteinander verknüpft. Ein Fremdschlüssel kann natürlich mehrfach in einer Tabelle vorkommen, genauso kann eine Tabelle über mehrere Fremdschlüssel verfügen.

**Referentielle Integrität**

Mithilfe der referentiellen Integrität kann die Konsistenz einer Datenbank gewährleistet werden. Bei der Eingabe wird überprüft, ob der Fremdschlüssel auf einen existierenden Primärschlüssel verweist. Auch wird das Löschen oder Verändern eines Primärschlüssels verhindert. Die referenzielle Integrität ist die logische Abhängigkeit eines Fremdschlüssels von einem Primärschlüssel

Wenn Sie z.B. eine Zeile löschen, die einen Primärschlüssel enthält, oder sie mit einem anderen Primärschlüssel aktualisieren, zerstören Sie die Bedeutung aller Zeilen, die diesen Wert als Fremdschlüssel enthalten. Die Integrität einer Zeile, die einen Fremdschlüssel enthält, hängt von der Integrität der Zeile ab, auf die sie verweist, d. h. von der Zeile, die den entsprechenden Primärschlüssel enthält.

Standardmäßig lässt der Datenbankserver nicht zu, dass Sie die referenzielle Integrität verletzten, und gibt eine Fehlernachricht aus, wenn Sie versuchen, Zeilen aus der übergeordneten Tabelle zu löschen, bevor Sie Zeilen aus der untergeordneten Tabelle löschen.

(Sie können jedoch die Option ON DELETE CASCADE verwenden, um durch das Löschen einer übergeordneten Tabelle das Löschen untergeordneter Tabellen auszulösen.)

Bezogen auf das obige Beispiel: Bei der Eingabe einer Kunden-ID in der Tabelle „Bestellungen“ wird sofort geprüft, ob es einen korrespondierenden Primärschlüssel dazu in der Tabelle „Kunden“ gibt.

**Redundanz**

Mit Redundanz wird das mehrfache Vorkommen derselben Information in einer Datenbank bezeichnet. Allerdings ist ein mehrfaches Vorkommen derselben Information erst als redundant zu bezeichnen, wenn man es weglassen kann, ohne dass es zu einem Informationsverlust kommt.

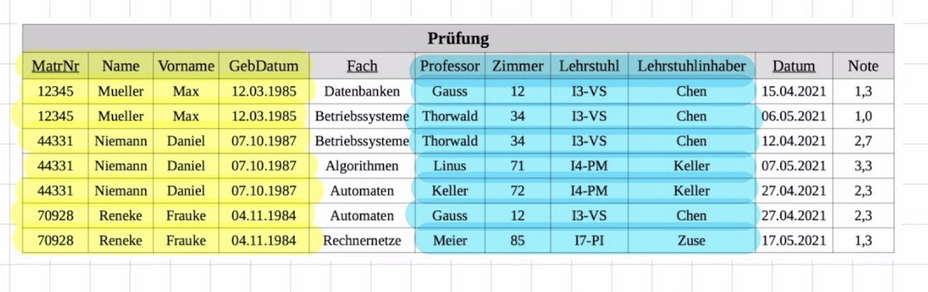
Beispiel: In einer Klasse heißen mehrere Jungen Marc (Marc Müller, Marc Schmitz und Marc Wagner). Die erste Voraussetzung für Redundanz ist erfüllt, da der Name Marc mehrfach vorkommt. Würde man ihn beispielsweise bei Marc Müller weglassen, würde man nicht mehr wissen, wie dieser Schüler mit Vornamen heißt, daher liegt keine Redundanz vor. Speichert man allerdings Marc Müller mit entweder all seinen oder nur einem Teil seiner Informationen in mehreren Tabellen (oder auch in einer Tabelle mehrfach), würde Redundanz vorliegen.

Wenn nun bei redundanten Daten nicht alle Kopien/Duplikate der Daten bei einer Änderung aktualisiert werden, führt dieses zu Inkonsistenzen. Diese sind der Tod einer jeden Datenbank

Allerdings kann eine „gezielte“ Redundanz durchaus sinnvoll sein und zur Performance einer Datenbank beitragen.

Auch bei der Hardware wird Redundanz gezielt eingesetzt s. RAID Systeme.

Insgesamt gilt: Redundanzen können in bestimmten Situationen nützlich sein, sollten aber ***gezielt*** und ***kontrolliert*** eingesetzt werden.



Beispieltabelle mit vielen Redundanzen.

Videotips

Primär- und Fremdschlüssel:

https://www.youtube.com/watch?v=Vgi3kfAJZQQ

Redundanzen und Anomalien: <https://www.youtube.com/watch?app=desktop&v=11bpaLfGnoQ>

https://www.youtube.com/watch?v=w5eAmcpDJsk

referentielle Integrität

https://www.youtube.com/watch?v=w5eAmcpDJsk