



# Datenbanken und SQL

(Woche 4 - Tag 3)

# Agenda

## VIEW

- Definition + Motivation
- 1. Beispielaufgabe (um die **Syntax** vorzustellen)
- 2. Beispielaufgabe (um den **komfortablen Umgang** mit VIEWS nachzuweisen)

# VIEW

# Definition + Motivation

- Ein VIEW ist – technisch betrachtet – die **Deklaration** und **Definition** einer **SELECT-Abfrage**.
- Aus Sicht der Einsetzbarkeit eines VIEWS können wir diesen allerdings auch als eine (virtuelle) **Tabelle** betrachten, die sich als Ausgabe-Tabelle des SELECT-Statements ergibt.
- VIEWS können zum einen eine Arbeitserleichterung darstellen (insbesondere dann, wenn der Code des VIEWS relativ aufwändig ist), weil man sich das erneute Codieren dieses Statements ersparen kann. Aus dieser Sicht erinnert ein VIEW dann an eine **Funktion**, die einmalig codiert, aber beliebig oft wiederverwendet werden kann.
- Zum anderen können VIEWS aber auch im Dienste der **Datensicherheit** genutzt werden. So kann zum Beispiel einem bestimmten User der Datenbank das Recht zugewiesen werden, mit bestimmten VIEWS zu arbeiten, während ihm die Nutzung anderer Tabellen untersagt bleibt.
- Aus dieser Sicht, verweisen dann VIEWS auch auf das **zweite Thema** des heutigen Tages.

# 1. Beispielaufgabe (Syntax)

## Aufgabenstellung:

Für jede Abrechnung soll die **Abrechnungs-ID**, das **Datum** sowie **Vor-** und **Nachname** des jeweiligen Kunden ausgegeben werden.

Diese Abfrage wird häufig benötigt, daher soll ein entsprechender **VIEW** eingeführt werden. Der Auftraggeber wünscht, dass dieser VIEW die Bezeichnung "**Abrechnung\_INFO**" tragen soll:

```
CREATE VIEW Abrechnung_INFO AS  
SELECT Abrechnung_ID, Datum, Vorname, Nachname  
FROM Kunde, Abrechnung WHERE Kunde.Kunde_ID=Abrechnung.Kunde_ID;
```

Dieser VIEW kann nun als **(virtuelle) Tabelle** genutzt werden:

```
SELECT * FROM Abrechnung_INFO;
```

Abrechnung_ID	Datum	Vorname	Nachname
1	2021-05-05	Elli	Rot
2	2021-10-07	Witali	Myrnow
3	2021-10-11	Vera	Deise
4	2021-10-16	Witali	Myrnow
5	2021-10-25	Eva	Hahn
6	2021-11-03	Rita	Myrnow
7	2021-11-05	Witali	Myrnow
8	2021-11-09	Vera	Deise
9	2021-11-17	Elli	Rot
10	2022-02-14	Peter	Kaufnix

## 2. Beispielaufgabe – Definition des VIEWS

### Zielsetzung des VIEWS:

Für jeden Einkauf sollen die gesamten Informationen ausgegeben werden.

**Insbesondere also:** Welcher Kunde, auf welcher Abrechnung, welches Produkt bestellte, wer dieses herstellte und wer es auslieferte.

Der entsprechende VIEW soll "Total\_Abfrage" genannt werden.

(Allein seine Größe macht bereits deutlich, wie viel Code eingespart werden kann, wenn man diesen zukünftig wiederverwendet.)

### Definition des VIEWS:

**CREATE VIEW** Total\_Abfrage **AS**

**SELECT** Kunde.Kunde\_ID,Vorname,Nachname,Email,

Abrechnung.Abrechnung\_ID,Datum,

Produkt.Produkt\_id,Produkt\_Name,Euro\_Preis,

Hersteller.Hersteller\_ID,Hersteller\_Name,

Spedition.Spedition\_ID,Spedition\_Name

**FROM** Kunde **INNER JOIN** Abrechnung **ON** Kunde.Kunde\_ID=Abrechnung.Kunde\_ID

**INNER JOIN** Abrechnung\_Produkt **ON** Abrechnung.Abrechnung\_ID=Abrechnung\_Produkt.Abrechnung\_ID

**INNER JOIN** Produkt **ON** Abrechnung\_Produkt.Produkt\_ID=Produkt.Produkt\_ID

**INNER JOIN** Hersteller **ON** Produkt.Hersteller\_ID=Hersteller.Hersteller\_ID

**INNER JOIN** Spedition **ON** Hersteller.Spedition\_ID=Spedition.Spedition\_ID;

## 2. Beispielaufgabe – Verwendung des VIEWS

### Aufgabenstellung:

Es sollen Vor- und Nachname all jener Kunden ausgegeben werden, die mindestens ein Produkt bestellten, das ihnen anschließend von der Spedition "Speedvan GmbH" ausgeliefert worden war.

Der Code einer entsprechenden Abfrage wäre ähnlich umfangreich, wie der des VIEWS **Total\_Abfrage**.

### Komfortable Alternative:

```
SELECT DISTINCT Vorname,Nachname FROM Total_Abfrage  
WHERE Spedition_Name="Speedvan GmbH";
```

#### Hinweis:

Diese Form von „Total-Abfragen“ sind enorm komfortabel, allerdings in der Praxis oft nicht praktikabel, da sie bei großen Datenbanken eine viel zu **geringe Abarbeitungs-Performance** hätten.

Vorname	Nachname
Elli	Rot
Vera	Deise
Witali	Myrnnow
Rita	Myrnnow
Eva	Hahn

# Gemeinsame Übung („Live-Coding“) -> A\_04\_03\_01



## Aufgabe\_04\_03\_01

Formulieren Sie bitte entsprechende SQL-Anweisungen für folgende Aufgabestellungen:

- Erstellen Sie bitte zunächst einen VIEW (namens „View\_1“), der für alle Abrechnungen die Abrechnungs-ID, das Datum sowie Vor- und Nachname des zugehörigen Kunden ermittelt. Nutzen Sie diesen VIEW bitte anschließend, um die folgende Abfrage zu formulieren:  
Für Jede Abrechnung soll Datum und Nachname des zugehörigen Kunden ausgegeben werden.
- Erstellen Sie bitte zunächst einen VIEW (namens „View\_2“), der für alle Produkte den Produktnamen, den Preis und zugehörigen Herstellernamen ermittelt. Nutzen Sie diesen VIEW bitte anschließend, um die folgende Abfrage zu formulieren:  
Ausgabe der drei teuersten Produkte (Produktname, Preis und zugehöriger Herstellername)
- Erstellen Sie bitte den neuen User „Admin“ (Passwort: „1234abcd“). Er soll für alle Datenbanken und für alle Tabellen auf dem MySQL-Server das Recht haben, Abfragen zu stellen, sofern er sich vom „localhost“ anmeldet.

Testen Sie dies, indem Sie sich anschließend als Admin anmelden und sich die gesamte Tabelle „Kunde“ anzeigen lassen.

WBS TRAINING AG  
Lorenzweg 5  
D-12099 Berlin  
Amtsgericht Berlin-We 68531  
Sitz der Gesellschaft: Berlin

Vorstand:  
Heinrich Kronbichler,  
Joachim Giese  
Aufsichtsrat (Vorsitz): Dr. Daniel Stadler  
USA-KNr.: DE 209 768 248

GLS Gemeinschaftsbank eG  
IBAN: DE18 4306 0967 1146 1814 00  
BIC: GENODEM33GLS



GLS zertifiziert nach  
ISO 9001:2015 und ISO 14001:2015  
Zertifizierung nach DIN EN ISO 26001:2010



# Vielen Dank für Ihre Aufmerksamkeit!

