

# Datenbankkonzept



Projektarbeit M153

Etienne Roulet  
Elia Reutlinger  
Modul 153 @GIBM Pratteln  
März 2019

<b>Management Summary</b>	<b>2</b>
<b>Selbstständigkeitserklärung</b>	<b>2</b>
<b>Namenskonzept</b>	<b>3</b>
<b>Views</b>	<b>3</b>
Nach Vorgaben	3
Abteilungsleiter	3
IT-Probleme/Tickets	4
Arbeitsplatz	5
Artikel	6
Vorträge	8
Reservationen	9
Selbst definierte	10
Fahrzeugstatus	10
Beanstandungen	11
Abteilungsmitarbeiter	12

## Management Summary

Dieses Dokument soll bestimmte Punkte zur erstellten Datenbank im Modul 153 genauer erläutern. Dazu wird das Namenskonzept aufgeführt sowie die unterschiedlichen Views, welche laut Auftrag zu erstellen waren. Diese bestehen aus 6 Stück nach Vorlage und 3 weiteren, welche individuell bestimmt werden konnten. Zu jeder View werden Details wie Eigenschaften und Datentypen, sowie 2 Beispiele zur Ausführung festgehalten.

## Selbstständigkeitserklärung

Wir bestätigen hiermit, dass wir die vorstehende Modularbeit (Konzept & ERM) selbstständig angefertigt, keine anderen als die angegebenen Hilfsmittel benutzt und sowohl wörtliche, als auch sinngemäss verwendete Textteile, Grafiken oder Bilder kenntlich gemacht haben. Diese Arbeit ist in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt worden.

Ort, Datum

Etienne Roulet

Elia Reutlinger

# Namenskonzept

Um einen reibungslosen Import und eine gute Übersicht zu garantieren, orientieren sich alle Elemente der Datenbank an einem bestimmten Namenskonzept:

1. **Tabellennamen, Eigenschaften, Beziehungen u.ä.** sind auf Deutsch, und **Umlaute** werden als ue, oe, au, etc. geschrieben.
2. **Tabellennamen** sowie der **Datenbankname** sind kleingeschrieben.
3. **Eigenschaften** haben einen grossen Anfangsbuchstaben.
4. **Eigenschaften**, welche als Fremdschlüssel auf die ID einer anderen Tabelle verweisen, haben einen zweckgebundenen Namen (also nicht unbedingt den Namen der Tabelle) und den Suffix '\_ID' (Ausser ID's bei Sub-Entitätsmengen).
5. Jede **Tabelle** (ausser bei n:m Beziehungen) hat einen Integer Primärschlüssel 'ID', auch wenn sich eine Eigenschaft in der Tabelle als Primärschlüssel eignen würde (Da ihre Werte einzigartig sind).
6. **Eigenschaften**, welche einzigartig sein sollten, tragen einen 'UNIQUE'-Index, um mehrfacheingaben zu verhindern.
7. **Views** tragen zur Unterscheidung von Tabellen einen Präfix 'view\_'.
8. **Beziehungen** sind in Gross- und Kleinbuchstaben beschriftet und lassen ihren Zweck in der Beschriftung nachvollziehen.
9. **Beschriftungen** folgen wenn nötig dem 'Snake\_Case' Format.
10. **Datentypen** wie VARCHAR haben einen sinnvollen Wertebereich. Was hier sinnvoll ist soll im Internet recherchiert werden (z.B. nach Best-Practice).
11. **Geldbeträge** werden mit dem Datentyp FLOAT gespeichert.
12. Es wird wenn möglich und sinnvoll immer generalisiert/spezialisiert.
13. Wenn bei textuellen Werten die maximale Länge nicht definitiv klar ist, wird MEDIUMTEXT verwendet.
14. ENUM-Eigenschaften nutzen keine Abkürzungen in ihren Optionen.

# Views

## Nach Vorgaben

### Abteilungsleiter

#### View-Name

view\_abteilungsleiter

#### Zweck/Voraussetzungen

Zeigen Sie von allen Abteilungsleitern den Namen, den Benutzernamen, die vollständige Adresse, den Namen der Abteilung und den Lohn an.

#### Enthaltene Attribute

Name (Ursprung)	Datentyp	Beschreibung
Abteilungsname (abteilung)	varchar	Der Name der Abteilung
Abteilungsleiter (mitarbeiter)	varchar (mehrere)	Vorname, Nachname und (in Klammern) Benutzername des Abteilungsleiters.
Anschrift (kontaktdaten)	varchar (mehrere)	Strasse und Hausnummer des Abteilungsleiters
Ort (ortschaft)	int + varchar	Postleitzahl und Ortschaft zur Anschrift
Gehalt (gehalt)	float	Gehalt des Abteilungsleiters

#### Beispiele

Suche nach Details zu einem Benutzernamen:

```
SELECT * FROM `view_abteilungsleiter`
WHERE Abteilungsleiter LIKE '%chef.desonespaar%'
```

Abteilungsname	Abteilungsleiter	Anschrift	Ort	Gehalt
Reiche Kunden	Chef Desonespaar (chef.desonespaar)	Zur Holle 22	4001 Basel	9560
Arme Kunden	Chef Desonespaar (chef.desonespaar)	Zur Holle 22	4001 Basel	9560

Suche nach Abteilungsleitern mit einem Gehalt grösser 10'000:

```
SELECT Abteilungsleiter, Gehalt FROM `view_abteilungsleiter` WHERE Gehalt > 10000
```

Abteilungsleiter	Gehalt
Chef DeCheffe (chef.decheffe)	10315.45

## IT-Probleme/Tickets

### View-Name

view\_itprobleme

### Zweck/Voraussetzungen

Zeigen Sie alle IT-Geräte an, für die von Mitarbeitern Fehlermeldungen erfasst wurden, die noch offen sind. Angezeigt werden sollen: interne Bezeichnung, Geräteart, Name des Herstellers, Modellname, Name des Mitarbeiters, der die Meldung erstellt hat, die Fehlermeldung und der Raum, in dem sich das Gerät befindet.

### Enthaltene Attribute

Name (Ursprung)	Datentyp	Beschreibung
Interne Bezeichnung (komponente_inventar)	varchar	Die interne Beschreibung des Gerätes.
Geraeteart (komponente_inventar)	enum (varchar)	Der Geräte-Typ (PC, Drucker, Switch, etc...).
Modell (komponente_katalog)	varchar	Der Modellname des Geräts.
Hersteller (komponente_katalog)	varchar	Der Hersteller des Geräts.
Erfasser (mitarbeiter)	varchar (mehrere)	Vor- und Nachname des Erfassers der Meldung.
Initiale Nachricht (nachricht)	mediumtext	Die erste Nachricht, welche zur Meldung erfasst wurde (Fehlermeldung).
Raum (raum)	int	Die ID des Raums, in welchem sich das Gerät befindet.

### Beispiele

Drucker finden, welche gemeldet wurden:

```
SELECT * FROM `view_itprobleme` WHERE Geraeteart = 'Drucker'
```

Interne Bezeichnung	Geraeteart	Modell	Hersteller	Erfasser	Initiale Nachricht	Raum
DRUCKER_01	Drucker	0815 Office Printer	Ich mache herstellen diese	Chef Desonespaar	Der Drucker isch hinüber lan.	3

Suche gemeldeten Geräten nach Hersteller:

```
SELECT Geraeteart, Modell, Hersteller FROM `view_itprobleme`  
WHERE Hersteller LIKE '%mache herstellen%'
```

Geraeteart	Modell	Hersteller
Drucker	0815 Office Printer	Ich mache herstellen diese
Monitor	MyMonitor1000+	Ich mache herstellen diese

## Arbeitsplatz

### View-Name

view\_arbeitsplatz

### Zweck/Voraussetzungen

Erstellen Sie für einen anzugebenden Arbeitsplatz eine Liste aller IT-Geräte, die ausschliesslich zu diesem Arbeitsplatz gehören. Angezeigt werden sollen: der Arbeitsplatz, der Name des Herstellers, der Modellname, die Geräteart und die interne Bezeichnung.

### Enthaltene Attribute

Name (Ursprung)	Datentyp	Beschreibung
Arbeitsplatz (arbeitsplatz)	int	ID des Arbeitsplatzes.
PC (komponente_inventar) (komponente_katalog) (firma)	varchar (mehrere) + enum	Geräte-Typ, Interne Bezeichnung, Modellname und Hersteller des PCs.
Tastatur (komponente_inventar) (komponente_katalog) (firma)	varchar (mehrere) + enum	Geräte-Typ, Interne Bezeichnung, Modellname und Hersteller der Tastatur.
Maus (komponente_inventar) (komponente_katalog) (firma)	varchar (mehrere) + enum	Geräte-Typ, Interne Bezeichnung, Modellname und Hersteller der Maus.
Monitor (komponente_inventar) (komponente_katalog) (firma)	varchar (mehrere) + enum	Geräte-Typ, Interne Bezeichnung, Modellname und Hersteller des Monitors.

### Beispiele

Alle Komponenten an einem Arbeitsplatz finden:

```
SELECT * FROM `view_arbeitsplatz` WHERE Arbeitsplatz = 2
```

Arbeitsplatz	PC	Tastatur	Maus	Monitor
2	Gereateart: PC Bezeichnung: PC_02 Modell: Einfachster Arbeits-PC Hersteller: Ich mache herstellen diese	Gereateart: Tastatur Bezeichnung: TASTA_02 Modell: Klapapapa Tastatur Hersteller: Ich mache herstellen diese	Gereateart: Maus Bezeichnung: MAUS_02 Modell: Klicketiklick Maus Hersteller: Ich mache herstellen diese	Gereateart: Monitor Bezeichnung: MONITOR_02 Modell: MyMonitor1000+ Hersteller: Ich mache herstellen diese

Arbeitsplatz zu Gerät finden (anhand interner Bezeichnung):

```
SELECT Arbeitsplatz FROM `view_arbeitsplatz`  
WHERE Monitor LIKE '%MONITOR_02%'
```

**Arbeitsplatz**

2

## Artikel

### View-Name

view\_artikel

### Zweck/Voraussetzungen

Zeigen Sie alle Artikel. Angezeigt werden sollen der Artikelname, der Name der Kategorie, der Name des jeweiligen Lieferanten, die Anzahl der Zubehörartikel.

### Enthaltene Attribute

Name (Ursprung)	Datentyp	Beschreibung
Artikelname (artikel)	varchar	Name des Artikels
Kategorie (kategorie)	varchar	Kategorie des Artikels
Lieferant (firma)	varchar	Lieferant des Artikels
Zubehoer (artikel_hat_zubehoer)	int	Anzahl Zubehör-Teile zum Artikel

### Beispiele

Artikel zu einer bestimmten Kategorie finden:

```
SELECT * FROM `view_artikel`  
WHERE Kategorie LIKE '%Entspannung%'
```

Artikelname	Kategorie	Lieferant	Zubehoer
Fidget Spinner	Entspannung	Ich bringe diese	0

Artikel mit Zubehör finden:

```
SELECT * FROM `view_artikel` WHERE Zubehoer > 0
```

Artikelname	Kategorie	Lieferant	Zubehoer
SuperHaltung	Buchhaltungsprogramme	Ich bringe diese	3

## Vorträge

### View-Name

view\_vortraege

### Zweck/Voraussetzungen

Erstellen Sie eine Liste aller gehaltenen Vorträge. Angezeigt werden sollen das Datum, die Uhrzeit, das Thema, der Name des Haupt-Referenten und die Anzahl angemeldeter Personen. Falls es einen Vorgänger und/oder Nachfolger des Vortrags gibt, sind jeweils deren Themen anzugeben.

### Enthaltene Attribute

Name (Ursprung)	Datentyp	Beschreibung
Thema (vortrag)	varchar	Thema des Vortrags.
Datum_Uhrzeit (vortrag)	datetime	Datum und Zeitpunkt des Vortrags.
Hauptreferent (mitarbeiter)	varchar (mehrere)	Vor- und Nachname des Hauptreferenten
Teilnehmer (vortrag_hat_teilnehmer)	int (Summe)	Gesamtanzahl Teilnehmer
Nachfolger (vortrag)	varchar	Thema des Nachfolgers, falls vorhanden.
Vorgaenger (vortrag)	varchar	Thema des Vorgängers, falls vorhanden.

### Beispiele

Vorträge eines bestimmten Hauptreferenten finden:

```
SELECT * FROM `view_vortraege`
WHERE Hauptreferent LIKE '%DeCheffe%'
```

Thema	Datum_Uhrzeit	Hauptreferent	Teilnehmer	Nachfolger	Vorgaenger
Tu was du net lassen kannsch	2019-03-29 12:00:00	Chef DeCheffe	34	Tu was du net lassen sollsch	NULL
Tu was du net lassen sollsch	2019-04-05 13:30:00	Chef DeCheffe	57	NULL	Tu was du net lassen kannsch

Vorträge in einem bestimmten Zeitraum finden:

```
SELECT * FROM `view_vortraege`
WHERE Datum_Uhrzeit between '2019/04/01' and '2019/04/25'
```

Thema	Datum_Uhrzeit	Hauptreferent	Teilnehmer	Nachfolger	Vorgaenger
Tu was du net lassen sollsch	2019-04-05 13:30:00	Chef DeCheffe	57	NULL	Tu was du net lassen kannsch
Was Hans niet lernt fällt selbscht hinein	2019-04-13 12:00:00	Chef Desonespaar	NULL	NULL	NULL



## Reservationen

### View-Name

view\_reservationen

### Zweck/Voraussetzungen

Zeigen Sie alle Reservationen von FCB-Jahreskarten an, die für die Kundenpflege verwendet wurden. Angezeigt werden muss das Datum, der Name des Mitarbeiters, der Name des Kunden, der Rang, der Sektor, die Reihe und die Sitzplatznummer.

### Enthaltene Attribute

Name (Ursprung)	Datentyp	Beschreibung
Ressource (ressource)	enum (varchar)	Typ der Ressource
Mitarbeiter (mitarbeiter)	varchar (mehrere)	Vor- und Nachname des Mitarbeiters
Kunde (firma)	varchar	Name des Kunden
Rang (jahreskarte)	int	Rang der Karte
Sektor (jahreskarte)	varchar	Sektor der Karte
Reihe (jahreskarte)	int	Reihe der Karte
Sitzplatznummer (jahreskarte)	int	Sitzplatznummer der Karte

### Beispiele

Reservationen eines bestimmten Mitarbeiters finden:

```
SELECT * FROM `view_reservationen`
WHERE Mitarbeiter LIKE '%DeCheffe%'
```

Ressource	Mitarbeiter	Kunde	Datum	Rang	Sektor	Reihe	Sitzplatznummer
Jahreskarte	Chef DeCheffe	Ich habe zu viel Geld	2019-03-20	4	4b	18	33

Reservationen in einem bestimmten Zeitraum finden:

```
SELECT * FROM `view_reservationen`
WHERE Datum between '2019/03/01' and '2019/03/25'
```

Ressource	Mitarbeiter	Kunde	Datum	Rang	Sektor	Reihe	Sitzplatznummer
Jahreskarte	Chef DeCheffe	Ich habe zu viel Geld	2019-03-20	4	4b	18	33

## Selbst definierte

### Fahrzeugstatus

#### View-Name

view\_fahrzeugstatus

#### Zweck/Voraussetzungen

Eine Liste, welche alle Firmenfahrzeuge mit Modellname, Kennzeichen und dem Namen der zugehörigen Versicherung auflistet. Damit die Verfügbarkeit des Fahrzeuges geprüft werden kann, soll die Anzahl an ungelösten Problemen beim Fahrzeug angezeigt werden (Ein Fahrzeug mit Problemen sollte nicht genutzt werden). Zusätzlich soll ersichtlich sein, wie viele Reparaturen das Fahrzeug schon hatte. Schlussendlich kann man sehen, welches Fahrzeug gerade genutzt werden könnte und welches am meisten Probleme macht.

#### Enthaltene Attribute

Name (Ursprung)	Datentyp	Beschreibung
Modell (dienstfahrzeug)	varchar	Modell des Fahrzeugs
Kennzeichen (dienstfahrzeug)	varchar	Kennzeichen des Fahrzeugs
Versicherung (firma)	varchar	Name der Versicherung
Probleme (problem)	int	Anzahl nicht geschlossener/behandelter Probleme
Reparaturen (reparatur)	int	Anzahl durchgeführter Reparaturen

#### Beispiele

Fahrzeug ohne Probleme finden:

```
SELECT * FROM `view_fahrzeugstatus` WHERE Probleme = 0
```

Modell	Kennzeichen	Versicherung	Probleme	Reparaturen
Tesla S3	AG123456	Zu teure Haftpflicht ohne Grund	0	0
Ford Focus	BL123456	Zu teure Haftpflicht ohne Grund	0	0

Fahrzeug mit den meisten Reparaturen finden:

```
SELECT * FROM `view_fahrzeugstatus`  
ORDER BY Reparaturen DESC LIMIT 1
```

Modell	Kennzeichen	Versicherung	Probleme	Reparaturen
Punto	BS123456	Zu teure Haftpflicht ohne Grund	1	3

## Beanstandungen

### View-Name

view\_beanstandungen

### Zweck/Voraussetzungen

Hierbei sollen für jede vorhandene Beanstandung der betroffene Artikel, die Nachricht des Kunden in der Beanstandung und der Hersteller des Artikels sowie dessen Telefonnummer aufgelistet werden. Dadurch kann der Support schnell den Hersteller kontaktieren.

### Enthaltene Attribute

Name (Ursprung)	Datentyp	Beschreibung
Artikel (artikel)	varchar	Name des Artikels
Beschreibung (beanstandung)	mediumtext	Beschreibung der Beanstandung
Hersteller (firma)	varchar	Name des Herstellers zum Artikel
Telefon (kontaktdaten)	varchar	Telefonnummer des Herstellers

### Beispiele

Alle Beanstandungen zu Artikel von bestimmtem Hersteller finden:

```
SELECT * FROM `view_beanstandungen`
WHERE Hersteller LIKE '%herstellen diese%'
```

Artikel	Beschreibung	Hersteller	Telefon
SuperHaltung	Wasch da los eyyy	Ich mache herstellen diese	+41 82 232 94 84
Origami	Dasch mega blöd odr	Ich mache herstellen diese	+41 82 232 94 84
Type P	Wurde deswegen gekündigt. Braucht das jetzt nicht ...	Ich mache herstellen diese	+41 82 232 94 84

Beanstandungen zu einem bestimmten Artikel finden:

```
SELECT * FROM `view_beanstandungen`
WHERE Artikel LIKE '%Origami%'
```

Artikel	Beschreibung	Hersteller	Telefon
Origami	Dasch mega blöd odr	Ich mache herstellen diese	+41 82 232 94 84

## Abteilungsmitarbeiter

### View-Name

view\_abteilungsmitarbeiter

### Zweck/Voraussetzungen

Diese Liste soll alle Abteilungen mit ihrem Namen anzeigen. Dazu soll ersichtlich sein, wer der Abteilungsleiter ist (Name) und wie viele Angestellte die Abteilung insgesamt hat. Wenn eine Abteilung eine Übergeordnete Abteilung hat, soll der Name von dieser ebenfalls erscheinen.

### Enthaltene Attribute

Name (Ursprung)	Datentyp	Beschreibung
Abteilung (abteilung)	varchar	Name der Abteilung
Abteilungsleiter (mitarbeiter)	varchar (mehrere)	Vor- und Nachname des Abteilungsleiters
Mitarbeiter (mitarbeiter)	int	Anzahl Mitarbeiter in der Abteilung
Uebergeordnete_Abteilung (abteilung)	varchar	Name der Übergeordneten Abteilung, falls vorhanden.

### Beispiele

Abteilung nach Abteilungsleiter finden:

```
SELECT * FROM `view_abteilungsmitarbeiter`  
WHERE Abteilungsleiter LIKE '%DeCheffe%'
```

Abteilung	Abteilungsleiter	Mitarbeiter	Uebergeordnete_Abteilung
Abzocken	Chef DeCheffe	1	NULL

Die Summe Mitarbeitenden in allen untergeordneten Abteilungen einer bestimmten übergeordneten Abteilung herausfinden:

```
SELECT Uebergeordnete_Abteilung, SUM(Mitarbeiter) AS Total_Mitarbeiter  
FROM `view_abteilungsmitarbeiter`  
WHERE Uebergeordnete_Abteilung LIKE '%Abzocken%'
```

Uebergeordnete_Abteilung	Total_Mitarbeiter
Abzocken	5