

# MODUL 164

## THEMA 7: DATENBANKPRO- JEKT

Michael Abplanalp

Version 2024

# Modul 164: Datenbanken erstellen und Daten einfügen

## Thema 7: Datenbankprojekt

### Inhaltsverzeichnis

Projektauftrag.....	2
Anforderungen.....	2
Projektablauf .....	3
Projektvorschläge.....	4
Konzeptionelles Datenmodell.....	6
Logisches Datenmodell .....	6
Sinnvolle Abfragen .....	6
Abgabe .....	7

# Projektauftrag

Sie erstellen eine Datenbank von Grund auf und durchlaufen alle Phasen, bis sich die Daten in der Datenbank befinden. Danach werden die Daten exportiert, um diese auf einem anderen System wieder zu importieren.



Sie arbeiten in Ihrer Datenbank mit wenigen Entitäten. Die Datenbank soll so konzipiert werden, dass sie auch mit grossen Datenmengen funktioniert bzw. funktionieren würde. Beispiel: Wenn Sie den Export und Import der Daten vorbereiten, dann denken Sie daran, dass die referentielle Integrität sichergestellt werden muss:

- › Stellen Sie sicher, dass die Fremdschlüsselwerte nach dem Import korrekt sind.
- › Wenn Sie beim Import die referentielle Integrität ausschalten, dann muss diese nach dem wieder Einschalten überprüft werden.

## Anforderungen

- › Es soll sich um eine realistische Datenbank handeln, also eine, die in der Realität vorkommt bzw. vorkommen könnte.
- › Das **konzeptionelle Datenmodell** enthält mind. 6 Tabellen, wovon 1 ein Ereignis (Zwischentabelle mit Attributen) sein kann.
- › Die Tabellen enthalten genügend Attribute, so dass die Datenbank sinnvoll bewirtschaftet werden kann. Denken Sie an eine Endbenutzer-Applikation, die später erstellt wird.
- › Das konzeptionelle Datenmodell wird mit diagrams.net (draw.io) oder einem ähnlichen Tool gezeichnet. Reverse-Engineering (zuerst die Tabellen erstellen und daraus das Datenmodell generieren) ist nicht erlaubt.
- › Mind. 2 m:m-Beziehungen (m:m, m:mc oder mc:mc) sind vorhanden, 1 davon soll wenn möglich ein Ereignis sein.  
Beispiel Bibliothek: Jeder Kunde kann viele Bücher ausleihen. Jedes Buch kann von vielen Kunden ausgeliehen werden. Die Ausleihe ist ein Ereignis (Datum von, Datum bis und Preis).
- › Mind. 2 1:m-Beziehungen (1:m, c:m, 1:mc oder c:mc) sind vorhanden, wovon mind. 1 eine c:m- bzw. c:mc-Beziehung sein muss..  
Beispiel Bibliothek: Jedes Buch hat einen Verlag, jeder Verlag publiziert viele Bücher.
- › Das **logische Datenmodell** wird aus dem konzeptionellen abgeleitet.
- › Das logische Datenmodell wird mit diagrams.net (draw.io) oder einem ähnlichen Tool gezeichnet. Reverse-Engineering ist nicht erlaubt.
- › Für das **physische Datenmodell** werden die CREATE DATABASE- und CREATE TABLE-Befehle erstellt und in einer Skriptdatei gespeichert. Mit dieser Datei soll die Datenbank und die Datenstrukturen auf einem anderen System (identisches RDBMS) erstellt werden können. Es ist nicht erlaubt, die Tabellen mit einem GUI-Tool zu erstellen und für die Abgabe Skripts zu generieren!
- › Die Attribute sind mit dem korrekten Datentyp definiert, es sind mind. 4 verschiedene Datentypen vorhanden (neben INT und VARCHAR/NVARCHAR mind. 2 weitere).
- › Die Primärschlüsselwerte werden von der Datenbank verwaltet (AUTO INCREMENT).
- › Die referentielle Integrität muss sichergestellt sein (Primärschlüssel-Fremdschlüssel-Beziehung).
- › Weitere Constraints:
  - › Die wichtigen Attribute werden als obligatorisch definiert (NOT NULL).
  - › Mind. 1 weiterer Constraint (UNIQUE, CHECK, DEFAULT) ist vorhanden.
- › ON DELETE-Regeln: NO ACTION und CASCADE müssen beide vorhanden sein, alle Regeln sind explizit zu definieren (also auch NO ACTION, wenn dies der Standardwert ist).

- › Jede Tabelle enthält mind. 10 Datensätze. Ausnahme: Tabellen, die nur Primär- aber nicht Detailtabellen sind, können weniger Datensätze enthalten. Z.B. wenn in einer Tabelle ein Status festgehalten wird und es nicht 10 verschiedene Stati gibt.
- › Die Daten sind so weit wie möglich realistisch und sinnvoll.
- › In Primärtabellen sollen einige Datensätze mehrfach referenziert werden. Beispiel: Bei einem Onlineshop sollen einige Kunden mehr als 1 Bestellung gemacht haben und einige Artikel sollen mehr als 1x bestellt worden sein.
- › Mind. 10 Abfragen (SELECT-Befehle) liegen vor: Abfragen, die der Endkunde durchführen würde, siehe Kapitel weiter unten. Bei jedem SELECT-Befehl soll als Kommentar kurz erklärt werden, was das Ziel der Abfrage ist.
- › Die Abfragen dürfen keine Primärschlüsselwerte in den WHERE-Bedingungen enthalten.
- › In den Abfragen sollen keine Fremdschlüsselwerte ausgegeben werden, anstelle dessen sollen die entsprechenden Datenwerte angezeigt werden (Verknüpfungen mit JOIN).
- › Die Daten werden **exportiert**, so dass diese auf einem anderen System importiert werden können: Mit INSERT-Befehlen oder als CSV-Import.



Am Ende der Projektarbeit müssen die SQL-Skripts abgegeben werden, um die Datenbank inkl. Daten auf einem anderen System zu reproduzieren. Damit alle Datenbanken parallel installiert werden können, sind folgende Vorgaben für den Namen der Datenbank zwingend einzuhalten:

[Klasse]\_[Laufnummer]\_[Datenbankname]

Beispiel: INF2022z\_01\_Musikverwaltung

Die Laufnummer wird von der Lehrperson vergeben.

Die Lehrperson ist frei, andere Vorgaben für die Benennung der Datenbanken zu definieren!

**Falls die Daten mit CSV-Dateien importiert werden, muss das Import-Verzeichnis zwingend wie folgt sein: C:\Daten\CSV\**

## Projektablauf

1. Sie bilden ein Lerntandem mit einer Kollegin/einem Kollegen
2. Sie entscheiden sich für ein Thema/Projekt, siehe nächstes Kapitel *Projektvorschläge*.
3. Sie erstellen das konzeptionelle Datenmodell aufgrund Ihrer eigenen Anforderungen.
4. Ausgehend vom konzeptionellen Datenmodell erstellen Sie das logische Datenmodell.
5. Sie überprüfen, ob sich das logische Datenmodell in der dritten Normalform befindet. Falls nicht, führen Sie die Normalisierung bis zur 3. Normalform durch.
6. Sie bereiten das physische Datenmodell vor, indem Sie sich Überlegungen zu den Datentypen, Constraints und ON DELETE-Regeln machen.
7. Sie erstellen die Datenbank und Tabellen mit SQL-DDL-Befehlen: CREATE DATABASE, CREATE TABLE. Dabei stellen Sie die referentielle Integrität mit Beziehungen (Primär- und Fremdschlüssel) sicher und definieren weitere Constraints und legen Regeln für das Löschen von Datensätzen fest.
8. Sie fügen Daten in die Tabellen ein: Es müssen mind. 10 Datensätze pro Tabelle vorhanden sein (Ausnahmen: Siehe Anforderungen oben).
9. Sie bereiten folgende Dateien für die Abgabe vor:
  - a. Das Skript, das Sie zum Erstellen der Datenbank und der Tabellen gemacht haben.
  - b. Das Skript, um die Daten in die Datenbank einzufügen.
  - c. Im Falle von CSV kommt pro Tabelle eine CSV-Datei dazu.

10. Testen Sie alle Ihre Skripts, damit es dann auf dem System der Lehrperson auch funktioniert!



Das konzeptionelle Datenmodell muss in der ersten Woche des Projektes gezeigt bzw. abgegeben werden. Sie erhalten ein Feedback, ob Sie mit dem Datenmodell weiterfahren können oder ob Korrekturen notwendig sind.

## Projektvorschläge

Am besten arbeiten Sie mit einer eigenen Projektidee. Einem Thema, das Sie interessiert. Falls Ihnen nichts Passendes einfällt, dürfen Sie gerne mit einem der Projektvorschläge arbeiten.

### Web-Börse

Sie erstellen eine Datenbank für eine Web-Börse, ähnlich wie [eBay](#) oder [Ricardo](#). Sowohl die Anbieter als auch die Bieter müssen sich im System registrieren. Die Anbieter erfassen zuerst ihre Artikel, bevor sie Angebote erstellen können.

Die Angebote haben ein Startdatum, danach sind sie für die Bieter sichtbar. Bis zum Enddatum haben die Bieter Gelegenheit, mit einem Betrag für den/die Artikel zu bieten (= Gebot). Derjenige, der am Enddatum das höchste Angebot gemacht hat, erhält den Zuschlag für den/die Artikel.

Minimale Anforderungen für die Datenspeicherung:

- › Personenangaben (Anbieter und Bieter): Name, Adresse, Kontaktangaben, usw.
- › Artikel:
  - › können in Kategorien eingeteilt werden, z.B. "Computer", "TV", "Hi-Fi", usw.
  - › haben einen Zustand, z.B. "neu", "wie neu", "sehr gut", usw.
- › Angebot: Startdatum, Enddatum, Anfangspreis, usw.
- › Jedes Gebot ist ein Ereignis und wird festgehalten mit: Datum, Preis, Bieter, usw.

Mögliche 1:m-Beziehungen: Orte-Personen, Zustand-Artikel, Artikel-Angebote.

Mögliche m:n-Beziehungen: Artikel-Kategorien, Personen-Angebote (wobei die Zwischentabelle Gebote ein Ereignis ist mit Datum und Preis).

### Verwaltung von Ferienwohnungen

Sie erstellen für einen Tourismusanbieter ein relationales Datenbankmodell für eine Ferienwohnungsverwaltung. Auf der einen Seite gibt es Eigentümer, die Wohnungen anbieten und vermieten. Auf der anderen Seite stehen die Kunden, die Wohnungen mieten können.

Minimale Anforderungen für die Datenspeicherung:

- › Eigentümerangaben, Mieterangaben: Name, Adresse, Kontaktangaben, usw.
- › Liegenschaften:
  - › haben eine Adresse
  - › es befinden sich eine oder mehrere Wohnungen darin
- › Wohnungen:
  - › haben Eigenschaften, z.B. Vermieter, Preis pro Tag oder Woche, usw.
  - › haben eine Ausstattung, z.B. 2 Schlafzimmer, Waschmaschine, Balkon, usw.
  - › werden in Kategorien eingeteilt: 1-5 Sterne
- › Jede Vermietung ist ein Ereignis und wird festgehalten mit: Datum von, Datum bis, Preis, Mieter, usw.

## Musikverwaltung

Sie erstellen für einen Anbieter von Dienstleistungen im Bereich Musik eine Applikation zur Verwaltung von Songs und Alben. Es werden Interpreten, Songs und Alben erfasst. Die Songs und Alben können auf CD, LP oder digital erscheinen. Zudem gibt es Benutzer, die sich registrieren und Listen mit Alben erstellen können. Benutzer können auch ihre eigenen Alben zum Verkauf anbieten.

Die Funktionen sollen etwa denjenigen von [Discogs](#) entsprechen, mit reduziertem Umfang (siehe Anforderungen).

Minimale Anforderungen für die Datenspeicherung:

- › Interpreten werden mit den notwendigen Informationen erfasst.
- › Es gibt Singles, EPs und Alben. Diese bestehen jeweils aus einem oder mehreren Songs und werden mit Ausgabejahr, Genre und Interpret gespeichert.
- › Jeder Song wird mit Titel und Dauer in Sekunden gespeichert.
- › Interpreten, Songs und Medien werden von einem Redaktionsteam erfasst.
- › Benutzer können sich im System registrieren.
- › Benutzer können Listen mit Singles, EPs und Alben anlegen. Dabei können Sie keine eigenen Medien erfassen, sondern nur aus den bestehenden auswählen.
- › Benutzer können Ihre Singles, EPs und Alben zu einem Fixpreis zum Verkauf anbieten (neu oder gebraucht).
- › Der Kauf eines Artikels kann weggelassen werden.

## Web-Shop Computerartikel

Sie erstellen eine Datenbank für einen Onlineshop im Bereich IT und Multimedia, ähnlich wie [Brack.ch](#) oder [digitec.ch](#). Im Shop werden Artikel in verschiedenen Kategorien angeboten. Der Kunde kann Artikel in den Warenkorb legen und kaufen.

Minimale Anforderungen für die Datenspeicherung:

- › Artikel:
  - › haben eine Beschreibung
  - › haben 1 oder mehrere Bilder hinterlegt (nur Dateipfad)
  - › haben Preise in unterschiedlichen Preisgruppen (Endkunde, Händler, Schüler usw.)
  - › sind in Kategorien und Unterkategorien eingeteilt
  - › können sich in mehreren Kategorien befinden
  - › Artikel werden wegen alten Bestellungen nicht gelöscht, aber es sollte ein Merkmal geben, das angibt, ob der Artikel noch bestellt werden kann
- › Kunden:
  - › müssen sich vor dem Kauf registrieren (Name, Adresse, Kontaktangaben)
  - › können alle getätigten Käufe einsehen
- › Jede Bestellung ist ein Ereignis und wird festgehalten mit: Kunde, Datum, Preis, usw.
  - › Bestellungen können aus mehreren Artikeln bestehen
  - › Bestellungen haben einen Status (bestellt, in Bearbeitung, geliefert, usw.)

# Konzeptionelles Datenmodell

Zur Wiederholung:

Das konzeptionelle Modell stellt eine globale, unternehmensweite Sicht auf die in der Datenbank zu verwaltenden Daten dar. Das konzeptionelle Modell wird aufgrund der Anforderungen des Kunden erstellt und dient zur Überprüfung mit dem Endkunden, ob an alles gedacht wurde. Aufgrund des Übersichtscharakters des konzeptionellen Modells enthält es keine Details.

**Das konzeptionelle Datenmodell ist neutral gegenüber dem Datenbankmanagementsystem DBMS.** Die Realisierung kann mit einem relationalen oder einem anderen Datenbanksystem (z.B. einem objektorientierten) erfolgen.

Im konzeptionellen Datenmodell werden alle vorher ermittelten Tabellen und deren Beziehung untereinander festgehalten.

Enthalten sind: Entitätsmengen, Attribute, Beziehungen.

Nicht enthalten sind: Primär- und Fremdschlüssel, Datentyp der Attribute, Zwischentabellen



Da das DBMS zu diesem Zeitpunkt noch nicht festgelegt ist, muss das konzeptionelle Datenmodell mit einem neutralen Tool erstellt werden. Vorschlag: Diagrams.net (ehemals Draw.io).

# Logisches Datenmodell

Zur Wiederholung:

Das logische Modell wird aus dem konzeptionellen heraus entwickelt und stellt das Datenmodell so dar, wie es vom DBMS gesehen wird. In unserem Fall verwenden wir das Modell der relationalen Datenbank, das sich von anderen Modellen unterscheidet. Das konzeptionelle Modell wird ergänzt mit: Primär- und Fremdschlüsseln und Zwischentabellen bei m:m-Beziehungen. Wir können auch bereits die Datentypen der Attribute bestimmen. Jedoch kann es im physikalischen Modell noch zu Anpassungen kommen, weil noch nicht klar ist, welches RDBMS eingesetzt wird.

**Für das logische Datenmodell ist es irrelevant, welches RDBMS (also welches Produkt) eingesetzt wird (MS SQL, Oracle, usw.).**



Da zu diesem Zeitpunkt noch nicht klar ist, welches RDBMS eingesetzt wird, muss das konzeptionelle Datenmodell mit einem neutralen Tool erstellt werden. Vorschlag: Diagrams.net (ehemals Draw.io).

Tools wie MySQL Workbench sind erst nach dem Erstellen des logischen Datenmodells erlaubt!

# Sinnvolle Abfragen

Was sind sinnvolle Abfragen? Das soll anhand eines Webshops erklärt werden, siehe den Projektvorschlag *Web-Shop Computerartikel* oben.

Im Web-Shop gibt es grundsätzlich die Sicht des Kunden, der Bestellungen tätigt und die Sicht der Artikel. Wie viele Artikel sind am Lager, wie oft werden diese bestellt, usw.

#### Sinnvolle und notwendige Abfragen mit SELECT:

- › Eine bestimmte Bestellung eines Kunden anzeigen → notwendig
- › Alle Bestellungen eines Kunden anzeigen → notwendig
- › Die Bestellungen von allen Kunden an einem bestimmten Datum anzeigen
- › Alle Kunden einer bestimmten Ortschaft anzeigen
- › Alle Kunden anzeigen, die einen bestimmten Artikel bestellt haben
- › Alle Kunden anzeigen, die Artikel von einer bestimmten Kategorie bestellt haben
- › Alle Artikelkategorien anzeigen
- › Alle Artikel einer bestimmten Kategorie anzeigen → notwendig
- › Alle bestellbaren Artikel einer bestimmten Kategorie anzeigen
- › Den Lagerbestand eines bestimmten Artikels anzeigen → notwendig
- › Den Lagerbestand von allen Artikeln anzeigen → notwendig
- › Den Lagerbestand von allen bestellbaren Artikeln anzeigen
- › Artikel sortiert nach Preis auf- oder absteigend anzeigen
- › Artikel einer bestimmten Kategorie sortiert nach Preis auf- oder absteigend anzeigen
- › Bestellungen mit einem bestimmten Status anzeigen → notwendig
- › usw.

## Testen und Abgabe



Testen Sie alle Skripts, bevor Sie abgeben. Entweder löschen Sie die Datenbank auf Ihrem PC und führen danach alle Skripts aus, oder Sie testen die Skripts auf einem anderen System.

**Nur so sind Sie sicher, dass Ihre Datenbank auf einem anderen System ohne Fehler reproduziert werden kann!**

Lieferobjekte:

- › Konzeptionelles Datenmodell als png-Grafikdatei
- › Logisches Datenmodell als png-Grafikdatei
- › Skript 1: CREATE DATABASE + CREATE TABLE
- › Skript 2: INSERT-Befehle bzw. Befehle für den Import der CSV-Dateien
- › Skript 3: SELECT-Befehle

Abgabe:

1. Erstellen Sie eine zip-Datei mit allen Skripts und den beiden Datenmodellen, aber ohne CSV-Dateien (falls der Datenimport mit CSV-Dateien erfolgt). Name der zip-Datei:  
`[Klasse]_[Laufnummer]_Skripts.zip`  
Beispiel: INF2023z\_01\_Skripts.zip
2. Falls der Datenimport mit CSV-Dateien erfolgt: Erstellen Sie eine zip-Datei mit allen CSV-Dateien. Name der zip-Datei:  
`[Klasse]_[Laufnummer]_CSV.zip`  
Beispiel: INF2023z\_01\_CSV.zip
3. Geben Sie die zip-Datei bzw. die beiden zip-Dateien gemäss Anweisung der Lehrperson ab.