

SQL Island – Ein Textadventure

Redaktion Informatik



© graphixel/E+/Getty Images

In dieser Unterrichtseinheit lernt Ihre Klasse grundlegende Konzepte relationaler Datenbanken und der Abfragesprache SQL kennen. Sie eignen sich Kenntnisse im Lernspiel **SQL Island** selbstständig an und dokumentieren zentrale Abfragen in einem SQL-Logbuch. Ergänzende Aufgaben fördern die sprachliche Reflexion und die systematische Sicherung wichtiger SQL-Befehle. In einer Transferaufgabe übertragen die Lernenden ihr Wissen auf die Datenbank einer Schulbibliothek. Ein Abschluss-test ermöglicht Ihnen, den Lernerfolg zu kontrollieren.

KOMPETENZPROFIL

Klassenstufe/Lernjahr:	11/12/13
Dauer:	5 Unterrichtsstunden
Inhalt:	relationale Datenbanksysteme, Attribut, Tabelle, Datensatz, Schlüssel, Primärschlüssel, Fremdschlüssel, Abfragen, SQL, Projektion, Selektion, Verbund

Auf einen Blick

1. Stunde

Thema: Einstieg und Datenbankschema

M 1 Aufbau einer Datenbank

2./3. Stunde

Thema: Spielbegleitung

M 2 Spielprotokoll und Logbuch

Benötigt: ☐ PC im Klassensatz oder für Paararbeit
☐ Link zum Spiel: <https://sql-island.informatik.uni-kl.de/>

4./5. Stunde

Thema: Sicherung und Transfer

M 3 Meine eigene Datenbank – Schulbibliothek

M 4 Tipps und Hilfestellungen

M 5 Lernerfolgskontrolle

Erklärung zu den Symbolen



Dieses Symbol markiert differenziertes Material. Wenn nicht anders ausgewiesen, befinden sich die Materialien auf mittlerem Niveau.



einfaches Niveau



mittleres Niveau



schwieriges Niveau

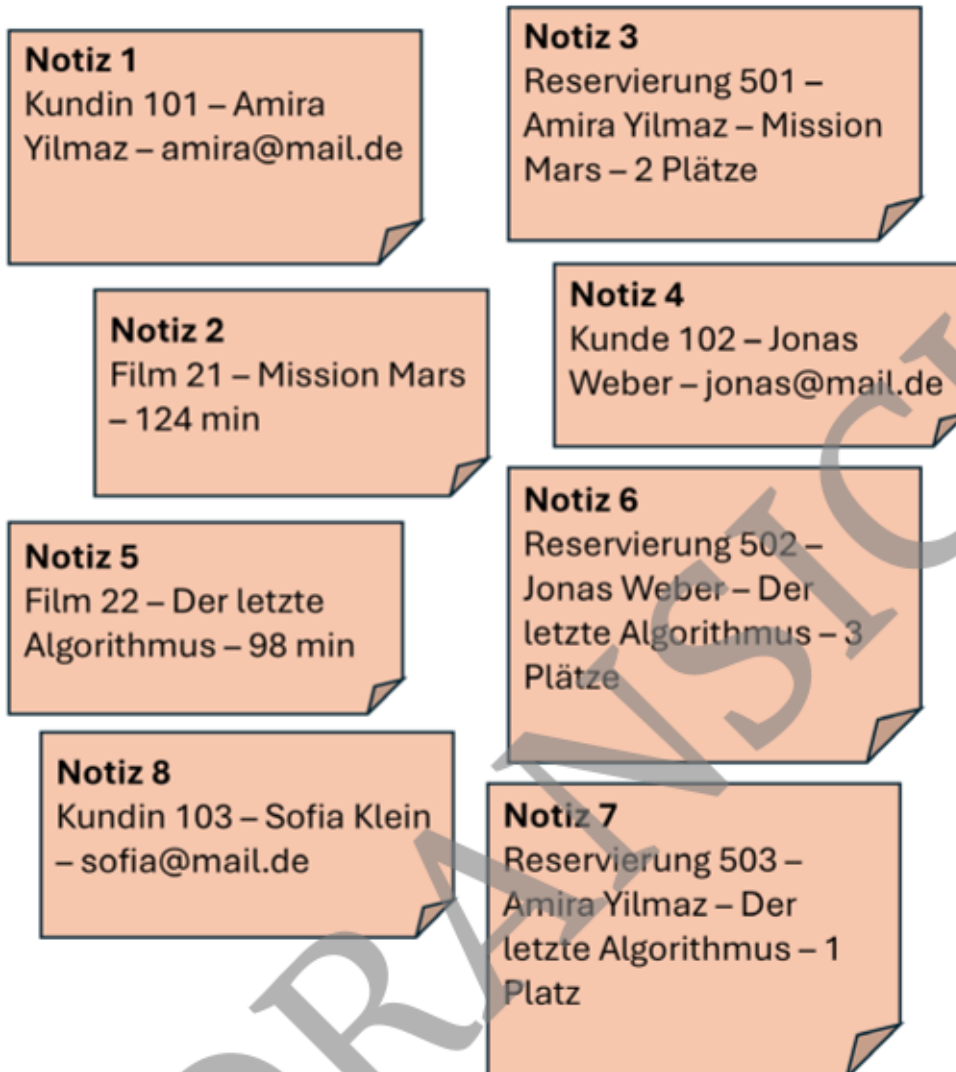
Aufbau einer Datenbank: Ordnung im Datenchaos

M 1



Das Kino am Park verwaltet Filme und Reservierungen bisher mithilfe einzelner Notizen. Das funktioniert inzwischen nicht mehr besonders gut.

Auf dem Schreibtisch liegen diese Informationen:



© RAABE

Die Kinoleitung möchte die Informationen künftig in einer **Datenbank** speichern.

Aufgabe 1

a) **Untersucht** die acht Notizen. **Ordnet** die Informationen in sinnvolle Gruppen.

Überlegt zunächst gemeinsam:

- Welche Informationen gehören zusammen?
- Welche Informationen beschreiben Personen?
- Welche beschreiben Filme?
- Welche beschreiben Reservierungen?

Aufgabe 2

Notiere vor jeder neuartigen Eingabe in eigenen Worten, was du mit der Eingabe erreichen möchtest und wie die Eingabe genau lautet.

Notiere außerdem, ob du die gewünschte Ausgabe erreicht hast. Wenn nicht, wo lag dein Fehler? Wiederholt sich lediglich ein bereits verwendetes SQL-Muster, musst du die Abfrage nicht erneut vollständig dokumentieren.

Tipp

Statt auf "Ausführen" zu klicken, kannst du auch STRG + Enter drücken.

**Protokoll**

Ziel	Eingabe	Ziel erreicht? / Fehler
Liste aller Bewohner anzeigen.	SELECT * FROM bewohner	

Meine eigene Datenbank – Schulbibliothek

M 3

Die Schulbibliothek möchte ihre Ausleihe künftig mithilfe einer relationalen Datenbank verwalten. Gespeichert werden sollen Informationen über **Bücher**, **Lernende** sowie **Ausleihen**. Du entwickelst zunächst ein geeignetes Datenbankschema. Anschließend formulierst du SQL-Befehle für typische Vorgänge in der Bibliothek.

Tipp

Auf der Seite www.w3schools.com findest du eine Übersicht über viele SQL-Anfragen, nutze sie bei Bedarf zur Recherche.



Aufgabe 1

Für die Schulbibliothek sollen drei Tabellen angelegt werden:

- **BUCH**
- **SCHUELER**
- **AUSLEIHE**

Folgende Informationen müssen gespeichert werden:

- eindeutige Nummer eines Buches
- Titel eines Buches
- Autorin oder Autor
- eindeutige Nummer einer Schülerin oder eines Schülers
- Name der Schülerin oder des Schülers
- Klasse
- eindeutige Nummer einer Ausleihe
- ausgeliehenes Buch
- Person, die das Buch ausgeliehen hat
- Ausleihdatum
- Rückgabedatum
- Säumnisgebühr

a) **Ordne** die Informationen den drei Tabellen zu und **entwickle** daraus ein Datenbankschema.

BUCH (

SCHUELER (

AUSLEIHE (

b) **Kennzeichne** in deinem Datenbankschema

- Primärschlüssel mit **PK**,
- Fremdschlüssel mit **FK**.

Tipps und Hilfestellungen zu M 3

M 4

Tipp Aufgabe 1

Aus der Liste der Informationen kannst du die Attribute der Tabellen entwickeln. Tabellen und ihre Attribute werden üblicherweise in der Form tabellenname (attribut1, attribut2, ...) angegeben.

Frage dich bei jeder Information: Beschreibt sie ein Buch, eine Person oder einen Ausleihvorgang? Eine Ausleihe muss außerdem eindeutig mit einem Buch und einer Person verbunden werden können.

Tipp Aufgabe 2

Du möchtest einen neuen Datensatz anlegen.

Erinnere dich an:

```
INSERT INTO ... (...) VALUES (...)
```

Achte darauf, Textwerte in Anführungszeichen zu setzen.

Tipp Aufgabe 3

Auch hier wird ein neuer Datensatz angelegt – diesmal in der Tabelle AUSLEIHE.

Für einen Wert, der noch nicht bekannt ist, kannst du NULL verwenden.

Tipp Aufgabe 4

Du benötigst Informationen aus allen drei Tabellen:

SCHUELER – AUSLEIHE – BUCH

Überlege:

- Welcher Fremdschlüssel in AUSLEIHE verweist auf SCHUELER?
- Welcher Fremdschlüssel in AUSLEIHE verweist auf BUCH?

Die Grundstruktur lautet:

```
SELECT ... FROM Tabelle1 JOIN Tabelle2 ON gemeinsames_Attribut
```

Tipp Aufgabe 5

Der Datensatz existiert bereits. Deshalb benötigst du keinen INSERT-Befehl.

Überlege:

- Welches Attribut soll verändert werden?
- Welcher Datensatz soll verändert werden?

Nutze:

```
UPDATE ... SET ... WHERE ...
```

Tipp Aufgabe 6

Auch hier wird ein bereits vorhandener Datensatz verändert.

Achte darauf, mit WHERE genau die richtige Ausleihe auszuwählen.