



Datenbank-Abfragen zum „School Adventure“ mit SQL



Alle theoretischen Vorüberlegungen zur „School Adventure“ - Datenbank sind abgeschlossen!

Die Datenbank ist aufgesetzt und mit Inhalt gefüllt – SUPER! 😊

Jetzt wollen wir sie auch benutzen und alle Details des „School Adventure“ abfragen können! Aber wie??? Mit SQL natürlich! Diese Datenbanksprache sollte jede/r können!

AUFGABEN:

1. **Formuliere zu folgenden Fragestellungen die passenden SQL-Anfragen!**
(Überlege Dir, welche Ergebnisse die jeweilige Anfrage liefern sollte!)
2. **Überprüfe am Rechner, ob Deine SQL-Anfragen die richtigen Ergebnisse liefern und überarbeite ggf. Deine Lösungen.**
- 3.* **Zusatz-Aufgabe für Ambitionierte:**
Formuliere die Anfragen 1 - 6 auch in der Syntax der Relationalen Algebra (mit π , σ , \bowtie)!

Folgende Relationen sind in der „School Adventure“ - Datenbank enthalten:

Reminder: \$ Schlüssel, ! Fremdschlüssel

NPCs (\$ NPCNr, NPCName)

NPC: Non Playing Character

Lehrer_innen (! NPCNr, Lieblingsgetränk)

sonstigeNPCs (! NPCNr, Aufgabe)

Schulfächer (\$ FachNr, FachName, Kürzel, Wochenstunden, Klassenstufe, ! FBNr)

Fachbereiche (\$ FBNr, FBName)

Minigames (\$ GameNr, GameName, ! FachNr)

Spieler_innen (\$ SpNr, SpName, Schlüsselanzahl, ! RaumNr)

Räume (\$ RaumNr, RaumName, Ort, Funktion)

Unterricht (!\$ FachNr, ! NPCNr, ! RaumNr)

Spielstände (!\$ GameNr, !\$ SpNr, Note, Punkte)

Aufenthaltsorte (!\$ NPCNr, !\$ RaumNr)

Assistenz (!\$ FachNr, ! NPCNr)



Hinweis: Du darfst das SQL-Cheat-Sheet von [LearnSQL.com](https://www.learnsql.com/) zur Hilfe nehmen!

**1. Abfrage der Inhalte einzelner Datenbank-Tabellen** (alle Einträge)

a) Welche Räume gibt es in der „School Adventure“ - World?

`SELECT * FROM raeume;` RA: `Raeume`

2. Selektionen (Anzeigen einzelner Zeilen einer Datenbank-Tabelle, die bestimmte Bedingungen erfüllen)

a) Welche Schulfächer haben 4 Wochenstunden?

`SELECT * FROM schulfaecher WHERE wochenstunden = 4;` RA: `σWochenstunden = 4(Schulfächer)`

b) Welche Schulfächer gibt es in der 9. Klasse?

`SELECT * FROM schulfaecher WHERE klassenstufe = 9;` RA: `σKlassenstufe = 9(Schulfächer)`

3. Projektionen (Anzeigen bestimmter einzelner Spalten einer Datenbank-Tabelle)

a) Wie heißen die Spieler_innen, die bisher das „School Adventure“ gespielt haben?

`SELECT spname FROM spieler_innen;` RA: `πSpName(Spieler_innen)`

b) Welche Aufgaben haben die sonstigen NPCs im „School Adventure“?

`SELECT aufgabe FROM sonstigenpcs;` RA: `πAufgabe(sonstigeNPCs)`

4. Natural Joins (inhaltliche Kombination für die Abfrage mehrerer Tabellen mit gleichnamigen Spalten)

a) Welche Lehrer_innen gibt es im „School Adventure“?

`SELECT * FROM dozent_innen NATURAL JOIN npcs;` RA: `Dozent_innen ⋈ NPCs`

b) Welche Minigames gibt es in der 8. Klasse?

`SELECT * FROM minigames NATURAL JOIN schulfaecher WHERE klassenstufe = 8;`
RA: `σKlassenstufe = 8(Minigames ⋈ Schulfächer)`

c) Welche Schulfächer gehören zum Fachbereich 'Gesellschaftswissenschaften'?

`SELECT * FROM schulfaecher NATURAL JOIN fachbereiche WHERE fbname = 'Gesellschaftswissenschaften';`
RA: `σFBName = 'Gesellschaftswissenschaften'(Schulfächer ⋈ Fachbereiche)`

d) Welche Aufgabe hat NPC 'Hausi'?

`SELECT aufgabe FROM sonstigenpcs NATURAL JOIN npcs WHERE npcname = 'Hausi';`
RA: `πAufgabe(σNPCName = 'Hausi'(sonstigeNPCs ⋈ NPCs))`

e) Was ist das Lieblingsgetränk von 'Math Master'?

`SELECT Lieblingsgetraenk FROM lehrer_innen NATURAL JOIN npcs WHERE npcname = 'Math Master';`
RA: `πLiebblingsgetränk(σNPCName = 'Math Master'(Lehrer_innen ⋈ NPCs))`

5. LIKE und OR (Suche nach einem (Buchstaben-)Muster/Wortausschnitt mithilfe von '%' als Platzhalter)

Welche Schulfächer haben etwas mit 'Sprache' oder 'Natur' zu tun?

`SELECT * FROM schulfaecher NATURAL JOIN fachbereiche WHERE fbname LIKE '%Sprache%' OR fbname LIKE '%Natur%';`

RA: nicht möglich (keine Platzhalter)

MUSTERLÖSUNG**6. Negation, EXCEPT und Unterabfrage**

(Anzeigen von Einträgen, die bestimmte Bedingungen NICHT erfüllen)

a) Welche Schulfächer finden nicht im 'Hauptgebäude' statt?

```
SELECT fachname, ort FROM raeume NATURAL JOIN unterricht NATURAL JOIN schulfaeher WHERE
ort != 'Hauptgebäude';
```

RA: $\pi_{\text{FachName, Ort}}((\sigma_{\text{Ort} \neq \text{'Hauptgebäude'}}(\text{Räume})) \bowtie \text{Unterricht} \bowtie \text{Schulfächer})$
 oder $\pi_{\text{FachName, Ort}}((\text{Räume} \setminus \sigma_{\text{Ort} = \text{'Hauptgebäude'}}(\text{Räume})) \bowtie \text{Unterricht} \bowtie \text{Schulfächer})$

b) Welche sonstigenNPCs haben keine Assistenz-Funktion?

```
SELECT npcname FROM npcs NATURAL JOIN (SELECT npcnr FROM lehrer_innen EXCEPT SELECT
npcnr FROM assistenz) AS xyz;
```

RA: $\pi_{\text{NPCName}}(\text{NPCs} \bowtie (\pi_{\text{NPCNr}}(\text{Lehrer_innen}) \setminus \pi_{\text{NPCNr}}(\text{Assistenz})))$

7. COUNT, SUM, MAX und AS (Aggregat-Funktionen: Anzeigen von Anzahl, Summe oder Maximum bestimmter Einträge mit neuer Spalten-Überschrift)

a) Wieviele Mini-Games gibt es in der „School Adventure“ - World?

(Ausgaben-Titel: AnzahlMinigames)

```
SELECT COUNT(*) AS AnzahlMinigames FROM minigames;
```

b) Wieviele Wochenstunden müssen im „School Adventure“ insgesamt absolviert werden?

(Ausgaben-Titel: GesamtanzahlWochenstunden)

```
SELECT SUM(wochenstunden) AS GesamtanzahlWochenstunden FROM schulfaeher;
```

c) Wie heißt das Fach mit den meisten Wochenstunden?

```
SELECT fachname FROM schulfaeher WHERE wochenstunden = (SELECT MAX(wochenstunden)
FROM schulfaeher);
```

8. ORDER BY ASC/DESC (Sortierung der angezeigten Einträge aufsteigend/absteigend)

Gesucht sind die Namen, Klassenstufen und Wochenstunden aller Fächer von 'Mr. Multitalent' absteigend sortiert nach Wochenstunden-Anzahl!

```
SELECT fachname, klassenstufe, wochenstunden FROM schulfaeher NATURAL JOIN unterricht
NATURAL JOIN npcs WHERE npcname = 'Mr. Multitalent' ORDER BY wochenstunden DESC;
```

9. GROUP BY (Gruppierung ausgewählter Einträge nach bestimmten Attributen)

Wieviele Fächer gibt es pro Raum?

```
SELECT raumname, COUNT(*) AS AnzahlFaecher FROM raeume NATURAL JOIN unterricht GROUP BY
raumname ORDER BY COUNT(*);
```

10. HAVING, NOT, BETWEEN und AVG (weitere Operatoren im Zusammenhang mit GROUP BY)

Welche Spieler_innen haben einen Gesamt-Notendurchschnitt, der nicht zwischen 2.0 und 4.0 liegt? (Sortierung nach Gesamt-Notendurchschnitt aufsteigend, also bester Schnitt zuerst)

```
SELECT sname FROM spieler_innen NATURAL JOIN spielstaende GROUP BY sname
HAVING AVG(note) NOT BETWEEN 2.0 AND 4.0 ORDER BY AVG(note), sname;
```

