

# Aufgaben-Sequenz zur LWBAdventure-Datenbank

## RELATIONEN

NPCs (\$ NPCNr, NPCName)

*NPC: Non Playing Character*

Dozent\_innen (! NPCNr, Lieblingsgetränk)

sonstigeNPCs (! NPCNr, Aufgabe)

Veranstaltungen (\$ VNr, VName, Kürzel, SWS, Semester, ! GebietNr)

Themengebiete (\$ GebietNr, GebietName)

Minigames (\$ GameNr, GameName, ! VNr)

Spieler\_innen (\$ SpNr, SpName, Schlüsselanzahl, ! RaumNr)

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

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

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

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

Assistenz (! \$ VNr, ! NPCNr)

## ANFRAGEN

### 1a. Welche Räume gibt es in der LWB-Adventure-World?

Räume

RA: `/listing Räume`

DRC: { RaumNr, RaumName, Ort, Funktion | Räume (RaumNr, RaumName, Ort, Funktion) }

SQL: `SELECT * FROM raume;`

### 1b. Welche Aufgaben haben die sonstigen NPCs im LWB-Adventure?

$\pi_{\text{Aufgabe}}(\text{sonstigeNPCs})$

RA: `project Aufgabe (sonstigeNPCs);`

DRC: { Aufgabe | ( $\exists \text{NPCNr}$ ) (sonstigeNPCs (NPCNr, Aufgabe)) }

SQL: `SELECT aufgabe FROM sonstigenpcs;`

### 2a. Welche Lehrveranstaltungen haben 6 SWS?

$\sigma_{\text{SWS} = 6}(\text{Veranstaltungen})$

RA: `select SWS = 6 (Veranstaltungen);`

DRC: { VNr, VName, Kürzel, SWS, GebietNr | Veranstaltungen (VNr, VName, Kürzel, SWS, GebietNr)  $\wedge$  SWS = 6 }

SQL: `SELECT * FROM veranstaltungen WHERE sws = 6;`

### 2b. Welche Lehrveranstaltungen gibt es im 4. Semester?

$\sigma_{\text{Semester} = 4}(\text{Veranstaltungen})$

RA: `select Semester = 4 (Veranstaltungen);`

DRC: { VNr, VName, Kürzel, SWS, Semester, GebietNr | Veranstaltungen (VNr, VName, Kürzel, SWS, Semester, GebietNr)  $\wedge$  Semester = 4 }

SQL: `SELECT * FROM veranstaltungen WHERE semester = 4;`

## 2c. Welche Minigames gibt es im 4. Semester?

$\sigma_{\text{Semester} = 4}(\text{Minigames} \bowtie \text{Veranstaltungen})$

RA: `select Semester = 4 (Minigames njoin Veranstaltungen);`

DRC: { GameNr, GameName, VNr, VName, Kürzel, SWS, Semester, GebietNr | Minigames (GameNr, GameName)  $\wedge$  Veranstaltungen (VNr, VName, Kürzel, SWS, Semester, GebietNr)  $\wedge$  Semester = 4 }

SQL: `SELECT * FROM minigames NATURAL JOIN veranstaltungen WHERE semester = 4;`

## 3a. Wie heißen die Spieler\_innen, die bisher das LWB-Adventure gespielt haben?

$\pi_{\text{SpName}}(\text{Spieler\_innen})$

RA: `project SpName (Spieler_innen);`

DRC: { SpName | ( $\exists$  SpNr)( $\exists$  Schlüsselanzahl)( $\exists$  RaumNr) (Spieler\_innen (SpNr, SpName, Schlüsselanzahl, RaumNr)) }

SQL: `SELECT spname FROM spieler_innen;`

## 3b. Wie heißen die Dozenten im LWB-Adventure?

$\pi_{\text{NPCName}}(\text{Dozent\_innen} \bowtie \text{NPCs})$

RA: `project NPCName (Dozent_innen njoin NPCs);`

DRC: { NPCName | ( $\exists$  NPCNr)( $\exists$  Lieblingsgetränk) (Dozent\_innen (NPCNr, Lieblingsgetränk)  $\wedge$  NPCs (NPCNr, NPCName)) }

SQL: `SELECT npcname FROM dozent_innen NATURAL JOIN npcs;`

## 3c. Welche Aufgabe hat NPC 'Heidi'?

$\pi_{\text{Aufgabe}}(\sigma_{\text{NPCName} = \text{'Heidi'}}(\text{sonstigeNPCs} \bowtie \text{NPCs}))$

RA: `project Aufgabe (select NPCName = 'Heidi' (sonstigeNPCs njoin NPCs));`

DRC: { Aufgabe | ( $\exists$  NPCNr)( $\exists$  NPCName) (sonstigeNPCs (NPCNr, Aufgabe)  $\wedge$  NPCs (NPCNr, NPCName)  $\wedge$  NPCName = 'Heidi') }

SQL: `SELECT aufgabe FROM sonstigenpcs NATURAL JOIN npcs WHERE npcname = 'Heidi';`

## 4a. Welche Lehrveranstaltungen gehören zum Themengebiet 'Programmierung'?

$\sigma_{\text{GebietName} = \text{'Programmierung'}}(\text{Veranstaltungen} \bowtie \text{Themengebiete})$

RA: `select GebietName = 'Programmierung' (Veranstaltungen njoin Themengebiete);`

DRC: { VNr, VName, Kürzel, SWS, Semester, GebietNr, GebietName | Veranstaltungen (VNr, VName, Kürzel, SWS, Semester, GebietNr)  $\wedge$  Themengebiete (GebietNr, GebietName)  $\wedge$  GebietName = 'Programmierung' }

SQL: `SELECT * FROM veranstaltungen NATURAL JOIN themengebiete WHERE gebietname = 'Programmierung';`

## 4b. Welche Lehrveranstaltungen haben etwas mit 'Daten' oder 'Programmierung' zu tun?

RA: Wildcards in RA nicht möglich!

DRC: Wildcards in DRC nicht möglich!

SQL: `SELECT * FROM veranstaltungen WHERE vname LIKE '%Daten%' OR vname LIKE '%Programmierung%';`  
oder

SQL: `SELECT * FROM veranstaltungen NATURAL JOIN themengebiete WHERE gebietname LIKE '%Daten%' OR gebietname LIKE '%Programmierung%';`

## 5. Was ist das Lieblingsgetränk von Darth Schmidter?

$\pi_{\text{Lieblingsgetränk}}(\sigma_{\text{NPCName} = \text{'Darth Schmidter'}}(\text{Dozent\_innen} \bowtie \text{NPCs}))$

RA: `project Lieblingsgetraenk (select NPCName = 'Darth Schmidter' (Dozent_innen njoin NPCs));`

DRC: `{ Lieblingsgetränk | (∃ NPCNr)(∃ NPCName) (Dozent_innen (NPCNr, Lieblingsgetränk) ∧ NPCs (NPCNr, NPCName) ∧ NPCName = 'Darth Schmidter') }`

SQL: `SELECT Lieblingsgetraenk FROM dozent_innen NATURAL JOIN npcs WHERE npcname = 'Darth Schmidter';`

## 6. Welche Lehrveranstaltungen finden nicht in der 'FU Berlin' statt?

$\pi_{\text{VName, Ort}}((\sigma_{\text{Ort} \neq \text{'FU Berlin'}}(\text{Räume})) \bowtie \text{Unterricht} \bowtie \text{Veranstaltungen})$

RA: `project VName, Ort ((select Ort != 'FU Berlin' (Raeume)) njoin Unterricht njoin Veranstaltungen);`

oder mit Differenz:

$\pi_{\text{VName, Ort}}((\text{Räume} \setminus \sigma_{\text{Ort} = \text{'FU Berlin'}}(\text{Räume})) \bowtie \text{Unterricht} \bowtie \text{Veranstaltungen})$

RA: `project VName, Ort ((Raeume difference (select Ort = 'FU Berlin' (Raeume))) njoin Unterricht njoin Veranstaltungen);`

DRC: `{ VName, Semester, Ort | (∃ RaumNr)(∃ RaumName)(∃ Funktion)(∃ VNr)(∃ NPCNr)(∃ VName)(∃ Kürzel) (∃ SWS)(∃ GebietNr) (Räume (RaumNr, RaumName, Ort, Funktion) ∧ Unterricht (VNr, NPCNr, RaumNr) ∧ Veranstaltungen (VNr, VName, Kürzel, SWS, Semester, GebietNr) ∧ Ort ≠ 'FU Berlin') }`

SQL: `SELECT vname, semester, ort FROM raeume NATURAL JOIN unterricht NATURAL JOIN veranstaltungen WHERE ort != 'FU Berlin';`

## 7. Welche Dozenten sind in der LWB nur leitend tätig und machen keine Assistenz?

$\pi_{\text{NPCName}}(\text{NPCs} \bowtie (\pi_{\text{NPCNr}}(\text{Dozent\_innen}) \setminus \pi_{\text{NPCNr}}(\text{Assistenz})))$

RA: `project NPCName (NPCs njoin (project NPCNr (Dozent_innen) difference project NPCNr (Assistenz)));`

DRC: `{ NPCName | (¬∃ NPCNr) (Assistenz (VNr, NPCNr)) ∧ (∃ NPCNr)(∃ Lieblingsgetränk) (Dozent_innen (NPCNr, Lieblingsgetränk) ∧ NPCs (NPCNr, NPCName)) }`

SQL: `SELECT npcname FROM npcs NATURAL JOIN (SELECT npcnr FROM dozent_innen EXCEPT SELECT npcnr FROM assistenz) AS xyz;`

Kommentar: Hier braucht es einen Alias, damit der NATURAL JOIN mit der Unterabfrage funktioniert. Die Bezeichnung ist jedoch egal, da nur auf den NPCNamen projiziert wird.

## Anfragen, die nur mit erweiterter relationaler Algebra beschrieben werden können:

### 8. Wieviele Mini-Games gibt es in der LWB-Adventure-World? (Ausgaben-Titel: AnzahlMinigames)

$\text{AnzahlMinigames} \leftarrow \gamma_{\text{COUNT}(*)}(\text{Minigames})$

RA: `rename Anzahl (Minigames) (group_by [] count(*) true (Minigames));`

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

### 9. Wieviele SWS müssen in der LWB insgesamt absolviert werden?

(Ausgaben-Titel: GesamtanzahlSWS)

$\text{GesamtanzahlSWS} \leftarrow \gamma_{\text{SUM(SWS)}}(\text{Veranstaltungen})$

RA: `rename Gesamtanzahl (SWS) (group_by [] sum(SWS) true (Veranstaltungen));`

SQL: `SELECT SUM(sws) AS GesamtanzahlSWS FROM veranstaltungen;`

### 10. Wie heißt die Veranstaltung mit den meisten SWS?

$\pi_{Vname}(\text{select SWS} = \text{MaxAnzahl.SWS} ((\text{MaxAnzahl.SWS} \leftarrow \gamma_{\text{MAX(SWS)}}(\text{Veranstaltungen})) \bowtie \text{Veranstaltungen}))$

RA: `project VName (select SWS = MaxAnzahl.SWS (rename MaxAnzahl (SWS) (group_by [] max(SWS) true (Veranstaltungen)) njoin Veranstaltungen));`

SQL: `SELECT vname FROM veranstaltungen WHERE sws = (SELECT MAX(sws) FROM veranstaltungen);`

### 11. Gesucht sind die Namen, Semester und SWS aller Veranstaltungen von Winnie the K absteigend sortiert nach SWS-Anzahl!

$\tau_{\text{SWS}} \downarrow (\pi_{VName, SWS, Semester}(\sigma_{NPCName = 'Winnie the K'}(\text{Veranstaltungen} \bowtie \text{Unterricht} \bowtie \text{NPCs})))$

$\tau_{\text{SWS}} \downarrow (\pi_{VName, SWS, Semester}(\rho_{Semester \leftarrow \text{Raumnr}}(\sigma_{NPCName = 'Winnie the K'}(\text{Veranstaltungen} \bowtie \text{Unterricht} \bowtie \text{NPCs}))))$

RA: `sort SWS desc (project VName, SWS, Semester (select NPCName = 'Winnie the K' (Veranstaltungen njoin Unterricht njoin NPCs)));`

SQL: `SELECT vname, sws, semester FROM veranstaltungen NATURAL JOIN unterricht NATURAL JOIN npc WHERE npcname = 'Winnie the K' ORDER BY sws DESC;`

### 12. Wieviele Veranstaltungen gibt es pro Standort?

$\tau_{\text{AnzahlVeranstaltungen}}(\gamma_{\text{Ort}}, \text{AnzahlVeranstaltungen} \leftarrow \text{COUNT}(*)(\text{Räume} \bowtie \text{Unterricht}))$

RA: `sort AnzahlVeranstaltungen (rename Standorte (OrtName, AnzahlVeranstaltungen) (group_by Ort Ort, count(VNr) true (Raeume njoin Unterricht)));`

SQL: `SELECT ort, COUNT(*) AS AnzahlVeranstaltungen FROM raeume NATURAL JOIN unterricht GROUP BY ort ORDER BY COUNT(*);`

### 13. 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)

$Q1 \leftarrow \gamma_{\text{SpName}}, \text{AVG}(\text{Note})(\text{Spielstände} \bowtie \text{Spieler\_innen})$

$Q2 \leftarrow \rho_{\text{Schnitt}} \leftarrow \text{AVG}(\text{Note})(Q1)$

$Q3 \leftarrow \sigma_{\text{Schnitt} < 2.0 \wedge \text{Schnitt} > 4.0}(Q2)$

$Q4 \leftarrow \tau_{\text{Schnitt}, \text{SpName}}(Q3)$

$Q5 \leftarrow \pi_{\text{SpName}}(Q4)$

oder

$\pi_{\text{SpName}}(\tau_{\text{Schnitt}, \text{SpName}}(\sigma_{\text{Schnitt} < 2.0 \wedge \text{Schnitt} > 4.0}(\rho_{\text{Schnitt}} \leftarrow \text{AVG}(\text{Note})(\gamma_{\text{SpName}}, \text{AVG}(\text{Note})(\text{Spielstände} \bowtie \text{Spieler\_innen}))))))$

RA: `project SpName (sort Schnitt, SpName (select Schnitt < 2.0 or Schnitt > 4.0 (rename Noten (SpName, Schnitt) (group_by SpName SpName, avg(Note) true (Spielstaende njoin Spieler_innen))));`

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

**Kommentar:** Hier könnte die Projektion auch weggelassen werden, wenn die Ausgabe außer den SpNamen auch den Notendurchschnitt enthalten soll.

$\tau_{\text{Schnitt}, \text{SpName}}(\sigma_{\text{Schnitt} < 2.0 \wedge \text{Schnitt} > 4.0}(\rho_{\text{Schnitt}} \leftarrow \text{AVG}(\text{Note})(\gamma_{\text{SpName}}, \text{AVG}(\text{Note})(\text{Spielstände} \bowtie \text{Spieler\_innen}))))$

RA: `sort Schnitt, SpName (select Schnitt < 2.0 or Schnitt > 4.0 (rename Noten (SpName, Schnitt) (group_by SpName SpName, avg(Note) true (Spielstaende njoin Spieler_innen))));`

SQL: `SELECT SpName, ROUND(AVG(note), 2) AS Notendurchschnitt FROM spieler_innen NATURAL JOIN spielstaende GROUP BY spname HAVING AVG(note) NOT BETWEEN 2.0 AND 4.0 ORDER BY AVG(note), spname;`