

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

`/listing Räume`

`SELECT * FROM räume;`

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

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

`project Aufgabe (sonstigeNPCs);`

`SELECT aufgabe FROM sonstigenpcs;`

2a. Welche Lehrveranstaltungen haben 6 SWS?

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

`select SWS = 6 (Veranstaltungen);`

`SELECT * FROM veranstaltungen WHERE sws = 6;`

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

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

`select Semester = 4 (Veranstaltungen);`

`SELECT * FROM veranstaltungen WHERE semester = 4;`

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

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

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

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

3a. Wie heißen die Spieler_innen, die bisher das LWB-Adventure gespielt haben?

$\pi_{\text{SpName}}(\text{Spieler_innen})$

`project SpName (Spieler_innen);`

`SELECT spname FROM spieler_innen;`

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

$\pi_{\text{NPCName}}(\text{Dozent_innen} \bowtie \text{NPCs})$

`project NPCName (Dozent_innen njoin NPCs);`

`SELECT npcname FROM dozent_innen NATURAL JOIN npcs;`

3c. Welche Aufgabe hat NPC 'Heidi'?

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

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

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

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

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

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

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

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

`% Wildcards in RA nicht möglich!`

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

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

5. Was ist das Lieblingsgetränk von Darth Schmidter?

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

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

`SELECT lieblingsgetraenk FROM dozent_innen NATURAL JOIN npcs WHERE npcname = 'Darth Schmidter';`

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

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

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

oder mit Differenz:

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

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

```
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?

$\pi_{NPCName}(NPCs \bowtie (\pi_{NPCNr}(Dozent_innen) \setminus \pi_{NPCNr}(Assistenz)))$

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

```
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)

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

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

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

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

(Ausgaben-Titel: GesamtanzahlSWS)

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

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

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

10. Wie heißt die Veranstaltung mit den meisten SWS?

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

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

```
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!

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

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

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

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

12. Wieviele Veranstaltungen gibt es pro Standort?

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

sort AnzahlVeranstaltungen (rename Standorte (OrtName, AnzahlVeranstaltungen) (group_by Ort Ort, count(VNr) true (Räume njoin Unterricht)));

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 $\leftarrow \gamma_{SpName, AVG(Note)}(Spielstände \bowtie Spieler_innen)$

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

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

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

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

oder

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

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)))));

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.

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

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))));

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;