

Datenbanksysteme: Übung 6

Tutor: Miao Wang

Lisa Dohrmann (4130066), Adrian Neumann (4140810), Naja v. Schmude (4127652)

29. Mai 2008

Aufgabe 1

- `SELECT p.Vorname, a.Name FROM Autoren NATURAL JOIN Personen p
JOIN Abenteuer a ON Autoren.ID = a.Autor;`
- `SELECT DISTINCT w.Name FROM Welten w JOIN Abenteuer a ON w.ID=a.Welt;`
- `SELECT w.Genre FROM Welten w JOIN Abenteuer a ON w.ID = a.Welt
JOIN Autor au ON a.Autor = au.ID JOIN Spieler s ON au.ID = s.ID
JOIN Spielfigur sf ON sf.Spieler = s.ID;`
- `SELECT Vorname, Nachname FROM Spielfigur EXCEPT
(SELECT sf.Vorname, sf.nachname FROM Spielfigur sf
JOIN Welten w ON sf.welt = w.ID JOIN Abenteuer a ON w.ID = a.Welt
JOIN Autorn au ON a.Autor = au.ID JOIN Personen p ON au.ID = p.ID
WHERE p.Vorname = 'Klaus' AND p.Nachname='Schmidt')
WHERE alter > 40;`
- `SELECT p.Vorname, sf.Vorname FROM Person p NATURAL JOIN Spieler s
JOIN Spielfigur sf ON s.ID = sf.ID JOIN Abenteuer a ON sf.Welt=a.Welt
JOIN Autor au ON au.ID = a.Autor;`
- `SELECT * FROM Person p JOIN Spieler s ON p.ID != s.ID
UNION SELECT * FROM Person NATURAL JOIN Spieler;`

Aufgabe 2

1. `CREATE FUNCTION jahresende RETURNS SETOF (Movie.title%TYPE, Movie.price_Day%TYPE) AS $$
DECLARE
title Movie.title%TYPE
price_All Movie.price_Day%TYPE;
BEGIN
FOR (title, price_All) IN SELECT M.title, SUM(M.price_Day * (R.until_date-R.from_date)) FR
JOIN Tape JOIN Rental R ON Tape.t_id = R.tape_id GROUP BY M.title
UNION SELECT M.cat, SUM(M.price_Day * (R.until_date-R.from_date))
FROM Movie M NATURAL JOIN Tape JOIN Rental R
ON Tape.t_id = R.tape_id GROUP BY M.cat
LOOP
RETURN NEXT (title, price_All);
END LOOP;
RETURN;`

```
END;  
$$ LANGUAGE plpgsql;
```