Ausarbeitung 05

Niklas Vest

November 6, 2018

Abfragen

```
1 -- 1.1
 2 SELECT release_year, title,
          LISTAGG(SUBSTR(first_name, 0, 1) || '. ' || last_name, ', ')
 3
 4
              WITHIN GROUP (ORDER BY last_name) AS actors
 5 FROM film
          INNER JOIN film_actor USING (film_id)
          INNER JOIN actor USING (actor_id)
 8 GROUP BY film_id, title, release_year
 9 ORDER BY release_year ASC, title ASC;
10
12 SELECT first_name || ' ' || last_name AS customer,
          LISTAGG(title || ' (' || to_char(release_year) || ')', ',')
13
              WITHIN GROUP (ORDER BY release_year DESC) AS films
14
15 FROM rental
          INNER JOIN customer USING (customer_id)
          INNER JOIN inventory USING (inventory_id)
17
          INNER JOIN film USING (film_id)
18
19 WHERE (
             (SELECT sysdate
20
             FROM dual) - rental_date) / 365 <= 4
22 GROUP BY customer_id, first_name, last_name
23 ORDER BY last_name ASC;
24
25 -- 1.3
26 WITH at_least_three AS
27
       (SELECT customer_id,
28
               category_id,
               COUNT(film_id) AS films_of_category
29
30
        FROM customer c
               INNER JOIN store s ON c.store_id = s.store_id
31
32
               INNER JOIN address a ON s.address_id = a.address_id
33
               INNER JOIN city USING (city_id)
               INNER JOIN rental USING (customer_id)
34
               INNER JOIN inventory USING (inventory_id)
               INNER JOIN film USING (film_id)
36
37
               INNER JOIN film_category USING (film_id)
        WHERE city = 'Linz'
38
        GROUP BY customer_id, category_id
39
        HAVING COUNT(film_id) >= 3)
```

```
41 SELECT first_name || ' ' || last_name AS customer,
          LISTAGG(name, ', ')
42
              WITHIN GROUP (ORDER BY name ASC) AS interests
43
44 FROM at_least_three
          INNER JOIN customer USING (customer_id)
45
46
          INNER JOIN category USING (category_id)
47 GROUP BY customer_id, first_name, last_name
48 ORDER BY last_name ASC;
50 -- 2.1
51 CREATE OR REPLACE VIEW ue5_02a AS
52 WITH revenues AS
53
   (SELECT store_id,
54
             category_id,
             SUM(amount) AS revenue
55
56
      FROM payment
57
             INNER JOIN rental USING (rental_id)
             INNER JOIN inventory USING (inventory_id)
58
59
             INNER JOIN store USING (store_id)
60
             INNER JOIN film_category USING (film_id)
61
      WHERE store_id IN (1, 2)
      GROUP BY store_id, category_id
62
63
      ORDER BY category_id ASC)
     SELECT category_id,
64
65
            store1_revenue,
66
            store2_revenue,
67
            ROUND(store1_revenue / store2_revenue, 2) AS s1_relative_to_s2
68
          (SELECT *
69
70
           FROM revenues
71
               PIVOT
               -- it's only one row anyway
72
73
               -- so AVG should be fine
               (AVG(revenue) AS revenue
74
75
               FOR store_id
76
               IN (1 AS store1,
77
                 2 AS store2)));
78
79 SELECT * FROM ue5_02a;
80
81 -- 2.2
82 CREATE MATERIALIZED VIEW ue05_02b
83 BUILD IMMEDIATE
84 REFRESH COMPLETE
85 ON DEMAND
86 AS
       (SELECT *
87
        FROM ue5_02a);
88
89
90 SELECT * FROM ue05_02b;
91
92 DROP MATERIALIZED VIEW ue05_02b;
93 -- 2.3
94 CREATE MATERIALIZED VIEW ue05_02c
95 REFRESH COMPLETE
96 START WITH TRUNC(SYSDATE) + ((1/24/60) * (23 * 60 + 30))
97 NEXT TRUNC(SYSDATE) + 1 + ((1/24/60) * (23 * 60 + 30))
```

```
98
     AS
99
        (SELECT *
100
        FROM ue05_02b);
101
102 DROP MATERIALIZED VIEW ue05_02c;
103
104
105 SELECT column_name,
106
           data_type,
107
           data_length,
108
           nullable,
           data_precision AS significant_digits,
109
110
           data_scale AS fixed_point_digits
111 FROM user_tab_columns
112 WHERE table_name = &tab_name;
113
114 -- 3.2
115 COMMENT ON TABLE store IS 'This is one hell of a store collection.';
116 SELECT table_name,
117
           comments
118 FROM user_tab_comments
119 WHERE table_name = 'STORE';
120
121 -- 3.3
122 SELECT cols.column_name,
123
           cols.constraint_name,
124
           cons.constraint_type,
           cons.search_condition,
125
126
           cons.status
127 FROM user_constraints cons INNER JOIN user_cons_columns cols
128
             ON cons.constraint_name = cols.constraint_name
129 WHERE cons.table_name = &tab_name;
130
131 -- 4.1
132 CREATE SEQUENCE probe_seq;
133
134 CREATE TABLE probe (
135 id VARCHAR2(50),
136
    timestamp DATE NOT NULL,
137
     type VARCHAR2(50) NOT NULL,
      "comment" VARCHAR2(255),
138
139
    CONSTRAINT probe_pk PRIMARY KEY (id)
140);
141
142 -- There must be some sort of information about
143 -- who is doing the insert because otherwise the
144 -- DB could not distinguish the teams since I assume
145 -- there is only one DB user for all teams (as the
146 -- task description suggests). Hence I decided to
147 -- concatenate the next sequence value with a unique
148 -- team descriptor.
149 INSERT
       INTO probe (id, timestamp, type)
150
     VALUES (probe_seq.nextval || '_team1', SYSDATE, 'Milch');
151
152
153 INSERT
154 INTO probe (id, timestamp, type)
```

```
155 VALUES (probe_seq.nextval || '_team2', SYSDATE, 'Eier');
156
157 SELECT * FROM probe;
158
159 DROP TABLE probe;
160 DROP SEQUENCE probe_seq;
```

Ergebnisse

	RELEASE_YEAR \$	TITLE \$	ACTORS \$
1	1983	BORN SPINAL	M. ALLEN, R. JOHANSSON, K. PALT
2	1983	BOWFINGER GABLES	K. BERRY, C. HUNT, M. MCCONAUGH
3	1983	BUNCH MINDS	K. BERRY, C. BRIDGES, L. BULLOC
4	1983	CHITTY LOCK	V. BOLGER, S. DAVIS, N. DEGENER
5	1983	CIDER DESIRE	J. CHASE, F. DAY-LEWIS, J. DEGE
6	1983	CLOSER BANG	J. DEGENERES, J. FAWCETT, E. GU
7	1983	DIVIDE MONSTER	C. AKROYD, H. BERRY, A. DREYFUS
8	1983	DRAGON SQUAD	A. CRONYN, S. DAVIS, S. DEPP, E
9	1983	DRUMS DYNAMITE	V. BERGEN, J. CRUISE, L. DEE, M
10	1983	INNOCENT USUAL	C. AKROYD, J. PITT, D. TORN
11	1983	LOST BIRD	C. BIRCH, K. JOVOVICH, K. TORN
12	1983	MAJESTIC FLOATS	N. HOPKINS, E. MANSFIELD, G. SI
13	1983	MOONSHINE CABIN	M. CARREY, J. DEGENERES, G. GOO
14	1983	NASH CHOCOLAT	H. BALE, J. DEAN, H. GARLAND, B
15	1983	PET HAIINTING	I RRODY 1 DEGENERES N DEGE

Figure 1: Resultat der Abfrage 1.1

	CUSTOMER \$	FILMS
1	RAFAEL ABNEY	CHOCOLAT HARRY (2006), CONEHEADS SMOOCHY (20
2	NATHANIEL ADAM	TIGHTS DAWN (2006), GATHERING CALENDAR (2005
3	KATHLEEN ADAMS	SPY MILE (2008), SWARM GOLD (2003), SUNDANCE
4	DIANA ALEXANDER	REBEL AIRPORT (2008), SHOW LORD (2006), JASON
5	GORDON ALLARD	SILVERADO GOLDFINGER (2008),BINGO TALENTED
6	SHIRLEY ALLEN	BIRD INDEPENDENCE (2006), ORANGE GRAPES (200
7	CHARLENE ALVAREZ	BIRDS PERDITION (2006), INDEPENDENCE HOTEL (
8	LISA ANDERSON	MOTHER OLEANDER (2007), COLOR PHILADELPHIA (
9	JOSE ANDREW	CROOKED FROGMEN (2008), GARDEN ISLAND (2007)
10	IDA ANDREWS	CITIZEN SHREK (2008), VOLCANO TEXAS (2008), A
11	OSCAR AQUINO	AFFAIR PREJUDICE (2008), OPEN AFRICAN (2007)
12	HARRY ARCE	COMA HEAD (2007), FROST HEAD (2007), JASON TR
13	JORDAN ARCHULETA	AFFAIR PREJUDICE (2008),LORD ARIZONA (2008)
14	MELANIE ARMSTRONG	DOLLS RAGE (2006), TOURIST PELICAN (2000), BO
15	DEVIDICE VONUIU	DALLOOM HOMENADD (2007) DOLLG DAGE (2004) M

Figure 2: Resultat der Abfrage 1.2

	CUSTOMER	INTERESTS #
1	KATHLEEN ADAMS	Children, Classics, Music
2	DIANA ALEXANDER	Classics, Documentary, Drama, Foreign
3	SHIRLEY ALLEN	Animation, Children, Classics, Documentary, Family, Foreign, Sports
4	KENT ARSENAULT	Animation, Drama, Sci-Fi, Sports
5	TYRONE ASHER	Documentary, Sci-Fi
6	REGINA BERRY	Animation, Family, Foreign, Games, Horror
7	CLINTON BUFORD	Children, Family, Foreign
8	LYDIA BURKE	Children, Documentary, Games
9	ADRIAN CLARY	Family, Games, Music
10	GARY COY	Comedy, Games, Music
11	EUGENE CULPEPPER	Comedy
12	STACY CUNNINGHAM	Documentary, Family
13	LEAH CURTIS	Children, Music
14	EMILY DIAZ	Animation, New, Sports
15	CDACE FILIS	Action Children Classics Foreign New Courts

Figure 3: Resultat der Abfrage 1.3



Figure 4: Resultat der Abfrage 2.1

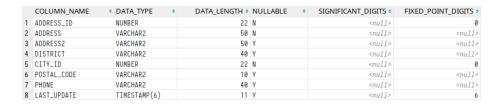


Figure 5: Resultat der Abfrage 3.1

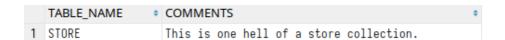


Figure 6: Resultat der Abfrage 3.2

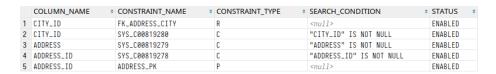


Figure 7: Resultat der Abfrage 3.3



Figure 8: Resultat der Abfrage 4.1

Theorie

4.2 Autokennzeichen

Autokennzeichen sind künstliche Schlüssel, die sich aus verschiedenen Informationen zusammensetzen, wie zum Beispiel dem Sitz der Behörde, bei der ein besimmtes Auto angemeldet ist. Es eignet sich nicht als Primärschlüssel für Kunden, da man mehrere Autos auf seinen Namen anmelden kann und deswegen Kunden der Autostatt zwei mal aufgeführt wären. Weiters ist das Kennzeichen ungeeignet als Primärschlüssel für Auto-Relationen, da bei einem Umzug das Auto (Modell, Zustand etc.) gleich bleibt, sich die Kenntafel aber ändert.

4.3 ISBN

Die ISBN ist eine Kombination von Nummern, welche Informationen über Titel, Verlag etc. geben, sie ist demnach ein zusammengesetzter / künstlicher Schlüssel. Sie identifiziert ein Buch und kein Exemplar eines Buches und ist deshalb ungeeignet als (einziger) Primärschlüssel für Buchexemplare, da eine Bibliothek ohne weiteres mehrere Exemplare eines Buches im Bestand haben kann. Wird der ISBN aber als Schlüssel für Bücher (nicht Exemplare) verwendet, ist das kein Problem - schließlich ist das der Sinn einer ISBN.