

Abgabetermin: 07.11.2018, 13:30 Uhr

<input type="checkbox"/> DES31UE Niklas	Name	Niklas Vest	Aufwand in h	5
<input type="checkbox"/> DES32UE Niklas				
<input checked="" type="checkbox"/> DES33UE Traxler	Punkte		Kurzzeichen Tutor	

In dieser Übung sollen materialisierte Sichten und Abfragen mit dem LISTAGG-Operator anhand der Sakila Datenbank vertieft werden. Weiters werden Data Dictionary und künstliche Schlüssel anhand theoretischer und praktischer Beispiele erarbeitet.

**1. LISTAGG – Sakila****(8 Punkte - 2 + 2 + 4)**

- Geben Sie für alle Filme das Erscheinungsjahr (,year‘) und den Filmtitel (,film‘) aus, sortiert nach Jahr und Titel. Zusätzlich soll für jeden Film eine Liste aller Schauspieler (,actors‘), die in dem Film mitspielen, ausgegeben werden (siehe Abbildung). In der Actors-Liste sollen die Namen nach Nachname und Vorname sortiert sein. Die Namen sollen so ausgegeben werden, dass jeweils der erste Buchstabe des Vornamens, ‘. ‘, und der Nachname angezeigt werden. Die einzelnen Schauspieler sind durch Komma ‘,’ voneinander zu trennen.

	RELEASE_YEAR	Film	actors
1	1983	BORN SPINAL	M. ALLEN, R. JOHANSSON, K. PALTROW, K. PALTROW, R. REYN
2	1983	BOWFINGER GABLES	K. BERRY, C. HUNT, M. MCCONAUGHEY, W. WILSON, M. ZELLWE
3	1983	BUNCH MINDS	K. BERRY, C. BRIDGES, L. BULLOCK, J. CAGE, E. GOODING,
4	1983	CHITTY LOCK	V. BOLGER, S. DAVIS, N. DEGENERES, R. DUKAKIS, A. GARLA
5	1983	CIDER DESIRE	J. CHASE, F. DAY-LEWIS, J. DEGENERES, J. MCQUEEN, P. PI
6	1983	CLOSER BANG	J. DEGENERES, J. FAWCETT, E. GUINNESS, G. MALDEN, K. PAI
7	1983	DIVIDE MONSTER	C. AKROYD, H. BERRY, A. DREYFUSS, S. KILMER, G. MCKELLE
8	1983	DRAGON SQUAD	A. CRONYN, S. DAVIS, S. DEPP, E. GUINNESS, G. HOPKINS, J
9	1983	DRUMS DYNAMITE	V. BERGEN, J. CRUISE, L. DEE, M. HOPKINS, W. HURT, J. S

- Geben Sie für zu jedem Kunden (Vorname Nachname) eine Liste der Filme aus, die sich der Kunde innerhalb der letzten vier Jahre ausgeborgt hat. Zusätzlich zum Titel des Films geben Sie das Erscheinungsjahr in Klammer an. Sortieren Sie die Film-Liste so, dass die jüngsten Filme zuerst aufscheinen.

CUSTOMER	FILMS
1 RAFAEL ABNEY	CHOCOLAT HARRY (2006), CONEHEADS SMOOCHY (2005), GOODFELLAS SALUTE (2002), POCUS PULP (1995)
2 NATHANIEL ADAM	TIGHTS DAWN (2006), GATHERING CALENDAR (2005), ORIENT CLOSER (2005), ROCKY WAR (2005), HANGI
3 KATHLEEN ADAMS	SPY MILE (2008), SWARM GOLD (2003), SUNDANCE INVASION (2002), ALONE TRIP (1998), INSIDER ARI
4 DIANA ALEXANDER	REBEL AIRPORT (2008), SHOW LORD (2006), JASON TRAP (2003), GENTLEMEN STAGE (2001), ROXANNE R
5 GORDON ALLARD	SILVERADO GOLDFINGER (2008), BINGO TALENTED (2006), DETAILS PACKER (2005), ALADDIN CALENDAR

- Der Store in Linz möchte die Interessen seiner Kunden näher bestimmen. Besonders interessant sind die Filmkategorien, die jeden einzelnen Kunden interessieren. Geben Sie Vor- und Nachname des Kunden als „customer“ und eine Liste der Kategorien aller Filme, die sich der jeweilige Kunde ausgeborgt hat, als „interests“ aus. Dabei sollen nur jene Kategorien aufgelistet werden, aus denen sich der Kunde mindestens 3 Filme ausgeborgt hat. Die Kategorien sollen durch Komma getrennt werden. Beschränken Sie sich auf Kunden, die im Store in der Stadt Linz registriert sind. Achten Sie weiters darauf, dass Kategorien nicht mehrfach in der Liste erscheinen. Dies können Sie unter anderem durch eine entsprechende Vorverarbeitung der Datenmenge, die aggregiert wird, erreichen, z.B. durch die Verwendung von WITH. Recherchieren Sie bei Bedarf.

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, Documentarv, Games

Anmerkung: WITH bringt gegenüber einer VIEW Performance-Vorteile: WITH wird zu Beginn ausgewertet und die Ergebnis-Menge dann an den Stellen in der Abfrage eingesetzt (= 1 Auswertung, mehrfache Verwendung), bei einer VIEW wird jedes Mal in der Abfrage, in der sie eingesetzt wird, das Statement neu in der Datenbank ausgeführt.

## 2. Materialisierte Sichten (Sakila-Datenbank)

(6 Punkte – 3+1+2)

- Formulieren Sie eine Anfrage, die den Umsatz der beiden Stores (store\_id = 1 bzw. 2) pro Filmkategorie vergleicht und geben Sie auch das Verhältnis der Umsätze (pro Kategorie) aus. „Speichern“ Sie diese Abfrage als **virtuelle Sicht** UE05\_02a“.

Tipp: Erstellen Sie hierfür zuerst einen Subquery-Block „revenues“, der Ihnen für jeden Store (store\_id) den Umsatz pro Filmkategorie (Name der Filmkategorie) berechnet.

- Erzeugen Sie aus der virtuellen Sicht UE05\_02a eine manuell zu aktualisierende materialisierte Sicht UE05\_02b mit kompletter Neuberechnung (Re-Materialisierung).
- Verändern Sie den Aktualisierungszeitpunkt der erstellten materialisierten Sicht UE05\_02b in der Weise, dass sie automatisch jeden Tag um 23:30 aktualisiert wird. Speichern Sie die geänderte materialisierte Sicht unter dem Namen UE05\_02c. Löschen Sie die materialisierte Sicht UE05\_02c anschließend wieder.

## 3. Data Dictionary

(4 Punkte – 1 + 1 + 2)

Das Data Dictionary von Oracle speichert alle Informationen, die zur Verwaltung der Objekte (z.B. Tabellen, Sichten, Indizes, Prozeduren, Funktionen, Trigger) in der Datenbank benötigt werden. Obwohl dies üblicherweise in den Zuständigkeitsbereich des Datenbankadministrators fällt, stellt das Data Dictionary auch für Entwickler und Datenbankanwender eine wertvolle Informationsquelle dar. Sie lernen in den folgenden Aufgaben ausgewählte Bereiche des Data Dictionary aus der Perspektive des Datenbankanwenders kennen.

- Erstellen Sie ein Skript für eine angegebene Tabelle, das die Spaltennamen, die Datentypen und die Länge der Datentypen sowie eine Information darüber liefert, ob Nullwerte zulässig sind. Fordern Sie den Benutzer auf, den Tabellennamen einzugeben (&-Operator). Weisen Sie den Spalten DATA\_PRECISION und DATA\_SCALE geeignete Aliasnamen zu.
- Fügen Sie der Tabelle STORE einen Kommentar (SQL: COMMENT ON <Tablename> IS '<Comment>') hinzu. Fragen Sie anschließend die View USER\_TAB\_COMMENTS ab, um zu prüfen, ob der Kommentar hinzugefügt wurde.
- Erstellen Sie ein Skript, das den Spaltennamen (COLUMN\_NAME), den Constraint-Namen (CONSTRAINT\_NAME), den Constraint-Typ (CONSTRAINT\_TYPE), das Suchkriterium (SEARCH\_CONDITION) und den Status (STATUS) für eine angegebene Tabelle liefert. Sie müssen die Tabellen USER\_CONSTRAINTS und USER\_CONS\_COLUMNS verknüpfen, um alle diese Informationen zu erhalten. Fordern Sie den Benutzer auf, den Tabellennamen einzugeben.

#### 4. Künstliche Schlüssel

(6 Punkte – 3 + 1,5 + 1,5)

Bewerten Sie folgende Fallbeschreibungen und führen Sie die gestellten Aufgaben durch. Recherchieren Sie wenn notwendig den Sachverhalt und erörtern Sie die Fragestellungen ausführlich.

1. Für das AMA Gütesiegel werden in Oberösterreich stichprobenartig mehrere Kontrollen durchgeführt. In Oberösterreich gibt es dafür vier Teams (diese decken sich mit den Vierteln des Bundeslandes). Diese Teams sind unabhängig voneinander unterwegs und nehmen Proben bei den jeweiligen Betrieben (z.B. Milchproben). Jede Probe soll mit einer landesweit eindeutigen ID (Primärschlüssel) in der gemeinsam verwalteten Datenbank gespeichert werden. Wie können Sie dies sicherstellen? Legen Sie dafür eine Tabelle „Probe“ an (ID, Zeitpunkt der Probennahme, Art der Probe, Kommentar) und die notwendigen Sequenzen (erstellen Sie auch die dazugehörigen DROP Befehle). Gehen Sie bei der Vergabe der IDs so vor, dass Sie zukünftig sofort ablesen können, welches Team die Probe genommen hat. Zeigen Sie beispielhaft, wie die vier Teams Datensätze einfügen können (z.B. Team 1 entnimmt heute Milch, Team 2 entnimmt eine Probe aus Eiern usw.). Fragen Sie anschließend die Daten der Proben-Tabelle ab.
2. Eine Autowerkstatt verwaltet die ihr bekannten Autos (Reparaturen, Service) anhand des Kennzeichens, dieses ist für die Fahrzeuge eindeutig. Bewerten Sie die Information eines Kennzeichens: Ist dies ein künstlicher Schlüssel? Eignet sich ein Kennzeichen als Primärschlüssel? Welche Probleme können auftreten. Welche Verbesserungen hinsichtlich der Schlüsselwahl würden Sie der Autowerkstatt vorschlagen? Denken Sie bei Ihren Antworten auch an Umzug, Besitzer-/Fahrzeugwechsel, Wunsch- und Wechselkennzeichen!
3. Ist die ISBN ein künstlicher Schlüssel? Ist die Anwendung der ISBN als Primärschlüssel in einer Bücherei zulässig/ratsam, die damit die Bücher eindeutig zuordnen kann?

# Ausarbeitung 05

Niklas Vest

November 6, 2018

## Abfragen

```
1  -- 1.1
2  SELECT release_year, title,
3         LISTAGG(SUBSTR(first_name, 0, 1) || '. ' || last_name, ', ')
4         WITHIN GROUP (ORDER BY last_name) AS actors
5  FROM film
6       INNER JOIN film_actor USING (film_id)
7       INNER JOIN actor USING (actor_id)
8  GROUP BY film_id, title, release_year
9  ORDER BY release_year ASC, title ASC;
10
11 -- 1.2
12 SELECT first_name || ' ' || last_name AS customer,
13        LISTAGG(title || ' (' || to_char(release_year) || ')', ', ')
14        WITHIN GROUP (ORDER BY release_year DESC) AS films
15 FROM rental
16      INNER JOIN customer USING (customer_id)
17      INNER JOIN inventory USING (inventory_id)
18      INNER JOIN film USING (film_id)
19 WHERE (
20        (SELECT sysdate
21         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,
29          COUNT(film_id) AS films_of_category
30   FROM customer c
31        INNER JOIN store s ON c.store_id = s.store_id
32        INNER JOIN address a ON s.address_id = a.address_id
33        INNER JOIN city USING (city_id)
34        INNER JOIN rental USING (customer_id)
35        INNER JOIN inventory USING (inventory_id)
36        INNER JOIN film USING (film_id)
37        INNER JOIN film_category USING (film_id)
38   WHERE city = 'Linz'
39   GROUP BY customer_id, category_id
40   HAVING COUNT(film_id) >= 3)
```

```

41 SELECT first_name || ' ' || last_name AS customer,
42        LISTAGG(name, ', ')
43        WITHIN GROUP (ORDER BY name ASC) AS interests
44 FROM at_least_three
45     INNER JOIN customer USING (customer_id)
46     INNER JOIN category USING (category_id)
47 GROUP BY customer_id, first_name, last_name
48 ORDER BY last_name ASC;
49
50 -- 2.1
51 CREATE OR REPLACE VIEW ue5_02a AS
52   WITH revenues AS
53     (SELECT store_id,
54            category_id,
55            SUM(amount) AS revenue
56     FROM payment
57          INNER JOIN rental USING (rental_id)
58          INNER JOIN inventory USING (inventory_id)
59          INNER JOIN store USING (store_id)
60          INNER JOIN film_category USING (film_id)
61     WHERE store_id IN (1, 2)
62     GROUP BY store_id, category_id
63     ORDER BY category_id ASC)
64   SELECT category_id,
65          store1_revenue,
66          store2_revenue,
67          ROUND(store1_revenue / store2_revenue, 2) AS s1_relative_to_s2
68   FROM
69     (SELECT *
70      FROM revenues
71       PIVOT
72         -- it's only one row anyway
73         -- so AVG should be fine
74         (AVG(revenue) AS revenue
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
87   (SELECT *
88    FROM ue5_02a);
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 -- 3.1
105 SELECT column_name,
106        data_type,
107        data_length,
108        nullable,
109        data_precision AS significant_digits,
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,
125        cons.search_condition,
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,
138   "comment" VARCHAR2(255),
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
150 INTO probe (id, timestamp, type)
151 VALUES (probe_seq.nextval || '_team1', SYSDATE, 'Milch');
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 HAIRTIME	I. ARROYD, J. DEGENERES, M. 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	DEATDICE ADMOND	DAIIONM HOMEMADD (2007) DOLLS RAGE (2006) 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	GRACE FLITS	Action, Children, Classics, Foreign, New, Sports

Figure 3: Resultat der Abfrage 1.3



	category_id	store1_revenue	store2_revenue	s1_relative_to_s2
1	1	1161.28	1027.74	1.13
2	2	1264.12	1670.32	0.76
3	3	1076.2	1128.88	0.95
4	4	852.94	1091.73	0.78
5	5	1301.89	1105.1	1.18
6	6	1441.73	1580.36	0.91
7	7	856.05	941.4	0.91
8	8	1021	1111.42	0.92
9	9	1339.4	1106.01	1.21
10	10	1182.9	1154.64	1.02
11	11	1224.96	1097.72	1.12
12	12	1045.16	839.85	1.24
13	13	1109.69	871.42	1.27
14	14	1437.74	1710.23	0.84
15	15	1255.83	1459.38	0.86
16	16	1365.7	936.82	1.46

Figure 4: Resultat der Abfrage 2.1

	COLUMN_NAME	DATA_TYPE	DATA_LENGTH	NULLABLE	SIGNIFICANT_DIGITS	FIXED_POINT_DIGITS
1	ADDRESS_ID	NUMBER	22	N	<null>	0
2	ADDRESS	VARCHAR2	50	N	<null>	<null>
3	ADDRESS2	VARCHAR2	50	Y	<null>	<null>
4	DISTRICT	VARCHAR2	40	Y	<null>	<null>
5	CITY_ID	NUMBER	22	N	<null>	0
6	POSTAL_CODE	VARCHAR2	10	Y	<null>	<null>
7	PHONE	VARCHAR2	40	Y	<null>	<null>
8	LAST_UPDATE	TIMESTAMP(6)	11	Y	<null>	6

Figure 5: Resultat der Abfrage 3.1

	TABLE_NAME	COMMENTS
1	STORE	This is one hell of a store collection.

Figure 6: Resultat der Abfrage 3.2

	COLUMN_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION	STATUS
1	CITY_ID	FK_ADDRESS_CITY	R	<null>	ENABLED
2	CITY_ID	SYS_C00819280	C	"CITY_ID" IS NOT NULL	ENABLED
3	ADDRESS	SYS_C00819279	C	"ADDRESS" IS NOT NULL	ENABLED
4	ADDRESS_ID	SYS_C00819278	C	"ADDRESS_ID" IS NOT NULL	ENABLED
5	ADDRESS_ID	ADDRESS_PK	P	<null>	ENABLED

Figure 7: Resultat der Abfrage 3.3

	id	timestamp	type	"comment"
1	0_team1	2018-11-06 11:10:15	Milch	<null>
2	1_team2	2018-11-06 11:10:16	Eier	<null>

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 bestimmtes 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.