

Abgabetermin: 07.11.2018, 13:30 Uhr

<input checked="" type="checkbox"/> DES31UE Niklas	Name <u>Papesh Konstantin</u>	Aufwand in h <u>6</u>
<input type="checkbox"/> DES32UE Niklas		
<input 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
10	1983	DRAGON SQUAD	C. AKROYD, H. BERRY, A. DREYFUSS, S. KILMER, G. MCKELLE

- 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?

1.1

```
SELECT DISTINCT release_year, title, LISTAGG(SUBSTR(first_name, 1, 1) || '. ' || last_name, ', ')
  WITHIN GROUP (ORDER BY last_name, first_name, a3.actor_id)
    OVER (PARTITION BY title) AS "actors"
FROM film
  INNER JOIN film_actor a2 ON film.film_id = a2.film_id
  INNER JOIN actor a3 ON a2.actor_id = a3.actor_id
ORDER BY release_year, title;
```

RELEASE_YEAR	TITLE	actors
1983	BORN SPINAL	M. ALLEN, R. JOHANSSON, K. PALTROW, K. PALT...
1983	BOWFINGER GABLES	K. BERRY, C. HUNT, M. MCCONAUGHEY, W. WILSO...
1983	BUNCH MINDS	K. BERRY, C. BRIDGES, L. BULLOCK, J. CAGE, ...
1983	CHITTY LOCK	V. BOLGER, S. DAVIS, N. DEGENERES, R. DUKAK...
1983	CIDER DESIRE	J. CHASE, F. DAY-LEWIS, J. DEGENERES, J. MC...
1983	CLOSER BANG	J. DEGENERES, J. FAWCETT, E. GUINNESS, G. MA...
1983	DIVIDE MONSTER	C. AKROYD, H. BERRY, A. DREYFUSS, S. KILMER...
1983	DRAGON SQUAD	A. CRONYN, S. DAVIS, S. DEPP, E. GUINNESS, G...
1983	DRUMS DYNAMITE	V. BERGEN, J. CRUISE, L. DEE, M. HOPKINS, W...
1983	INNOCENT USUAL	C. AKROYD, J. PITT, D. TORN

997 Zeilen

1.2

```
SELECT DISTINCT first_name || ' ' || last_name AS customer,
  LISTAGG(title || '(' || release_year || ')', ', ')
    WITHIN GROUP (ORDER BY release_year DESC)
    OVER (PARTITION BY customer.customer_id) AS films
FROM customer
  INNER JOIN rental r ON customer.customer_id = r.customer_id
  INNER JOIN inventory i ON r.inventory_id = i.inventory_id
  INNER JOIN film f ON i.film_id = f.film_id
WHERE rental_date > ADD_MONTHS(CURRENT_DATE, -(12 * 4));
```

CUSTOMER	FILMS
PATRICIA JOHNSON	CHICKEN HELLFIGHTERS(2008), TEEN APOLLO(200...
MARIA MILLER	WAIT CIDER(2008), CLUELESS BUCKET(2008), NO...
MARGARET MOORE	MADNESS ATTACKS(2008), COLOR PHILADELPHIA(2...
KAREN JACKSON	GARDEN ISLAND(2007), CHANCE RESURRECTION(20...
SANDRA MARTIN	TALENTED HOMICIDE(2006), TOURIST PELICAN(20...
CAROL GARCIA	WAIT CIDER(2008), HYDE DOCTOR(2005), YENTL ...
RUTH MARTINEZ	HEAVYWEIGHTS BEAST(2003), ARACHNOPHOBIA ROL...
AMANDA CARTER	SUN CONFESSIONS(2003), CURTAIN VIDEOTAPE(20...
ALICE STEWART	OPEN AFRICAN(2007), ORANGE GRAPES(2005), SP...
CHRISTINA RAMIREZ	VACATION BOONDOCK(2001), MASSACRE USUAL(200...
BEVERLY BROOKS	MALKOVICH PET(2007), SECRETS PARADISE(2007)...
LORI WOOD	CLUELESS BUCKET(2008), SILVERADO GOLDFINGER...

599 Zeilen

1.3

```

WITH three_or_more AS (SELECT c.customer_id, name, COUNT(f.film_id)
                        FROM customer c
                        INNER JOIN rental r ON c.customer_id = r.customer_id
                        INNER JOIN inventory i ON r.inventory_id = i.inventory_id
                        INNER JOIN film f ON i.film_id = f.film_id
                        INNER JOIN film_category ON f.film_id = film_category.film_id
                        INNER JOIN category ON film_category.category_id =
category.category_id
                        GROUP BY c.customer_id, name
                        HAVING count(f.film_id) >= 3)
SELECT DISTINCT first_name || ' ' || last_name AS customer, LISTAGG(name, ', ')
    WITHIN GROUP (ORDER BY name)
    OVER (PARTITION BY customer.customer_id) AS interests
FROM customer
    INNER JOIN three_or_more ON three_or_more.customer_id = customer.customer_id
    INNER JOIN address a2 ON customer.address_id = a2.address_id
    INNER JOIN city c2 ON a2.city_id = c2.city_id
WHERE city = 'Linz';

```

CUSTOMER	INTERESTS
JILL HAWKINS	Action, Animation, Classics

2.1

```

CREATE VIEW "UE05_02a" AS
SELECT category, store1, store2, ROUND(1 + (store2 - store1) / store1, 3) AS "STORE1:STORE2"
FROM (SELECT *
      FROM (SELECT store_id, name AS category, SUM(amount) AS revenue
            FROM inventory i
            INNER JOIN film f ON i.film_id = f.film_id
            INNER JOIN film_category c2 ON f.film_id = c2.film_id
            INNER JOIN category c3 ON c2.category_id = c3.category_id
            INNER JOIN rental r ON i.inventory_id = r.inventory_id
            INNER JOIN payment p ON r.rental_id = p.rental_id
            GROUP BY store_id, name
            HAVING store_id = 1
            OR store_id = 2)
      PIVOT
      (
        SUM(revenue)
        FOR (store_id)
        IN (1 AS store1,
            2 AS store2)
      )
      ORDER BY category);

-- Show view
SELECT *
FROM "UE05_02a";

```

category	store1	store2	"STORE1:STORE2"
Action	1161.28	1027.74	0.885
Animation	1264.12	1670.32	1.321
Children	1076.2	1128.88	1.049
Classics	852.94	1091.73	1.28
Comedy	1301.89	1105.1	0.849
Documentary	1441.73	1580.36	1.096
Drama	856.05	941.4	1.1
Family	1021	1111.42	1.089
Foreign	1339.4	1106.01	0.826
Games	1182.9	1154.64	0.976

16 Zeilen

2.2

```
CREATE MATERIALIZED VIEW "UE05_02b"
REFRESH COMPLETE ON DEMAND
AS
  SELECT *
  FROM "UE05_02a";
```

2.3

```
ALTER MATERIALIZED VIEW "UE05_02b"
REFRESH START WITH trunc(SYSDATE) + 23.5 / 24
NEXT trunc(sysdate) + 1 / 24 * 23.5;

DROP VIEW "UE05_02a";
DROP MATERIALIZED VIEW "UE05_02b";
```

3.1

```
SELECT column_name, data_type, data_length, nullable, data_precision AS decimal_points,
data_scale AS digits_right
FROM all_tab_columns
WHERE table_name = '&tablename';
```

3.2

```
COMMENT ON TABLE store
IS 'Test';

SELECT *
FROM user_tab_comments;
```

STORE	TABLE	Test	0
-------	-------	------	---

3.3

```
SELECT DISTINCT column_name, constr.constraint_name, constr.constraint_type,
constr.search_condition_vc, constr.status
FROM user_constraints constr
      JOIN user_cons_columns ucc USING (table_name)
WHERE '' || ucc.column_name || '' = REGEXP_SUBSTR(search_condition_vc, '"(.*)"');
```

COLUMN_NAME	CONSTRAINT_NAME	CONSTRAINT_TYPE	SEARCH_CONDITION_VC	STATUS
CITY_ID	SYS_C00797547	C	"CITY_ID" IS NOT NULL	ENABLED
COUNTRY_ID	SYS_C00797541	C	"COUNTRY_ID" IS NOT NULL	ENABLED
START_DATE	JHIST_START_DATE_NN	C	"START_DATE" IS NOT NULL	ENABLED
FILM_ID	SYS_C00809804	C	"FILM_ID" IS NOT NULL	ENABLED
STAFF_ID	SYS_C00797617	C	"STAFF_ID" IS NOT NULL	ENABLED
FILM_ID	SYS_C00797607	C	"FILM_ID" IS NOT NULL	ENABLED
INVENTORY_ID	SYS_C00797606	C	"INVENTORY_ID" IS NOT NULL	ENABLED
FILM_ID	SYS_C00797595	C	"FILM_ID" IS NOT NULL	ENABLED
TITLE	SYS_C00797585	C	"TITLE" IS NOT NULL	ENABLED
LAST_UPDATE	SYS_C00797566	C	"LAST_UPDATE" IS NOT NULL	ENABLED

86 Zeilen

4.1

```
CREATE SEQUENCE id_seq
START WITH 1
INCREMENT BY 1
NOCACHE
NOCYCLE;
```

```
CREATE OR REPLACE FUNCTION seq(a INT)
RETURN VARCHAR2 IS
BEGIN
  RETURN a || '_' || id_seq.nextval;
END;
```

```
CREATE TABLE probe
(
  probe_id VARCHAR2(30) PRIMARY KEY,
  zeit      TIMESTAMP    NOT NULL,
  art       VARCHAR2(30) NOT NULL,
  kommentar VARCHAR2(30)
);
```

```
INSERT INTO probe (probe_id, zeit, art)
VALUES (seq(1), SYSDATE, 'Milch');
```

```
INSERT INTO probe (probe_id, zeit, art, kommentar)
VALUES (seq(2), SYSDATE, 'Eier', 'Sehr duenne Schale');
```

```
INSERT INTO probe (probe_id, zeit, art)
VALUES (seq(3), SYSDATE, 'Schafskaese');
```

```
INSERT INTO probe (probe_id, zeit, art)
VALUES (seq(3), SYSDATE, 'Butter');
```

```
SELECT *
FROM probe;
```

```
COMMIT;
```

```
DROP TABLE probe;
```

```
DROP FUNCTION seq;
```

```
DROP SEQUENCE id_seq;
```

probe_id	zeit	art	kommentar
1_1	2018-11-07 10:21:51.000000	Milch	<null>
2_2	2018-11-07 10:21:53.000000	Eier	Sehr duenne Schale
3_3	2018-11-07 10:21:54.000000	Schafskaese	<null>
3_4	2018-11-07 10:21:56.000000	Butter	<null>

4.2

Beim Autokennzeichen an sich handelt es sich um einen künstlichen Schlüssel.

Da es nicht auf existierende Eigenschaften zurückgreift.

Als alleiniger Primärschlüssel eignet es sich jedoch nicht, da beim Verkauf der frühere Besitzer manchmal die Nummerntafel behalten kann.

dh es sich dann um ein anderes Auto handelt. Außerdem wird bei Umzug in ein anderes Verwaltungsgebiet der Anfangsfolge des Kennzeichens abgeändert.

Daher wäre es sinnvoll, die Fahrgestellnummer als Primärschlüssel zu verwenden, da diese eindeutig an das Fahrzeug gebunden ist.

AUSNAHME: In England ist die Nummerntafel fest an das Auto gebunden. Hier wäre es zulässig, diese als Primärschlüssel zu verwenden.

4.3

Ja, die ISBN ist eine Kombination aus verschiedenen serialisierten Daten wie Publisher und Titel.

Jedoch kann eine Bibliothek ein Buch öfter führen, wodurch es zu Problemen kommen könnte.

Daher ist eine zweite ID für die Bibliothek sehr ratsam.