

- DES3UEG1 (Niklas)  
 DES3UEG2 (Niklas)  
 DES3UEG3 (Teufl)

Name Egor Podverbni Aufwand in h 16  
Punkte \_\_\_\_\_ Kurzzeichen Tutor\*in \_\_\_\_\_

Ziel dieser Übung ist die Modellierung und Erstellung einer kleinen **Data Warehouse Anwendung** und die Durchführung von Abfragen darauf.

Arbeiten Sie die Übung in **2er Gruppen** aus und geben Sie Ihre\*n Partner\*in an. Eine Abgabe für beide gemeinsam ist ausreichend.

### Beschreibung Sachverhalt

Ihr Auftraggeber ist der Besitzer der Film-Verleihkette „**Sakila**“, die in mehreren Städten Stores zum Verleih von Filmen betreibt. Die Grundlage für den laufenden Betrieb bildet eine Datenbank, die alle Filme enthält, außerdem werden Kunden und Personal erfasst, sowie die Entlehnungen und Rückgaben (inkl. Bezahlung).

Ihr Auftraggeber möchte eine stabile Datenbasis erstellen, die es ihm ermöglicht, viele **analytische Abfragen** einfach durchzuführen, um den laufenden Betrieb zu beobachten und maßgebliche Entscheidungen zu treffen. Es soll möglich werden, **Kennzahlen des laufenden Geschäftsbetriebs** zu erfassen. Es ist ausreichend, wenn der Datenstand **monatsweise aktualisiert** wird, eine täglich aktuelle Auskunft ist nicht notwendig.

Die Filmkette ist dafür bekannt, auf der ganzen Welt Filme in verschiedenen Sprachen zu verleihen und verschiedene Film-Genres zu besitzen. Für die Analyse der Geschäftsprozesse ist es nicht wichtig, Details zu kennen wie den Namen eines Films oder die mitspielenden SchauspielerInnen. Die Länge des Films ist relevant (unterteilt nach short <= 60 Minuten, medium <= 120 Minuten, long > 120 Minuten) und die Stadt/das Land in dem der Film verliehen wurde. Ihr Auftraggeber ist auch an den Umsatzzahlen interessiert und möchte diese quartals- und monatsweise erfassen (Q1 = Jan, Feb, Mar; Q2 = Apr, Mai, Jun, ...); maßgeblich soll das Datum des Umsatzes sein (dh. Bezahlungseingang).

Die Film-Genres werden zu Analysezwecken innerhalb der Verleih-Kette in vier Klassen eingeteilt: Storyline (Animation, Sci-Fi, Sports), Narrative (Children, Comedy, Documentary, Drama, Family, Foreign, Travel), Mood (Action, Horror, Music) und Others (alle übrigen). Dies ist zwar für den laufenden Betrieb unerheblich, sollte jedoch bei der Auswertung zur Verfügung stehen. Beachten Sie dazu auch die bereits beschriebenen Business-Fälle (Abfragen) ganz unten.

### 1. Modellierung ADAPT

(8 Punkte)

Modellieren Sie den oberhalb beschriebenen Sachverhalt (unter Berücksichtigung der beschriebenen Business-Fälle) in der Modelliersprache ADAPT.

### 2. Modellierung STAR-Schema

(7 Punkte)

Erstellen Sie aus der ADAPT Modellierung, die Sie im vorigen Schritt erstellt haben, ein STAR-Schema. Verwenden Sie dazu ein Werkzeug Ihrer Wahl.

**3. Erstellung STAR-Schema****(11 Punkte)**

Erstellen Sie das modellierte STAR-Schema physisch in der Datenbank. Verwenden Sie dazu

- eine Sequenz (um die Primärschlüssel zu generieren),
- Tabellen für die Dimensionstabellen (Präfix dim) und
- eine Materialisierte Sicht für die Faktentabelle (Präfix fact), wobei das Update jeweils am Ersten des Monats erfolgen soll. Recherchieren Sie bei Bedarf die benötigte Datums-Syntax.

Befüllen Sie Ihre Dimensions-Tabellen, bevor Sie die Materialisierte Sicht erstellen: zB CREATE TABLE dimTable AS, DML-Statements, anonymer Block, gespeicherte Prozedur, EXTRACT (datetime). Rechnen Sie mit ganzen Tagen (d.h. Aufrunden).

**4. Abfragen & Interpretation****(22 Punkte)**

Sie haben mehrere Fragestellungen von Ihrem Auftraggeber erhalten. Erstellen Sie **pro Frage EINE Abfrage**, mit der Sie den Sachverhalt erörtern können. **Zusätzlich** zur gewohnten Formatierung und Abgabe der Ergebnisse Ihrer Statements, beschreiben Sie das jeweilige Ergebnis der gestellten Frage in **textueller Form**.

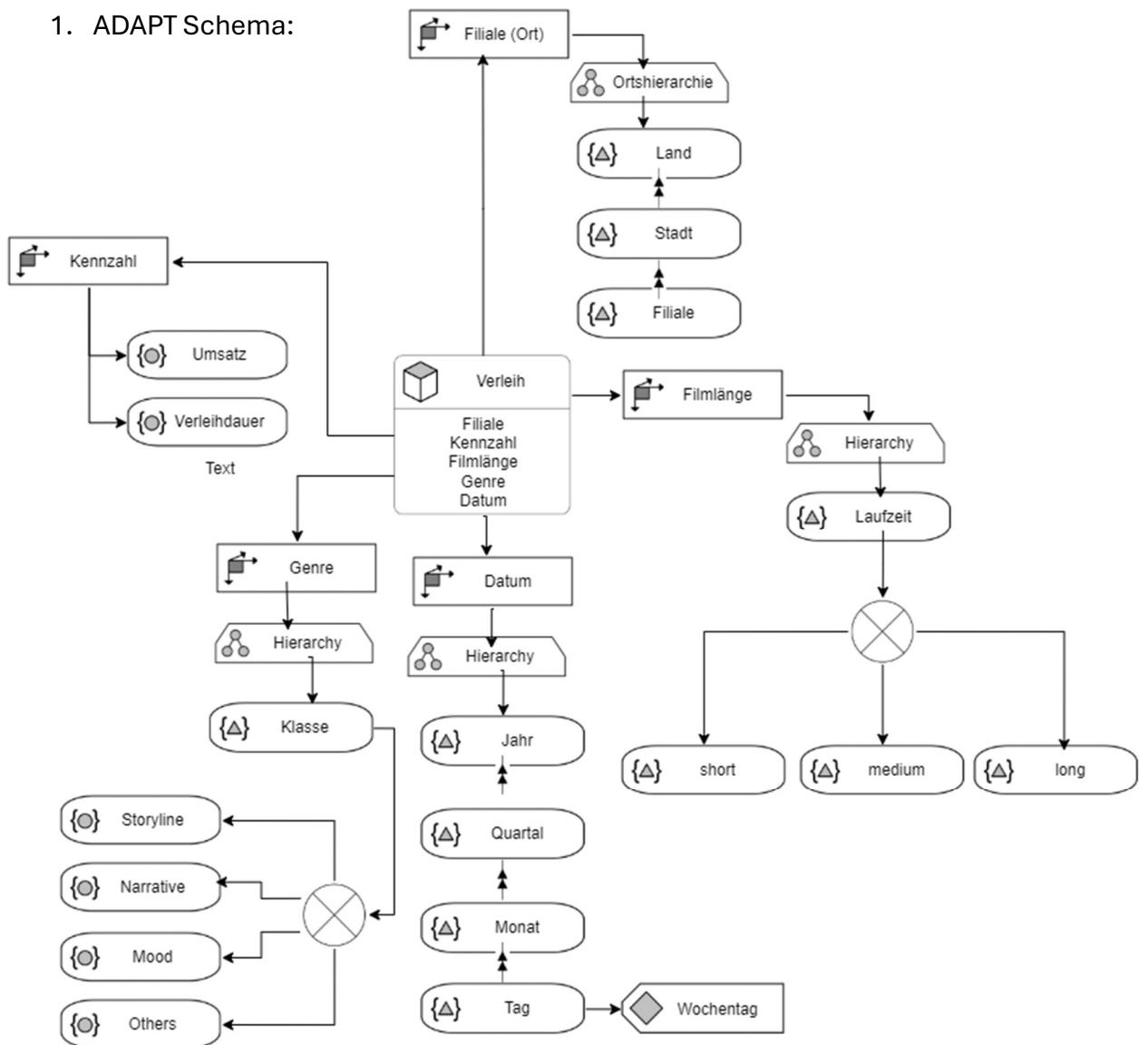
*Hinweise* (je Frage in *kursiv*): Mit den angegebenen Hinweisen können Sie Ihre Lösung überprüfen. Die Hinweise sind dabei „Nebenprodukte“ eines Statements, das zur Analyse herangezogen werden kann.

*Allgemeiner Hinweis: Jahr, Monat und Quartal, Wochentag sollen nicht aus dem Datumsfeld neu ermittelt werden, wenn diese Werte bereits vorberechnet vorliegen.*

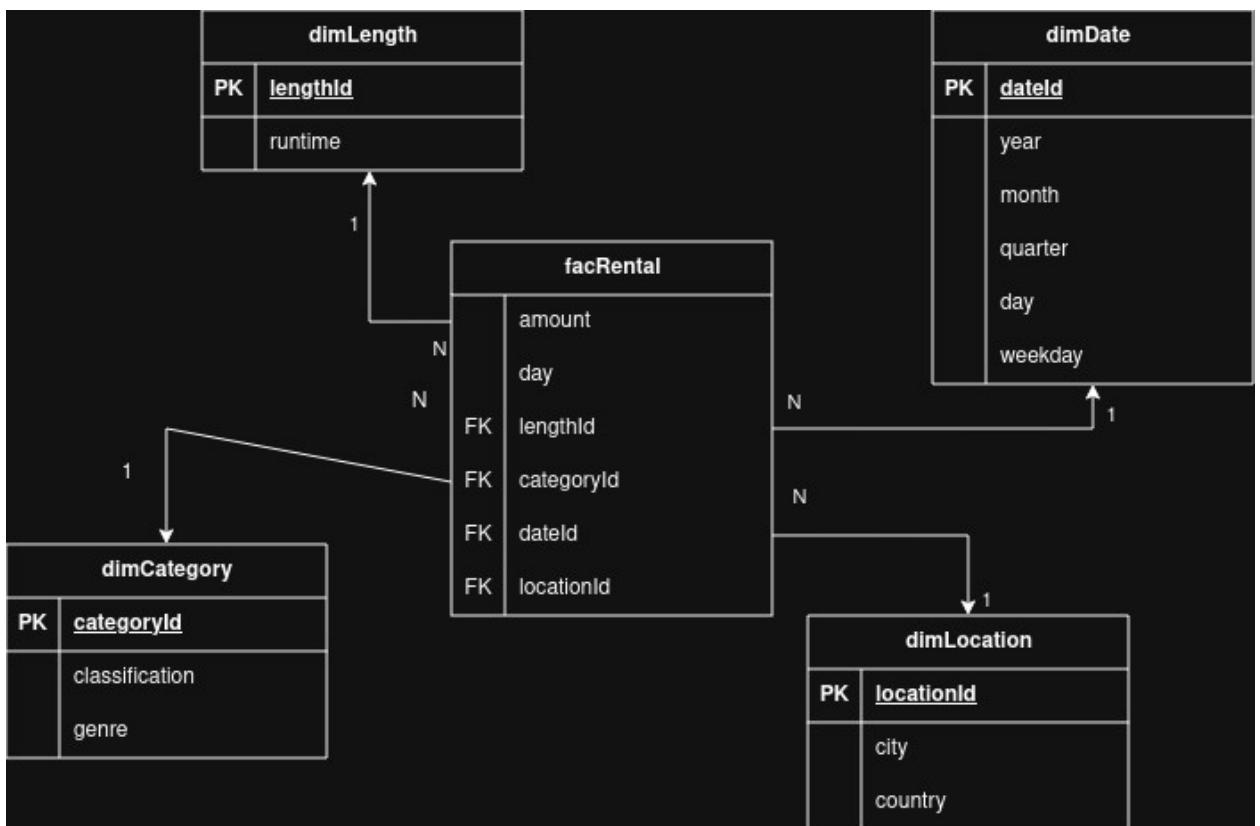
1. Stellen Sie fest, wie häufig unterschiedliche Film-Laufzeiten (short, medium, long) ausgeliehen werden (*3 Zeilen*). Welche Laufzeit wird am seltensten ausgeliehen?
2. Geben Sie den Namen jener Stadt aus, in der insgesamt die meisten Einnahmen erzielt wurden.
3. Ermitteln Sie welche Genres in welchen Ländern durchschnittlich den meisten Umsatz einbringen. Die Verleihdauer braucht nicht berücksichtigt werden. Welches Genre liegt an Platz 1, d.h. ist am lukrativsten? (*zweitbeste Kategorie ist Travel in Australien.*)
4. Stellen Sie die durchschnittlichen Einnahmen pro Kategorie und Land in einer Kreuzwert-Tabelle (inkl. Gesamt-Durchschnitt) dar. Die Kategorien Storyline, Mood, Narrative und Others stellen die Spalten dar, die Länder werden in den Zeilen angeführt.
5. Ermitteln Sie die Gesamt-Verleihdauer pro Jahr, gibt es einen Zusammenhang? (*2013 ist das schwächste Jahr*)
6. Analysieren Sie, ob in einem bestimmten Quartal (ermittelt über alle Jahre) mehr Umsatz erzielt wurde. Welches Quartal liegt an Platz 1? (*Platz 4 ist Quartal 4.*)
7. Stellen Sie gegenüber, wie lange Filme durchschnittlich ausgeliehen werden, analysieren Sie den Sachverhalt monatsweise und quartalsweise. Können Sie einen Zusammenhang feststellen? In welchem Monat werden Filme im Durchschnitt am längsten ausgeliehen, in welchem Quartal (Angabe Monat/Quartal genügt)? (*Sehr kleiner Unterschied; 17 Zeilen.*)
8. Analysieren Sie anhand der Anzahl an ausgeliehenen Filmen, ob ein Zusammenhang besteht zwischen Spieldauer (Film-Lauffzeit) und Land. Stellen Sie auch die Einnahmen dafür gegenüber. Welche Filmlängen (short, medium, long) würden Sie in welchen Ländern ausmustern (welche erzielen die wenigsten Einnahmen)? (*20 Zeilen, Australien hat – alleine betrachtet – die wenigsten Einnahmen.*)

9. Sie möchten ermitteln, ob es bestimmte Wochentage gibt, an denen häufiger Filme zurückgegeben werden (Tage der Bezahlvorgänge). Unterscheiden sich diese in verschiedenen Ländern? (*Die wenigsten Bezahlvorgänge gibt es an Montagen in Australien.*)
10. Ermitteln Sie für jedes Land die maximale Anzahl an Filmen, die innerhalb von 365 Tagen ausgeliehen wurden. Hierfür berechnen Sie von jedem Bezahlvorgang ausgehend die Anzahl der in den vorhergehenden 365 Tagen ausgeliehenen Filme, selektieren Sie davon das das Maximum je Land. (*4 Zeilen, 1419 für Austria*)
11. Geben Sie die kumulierte Umsatz-Summe aller Bezahlvorgänge für die Stadt „Tsuyama“ aus. Ermitteln Sie dazu zuerst die Gesamtsumme der Bezahlvorgänge jedes Monats und kumulieren Sie diese Anzahl innerhalb des jeweiligen Jahres. (*kumulierte Summe für das Jahr 2014 ist 10441,24*)

## 1. ADAPT Schema:



## 2. STAR Schema:



### 3. Erstellung STAR-Schema:

```
-- Sequenz für künstliche Schlüssel der Dimensionstabellen
CREATE SEQUENCE identifiers;
```

```
-- dimLength
CREATE TABLE dimLength AS
SELECT identifiers.nextVal AS lengthId, length,
CASE
    WHEN length <= 60 THEN 'short'
    WHEN length > 60 AND length <= 120 THEN 'medium'
    ELSE 'long'
    END AS runtime
FROM
(SELECT DISTINCT length FROM film);
```

```
ALTER TABLE dimLength ADD CONSTRAINT dimLengthPK PRIMARY KEY (lengthId);
```

```
SELECT * FROM dimLength;
```

	LENGTHID	LENGTH	RUNTIME	
1	141	58	short	
2	142	173	long	
3	143	166	long	
4	144	64	medium	
5	145	57	short	
6	146	50	short	
7	147	113	medium	
8	148	72	medium	
9	149	161	long	
10	150	105	?	

```
-- dimLocation
```

```
CREATE TABLE dimLocation AS
SELECT identifiers.nextval AS locationId, city, country, address_id
FROM
(
    SELECT city, country, address_id
    FROM store INNER JOIN address USING (address_id)
    INNER JOIN city USING (city_id)
    INNER JOIN country USING (country_id)
);
```

```
ALTER TABLE dimLocation ADD CONSTRAINT dimLocationPK PRIMARY KEY (locationId);
```

```
SELECT * FROM dimLocation;
```

	LOCATIONID	CITY	COUNTRY	ADDRESS_ID
1	281	Woodridge	Australia	2
2	282	Linz	Austria	606
3	283	Vancouver	Canada	441
4	284	Lethbridge	Canada	1
5	285	Tsuyama	Japan	434
6	286	Izumisano	Japan	607

-- dimDate

```

CREATE TABLE dimDate(
    dateId NUMBER PRIMARY KEY,
    ddate DATE, --dient nur zur Verknüpfung mit der Faktentabelle
    year NUMBER NOT NULL,
    month NUMBER NOT NULL,
    quarter NUMBER NOT NULL,
    day NUMBER NOT NULL,
    weekday VARCHAR2(10)
);

DECLARE
    minDate DATE;
    maxDate DATE := TRUNC(SYSDATE);
    currDate DATE;
BEGIN
    SELECT MIN(TRUNC(payment_date)) INTO minDate FROM payment;
    FOR d IN 0 .. (maxDate - minDate) LOOP
        currDate := minDate + d;
        INSERT INTO dimDate (dateId, ddate, year, month, quarter, day,
    weekday)
        VALUES (
            TO_CHAR(currDate, 'YYYYMMDD'),
            currDate,
            EXTRACT(YEAR FROM currDate),
            EXTRACT(MONTH FROM currDate),
            TO_CHAR(currDate, 'Q'),
            TO_CHAR(currDate, 'DD'),
            TO_CHAR(currDate, 'Day')
        );
    END LOOP;
END;

```

```
SELECT * FROM dimDate;
```

	DATEID	DDATE	YEAR	MONTH	QUARTER	DAY	WEEKDAY
1	20060506	2006-05-06	2006	5		2	6 Saturday
2	20060507	2006-05-07	2006	5		2	7 Sunday
3	20060508	2006-05-08	2006	5		2	8 Monday
4	20060509	2006-05-09	2006	5		2	9 Tuesday
5	20060510	2006-05-10	2006	5		2	10 Wednesday
6	20060511	2006-05-11	2006	5		2	11 Thursday
7	20060512	2006-05-12	2006	5		2	12 Friday
8	20060513	2006-05-13	2006	5		2	13 Saturday
9	20060514	2006-05-14	2006	5		2	14 Sunday

```
-- dimCategory
CREATE TABLE dimCategory AS
SELECT
    identifiers.NEXTVAL AS categoryId,
    c.name AS genre,
    CASE
        WHEN c.name IN ('Animation', 'Sci-Fi', 'Sports')
            THEN 'Storyline'
        WHEN c.name IN ('Children', 'Comedy', 'Documentary', 'Drama',
'Family', 'Foreign', 'Travel')
            THEN 'Narrative'
        WHEN c.name IN ('Action', 'Horror', 'Music')
            THEN 'Mood'
        ELSE 'Others'
    END AS classification
FROM (
    SELECT DISTINCT name
    FROM category
) c;
```

```
ALTER TABLE dimCategory
ADD CONSTRAINT dimCategoryPK PRIMARY KEY (categoryId);
```

```
SELECT * FROM dimCategory;
```

	CATEGORYID	GENRE	CLASSIFICATION
1	301	Children	Narrative
2	302	Classics	Others
3	303	Documentary	Narrative
4	304	Family	Narrative
5	305	Foreign	Narrative
6	306	New	Others
7	307	Sports	Storyline
8	308	Animation	Storyline
9	309	Horror	Mood

```
DROP MATERIALIZED VIEW factRental;
```

```
CREATE MATERIALIZED VIEW factRental
REFRESH COMPLETE
NEXT ADD_MONTHS(TRUNC(SYSDATE, 'mm'), 1) + 1/24
AS
SELECT
    p.amount AS amount,
    CEIL(r.return_date - r.rental_date) AS duration,
    dl.lengthId AS lengthId,
    dloc.locationId AS locationId,
    dd.dateId AS dateId,
    dc.categoryId AS categoryId
FROM rental r
JOIN payment p      ON p.rental_id = r.rental_id
JOIN inventory i   ON i.inventory_id = r.inventory_id
```

```

JOIN film f          ON f.film_id      = i.film_id
JOIN film_category fc ON fc.film_id    = f.film_id
JOIN category c     ON c.category_id = fc.category_id
JOIN store s         ON s.store_id    = i.store_id
JOIN dimLength dl   ON dl.length     = f.length
JOIN dimLocation dloc ON dloc.address_id = s.address_id
JOIN dimDate dd     ON dd.ddate      = TRUNC(p.payment_date)
JOIN dimCategory dc  ON dc.genre      = c.name;

```

`SELECT * FROM factRental;`

	AMOUNT	DURATION	LENGTHID	LOCATIONID	DATEID	CATEGORYID	
1	0.27	3	157	285	20150928	307	
2	16.42	7	278	283	20140209	313	
3	8.23	5	176	285	20150706	311	
4	10.43	7	239	281	20140327	309	
5	9.03	7	170	284	20140520	313	
6	18.13	7	152	285	20151004	301	
7	1.17	3	163	283	20150404	311	
8	15.45	5	237	284	20140928	310	
9	9.54	6	224	286	20140925		

```

BEGIN
  dbms_mview.REFRESH('factRental');
end;

```

#### 4. Abfragen

-- 1 Das Ergebnis ist eine Liste der drei Laufzeitklassen (short, medium, long) mit der jeweiligen Anzahl an Rentals.  
-- Die kleinste Anzahl (erste Zeile wegen Sortierung) zeigt, welche Laufzeitklasse am seltensten vorkommt.

```
SELECT
    dl.runtime,
    COUNT(*) AS rentals_cnt
FROM factRental fr
JOIN dimLength dl ON dl.lengthId = fr.lengthId
GROUP BY dl.runtime
ORDER BY rentals_cnt;
```

	RUNTIME	RENTALS_CNT
1	short	1651
2	medium	7092
3	long	7298

-- 2  
-- Das Ergebnis ist genau eine Stadt: jene, die über alle Rentals hinweg die höchste Summe an Einnahmen (SUM(amount)) erzielt hat.

```
SELECT city
FROM (
    SELECT
        dloc.city,
        SUM(fr.amount) AS revenue
    FROM factRental fr
    JOIN dimLocation dloc ON dloc.locationId = fr.locationId
    GROUP BY dloc.city
    ORDER BY revenue DESC
)
FETCH FIRST 1 ROW ONLY;
```

	CITY
1	Tsuyama

-- 3  
-- Das Ergebnis zeigt pro Land das Genre, das im jeweiligen Land den höchsten durchschnittlichen Umsatz pro Zahlung (AVG(amount)) hat, inkl. dieses Durchschnittswerts. Wenn mehrere Genres gleichauf sind, werden alle mit Rang 1 ausgegeben.

```
SELECT country, genre, avg_amount
FROM (
    SELECT
        dloc.country,
        dc.genre,
        AVG(fr.amount) AS avg_amount,
        DENSE_RANK() OVER (
            PARTITION BY dloc.country
            ORDER BY AVG(fr.amount) DESC
        ) AS rnk
```

```

        FROM factRental fr
        JOIN dimLocation dloc ON dloc.locationId = fr.locationId
        JOIN dimCategory dc   ON dc.categoryId    = fr.categoryId
        GROUP BY dloc.country, dc.genre
    )
WHERE rnk = 1
ORDER BY avg_amount DESC;

```

	COUNTRY	GENRE	Avg_Amount
1	Austria	Sci-Fi	9.10208737864077669902912621359223300971
2	Australia	Sci-Fi	9.04883597883597883597883597883597883598
3	Canada	Sci-Fi	8.80973760932944606413994169096209912536
4	Japan	Family	8.44618798955613577023498694516971279373

-- 4  
-- Das Ergebnis ist eine Tabelle mit einer Zeile pro Land und Spalten für die 4 Klassifizierungen (Storyline/Mood/Narrative/Others).  
-- In jeder Zelle steht der durchschnittliche Umsatz für diese Kategorie in diesem Land;  
-- zusätzlich gibt es eine Spalte total\_avg als Gesamt-Durchschnitt über alle Kategorien im Land.

```

SELECT
    dloc.country,
    ROUND(AVG(CASE WHEN dc.classification = 'Storyline' THEN fr.amount
END), 2) AS storyline_avg,
    ROUND(AVG(CASE WHEN dc.classification = 'Mood'      THEN fr.amount
END), 2) AS mood_avg,
    ROUND(AVG(CASE WHEN dc.classification = 'Narrative' THEN fr.amount
END), 2) AS narrative_avg,
    ROUND(AVG(CASE WHEN dc.classification = 'Others'    THEN fr.amount
END), 2) AS others_avg,
    ROUND(AVG(fr.amount), 2) AS total_avg
FROM factRental fr
JOIN dimLocation dloc ON dloc.locationId = fr.locationId
JOIN dimCategory dc   ON dc.categoryId    = fr.categoryId
GROUP BY dloc.country
ORDER BY dloc.country;

```

COUNTRY	STORYLINE_AVG	MOOD_AVG	NARRATIVE_AVG	OTHERS_AVG	TOTAL_AVG
1 Australia	7.52	6.85	7.35	6.42	7.14
2 Austria	7.82	7.53	7.08	6.52	7.19
3 Canada	7.46	6.87	7.48	6.99	7.28
4 Japan	7.12	7.22	7.46	6.5	7.19

-- 5  
-- Das Ergebnis listet pro Jahr die Summe der Verleihdauer (duration) (z. B. in Tagen, je nach Definition deiner duration).  
-- Damit siehst du, ob die Gesamtverleihdauer über die Jahre eher steigt, sinkt oder schwankt.

```

SELECT
    dd.year,
    SUM(fr.duration) AS total_duration_days
FROM factRental fr
JOIN dimDate dd ON dd.dateId = fr.dateId
GROUP BY dd.year

```

```
ORDER BY dd.year;
```

	YEAR	TOTAL_DURATION_DAYS
1	2013	2091
2	2014	38862
3	2015	32891

```
-- 6
```

-- Das Ergebnis listet die Quartale Q1-Q4 mit der jeweiligen Umsatzsumme.  
-- Durch Sortierung ist oben das Quartal mit dem höchsten Umsatz.

```
SELECT
```

```
    dd.quarter,  
    SUM(fr.amount) AS revenue
```

```
FROM factRental fr
```

```
JOIN dimDate dd ON dd.dateId = fr.dateId
```

```
GROUP BY dd.quarter
```

```
ORDER BY revenue DESC;
```

	QUARTER	REVENUE
1	1	31764.09
2	3	29326.91
3	2	28124
4	4	26416.06

```
-- 7
```

-- Das Ergebnis enthält Durchschnittswerte der Verleihdauer getrennt nach Monat und nach Quartal (jeweils innerhalb eines Jahres).

-- Du bekommst dadurch einen Vergleich, ob Rentals z. B. in bestimmten Monaten/Quartalen im Schnitt länger oder kürzer dauern.

-- Filme werden durchschnittlich im 1. Quartal und im Januar am längsten ausgeborgt.

```
SELECT
```

```
    grp,  
    period,  
    ROUND(avg_duration, 2) AS avg_duration_days
```

```
FROM (
```

```
-- 12 Monate
```

```
SELECT
```

```
    'MONTH' AS grp,  
    TO_CHAR(dd.month) AS period,  
    AVG(fr.duration) AS avg_duration
```

```
FROM factRental fr
```

```
JOIN dimDate dd ON dd.dateId = fr.dateId
```

```
GROUP BY dd.month
```

```
UNION ALL
```

```
-- 4 Quartale
```

```
SELECT
```

```
    'QUARTER' AS grp,  
    TO_CHAR(dd.quarter) AS period,  
    AVG(fr.duration) AS avg_duration
```

```

        FROM factRental fr
    JOIN dimDate dd ON dd.dateId = fr.dateId
    GROUP BY dd.quarter

    UNION ALL

    -- 1 Gesamtwert
    SELECT
        'TOTAL' AS grp,
        '-' AS period,
        AVG(fr.duration) AS avg_duration
    FROM factRental fr
)
ORDER BY
    CASE grp WHEN 'MONTH' THEN 1 WHEN 'QUARTER' THEN 2 ELSE 3 END,
    CASE
        WHEN grp = 'MONTH' THEN TO_NUMBER(period)
        WHEN grp = 'QUARTER' THEN TO_NUMBER(period)
        ELSE 0
    END;

```

	GRP	PERIOD	AVG_DURATION_DAYS
1	MONTH	1	5.54
2	MONTH	2	5.14
3	MONTH	3	4.68
4	MONTH	4	4.51
5	MONTH	5	4.21
6	MONTH	6	3.99
7	MONTH	7	4.05
8	MONTH	8	4.35
9	MONTH	9	4.5
10	MONTH	10	4.49
11	MONTH	11	4.77
12	MONTH	12	5.36
13	QUARTER	1	5.11
14	QUARTER	2	4.23
15	QUARTER	3	4.29
16	QUARTER	4	4.84
17	TOTAL	-	4.6