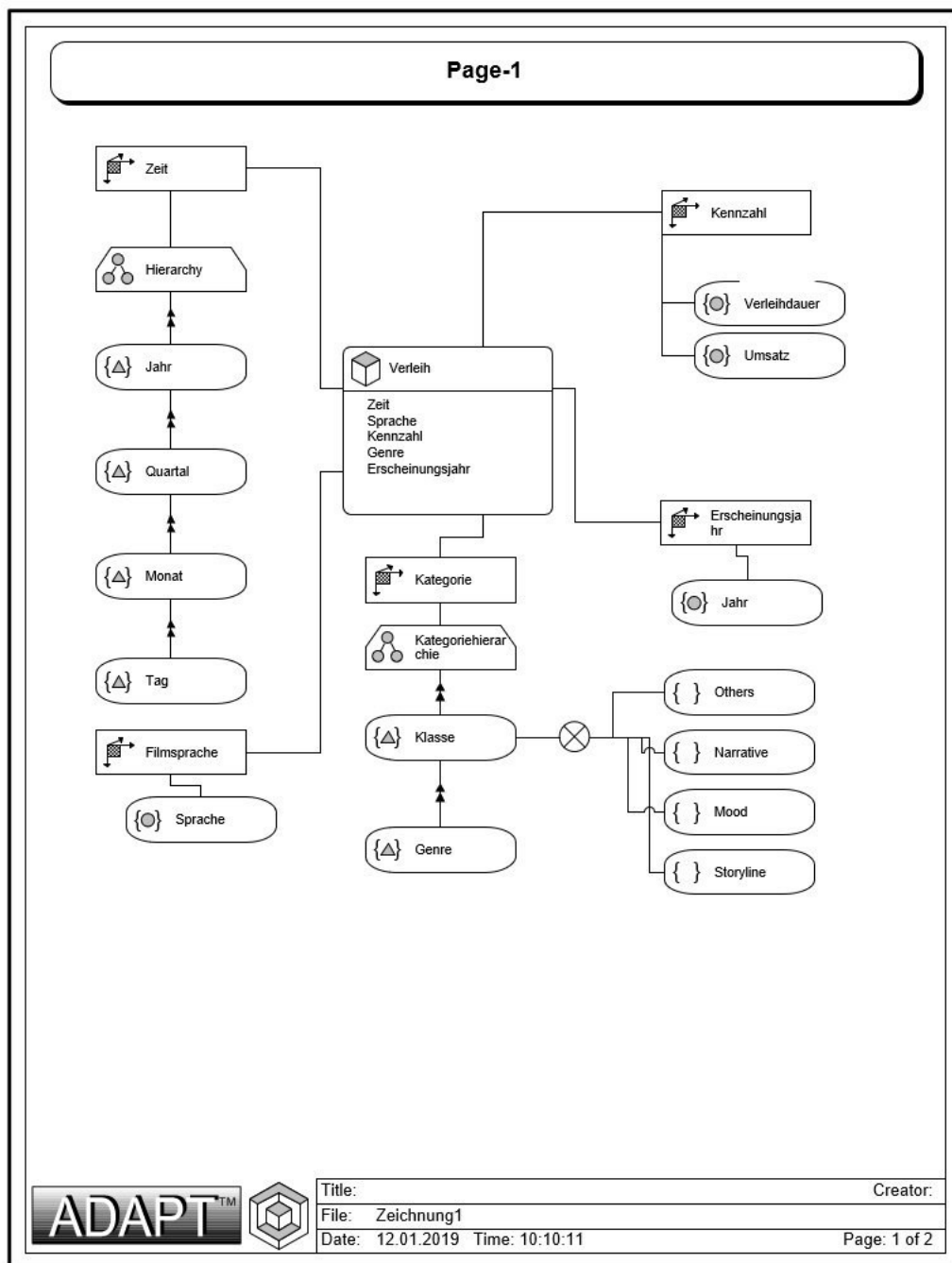
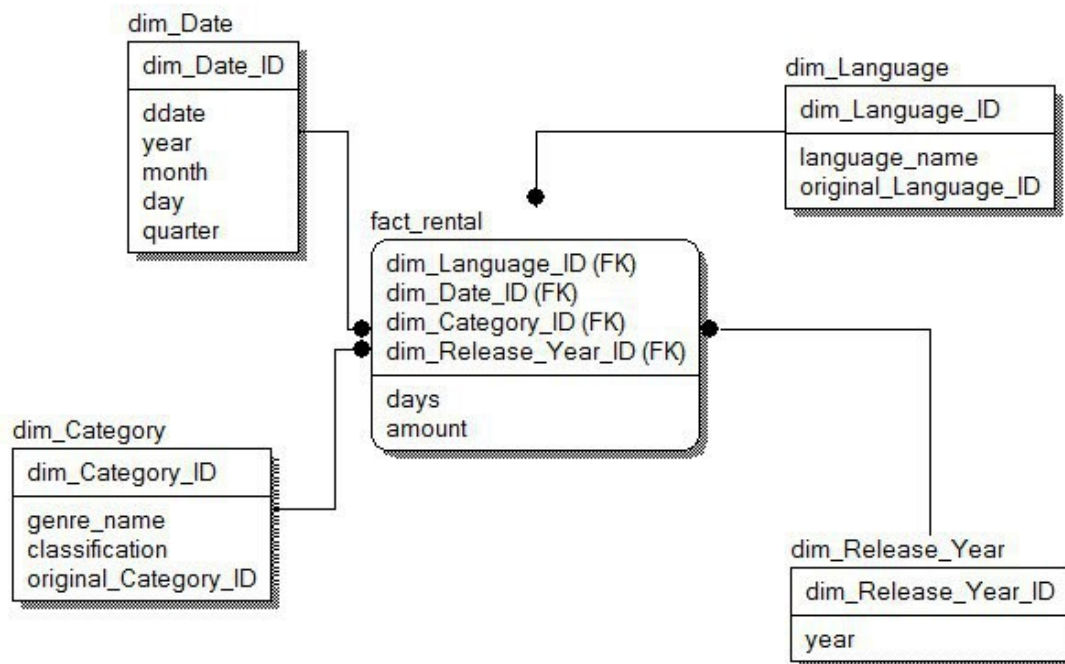


# Ausarbeitung UE12

## 1. Modellierung ADAPT



## 2. Modellierung STAR-Schema



## 3. Erstellung STAR-Schema

```

DROP TABLE dim_category;
DROP TABLE dim_date;
DROP TABLE dim_language;
DROP TABLE dim_release_year;
DROP TABLE fact_rental;
DROP SEQUENCE identifiers;
  
```

```

----- GLOBAL PK GENERATOR -----
CREATE SEQUENCE identifiers
  START WITH 1
  INCREMENT BY 1
  MAXVALUE 10000000000;

----- DIMENSION RELEASE YEAR -----
CREATE TABLE dim_release_year AS
SELECT identifiers.nextval AS dim_release_year_id, year
FROM (
  SELECT DISTINCT release_year AS year
  FROM film
);

ALTER TABLE dim_release_year
  ADD CONSTRAINT dim_release_year_pk
  PRIMARY KEY (dim_release_year_id);
  
```

```
ALTER TABLE dim_release_year
  MODIFY year NUMBER NOT NULL;
```

```
----- DIMENSION CATEGORY -----
CREATE TABLE dim_category AS
SELECT identifiers.nextval AS dim_category_id,
       name                AS genre_name,
       'no_classification' AS classification,
       category_id         AS original_category_id
FROM (
  SELECT name, category_id
  FROM category
);
```

```
ALTER TABLE dim_category
  ADD CONSTRAINT
    dim_category_pk PRIMARY KEY (dim_category_id);
```

```
UPDATE dim_category
SET classification = 'Storyline'
WHERE genre_name IN ('Animation', 'Sci-Fi', 'Sports');
```

```
UPDATE dim_category
SET classification = 'Narrative'
WHERE genre_name IN ('Children', 'Comedy', 'Documentary',
                    'Drama', 'Family', 'Foreign', 'Travel');
```

```
UPDATE dim_category
SET classification = 'Mood'
WHERE genre_name IN ('Action', 'Horror', 'Music');
```

```
UPDATE dim_category
SET classification = 'Others'
WHERE genre_name NOT IN ('Animation', 'Sci-Fi', 'Sports',
                        'Children', 'Comedy', 'Documentary',
                        'Drama', 'Family', 'Foreign',
                        'Travel', 'Action', 'Horror', 'Music');
```

```
----- DIMENSION LANGUAGE -----
CREATE TABLE dim_language AS
SELECT identifiers.nextval AS dim_language_id,
       name                AS language_name,
       language_id         AS original_language_id
FROM (
  SELECT name, language_id
  FROM language
);
```

```
ALTER TABLE dim_language
  ADD CONSTRAINT
    dim_language_pk PRIMARY KEY (dim_language_id);
```

```
----- DIMENSION DATE -----
CREATE TABLE dim_date
(
  dim_date_id NUMBER,
  ddate       DATE NOT NULL,
  year        NUMBER NOT NULL,
  month       NUMBER NOT NULL,
  quarter     NUMBER NOT NULL,
  day         NUMBER NOT NULL
);
```

```
ALTER TABLE dim_date
  ADD CONSTRAINT dim_date_pk PRIMARY KEY (dim_date_id);
```

```

DECLARE
    mindate DATE;
    maxdate DATE := TRUNC(SYSDATE);
BEGIN
    SELECT MIN(payment_date) INTO mindate FROM payment;
    FOR i IN 0..(maxdate - mindate)
        LOOP
            INSERT INTO dim_date (dim_date_id, ddate, year, month, quarter, day)
            VALUES (TO_CHAR(mindate + NUMTODSINTERVAL(i, 'day'), 'yyyymmdd'),
                    mindate + NUMTODSINTERVAL(i, 'day'),
                    EXTRACT(YEAR FROM (mindate + NUMTODSINTERVAL(i, 'day'))),
                    EXTRACT(MONTH FROM (mindate + NUMTODSINTERVAL(i, 'day'))),
                    TO_CHAR(mindate + NUMTODSINTERVAL(i, 'day'), 'Q'),
                    TO_CHAR(mindate + NUMTODSINTERVAL(i, 'day'), 'DD'));
        END LOOP;
    END;
/

CREATE MATERIALIZED VIEW fact_rental REFRESH COMPLETE
START WITH (add_months(TRUNC(SYSDATE, 'mm'), 1) + 0) - 1 / 24
NEXT (add_months(TRUNC(SYSDATE, 'mm'), 2) + 0) - 1 / 24
AS
SELECT CEIL(return_date - rental_date) days,
       amount,
       dim_language.dim_language_id,
       dim_release_year.dim_release_year_id,
       dim_category.dim_category_id,
       dim_date.dim_date_id
FROM rental
   LEFT JOIN payment USING (rental_id)
   INNER JOIN inventory USING (inventory_id)
   INNER JOIN film USING (film_id)
   INNER JOIN film_category USING (film_id)
   INNER JOIN dim_category ON (film_category.category_id =
dim_category.original_category_id)
   INNER JOIN dim_language ON (film.language_id = dim_language.original_language_id)
   INNER JOIN dim_release_year ON (film.release_year = dim_release_year.year)
   INNER JOIN dim_date ON (TRUNC(payment_date) = ddate);
BEGIN
    dbms_mview.REFRESH('fact_rental');
END;

SELECT *
FROM fact_rental;

```

## 4. Abfragen & Interpretation

```

----- Nr. 1 -----
SELECT count,
       language_name
FROM (
    SELECT dim_language_id, COUNT(*) AS count
    FROM fact_rental
    GROUP BY dim_language_id
    ORDER BY COUNT(*) DESC FETCH FIRST 2 ROWS
    WITH TIES
)
   INNER JOIN dim_language USING (dim_language_id);

-- Count Language

```

```
-- -----
-- 3111   German
-- 2940   Italian
```

```
-- Interpretation: Vielleicht hat das Unternehmen in Deutschland und Italien die meisten
--                  Filialen oder einfach nur die größte Kundschaft.
```

```
----- Nr. 2 -----
SELECT ROUND(SUM(amount) / COUNT(*), 3) AS avg_revenue
FROM fact_rental;
```

```
-- Average Revenue
-- -----
-- 7.208
```

```
-- Interpretation: Im Durchschnitt wird mit einem Verleihvorgang 7.21€ Umsatz erzielt.
```

```
----- Nr. 3 -----
SELECT genre_name,
       ROUND(SUM(amount) / COUNT(day), 3) AS avg_revenue_per_day
FROM fact_rental
     INNER JOIN dim_date USING (dim_date_id)
     INNER JOIN dim_category USING (dim_category_id)
GROUP BY genre_name
ORDER BY avg_revenue_per_day DESC FETCH FIRST ROW ONLY;
```

```
-- Genre Name      Average revenue per day
-- -----
-- Sci-Fi          8.7
```

```
-- Interpretation: Man sollte sich überlegen, mehr Sci-Fi Filme in den Bestand
aufzunehmen.
```

```
----- Nr. 4 -----
SELECT SUM(amount) AS total_revenue,
       quarter
FROM fact_rental
     INNER JOIN dim_date USING (dim_date_id)
GROUP BY quarter
ORDER BY total_revenue DESC FETCH FIRST ROW ONLY;
```

```
-- Total Revenue   Quarter
-- -----
-- 31764.09         1
```

```
-- Interpretation: Weihnachts-, Oster- und Semesterferien sind alle im 1. Quartal,
vielleicht haben
--                  da einfach die meisten Kunden Zeit zum Filmschauen?
```

```
----- Nr. 5 -----
SELECT ROUND(AVG(days), 2) AS duration,
       quarter,
       month
FROM fact_rental
     INNER JOIN dim_date USING (dim_date_id)
GROUP BY ROLLUP (quarter, month);
```

```
-- Duration   Quarter   Month
-- -----
-- 5.54        1         1
-- 5.14        1         2
-- 4.68        1         3
-- 5.11        1         <null>
-- 4.51        2         4
```

```
-- 4.21      2      5
-- 3.99      2      6
-- 4.23      2      <null>
-- 4.05      3      7
-- 4.34      3      8
-- 4.5        3      9
-- 4.29      3      <null>
-- 4.49      4      10
-- 4.78      4      11
-- 5.36      4      12
-- 4.84      4      <null>
-- 4.6       <null>  <null>
```

-- Interpretation: Im 1. Quartal ist die Verleihdauer am längsten, was sich mit der Vermutung aus Nr. 4

-- deckt, dass Kunden in Q1 mehr Zeit zum Filmschaun haben. Nachdem Q1 M1 und Q4 M12 die längste Verleihdauer aufweisen, wird dies weiter bestätigt.

----- Nr. 6 -----

```
SELECT genre_name,
       language_name,
       SUM(amount) AS revenue,
       COUNT(*)
FROM fact_rental
     INNER JOIN dim_category USING (dim_category_id)
     INNER JOIN dim_language USING (dim_language_id)
GROUP BY CUBE (genre_name, language_name);
```

```
-- ROW  Genre      Revenue
-- ---  -
-- 8     Sci-Fi      9579.03
-- 9     Family      8380.94
-- 10    Documentary  8147.83
-- 11    Sports      8143.9
-- ROW  Language    Revenue
-- ---  -
-- 2     German      22050.77
-- 3     Italian     21423.05
-- 4     Japanese    21086.4
```

-- Interpretation: Die Genres Sci-Fi, Family, Documentary und Sports sowie die Sprachen German, Italian und Japanese sind am lukrativsten und aus diesem Grund sollten Filme mit diesen Attributen nachgekauft werden.

----- Nr. 7 -----

```
SELECT year, language_name, COUNT(*) AS rental_amounts
FROM fact_rental
     INNER JOIN dim_release_year USING (dim_release_year_id)
     INNER JOIN dim_language USING (dim_language_id)
GROUP BY ROLLUP (year, language_name)
ORDER BY rental_amounts ASC, year ASC;
```

```
-- Year  Rental Amounts
-- ---  -
-- 1983  384
-- Language  Rental Amounts
-- ---  -
-- Mandarin  29
-- English    53
-- German     54
```

-- Interpretation: Filme aus dem Jahr 1983 wurden am wenigsten ausgeliehen. Warum auch immer man die am wenigsten  
 -- oft verliehenen Filme nachkaufen wollen würde? Filme der Sprachen  
 Mandarin, Englisch und  
 -- German wurden in diesem Jahr am wenigsten oft ausgeliehen.

----- Nr. 8 -----

```

SELECT year,
       month,
       rental_amount,
       total_revenue,
       SUM(rental_amount) OVER (
         PARTITION BY year ORDER BY year ASC
         ROWS BETWEEN UNBOUNDED PRECEDING AND
         CURRENT ROW) AS rentals_up_to_month_x -- adds the accumulation of rental
amounts during a year
FROM (
  SELECT year,
         month,
         COUNT(*) AS rental_amount,
         SUM(amount) AS total_revenue
  FROM fact_rental
       INNER JOIN dim_date USING (dim_date_id)
       INNER JOIN dim_category USING (dim_category_id)
  WHERE classification = 'Narrative'
  GROUP BY year, month
  ORDER BY year ASC, month ASC, rental_amount ASC, total_revenue ASC
);

```

-- Year	Month	Rental Amount	Total Revenue	Rentals Up To Month X
-- ----	----	-----	-----	-----
-- 2013	12	176	1431.96	176
-- 2014	1	312	2670.62	312
-- 2014	2	260	2258.82	572
-- 2014	3	301	2289.47	873
-- 2014	4	298	2195.34	1171
-- 2014	5	287	2044.02	1458
-- 2014	6	284	1798.96	1742
-- 2014	7	333	2130.33	2075
-- 2014	8	328	2253.95	2403
-- 2014	9	256	1885.39	2659
-- 2014	10	322	2378.18	2981
-- 2014	11	287	2230.53	3268
-- 2014	12	309	2527.21	3577
-- 2015	1	301	2512.96	301
-- 2015	2	279	2197.39	580
-- 2015	3	322	2367.02	902
-- 2015	4	281	2223.34	1183
-- 2015	5	316	2245.68	1499
-- 2015	6	317	1959.34	1816
-- 2015	7	347	2418.49	2163
-- 2015	8	319	2166.48	2482
-- 2015	9	282	1979.1	2764
-- 2015	10	318	2298.68	3082
-- 2015	11	127	933.16	3209

-- Interpretation: Wir haben die ganze Tabelle eingefügt, weil in der Angabe keine  
 explizite Interpretation  
 -- gefordert war und wir auch nicht wissen, was wir mit den erhaltenen  
 Daten anstellen sollen.  
 -- Die Tabelle ist sicher nützlich als Zwischenprodukt für weitere  
 Datenauswertungen, sagt in  
 -- dieser Form aber wenig aus. Gegen Ende des Jahres wurden meistens  
 (akkumuliert) mehr Filme

11.1.2019

DES3

Johann Hoffmann & Niklas Vest

--

*ausgeliehen als im Jänner "\\_ツ\_/"*