Dani Schnider

Nach den Sternen greifen – Herausforderungen im Star Schema

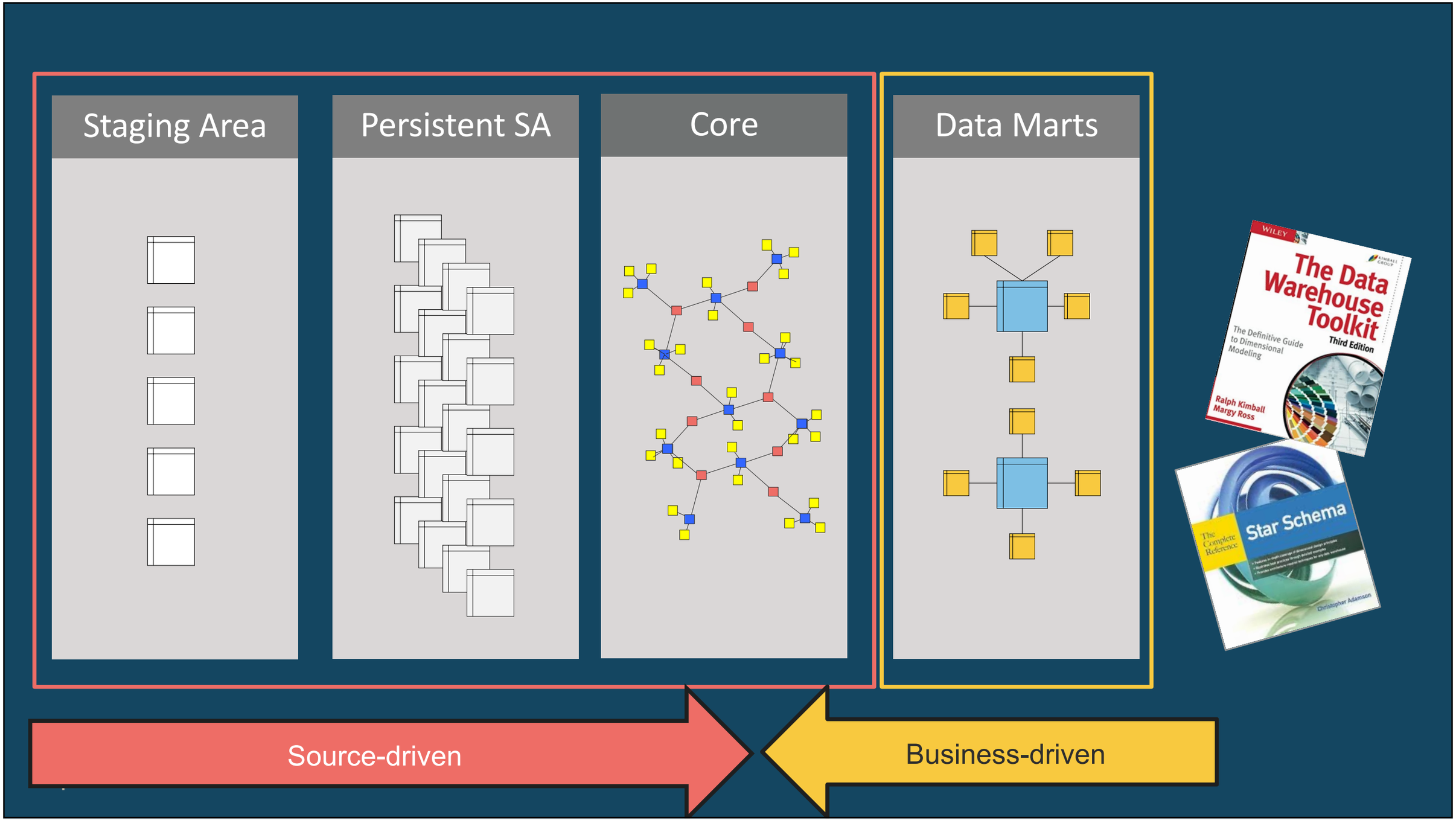
Callista

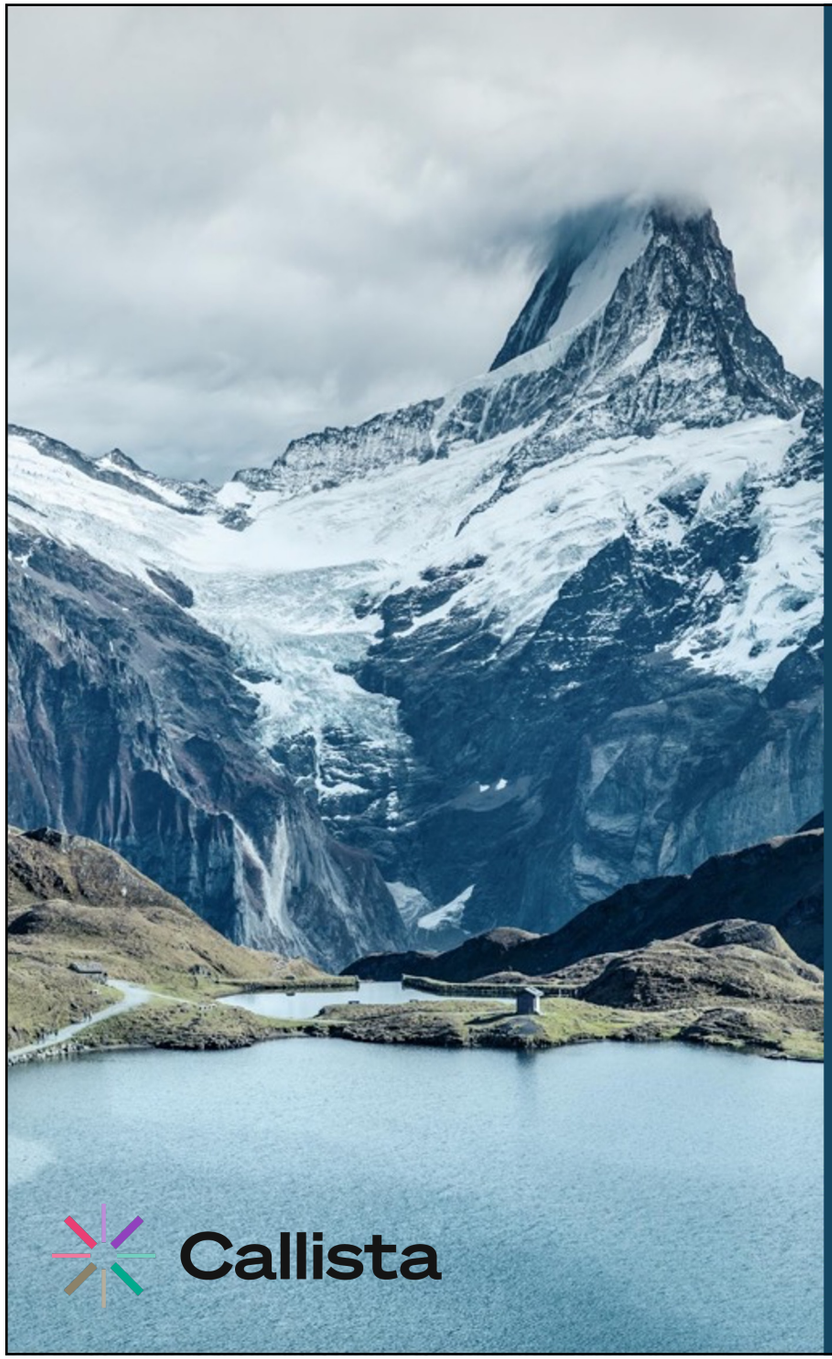
About me

Dani Schnider

- Working for Callista
- Oracle ACE Director
- Member of Symposium 42
- Hobby: Craft Beer Brewing

@dani_schnider danischnider.wordpress.com www.linkedin.com/in/danischnider/


Staging Area
Persistent SA
Core
Data Marts
WILEY
The Data Warehouse Toolkit
Third Edition
The Definitive Guide to Dimensional Modeling
Ralph Kimball
Margy Ross
The Complete Reference
Star Schema
Christopher Adamson
Source-driven
Business-driven

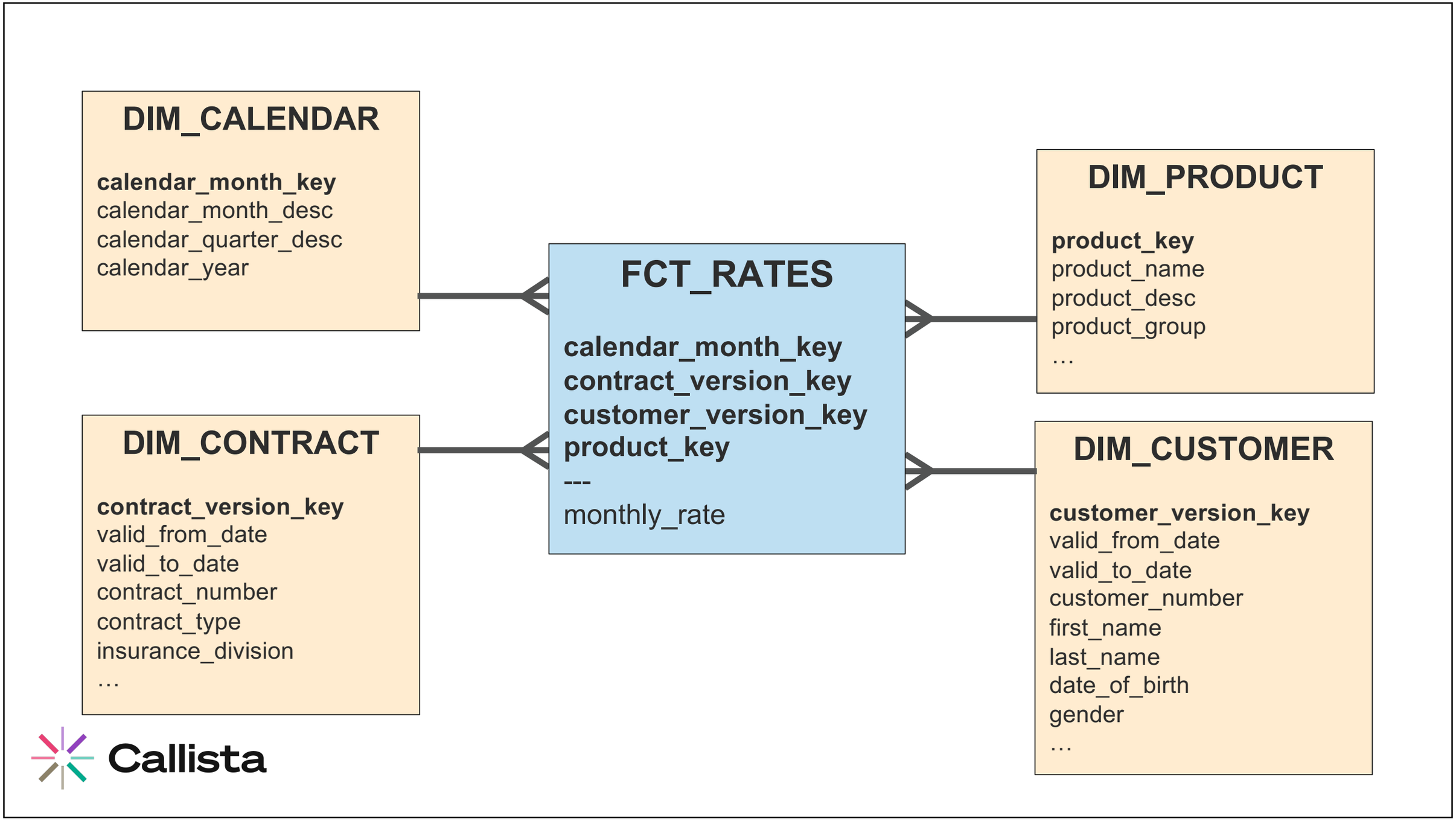


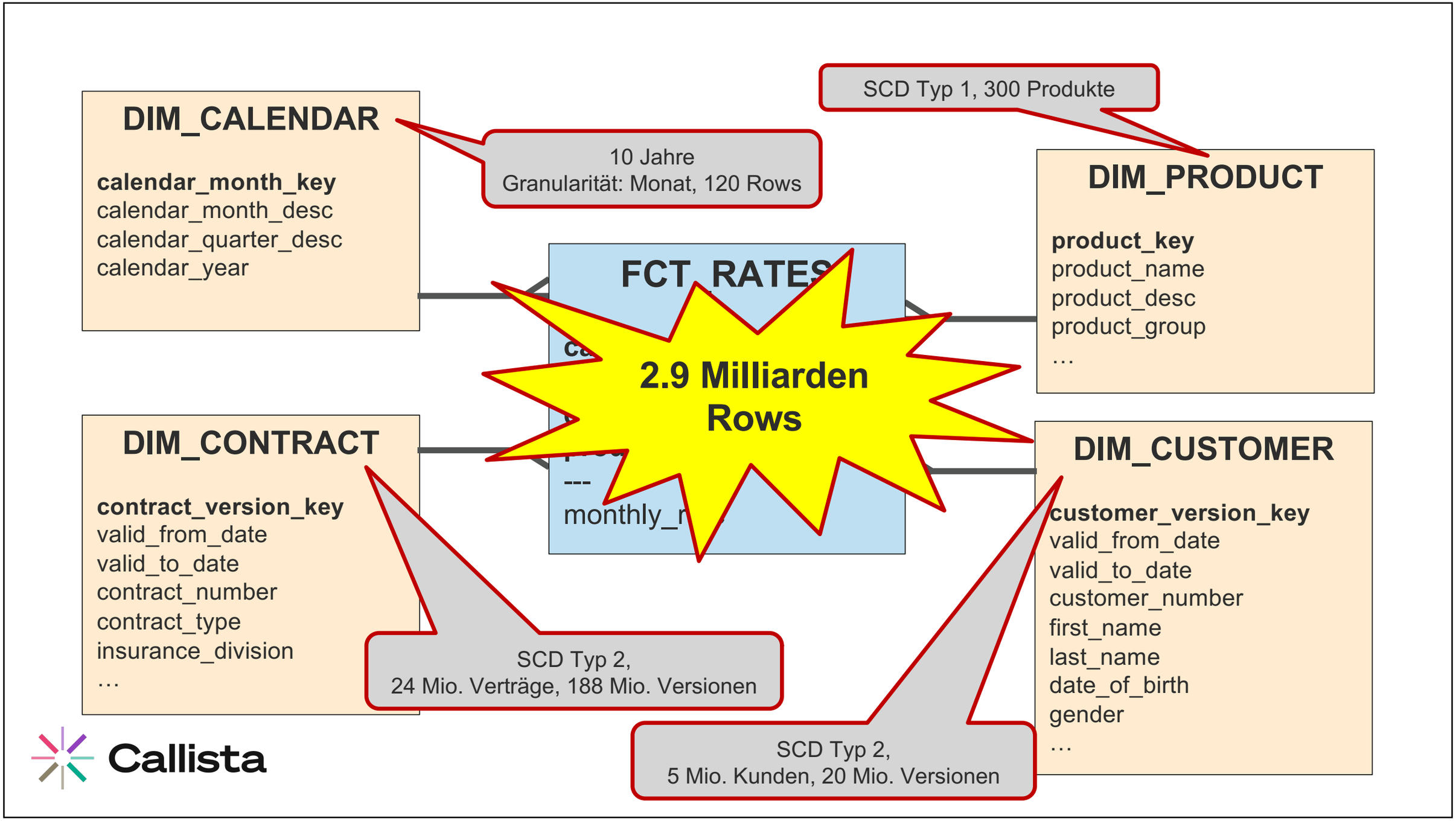
Auswertungen über Versicherungsprämien

- Jahresprämienvolumen pro Versicherungssparte
- Prämienvolumen pro Monat und Wohnregion der Kunden
- Durchschnittliche Monatsprämie pro Produkt
- Entwicklung der Jahresprämien pro Vertragsart
- …

«Star Schemas
sind einfach und
übersichtlich.»

Callista


DIM_CALENDAR
calendar_month_key
calendar_month_desc
calendar_quarter_desc
calendar_year
DIM_CONTRACT
contract_version_key
valid_from_date
valid_to_date
contract_number
contract_type
insurance_division
…
FCT_RATES
calendar_month_key
contract_version_key
customer_version_key
product_key
---
monthly_rate
DIM_PRODUCT
product_key
product_name
product_desc
product_group
…
DIM_CUSTOMER
customer_version_key
valid_from_date
valid_to_date
customer_number
first_name
last_name
date_of_birth
gender
…
Callista

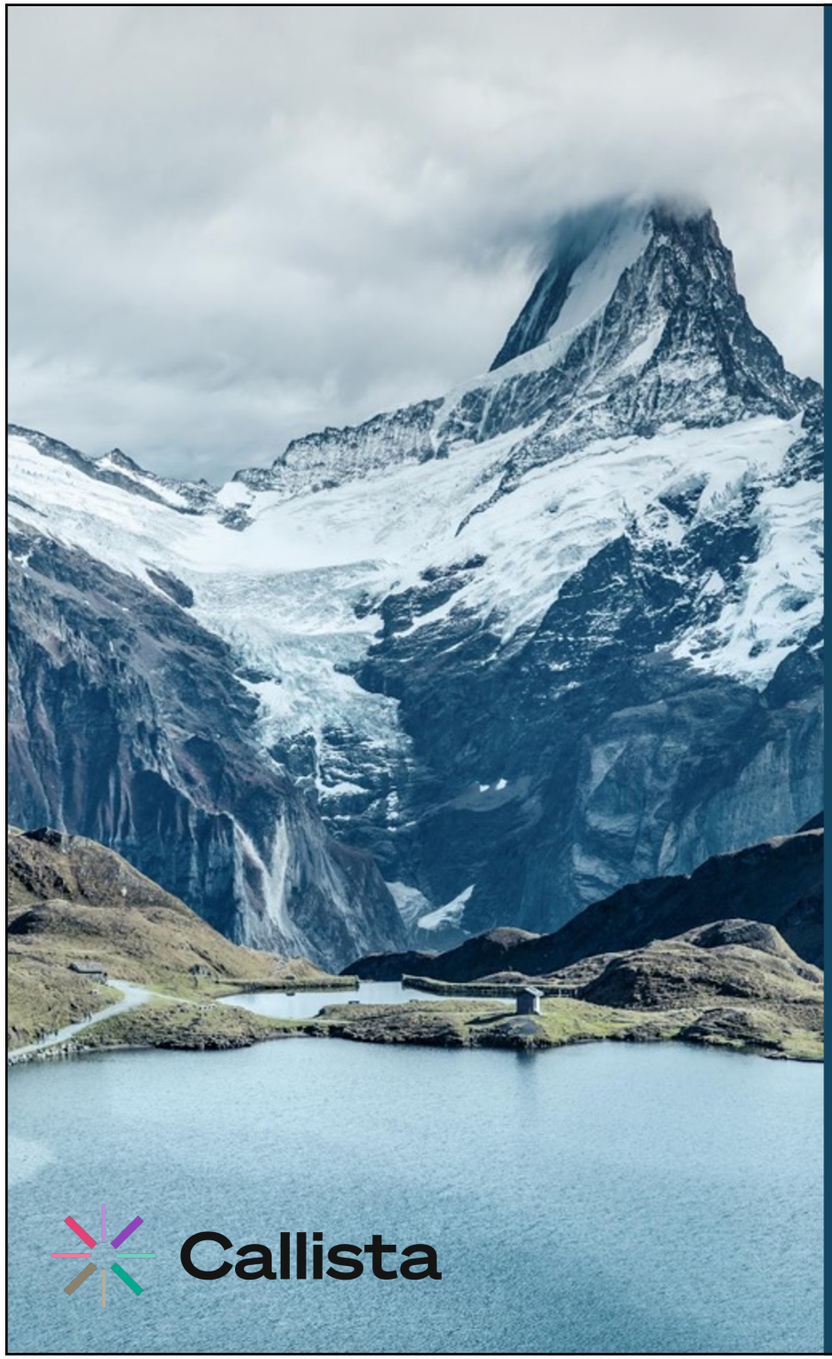

DIM_CALENDAR
calendar_month_key
calendar_month_desc
calendar_quarter_desc
calendar_year
10 Jahre
Granularität: Monat, 120 Rows
SCD Typ 1, 300 Produkte
DIM_PRODUCT
product_key
product_name
product_desc
product_group
…
FCT_RATES
2.9 Milliarden Rows
DIM_CONTRACT
contract_version_key
valid_from_date
valid_to_date
contract_number
contract_type
insurance_division
…
SCD Typ 2,
24 Mio. Verträge, 188 Mio. Versionen
DIM_CUSTOMER
customer_version_key
valid_from_date
valid_to_date
customer_number
first_name
last_name
date_of_birth
gender
…
SCD Typ 2,
5 Mio. Kunden, 20 Mio. Versionen
Callista

Abfrage Prämienvolumen

- Pro Versicherungssparte
- Für das Jahr 2023

```
SELECT cont.insurance_division
     , SUM(fct.monthly_rate)
  FROM fct_rates fct
  JOIN dim_calendar cal
    ON cal.calendar_month_key = fct.calendar_month_key
  JOIN dim_contract cont
    ON cont.contract_version_key = fct.contract_version_key
 WHERE cal.calendar_year = 2023
GROUP BY cont.insurance_division
ORDER BY cont.insurance_division
```

Callista



Zusätzliche Anforderungen

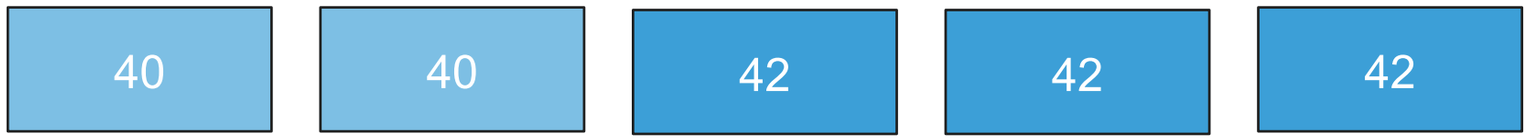
- Prämien sollen auf Tagesebene ausgewertet werden können (oder über beliebige Zeitperiode)
- Verträge können jederzeit angepasst werden (z.B. Geburt, Zusatzversicherungen, Prämienanpassungen bei Umzug, Todesfall)
- Auswertung Prämienvolumen zu bestimmten Stichtagen

Callista

Prämienvolumen pro Zeitperiode

November **Dezember** **Januar** **Februar** **März**

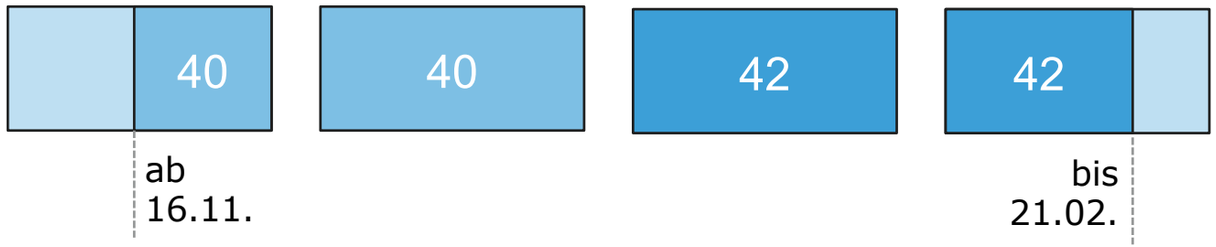
01.11.2022 – 31.03.2023:



$$(2 * 40) + (3 * 42)$$

$$= \mathbf{206.00}$$

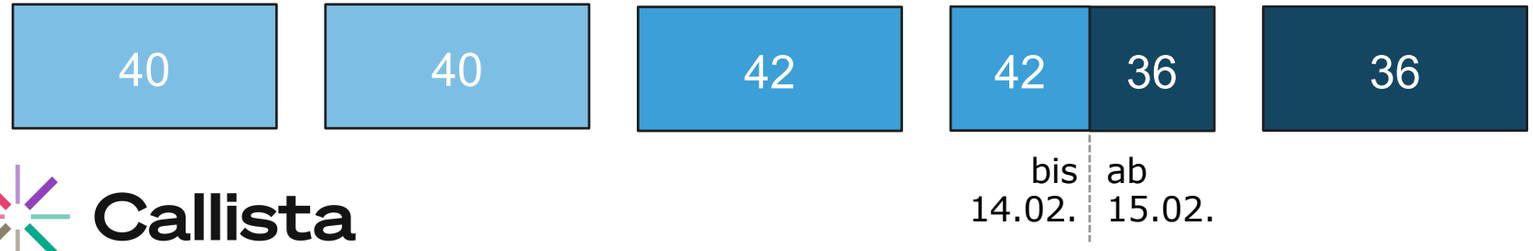
16.11.2022 – 21.02.2023:



$$\frac{15}{30} * 40 + 40 + 42 + \frac{21}{28} * 42$$

$$= \mathbf{133.50}$$

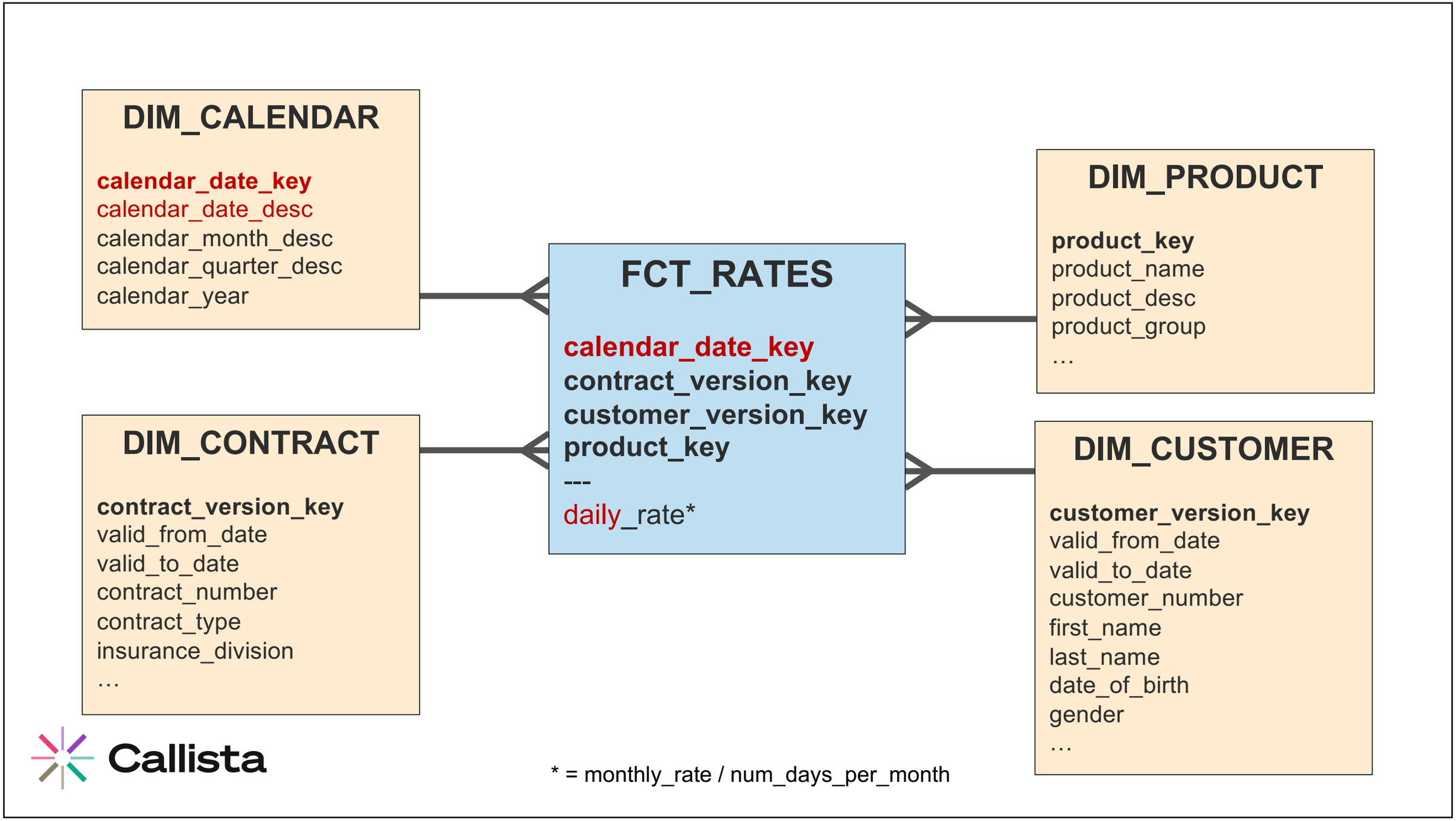
01.11.2022 – 31.03.2023:

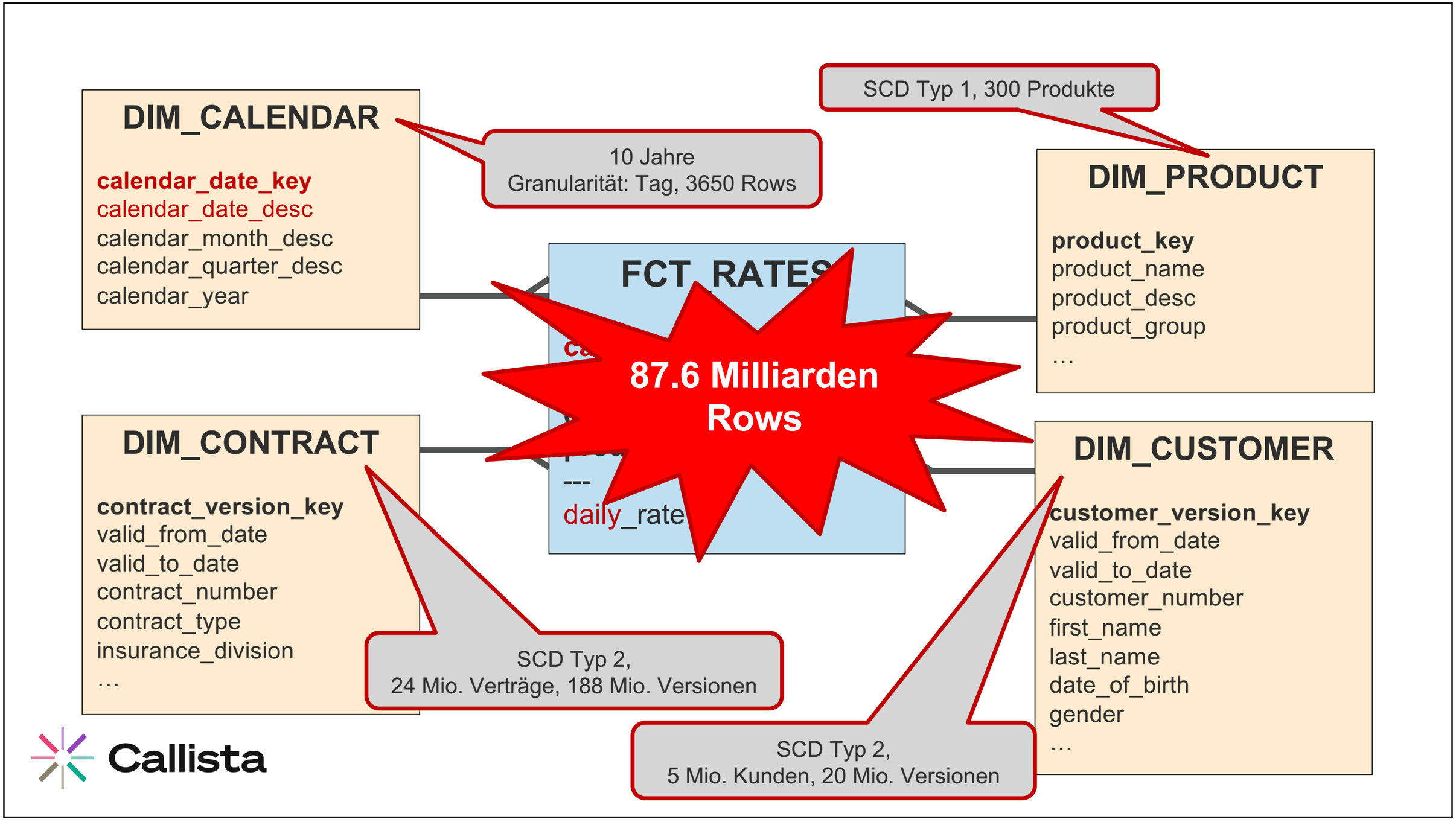


$$2 * 40 + (1 + \frac{14}{28}) * 42 + (\frac{14}{28} + 1) * 36$$

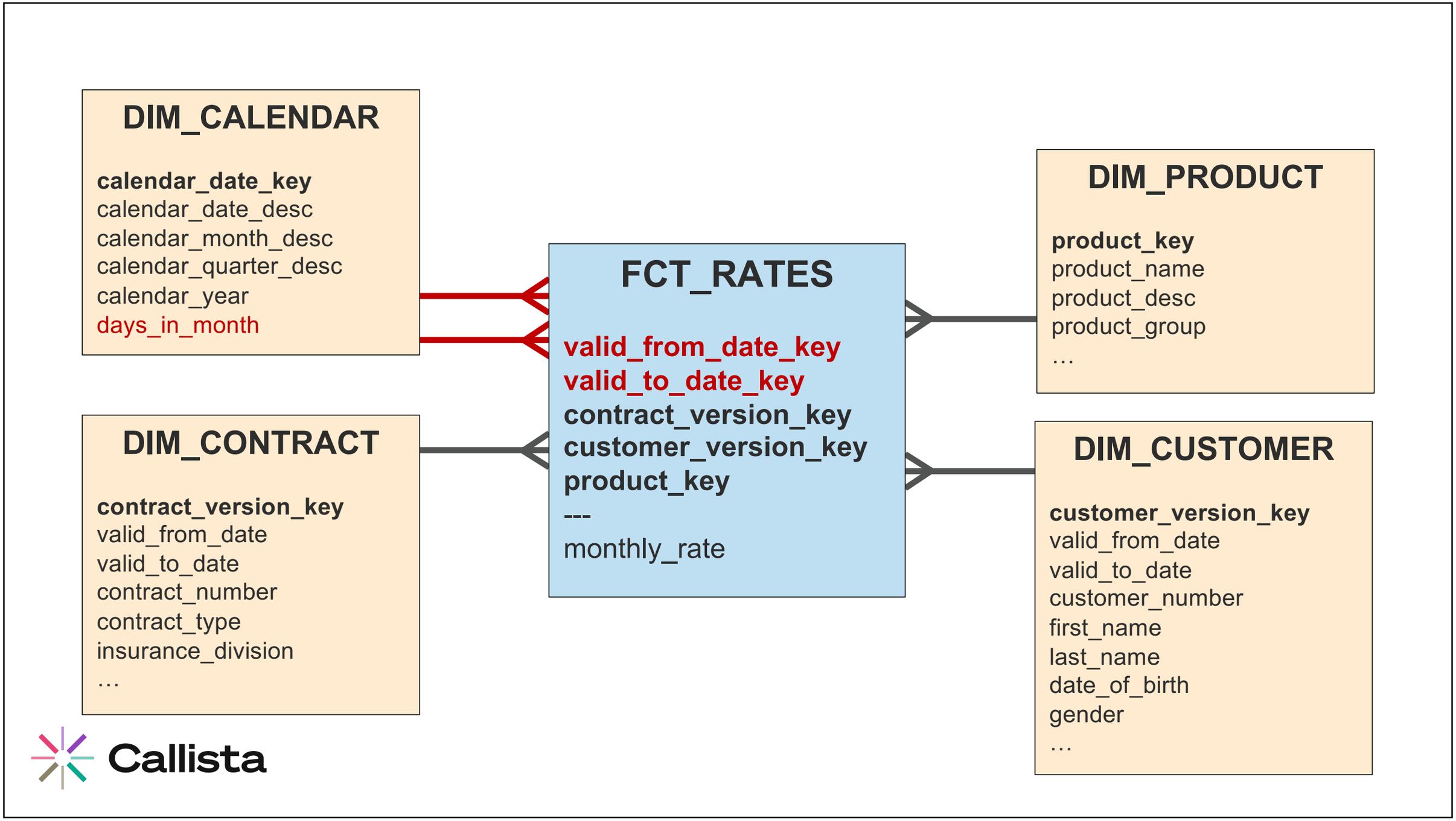
$$= \mathbf{197.00}$$

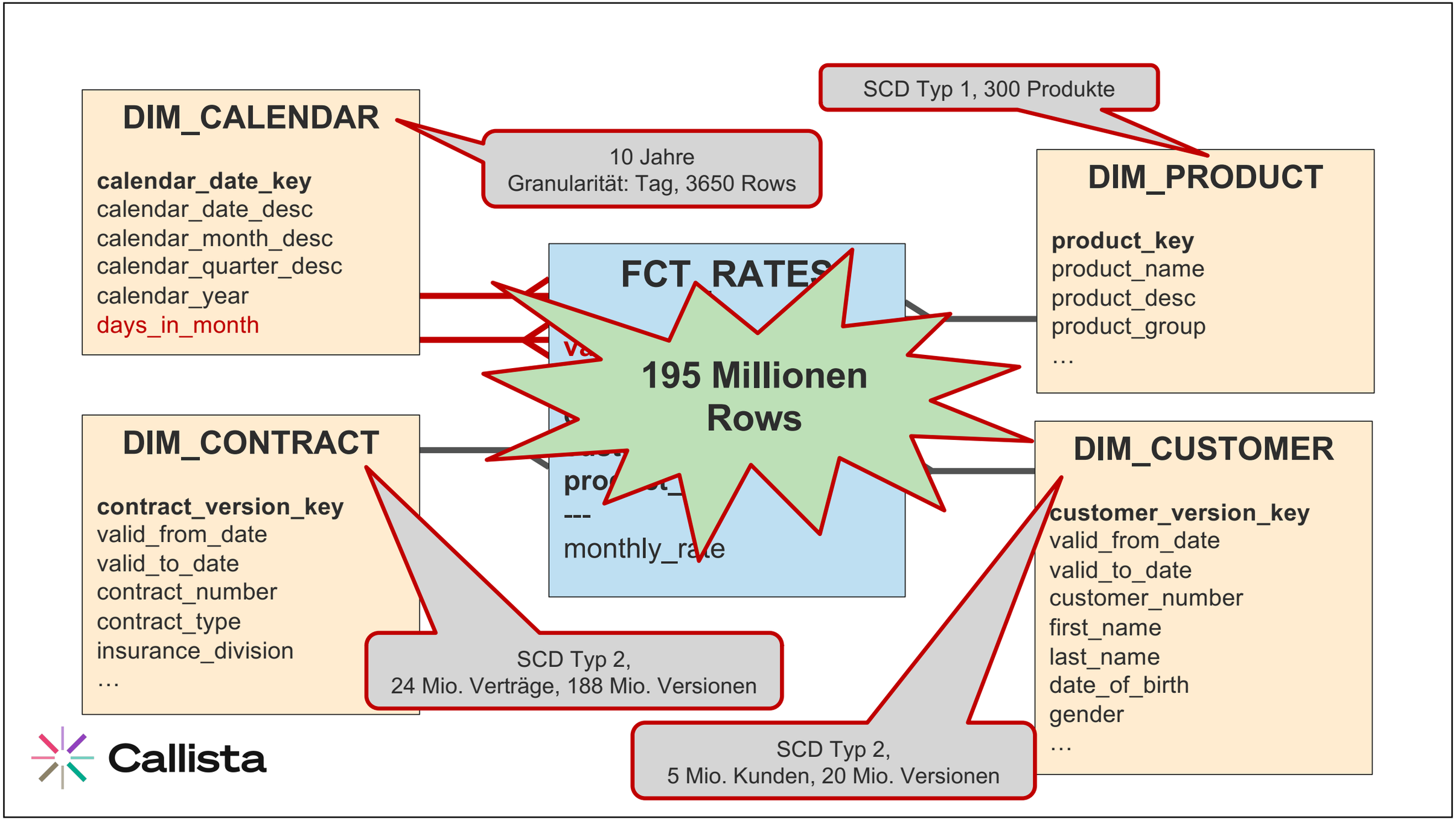
Callista


DIM_CALENDAR
calendar_date_key
calendar_date_desc
calendar_month_desc
calendar_quarter_desc
calendar_year
DIM_CONTRACT
contract_version_key
valid_from_date
valid_to_date
contract_number
contract_type
insurance_division
…
FCT_RATES
calendar_date_key
contract_version_key
customer_version_key
product_key
---
daily_rate*
DIM_PRODUCT
product_key
product_name
product_desc
product_group
…
DIM_CUSTOMER
customer_version_key
valid_from_date
valid_to_date
customer_number
first_name
last_name
date_of_birth
gender
…
Callista
* = monthly_rate / num_days_per_month


DIM_CALENDAR
calendar_date_key
calendar_date_desc
calendar_month_desc
calendar_quarter_desc
calendar_year
10 Jahre
Granularität: Tag, 3650 Rows
SCD Typ 1, 300 Produkte
DIM_PRODUCT
product_key
product_name
product_desc
product_group
…
FCT_RATES
---
daily_rate
87.6 Milliarden
Rows
DIM_CONTRACT
contract_version_key
valid_from_date
valid_to_date
contract_number
contract_type
insurance_division
…
SCD Typ 2,
24 Mio. Verträge, 188 Mio. Versionen
DIM_CUSTOMER
customer_version_key
valid_from_date
valid_to_date
customer_number
first_name
last_name
date_of_birth
gender
…
SCD Typ 2,
5 Mio. Kunden, 20 Mio. Versionen
Callista

Wie können wir die Anzahl Fakten reduzieren?


DIM_CALENDAR
calendar_date_key
calendar_date_desc
calendar_month_desc
calendar_quarter_desc
calendar_year
days_in_month
DIM_CONTRACT
contract_version_key
valid_from_date
valid_to_date
contract_number
contract_type
insurance_division
…
FCT_RATES
valid_from_date_key
valid_to_date_key
contract_version_key
customer_version_key
product_key
---
monthly_rate
DIM_PRODUCT
product_key
product_name
product_desc
product_group
…
DIM_CUSTOMER
customer_version_key
valid_from_date
valid_to_date
customer_number
first_name
last_name
date_of_birth
gender
…
Callista


DIM_CALENDAR
calendar_date_key
calendar_date_desc
calendar_month_desc
calendar_quarter_desc
calendar_year
days_in_month
10 Jahre
Granularität: Tag, 3650 Rows
SCD Typ 1, 300 Produkte
DIM_PRODUCT
product_key
product_name
product_desc
product_group
…
FCT_RATES
monthly_rate
195 Millionen Rows
DIM_CONTRACT
contract_version_key
valid_from_date
valid_to_date
contract_number
contract_type
insurance_division
…
SCD Typ 2,
24 Mio. Verträge, 188 Mio. Versionen
DIM_CUSTOMER
customer_version_key
valid_from_date
valid_to_date
customer_number
first_name
last_name
date_of_birth
gender
…
SCD Typ 2,
5 Mio. Kunden, 20 Mio. Versionen
Callista

2’900’000’000

87’600’000’000

195’000’000

Callista

MONTHLY_RATE ist nicht mehr additiv.

Callista

View V_FCT_RATES_DAILY

```
CREATE OR REPLACE VIEW v_fct_rates_daily AS
SELECT cal.calendar_date_key
     , fct.*
     , fct.monthly_rate / cal.days_in_month AS daily_rate
  FROM fct_rates fct
  JOIN dim_calendar cal
    ON cal.calendar_date_key BETWEEN fct.valid_from_date_key
                                 AND fct.valid_to_date_key
```

Callista

Abfrage Prämienvolumen

```
SELECT cont.insurance_division
     , ROUND(SUM(fct.daily_rate), 2)
  FROM v_fct_rates_daily fct
  JOIN dim_calendar cal
    ON cal.calendar_date_key = fct.calendar_date_key
  JOIN dim_contract cont
    ON cont.contract_version_key = fct.contract_version_key
 WHERE cal.calendar_year = 2023
GROUP BY cont.insurance_division
ORDER BY cont.insurance_division
```

Probleme:

- Rundungsfehler
- Performance!!!!

Callista

View V_FCT_RATES_INTERVAL

```
CREATE OR REPLACE VIEW v_fct_rates_interval AS
WITH cte_dates AS (
   SELECT fct.*
        , cal_from.calendar_date_key AS input_from_date
        , cal_to.calendar_date_key AS input_to_date
        , GREATEST(fct.valid_from_date_key, cal_from.calendar_date_key) AS interval_from_date
        , LEAST(fct.valid_to_date_key, cal_to.calendar_date_key) AS interval_to_date
     FROM fct_rates fct
     JOIN dim_calendar cal_to
       ON fct.valid_from_date_key <= cal_to.calendar_date_key
     JOIN dim_calendar cal_from
       ON fct.valid_to_date_key >= cal_from.calendar_date_key
      AND cal_from.calendar_date_key <= cal_to.calendar_date_key
)
SELECT cte_dates.*
     , monthly_rate *
       (MONTHS_BETWEEN(TRUNC(interval_to_date, 'MONTH'), TRUNC(interval_from_date, 'MONTH'))
          - CAST(interval_from_date - TRUNC(interval_from_date, 'MONTH') AS NUMBER)
            / (LAST_DAY(interval_from_date) - TRUNC(interval_from_date, 'MONTH') + 1)
          + CAST(interval_to_date - TRUNC(interval_to_date, 'MONTH') + 1 AS NUMBER)
            / (LAST_DAY(interval_to_date) - TRUNC(interval_to_date, 'MONTH') + 1)) AS interval_rate
  FROM cte_dates;
```

Callista

Abfrage 1. Januar 2022 – 31. Dezember 2023

Worksheet | Query Builder

```
1  SELECT valid_from_date_key, valid_to_date_key, monthly_rate
2       , interval_from_date, interval_to_date, interval_rate
3    FROM v_fct_rates_interval fct
4    JOIN dim_calendar cal
5      ON cal.calendar_date_key = fct.valid_from_date_key
6    JOIN dim_contract cont
7      ON cont.contract_version_key = fct.contract_version_key
8   WHERE fct.input_from_date = DATE'2022-01-01'
9     AND fct.input_to_date = DATE'2023-12-31'
10    AND cont.contract_number = '4711';
```

Script Output | Query Result

SQL | All Rows Fetched: 4 in 0,013 seconds

	VALID_FROM_DATE_KEY	VALID_TO_DATE_KEY	MONTHLY_RATE	INTERVAL_FROM_DATE	INTERVAL_TO_DATE	INTERVAL_RATE
1	01.01.2022	15.11.2022	40	01.01.2022	15.11.2022	420
2	16.11.2022	31.12.2022	40	16.11.2022	31.12.2022	60
3	01.01.2023	14.02.2023	42	01.01.2023	14.02.2023	63
4	15.02.2023	31.12.2023	36	15.02.2023	31.12.2023	378

Callista

Abfrage 25. November 2022 – 10. April 2023

Worksheet | Query Builder

```
1  SELECT valid_from_date_key, valid_to_date_key, monthly_rate
2       , interval_from_date, interval_to_date, interval_rate
3    FROM v_fct_rates_interval fct
4    JOIN dim_calendar cal
5      ON cal.calendar_date_key = fct.valid_from_date_key
6    JOIN dim_contract cont
7      ON cont.contract_version_key = fct.contract_version_key
8   WHERE fct.input_from_date = DATE'2022-11-25'
9     AND fct.input_to_date = DATE'2023-04-10'
10    AND cont.contract_number = '4711';
```

Script Output | Query Result

SQL | All Rows Fetched: 3 in 0,004 seconds

	VALID_FROM_DATE_KEY	VALID_TO_DATE_KEY	MONTHLY_RATE	INTERVAL_FROM_DATE	INTERVAL_TO_DATE	INTERVAL_RATE
1	16.11.2022	31.12.2022	40	25.11.2022	31.12.2022	48
2	01.01.2023	14.02.2023	42	01.01.2023	14.02.2023	63
3	15.02.2023	31.12.2023	36	15.02.2023	10.04.2023	66

Callista

Abfrage Prämienvolumen

```
SELECT cont.insurance_division
     , SUM(fct.interval_rate)
  FROM v_fct_rates_interval fct
  JOIN dim_calendar cal
    ON cal.calendar_date_key = fct.valid_from_date_key
  JOIN dim_contract cont
    ON cont.contract_version_key = fct.contract_version_key
 WHERE fct.input_from_date = DATE'2023-01-01'
   AND fct.input_to_date = DATE'2023-12-31'
GROUP BY cont.insurance_division
ORDER BY cont.insurance_division
```

Hinweis:

- input_from_date und input_to_date müssen zwingend angegeben werden

Callista

SQL Macro FN_MONTHS_BETWEEN

```
CREATE OR REPLACE FUNCTION fn_months_between (p_from_date DATE, p_to_date DATE)
   RETURN VARCHAR2 SQL_MACRO(SCALAR)
IS
BEGIN
   RETURN q'{MONTHS_BETWEEN(TRUNC(p_to_date, 'MONTH'), TRUNC(p_from_date, 'MONTH'))
              - CAST(p_from_date - TRUNC(p_from_date, 'MONTH') AS NUMBER)
                / (LAST_DAY(p_from_date) - TRUNC(p_from_date, 'MONTH') + 1)
              + CAST(p_to_date - TRUNC(p_to_date, 'MONTH') + 1 AS NUMBER)
                / (LAST_DAY(p_to_date) - TRUNC(p_to_date, 'MONTH') + 1)
   }';
END;
/
```

Callista

ORACLE DATABASE 21c

SQL Macro FN_FCT_RATES_INTERVAL

```
CREATE OR REPLACE FUNCTION fn_fct_rates_interval (p_from_date DATE, p_to_date DATE)
   RETURN VARCHAR2 SQL_MACRO(TABLE)
IS
BEGIN
   RETURN q'{
   SELECT fct.*
        , GREATEST(fct.valid_from_date_key, p_from_date) AS interval_from_date
        , LEAST(fct.valid_to_date_key, p_to_date) AS interval_to_date
        , fct.monthly_rate *
          fn_months_between (GREATEST(fct.valid_from_date_key, p_from_date),
                             LEAST(fct.valid_to_date_key, p_to_date)) interval_rate
     FROM fct_rates fct
    WHERE fct.valid_from_date_key <= p_to_date
      AND fct.valid_to_date_key >= p_from_date
      AND p_from_date <= p_to_date
   }';
END;
/
```

Callista

ORACLE DATABASE 21c

Abfrage Prämienvolumen

```
SELECT cont.insurance_division
     , SUM(fct.interval_rate)
  FROM fn_fct_rates_interval(DATE'2023-01-01',DATE'2023-12-31') fct
  JOIN dim_calendar cal
    ON cal.calendar_date_key = fct.valid_from_date_key
  JOIN dim_contract cont
    ON cont.contract_version_key = fct.contract_version_key
GROUP BY cont.insurance_division
ORDER BY cont.insurance_division;
```

Hinweise:

- Intervalleinschränkung über Parameter von SQL Macro
- Performance deutlich besser als mit Intervall-View

Callista

Abfrage 25. November 2022 – 10. April 2023

Worksheet | Query Builder

```
1  SELECT valid_from_date_key, valid_to_date_key, monthly_rate
2       , interval_from_date, interval_to_date, interval_rate
3    FROM fn_fct_rates_interval(DATE'2022-11-25', DATE'2023-04-10') fct
4    JOIN dim_calendar cal
5      ON cal.calendar_date_key = fct.valid_from_date_key
6    JOIN dim_contract cont
7      ON cont.contract_version_key = fct.contract_version_key
8   WHERE cont.contract_number = '4711';
9
```

Script Output | Query Result

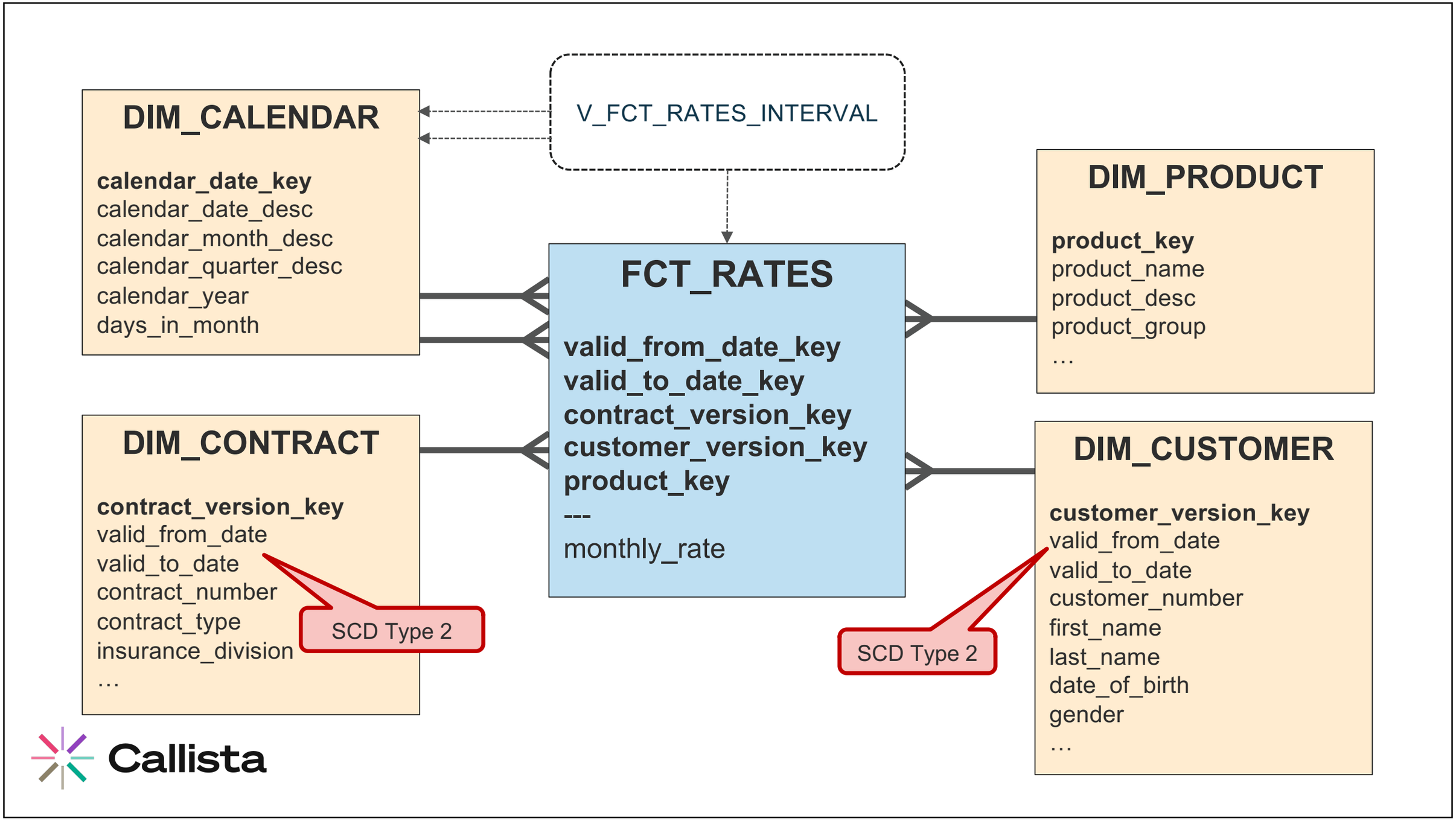
SQL | All Rows Fetched: 3 in 0,006 seconds

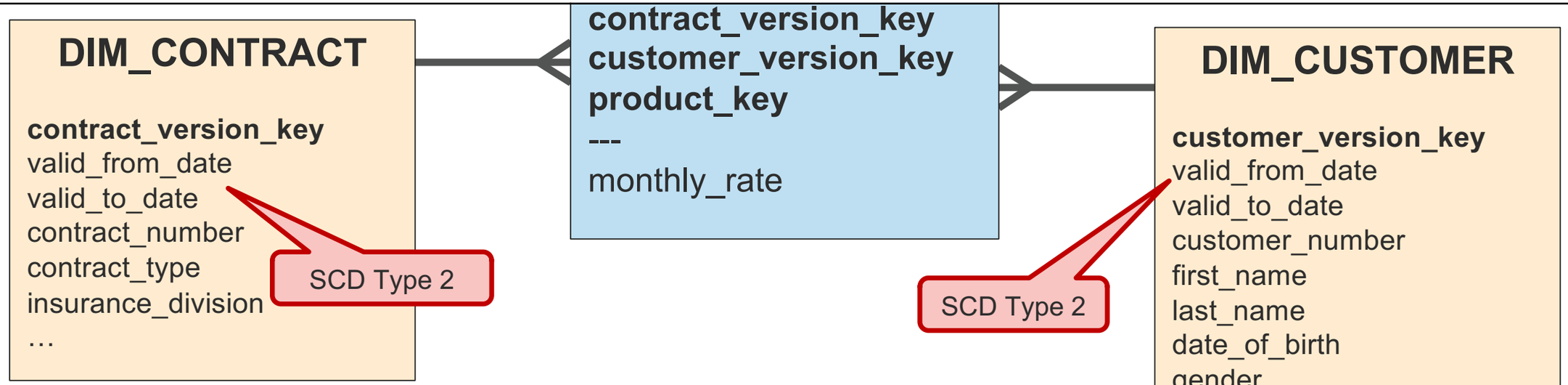
	VALID_FROM_DATE_KEY	VALID_TO_DATE_KEY	MONTHLY_RATE	INTERVAL_FROM_DATE	INTERVAL_TO_DATE	INTERVAL_RATE
1	16.11.2022	31.12.2022	40	25.11.2022	31.12.2022	48
2	01.01.2023	14.02.2023	42	01.01.2023	14.02.2023	63
3	15.02.2023	31.12.2023	36	15.02.2023	10.04.2023	66

Callista

Wir haben noch ein anderes Problem.

Callista


V_FCT_RATES_INTERVAL
DIM_CALENDAR
calendar_date_key
calendar_date_desc
calendar_month_desc
calendar_quarter_desc
calendar_year
days_in_month
FCT_RATES
valid_from_date_key
valid_to_date_key
contract_version_key
customer_version_key
product_key
---
monthly_rate
DIM_PRODUCT
product_key
product_name
product_desc
product_group
…
DIM_CONTRACT
contract_version_key
valid_from_date
valid_to_date
contract_number
contract_type
insurance_division
…
SCD Type 2
DIM_CUSTOMER
customer_version_key
valid_from_date
valid_to_date
customer_number
first_name
last_name
date_of_birth
gender
…
SCD Type 2
Callista



VERS_KEY	VALID_FROM_DATE	VALID_TO_DATE	CONTRACT_NUMBER
111	01.01.2022	15.11.2022	4711
112	16.11.2022	31.12.2022	4711
113	01.01.2023	14.02.2023	4711
114	15.02.2023	31.12.2023	4711

VERS_KEY	VALID_FROM_DATE	VALID_TO_DATE	CUSTOMER_NUMBER
221	27.06.2020	08.05.2022	0815
222	09.05.2022	20.07.2023	0815
223	21.07.2023	31.12.2999	0815

Callista

VALID_FROM _DATE	VALID_TO _DATE	CONTRACT _VERSION	CUSTOMER _VERSION	PRODUCT _KEY	MONTHLY _RATE
01.01.2022	08.05.2022	111	221	9876	40
09.05.2022	15.11.2022	111	222	9876	40
16.11.2022	31.12.2022	112	222	9876	40
01.01.2023	14.02.2023	113	222	9876	42
15.02.2023	20.07.2023	114	222	9876	36
21.07.2023	31.12.2023	114	223	9876	36

VERS _KEY	VALID_FROM _DATE	VALID_TO _DATE	CONTRACT _NUMBER
111	01.01.2022	15.11.2022	4711
112	16.11.2022	31.12.2022	4711
113	01.01.2023	14.02.2023	4711
114	15.02.2023	31.12.2023	4711

VERS _KEY	VALID_FROM _DATE	VALID_TO _DATE	CUSTOMER _NUMBER
221	27.06.2020	08.05.2022	0815
222	09.05.2022	20.07.2023	0815
223	21.07.2023	31.12.2999	0815

Callista

```
INSERT INTO data_mart.fct_rates
WITH valid_dates AS (
 SELECT cust_nr_fk AS cnum, valid_from_date FROM core_dwh.v_cor_contract_cur
 UNION
 SELECT cust_nr_fk AS cnum, valid_to_date+1 FROM core_dwh.v_cor_contract_cur
 UNION
 SELECT cust_nr_pk AS cnum, valid_from_date FROM core_dwh.v_cor_customer_cur
 UNION
 SELECT cust_nr_pk AS cnum, valid_to_date+1 FROM core_dwh.v_cor_customer_cur
)
SELECT vd.valid_from_date AS valid_from_date_key
     , LEAD(vd.valid_from_date) OVER
          (PARTITION BY vd.cnum ORDER BY vd.valid_from_date)
       AS valid_to_date_key
     , cont.version_key AS contract_version_key
     , cust.version_key AS customer_version_key
     , cont.product_nr_fk AS product_key
     , cont.monthly_rate AS monthly_rate
  FROM valid_dates vd
  JOIN core_dwh.v_cor_contract_cur cont ON (vd.cnum = cont.cust_nr_fk
       AND vd.valid_from_date BETWEEN cont.valid_from_date AND cont.valid_to_date)
  JOIN core_dwh.v_cor_customer_cur cust ON (vd.cnum = cust.cust_nr_pk
       AND vd.valid_from_date BETWEEN cust.valid_from_date AND cust.valid_to_date)
```

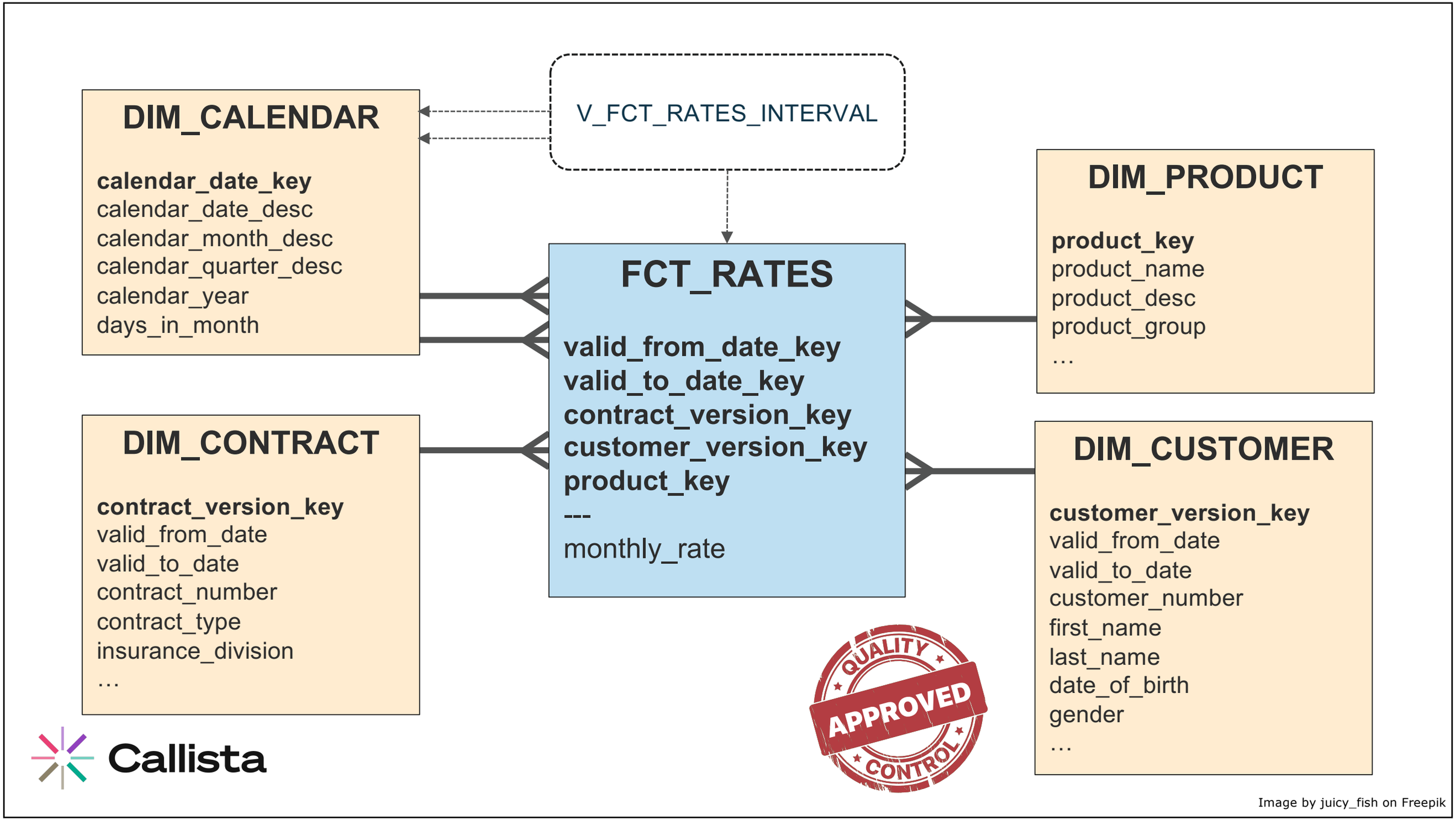
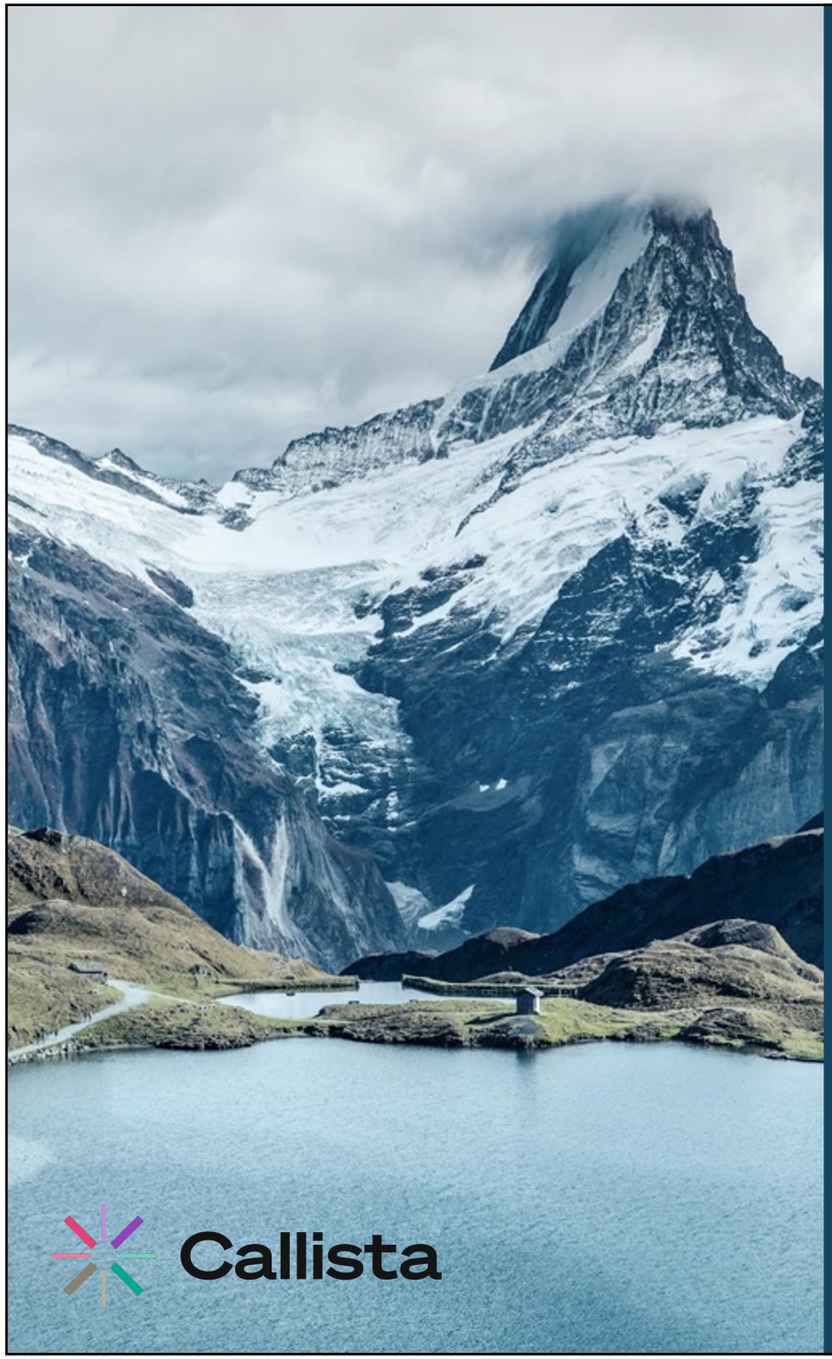
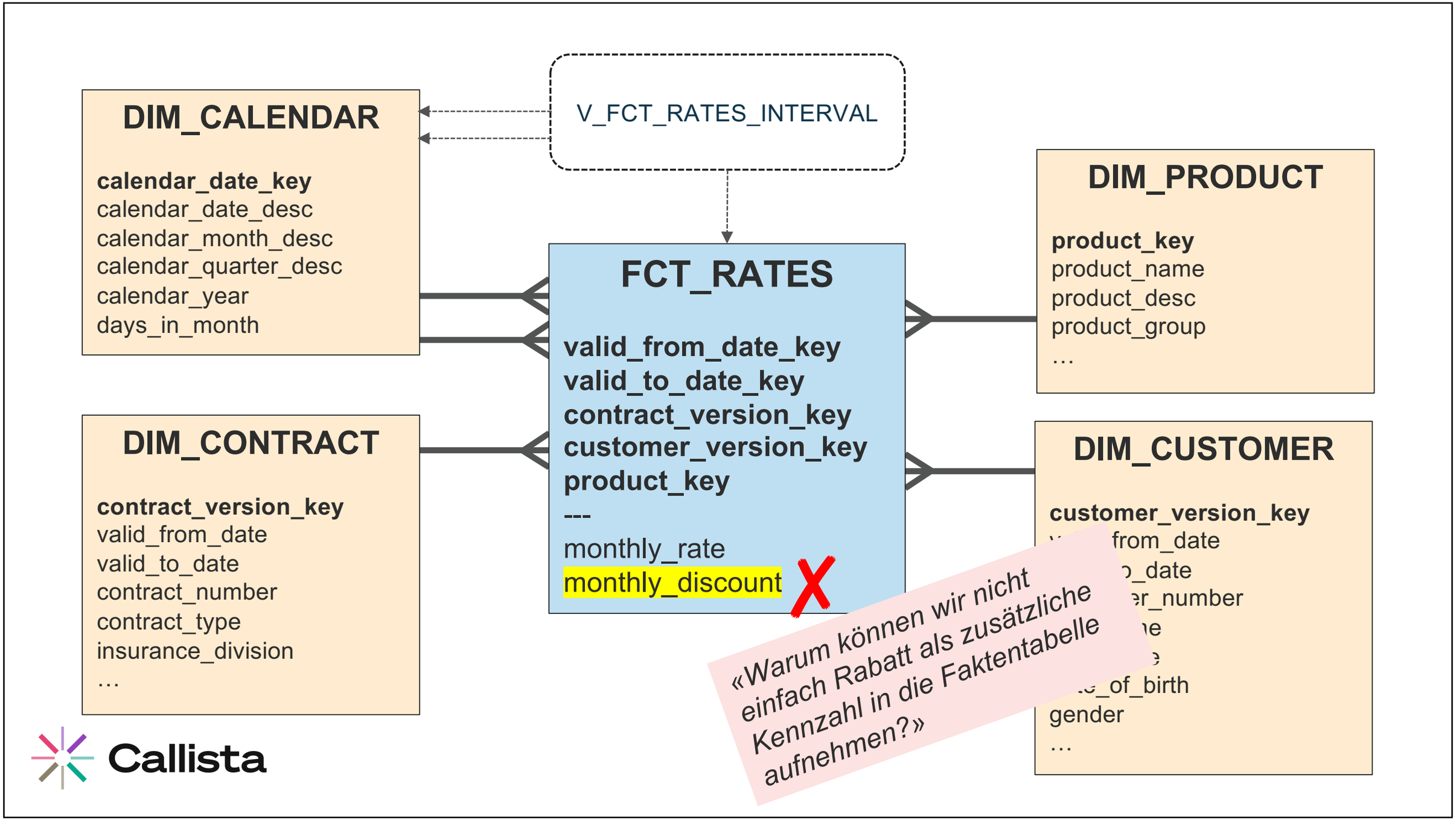


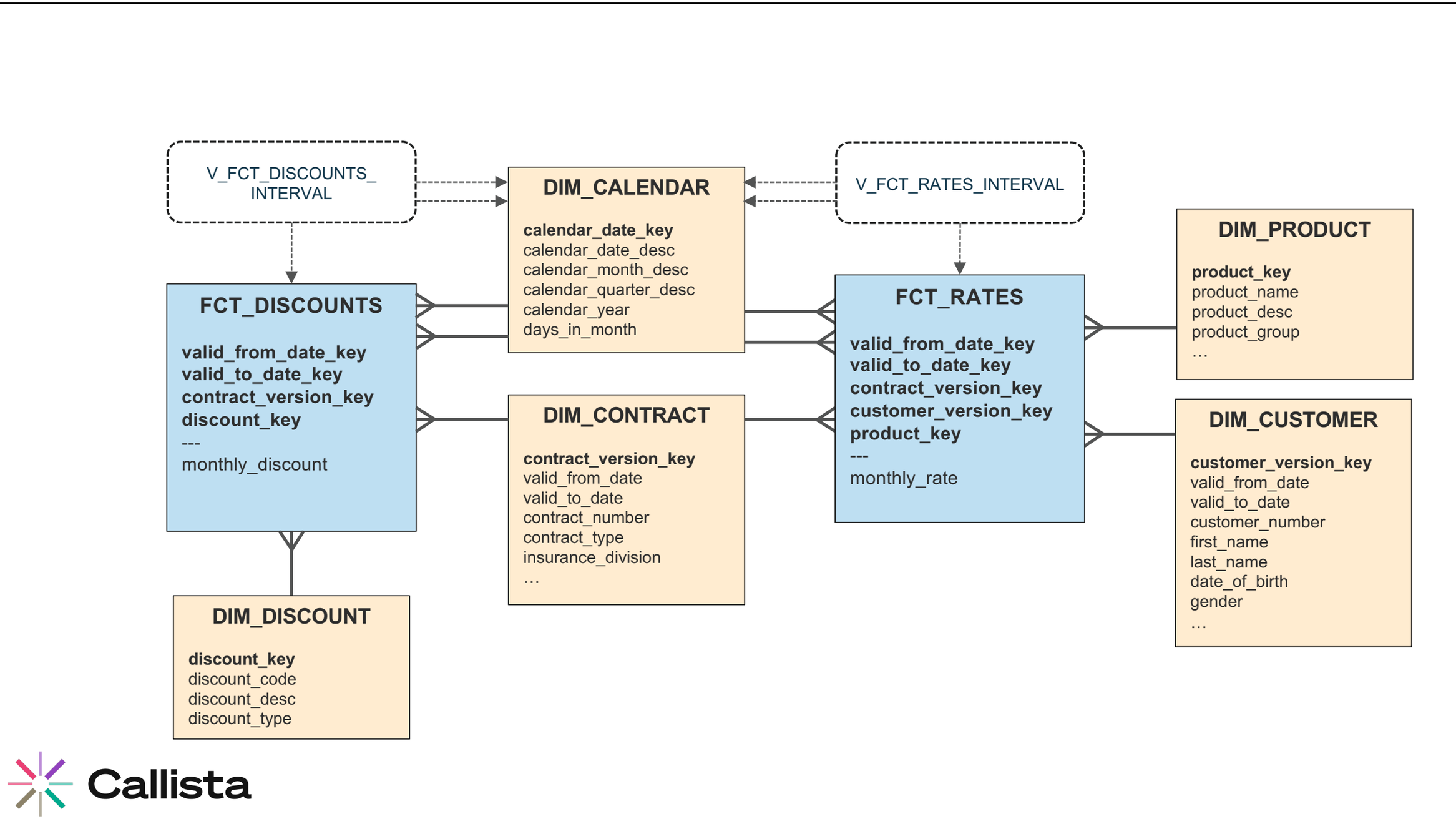
Image by juicy_fish on Freepik



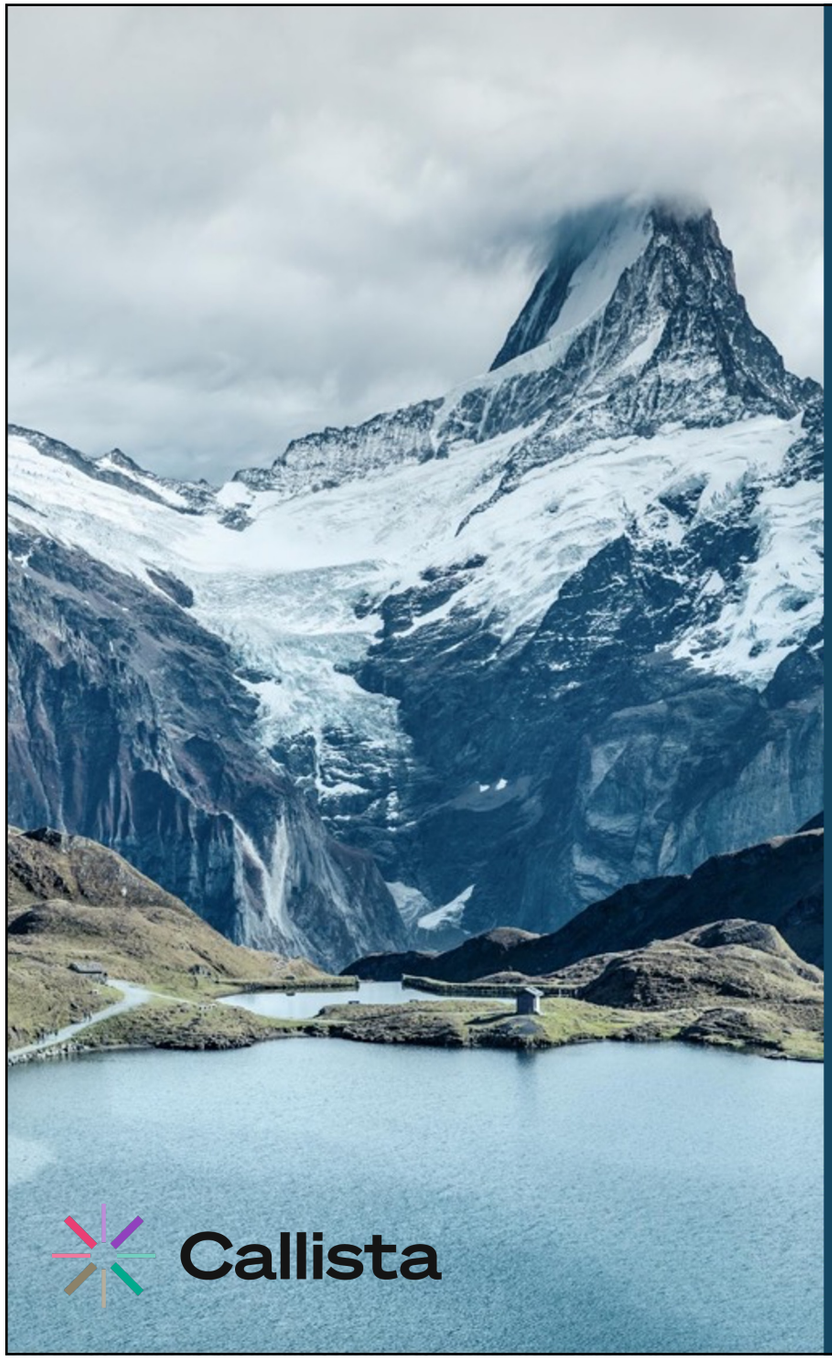
Zusätzliche Anforderungen

- Rabatte müssen in den Auswertungen auch berücksichtigt werden
- Pro Vertrag können unterschiedliche Rabatte vorkommen (z.B. Familienrabatt, Mitarbeiterrabatt) – auch mehrere gleichzeitig
- Aus Rabatte sollen auf Tagesebene ausgewertet werden können (oder über beliebige Zeitperiode)


V_FCT_RATES_INTERVAL
DIM_CALENDAR
calendar_date_key
calendar_date_desc
calendar_month_desc
calendar_quarter_desc
calendar_year
days_in_month
FCT_RATES
valid_from_date_key
valid_to_date_key
contract_version_key
customer_version_key
product_key
---
monthly_rate
monthly_discount
DIM_PRODUCT
product_key
product_name
product_desc
product_group
…
DIM_CONTRACT
contract_version_key
valid_from_date
valid_to_date
contract_number
contract_type
insurance_division
…
DIM_CUSTOMER
customer_version_key
from_date
_date
er_number
e
e
_of_birth
gender
…
«Warum können wir nicht einfach Rabatt als zusätzliche Kennzahl in die Faktentabelle aufnehmen?»
Callista

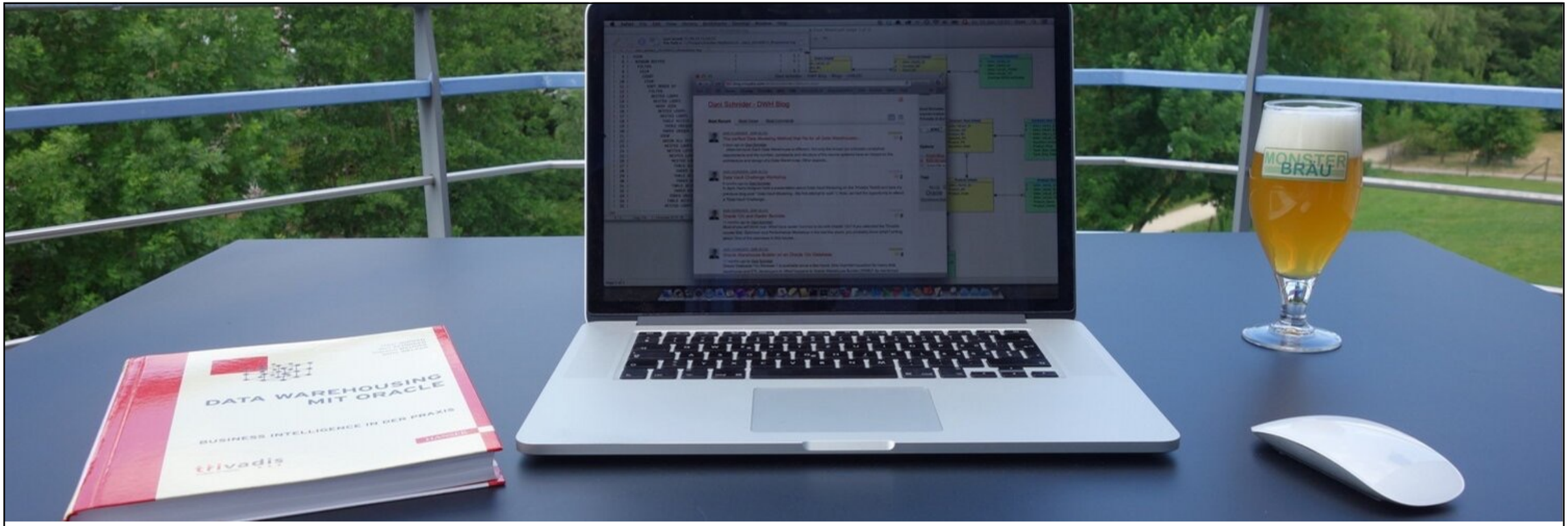

V_FCT_DISCOUNTS_
INTERVAL
DIM_CALENDAR
calendar_date_key
calendar_date_desc
calendar_month_desc
calendar_quarter_desc
calendar_year
days_in_month
V_FCT_RATES_INTERVAL
DIM_PRODUCT
product_key
product_name
product_desc
product_group
…
FCT_DISCOUNTS
valid_from_date_key
valid_to_date_key
contract_version_key
discount_key
---
monthly_discount
DIM_CONTRACT
contract_version_key
valid_from_date
valid_to_date
contract_number
contract_type
insurance_division
…
FCT_RATES
valid_from_date_key
valid_to_date_key
contract_version_key
customer_version_key
product_key
---
monthly_rate
DIM_CUSTOMER
customer_version_key
valid_from_date
valid_to_date
customer_number
first_name
last_name
date_of_birth
gender
…
DIM_DISCOUNT
discount_key
discount_code
discount_desc
discount_type
Callista

```
SELECT NVL(star1.insurance_division, star2.insurance_division) insurance_division
     , SUM(interval_rate) interval_rate
     , SUM(interval_discount) interval_discount
  FROM
(
   SELECT cont.insurance_division
        , SUM(fct.interval_rate)
     FROM v_fct_rates_interval fct
     JOIN dim_calendar cal
       ON cal.calendar_date_key = fct.valid_FROM_date_key
     JOIN dim_contract cont
       ON cont.contract_version_key = fct.contract_version_key
    WHERE fct.input_FROM_date = DATE'2023-01-01'
      AND fct.input_to_date = DATE'2023-12-31'
   GROUP BY cont.insurance_division
) star1
FULL OUTER JOIN
(
   SELECT cont.insurance_division
        , SUM(fct.interval_discount)
     FROM v_fct_discounts_interval fct
     JOIN dim_calendar cal
       ON cal.calendar_date_key = fct.valid_FROM_date_key
     JOIN dim_contract cont
       ON cont.contract_version_key = fct.contract_version_key
    WHERE fct.input_FROM_date = DATE'2023-01-01'
      AND fct.input_to_date = DATE'2023-12-31'
   GROUP BY cont.insurance_division
) star2
ON (star1.contract_version_key = star2.contract_version_key)
GROUP BY 1
ORDER BY 1
```



Fazit

- Star Schemas sind nicht immer einfach und übersichtlich
- Zeitintervalle (von-bis) in Faktentabellen führen zu komplexen Abfragen
- Slowly Changing Dimensions Typ 2 müssen speziell behandelt werden
- All dieses Probleme sind lösbar – und spannend ☺



https://danischnider.wordpress.com

https://twitter.com/dani_schnider

https://www.linkedin.com/in/danischnider/

Q&A


DRIVEN BY EXCELLENCE
Callista