ADMT 2018 - Project report

Group 02: Andreas Vieider (13177) & Laurin Stricker (13412)

January 9, 2019

Contents

List of Figures

List of Tables

1 Introduction

The domain of our fictional company is the one of furniture production and retail. The company is located in the province of Bolzano and has several showrooms in the area and one production center.

1.1 Business processes

1.1.1 CRM - Showroom visit

One CRM process is the collection of data about visitors at the different showrooms. A visitor can either be one who is just looking around without intention of buying anything (Seeleute), a future potential customer or an already existing customer. A visit can lead to an order.

Business questions:

- Which is the best running showroom (most visitors, most orders, etc.)
- Where are the customers from (with different granularity)
- Which department are the customers the most interested in
- Compare the number of visitors to the number of customers for a time period and/or showroom

1.1.2 Production

The company logs every step in the production process, especially duration, defects and machine failures.

Business questions:

- What is the average time to produce a particular product
- Which is the product with the highest/lowest quality
- How much does a product cost in terms of raw material cost

2 Conceptual Design

The first fact of our Data Warehouse represents a showroom visit. The company is registering each visit in a particular showroom and is interested in some very specific details about a the visit. Namely, for each visit they store the date, the visitor and visitor type, the showroom, the department in which the visitor was particularly interested, the order if the visitor placed one, the sales representative who took care about the visitor and the duration and the number of people with respect to the visit.

The second fact collects some relevant information of a production stage. For each production stage of a particular product, in addition to those two information, also start-

and end-date, the machine, the result of the quality control, the operator, the costs of the raw material and the duration of the process are stored.

Table 1: Fact table

Fact	Dimensions	Measures
Showroom visit	Date, Showroom, Visitor, Visitor type, Order, Department, Sales representative	Duration (AVG - additive), Amount of people (SUM - additive, AVG - additive)
Production	Start Date, End date, Product, Production Stage, Machine, Quality control, Operator	Duration (AVG), Raw material cost (SUM - semi-additive; AVG - semi-additive)

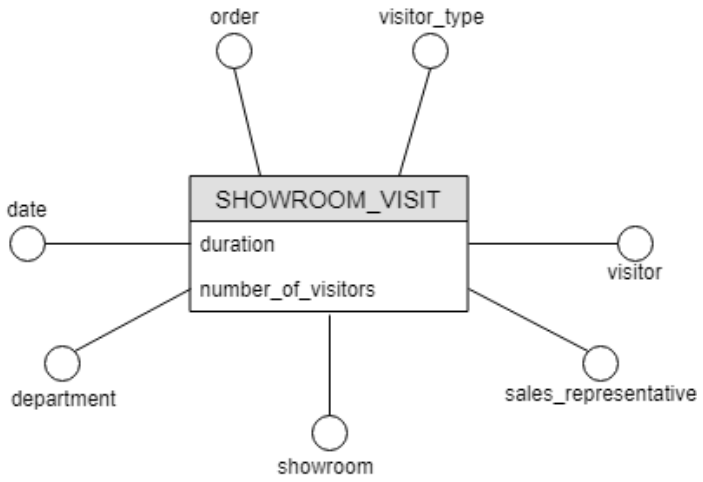

Figure 1: DFM of the showroom visit

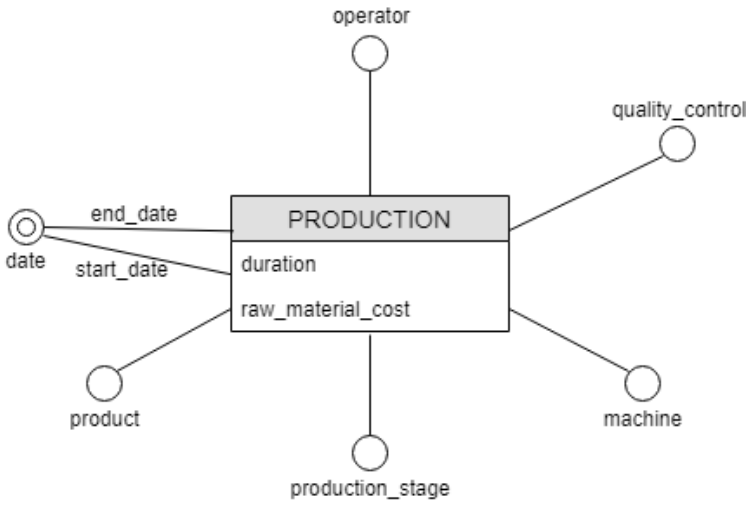

Figure 2: DFM of the production

2.1 Showroom visit

Table 2: Fact table

Dimension	Attributes
Date	Day, Month, Year, Quartal, Week, Day of Week, Season, Holiday
Showroom	Name, City, District, Province, Region, Country, Manager, Address, Telephone, Size
Visitor	Name, City, District, Province, Region, Country, Language, Telephone, E-Mail, Type, Sector, Gender, Customer number
Order	Order Number, Total Price, Discount
Order Detail	Quantity, Quantity Type, Product, Unit price, Total price
Department	Name
Sales representative	Name, City, District, Province, Region, Country, Language, Telephone, E-Mail, Gender

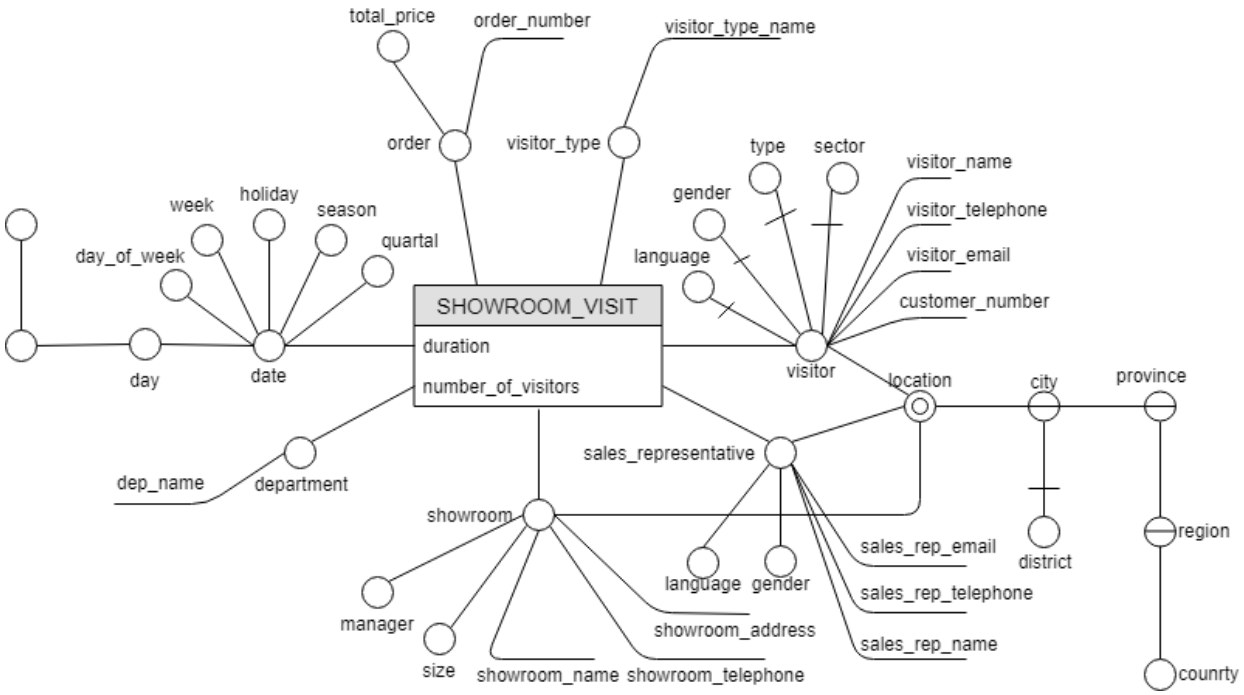

Figure 3: Dimension fact model (DFM) of the showroom visit with attributes

2.2 Production

Table 3: Fact table

Dimension	Attributes
Start date	Day, Month, Year, Week
End date	Day, Month, Year, Week
Product	Product number, Name, Department, Category
Production stage	Name
Machine	Name, Purchasing year, Vendor
Quality control	Grade
Operator	Name

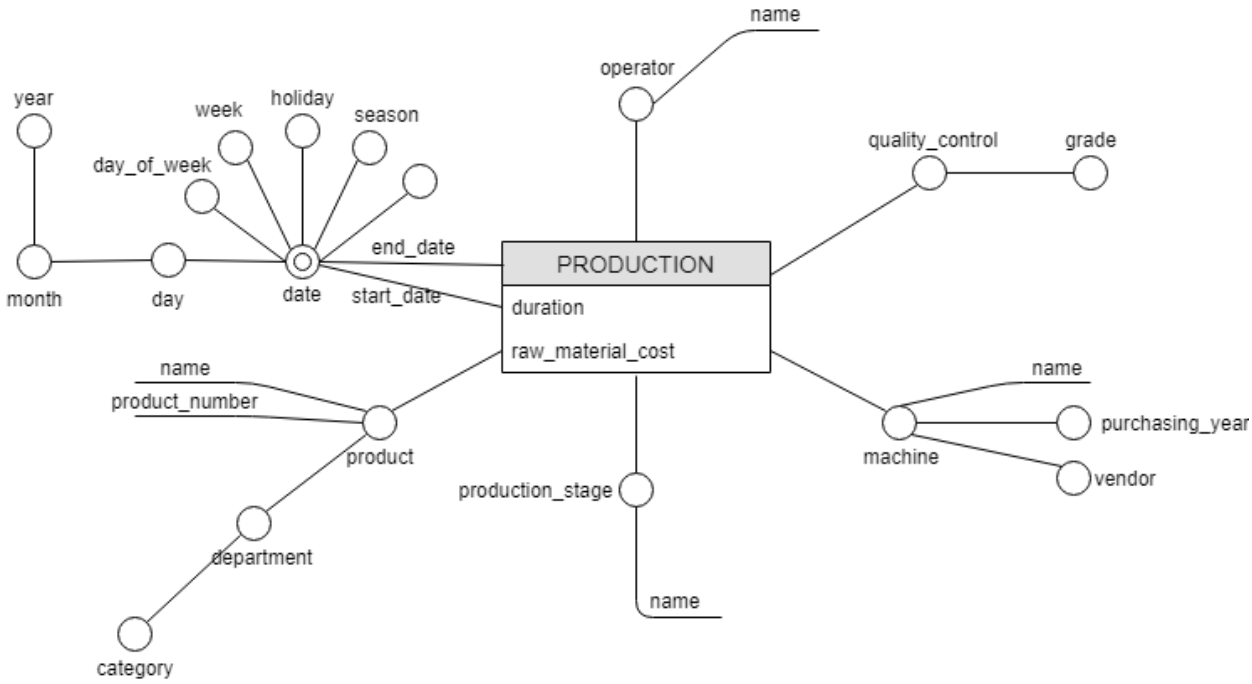

Figure 4: Dimension fact model (DFM) of the production with attributes

3 Logical Design

3.1 Star schemas

The following star schema fig. 5 represent the first business process, namely the showroom visit.

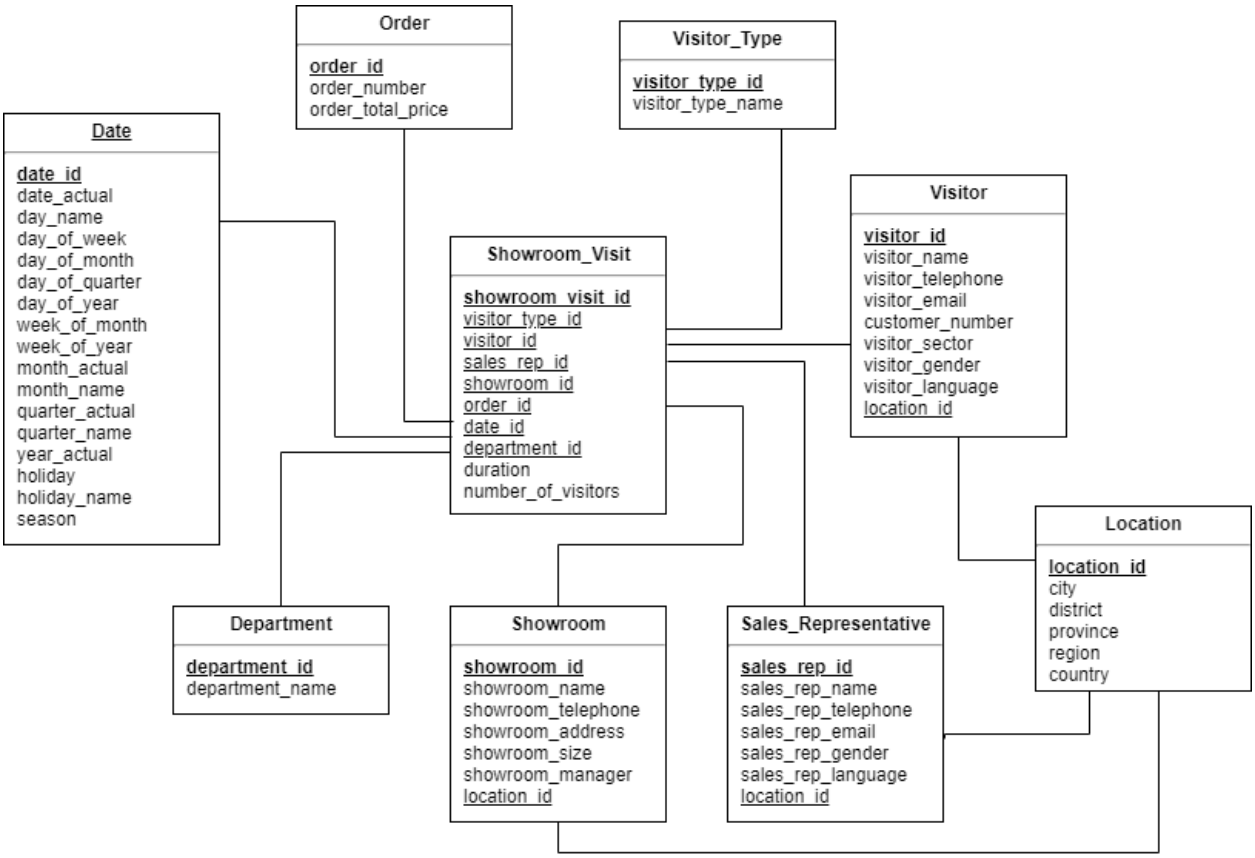

Figure 5: Star schema of the showroom visit

Instead, the star schema fig. 6 represents the production business process.

3.2 Two business questions

3.2.1 Fact: Showroom visit

In order to be able to make the right marketing decisions, it is very important for the management to know from which sector the various customers or interested parties of a particular showroom come from. So, for example the management wants to know, from which sectors the various customers of showroom "Showroom-Bozen" were coming in the last year.

SQL query:

```
SELECT v.visitor_sector, count(*)
FROM warehouse.visitor v
INNER JOIN warehouse.showroom_visit sv on v.visitor_id = sv.visitor_id
INNER JOIN warehouse.showroom s on sv.showroom_id = s.showroom_id
INNER JOIN warehouse.date d on sv.date_id = d.date_id
WHERE s.showroom_name = 'Showroom-BOZEN'
```

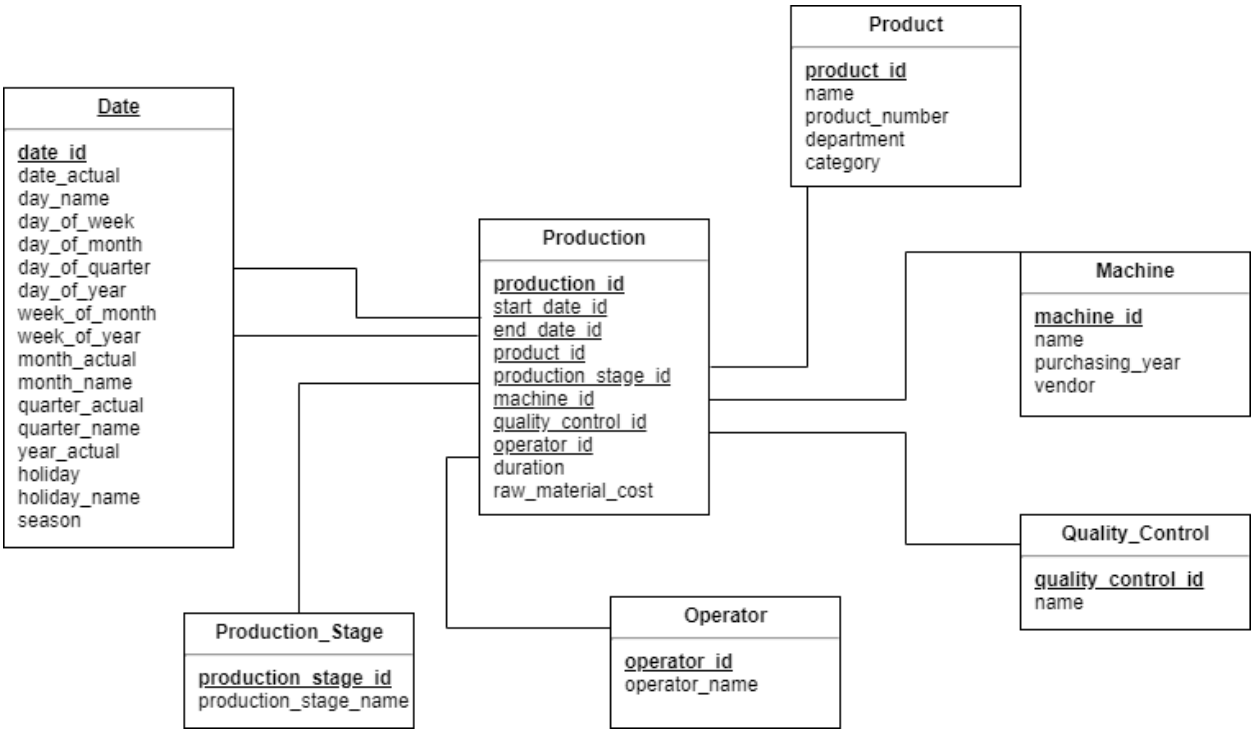

Figure 6: Star schema of the production

```
AND d.date_actual >= '2018-01-01' AND d.date_actual <= '2018-12-31'
GROUP by v.visitor_sector
```

Table 4: Showroom visit

ID	Visitor_id	Sales_rep_id	Showr._id	Depart._id	Date_id	Type_id	Duration	Nr._of_visit.
1282369	570822	6	5	4	20180323	2	90	2
1282370	570823	5	5	2	20160107	4	167	4
1282371	570823	7	5	1	20130526	3	173	6
1282372	570823	11	5	6	20150806	3	100	10
1282373	570823	7	5	1	20121116	4	169	5
1282374	570824	7	5	1	20171210	3	57	3
1282375	570824	18	5	2	20110212	3	166	7
1282376	570824	9	5	4	20130811	3	84	5
1282377	570825	11	5	6	20170507	3	184	10
1282378	570825	12	5	2	20111127	2	26	2
1282379	570825	7	5	1	20150425	3	141	10
1282380	570826	11	5	6	20130208	2	8	2
1282381	570826	12	5	1	20111214	3	61	8

ID	Visitor_id	Sales_rep_id	Showr._id	Depart._id	Date_id	Type_id	Duration	Nr._of_visit.
1282382	570827	12	5	1	20170202	3	139	9
1282383	570827	12	5	2	20121012	3	71	7

Table 5: Visitor

ID	Name	Telephone	E-Mail	Sector	Sex	Lang.	Loc._id
570822	Melanie Eder			Gastronomy	F	german	9
570823	Julian Schmidt		j.schmidt@email.com	Private	M	german	9
570824	Marcel Schwarz	306 9579783	m.schwarz@email.com	Hotel	M	german	9
570825	Denise Fuchs	396 5305260	d.fuchs@email.com	Public	F	german	9
570826	Sophie Wimmer	322 7641804	s.wimmer@email.com	Private	F	german	9

Table 6: Showroom

ID	Name	Telephone	Address	Size	Manager	Loc._id
1	Showroom-LATSCH	0477 069655	Herrengasse 8	581	Paul Wolf	42
2	Showroom-MÜHLBACH	0474 039227	Platzerstr. 58	349	Christoph Steiner	54
3	Showroom-MÖLTEN	0470 429676	Vernag 97	857	Christoph Steiner	51
4	Showroom-SALURN	0475 248487	Gewerbezone 44	198	Johannes Egger	77
5	Showroom-BOZEN	0473 723301	St. Urban 73	447	Sabine Schneider	9

Table 7: Date

ID	Date	Day_week	Day	Month	Quartal	Year	Holiday	Season
20160102	2010-01-02	6	Saturday	January	First	2016	false	Winter
20170103	2010-01-03	7	Sunday	January	First	2017	false	Winter
20180108	2018-01-08	5	Friday	January	First	2018	false	Winter
20190109	2010-01-09	6	Saturday	January	First	2019	false	Winter
20200110	2010-01-10	7	Sunday	January	First	2020	false	Winter

Table 8: Result of the query

Sector	Number of visitors
Gastronomy	2985
Hotel	4223
Private	5629
Public	1371

3.2.2 Fact: Production

The company's quality control is always interested in optimizing processes. It is therefore interesting for employees to know whether a machine has significant time differences in production in relation to a particular product in comparison to the other machines.

SQL query:

```
SELECT m.machine_name, avg(p.duration) AS avg_production_duration
FROM warehouse.machine m
INNER JOIN warehouse.production p ON m.machine_id = p.machine_id
INNER JOIN warehouse.product o ON p.product_id = o.product_id
WHERE o.product_number = 'Warteraum-Couch␣-␣10'
GROUP BY m.machine_id
ORDER BY avg_production_duration DESC LIMIT 10
```

Table 9: Production

ID	Operator*	Machine*	Stage*	Product*	Start_date*	End_date*	Duration	Raw_mat._cost
591814	779	1144	1	361016	20101105	20101202	152	76
591815	780	1174	2	361016	20101202	20101203	1	395
591816	775	1213	3	361016	20101203	20101207	2	277
591817	770	1055	1	361016	20101122	20101214	30	66
591818	722	1176	2	361016	20101214	20110111	133	391
591819	755	1079	3	361016	20110111	20110204	36	275
591820	740	1069	1	361016	20150511	20150520	49	73
591821	756	1025	2	361016	20150520	20150603	54	398
591822	758	1130	3	361016	20150603	20150625	96	278
27064	754	1164	1	361016	20101022	20101026	8	66
27065	739	1028	2	361016	20101026	20101104	6	407
27066	798	1098	3	361016	20101104	20101105	6	280
27067	780	1013	1	361016	20130327	20130411	70	74
27068	737	1145	2	361016	20130411	20130509	18	404

ID	Operator*	Machine*	Stage*	Product*	Start_date*	End_date*	Duration	Raw_mat._cost
27069	772	1032	3	361016	20130509	20130520	14	281

Note: all columns with the * are foreign key columns and are carrying only the id

Table 10: Machine

ID	Machine_name	Machine_vendor	Purchasing_year
1172	Melichár	Durán	1998
1173	Horn	Lóntos	2009
1174	Chihaia	Murtazaev	2002
1175	Korčák	Durán	2006
1176	Ramóna	Barbora	1996

Table 11: Product

ID	Product_name	Product_number	Product_department	Product_category
361013	Warteraum-Couch	Warteraum-Couch - 7	Büro	Arztpraxis-Set
361014	Warteraum-Couch	Warteraum-Couch - 8	Büro	Arztpraxis-Set
361015	Warteraum-Couch	Warteraum-Couch - 9	Büro	Arztpraxis-Set
361016	Warteraum-Couch	Warteraum-Couch - 10	Büro	Arztpraxis-Set
361017	Warteraum-Couch	Warteraum-Couch - 11	Büro	Arztpraxis-Set

Table 12: Result of the query

Machine_name	AVG_Production_duration
Vajda	152.00
Ramóna	133.00
Papandreou	96.00
Kontoléon	70.00
Mitu	54.00
Bercu	49.00
Heinrich	36.00

Machine_name	AVG_Production_duration
Martinez	30.00
Pál	18.00
Aguilar	14.00

4 Implementation

4.1 ROLLUP

4.1.1 SQL query using ROLLUP for business process 1 (showroom visit)

The following sql query shows the number of visitors per showroom, in the different areas and in the different seasons. In addition there are the different partial sums. For example, for the showroom in Bolzano, first the number of visitors for the 'bBedroom' area in autumn is shown, then the total number of visitors for the 'bedroom' area, regardless of the season, and finally the total number of visitors for the showroom in Bolzano, regardless of the area and the season.

```
1 SELECT showroom_name, department_name, season, count(visitor_id)
2 FROM warehouse.showroom_visit
3 JOIN warehouse.showroom using (showroom_id)
4 JOIN warehouse.department using (department_id)
5 JOIN warehouse.date using (date_id)
6 GROUP BY ROLLUP(showroom_name, department_name, season);
```

4.1.2 SQL query with ROLLUP for business process 2 (production)

The following sql query shows the average machining time for a particular production stage of a particular product of a particular product category. The query also returns the average machining times of the higher levels, in other words, a granularity is removed step by step. For example, the average machining time of 'table XY' is shown first for the 'fine grinding' process. Then you get the average machining time of all processes on 'table XY' and finally the average machining time of all processes on all table models, thus of the whole product category 'table'.

```
1 SELECT product_category, product_name,
2         production_stage_name, ROUND(avg(duration)::numeric,2)
3 FROM warehouse.production
4 JOIN warehouse.product using (product_id)
5 JOIN warehouse.production_stage using (production_stage_id)
6 GROUP BY ROLLUP(product_category, product_name, production_stage_name);
```

4.2 CUBE

4.2.1 SQL query using CUBE for business process 1 (showroom visit)

The following query shows the number of visitors from the province of Bolzano and its commercial sector in the different districts of the showrooms. In addition, the query shows all possible sub-totals, removing step by step different granularities. In other words, for each combination of values, the sum is shown, finally the total sum of all visits from visitors from the province of Bolzano.

```
SELECT visitor_sector, vl.district, sl.district,
        sum(number_of_visitors)
FROM warehouse.showroom_visit
JOIN warehouse.visitor using (visitor_id)
JOIN warehouse.location as vl
        on warehouse.visitor.location_id = vl.location_id
JOIN warehouse.showroom using (showroom_id)
JOIN warehouse.location as sl
        on warehouse.showroom.location_id = sl.location_id
WHERE vl.province = 'Bozen'
GROUP BY CUBE(vl.district, visitor_sector, sl.district)
ORDER BY visitor_sector, vl.district, sl.district;
```

4.2.2 SQL query using CUBE for business process 2 (production)

The following query shows the average grade of the quality control for a machine and for the product category. Also all partial average values of all different combinations and groupings can be read off.

```
SELECT product_department, machine_name,
        ROUND(avg(quality_control_grade)::numeric,2)
FROM warehouse.production
JOIN warehouse.product using (product_id)
JOIN warehouse.machine using (machine_id)
JOIN warehouse.quality_control using (quality_control_id)
WHERE quality_control_grade is not NULL
GROUP BY CUBE(product_department, machine_name)
ORDER BY avg(quality_control_grade) desc;
```

4.3 GROUPING SETS

4.3.1 SQL query using GROUPING SETS for business process 1 (showroom visit)

The following query shows the number of visitors per language served by a sales representative in a showroom. Also the total number of visitors can be taken from a language in that showroom as well as the total number of visitors served by that sales representative.

```
SELECT showroom_name, sales_rep_name,
        visitor_language, sum(order_total_price)
FROM warehouse.showroom_visit
JOIN warehouse.visitor using (visitor_id)
JOIN warehouse.sales_representative using (sales_rep_id)
JOIN warehouse.order using (order_id)
JOIN warehouse.showroom using (showroom_id)
GROUP BY GROUPING SETS(
        (showroom_name, sales_rep_name, visitor_language),
        (showroom_name, visitor_language),
        (showroom_name, sales_rep_name));
```

4.3.2 SQL query using GROUPING SETS for business process 2 (production)

The following query shows the number of a certain grade for a product category in a specific year. The query also shows the number of a certain rating in a certain year.

```
SELECT product_category, year_actual,
        quality_control_grade, count(product_id)
FROM warehouse.production
JOIN warehouse.product using (product_id)
JOIN warehouse.date ON date.date_id = production.end_date_id
JOIN warehouse.quality_control using (quality_control_id)
GROUP BY GROUPING SETS(
        (product_category, year_actual, quality_control_grade),
        (year_actual, quality_control_grade));
```

5 Querying

5.1 NTILE

5.1.1 SQL query using NTILE for business process 1 (showroom visit)

The following sql statement calculates the number of visitors coming from a particular location of the province of Bolzano and assigns each row to a group from 1-4, depending on the size of the number of visitors.

```
SELECT vl.city, count(visitor_id),
        NTILE(4) OVER (ORDER BY count(visitor_id)) AS TILE4
FROM warehouse.showroom_visit
JOIN warehouse.visitor using (visitor_id)
JOIN warehouse.location as vl
        on warehouse.visitor.location_id = vl.location_id
WHERE vl.province = 'Bozen'
GROUP BY vl.city;
```

5.1.2 SQL query using NTILE for business process 2 (production)

The next sql query sums all processing times of an operator and groups them to 4 groups, were each operators gets assigned to a specific group relatively to the sum of duration of all production steps.

```
SELECT operator_name, ROUND(sum(duration)::numeric,2),
        NTILE(4) OVER (ORDER BY sum(duration)) AS TILE4
FROM warehouse.production
JOIN warehouse.operator using (operator_id)
GROUP BY operator_name;
```

5.2 RANK

5.2.1 SQL query using RANK for business process 1 (showroom visit)

The following query identifies the overall total number of visitors per showroom and ranks them according to their number of visitors.

```
SELECT showroom_name, count(distinct visitor_id),
        RANK() OVER (ORDER BY count(distinct visitor_id) DESC)
FROM warehouse.showroom_visit
JOIN warehouse.showroom using (showroom_id)
GROUP BY showroom_name;
```

5.2.2 SQL query using RANK for business process 2 (production)

The following sql query ranks the different products with respect to their average raw material costs.

```
SELECT product_category, ROUND(avg(raw_material_cost)::numeric,2),
        RANK() OVER (ORDER BY (avg(raw_material_cost)) DESC)
FROM warehouse.production
JOIN warehouse.product using (product_id)
GROUP BY product_category;
```

5.3 ??????

5.3.1 SQL query ??????

(text missing)

```
SELECT date_actual, sum(order_total_price),
        ROUND(AVG(SUM(order_total_price))
                OVER ( ORDER BY date_actual
                ROWS BETWEEN 7 PRECEDING
                AND CURRENT ROW)::numeric,2)
FROM warehouse.showroom_visit
```

```
JOIN warehouse.date using (date_id)
JOIN warehouse.order using (order_id)
WHERE year_actual > 2017
GROUP BY date_actual
ORDER BY date_actual;
```

5.3.2 SQL query ??????

(text missing)

```
SELECT year_actual, month_actual, sum(raw_material_cost),
        ROUND(AVG(SUM(raw_material_cost))
                OVER ( ORDER BY year_actual, month_actual
                ROWS BETWEEN 6 PRECEDING
                AND CURRENT ROW)::numeric,2)
FROM warehouse.production
JOIN warehouse.date ON date.date_id = production.end_date_id
GROUP BY year_actual, month_actual
ORDER BY year_actual, month_actual;
```

5.4 ??????

5.4.1 SQL query ??????

(text missing)

```
SELECT year_actual, quarter_actual,
visitors_this_year, visitors_last_year,
visitors_this_year - visitors_last_year as difference
FROM (
        SELECT year_actual, quarter_actual,
                count(visitor_id) as visitors_this_year,
                LAG(count(visitor_id), 4)
                OVER (ORDER BY year_actual, quarter_actual)
                as visitors_last_year
        FROM warehouse.showroom_visit
                JOIN warehouse.date using (date_id)
                JOIN warehouse.order using (order_id)
                GROUP BY year_actual, quarter_actual
                ORDER BY year_actual, quarter_actual) as last_year
        WHERE year_actual > 2010;
```

6 Data Analysis Tool