Documentation of Data Option Data Science

Definition of the data

Text Broker provides authors to clients who are looking for text for websites, publications, etc. Text Broker has registered customers for this purpose. If a customer wants to have a text, he places an order and describes the text and the content he wants, including price ideas, and makes it available on the platform. An author can record, write and submit this song. If an order is completed, it will be used for statistics, ie it will appear in the records. The data sets contain only completed orders.

List of orders Fields and explanations:

title	**explanation**
Order ID	Order number (1 text order)
Client ID	Customer ID
Client segment	We distinguish between self and managed service. Self Service customers, who order themselves through the platform, manage customer book project management through us and we keep their accounts for them. That's why I took out Managed Service customers.
Client Zip_code	Postcode customer
Client City	City customer
Client country	Country customer
Client Register Date	Date registration customer
Author ID	Author ID
Author ZIP code	Postal Author
Author City	City author
Author Country	Country author
Author Register Date	Date registration author
Date	Setting date of the order (order = a text order) , ie the order is completed and appears here
Order Title	Title text order
Category	Category text order
Written words	Actual number of written words
Max Words	Maximum number of words (which is the maximum number of words to be paid -> If the author writes less, he will get the written words , the author writes more, he gets the max words remunerated)
gain Author	Price per word in euros, which the author receives for this order, eg 0.0095 = 0.95 cents per word
Classification	We offer different order types (OpenOrder3,4 or 5, DirectOrder = 10, TeamOrder = 20). In the OpenOrder , the author wins the quickest and it is the star ranking of the authors, 3 star authors can write 3 star OpenOrders , 4 star authors 3 and 4 star OOs, 5 star authors everything. There is a fixed word price per star level; TeamOrders -> here the customer sets the word price and invites authors to his team or authors apply, DirectOrders -> word price is negotiated between author and customer, an author is assigned directly, DO and TO are star level independent
rating	Should be the star rating of the author, star ratings are from 1 to 5 (see above explanation at classification), 2 to 5 stars authors can write texts on the platform, but we have changed our way to rates 2016, before was the rating order based , for these orders there is still a rating z. 1

	and 5 in this table, since mid-2016 is based as rating rating author, desalb only the value -1 is displayed, you can read the rating no longer order-based.
system	Textbroker- Platform (DE = Germany, NL = Netherlands etc.) Customers are usually in one platform, but there are customers who are traveling in multiple platforms.
Client First Deposit	First deposit Customer on account (prepaid system) Is the same for all orders with the same customer
Client load deposit	Last deposit customer on account) Is the same for all orders with the same customer
Client First Order Closed	First completed order Customer -> the first text the customer has accepted (order accepted means author will be remunerated, blocked credit will be deducted by the customer, text is purchased and completed) . Is identical for all orders with the same customer
Client Last Order Closed	Last completed order Customer -> last accepted text by the customer . Is identical for all orders with the same customer
Revision Request	Request for revision -> Change request of the customer to the author. Simply yes or no. Yes, it can also mean that the customer has repeatedly returned the text for modification
Days to write order	Period from order placement to fulfillment -> number of days from the order setting to the final delivery by the author (but not yet accepted by the customer) The customer has 3 days to accept the order or a change request after the author has submitted the order If he does not do anything, the order will automatically be accepted on the 4th day)
Days or pending before the 1st time	Number of days that an order placed by the customer is in the system until the first time an author selects it for writing / editing (does not mean that he actually writes or submits it = submitted)
Count picked by author	Number of times an order has been processed by possibly different authors. Every time an author either accepts an order begins the processing time, the customer can adjust (1 to 10 days) to run again at. Orders with lots of picks usually take longer to receive a finished text. One and the same author can cancel and accept a text up to 3 times.
Author last Rated	Author last review ->, on this date the author got his last rating from Textbroker

Files for further analysis

The previous test file has been extended and new files have been created:

file	**purpose**
ORDER_PREPARATION_01.csv	Contains a record for each customer with values for the customer or the summary of various values of the single order from the years 2016 and 2017. Two values from 2018 were calculated: on which the customer made business in 2018 and how high the customer's turnover is. File is also available on the MySQL server as a table
ORDER_PREPARATION_YEAR.csv	If this contains the same data as ORDER_PREPARATION_01 .csv , only here the calculated values are split up again into the years 2016 and 2017. File is also available on the MySQL server as a table

Association_category.csv	Contains the data for an association analysis of the categories of written texts summarized at customer level

ORDER_PREPARATION_01.csv

TABLE ORDER Preparation	Type	statement
CLIENT_ID	NUMBER	ID of the customer
CLIENT_SEGMENT	VARCHAR2 (10)	Self Serviced or managed customer
CLIENT_COUNTRY	VARCHAR2 (6)	Country of the customer
CLIENT_REGISTER_DATE	DATE,	Date of registration
CLIENT_FIRST_DEPOSIT	DATE,	First Loan
CLIENT_LAST_DEPOSIT	DATE,	Latest Loan
CLIENT_FIRST_ORDER_CLOSED	DATE,	Conclusion of the first order
CLIENT_LAST_ORDER_CLOSED	DATE,	Conclusion of the last order
YEARS_SINCE_REGISTRATION	NUMBER	years since the registration
YEARS_SINCE_FIRST_DEPOSIT	NUMBER	Years since the first deposit
YEARS_SINCE_FIRST_ORDER_ CLOSED	NUMBER	Years since the first graduation . Reference here is always 1.1.2018
SYSTEM_S	VARCHAR (2)	System in which the customer is traveling
COUNT_ORDER	NUMBER	Number of orders
COUNT_ORDER _CLASS_OPEN	NUMBER	Number of open orders (classification 2,3,4,5)
AVG_ORDER_CLASS_OPEN	NUMBER	Average of the open order
COUNT_ ORDER _CLASS_TEAM	NUMBER	Number of Team Order (Classification 20)
AVG_ORDER_CLASS_TEAM	NUMBER	Average of Team Order
COUNT_ORDER_CLASS_DIRECT	NUMBER	Number of direct orders (Classification 10)
AVG_ORDER_CLASS_DIRECT	NUMBER	Average of Direct Order
COUNT_AUTHOR	NUMBER	Number of different e ne authors who have worked for the customer
AVG_AUTHOR	NUMBER	Durchsc h nittliche number of the various authors who have worked reasons for the K.
SUM_WRITTEN_WORDS	NUMBER	Sum of the written words
AVG_WRITTEN_WORDS	NUMBER	Average of written words per job
SUM_GAIN_AUTHOR	NUMBER	Sum of the average price per word
AVG_GAIN_AUTHOR	NUMBER	Average of the average price per word
SUM_REVENUE	NUMBER	total Sales (gain * written_words)
AVG_REVENUE	NUMBER	Durchsc h Nitt sales (gain * written_words)
SUM_DAYS_TO_WRITE_ORDER	NUMBER	Sum of days from order fulfillment to completion
AVG_DAYS_TO_WRITE_ORDER	NUMBER	Durchsc h Nitt the days of de r order fulfillment to final
SUM_DAYS_ORDER_PENDING	NUMBER	Sum of the days until the first application is received
AVG_DAYS_ORDER_PENDING	NUMBER	Average of the days until the first application is received
SUM_COUNT_PICKED_BY_AUTHOR	NUMBER	Number of author changes
AVG_COUNT_PICKED_BY_AUTHOR	NUMBER	average Author exchange
COUNT _REVISION_REQUEST	NUMBER	Number of orders that had a revision request

AVG_REVISION_REQUEST	NUMBER	Average of the order that had a revision request
SUM_YEARS_AUTHOR_REG	NUMBER	Sum of years since the author is registered (relative to order date)
AVG_YEARS_AUTHOR_REG	NUMBER	Average of years since the author is registered (relative to order date)
ACTIVE_2018	NUMBER	1: Active in 2018, 0 not active in 2018
REVENUE_2018	NUMBER	Turnover in 2018, 0: no sales

All slightly green fields have been split in the table ORDER_PREPARATION_YEAR.csv in 2016 and 2017 .

Association_category.csv

The file contains all customers as rows and as columns the individual categories of the text order (Category). = in the column means the category was not booked, a value greater than 0 is the number of or in the Period category is 2016 - 2018

The Excel Spreadsheet Association_category.xlsx contains the data digest and performs the transposition.

The mapping of the column names to the categories is as follows:

Initial and continuing education	W1	Job & Career	W23
Cars & Transportation	W2	Arts & Crafts	W24
Beauty	W3	proofreading	W25
Business & Economy	W4	literature	W26
Computer & Software	W5	logistics	W27
electric	W6	marketing	W28
entertainment	W7	Media & Telecommunications	W29
Education & Family	W8	Fashion	W30
Eat Drink	W9	Music & Dance	W31
Events	W10	News & Current Affairs	W32
finances	W11	Not youth free (18+)	W33
Fitness & Lifestyle	W12	Product & Category Descriptions	W34
Garden & plants	W13	Law	W35
Hospitality	W14	Travel & Tourism	W36
History & Politics	W15	Jewelry & precious metals	W37
Health & Medicine	W16	Shopping	W38
craft	W17	miscellaneous	W39
House & Living	W18	Spiritual & Philosophy	W40
Hobbies	W19	Sports	W41
Horoscopes & Divination	W20	Languages & Translations (source language => German)	W42
real estate	W21	animals	W43
Internet	W22	Betting & Gambling	W44
		sciences	W45

A few examples of a data analysis:

The data should be analyzed in advance if possible. Here are a few examples of a data analysis.

In the Excel file **Analysis_Pivot_Excel.xlsx** is the example for a pivot .

1. Number of records: 1 . 412 . 627
2. Number of customers who placed an order: 24.301
3. Customer sharing:
   managed 243,466 orders from 807 customers
   Weekly 1 . 169 . 161 orders from 23494 customers.
4. min Max Client Register Date
   01.01.70 8:02:19
5. 01/01/1970 49. 979 order with client register date , all others from 11.01.2017
6. But there are only 67 customers who egister_date an R of 01/01/1970
7. Most customers are from Germany followed by France , Netherlands, Switzerland and Austria
   country number of customers Number of orders
   de 15 . 058 577 . 348
   Fri. 2 . 051 213 . 303
   nl 1 . 877 176 . 033
   ch 1 . 160 76 . 421
   at 1 . 039 39 . 301
8. And now the same thing after authors
   de 9 . 988 600383
   Fri. 3 . 389 200288
   nl 1 . 828 132495
   gb 1 . 456 109182
   at 852 49508
9. Distribution of categories
   miscellaneous 3913 214432
   Health & Medicine 3673 105292
   Product & Category Descriptions 3609 157442
   marketing 3401 27625
   Business & Economy 3283 49496
   Travel & Tourism 2408 87275
   House & Living 2253 69791
   Eat Drink 2236 32279
   electric 2162 26921
   finances 2043 45238
10. Distribution by years
    2016 11577 464212
    2017 11475 455306
    2018 11352 436380
    2019 3326 56729

Analysis Statements:

```
1.) SELECT count (*) from text_broker_order;

2.) SELECT count (distinct client_id) from text_broker_order;

3.) SELECT client_segment,
       count (*) number_order,
       count (distinct client_id) Number_customers
from text_broker_order
group by client_segment;

4.) SELECT min (CLIENT_REGISTER_DATE),
           max (client_register_date)
from text_broker_order;


5.) SELECT CLIENT_REGISTER_DATE REG_date,
count (client_register_date)
from text_broker_order
group by CLIENT_REGISTER_DATE
order by 1;
```

```
6.) SELECT count (distinct client_id )
from text_broker_order
where client_register_date = (select min ( client_register_date ) from
text_broker_order );

7.) SELECT client _country , count (distinct client_id ) ,
count ( *) order
from text_broker_order
group by client_country
order by 2 desc ;

8.) SELECT author _country , count (distinct author _id ) ,
count ( *) order
from text_broker_order
group by author _country
order by 2 desc ;

9.) SELECT Category , count (distinct Client_id ) AnzClient , count (*) AnzOrder
from text_broker_order
group by category
order by 2 desc ;

10.)
SELECT to _char (date_s, ' yyyy ') as year, count (distinct Client_id ) AnzClient ,
count (*) AnzOrder
from text_broker_order
group by to_char (date_s, ' yyyy ')
order by 2 desc ;


Plus more analysis statements

Select client_id , client_last_deposit , count ( *)
from TEXT_BROKER_ORDER
group by client_id , client_last_deposit
order by 1;

Select Client_id , client_first _deposit , count ( *)
from TEXT_BROKER_ORDER
group by client_id , client_first _deposit
order by 1;

Select Client_id , CLIENT_LAST_ORDER_CLOSED, count ( *)
from TEXT_BROKER_ORDER
group by client_id , CLIENT_LAST_ORDER_CLOSED
order by 1;

Select Client_id , CLIENT_FIRST_ORDER_CLOSED, count ( *)
from TEXT_BROKER_ORDER
group by client_id , CLIENT_FIRST_ORDER_CLOSED
order by 1;


Select client_id , system_s , count ( *)
from TEXT_BROKER_ORDER
group by client_id , system_s
order by 1;

Select classification, count ( *)
from TEXT_BROKER_ORDER
group by classification
order by 1;
```

source

Enclosed the source code to create the tables and data. The tables and queries have been implemented with Oracle, so it must be customized for MySQL

Creation table:

```
CREATE TABLE "BI_LAB". "ORDER_PREPARATION_01"
(               "CLIENT_ID" NUMBER,
               "CLIENT_SEGMENT" VARCHAR2 ( 10 BYTE),
               "CLIENT_COUNTRY" VARCHAR2 ( 6 BYTE),
               "CLIENT_REGISTER_DATE" DATE,
               "CLIENT_FIRST_DEPOSIT" DATE,
               "CLIENT_LAST_DEPOSIT" DATE,
               "CLIENT_FIRST_ORDER_CLOSED" DATE,
               "CLIENT_LAST_ORDER_CLOSED" DATE,
               "YEARS_SINCE_REGISTRATION" NUMBER ( 15.5),
               "YEARS_SINCE_FIRST_DEPOSIT" NUMBER ( 15.5),
               "YEARS_SINCE_FIRST_ORDER_CLOSED" NUMBER ( 15.5),
               "SYSTEM_S" VARCHAR2 ( 2 BYTE),
               "COUNT_ORDER" NUMBER,
               "COUNT_ORDER_CLASS_OPEN" NUMBER,
               "AVG_ORDER_CLASS_OPEN" NUMBER ( 15.5),
               "COUNT_ORDER_CLASS_TEAM" NUMBER,
               "AVG_ORDER_CLASS_TEAM" NUMBER ( 15.5),
               "COUNT_ORDER_CLASS_DIRECT" NUMBER,
               "AVG_ORDER_CLASS_DIRECT" NUMBER ( 15.5),
               "COUNT_AUTHOR" NUMBER,
               "AVG_AUTHOR" NUMBER ( 15.5),
               "COUNT_REVISION_REQUEST" NUMBER,
               "AVG_REVISION_REQUEST" NUMBER ( 15.5),
               "SUM_WRITTEN_WORDS" NUMBER,
               "AVG_WRITTEN_WORDS" NUMBER ( 15.5),
               "SUM_GAIN_AUTHOR" NUMBER ( 15.5),
               "AVG_GAIN_AUTHOR" NUMBER ( 15.5),
               "SUM_REVENUE" NUMBER ( 15.5),
               "AVG_REVENUE" NUMBER ( 15.5),
               "SUM_DAYS_TO_WRITE_ORDER" NUMBER,
               "AVG_DAYS_TO_WRITE_ORDER" NUMBER ( 15.5),
               "SUM_DAYS_ORDER_PENDING" NUMBER,
               "AVG_DAYS_ORDER_PENDING" NUMBER ( 15.5),
               "SUM_COUNT_PICKED_BY_AUTHOR" NUMBER,
               "AVG_COUNT_PICKED_BY_AUTHOR" NUMBER ( 15.5),
               "SUM_YEARS_AUTHOR_REG" NUMBER ( 15.5),
               "AVG_YEARS_AUTHOR_REG" NUMBER ( 15.5),
               "ACTIVE_2018" NUMBER,
               "REVENUE_2018" NUMBER
) ;
```

Filling the table :

```
    truncate table order_preparation_01;

    insert into ORDER_PREPARATION_01
SELECT
    o.CLIENT_ID,
               MAX ( CLIENT_SEGMENT) CLIENT_SEGMENT,
               MAX ( CLIENT_COUNTRY) CLIENT_COUNTRY,
               MAX ( CLIENT_REGISTER_DATE) CLIENT_REGISTER_DATE,
               MAX ( CLIENT_FIRST_DEPOSIT) CLIENT_FIRST_DEPOSIT,
               MAX ( CLIENT_LAST_DEPOSIT) CLIENT_LAST_DEPOSIT,
               MAX ( CLIENT_FIRST_ORDER_CLOSED) CLIENT_FIRST_ORDER_CLOSED,
               MAX ( CLIENT_LAST_ORDER_CLOSED) CLIENT_LAST_ORDER_CLOSED,
               AVG ( months_between ( to_date ('01 .01.2018 ',' dd.mm.yyyy '), CLIENT_REGISTER_DATE) /
12) YEARS_SINCE_REGISTRATION,
               AVG ( months_between ( to_date ('01 .01.2018 ',' dd.mm.yyyy '), CLIENT_FIRST_DEPOSIT) /
12) YEARS_SINCE_FIRST_DEPOSIT,
    AVG ( months_between ( to_date ('01 .01.2018 ',' dd.mm.yyyy '), CLIENT_FIRST_ORDER_CLOSED) / 12)
               YEARS_SINCE_FIRST_ORDER_CLOSED,
               MAX ( SYSTEM_S) SYSTEM_S,
               COUNT ( ORDER_ID) COUNT_ORDER,
               sum (decode ( o.classification, 2,1,3,1,4,1,5,1,0)) COUNT_ORDER_CLASS_OPEN,
```

```
               avg (decode ( o.classification , 2,1,3,1,4,1,5,1,0)) AVG_ORDER_CLASS_OPEN,
               sum ( decode ( o.classification , 20,1,0)) COUNT_ORDER_CLASS_TEAM,
               avg ( decode ( o.classification , 20,1,0)) AVG_ORDER_CLASS_TEAM,
               sum ( decode ( o.classification , 10,1,0)) COUNT_ORDER_CLASS_DIRECT,
               avg ( decode ( o.classification , 10,1,0)) AVG_ORDER_CLASS_DIRECT,
               count ( distinct author_id ) COUNT_AUTHOR,
               count ( distinct author_id ) / COUNT (ORDER_ID) AVG_AUTHOR,
               sum ( decode (o.revision_request, 'yes', 1,0)) COUNT_REVISION_REQUEST,
               avg ( decode (o.revision_request, 'yes', 1,0)) AVG_REVISION_REQUEST,
               sum ( o.written_words ) SUM_WRITTEN_WORDS,
               avg ( o.written_words ) AVG_WRITTEN_WORDS,
               sum ( o.gain_author ) SUM_GAIN_AUTHOR,
               avg ( o.gain_author ) AVG_GAIN_AUTHOR,
               sum ( o.written_words * o.gain_author ) SUM_REVENUE,
               avg ( o.written_words * o.gain_author ) AVG_REVENUE,
               sum ( o.days_to_write_order ) SUM_DAYS_TO_WRITE_ORDER,
               avg ( o.days_to_write_order ) AVG_DAYS_TO_WRITE_ORDER,
               sum ( o.days_order_pending_bp_1st_time) SUM_DAYS_ORDER_PENDING,
               avg ( o.days_order_pending_bp_1st_time) AVG_DAYS_ORDER_PENDING,
               sum ( count_picked_by_author ) SUM_COUNT_PICKED_BY_AUTHOR,
               avg ( count_picked_by_author ) AVG_COUNT_PICKED_BY_AUTHOR,
               sum ( months_between ( to_date ('01 .01.2018 ',' dd.mm.yyyy '), o.AUTHOR_REGISTER_DATE ) /
12) SUM_YEARS_AUTHOR_REG,
               avg ( months_between ( to_date ('01 .01.2018 ',' dd.mm.yyyy '), o.AUTHOR_REGISTER_DATE ) /
12) AVG_YEARS_AUTHOR_REG,
               0 ACTIVE_2018,
               0 REVENUE_2018
    from TEXT_BROKER_ORDER o
   where date_s < to_date ('01 .01.2018 ',' dd.mm.yyyy ')
   and date_s > = to_date ('01 .01.2016 ',' dd.mm.yyyy ')
GROUP BY o.client_ID , 0;


    update order_preparation_01 p
set (ACTIVE_2018, REVENUE_2018) = (SELECT decode ( nvl (count ( order_id ), 0), 0,0,1), nvl (sum (
o.written_words * o.gain_author ), 0)
                                        from TEXT_BROKER_ORDER o
                                        where o.client_id = p.client_id
                                        and date_s < to_date ('01 .01.2019 ',' dd.mm.yyyy ')
                                        and date_s > = to_date ('01 .01.2018 ',' dd.mm.yyyy '));
```

Creation table:

```
drop table "BI_LAB". "ORDER_PREPARATION_BY_YEAR";
CREATE TABLE "BI_LAB". "ORDER_PREPARATION_BY_YEAR"
("CLIENT_ID" NUMBER,
               "CLIENT_SEGMENT" VARCHAR2 ( 10 BYTE),
               "CLIENT_COUNTRY" VARCHAR2 ( 6 BYTE),
               "CLIENT_REGISTER_DATE" DATE,
               "CLIENT_FIRST_DEPOSIT" DATE,
               "CLIENT_LAST_DEPOSIT" DATE,
               "CLIENT_FIRST_ORDER_CLOSED" DATE,
               "CLIENT_LAST_ORDER_CLOSED" DATE,
               "YEARS_SINCE_REGISTRATION" NUMBER ( 15.5),
               "YEARS_SINCE_FIRST_DEPOSIT" NUMBER ( 15.5),
               "YEARS_SINCE_FIRST_ORDER_CLOSED" NUMBER ( 15.5),
               "SYSTEM_S" VARCHAR2 ( 2 BYTE),
               "COUNT_ORDER_2016" NUMBER,
               "COUNT_ORDER_CLASS_OPEN_2016" NUMBER,
               "AVG_ORDER_CLASS_OPEN_2016" NUMBER ( 15.5),
               "COUNT_ORDER_CLASS_TEAM_2016" NUMBER,
               "AVG_ORDER_CLASS_TEAM_2016" NUMBER ( 15.5),
               "COUNT_ORDER_CLASS_DIRECT_2016" NUMBER,
               "AVG_ORDER_CLASS_DIRECT_2016" NUMBER ( 15.5),
               "COUNT_AUTHOR_2016" NUMBER,
               "AVG_AUTHOR_2016" NUMBER ( 15.5),
               "COUNT_REVISION_REQUEST_2016" NUMBER,
               "AVG_REVISION_REQUEST_2016" NUMBER ( 15.5),
               "SUM_WRITTEN_WORDS_2016" NUMBER,
               "AVG_WRITTEN_WORDS_2016" NUMBER ( 15.5),
               "SUM_GAIN_AUTHOR_2016" NUMBER ( 15.5),
               "AVG_GAIN_AUTHOR_2016" NUMBER ( 15.5),
               "SUM_REVENUE_2016" NUMBER ( 15.5),
```

```
                "AVG_REVENUE_2016" NUMBER (15.5),
                "SUM_DAYS_TO_WRITE_ORDER_2016" NUMBER,
                "AVG_DAYS_TO_WRITE_ORDER_2016" NUMBER (15.5),
                "SUM_DAYS_ORDER_PENDING_2016" NUMBER,
                "AVG_DAYS_ORDER_PENDING_2016" NUMBER (15.5),
                "SUM_COUNT_PICKED_BY_AUTH_2016" NUMBER,
                "AVG_COUNT_PICKED_BY_AUTH_2016" NUMBER (15.5),
                "SUM_YEARS_AUTHOR_REG_2016" NUMBER (15.5),
                "AVG_YEARS_AUTHOR_REG_2016" NUMBER (15.5),
                "COUNT_ORDER_2017" NUMBER,
                "COUNT_ORDER_CLASS_OPEN_2017" NUMBER,
                "AVG_ORDER_CLASS_OPEN_2017" NUMBER (15.5),
                "COUNT_ORDER_CLASS_TEAM_2017" NUMBER,
                "AVG_ORDER_CLASS_TEAM_2017" NUMBER (15.5),
                "COUNT_ORDER_CLASS_DIRECT_2017" NUMBER,
                "AVG_ORDER_CLASS_DIRECT_2017" NUMBER (15.5),
                "COUNT_AUTHOR_2017" NUMBER,
                "AVG_AUTHOR_2017" NUMBER (15.5),
                "COUNT_REVISION_REQUEST_2017" NUMBER,
                "AVG_REVISION_REQUEST_2017" NUMBER (15.5),
                "SUM_WRITTEN_WORDS_2017" NUMBER,
                "AVG_WRITTEN_WORDS_2017" NUMBER (15.5),
                "SUM_GAIN_AUTHOR_2017" NUMBER (15.5),
                "AVG_GAIN_AUTHOR_2017" NUMBER (15.5),
                "SUM_REVENUE_2017" NUMBER (15.5),
                "AVG_REVENUE_2017" NUMBER (15.5),
                "SUM_DAYS_TO_WRITE_ORDER_2017" NUMBER,
                "AVG_DAYS_TO_WRITE_ORDER_2017" NUMBER (15.5),
                "SUM_DAYS_ORDER_PENDING_2017" NUMBER,
                "AVG_DAYS_ORDER_PENDING_2017" NUMBER (15.5),
                "SUM_COUNT_PICKED_BY_AUTH_2017" NUMBER,
                "AVG_COUNT_PICKED_BY_AUTH_2017" NUMBER (15.5),
                "SUM_YEARS_AUTHOR_REG_2017" NUMBER (15.5),
                "AVG_YEARS_AUTHOR_REG_2017" NUMBER (15.5),
                "ACTIVE_2018" NUMBER,
                "REVENUE_2018" NUMBER
) SEGMENT CREATION IMMEDIATE
PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255 NOCOMPRESS LOGGING
  STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE
DEFAULT)
TABLESPACE "USERS";

CREATE INDEX "BI_LAB". "IX_ORDER_PREP_02_CLIENT_ID" ON "BI_LAB". "ORDER_PREPARATION_BY_YEAR"
("CLIENT_ID")
PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS
  STORAGE (INITIAL 65536 NEXT 1048576 MINEXTENTS 1 MAXEXTENTS 2147483645
PCTINCREASE 0 FREELISTS 1 FREELIST GROUPS 1 BUFFER_POOL DEFAULT FLASH_CACHE DEFAULT CELL_FLASH_CACHE
DEFAULT)
TABLESPACE "USERS";
```

Filling the table :

```
    insert into ORDER_PREPARATION_BY_YEAR
SELECT
    o.CLIENT_ID,
                MAX (CLIENT_SEGMENT) CLIENT_SEGMENT,
                MAX (CLIENT_COUNTRY) CLIENT_COUNTRY,
                MAX (CLIENT_REGISTER_DATE) CLIENT_REGISTER_DATE,
                MAX (CLIENT_FIRST_DEPOSIT) CLIENT_FIRST_DEPOSIT,
                MAX (CLIENT_LAST_DEPOSIT) CLIENT_LAST_DEPOSIT,
                MAX (CLIENT_FIRST_ORDER_CLOSED) CLIENT_FIRST_ORDER_CLOSED,
                MAX (CLIENT_LAST_ORDER_CLOSED) CLIENT_LAST_ORDER_CLOSED,
                AVG (months_between (to_date ('01 .01.2018 ',' dd.mm.yyyy '), CLIENT_REGISTER_DATE) /
12) YEARS_SINCE_REGISTRATION,
                AVG (months_between (to_date ('01 .01.2018 ',' dd.mm.yyyy '), CLIENT_FIRST_DEPOSIT) /
12) YEARS_SINCE_FIRST_DEPOSIT,
    AVG (months_between (to_date ('01 .01.2018 ',' dd.mm.yyyy '), CLIENT_FIRST_ORDER_CLOSED) / 12)
                YEARS_SINCE_FIRST_ORDER_CLOSED,
                MAX (SYSTEM_S) SYSTEM_S,
                0 COUNT_ORDER_2016,
                0 COUNT_ORDER_CLASS_OPEN_2016,
                0 AVG_ORDER_CLASS_OPEN_2016,
                0 COUNT_ORDER_CLASS_TEAM_2016,
                0 AVG_ORDER_CLASS_TEAM_2016,
                0 COUNT_ORDER_CLASS_DIRECT_2016,
                0 AVG_ORDER_CLASS_DIRECT_2016,
```

```
                0 COUNT_AUTHOR_2016,
                0 AVG_AUTHOR_2016,
                0 COUNT_REVISION_REQUEST_2016,
                0 AVG_REVISION_REQUEST_2016,
                0 SUM_WORDS_WORDS_2016,
                0 AVG_WRITTEN_WORDS_2016,
                0 SUM_GAIN_AUTHOR_2016,
                0 AVG_GAIN_AUTHOR_2016,
                0 SUM_REVENUE_2016,
                0 AVG_REVENUE_2016,
                0 SUM_DAYS_TO_WRITE_ORDER_2016,
                0 AVG_DAYS_TO_WRITE_ORDER_2016,
                0 SUM_DAYS_ORDER_PENDING_2016,
                0 AVG_DAYS_ORDER_PENDING_2016,
                0 SUM_COUNT_PICKED_BY_AUTH_2016,
                0 AVG_COUNT_PICKED_BY_AUTH_2016,
  0 SUM_YEARS_AUTHOR_REG_2016,
                0 AVG_YEARS_AUTHOR_REG_2016,
  0 COUNT_ORDER_2017,
                0 COUNT_ORDER_CLASS_OPEN_2017,
                0 AVG_ORDER_CLASS_OPEN_2017,
                0 COUNT_ORDER_CLASS_TEAM_2017,
                0 AVG_ORDER_CLASS_TEAM_2017,
                0 COUNT_ORDER_CLASS_DIRECT_2017,
                0 AVG_ORDER_CLASS_DIRECT_2017,
                0 COUNT_AUTHOR_2017,
                0 AVG_AUTHOR_2017,
                0 COUNT_REVISION_REQUEST_2017,
                0 AVG_REVISION_REQUEST_2017,
                0 SUM_WORDS_WORDS_2017,
                0 AVG_WRITTEN_WORDS_2017,
                0 SUM_GAIN_AUTHOR_2017,
                0 AVG_GAIN_AUTHOR_2017,
                0 SUM_REVENUE_2017,
                0 AVG_REVENUE_2017,
                0 SUM_DAYS_TO_WRITE_ORDER_2017,
                0 AVG_DAYS_TO_WRITE_ORDER_2017,
                0 SUM_DAYS_ORDER_PENDING_2017,
                0 AVG_DAYS_ORDER_PENDING_2017,
                0 SUM_COUNT_PICKED_BY_AUTH_2017,
                0 AVG_COUNT_PICKED_BY_AUTH_2017,
  0 SUM_YEARS_AUTHOR_REG_2017,
                0 AVG_YEARS_AUTHOR_REG_2017,
                0 ACTIVE_2018,
                0 REVENUE_2018
     from TEXT_BROKER_ORDER o
    where date_s < to_date ('01 .01.2018 ',' dd.mm.yyyy ')
    and date_s > = to_date ('01 .01.2016 ',' dd.mm.yyyy ')
  GROUP BY o.client_ID, 0;

     merge into ORDER_PREPARATION_BY_YEAR p
     USING ( SELECT
     o.CLIENT_ID,
                COUNT ( ORDER_ID) COUNT_ORDER_2016,
                sum (decode ( o.classification, 2,1,3,1,4,1,5,1,0)) COUNT_ORDER_CLASS_OPEN_2016,
                avg (decode ( o.classification, 2,1,3,1,4,1,5,1,0)) AVG_ORDER_CLASS_OPEN_2016,
                sum ( decode ( o.classification, 20,1,0)) COUNT_ORDER_CLASS_TEAM_2016,
                avg ( decode ( o.classification, 20,1,0)) AVG_ORDER_CLASS_TEAM_2016,
                sum ( decode ( o.classification, 10,1,0)) COUNT_ORDER_CLASS_DIRECT_2016,
                avg ( decode ( o.classification, 10,1,0)) AVG_ORDER_CLASS_DIRECT_2016,
                count ( distinct author_id ) COUNT_AUTHOR_2016,
                count ( distinct author_id ) / COUNT (ORDER_ID) AVG_AUTHOR_2016,
                sum ( decode (o.revision_request, 'yes', 1,0)) COUNT_REVISION_REQUEST_2016,
                avg ( decode (o.revision_request, 'yes', 1,0)) AVG_REVISION_REQUEST_2016,
                sum ( o.written_words ) SUM_WRITTEN_WORDS_2016,
                avg ( o.written_words ) AVG_WRITTEN_WORDS_2016,
                sum ( o.gain_author ) SUM_GAIN_AUTHOR_2016,
                avg ( o.gain_author ) AVG_GAIN_AUTHOR_2016,
                sum ( o.written_words * o.gain_author ) SUM_REVENUE_2016,
                avg ( o.written_words * o.gain_author ) AVG_REVENUE_2016,
                sum ( o.days_to_write_order ) SUM_DAYS_TO_WRITE_ORDER_2016,
                avg ( o.days_to_write_order ) AVG_DAYS_TO_WRITE_ORDER_2016,
                sum ( o.days_order_pending_bp_1st_time) SUM_DAYS_ORDER_PENDING_2016,
                avg ( o.days_order_pending_bp_1st_time) AVG_DAYS_ORDER_PENDING_2016,
                sum ( count_picked_by_author ) SUM_COUNT_PICKED_BY_AUTH_2016,
```

```
              avg ( count_picked_by_author ) AVG_COUNT_PICKED_BY_AUTH_2016,
              sum ( months_between ( to_date ('01 .01.2018 ',' dd.mm.yyyy '), o.AUTHOR_REGISTER_DATE ) /
12) SUM_YEARS_AUTHOR_REG_2016,
              avg ( months_between ( to_date ('01 .01.2018 ',' dd.mm.yyyy '), o.AUTHOR_REGISTER_DATE ) /
12) AVG_YEARS_AUTHOR_REG_2016
    from TEXT_BROKER_ORDER o
   where date_s < to_date ('01 .01.2017 ',' dd.mm.yyyy ')
   and date_s > = to_date ('01 .01.2016 ',' dd.mm.yyyy ')
GROUP BY o.client_ID, 0) om
    on ( om.client_id = p.client_id )
   when matched then update set
p.COUNT_ORDER_2016 = om.COUNT_ORDER_2016,
p.COUNT_ORDER_CLASS_OPEN_2016 = om.COUNT_ORDER_CLASS_OPEN_2016,
p.AVG_ORDER_CLASS_OPEN_2016 = om.AVG_ORDER_CLASS_OPEN_2016,
p.COUNT_ORDER_CLASS_TEAM_2016 = om.COUNT_ORDER_CLASS_TEAM_2016,
p.AVG_ORDER_CLASS_TEAM_2016 = om.AVG_ORDER_CLASS_TEAM_2016,
p.COUNT_ORDER_CLASS_DIRECT_2016 = om.COUNT_ORDER_CLASS_DIRECT_2016,
              p.AVG_ORDER_CLASS_DIRECT_2016 = om.AVG_ORDER_CLASS_DIRECT_2016,
              p.COUNT_AUTHOR_2016 = om.COUNT_AUTHOR_2016,
              p.AVG_AUTHOR_2016 = om.AVG_AUTHOR_2016,
              p.COUNT_REVISION_REQUEST_2016 = om.COUNT_REVISION_REQUEST_2016,
              p.AVG_REVISION_REQUEST_2016 = om.AVG_REVISION_REQUEST_2016,
              p.SUM_WRITTEN_WORDS_2016 = om.SUM_WRITTEN_WORDS_2016,
              p.AVG_WRITTEN_WORDS_2016 = om.AVG_WRITTEN_WORDS_2016,
              p.SUM_GAIN_AUTHOR_2016 = om.SUM_GAIN_AUTHOR_2016,
              p.AVG_GAIN_AUTHOR_2016 = om.AVG_GAIN_AUTHOR_2016,
              p.SUM_REVENUE_2016 = om.SUM_REVENUE_2016,
              p.AVG_REVENUE_2016 = om.AVG_REVENUE_2016,
              p.SUM_DAYS_TO_WRITE_ORDER_2016 = om.SUM_DAYS_TO_WRITE_ORDER_2016,
              p.AVG_DAYS_TO_WRITE_ORDER_2016 = om.AVG_DAYS_TO_WRITE_ORDER_2016,
              p.SUM_DAYS_ORDER_PENDING_2016 = om.SUM_DAYS_ORDER_PENDING_2016,
              p.AVG_DAYS_ORDER_PENDING_2016 = om.AVG_DAYS_ORDER_PENDING_2016,
              p.SUM_COUNT_PICKED_BY_AUTH_2016 = om.SUM_COUNT_PICKED_BY_AUTH_2016,
              p.AVG_COUNT_PICKED_BY_AUTH_2016 = om.AVG_COUNT_PICKED_BY_AUTH_2016,
p.SUM_YEARS_AUTHOR_REG_2016 = om.SUM_YEARS_AUTHOR_REG_2016,
              p.AVG_YEARS_AUTHOR_REG_2016 = om.AVG_YEARS_AUTHOR_REG_2016

;

    merge into ORDER_PREPARATION_BY_YEAR p
    USING ( SELECT
    o.CLIENT_ID,
              COUNT ( ORDER_ID) COUNT_ORDER_2017,
              sum (decode ( o.classification , 2,1,3,1,4,1,5,1,0)) COUNT_ORDER_CLASS_OPEN_2017,
              avg (decode ( o.classification , 2,1,3,1,4,1,5,1,0)) AVG_ORDER_CLASS_OPEN_2017,
              sum ( decode ( o.classification , 20,1,0)) COUNT_ORDER_CLASS_TEAM_2017,
              avg ( decode ( o.classification , 20,1,0)) AVG_ORDER_CLASS_TEAM_2017,
              sum ( decode ( o.classification , 10,1,0)) COUNT_ORDER_CLASS_DIRECT_2017,
              avg ( decode ( o.classification , 10,1,0)) AVG_ORDER_CLASS_DIRECT_2017,
              count ( distinct author_id ) COUNT_AUTHOR_2017,
              count ( distinct author_id ) / COUNT (ORDER_ID) AVG_AUTHOR_2017,
              sum ( decode (o.revision_request, 'yes', 1,0)) COUNT_REVISION_REQUEST_2017,
              avg ( decode (o.revision_request, 'yes', 1,0)) AVG_REVISION_REQUEST_2017,
              sum ( o.written_words ) SUM_WRITTEN_WORDS_2017,
              avg ( o.written_words ) AVG_WRITTEN_WORDS_2017,
              sum ( o.gain_author ) SUM_GAIN_AUTHOR_2017,
              avg ( o.gain_author ) AVG_GAIN_AUTHOR_2017,
              sum ( o.written_words * o.gain_author ) SUM_REVENUE_2017,
              avg ( o.written_words * o.gain_author ) AVG_REVENUE_2017,
              sum ( o.days_to_write_order ) SUM_DAYS_TO_WRITE_ORDER_2017,
              avg ( o.days_to_write_order ) AVG_DAYS_TO_WRITE_ORDER_2017,
              sum ( o.days_order_pending_bp_1st_time) SUM_DAYS_ORDER_PENDING_2017,
              avg ( o.days_order_pending_bp_1st_time) AVG_DAYS_ORDER_PENDING_2017,
              sum ( count_picked_by_author ) SUM_COUNT_PICKED_BY_AUTH_2017,
              avg ( count_picked_by_author ) AVG_COUNT_PICKED_BY_AUTH_2017,
              sum ( months_between ( to_date ('01 .01.2018 ',' dd.mm.yyyy '), o.AUTHOR_REGISTER_DATE ) /
12) SUM_YEARS_AUTHOR_REG_2017,
              avg ( months_between ( to_date ('01 .01.2018 ',' dd.mm.yyyy '), o.AUTHOR_REGISTER_DATE ) /
12) AVG_YEARS_AUTHOR_REG_2017
    from TEXT_BROKER_ORDER o
   where date_s < to_date ('01 .01.2018 ',' dd.mm.yyyy ')
   and date_s > = to_date ('01 .01.2017 ',' dd.mm.yyyy ')
GROUP BY o.client_ID, 0) om
    on ( om.client_id = p.client_id )
   when matched then update set
p.COUNT_ORDER_2017 = om.COUNT_ORDER_2017,
p.COUNT_ORDER_CLASS_OPEN_2017 = om.COUNT_ORDER_CLASS_OPEN_2017,
```

```
p.AVG_ORDER_CLASS_OPEN_2017 = om.AVG_ORDER_CLASS_OPEN_2017,
p.COUNT_ORDER_CLASS_TEAM_2017 = om.COUNT_ORDER_CLASS_TEAM_2017,
p.AVG_ORDER_CLASS_TEAM_2017 = om.AVG_ORDER_CLASS_TEAM_2017,
p.COUNT_ORDER_CLASS_DIRECT_2017 = om.COUNT_ORDER_CLASS_DIRECT_2017,
              p.AVG_ORDER_CLASS_DIRECT_2017 = om.AVG_ORDER_CLASS_DIRECT_2017,
              p.COUNT_AUTHOR_2017 = om.COUNT_AUTHOR_2017,
              p.AVG_AUTHOR_2017 = om.AVG_AUTHOR_2017,
              p.COUNT_REVISION_REQUEST_2017 = om.COUNT_REVISION_REQUEST_2017,
              p.AVG_REVISION_REQUEST_2017 = om.AVG_REVISION_REQUEST_2017,
              p.SUM_WRITTEN_WORDS_2017 = om.SUM_WRITTEN_WORDS_2017,
              p.AVG_WRITTEN_WORDS_2017 = om.AVG_WRITTEN_WORDS_2017,
              p.SUM_GAIN_AUTHOR_2017 = om.SUM_GAIN_AUTHOR_2017,
              p.AVG_GAIN_AUTHOR_2017 = om.AVG_GAIN_AUTHOR_2017,
              p.SUM_REVENUE_2017 = om.SUM_REVENUE_2017,
              p.AVG_REVENUE_2017 = om.AVG_REVENUE_2017,
              p.SUM_DAYS_TO_WRITE_ORDER_2017 = om.SUM_DAYS_TO_WRITE_ORDER_2017,
              p.AVG_DAYS_TO_WRITE_ORDER_2017 = om.AVG_DAYS_TO_WRITE_ORDER_2017,
              p.SUM_DAYS_ORDER_PENDING_2017 = om.SUM_DAYS_ORDER_PENDING_2017,
              p.AVG_DAYS_ORDER_PENDING_2017 = om.AVG_DAYS_ORDER_PENDING_2017,
              p.SUM_COUNT_PICKED_BY_AUTH_2017 = om.SUM_COUNT_PICKED_BY_AUTH_2017,
              p.AVG_COUNT_PICKED_BY_AUTH_2017 = om.AVG_COUNT_PICKED_BY_AUTH_2017,
p.SUM_YEARS_AUTHOR_REG_2017 = om.SUM_YEARS_AUTHOR_REG_2017,
              p.AVG_YEARS_AUTHOR_REG_2017 = om.AVG_YEARS_AUTHOR_REG_2017

;
```

basis for the association table :

(Further processing in Excel for transposition)

```
select client_id, category, count (client_id) anz

from TEXT_BROKER_ORDER where category in
('Miscellaneous',
'Health' || '&' || 'Medicine' ,
'Product' || '&' || ' Category Descriptions ',
'Marketing',
'Business' || '&' || 'Economy'
'Travel' || '&' || 'Tourism',
'House' || '&' || 'Living'
'Food' || '&' || 'Drink',
'Electrical' || '&' || 'Electronics',
'Finance'
'Languages' || '&' || 'Translations (source language => German)',
'Sports',
'Internet',
'Craft'
'Shopping'
'Cars' || '&' || 'Transport'
'Beauty',
'Computer' || ' & ' || 'Software',
'Medien' || ' & ' || 'Telekommunikation',
'Entertainment',
'Mode',
'Immobilien',
'Aus- und Weiterbildung',
'Hobbys',
'Erziehung' || ' & ' || 'Familie',
'Tiere',
'News' || ' & ' || 'Zeitgeschehen',
'Events',
' Lektorat ',
' Recht ',
'Fitness' || ' & ' || 'Lifestyle',
' Literatur ',
'Job' || ' & ' || ' Karriere ',
' Musik ' || ' & ' || ' Tanz ',
'Garten' || ' & ' || ' Pflanzen ',
'Wissenschaften',
'Geschichte' || ' & ' || 'Politik',
'Schmuck' || ' & ' || 'Edelmetalle',
'Kunst' || ' & ' || 'Kunsthandwerk',
'Logistik',
'Gastgewerbe',
'Spirituelles' || ' & ' || 'Philosophie',
'Horoskope' || ' & ' || 'Wahrsagen',
'Nicht jugendfrei (18+)',
```

```
  'Wetten' || ' & ' || 'Glücksspiel'
  )
  group by Client_id, category
  ;
```