https://unsplash.com/photos/gm3bxHin8VA

Aalto University
School of Business

MySQL for Data Analytics

Lecturer: Yong Liu
Contact me at: Yong.liu@aalto.fi

Big Assignment

- Deadline: **November 08, 2024**. You have one month to work on the assignment after the last lecture of the course.
- Please do not share your code or copy others' code. Cheating (such as copying) on assignments will result in a failing grade for the course.
- The **evaluation scheme** is available in slides of the first lecture.
- **Hints**: Please do the hands-on session assignments first! After finishing the hands-on session assignments, it should not be difficult for you to pass the course by doing the big assignment.
- **Aalto VDI breakdown**: Kindly be aware that there might be instances when Aalto VDI experiences technical issues at the time of your assignment submission. Therefore, it is advisable to complete your assignment well in advance of the deadline. Unless Aalto VDI experiences an unusually prolonged outage (e.g., lasting over 48 hours) within the final three days leading up to the deadline, no extension will be granted.

An Excellent summary of basic MySQL commands

- **MySQL by Examples for Beginners**

  https://www3.ntu.edu.sg/home/ehchua/programming/sql/MySQL_Beginner.html

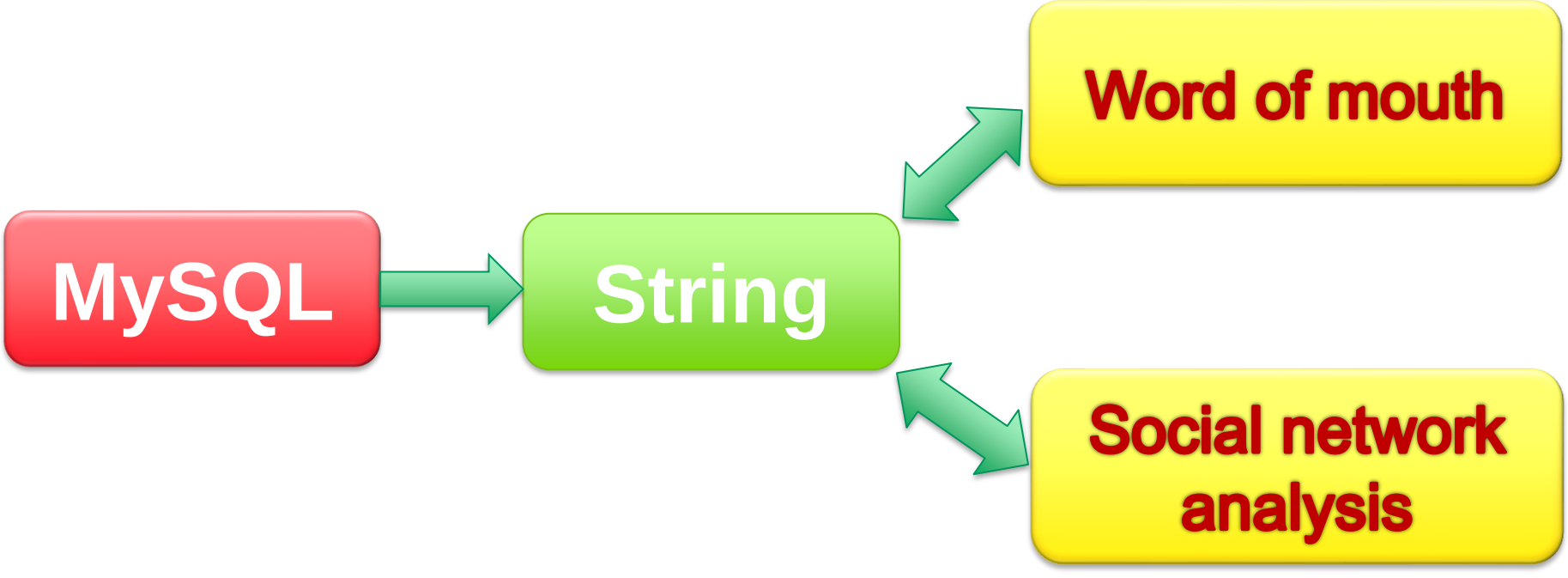
Objectives for class 4

- **String Analytics**
- **MySQL Keyword: Select (2)**
  - **Operation on string**
  - **Group by function (basics)**

‘String Analytics’

- **Not well handled by existing BI tools.**
- **An important component of tasks for many BI roles.**

MySQL for String Manipulation



A case of Supercell Game

- **Who is the main competitor?**

https://www.reddit.com/r/ClashOfClans/comments/1asedgl/project_global_chat/

Word of mouth

"Excellent place in Helsinki"

Reviewed February 7, 2015

Realy nice place in the heart of Helsinki. Excellent food and wines combimed with passionate service. I can warmly recommend this place.

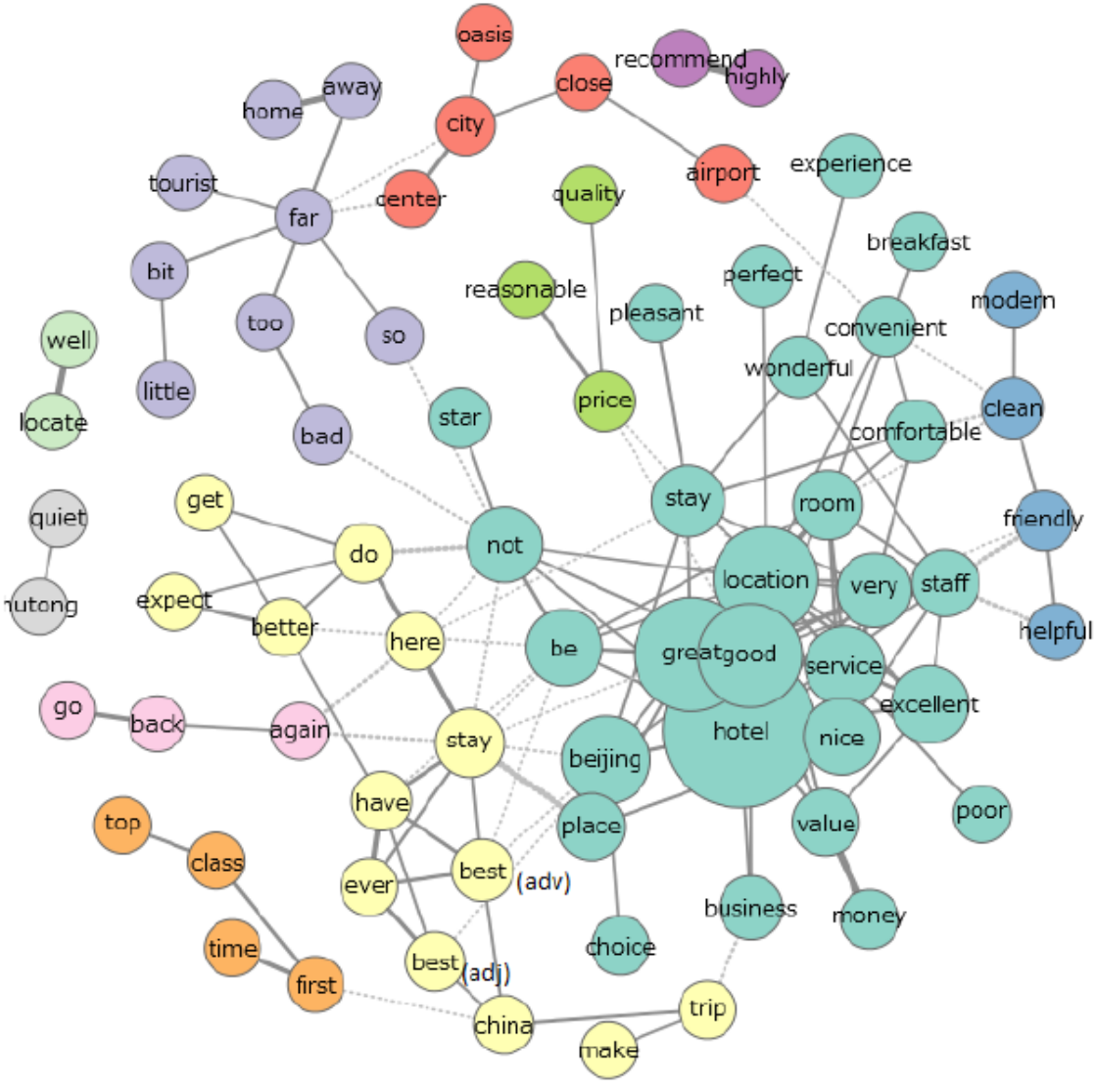
Was this review helpful? Yes 1

"Unique dinning experience"

Reviewed August 13, 2013 via mobile

Enjoyed all the dishes. Beatifully presented and tasted good. Service was very attentive. I didn't try the drink menu but 8 course meal was a unique dinning experience.

Was this review helpful? Yes


oasis
recommend
highly
close
away
home
city
tourist
far
center
airport
experience
quality
breakfast
perfect
reasonable
pleasant
modern
convenient
bit
too
so
wonderful
well
little
price
clean
locate
bad
star
comfortable
stay
room
get
quiet
do
not
location
very
staff
friendly
hutong
expect
better
here
be
greatgood
service
helpful
go
back
again
stay
beijing
hotel
nice
excellent
have
place
value
poor
top
class
best (adv)
ever
business
money
time
best (adj)
choice
first
china
trip
make

‘String’ to strategic insights

- **Yong Liu, Thorsten Teichert, Feng Hu, Hongxiu Li (2016) How do tourists evaluate Chinese hotels at different cities? Mining online tourist reviewers for new insights.**

https://pdfs.semanticscholar.org/4558/06c1da761e13d8b4d2481d8d64f1639ccebe.pdf

Most often appeared words

Table 2. The hotel attributes that perform poorly in tourists' evaluation

Shanghai (n =3298)	Sanya (n = 347)	Hangzhou (n = 451)	Guangzhou (n = 1172)	Beijing (n = 3752)
Service (12.5 %)	Service (13.5 %)	Service (9.0 %)	Service (9.3 %)	Service (9.9 %)
Location (6.6 %)	Resort (4.3 %)	Location (5.3 %)	Location (4.6 %)	Room (6.5 %)
Room (5.4 %)	Staff (2.5 %)	Room (5.0 %)	Room (4 %)	Location (6.0 %)
Staff (4.0 %)	Food (2.0 %)	Staff (3.7 %)	Dirty (2.7 %)	Staff (3.6 %)
Dirty (1.5 %)	Room (2.0 %)	Air (1.7 %)	Staff (2.6 %)	Dirty (2.5 %)

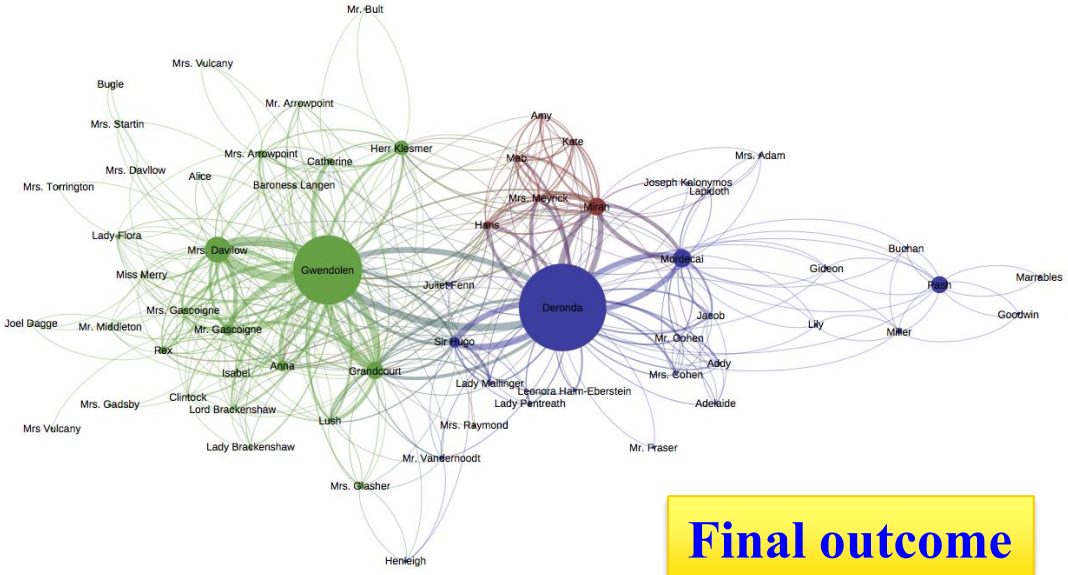
Yong Liu, Thorsten Teichert, Feng Hu, Hongxiu Li (2016)
How do tourists evaluate Chinese hotels at different cities? Mining online tourist reviewers for new insights.

Raw data

Communication ID	From	To	time
1	Alice	Ben	2005-02-03, 16:30:20
2	Alice	Ben	2005-05-08, 12:15:10
3	Alice	Danielle	2005-03-09, 09:32:26
4	Ben	Carl	…
5	…	…	…

Handled data

	Alice	Ben	Carl	Danielle	Edgar
Alice	0	25	17	28	15
Ben	25	0	42	3	10
Carl	17	42	0	45	32
Danielle	28	3	45	0	13
Edgar	15	10	32	13	0



Final outcome

http://www.umasocialmedia.com/socialnetworks/wp-content/uploads/2014/03/age1mode.png

https://ludicanalytics.files.wordpress.com/2013/11/daniel-deronda-in-gephi.png

Detecting personality using words

- **http://www.psychometrics.cam.ac.uk/news/the-language-of-social-media**
- **Tal Yarkoni (2010). Personality in 100,000 Words: A large-scale analysis of personality and word use among bloggers, J Res Pers. 2010 Jun 1; 44(3): 363–373.**

Datasets for hands-on session

- **CFPB Consumer Complaint Database**
  - **Consumer Financial Protection Bureau (CFPB) launched a public Consumer Complaint Database on credit cards, mortgages, private student loan etc.**

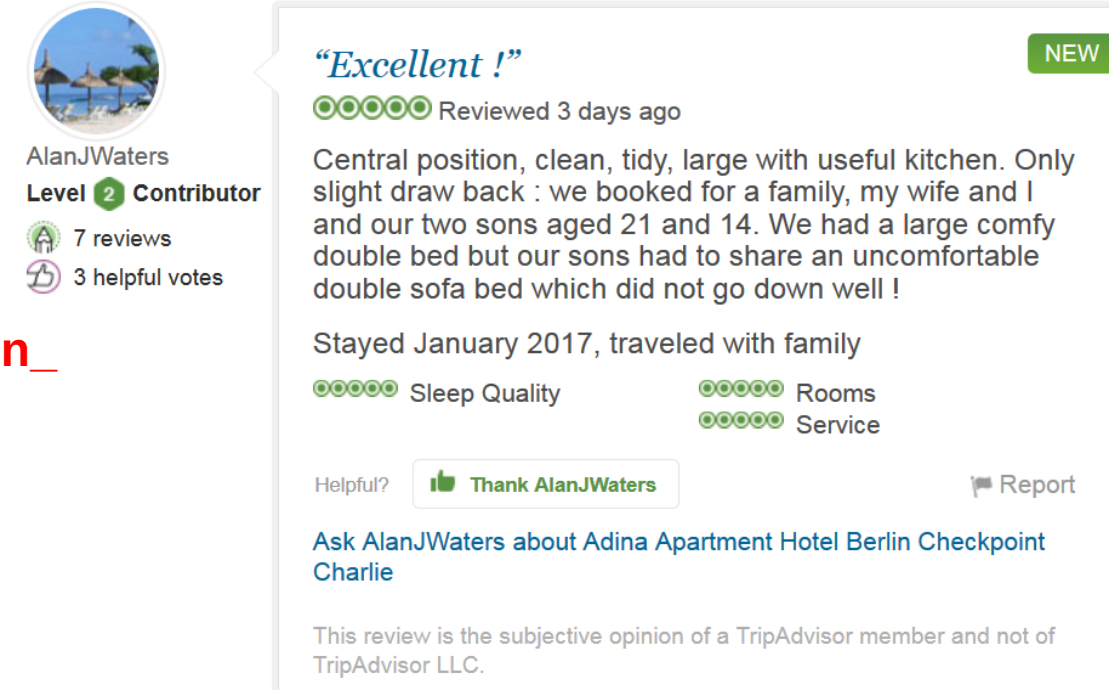
http://www.consumerfinance.gov/newsroom/consumer-financial-protection-bureau-launches-consumer-complaint-database/

https://www.consumerfinance.gov/data-research/consumer-complaints/#download-the-data

	Complaint ID	Product	Sub-product	Issue	Sub-issue	State	ZIP code	Submitted via	Date received	Date sent to company	Company	Company response	Timely response
1	1273083	Bank account or servic	(CD) Certificate of deposit	Account opening, closing, or managem		MA	20902	Phone	03/09/2015	03/09/2015	Ally Financial Inc.	In progress	Yes
2	1272928	Debt collection	Medical	Cont'd attempts collect debt not owed	Debt is not min	IL	61250	Web	03/09/2015	03/09/2015	Transworld Systems Inc.	In progress	Yes
3	1273115	Bank account or servic	Other bank product/service	Account opening, closing, or managem		PA	19149	Web	03/09/2015	03/09/2015	Santander Bank US	In progress	Yes
4	1272530	Debt collection	Other (phone, health club, etc.)	False statements or representation	Attempted to co	MA	02301	Web	03/08/2015	03/08/2015	Ability Recovery Services, LLC	Closed with explanation	Yes

Datasets for hands-on session

TripAdvisor data:
tripadvisor_data_for_handson_assignment_ONLY.sql



DRMCourse.tripadvisor_data_for_handson_assignment_ONLY: 1,979 rows total (approximately), limited to 1,000 — Next | Show all | Sorting | Columns

id	title	year_stayed	month_stayed	hotel_id	num_helpful_votes	review_date	via_mobile	rooms	value
145,862,422	"Good Place"	2,012	9	2,145,772	0	2012-11-21	FALSE	0	0
145,862,517	"Great Hotel With a Great Staff"	2,012	8	98,821	0	2012-11-21	FALSE	0	0
145,862,957	"Good quality budget hotel in expens...	2,012	11	81,042	1	2012-11-21	FALSE	3	3

Select…like [binary]…

customerNumber	customerName	contactLastName	contactFirstName	phone
103	Atelier graphique	Schmitt	Carine	40.32.2555
112	Signal Gift Stores	King	Jean	7025551838
114	Australian Collectors, Co.	Ferguson	Peter	03 9520 4555
119	La Rochelle Gifts	Labrune	Janine	40.67.8555
121	Baane Mini Imports	Bergulfsen	Jonas	07-98 9555
124	Mini Gifts Distributors Ltd.	Nelson	Susan	4155551450
125	Havel & Zbyszek Co	Piestrzeniewicz	Zbyszek	(26) 642-7555
128	Blauer See Auto, Co.	Keitel	Roland	+49 69 66 90 2555
129	Mini Wheels Co.	Murphy	Julie	6505555787
131	Land of Toys Inc.	Lee	Kwai	2125557818
141	Euro+ Shopping Channel	Freyre	Diego	(91) 555 94 44
144	Volvo Model Replicas, Co	Berglund	Christina	0921-12 3555
145	Danish Wholesale Imports	Petersen	Jytte	31 12 3555
146	Saveley & Henriot, Co.	Saveley	Mary	78.32.5555
148	Dragon Souveniers, Ltd.	Natividad	Eric	+65 221 7555
151	Muscle Machine Inc	Young	Jeff	2125557413
157	Diecast Classics Inc.	Leong	Kelvin	2155551555
161	Technics Stores Inc.	Hashimoto	Juri	6505556809
166	Handji Gifts& Co	Victorino	Wendy	+65 224 1555
167	Herkku Gifts	Oeztan	Veysel	+47 2267 3215
168	American Souvenirs Inc	Franco	Keith	2035557845
169	Porto Imports Co.	de Castro	Isabel	(1) 356-5555

How to retrieve records in which customers' first name starting with the word "ro"?

Select…like [binary]…

- **Keyword Like is used to compare strings.**

- "%" represents a string of any possible length

```
select * from customers where contactFirstName like 'ro%'
```

- "_" represents a single possible character.

```
select * from customers where contactFirstName like 'ro__'
```

customerNumber	customerName	contactLastName	contactFirstName	phone
128	Blauer See Auto, Co.	Keitel	Roland	+49 69 66 90 2555
452	Mini Auto Werke	Mendel	Roland	7675-3555
486	Motor Mint Distributors Inc.	Salazar	Rosa	2155559857

Select…like [binary]…

- **Which query will return the result below?**

Binary makes comparison **case-sensitive**

A.select * from customers where contactFirstName like **binary** 'ro%';

B.select * from customers where contactFirstName like **binary** 'Ro%';

customerNumber	customerName	contactLastName	contactFirstName	phone
128	Blauer See Auto, Co.	Keitel	Roland	+49 69 66 90 2555
452	Mini Auto Werke	Mendel	Roland	7675-3555
486	Motor Mint Distributors Inc.	Salazar	Rosa	2155559857

Questions

What does this query mean?

```
Select * from customers where contactFirstName not like 'ro__'
```

How to select customers whose first name consists of 4 characters?

```
select * from customers where contactFirstName like '____'
```

Questions

- **How to retrieve records in which ‘Issue’ contains the word ‘loan’?**

select * from cfpb_complaints_2500 where **'%loan%'**

Complaint ID	Product	Sub_product	Issue
2 298	Credit card		Billing disputes
2 315	Mortgage	Conventional adjustable mortgage (A...	Loan servicing, payments, escrow acc...
2 340	Credit card		Billing disputes
2 348	Credit card		Billing disputes

http://presemo.aalto.fi/drm

Select… IN…

- **Select …**
  [not] IN (element 1, element 2, …, element N)

```
Select * from customers
where contactFirstName in ('Michael','Elizabeth')
```

Select … REGEXP…

- **MySQL REGEXP performs a pattern match of a string expression against a pattern.**
  E.g. How to select the records in which addressline1 contains the words ‘road’, ‘tower’ or ‘airport’

```
Select * from customers
where addressLine1 regexp 'road|tower|airport'
```

Select … REGEXP…(2)

Function	Select * from *tableName*
^	Where *ColumnName REGEXP* '^L'
$	Where *ColumnName REGEXP* 'c$'
.	Where *ColumnName REGEXP* '^M..y$'
[ ]	Where *ColumnName REGEXP* '[abc]' or '[0-9]'
*	Where *ColumnName REGEXP* 'a*c'
+	Where *ColumnName REGEXP* 'a+c'
{N}	Where *ColumnName REGEXP* 'a{2}'
{M, N}	Where *ColumnName REGEXP* 'a{1,2}'

http://dev.mysql.com/doc/refman/5.7/en/regexp.html

- **^L:** string start with L
- **c$:** string ends with c
- **M..y$:** string with L, end with y, and two characters in between.
- **'[abc]':** string contain either a, or b or c.
- **'a*c':** string in which character a appear before character c for 0 or several time.
- **'a+c'** string in which character a appear before character c for at least one time.
- **'a{2}':** string in which ‘aa’ appears
- **'a{3}':** string in which ‘aaa’ appears
- **'a{1,2}':** string in which ‘a’ or ‘aa’ appears

What does the queries mean?

- **Select Issue**

from cfpb_complaints_2500 where Issue regexp '^a|^b';

- **select** Issue

from cfpb_complaints_2500 **where** Issue **not regexp** ' {1}';

classicmodels.customers: 122 rows total (approximately)

customerNumber	customerName	contactLastName	contactFirstName	phone
216	Enaco Distributors	Saavedra	Eduardo	(93) 203 4555
298	Vida Sport, Ltd	Holz	Mihael	0897-034555
344	CAF Imports	Fernandez	Jesus	+34 913 728 555

- **In the column ‘phone’ of table ‘customers’, a few phone numbers are found to be invalid because letters are also included. Please return valid phone numbers that do not contain ANY letter [a - z].**

```
Select * from customers where phone not regexp '[a-z]'
```

Select…Distinct…

- **A product (productCode) can be purchased by many different customers, and thus appear multiple times in the table (orderdetails). Your boss wants to obtain a neat list of products (productCode) that have been purchased (avoiding duplicated values)!**

select distinct productCode from orderdetails

orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber
10 100	S18_1749	30	136	3
10 100	S18_2248	50	55,09	2
10 100	S18_4409	22	75,46	4
10 100	S24_3969	49	35,29	1
10 101	S18_2325	25	108,06	4
10 101	S18_2795	26	167,06	1
10 101	S24_1937	45	32,53	3

What does this query mean?

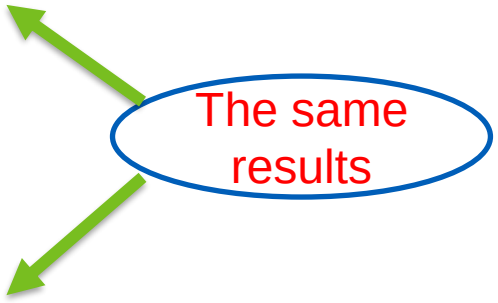
```
select distinct productCode, priceEach
 from orderdetails
```

orderNumber	productCode	quantityOrdered	priceEach	orderLineNumber
10 100	S18_1749	30	136	3
10 100	S18_2248	50	55,09	2
10 100	S18_4409	22	75,46	4
10 100	S24_3969	49	35,29	1
10 101	S18_2325	25	108,06	4
10 101	S18_2795	26	167,06	1
10 101	S24_1937	45	32,53	3
10 101	S24_2022	46	44,35	2
10 102	S18_1342	39	95,55	2
10 102	S18_1367	41	43,13	1
10 103	S10_1949	26	214,3	11
10 103	S10_4962	42	119,67	4

productCode	priceEach
S72_3212	43,68
S72_3212	44,23
S72_3212	44,77
S72_3212	46,96
S72_3212	47,5
S72_3212	48,05
S72_3212	48,59
S72_3212	49,14
S72_3212	51,32
S72_3212	51,87
S72_3212	52,42
S72_3212	52,96
S72_3212	53,51
S72_3212	54,05
S72_3212	54,6
S72_1253	39,73
S72_1253	40,22
S72_1253	41,22
S72_1253	41,71
S72_1253	42,71
S72_1253	43,2
S72_1253	43,7
S72_1253	44,2

Select…Group by…

- **"Group by" is similar to "Distinct" that returns unique records based on the column(s) specified.**
- **Select distinct productCode, priceEach from orderdetails**
- **Select productCode, priceEach from orderdetails group by productCode, priceEach**

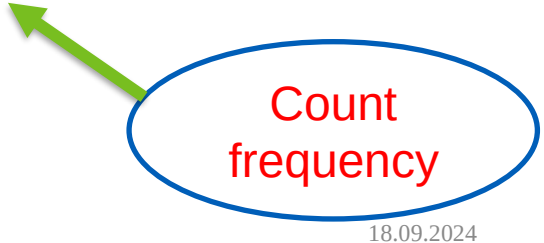


Select…Group by…

- **'Group by' creates virtual sub-tables that share some common characteristics.**

```
Select orderNumber, count(*)
from orderdetails
group by orderNumber
```

orderNumber	productCode
10,100	S18_1749
10,100	S18_2248
10,100	S18_4409
10,100	S24_3969
10,101	S18_2325
10,101	S18_2795
10,101	S24_1937
10,101	S24_2022
10,102	S18_1342
10,102	S18_1367
10,103	S10_1949



Select … ELT…

- **ELT() returns the *N*th element of the list of strings:**

```
Select ELT(1, 'ej', 'Heja', 'hej', 'foo');
```

ELT(1, 'ej', 'Heja', 'hej', 'foo')
ej

```
Select ELT(4, 'ej', 'Heja', 'hej', 'foo') as result;
```

result
foo

Think about this query

Name	Rank
David	3
John	1
Lily	2
Adam	4

Table `test`

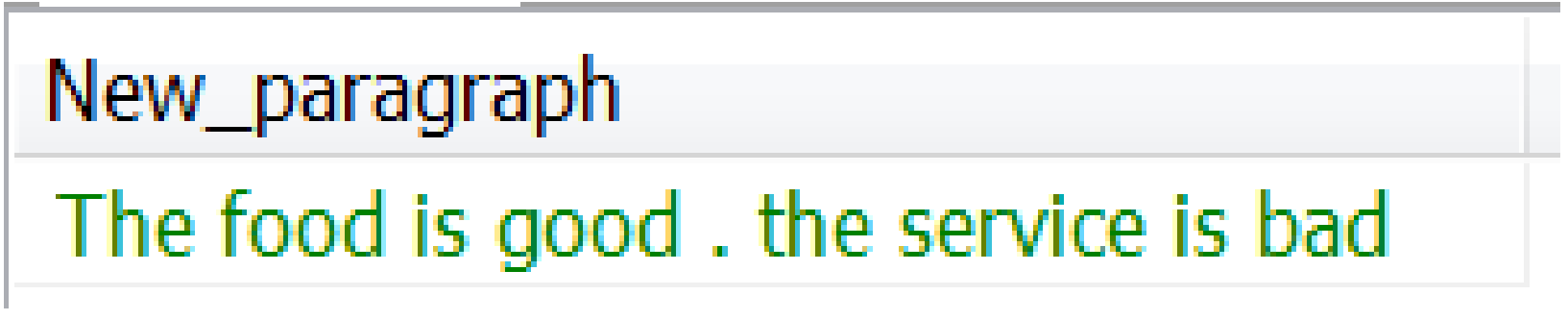
```
SELECT `Name`, `Rank`,
ELT(`Rank`, 'Ranks first',
            'Ranks second',
            'Ranks third',
            'Ranks forth')
as `rank`
from test;
```

Result of the above query

Name	Rank	rank
David	3	Ranks third
John	1	Ranks first
Lily	2	Ranks second
Adam	4	Ranks forth

Select… REPLACE(*str,from_str,to_str*) …

Select replace
('The food is good but the service is bad', 'but', '.')
as New_paragraph



What will be the result of the following command?

```
select replace (
replace('hello world', 'world', 'earth') , 'hello', 'hi')
as funny_test
```

http://presemo.aalto.fi/drm

Select … LENGTH(*str*) …

- **Possible applications: whether satisfied customers tend to write more words than those who are less satisfied?**

```
mysql> SELECT LENGTH('text');
        -> 4
```

```
select addressLine1, length(addressLine1) from customers
```

Popular problem

Select … Ltrim(*str*) …

- **Returns the string *str* with leading space characters removed.**

```
mysql> SELECT LTRIM('  barbar');
        -> 'barbar'
```

```
mysql> SELECT RTRIM('barbar   ');
        -> 'barbar'
```

Question!

- **In the table ‘customers’, please find the customers whose First Name contains five characters after removing empty space from the right side.**

customerNumber	customerName	contactLastName	contactFirstName
282	Souveniers And Things Co.	Huxley	Adrian
398	Tokyo Collectables, Ltd	Shimamura	Akiko
237	ANG Resellers	Camino	Alejandra
443	Feuer Online Stores, Inc	Feuer	Alexander

Answer

- **select** contactFirstName **from** customers
  where rtrim(contactFirstName) **like** '_____'

- **select** contactFirstName **from** customers
  where length(**rtrim**(contactFirstName)) =5

Select … LOWER(str) …

- **Returns the string str with all characters changed to lowercase.**

```
mysql> SELECT LOWER('QUADRATICALLY');
        -> 'quadratically'
```

- **Many software treats strings in a case-sensitive manner. Thus, it would be necessary to convert strings to lowercase letters before exporting data to these software for analysis.**

Select … SUBSTRING …

```
mysql> SELECT SUBSTRING('Quadratically',5);
        -> 'ratically'
mysql> SELECT SUBSTRING('foobarbar' FROM 4);
        -> 'barbar'
mysql> SELECT SUBSTRING('Quadratically',5,6);
        -> 'ratica'
mysql> SELECT SUBSTRING('Sakila', -3);
        -> 'ila'
mysql> SELECT SUBSTRING('Sakila', -5, 3);
        -> 'aki'
mysql> SELECT SUBSTRING('Sakila' FROM -4 FOR 2);
        -> 'ki'
```

SUBSTRING_INDEX(*str,delim,count*)

```
mysql> SELECT SUBSTRING_INDEX('www.mysql.com', '.', 2);
        -> 'www.mysql'
mysql> SELECT SUBSTRING_INDEX('www.mysql.com', '.', -2);
        -> 'mysql.com'
```

sent
first sentence. second sentence. third sentence.
first sentence. second sentence. third sentence.

Table name: test; Column name: sent

SUBSTRING_INDEX(*str,delim,count*)

- **Returns the substring from string *str* before *count* occurrences of the delimiter *delim*. If *count* is positive, everything to the left of the final delimiter (counting from the left) is returned. If *count* is negative, everything to the right of the final delimiter (counting from the right) is returned.**
- **SUBSTRING_INDEX() performs a case-sensitive match when searching for *delim*.**

Help me with this challenge!

"Excellent place in Helsinki"

Reviewed February 7, 2015

Realy nice place in the heart of Helsinki. Excellent food and wines combimed with passionate service. I can warmly recommend this place.

Was this review helpful? Yes 1

"Unique dinning experience"

Reviewed August 13, 2013 via mobile

Enjoyed all the dishes. Beatifully presented and tasted good. Service was very attentive. I didn't try the drink menu but 8 course meal wa unique dinning experience.

Was this review helpful? Yes

Hints

You can change * to be some other expression like SUBSTRING_INDEX

- **create table new_table as select * from old_table;**

You can create a test table using the following comment

- **create table test**
  (sent varchar(1000));
- **insert into test values**
  ('first sentence. second sentence. third sentence.');
- **insert into test values**
  ('first sentence. second sentence. third sentence.');

Solution I

- **create table** test2 **AS**
  (**select substring_index**(sent,'.',2) **as** first_two_sen **from** test);
- **Select substring_index**(first_two_sen,'.',-1) **as** second_sen **from** test2;

Challenge:
How to return the second sentence?

sent
first sentence. second sentence. third sentence.
first sentence. second sentence. third sentence.

Table name: test; Column name: sent

http://presemo.aalto.fi/drm

Hints

You can change * to be some other expression like SUBSTRING_INDEX

- **create table new_table as select * from old_table;**

You can create a test table using the following comment

- **create table test**
  (sent varchar(1000));
- **insert into test values**
  ('first sentence. second sentence. third sentence.');
- **insert into test values**
  ('first sentence. second sentence. third sentence.');

More sophisticated solution

```
select
substring_index(sent,'.',1) as sen1,
substring_index(substring_index(sent,'.',2),'.',-1)
as sen2,
substring_index(sent,'.',-2) as sen3
from test;
```

Reflection: How to do this research?

Detecting personality using words

- **http://www.psychometrics.cam.ac.uk/news/the-language-of-social-media**
- **Tal Yarkoni (2010). Personality in 100,000 Words: A large-scale analysis of personality and word use among bloggers, J Res Pers. 2010 Jun 1; 44(3): 363–373.**

Challenge II

DRMCourse.tripadvisor_data_for_handson_assignment_ONLY: 1,981 rows total (approximately), limited to 1,000

id	title	year_stayed	month_stayed	hotel_id	num_helpful_votes
145,862,422	"Good Place"	2,012	9	2,145,772	0
145,862,517	"Great Hotel With a Great Staff"	2,012	8	98,821	0
145,862,957	"Good quality budget hotel in expensive city"	2,012	11	81,042	1
145,863,456	"Just book it!"	2,012	11	113,329	5
145,863,801	"InterContinental Mark Hopkins"	2,012	11	115,623	0

Please calculate the number of words of titles!

Table name: test

http://presemo.aalto.fi/drm

https://support.zendesk.com/hc/user_images/qYB_ll7f1p5jVPNaaTFcGA.jpeg

- **Select** title,
  length(title) - **length**(**replace**(title, ' ', '')) + 1
  as number_of_words
  from test

Select … INSTR(str,substr) …

- **Returns the position of the first occurrence of substring *substr* in string *str*.**

```
mysql> SELECT INSTR('foobarbar', 'bar');
        -> 4
mysql> SELECT INSTR('xbar', 'foobar');
        -> 0
```

Select … LEFT(*str*,*len*) …

- **Returns the leftmost *len* characters from the string *str*.**

```
mysql> SELECT LEFT('foobarbar', 5);
        -> 'fooba'
```

```
mysql> SELECT RIGHT('foobarbar', 4);
        -> 'rbar'
```

Select … TRIM …

- **TRIM ([BOTH | LEADING | TRAILING FROM] *str*)**

```
mysql> SELECT TRIM('  bar   ');
        -> 'bar'
mysql> SELECT TRIM(LEADING 'x' FROM 'xxxbarxxx');
        -> 'barxxx'
mysql> SELECT TRIM(BOTH 'x' FROM 'xxxbarxxx');
        -> 'bar'
mysql> SELECT TRIM(TRAILING 'xyz' FROM 'barxxyz');
        -> 'barx'
```