Hands-on training session 5 (Answers)

- **Task 1**: In the orderdetails table at the car retailer database, please count the number of different prices that each product (productCode) have, and return the productCode of only those who have been sold in more than 5 different prices.

```
Select productCode, count(distinct(priceEach)) as freq from orderdetails group by
productCode having freq > 5
```

- **Task 2:**
  - Please count how many empty values there are in the column of Sub_product at the cfpb_complaints_2500.

    ```
    Select * from cfpb_complaints_2500 where Sub_product = '';
    or Select count(*) from cfpb_complaints_2500 where Sub_product = '';
    ```

  - Replace these empty value with NULL

    ```
    Update cfpb_complaints_2500 set Sub_product = NULL where Sub_product = ''
    ```

  - Drop the records that have NULL values in Sub_product.

    ```
    Delete from cfpb_complaints_2500 where Sub_product is NULL
    ```

  Note: after doing this task, please drop this incomplete table and import the complete data file from MyCourse again.

- **Task 3:** Which company received the largest amount of complaints from consumers on Wednesday? Assume the Data_received is the date that company received complaints.

```
Select Company, count(*) as freq from cfpb_complaints_2500 where dayname(Data_received)
= 'Wednesday' group by company order by freq desc
```

- **Task 4:** Which weekday did companies receive the largest amount of complaints?

```
Select dayname(Data_received), count(Data_received)as freq from cfpb_complaints_2500
group by dayname(Data_received) order by freq desc
```

- **Task 5:** Which month did the companies receive the largest amount of complaints?

```
Select monthname(Data_received), count(Data_received) as freq from
cfpb_complaints_2500 group by monthname(Data_received) order by freq desc
```

- **Task 6:** Considering only the first 7 days of each month, which month did the companies receive the largest amount of complaints?

```
Select monthname(Data_received), count(Data_received) as freq from cfpb_complaints_2500
where day(Data_received) < 8 group by monthname(Data_received) order by freq desc
```

- **Task 7:** in the table cfpb_complaints_2500, create a new column in the name of ‘datedifference’, this column contains the difference between Data_sent_to_company and Data_received.

```
Alter table cfpb_complaints_2500 add datedifference int;
```

```
Update cfpb_complaints_2500 set datedifference=
datediff(Data_sent_to_company,Data_received)
```

- **Task 8:** When will be date that you have living in this world for 10,000 days? How about 20,000 days?

Assume you are born on 1998-03-16:

```
Select date_add('1998-03-16', interval 10000 day);
Select date_add('1998-03-16', interval 20000 day)
```

- **Task 9:** How many days have you living in this word by today?

Assume you are born on 1998-03-16:

```
Select datediff(now(),'1998-03-16')
```

- **Task 10:** In the table **products** of the car retailing database, add two new columns of Price_difference, and Value_difference, given that:

  - **Price_difference = (MSRP-buyPrice)**

  - **Value_difference = (MSRP-buyPrice)* quantityInStock**

```
Alter table products add Price_difference decimal(10,2);
Alter table products add Value_difference decimal(10,2);

Update products set Price_difference = (MSRP-buyPrice);
Update products set Value_difference = (MSRP-buyPrice)*quantityInStock;
```

- **Task 11:** Remove two new columns of Price_difference, and Value_difference

You can actually do that by click on the HeidiSQL interface instead of using any comments.

```
Alter table products drop Price_difference;

Alter table products drop Value_difference
```

- **Task 12:** In the Classicmodels database, retrieve the names of the customers who purchased Ferrari in 2004!

Tips:

- In the table products, you will find productName including the word “Ferrari”
- In the table orderdetails, you will find the all the orderNumber that have ordered the products including “Ferrari”
- In the table orders, you will find the customerNumber connecting to each orderNumber as well as when the order was made.
- In the table customers, you will find the names of the customers connecting to their customerNumber.

Step 1: Select the orderNumber that has ordered “Ferrari”

Alternative 1:

```
Select orderNumber from orderdetails where productCode in (Select
distinct(productCode) from products where productName like '%Ferrari%')
```

You can also use this code (we will learn JOIN keyword next lecture):

Alternative 2:

```
Select orderNumber from orderdetails join products
on orderdetails.productCode=products.productCode
```

```
where products.productName like '%Ferrari%'
```

Step 2: Based on the above orderNumber, find the orders that were made in 2004 through the table orders.

You can create a temp table saving the orderNumber that you found from the step 1, like

```
create table temp as
(Select orderNumber from orderdetails where productCode in (Select distinct(productCode)
from products where productName like '%Ferrari%'))
```

Then from the table order, you find the ordersNumber and customerNumber that were made in 2004.

```
Select * from orders where orderNumber in (Select orderNumber from temp) and
year(orderDate) = 2004
```

- You may only simply Select the customerNumber from the table, like:

```
Select customerNumber from orders where orderNumber in (Select orderNumber from temp)
and  year(orderDate) = 2004
```

Step 3. You have now got the customerNumber who have ordered **"Ferrari"** in 2004, then it should be easy for you to find their name in the table **customers.**

```
Select customerName from customers where customerNumber in (Select customerNumber
from orders where orderNumber in (Select orderNumber from temp) and  year(orderDate)
= 2004)
```

- **Task 13:** In the table [cfpb_complaints_2500], count the frequencies of different **Issues** for different companies and show the starting date and ending date that different issues have been reported, and order the records based on first the company name from A-Z and then on frequencies in a descending manner. Save the results to a new table "Count_result"

```
CREATE TABLE Count_result AS
SELECT Company, Issue, COUNT(*) AS freq, MIN(Data_received) as startingday,
MAX(Data_received) as endingday
FROM cfpb_complaints_2500
GROUP BY Company, Issue
ORDER BY Company ASC, freq DESC
```

- **Task 14:** Based on the results in above table "Count_result", create a report like below.

```
SELECT GROUP_CONCAT('The company - ', Company, ' have issues related to ', Issue, ' for ',
freq, ' time(s) during ', startingday, ' and ', endingday, '.') AS sentence
FROM Count_result
GROUP BY Company, Issue
```

- **Task 15.** In the [tripadvisor_data_for_handson_assignment_ONLY] dataset, please evaluate whether and how the use of mobile device to write a review affects how people gave a rating to a hotel.

- The code to do the task is easy, but I hope you could think about why and how this happens. When review platform allows people to submit review via mobile phone, how will it affect the ratings on a hotel? This actually becomes a real business intelligence question that you need to get an answer by yourself. By reading the paper [Impact of Mobility and Timing on User-Generated Content] available in MyCourse, you may get some insights on your result.

```
Select via_mobile, avg(overall_rating), avg(service), avg(value), avg(rooms), avg(cleanliness),
avg(location), avg(sleep_quality), avg(reviewlength) from
tripadvisor_data_for_handson_assignment_ONLY group by via_mobile
```

- **Task 16.** Research shows that the review given at the same month of lodging vs. different month of lodging would be significantly different. Please find evidence from our data.

- the column ‘review_date’ provides the date of when the review is provided.

- the columns ‘year_stayed’ and ‘review_date’ provides the information of which month the accommodation in a hotel took place.

```
Select via_mobile, avg(overall_rating), avg(service), avg(value), avg(rooms), avg(cleanliness),
avg(location), avg(sleep_quality), avg(reviewlength) from
tripadvisor_data_for_handson_assignment_ONLY  where year(review_date) = year_stayed and
month(review_date) = month_stayed group by via_mobile
union
Select via_mobile, avg(overall_rating), avg(service), avg(value), avg(rooms), avg(cleanliness),
avg(location), avg(sleep_quality), avg(reviewlength) from
tripadvisor_data_for_handson_assignment_ONLY  where !(year(review_date) = year_stayed
and month(review_date) = month_stayed) group by via_mobile
```

- **Task 17.** Please import ‘tripadvisor_hotel_sample.sql’ to your database. This data table provides the information of hotels, while the [tripadvisor_data_for_handson_assignment_ONLY] table provides the information of reviews on hotels. Please get the list of hotels that are in [tripadivsor_hotel_sample] table, but not in [tripadvisor_data_ for_handson_assignment_ONLY] table.

```
Select hotel_id from tripadivsor_hotel_sample where hotel_id not in (Select hotel_id from
tripadvisor_data_for_handson_assignment_ONLY )
```

- **Task 18**. In the tripadvisor_hotel_sample table, please find out whether different hotels are actually using the same hotel name? Please sort the result by having the most frequently used name appear first.

```
Select name, count(*) fre from tripadivsor_hotel_sample group by name having fre > 1 order
by fre desc
```

- **Task 19.** In the tripadvisor_hotel_sample table, we find different hotels using the same name. However, whether different hotels but with the same name could appear in the same city?

```
Select locality, name, count(*) fre from tripadivsor_hotel_sample group by locality, name
order by fre desc
```

# Think about how you should name a hotel in the future!

- **Task 20.** Please write ONE query to count the frequency of different stars of hotels, calculate the average length of hotel name, average length of hotel address, and think about **why** there would be a difference among different stars of hotels.

```
Select hotel_class, count(*), avg(length(name)), avg(length(address)) from
tripadivsor_hotel_sample group by hotel_class
```

- **Task 21**. In the ‘orders’ table of ‘classicmodels’ database, we can find time (orderDate) of product order (orderNumber) was made by each customer (customerNumber). Could you please identify the **first** order (in term of orderDate) made by each customer in the database?

```
Select * FROM orders WHERE (orderDate, customerNumber) IN
(select MIN(orderDate),customerNumber FROM orders GROUP BY customernumber)
```

- **Task 22.** In the answer of task 21, what would be the results, if we replace “IN” with “NOT IN”.

The new results would be the sales order that exclude the first order of each customer.

- ➢ **Advance task 1:**

Task 1. If you check Mycourse page, you will find a paper [How do tourists evaluate Chinese hotels at different cities? Mining online tourist reviewers for new insights]. By quickly browsing the paper, you will find that the most popular words describe the conditions of tourism in a city. **However, how can we obtain the most popular words and their frequency across a large amount of review via the use of MySQL?** There were some analysis and data manipulation done in the paper, and now I would like you to duplicate the analysis via the use of our course dataset. Specifically, based on the review with maximal 10 words:

- Please ignore those titles like ‘<U+7ACB>’, which is probably caused by non-English like words.

1) Please provide me the most popular 30 words used in review titles. Remove punctuations like [. ! “”].

```
create table temp as (Select id, replace(replace(replace(replace(title, '!', ''), '“', ''), '.', ''), '”', '')
as title, author_location, overall_rating, (length(title) - length(replace(title, ' ', '')) + 1) as
wordcount from tripadvisor_data_for_handson_assignment_ONLY where (length(title) -
length(replace(title, ' ', '')) + 1)  < 11);
                                        # Note: there is a limit with regard to the length of a variable name.


create table tempword Select id, substring_index(title, ' ', 1) as title from temp
union all
Select id, substring_index(substring_index(title, ' ', 2), ' ', -1) as title from temp where
wordcount > 1
union all
Select id, substring_index(substring_index(title, ' ', 3), ' ', -1) as title from temp where
wordcount > 2
union all
Select id, substring_index(substring_index(title, ' ', 4), ' ', -1) as title from temp where
wordcount > 3
union all
Select id, substring_index(substring_index(title, ' ', 5), ' ', -1) as title from temp where
wordcount > 4
union all
Select id, substring_index(substring_index(title, ' ', 6), ' ', -1) as title from temp where
wordcount > 5
union all
Select id, substring_index(substring_index(title, ' ', 7), ' ', -1) as title from temp where
wordcount > 6
union all
Select id, substring_index(substring_index(title, ' ', 8), ' ', -1) as title from temp where
wordcount > 7
union all
```

```
Select id, substring_index(substring_index(title, ' ', 9), ' ', -1) as title from temp where
wordcount > 8
union all
Select id, substring_index(substring_index(title, ' ', 10), ' ', -1) as title from temp where
wordcount > 9
order by id
```

#Note: if you do not set up **where** condition, short title, e.g. a one-word title, will be computed multiple times in the command.

```
Select title, count(*) fre from tempword group by title order by fre desc limit 30
```

2) Please compare the ten most popular obtained from review title that are written by the authors whose ‘author_location’ is empty VS. those whose ‘author_location’ is not empty.

The code is also the same to the code for the last question, except for a small change in the where condition.

For author_location is empty

```
create table temp as (Select id, replace(replace(replace(replace(title, '!', ''), '“', ''), '.', ''), '”', '')
as title, author_location, overall_rating, (length(title) - length(replace(title, ' ', '')) + 1) as
wordcount from tripadvisor_data_for_handson_assignment_ONLY
where (length(title) - length(replace(title, ' ', '')) + 1)  < 11 and author_location = '');
```

For author_location is not empty

```
create table temp as (Select id, replace(replace(replace(replace(title, '!', ''), '“', ''), '.', ''), '”', '')
as title, author_location, overall_rating, (length(title) - length(replace(title, ' ', '')) + 1) as
wordcount from tripadvisor_data_for_handson_assignment_ONLY
where (length(title) - length(replace(title, ' ', '')) + 1)  < 11 and author_location != '');
```

3) What are the ten most popular words co-occurring with the word “great”? What are the ten most popular words co-occurring with the word “terrible”?

Similar to the above question, just change the where condition while the rest of code is the same.

For the words co-occur with the word “great”:

```
create table temp as (Select id, replace(replace(replace(replace(title, '!', ''), '“', ''), '.', ''), '”', '')
as title, author_location, overall_rating, (length(title) - length(replace(title, ' ', '')) + 1) as
wordcount from tripadvisor_data_for_handson_assignment_ONLY
where (length(title) - length(replace(title, ' ', '')) + 1)  < 11 and title like '%great%');
```

For the words co-occur with the word “terrible”:

```
create table temp as (Select id, replace(replace(replace(replace(title, '!', ''), '“', ''), '.', ''), '”', '')
as title, author_location, overall_rating, (length(title) - length(replace(title, ' ', '')) + 1) as
wordcount from tripadvisor_data_for_handson_assignment_ONLY
where (length(title) - length(replace(title, ' ', '')) + 1)  < 11 and title like '%terrible%');
```

4) What are the ten most popular words for 5-star overall rating review titles? What are the ten most popular words for 1-star overall rating review titles?

For the most popular words for 5-star overall rating review title.

```
create table temp as (Select id, replace(replace(replace(replace(title, '!', ''), '“', ''), '.', ''), '”', '')
as title, author_location, overall_rating, (length(title) - length(replace(title, ' ', '')) + 1) as
wordcount from tripadvisor_data_for_handson_assignment_ONLY
where (length(title) - length(replace(title, ' ', '')) + 1)  < 11 and overall_rating = 5);
```

For the most popular words for 1-star overall rating review title.

```
create table temp as (Select id, replace(replace(replace(replace(title, '!', ''), '“', ''), '.', ''), '”', '')
as title, author_location, overall_rating, (length(title) - length(replace(title, ' ', '')) + 1) as
wordcount from tripadvisor_data_for_handson_assignment_ONLY
where (length(title) - length(replace(title, ' ', '')) + 1)  < 11 and overall_rating = 1);
```