Practical script Nr. 6

# Sub-Queries

## Scalar sub-queries

SELECT column can queries with another SELECT statement, which returns a single column in a single row. For example – select network carrier name and how many reviews were there for that specific network, or average quality for each of the carriers.

SELECT CarrierName, (SELECT COUNT(\*) FROM NetworkQuality  
 WHERE Carrier.Carrier\_Id=NetworkQuality.Carrier\_ID)  
FROM Carrier

SELECT CarrierName, (SELECT COUNT(\*) FROM NetworkQuality  
 WHERE Carrier.Carrier\_Id=NetworkQuality.Carrier\_ID)  
 , (SELECT AVG(Quality\*1.0) FROM NetworkQuality  
 WHERE Carrier.Carrier\_Id=NetworkQuality.Carrier\_ID)  
FROM Carrier

This shows how you can use the query results as SELECT query column. In case the sub-query does not return single value in a single row, there will be an error.

An alternative way of writing the same query.

SELECT CarrierName, COUNT(\*)   
FROM Carrier, NetworkQuality  
WHERE Carrier.Carrier\_ID=NetworkQuality.Carrier\_ID  
GROUP BY CarrierName

## Using Sub-query results as a table for further queries

You can use a query result as a virtual table for further queries

SELECT CarrierName, Skaits  
FROM Carrier, (SELECT Carrier\_ID, COUNT(\*) AS Skaits FROM   
 NetworkQuality GROUP BY Carrier\_ID) A  
WHERE Carrier.Carrier\_ID=A.Carrier\_ID

A sub-query generates a resulting dataset, which is used a virtual table with columns, which can be queried.

## Comparison of sets

You can use sub-queries in WHERE condition in case you need to check if specific record exists or does not exist in pre-defined datasets. For example selecting carriers, which have network quality of 5.

SELECT CarrierName  
FROM Carrier  
WHERE EXISTS (SELECT \* FROM NetworkQuality   
 WHERE Carrier.Carrier\_ID=NetworkQuality.Carrier\_ID   
 AND Quality=5)

Or selecting carriers with not review with network quality = 5

SELECT CarrierName  
FROM Carrier  
WHERE NOT EXISTS (SELECT \* FROM NetworkQuality   
 WHERE Carrier.Carrier\_ID=NetworkQuality.Carrier\_ID   
 AND Quality=5)

Selecting carriers which average network quality > 4

SELECT CarrierName  
FROM Carrier  
WHERE (SELECT AVG(Quality\*1.0) FROM NetworkQuality  
 WHERE Carrier.Carrier\_Id=NetworkQuality.Carrier\_ID)>=4

Some more examples of using sub-queries in selects.

SELECT \* FROM PlaceFeedback, Feedback  
WHERE PlaceFeedback.Feedback\_ID=Feedback.Feedback\_ID  
AND Place\_ID IN (SELECT To\_ID FROM TravelMethod)

SELECT \* FROM PlaceFeedback, Feedback  
WHERE PlaceFeedback.Feedback\_ID=Feedback.Feedback\_ID  
AND Place\_ID NOT IN (SELECT To\_ID FROM TravelMethod)

Randomly selects a Quality from the sub-query result for comparison

SELECT \* FROM NetworkQuality   
WHERE Quality> SOME (SELECT Quality FROM NetworkQuality)

Perform comparison to all of the values in a sub-query.

SELECT \* FROM NetworkQuality   
WHERE Quality>= ALL (SELECT Quality FROM NetworkQuality)

You can use the comparison operators with both SOME and ALL expressions.

SELECT Carrier\_ID FROM NetworkQuality   
GROUP BY Carrier\_ID  
HAVING AVG(Quality\*1.)>=ALL (SELECT AVG(Quality\*1.0) FROM NetworkQuality   
 GROUP BY Carrier\_ID)

Select carrier names, with maximum average quality

SELECT CarrierName  
FROM Carrier,  
(SELECT Carrier\_ID FROM NetworkQuality   
GROUP BY Carrier\_ID  
HAVING AVG(Quality\*1.)>=ALL (SELECT AVG(Quality\*1.0) FROM NetworkQuality   
 GROUP BY Carrier\_ID)) A  
WHERE A.Carrier\_ID=Carrier.Carrier\_ID

Practical tasks

1. SELECT carriers with no network quality
2. SELECT users, who did not provide a single network quality feedback
3. SELECT carriers with most of the network quality feedbacks
4. SELECT users, who evaluated carriers with most 5 stars
5. SELECT place types with most places
6. SELECT network quality feedback for most popular place types
7. SELECT travel method types which nobody uses
8. SELECT places of interest, to which nobody travels to