

## Direct SQL

### Tools

- Database schema from Data Model Extension task.
- ANSI SQL vs Oracle SQL, PL/SQL.
- Oracle sqlDeveloper

General rule is to study standard (ANSI) SQL features first and then Oracle SQL specifics. That is, try to write the query within ANSI SQL features and if stuck – seek for Oracle SQL bonuses.

### Requirements

As usual, all artifacts should be stored in version control system.

\* – marks complex queries.

1. Develop a query to calculate the number of news, written by each author, the average number of comments per news for a current author and the most popular tag, referred to author news. All these information must be output in one single result set. Based on these query create a **custom db view**.
2. Write **SQL statement** to select author names who wrote more than 3 000 characters, but the average number of characters per news is more than 500. Think about the shortest statement notation.
3. Create **custom db function** that will return the list of all tags referred to a current news, concatenated by specified separator character. Function must accept the news id and separator character as input parameters and return a single string as a result of tag values concatenation.
4. Develop **single SQL statement** that will return the list of all available news (news id, news title columns) and one more column that will display all concatenated tags values, available for current news as a single string.  
Make two versions of statement:
  - a. By using *previously developed custom function (#3)*.
  - b. \* By using *Oracle 10g DB features*.
5. \* Make a writers competition cross-map. Create a **statement** that will generate random authors distribution by competition pairs.  
Statement will generate a list of author names pairs selected for a single round of tournament, displayed as two separate columns. Each author must be presented in resulting set (*in both columns*) once only. (The total number of authors must be *even*. Do not use custom functions during current implementation.)

6. \* Create custom LOGGING **db table**. The following corresponding columns must be created:
- a. Record insert date;
  - b. Referenced Table Name - Table Name where new record was inserted;
  - c. Description - The list of key-value pairs, separated by semicolon. Note: empty values and their column names must be omitted.

Add required changes to your DB Schema to **populate current table columns each time when new record was inserted to any db table** (in a scope of your schema).

7. Prepare a **SQL statement** to output total/available space within each tablespace. Calculate "Used %" column value.
8. Prepare a **SQL Statement** to calculate each table size in your schema.