

Лабораторна робота № 2

Тема: Конструювання реляційної бази даних в СКБД *Oracle*

Загальні завдання під час *Online*-заняття

(виконуються лише під час *Online*-заняття з можливістю демонструвати їх всім учасникам)

Етап 1. Особливості роботи з сервером СКБД *Oracle* через утиліту *SQLPlus*

Всі завдання виконуються в командному рядку з використанням утиліти *sqlplus*.

Всі команди та повідомлення системи зберегти у файлі *Прізвище_2_SQLPlus.sql*

Приклади команд представлено у файлі –

https://drive.google.com/file/d/1PJH_iiKxX11jHOo1XAIIdHqZ39QFCsk3M

1.1 Початок роботи з локальним сервером *Oracle*

1. Запустити термінал командного рядку від імені системного адміністратора ОС (для ОС *Windows* - команда *cmd*).
2. Змінити умовно забутий пароль головного адміністратора *SYS* та адміністратора *SYSTEM*, використовуючи команду *ALTER USER*.
3. Виконати команду зупинки сервера *Oracle*.
4. Виконати команду запуску сервера та визначити *SGA*-характеристики розмірів пам'яті
5. Встановити з'єднання з *CDB* БД від імені користувача *SYSTEM*, послідовно використовуючи наступні три варіанти (*CDB = SID = «XE»*, адреса = *localhost*, порт = *1521*):
 - 1) пряме підключення (без *TNS*) через команду *connect*
 - 2) пряме підключення з командного рядку
 - 3) *TNS*-підключення через логін/пароль/адреса
 - 4) *TNS*-підключення через логін/пароль/адреса/*SID*
 - 5) *TNS*-підключення через логін/пароль/адреса/порт/*SID*
6. Встановити з'єднання з *PDB* БД від імені користувача *SYSTEM*, використовуючи наступні варіанти (*PDB = «XEPDB1»*):
 - 1) пряме підключення через команду *alter session*
 - 2) *TNS*-підключення логін/пароль/адреса/порт/*PDB*

1.2 Створення нового користувача на віддаленому сервері *Oracle 18c XE*

На віддаленому сервері вже створено користувача з правами адміністратора.

1. У терміналі командного рядку встановити з'єднання з віддаленим сервером:

- ім'я користувача - збігається з ім'ям вашого *GitHub*-репозиторію
- пароль користувача = `oracle18xe`
- адреса = `91.219.60.189`
- порт = `1521`
- *PDB* = *XEPDB1*

2. Змінити пароль користувача.

3. Створити нового користувача, ім'я, якого співпадає з вашим прізвищем у транслітерації з додавання слова `_test`

4. В окремому терміналі встановити з'єднання з БД від імені нового користувача та перевірити реакцію системи.

5. У терміналі користувача-адміністратора надати новому користувачу право встановлення з'єднання з БД

6. Повторити крок 4 та перевірити реакцію системи.

7. У терміналі нового користувача виконати команду отримання імені поточного користувача.

8. У терміналі користувача-адміністратора надати новому користувачу права на керування схемою даних користувача.

9. Завершити роботу з терміналом

Етап 2. Основи роботи з інструментальним середовищем *SQL Developer*

1. Розпакувати архів з *SQL Developer* на локальному комп'ютері та запустити програму

2. Налаштувати з'єднання, використовуючи тип з'єднання *Basic*:

- назва з'єднання – *labwork2*
- ім'я користувача та пароль, які використано у завданні 1.3
- *Hostname* = `91.219.60.189`
- *Port* = `1521`
- *Service name* = *XEPDB1*

3. Перевірити з'єднання з БД (кнопка *Test*) та зберегти

4. Встановити з'єднання з БД

Етап 3. Конструювання реляційної БД з урахуванням особливостей СКБД *Oracle*

Вихідними даними для виконання етапу роботи є приклад файл-скриптів з *DDL*-командами створення таблиць БД у відповідності зі схемою реляційної моделі БД, яка була розглянута в лабораторній роботі №1:

- файл-скрипт *DDL*-команд опису таблиць БД –

<https://drive.google.com/file/d/1P86JM-fCjozj11CqT7KDcmNnse-AI5pD>

- файл-скрипт *DDL*-команд опису обмежень цілісності таблиць –

<https://drive.google.com/file/d/1P5ZM3ep3nKZoC6lX2eG0O41tuOYsQBL3>

1. Додати *DDL*-команди опису таблиць БД:

- таблиці було вами додано до реляційної моделі БД під час *Online*-заняття;
- *DDL*-команди повинні містити лише імена стопвчиків та їх типи (без опису обмежень цілісності);
- використовувати лише типи даних СКБД *Oracle*.

DDL-команди зберегти в окремому файлі-скрипті з ім'ям *Прізвище_2_DDL.sql*

2. Завантажити створений скрипт до віддаленої БД під керуванням нового користувача, створеного на попередніх етапах. Перевірити завантаження скрипту на можливі повідомлення про помилки (крім помилок, що виникають через спробу видалення ще не існуючих таблиць).

3. Спроекувати *DDL*-команди опису обмежень цілісності таблиць:

- потенційного ключа;
- первинного ключа;
- зовнішнього ключа;
- контролю значень даних (*CHECK*).

4. Всі команди зберегти в окремому файлі-скрипті з ім'ям *Прізвище_2_Constr.sql*

5. Завантажити створений скрипт у БД, перевіривши його на можливі повідомлення про помилки.

Індивідуальні завдання за межами *Online*-заняття

Етап 4. Конструювання реляційної БД з урахуванням особливостей СКБД *Oracle*.

Вихідними даними для виконання етапу роботи є схема реляційної моделі БД, створеної в лабораторній роботі №1.

Всі дії виконати від імені вашого користувача, ім'я якого співпадає з вашим Git-репозиторієм.

Крок 1. Описати *DDL*-команди створення таблиць БД, які включають лише імена стовпчиків та їх типи (без опису обмежень цілісності). Використовувати типи даних СКБД *Oracle*.

DDL-команди зберегти в окремому файлі-скрипті з ім'ям *TableCreate.sql*

Крок 2. На початку файл-скрипту додати *DDL*-команди видалення таблиць у порядку, зворотному порядку їх створення.

Крок 3. Завантажити створений скрипт у локальній або віддаленій у БД під керуванням нового користувача, створеного на попередніх етапах. Перевірити завантаження скрипту на можливі повідомлення про помилки (крім помилок, що виникають через спробу видалення ще не існуючих таблиць).

Етап 5. Конструювання обмежень цілісності таблиць БД.

Крок 1. Спроекувати *DDL*-команди опису обмежень цілісності таблиць, створених у попередньому етапі:

- потенційного ключа;
- первинного ключа;
- зовнішнього ключа;
- контролю значень даних (*CHECK*).

Крок 2. Всі команди зберегти в окремому файлі-скрипті з ім'ям *TableConstraints.sql*

Крок 3. Завантажити створений скрипт у БД, перевіривши його на можливі повідомлення про помилки.

Етап 6. Документування результатів роботи на Веб-сервісі *GitHub*

Крок 1. У вашому *GitHub*-репозиторії створено *Issue* з назвою «*tasks-of-laboratory-work-2*», в якому розміщено завдання з лабораторної роботи №2, яке пропонує задокументувати результати рішень цієї роботи.

Розпочинаючи роботу над документуванням рішень лабораторної роботи, необхідно:

- 1) підключити до *Issue* ваш *GitHub-project* (правий розділ «*Projects*» сторінки з *Issue*);
- 2) змінити статус *Issue* з «*Todo*» на «*In progress*», автоматично перевівши *Scrum*-картку з цим *Issue* на *Scrum*-дошку «*In progress*»;
- 3) створити нову *Git*-гілку з назвою, яка відповідає назві *Issue*, наприклад, «*tasks-of-laboratory-work-2*» (використовується посилання «*Create a branch*» у правому розділі «*Development*» сторінки з *Issue*).

Крок 2. Після створення *Git*-гілки перейти до цієї гілки для створення оновлень файлів *Git*-репозиторію.

Крок 3. У новій гілці *Git*-репозиторію створити каталог з назвою «*2-DDL*» (кнопка «*Add file*» - «*Create new file*»), при створенні якого одночасно створити файл *README.md* з першим рядком «*DDL-команди зі створення таблиць БД*» зі стилем «Заголовок 3-го рівня» мови розмітки *Markdown* (три символи решітка ###).

Крок 4. Розмістити в каталозі «*2-DDL*» *GitHub*-репозиторія файли *TableCreate.jpg* та *TableConstraints.sql*

Крок 5. Виконати запит *Pull Request*, розпочавши процес *Code Review*.

Під час створення *Pull Request* необхідно вказати:

- *Reviewers* = Oleksandr Blazhko, Maria Glava;
- *Labels* = *enhancement* (*New feature or request*);
- *Projects* = посилання на *GitHub-project*.

Завершення процесу *Code Review* відбудеться до початку нового заняття, після чого викладач закриє *Issue*, завершуючи процес виконання завдань з лабораторної роботи.