Mohamed Khider University - Biskra **2024/2025**
Department of Computer Science **Level: Master 1**
Module: Advanced Databases & Big Data **Option : RTIC**

Lab Work 1

Consider the following relational database

- **Products Table:**
  - IDProduit (primary key)
  - ProduitName
  - price
  - Stock
- **Clients Table:**
  - IDClient (primary key)
  - ClientName
  - Address
  - Email
- **Commandes Table:**
  - IDCommande (primary key)
  - Order Date
  - IDClient (Foreign key referencing the Clients table)
  - TotalAmount
- **Details Order Table:**
  - IDDetail (primary key)
  - IDCommande (Foreign key referencing the Orders table)
  - IDProduct (Foreign key referencing the Products table)
  - Quantity

Create the SQL database and Answer the following questions:

1. Select all products

2. Select the name and address of all customers

3. Select the details of all orders placed in the last month

4. Calculate the total amount of the last order for a given customer

5. Find out of stock products

6. Select the customer names and product name for each order, as well as the quantity ordered

7. Update the price of a product

8. Delete a customer and all their associated orders

Lab Work 2

An online video streaming company needs a database to manage users, videos, categories, and views.

The following tables are given:

- **Users** (UserID, Name, Email, RegistrationDate)
- **Videos** (VideoID, Title, URL, PublicationDate)
- **Categories** (CategoryID, CategoryName)
- **Views** (ViewID, UserID, VideoID, ViewDate)

Create the SQL database and Answer the following questions:

1. Find the name and email of all users who have watched a specific video (e.g., the video with VideoID = id).
2. Count how many videos each user has watched.
3. Find the videos that have been viewed more than 1,000 times.
4. Calculate the average number of views per video.
5. For each category, find the total number of videos and the most recently published video in that category.
6. Delete all users who registered more than two years ago and have not watched any videos.
7. Create an index on the **PublicationDate** column of the **Videos** table to speed up date-based queries.

Lab Work 3

Exercise 1:

Suppose we have two tables in a database: **clients** (ID, Name, City) and **orders** (ID, client_id, product, quantity).

- Write an SQL query to retrieve the client’s name, city, and the total quantity of products ordered by that client.
- Write an SQL query to retrieve the client’s name, the product they have ordered the most, and the number of times they have ordered it.
- Write another query to retrieve the most ordered product by all clients, along with the number of times it has been ordered.
-

Exercise 2

Suppose we have three tables in a database: **clients** (ID, Name, City), **orders** (ID, client_id, product, quantity), and **products** (ID, Product_Name).

- Write an SQL query to retrieve the client’s name, the product name, and the total quantity of that product ordered by the client.

Exercise 3:

Suppose we have two tables in a database: **employees** (ID, Name, Department_ID) and **departments** (ID, Name).

- Write an SQL query to retrieve the department name, the total number of employees in that department, and the number of employees whose name starts with the letter 'E'.

Lab Work 4

The goal of this practical work is to design, create, and manipulate a large-scale relational database using an open dataset from (Kaggle). You will import data, define relationships between tables, and perform complex SQL queries.

1. Choose a Dataset from Kaggle.com and select a **large** and **Big** dataset that is suitable for structuring into multiple related tables.

- Examples of datasets:
  - E-commerce transactions
  - Movie ratings and reviews
  - Financial transactions
  - Healthcare data
  - Social media interactions

2. Database Creation & Data Import

- Use **PostgreSQL**, **MySQL**, or **SQLite** to create your database.
- Write SQL scripts to create tables with appropriate data types and constraints.
- Import data from CSV files into the corresponding tables.

4. Data Analysis

Write and execute the following SQL queries:

- Perform aggregations like **SUM**, **AVG**, **COUNT**, and **MAX/MIN**.
- Implement indexing on large tables to optimize query performance.

5. Interfaces

Create a web-based interface using HTML, CSS, and JavaScript to interact with an SQL database. You will design a simple front-end that allows users to add, modify, delete, and view records in the database.

Lab Work 5

The goal of this practical work is to develop an intelligent web interface that enhances user experience by integrating **autocomplete** and **autocorrect** functionalities for queries. The system will use **Levenshtein distance** to suggest corrections for misspelled inputs and provide real-time query suggestions.

1. Database Setup
   - Use MySQL or PostgreSQL to create a database.
   - Define a table that will be used for search queries (e.g., users, products, books, etc.).
   - Populate the table with a large dataset for testing autocomplete and autocorrect functionalities.
   -
2. Implement Autocomplete (LIKE Query or Full-Text Search)
3. Implement Autocorrect (Levenshtein Distance)
4. Optimize database queries using **indexes** and **trigram-based similarity matching**.

Lab Work 6
Big Data Processing with Hadoop

The objective of this practical work is to introduce students to Hadoop, a powerful framework for distributed storage and processing of large datasets. Students will set up a Hadoop environment, process data using HDFS (Hadoop Distributed File System), and perform MapReduce operations to analyze a dataset.

1. Download and install **Hadoop** (Single-node), Configure core-site.xml, hdfs-site.xml, and mapred-site.xml.
2. Download a dataset (e.g., a Kaggle dataset like movie reviews, stock market data, or web logs).
3. Word Count Example in Java : Implement a MapReduce job that counts word occurrences in a dataset.
4. Download and process a large dataset (e.g., customer reviews, social media posts).
    a. Use HDFS to store the dataset.
    b. Implement a MapReduce job to analyze trends (e.g., most common words, user activity).

Lab Work 7

NoSQL Database Management with MongoDB

The objective of this practical work is to introduce students to **MongoDB**, a NoSQL database used for handling large amounts of unstructured and semi-structured data. Students will learn how to:

- and Install and configure MongoDB
- Create and manage collections and documents
- Perform CRUD (Create, Read, Update, Delete) operations
- Execute complex queries using MongoDB's aggregation framework

1. Install MongoDB on your system (MongoDB Download)
2. Create a Database
3. Manage Collections & Documents :
   i. Insert Data into a Collection
   ii. Retrieve Data
   iii. Delete Documents
   iv. Update Documents
4. Integrating MongoDB with a Web Application

Lab Work 8
Big Data Storage and Processing with HBase

The goal of this practical work is to introduce students to **HBase**, a distributed, scalable, and NoSQL database that runs on top of **Hadoop HDFS**. Students will learn how to:

- Set up an HBase environment
- Create and manage tables
- Perform CRUD (Create, Read, Update, Delete) operations
- Execute advanced queries using HBase Shell and Java API

Instructions

1. Download and **Install HBase (Standalone)**
2. Create a Table in HBase Shell
    a. Insert Data into the Table
    b. Retrieve Data from the Table
    c. Update Data
    d. Delete Data
3. Set Up a Java Project with HBase