**Лабораторна робота №2**

**Створення додатку бази даних, орієнтованого на взаємодію з СУБД**

**PostgreSQL**

Гаценко Марія, КП-02

**Мета роботи**

Здобуття вмінь програмування прикладних додатків баз даних PostgreSQL.

**Постановка завдання**

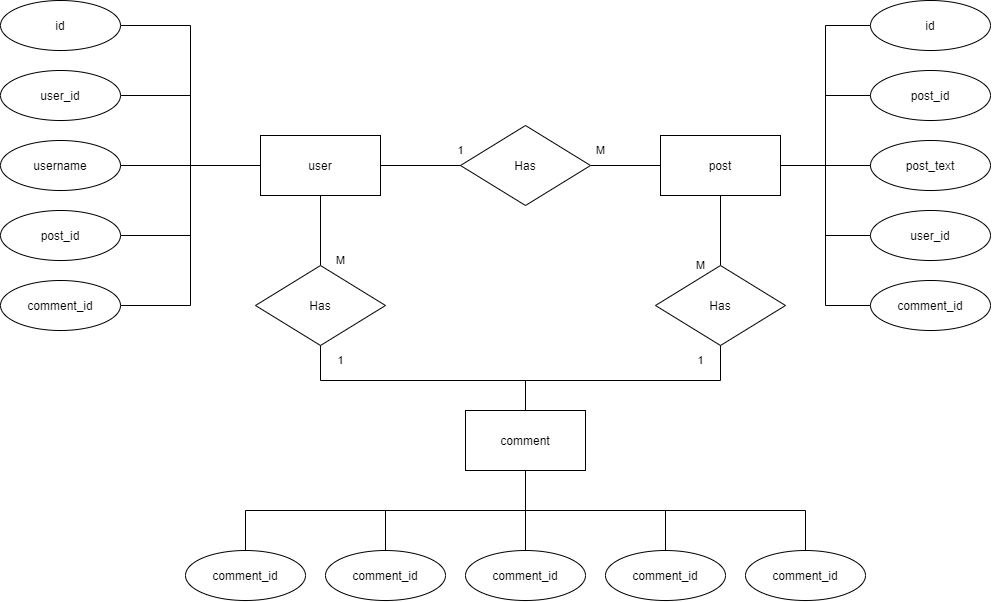
1. Реалізувати функції внесення, редагування та вилучення даних у таблицях бази даних, створених у лабораторній роботі No1, засобами консольного інтерфейсу.

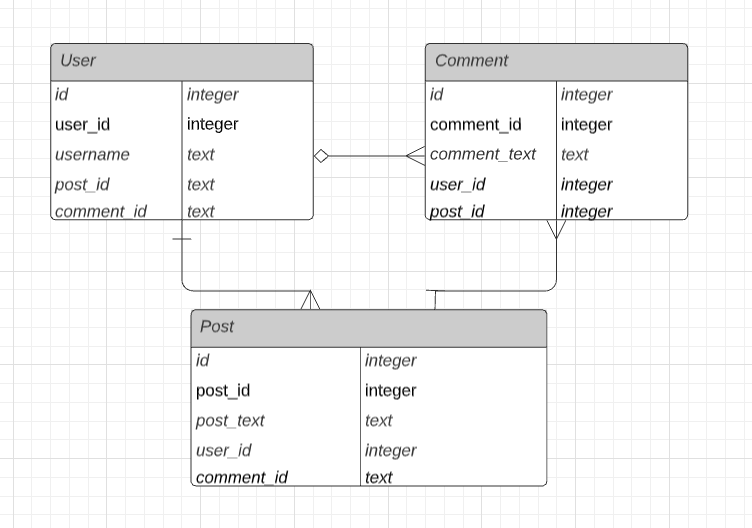
2. Передбачити автоматичне пакетне генерування «рандомізованих» даних у базі.

3. Забезпечити реалізацію пошуку за декількома атрибутами з двох та більше сутностей одночасно: для числових атрибутів – у рамках діапазону, для рядкових – як шаблон функції LIKE оператора SELECT SQL, для логічного типу – значення True/False, для дат – у рамках діапазону дат.

4. Програмний код виконати згідно шаблону MVC (модель-подання-контролер).

**Графічний файл моделі бази даних**

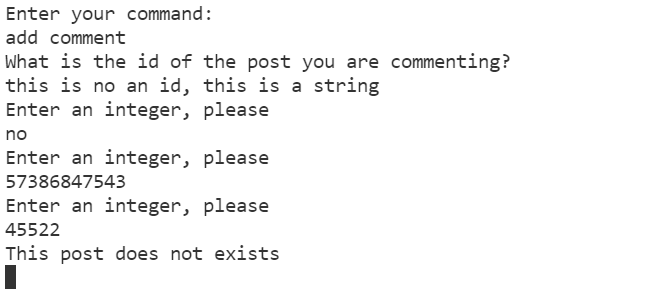
****

**Структура нормалізованої бази даних**

**Тексти коду програм**

**Перше завдання**

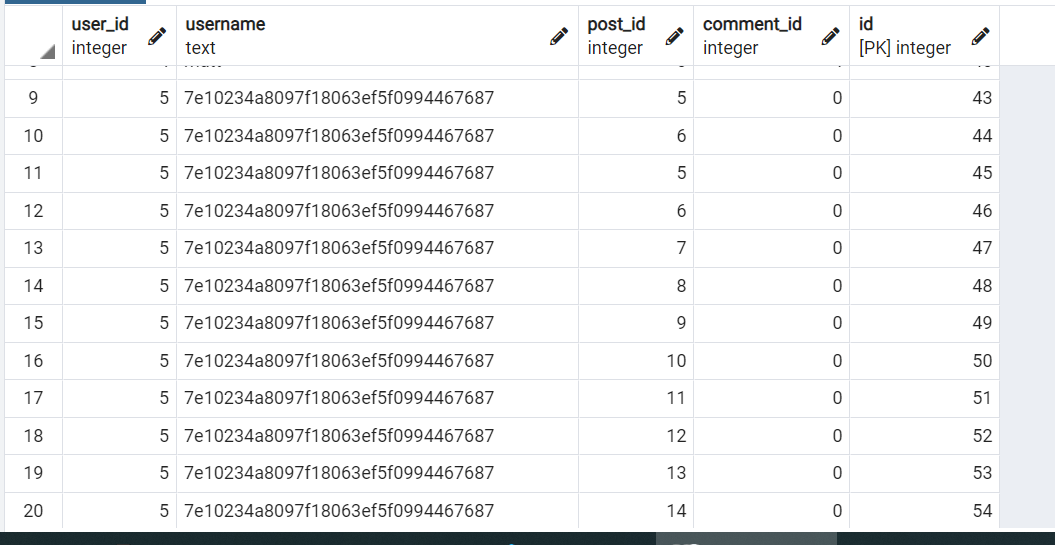
|  |
| --- |
| **Insert** |
| **Вставка поста**  public bool Insert(Post post)          {              Console.WriteLine("insert post");              string command = $"INSERT INTO posts (post\_id, post\_text, user\_id, comment\_id) VALUES ({post.postId}, '{post.postText}', {post.userId}, {post.commentId});";              NpgsqlCommand cmd = new NpgsqlCommand(command, connection);              cmd.ExecuteScalar();              UserRepo userRepo = new UserRepo(connection);              User user = userRepo.GetById(post.userId);              if (user.username == null)              {                  user.userId = post.userId;                  user.commentId = 0;                  connection.Close();                  connection.Open();                  command = $"SELECT md5(random()::text);";                  cmd = new NpgsqlCommand(command, connection);                  user.username = cmd.ExecuteScalar().ToString();                  connection.Close();                  connection.Open();              }              user.postId = post.postId;              userRepo.Insert(user);              return true;          }  **Вставка коментаря**  public bool Insert(Comment comment)          {              Console.WriteLine("insert comm");              string command = $"INSERT INTO comments (comment\_id, comment\_text, user\_id, post\_id) VALUES ({comment.commentId}, '{comment.commentText}', {comment.userId}, {comment.postId});";              NpgsqlCommand cmd = new NpgsqlCommand(command, connection);              //if (cmd.ExecuteScalar() == null) return false;              cmd.ExecuteScalar();              UserRepo userRepo = new UserRepo(connection);              User user = userRepo.GetById(comment.userId);              if (user.username == null)              {                  user.userId = comment.userId;                  user.postId = 0;                  connection.Close();                  connection.Open();                  command = $"SELECT md5(random()::text);";                  cmd = new NpgsqlCommand(command, connection);                  user.username = cmd.ExecuteScalar().ToString();                  connection.Close();                  connection.Open();              }              user.commentId = comment.commentId;              userRepo.Insert(user);              PostRepo postRepo = new PostRepo(connection);              Post post = postRepo.GetById(comment.postId);              if (post.postText == null)              {                  post.postId = comment.postId;                  connection.Close();                  connection.Open();                  command = $"SELECT md5(random()::text);";                  cmd = new NpgsqlCommand(command, connection);                  post.postText = cmd.ExecuteScalar().ToString();                  connection.Close();                  connection.Open();                  post.userId = userRepo.GetNewId();              }              post.commentId = comment.commentId;              postRepo.Insert(post);              return true;          }  **Вставка користувача**  public bool Insert(User user)          {              string command = $"INSERT INTO users (user\_id, username, post\_id, comment\_id) VALUES ({user.userId}, '{user.username}', {user.postId}, {user.commentId});";              NpgsqlCommand cmd = new NpgsqlCommand(command, connection);              if (cmd.ExecuteScalar() == null) return false;              return true;          }  **Обробка запитів на вставку**  public bool AddPost()          {              WriteLine("Type your post:");              string text = ReadLine();              Post post = new Post(postRepo.GetNewId(), text, mainUser.userId, 0);              return postRepo.Insert(post);          }          public bool AddComment(out bool exited)          {              WriteLine("What is the id of the post you are commenting?");              string strId = "";              Post post = new Post();              int id = 0;              exited = false;              while (true)              {                  strId = ReadLine();                  if (strId.Equals("exit", StringComparison.InvariantCultureIgnoreCase))                  {                      exited = true;                      break;                  }                  else if (!int.TryParse(strId, out id))                  {                      WriteLine("Enter an integer, please");                      continue;                  }                  post = postRepo.GetById(id);                  if (post.postText == null)                  {                      WriteLine("This post does not exists");                      continue;                  }                  break;              }              if (exited) return false;              WriteLine("\r\nType your comment:");              string text = ReadLine();              Comment comment = new Comment(commentRepo.GetNewId(), text, mainUser.userId, post.postId);              bool res = commentRepo.Insert(comment);              return res;          }  **Валідація даних, що ввів користувач**  void ProcessPostIdInput(out bool exited, out int id)          {              exited = false;              id = 0;              while (true)              {                  string strId = ReadLine();                  if (strId.Equals("exit", StringComparison.InvariantCultureIgnoreCase))                  {                      exited = true;                      break;                  }                  else if (!int.TryParse(strId, out id))                  {                      WriteLine("Enter an integer, please");                      continue;                  }                  Post post = postRepo.GetById(id);                  if (post.postText == null)                  {                      WriteLine("This post does not exists");                      continue;                  }                  break;              }          }          void ProcessCommentIdInput(out bool exited, out int id)          {              exited = false;              id = 0;              while (true)              {                  string strId = ReadLine();                  if (strId.Equals("exit", StringComparison.InvariantCultureIgnoreCase))                  {                      exited = true;                      break;                  }                  else if (!int.TryParse(strId, out id))                  {                      WriteLine("Enter an integer, please");                      continue;                  }                  Comment comment = commentRepo.GetById(id);                  if (comment.commentText == null)                  {                      WriteLine("This post does not exists");                      continue;                  }                  break;              }          } |
| **Edit** |
| **Редагування поста**   public bool Update(int postId, Post updatedPost)          {              string command = $"UPDATE posts SET post\_text = '{updatedPost.postText}' WHERE post\_id = {postId};";              NpgsqlCommand cmd = new NpgsqlCommand(command, connection);              cmd.ExecuteReader();              return true;          }  **Редагування коментаря**  public bool Update(int commentId, Comment updatedComment)          {              string command = $"UPDATE comments SET comment\_text = '{updatedComment.commentText}' WHERE comment\_id = {commentId};";              NpgsqlCommand cmd = new NpgsqlCommand(command, connection);              cmd.ExecuteReader();              return true;          }  **Обробка запитів на редагування**  public bool EditPost(out bool exited)          {              WriteLine("What post would you like to edit?");              int id = 0;              Post post = new Post();              ProcessPostIdInput(out exited, out id);              if (exited) return false;              post = postRepo.GetById(id);              WriteLine("Write new text for a post:");              string text = ReadLine();              post.postText = text;              return postRepo.Update(id, post);          }          public bool EditComment(out bool exited)          {              WriteLine("What comment would you like to edit?");              int id = 0;              Comment comment = new Comment();              ProcessCommentIdInput(out exited, out id);              comment = commentRepo.GetById(id);              if (exited) return false;              WriteLine("Write new text for a comment:");              string text = ReadLine();              comment.commentText = text;              return commentRepo.Update(id, comment);          } |
| **Delete** |
| **Видалення поста**  public bool DeleteById(int id)          {              string command = $"DELETE FROM posts WHERE post\_id = {id};";              NpgsqlCommand cmd = new NpgsqlCommand(command, connection);              int res = cmd.ExecuteNonQuery();              UserRepo userRepo = new UserRepo(connection);              userRepo.DeleteByPostId(id);              if (res == -1) return false;              return true;          }  **Видалення коментаря**  public bool DeleteById(int id)          {              string command = $"DELETE FROM comments WHERE comment\_id = {id};";              NpgsqlCommand cmd = new NpgsqlCommand(command, connection);              int res = cmd.ExecuteNonQuery();              PostRepo postRepo = new PostRepo(connection);              postRepo.DeleteByCommentId(id);              UserRepo userRepo = new UserRepo(connection);              userRepo.DeleteByCommentId(id);              if (res == -1) return false;              return true;          }  **Видалення користувача**  public bool DeleteById(int id)          {              string command = $"DELETE FROM users WHERE user\_id = {id};";              NpgsqlCommand cmd = new NpgsqlCommand(command, connection);              int res = cmd.ExecuteNonQuery();              PostRepo postRepo = new PostRepo(connection);              postRepo.DeleteByUserId(id);              CommentRepo commentRepo = new CommentRepo(connection);              commentRepo.DeleteByUserId(id);              if (res == -1) return false;              return true;          } |

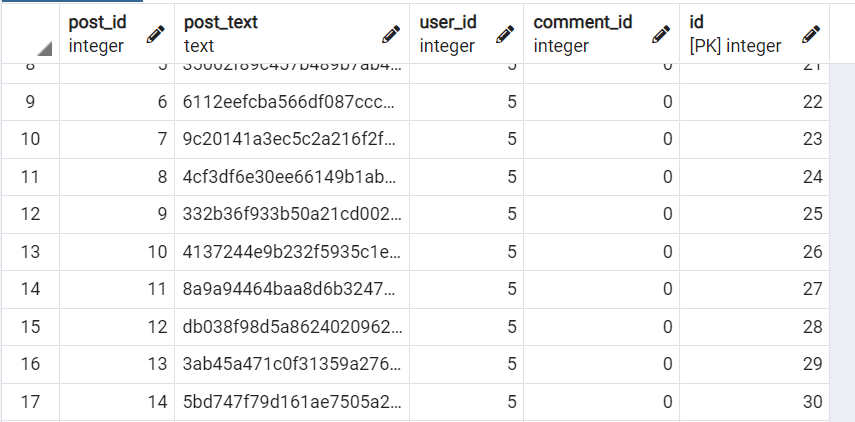


Обробка невірно введеного ідентифікаційного номера під час створення нового коментаря.

**Друге завдання**

|  |
| --- |
| **Генерація певної кількості випадкових коментарів**  public void GenerateXRandomComments(int number)          {              UserRepo userRepo = new UserRepo(connection);              int id = GetNewId();              for (int i = 0; i < number; i++)              {                  Comment comment = new Comment();                  PostRepo postRepo = new PostRepo(connection);                  comment.commentId = id + i;                  string command = $"SELECT md5(random()::text);";                  NpgsqlCommand cmd = new NpgsqlCommand(command, connection);                  comment.commentText = cmd.ExecuteScalar().ToString();                  connection.Close();                  connection.Open();                  comment.userId = userRepo.GetNewId();                  comment.postId = postRepo.GetNewId();                  this.Insert(comment);              }          } |
| **Генерація певної кількості випадкових постів**  public void GenerateXRandomPosts(int number)          {              UserRepo userRepo = new UserRepo(connection);              int id = GetNewId();              for (int i = 0; i < number; i++)              {                  Post post = new Post();                  post.postId = id + i;                  string command = $"SELECT md5(random()::text);";                  NpgsqlCommand cmd = new NpgsqlCommand(command, connection);                  post.postText = cmd.ExecuteScalar().ToString();                  connection.Close();                  connection.Open();                  post.userId = userRepo.GetNewId();                  this.Insert(post);              }          } |

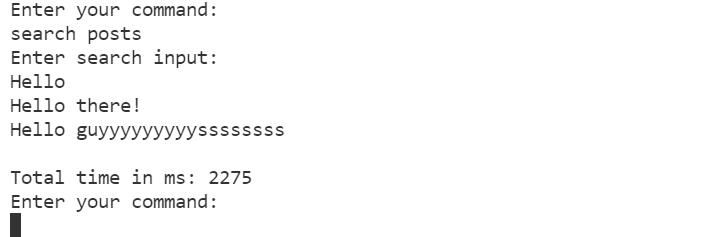


Випадково згенеровані користувачі 

Випадково згенеровані пости

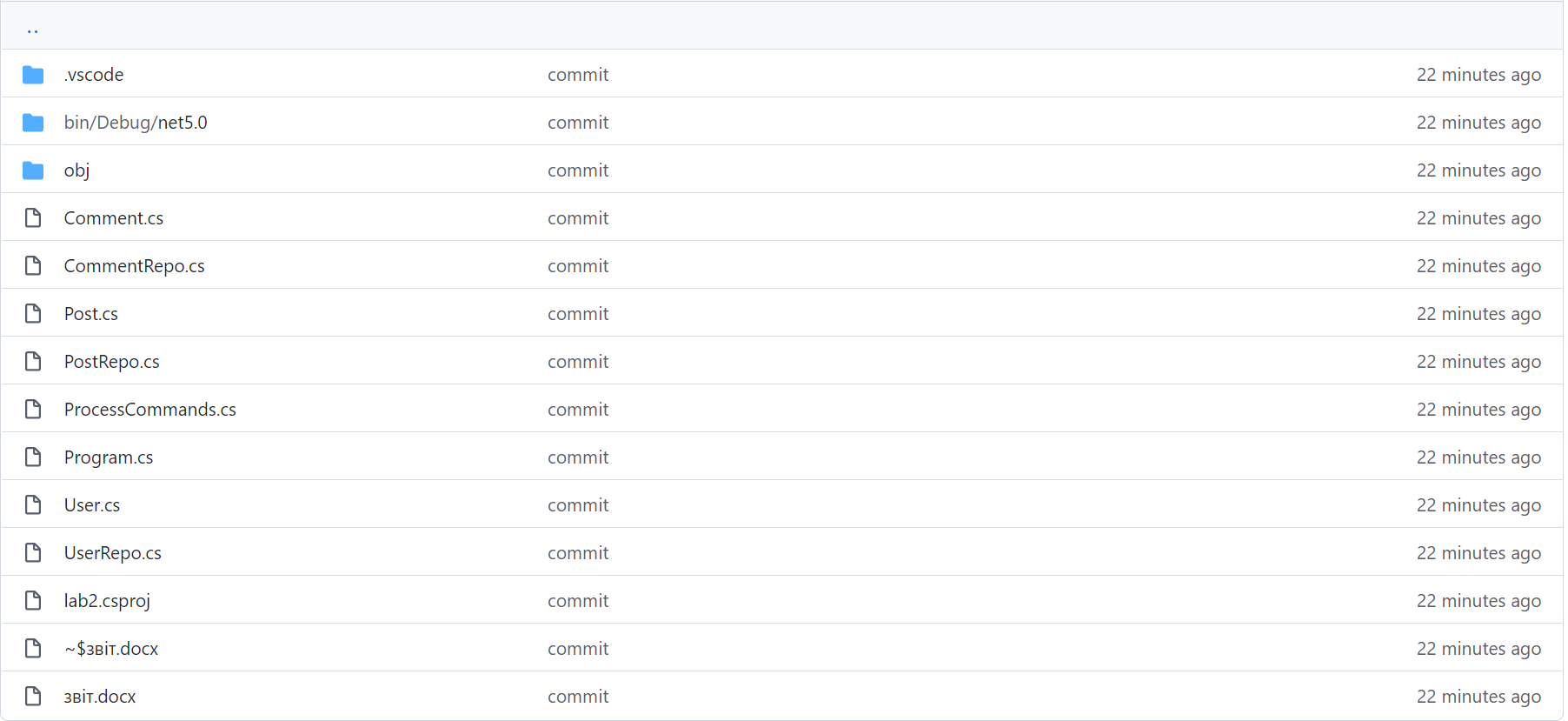
**Третє завдання**

|  |
| --- |
| **Пошук коментарів** |
| public List<string> SearchByText(string text)          {              List<string> comments = new List<string>();              string command = $"SELECT \* FROM comments WHERE comment\_text LIKE '%' || '{text}' || '%';";              NpgsqlCommand cmd = new NpgsqlCommand(command, connection);              NpgsqlDataReader reader = cmd.ExecuteReader();              while (reader.Read())              {                  string currText = "";                  currText = reader.GetString(1);                  if (!comments.Contains(currText))                  {                      comments.Add(currText);                  }              }              return comments;          }          public List<string> SearchByPostId(int id)          {              List<string> replies = new List<string>();              string command = $"SELECT \* FROM comments WHERE post\_id BETWEEN {id} AND {id};";              NpgsqlCommand cmd = new NpgsqlCommand(command, connection);              NpgsqlDataReader reader = cmd.ExecuteReader();              while (reader.Read())              {                  string currComment = "";                  currComment = reader.GetString(1);                  if (!replies.Contains(currComment))                  {                      replies.Add(currComment);                  }              }              connection.Close();              connection.Open();              return replies;          } |
| **Пошук постів** |
| public List<string> SearchByText(string text)          {              List<string> posts = new List<string>();              string command = $"SELECT \* FROM posts WHERE post\_text LIKE '%' || '{text}' || '%';";              NpgsqlCommand cmd = new NpgsqlCommand(command, connection);              NpgsqlDataReader reader = cmd.ExecuteReader();              while (reader.Read())              {                  string currText = "";                  currText = reader.GetString(1);                  if (!posts.Contains(currText))                  {                      posts.Add(currText);                  }              }              return posts;          } |



Консольний вивід для пошуку постів за словом “Hello”.

**Четверте завдання:**



Програма розділена на три групи:

1. **CommentRepo, PostRepo та UserRepo** напряму доступаються до бази даних через мову SQL
2. **ProcessCommands** – це посередник між основною частиною програми, яка складає консольний інтерфейс користувача та частиною програми, яка доступається о бази даних
3. **Program** – консольний інтерфейс користувача

**Висновки**

Під час роботи над цією лабораторною ми навчилися програмувати прикладні додатки баз даних PostgreSQL.