Mục lục:

[1. **Mô hình 3 lớp**: 3](#_Toc254734620)

[a. **Presentation Layer:** 3](#_Toc254734621)

[b. **Business Logic Layer:** 3](#_Toc254734622)

[c. **Data Access Layer:** 4](#_Toc254734623)

[2. **Auditing:** 4](#_Toc254734624)

[a. **Định nghĩa:** 4](#_Toc254734625)

[b. **Các kiểu giám sát (Types of Auditing):** 4](#_Toc254734626)

[c. **Audit Records và Audit Trails:** 5](#_Toc254734627)

[3. **Hiện thực**: 6](#_Toc254734628)

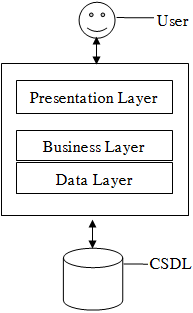
[a. **Môi trường Admin**: 6](#_Toc254734629)

[b. **Môi trường Client**: 12](#_Toc254734630)

[4. **Tổng kết**: 14](#_Toc254734631)

**Đề bài**: Xây dựng 1 ứng dụng windows theo mô hình 3 lớp (3-layer model), gồm:

1. ***Presentation layer***: xử lý phần hiển thị, giao diện. Áp dụng mô hình passive MVP.
2. ***Business layer***: chứa các thành phần xử lý các chức năng chính.
3. ***Data layer***: truy xuất dữ liệu từ CSDL. Dùng Oracle DBMS



Hiện thực các chức năng để kiểm soát các hoạt động của user cho ứng dụng đó, cụ thể:

1. Chức năng log in (áp dụng 1 phương pháp mã hóa cho việc lưu password).
2. Tạo/xóa/thay đổi user:

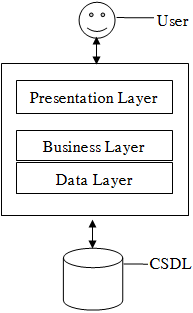
Cả hệ thống chỉ có 1 admin. Chỉ có admin mới có quyền tạo các user account. Các user thường chỉ được thay đổi password của mình.

Các thông tin mà 1 user account có thể được thiết lập (khi tạo mới hoặc khi thay đổi user) gồm :

* 1. Username
  2. Password

1. Chức năng enable/disable auditing của toàn hệ thống (chỉ admin được phép làm việc này).
2. Cho phép admin giám sát các hoạt động sau:
   1. Đối với mỗi user, admin có thể lựa chọn giám sát user đó hay không và nếu có thì sẽ giám sát những hoạt động sau:
      1. Khi user đăng nhập vào hệ thống.
      2. Khi user thay đổi thông tin account.
      3. Khi user thao tác để thực hiện các câu lệnh: SELECT, UPDATE, INSERT, DELETE ở bất kì table nào trong hệ thống hoặc trên những table cụ thể được admin chỉ ra trước.
   2. Admin có thể lựa chọn việc giám sát trên từng object (mà cụ thể ở đây là các table). Đối với mỗi object cần được giám sát, admin có thể chọn lựa:
      1. Chỉ giám sát khi có một hay nhiều lệnh hay tất cả các lệnh được thực hiện.
      2. Chỉ giám sát đối với một vài user nhất định nào đó hoặc đối với tất cả các user.
   3. Admin có thể tùy chọn chỉ audit khi các lệnh được thực thi thành công hoặc chỉ khi các lệnh không được thực thi thành công hoặc cả hai.
3. Cho phép mỗi user giám sát trên các object (cụ thể là table) thuộc schema của mình theo các tùy chọn như đã mô tả trong các phần đánh dấu **(\*)** ở trên (trong điều kiện chức năng Auditing được enable).
4. Cho phép mọi user xem các thông tin kết quả của việc giám sát mà user đó thiết lập.
5. **Mô hình 3 lớp**:

3 lớp này là **Presentation**, **Business Logic**, và **Data Access**. Các lớp này sẽ giao tiếp với nhau thông qua các dịch vụ(services) mà mỗi lớp cung cấp để tạo nên ứng dụng, lớp này cũng không cần biết bên trong lớp kia làm gì mà chỉ cần biết lớp kia cung cấp dịch vụ gì cho mình và sử dụng nó mà thôi.



* 1. **Presentation Layer:**

Lớp này làm nhiệm vụ giao tiếp với người dùng cuối để thu thập dữ liệu và hiển thị kết quả/dữ liệu thông qua các thành phần trong giao diện người sử dụng. Lớp này sẽ sử dụng các dịch vụ do lớp Business Logic cung cấp.

* + 1. *UI Components* là những phần tử chịu trách nhiệm thu thập và hiển thị thông tin cho người dùng cuối. Trong ASP.NET thì những thành phần này có thể là các TextBox, các Button…
    2. *UI Process Components* là thành phần chịu trách nhiệm quản lý các qui trình chuyển đổi giữa các UI Components. Ví dụ chịu trách nhiệm quản lý các màn hình nhập dữ liệu trong một loạt các thao tác định trước như các bước trong một Wizard…

**Lưu *ý****: lớp này không nên sử dụng trực tiếp các dịch vụ của lớp Data Access mà nên sử dụng thông qua các dịch vụ của lớp Business Logic vì khi bạn sử dụng trực tiếp như vậy, bạn có thể bỏ qua các ràng buộc, các logic nghiệp vụ mà ứng dụng cần phải có.*

* 1. **Business Logic Layer:**

Lớp này thực hiện các nghiệp vụ chính của hệ thống, sử dụng các dịch vụ do lớp Data Access cung cấp, và cung cấp các dịch vụ cho lớp Presentation. Lớp này cũng có thể sử dụng các dịch vụ của các nhà cung cấp thứ 3 (3rd parties) để thực hiện công việc của mình(ví dụ như sử dụng dịch vụ của các cổng thanh tóan trực tuyến như VeriSign, Paypal…).

* + 1. *Service Interface* là giao diện lập trình mà lớp này cung cấp cho lớp Presentationsử dụng. Lớp Presentationchỉ cần biết các dịch vụ thông qua giao diện này mà không cần phải quan tâm đến bên trong lớp này được hiện thực như thế nào.
    2. *Business Entities* là những thực thể mô tả những đối tượng thông tin mà hệ thống xử lý. Các business entities này cũng được dùng để trao đổi thông tin giữa lớp Presentationvà lớp Data Access.
    3. *Business Components* là những thành phần chính thực hiện các dịch vụ mà Service Interface cung cấp, chịu trách nhiệm kiểm tra các ràng buộc logic(constraints), các qui tắc nghiệp vụ (business rules), sử dụng các dịch vụ bên ngoài khác để thực hiện các yêu cầu của ứng dụng.
  1. **Data Access Layer:**

Lớp này thực hiện các nghiệp vụ liên quan đến lưu trữ và truy xuất dữ liệu của ứng dụng. Thường lớp này sẽ sử dụng các dịch vụ của các hệ quản trị cơ sở dữ liệu như SQL Server, Oracle,… để thực hiện nhiệm vụ của mình.

* + 1. *Data Access Logic components* (DALC) là thành phần chính chịu trách nhiệm lưu trữ vào và truy xuất dữ liệu từ các nguồn dữ liệu – Data Sources. Trong .NET Các DALCnày thường được hiện thực bằng cách sử dụng thư viện ADO.NET để giao tiếp với các hệ cơ sở dữ liệu.
    2. *Service Agents* là những thành phần trợ giúp việc truy xuất các dịch vụ bên ngoài một cách dễ dàng và đơn giản như truy xuất các dịch vụ nội tại.

1. **Auditing:**
   1. **Định nghĩa:**

Auditing là hoạt động giám sát và ghi lại… Được dựa trên các hoạt động cá nhân như thực hiện câu lệnh SQL, hay dựa trên sự kết hợp các yếu tố bao gồm tên, ứng dụng, thời gian,… Các chính sách bảo mật có thể dẫn đến việc audit khi những phần tử cụ thể trong CSDL Oracle bị truy cập hay thay thế.

Auditing nhìn chung được sử dụng:

* + 1. Cho phép giải trình những hành động hiện tại tham gia vào một schema, bảng, dòng riêng biệt, hay một nội dung cụ thể nào đó.
    2. Ngăn cản user khỏi hành động không thích hợp dựa trên trách nhiệm phải giải trình đó.
    3. Điều tra các hoạt động đáng ngờ. Ví dụ, nếu một user không được phép đang xóa dữ liệu từ một bảng nào đó thì người quản trị bảo mật sẽ ghi lại tất cả những kết nối CDSL và tất cả những hành động xóa các dòng từ bảng trong CSDL dù thành công hay không thành công.
    4. Thông báo cho người giám sát rằng có user bất hợp phát đang thao tác hay xóa dữ liệu hay user có nhiều quyền hệ thống hơn sự cho phép.
    5. Giám sát và thu thập dữ liệu về các hoạt động CSDL cụ thể. Ví dụ, người quản trị CSDL có thể thu thập thống kê về thông tin các bảng đang được update, hay bao nhiêu users cùng trung cập vào thời điểm cực đỉnh.
  1. **Các kiểu giám sát (Types of Auditing):**

Oracle cho phép giám sát theo 2 lựa chọn tập trung hoặc mở rộng. Bạn có thể giám sát:

* + 1. Sự thực thi câu lệnh thành công, hoặc không thành công, hoặc cả hai.
    2. Mỗi lần thực thi câu lệnh trong mỗi session của user, hay bất kì khi nào mà câu lệnh được thực thi.
    3. Hoạt động của tất cả các user hay của một user cụ thể nào đó

Có **bốn** kiểu giám sát:

* + 1. *Statement auditing:* chia thành hai nhóm:
       - Câu lệnh DDL: Ví dụ AUDIT TABLE giám sát tất cả các câu lệnh CREATE và DROP TABLE.
       - Câu lệnh DML: Ví dụ AUDIT SELECT TABLE giám sát tất cả câu lệnh SELECT trên bảng và trên view
    2. *Privilege auditing:* Kiểm tra việc sử dụng quyền hệ thống, ví dụ AUDIT CREATE TABLE. Privilege auditing được chú trọng hơn statement auditing vì nó chỉ kiểm tra việc sử dụng một số quyền nhất định. Bạn có thể đặt privilege auditing để kiểm tra những user được lựa chọn hay cho mọi user.
    3. *Schema object auditing:* Kiểm tra câu lệnh cụ thể trên đối tượng schema cụ thể, ví dụ AUDIT SELECT ON employees. (Rất được chú trọng). Schema object auditing luôn áp dụng cho tất cả các user.
    4. *Fine-grained auditing:* Kiểm tra dữ liệu truy xuất và các hoạt động dựa trên nội dung của dữ liệu đó. Ví dụ: Sử dụng DBMS\_FGA, người quản trị bảo mật tạo ra một chính sách kiểm tra trên một bảng. Nếu bất kì dòng nào trả về từ câu lệnh DML thỏa điều kiện kiểm tra thì một mục về sự kiện kiểm tra sẽ được chèn vào trong audit trail.
  1. **Audit Records và Audit Trails:**

Những thông tin được audit sẽ được lưu trong data dictionary table, gọi là database audit trail, hoặc lưu trong operating system files, gọi là operating system audit trail.

* + 1. *Bản ghi Audit (Audit trail records)*

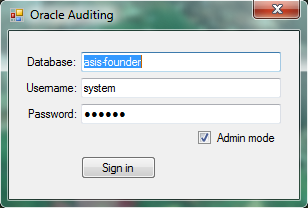
Chứa những loại thông tin khác nhau, phụ thuộc vào những sự kiện được giám sát và tập các lựa chọn giám sát. Thông tin sau đây được bao gồm trong mỗi bản ghi audit:

* + - * Database user name (DATABASE USER)
      * Operating system login user name (CLIENT USER)
      * Instance number (ko có trong Operation System…)
      * Process identifier
      * Session identifier
      * Terminal identifier
      * Name of the schema object accessed
      * Operation performed or attempted (ACTION)
      * Completion code of the operation
      * Date & time stamp in UTC format ( không có trong Operation System Audit Trail)
      * System privileges used (PRIVILEGE)
      * Chú ý: Audit trail không lưu thông tin về giá trị của dữ liệu dù nó liên quan đến trong câu lệnh được giám sát. Ví dụ, giá trị dữ liệu mới và giá trị dữ liệu cũ của hàng được update không được lưu lại khi câu lệnh UPDATE được giám sát. Tuy vậy, đối với phương pháp fine-grained auditing có khác.
    1. *Operating System Audit Trail*
    2. Oracle cho phép bản ghi dấu audit (audit trail records) được trực tiếp ghi vào operating system audit trail nếu hệ điều hành tạo một audit trail sẵn cho Oracle. Nếu không thì bản audit sẽ được ghi vào file bên ngoài CSDL, với định dạng tương tự như các file dấu tích Oracle (Oracle trace) khác.
    3. Oracle cho phép một hoạt động nào đó mà luôn bị giám sát được tiếp tục, thậm chí khi mà operating system audit trail ( file hệ điều hành chứa bản ghi audit) không được phép ghi lại bản ghi audit. Nguyên nhân thường gây ra vấn đề đó là do operating system audit trail hay file hệ thống bị đầy và không thể tiếp nhận thêm các bản ghi mới.
    4. Các nhà quản trị hệ thống mà cấu hình operating system auditing nên chắc chắn rằng audit trail hay file hệ thống ko bị đầy hoàn toàn. Phần lớn các hệ điều hành cung cấp cho nhà quản trị thông tin đầy đủ và cảnh báo để chắc chắn vấn đề đó không xảy ra. Tuy nhiên, chú ý rằng cấu hình auditing để sử dụng database audit trail loại bỏ được điểm yếu này, bởi vì hệ CSDL Oracle ngăn ngừa sự kiện được audit khỏi xảy ra nếu audit trail không thể tiếp nhận bản ghi database audit cho câu lệnh đó.

1. **Hiện thực**:

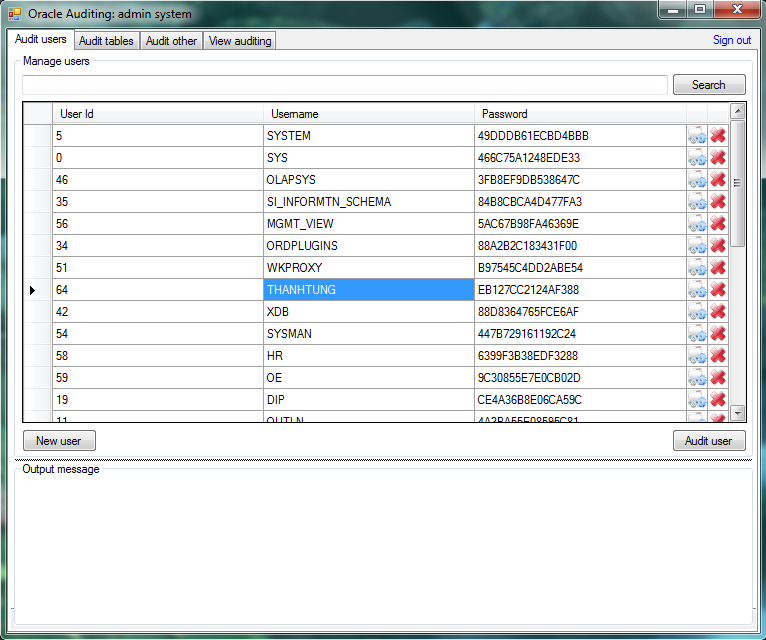
Chương trình được chia thành hai môi trường: admin & client.

* 1. **Môi trường Admin**:
     1. Cửa sổ Login:

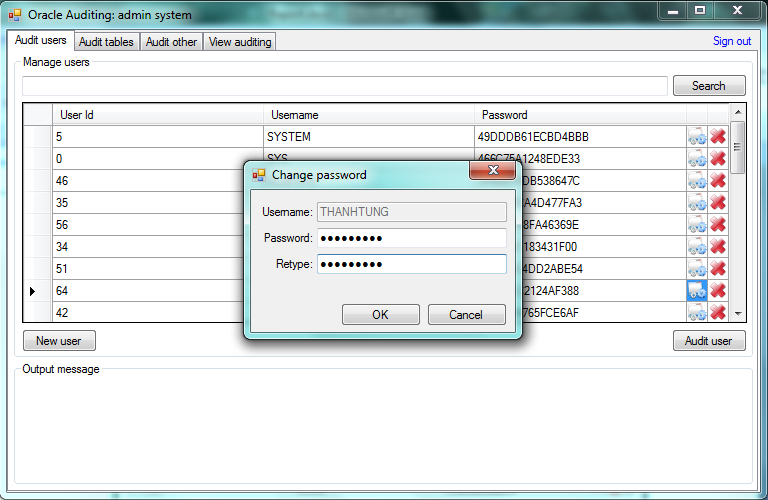


Password của người dùng được chương trình mã hóa bằng giải thuật mã hóa đối xứng 3DES.

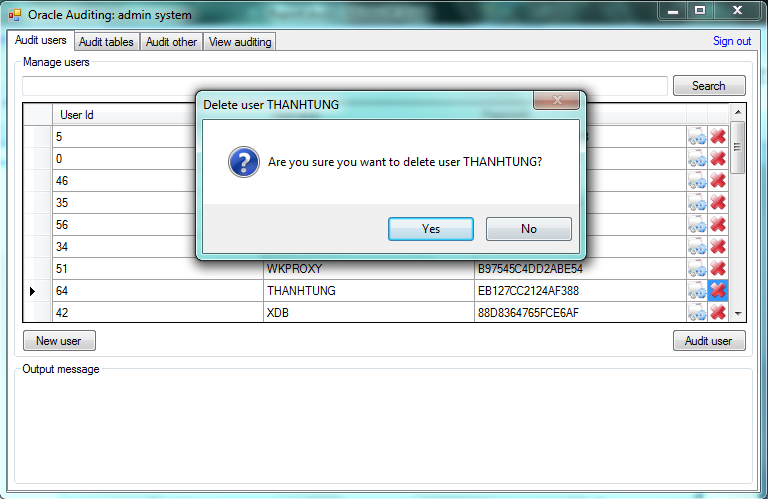
* + 1. Môi trường quản lý auditing của admin:



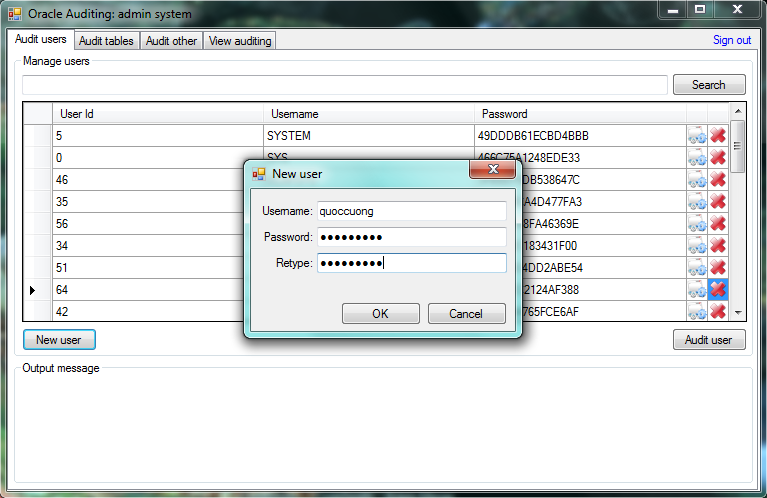
* + 1. Admin có thể thay đổi Password của user:



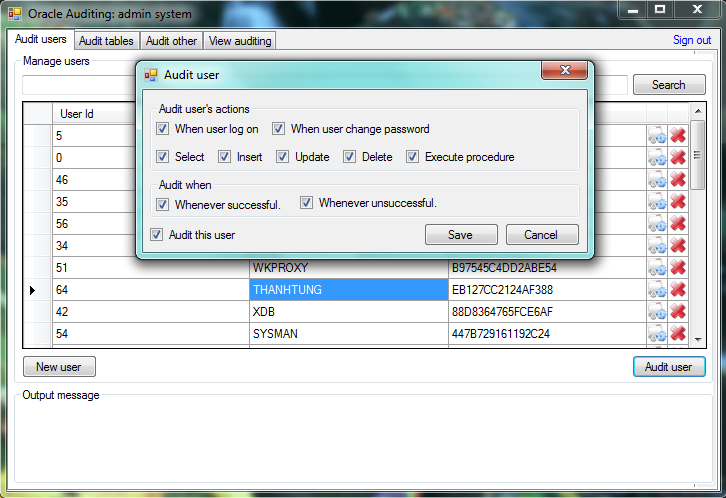
* + 1. Xóa user:



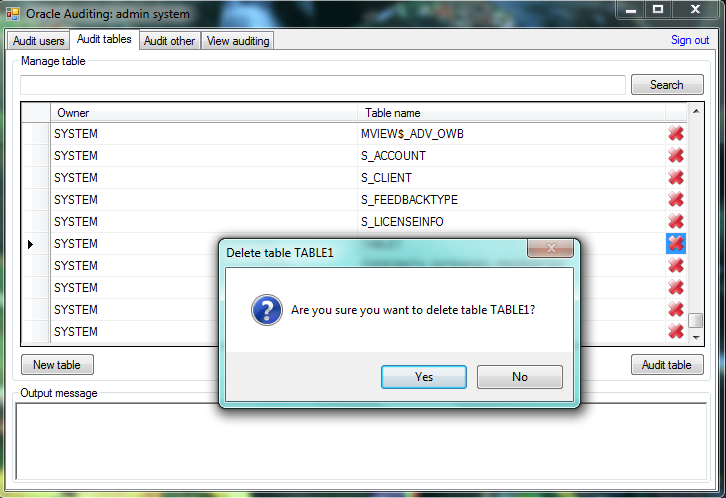
* + 1. Tạo mới user:



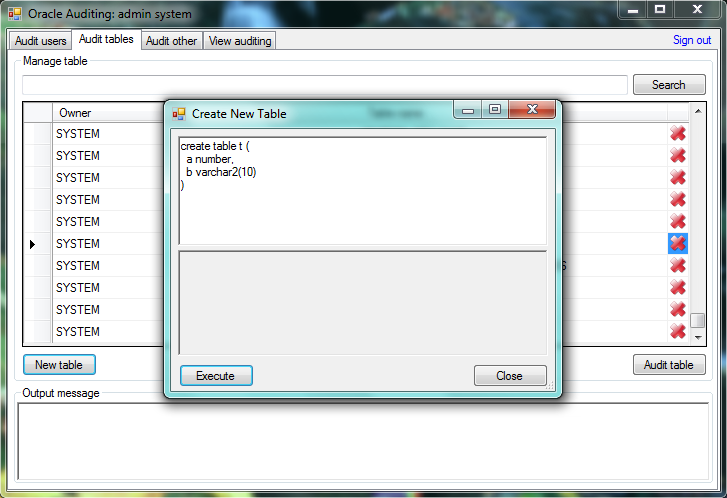
* + 1. Audit user:



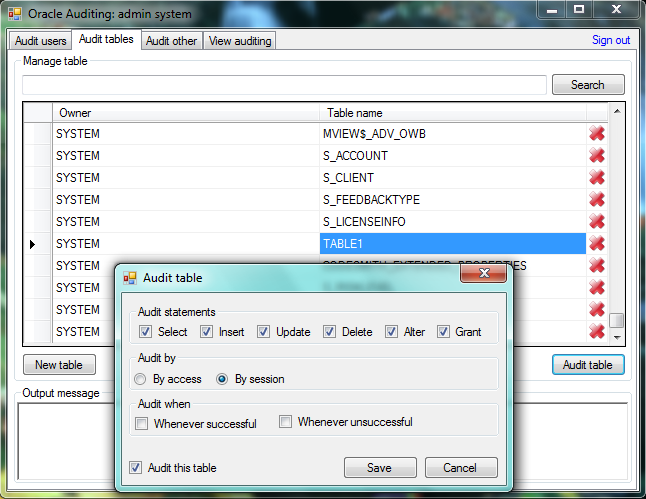
* + 1. Admin có thể xóa table:



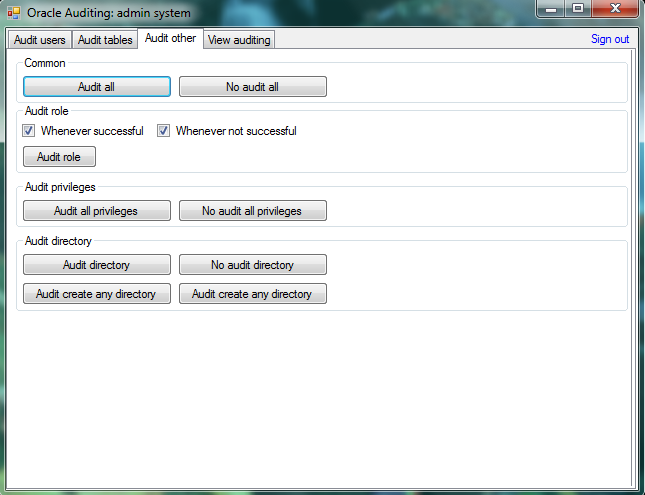
* + 1. Admin tạo mới table:



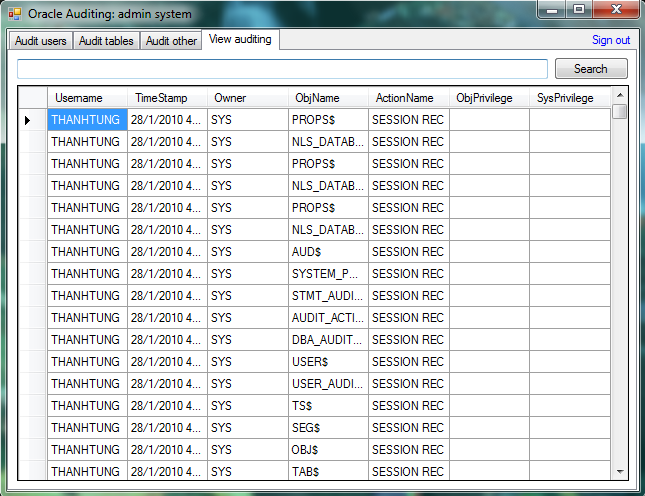
* + 1. Admin thực hiện auditing table:



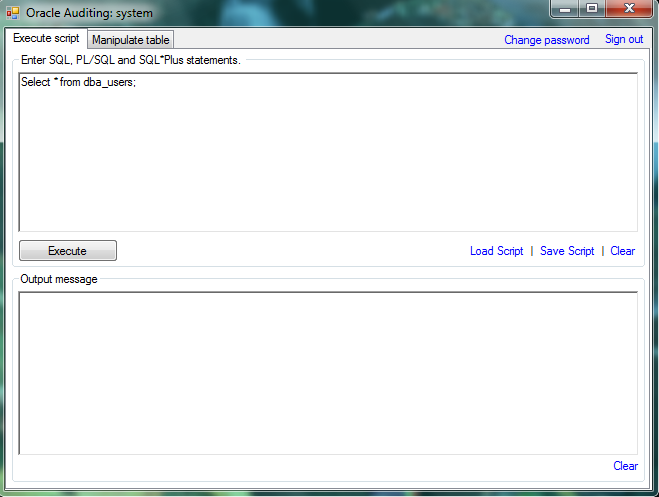
* + 1. Thực hiện một số chức năng auditing như thiết lập audit, audit privileges, audit directory:



* + 1. Hiển thị các thông tin audit:

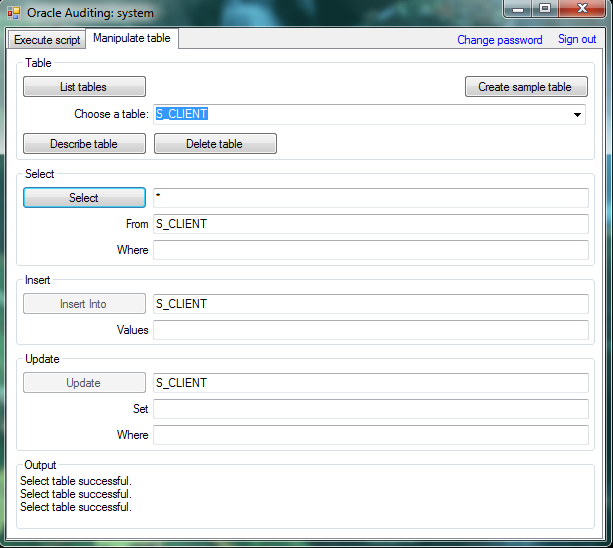


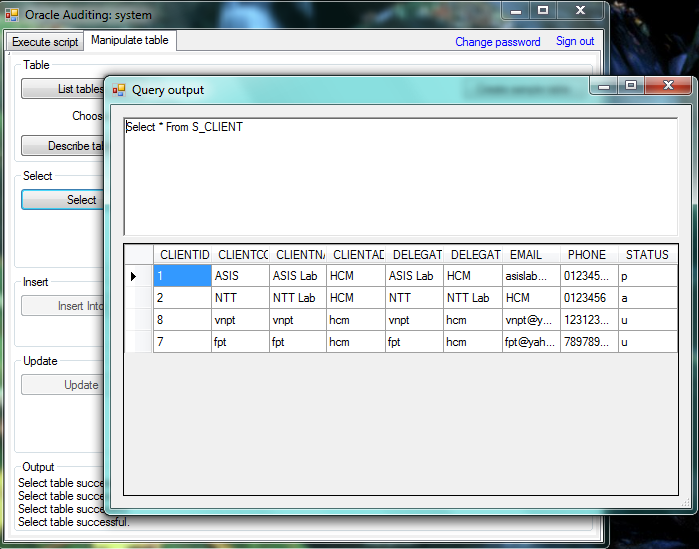
* 1. **Môi trường Client**:
     1. Client có thể thực thi các script:



Client có thể thay đổi password của mình, load 1 file script, save các script vào file.

* + 1. Cửa sổ hổ trợ một số thao tác lên table:





1. **Tổng kết**:

Qua bài tập lớn này, chúng em đã tìm hiểu thêm về chức năng ghi nhật ký của Oracle, cách viết ứng dụng theo mô hình ba lớp. Kết quả của quá trình tìm hiểu, chúng em đã hiện thực một chương trình nhỏ để admin có thể thiết lập việc ghi nhật ký cho cơ sở dữ liệu Oracle & xem các thông tin nhật ký mà Oracle đã ghi nhận.