МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПЕУБЛИКИ БЕЛАРУСЬ

Учреждение образования   
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных

технологий

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

«База данных «Салон автомобилей» с технологией шифрования и дешифрования данных DBMS\_CRYPTO»

Выполнил студент Мазенкова И.В.

(Ф.И.О.)

Руководитель проекта ассистент Нистюк О.А.

(учен. степень, звание, должность, Ф.И.О., подпись)

Заведующий кафедрой к.т.н., доц. Смелов В.В.

(учен. степень, звание, должность, Ф.И.О., подпись)

Курсовой проект защищен с оценкой

Минск 2022

Оглавление

[**Введение** 3](#_Toc120621030)

[**1.** **Аналитический обзор аналогов** 4](#_Toc120621031)

[**1.1** **Программное средство “Audy.by”** 4](#_Toc120621032)

[**1.2** **Программное средство “Hyundai.by”** 5](#_Toc120621033)

[**2.** **Анализ требований к программному средству и разработка функциональных требований** 6](#_Toc120621034)

[**3.3** **Разработка модели базы данных** 6](#_Toc120621037)

[**4.** **Разработка необходимых объектов** 8](#_Toc120621041)

[**4.1 Таблицы** 8](#_Toc120621042)

[**4.2 Пользователи** 11](#_Toc120621043)

[**4.3** **Хранимые процедуры** 11](#_Toc120621044)

[**4.4** **Описание процедур импорта и экспорта** 11](#_Toc120621045)

[**Cписок источников** 12](#_Toc120621046)

## **Введение**

Любая организация нуждается в своевременном доступе к информации. Ценность информации в современном мире очень высока. Роль распорядителей информации в современном мире чаще всего выполняют базы данных. Базы данных обеспечивают надежное хранение информации, в структурированном виде и своевременный доступ к ней. Практически любая современная организация нуждается в базе данных, удовлетворяющей те или иные потребности по хранению, управлению и администрированию данных.

За последние несколько лет наблюдается тенденция к усложнению структур данных. Простые виды информации, представимой в форме чисел и текстовых строк, не утратив своей значимости, дополняются сегодня многочисленными мультимедийными документами, графическими образами, хронологическими рядами, процедурными, или активными, данными и мириадами прочих сложных информационных форм.

На сегодняшний день на рынке представлено множество технологий доступа к данным и серверов баз данных, каждое, из которых имеет свои отличительные черты. Современные приложения обработки данных ориентированы на работу с большим количеством пользователей, на их удаленность от места расположения основного сервера БД.

Темой данного курсового проекта является разработка программы «Салон автомобилей».

Приложение «Салон автомобилей» имеет доступ к таблицам в базе данных и предназначено для взаимодействия с источником данных. Взаимодействие подразумевает получение данных, их представление в определенном формате для просмотра пользователем, редактирование в соответствии с реализованными в программе бизнес-алгоритмами и возврат обработанных данных обратно в базу данных.

База данных «Магазин автомобилей» существенно упрощает работу сотрудников магазина автомобилей, а также предоставляет возможность администратору своевременно вносить необходимые изменения.

## **Аналитический обзор аналогов**

В наши дни множество подобных программ можно найти на сайтах с тематикой по данному профилю. Перед тем как приступить к работе требовалось провести анализ и прочитать соответствующие статьи по данной теме. В ходе поиска данной литературы мною были найдены прототипы автомобильных салонов.

## **Программное средство “Audy.by”**

На рисунке 1.1 предствлено главное окно приложения «Audy.by».

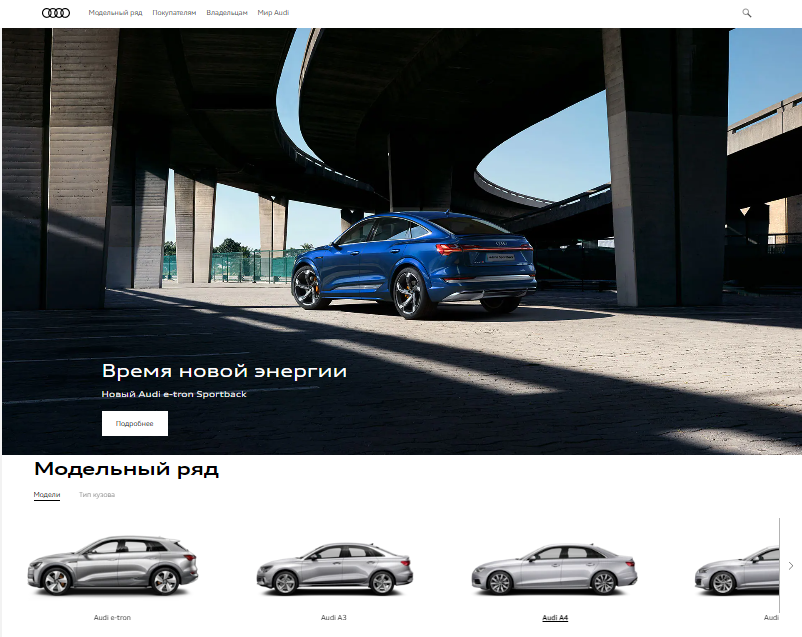


Рисунок 1.1 – Окно аналога приложения «Audy.by»

Ознакомившись с программным средством, был сделан вывод об основных функциональных особенностях:

* возможность выбора интересующей модели авто;
* возможность заказа товара;
* возможность изменения выбранного товара;
* возможность удалить выбранный товар и начать заново;

## **Программное средство “Hyundai.by”**

На рисунке 1.2 предствлено главное окно приложения «Hyundai.by».

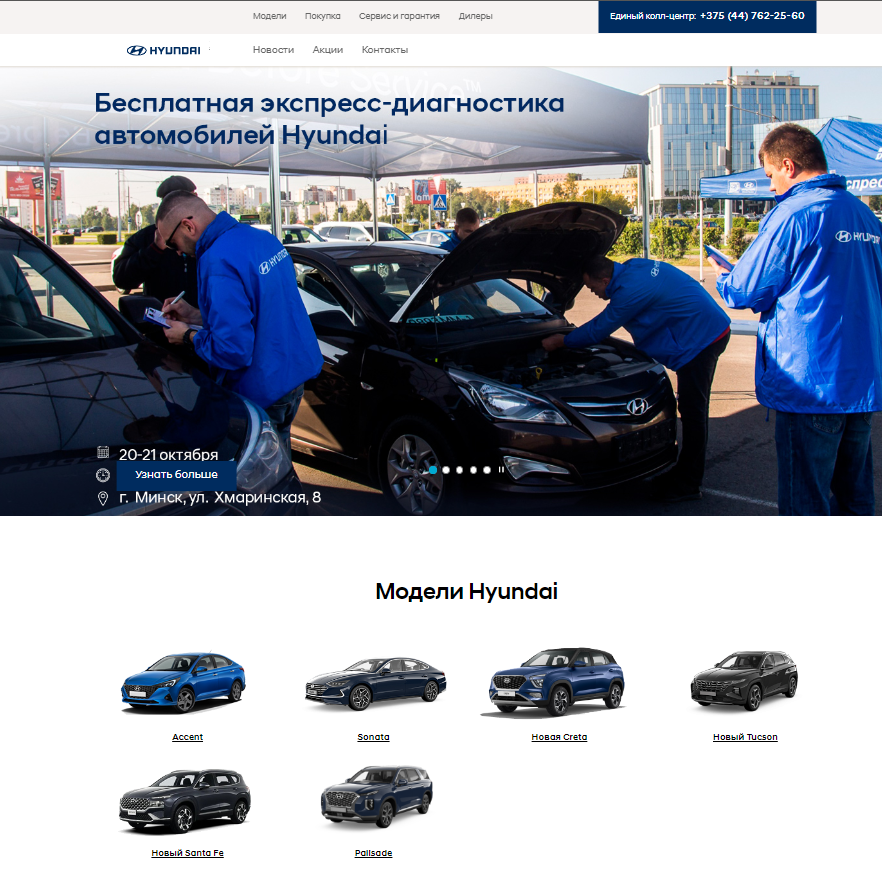


Рисунок 1.2 – Окно аналога приложения «Hyundai.by»

Ознакомившись с программным средством, был сделан вывод об основных функциональных особенностях:

* возможность выбора интересующей модели авто;
* возможность заказа товара;
* возможность изменения выбранного товара;

## **Анализ требований к программному средству и разработка функциональных требований**

Задачей проекта является проектирование базы данных.

Функционально должны быть выполнены следующие задачи:

* База данных содержит 2 роли: администратор и клиент;
* Администратор добавляет и удаляет информацию об автомобилях
* Клиент бронирует автомобили и отменяет бронь на них;
* Клиент просматривает список автомобилей.

Должны быть выполнены следующие требования:

* доступ к данным должен осуществляться только через соответствующие процедуры;
* должен быть проведен импорт данных из XML файлов, экспорт данных в формат XML;
* необходимо протестировать производительность базы данных на таблице, содержащей не менее 100 000 строк, и внести изменения в структуру в случае необходимости. Необходимо проанализировать планы запросов к таблице;
* применить технологию базы данных согласно выбранной теме: подробно описать применяемые системные пакеты, утилиты или технологии; показать применение указанной технологии в базе данных.

База данных должна быть спроектирована в Oracle Database 12c.



## **Разработка модели базы данных**

Первым этапом курсового проекта будет создание логически взаимосвязанных таблиц. Чтобы составить визуальную взаимосвязанную структуру базы данных, нам необходимо продумать, какая информация будет храниться в этих таблицах, после этого создать связи с помощью первичных и внешних ключей.

Диаграмма базы данных, спроектированной в ходе разработки показана на рисунке 2.1.

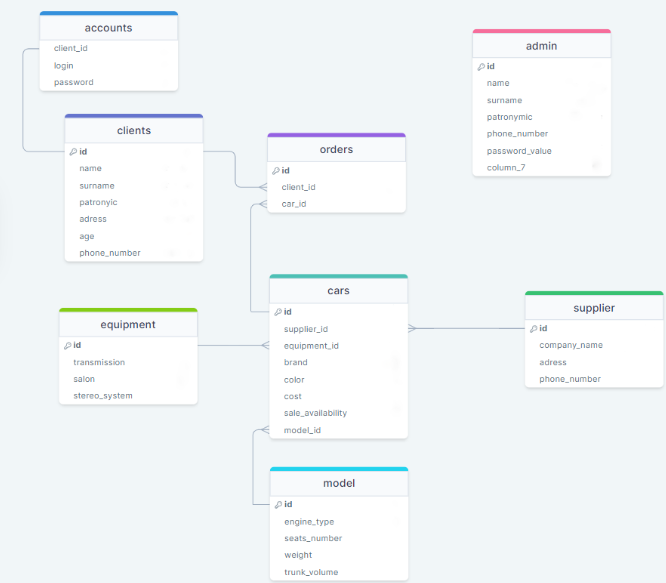


Рисунок 2.1 – Диаграмма базы данных

База данных приложения состоит из 9 таблиц, взаимосвязанных между собой внешними ключами:

* Cars;
* Equipment;
* Supplier;
* Model;
* Orders;
* Clients;
* Accounts;



# **Разработка необходимых объектов**

# **Таблицы**

Для реализации базы данных «Магазин автомобилей» было разработано 9 таблиц: Cars, Equipment, Supplier, Model, Orders, Clients, Workers, Accounts, Position.

Таблица Clients представляет список клиентов, состоит из столбцов (таблица 3.1):

Таблица 3.1 – Столбцы таблицы Clients

|  |  |  |
| --- | --- | --- |
| Наименование | Описание | Тип |
| id | идентификатор клиента, первичный ключ | int |
| surname | фамилия клиента | nvarchar2 |
| name | имя клиента | nvarchar2 |
| patronymic | отчество клиента | nvarchar2 |
| address | адрес | nvarchar2 |
| age | возраст | int |
| phone number | номер телефона | nvarchar2 |

Таблица Workers представляет список сотрудников состоит из столбцов (таблица 3.2):

Таблица 3.2 – Столбцы таблицы Admin

|  |  |  |
| --- | --- | --- |
| Наименование | Описание | Тип |
| id | идентификатор сотрудника, первичный ключ | int |
| surname | фамилия сотрудника, тип | nvarchar2 |
| name | имя сотрудника, тип | nvarchar2 |
| patronymic | отчество сотрудника, тип | nvarchar2 |
| phone number | номер телефона, тип | nvarchar2 |
| password value | пароль сотрудника | nvarchar2 |

Таблица Accounts представляет список учётных записей клиентов в системе, таблица состоит из столбцов (таблица 3.3):

Таблица 3.3 – Столбцы таблицы Accounts

|  |  |  |
| --- | --- | --- |
| Наименование | Описание | Тип |
| login | логин пользователя | nvarchar2 |
| password | пароль пользователя | nvarchar2 |
| client id | идентификатор клиента, внешний ключ | int |

Таблица Supplier представляет список поставщиков автомобилей, состоит из столбцов (таблица 3.4):

Таблица 3.4 – Столбцы таблицы Supplier

|  |  |  |
| --- | --- | --- |
| Наименование | Описание | Тип |
| id | идентификатор поставщика, первичный ключ | int |
| company name | название компании | nvarchar2 |
| address | адрес поставщика | number |
| phone number | номер телефона поставщика | nvarchar2 |

Таблица Model представляет список моделей автомобилей, состоит из столбцов (таблица 3.5):

Таблица 3.5 – Столбцы таблицы Model

|  |  |  |
| --- | --- | --- |
| Наименование | Описание | Тип |
| id | идентификатор модели, первичный ключ | int |
| engine type | тип двигателя | nvarchar2 |
| seats number | количество мест | int |
| trunk volume | вместимость багажника | int |
| weight | вес автомобиля | number |

Таблица Orders представляет заказы клиента, состоит из столбцов (таблица 3.6):

Таблица 3.6 – Столбцы таблицы Orders

|  |  |  |
| --- | --- | --- |
| Наименование | Описание | Тип |
| id | номер заказа, тип, первичный ключ | int |

Продолжение таблицы 3.6

|  |  |  |
| --- | --- | --- |
| Наименование | Описание | Тип |
| client id | идентификатор клиента, тип, внешний ключ для связи с таблицей clients | int |
| car id | идентификатор автомобиля, внешний ключ для связи с таблицей cars | int |
| client id | идентификатор клиента, тип, внешний ключ для связи с таблицей clients | int |
| ordering date | дата заказа | date |

Таблица Cars представляет список автомобилей, которые будет искать и заказывать пользователь через разработанное приложение. Данная таблица состоит из столбцов (таблица 3.7):

Таблица 3.7 – Столбцы таблицы Cars

|  |  |  |
| --- | --- | --- |
| Наименование | Описание | Тип |
| id | идентификатор автомобиля, тип, первичный ключ | int |
| supplier id | идентификатор поставщика, тип, внешний ключ для связи с таблицей Supplier | int |
| model id | идентификатор модели, тип, внешний ключ для связи с таблицей Models | int |
| equipment id | идентификатор оснащения, тип int, внешний ключ для связи с таблицей Equipment | int |
| brand | марка автомобиля | nvarchar2 |
| color | цвет | nvarchar2 |
| cost | цена автомобиля | int |
| sale availability | наличие для продажи | int |

Таблица Equipment представляет оборудование автомобиля, которое идёт в комплекте с его покупкой при заказе, поэтому связана с таблицей Car состоит из столбцов (таблица 3.8):

Таблица 3.8 – Столбцы таблицы Equipment

|  |  |  |
| --- | --- | --- |
| Наименование | Описание | Тип |
| id | идентификатор оснащения | int |

Продолжение таблицы 3.8

|  |  |  |
| --- | --- | --- |
| Наименование | Описание | Тип |
| transmission | коробка передач | nvarchar2 |
| salon | тип автомобиля | nvarchar2 |
| stereo system | стереосистема | nvarchar2 |

# **Пользователи**

Пользователь базы данных – это физическое или юридическое лицо, которое имеет доступ к БД и пользуется услугами информационной системы для получения информации.

При проектировании базы данных было использовано 2 пользователя. Первый пользователь – администратор, который является системным администратором, имеет доступ для чтения и изменения таблиц, связанных с клиентами.

Второй пользователь – клиент – имеет доступ к таблице с автомобилями и может оформить заказ автомобиля.

# **Хранимые процедуры**

Хранимая процедура – объект базы данных, представляющий собой набор SQL-инструкций, который компилируется один раз и хранится на сервере. При создании хранимой процедуры можно определить необязательный список параметров. Таким образом, процедура будет принимать соответствующие аргументы при каждом ее вызове. Все хранимые процедуры, созданные в данном курсовом проекте, содержат обработку исключений, для того, чтобы ограничить пользователя от возможных ошибок.

При разработке курсового проекта было создано 37 процедур для следующих целей:

* выборка данных для таблиц;
* добавление/удаление сотрудника;
* создание учетной записи, для увеличения безопасности учетной записи пользователя его пароль находится в зашифрованном виде.
* авторизация пользователей;
* регистрация пользователей;
* добавление клиента;
* добавления заказа;
* обновление информации о клиенте;
* добавления автомобиля;
* экспорт и импорт таблицы в формат xml.

# **Описание процедур импорта и экспорта**

В курсовом проекте был проведен импорт данных из XML файлов и экспорт данных в формат XML.

XML — это формат данных, используемый для обмена данными в форме, которая может быть легко использована и распространена. Часто возникает необходимость импортировать и экспортировать XML-файлы в Oracle

Для реализации обоих процедур потребовалось создать директорий с желаемым расположением при подключении под пользователем SYS к инстансу и выдать пользователю, владеющему базой данных, грант на чтение и запись из этого директория. Это изображено на рисунке 4.1.



Рисунок 4.1 – Создание директория DIR

В рамках задания нужно было разработать процедуру, экспортирующую данные из таблицы в XML документ. С этой целью разработана процедура exportXmlToEQUIPMENT. Она изображена на рисунке 4.2.

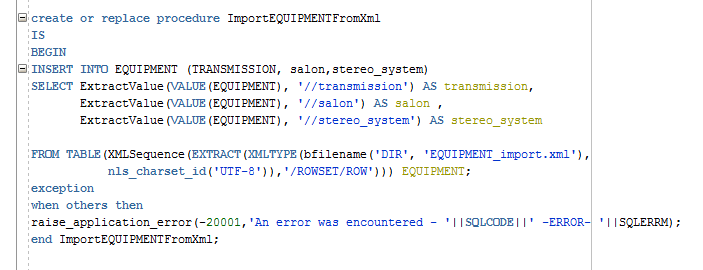


Рисунок 4.2 – Процедура exportXmlToEQUIPMENT

Также нужно было написать процедуру, которая бы, наоборот, импортировала существующие данные из XML документа в таблицу базы данных. В этом случае данные вставляются в таблицу EQUIPMENT . Процедура ImportEQUIPMENTFromXml изображена на рисунке 4.3.

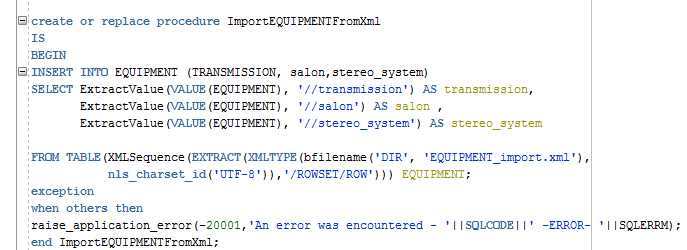


Рисунок 4.3. – Процедура ImportEQUIPMENTFromXml

# **Тестирование производительности**

Для проверки производительности базы данных необходимо заполнить ее большим количеством различных данных и узнать время выполнения одного запроса.

Для данной задачи мы создали анонимный блок, и так как работаем именно с действующей базой данных, то можем использовать готовую процедуру для вставки данных в таблицу. Разработанный анонимный блок позволяет добавить большое количество строк за одно выполнение (рисунок 5.1).

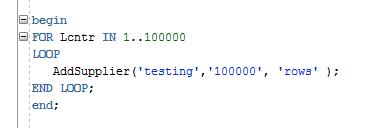


Рисунок 5.1 – Анонимный блок для заполнения таблицы большим количеством данных

Таким образом было добавлено 100000 строк в таблицу.

После этого в другом анонимном блоке, который содержит в себе дополнительно переменную для отслеживания времени выполнения, был сделан запрос, который должен возвращать в результате 20000 строк. Данный блок показан на рисунке 5.2.

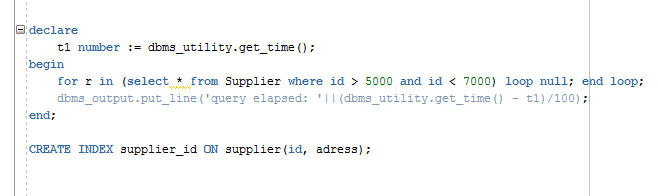


Рисунок 5.2 – Блок отбора данных

Далее после выполнения данного блока наша таблица будет содержать большое количество данных, и мы можем проанализировать время выполнения запроса. Также можно просмотреть план запроса используя стандартные средства Oracle, а именно кнопку на главной панели, предварительно выделив данный запрос. Покажем результат, в котором будет заметна разница во времени выполнениея запроса до создания индекса на рисунке 5.3.



Рисунок 5.3 – Результат отбора данных

Теперь остаётся лишь создать индекс и протестировать время выполнения аналогичного запроса. Покажем это на рисунке 5.4.



Рисунок 5.4 – Результат после создания индекса

Как видим, время запроса уменьшилось, но тем не менее даже без индекса наша база данных успешно прошла тест на производительность.

# **Описание технологии и ее применения в базе данных**

# **Технология «Шифрования данных»**

Шифрование – неотъемлемая часть базы данных. В наше время существует множество конфиденциальных данных, и множество баз данных хранящие их. Во избежание кражи или несанкционированного доступа к ним данные следует скрывать, а еще лучше шифровать. При шифровании данных даже при их краже от них не будет большой пользы, так как при современных алгоритмах шифрования расшифровать данные очень сложно даже практически невозможно.

В данном курсовом проекте я использовал встроенные функции Oracle, а именно: пакет DBMS\_CRYPTO, который помог мне зашифровать необходимые данные в моей базе данных.

# **Применение технологии**

Мною были написаны функции, которые шифруют и дешифруют данные по алгоритму AES-128. Алгоритм является блочным и симметричным. AES-128 преобразует один 128-битный блок в другой, используя секретный ключ, который нужен для такого преобразования. Для расшифровки полученного 128-битного блока используют второе преобразование с тем же секретным ключом.

Выбранный мной алгоритм обладает большой криптостойкостью и обеспечит моим данным надежную сохранность. Функцию шифрования можно увидеть на рисунке 5.1.

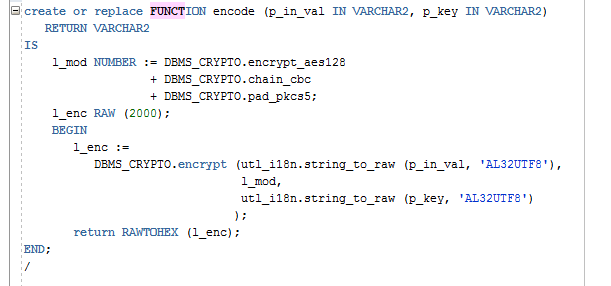


Рисунок 5.1 – Скрипт функции щифрования

Функция принимает 2 параметра: сообщение и ключ для шифрования. Ключ является основным параметром при шифровании, ведь этот же ключ необходим и для расшифрования данных. Шифрование данных имеет смысл только в том случае, если зашифрованные данные в какой-то момент будут прочитаны и использованы в приложении. Шифрование происходит при помощи функции encrypt и дешифрование функцией decrypt.

Функцию дешифрования можно увидеть на рисунке 5.2.

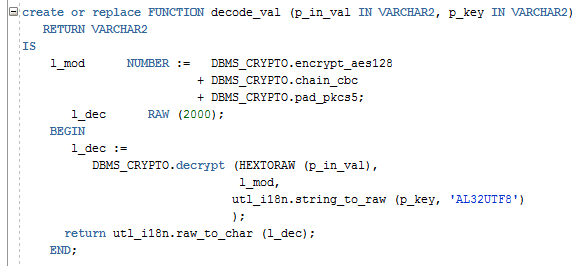


Рисунок 5.2 – скрипт функции дешифрования

Данный алгоритм обладает большой криптостойкостью и обеспечивает моим данным надежную сохранность. Результат работы функции шифрования представлен на рисунке 5.3.

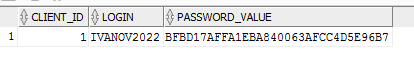


Рисунок 5.3 – результат работы функции шифрования

# **Заключение**

В процессе решения поставленной задачи была достигнута поставленная цель по созданию базы данных для программного средства «Магазин автомобилей», которая в совокупности с приложением формирует полноценное desktop-приложение для управления магазином. В данной работе использовалось СУБД Oracle DataBase 12c.При разработке курсового проекта использовались объекты: таблицы, хранимые процедуры, индексы, функции, последовательности, триггеры.

Основной целью курсового проекта стало проектирование базы данных

БД прошла тестирование при использовании большого количества данных. Также были реализованы процедуры для импорта, экспорта данных в формат XML.

Была реализована технология «Резервное копирование и восстановление». Шифрование данных было реализовано с помощью пакета DBMS\_CRYPTO.

В соответствии с полученным результатом работы программы можно сделать вывод, что разработанная бд работает корректно, а требования технического задания выполнены в полном объёме.

# **Cписок источников**

1. Документация Oracle [Электронный ресурс] / Foundation, Inc. https://docs.oracle.com/cd/B28359\_01/server.111/b31222/toc.htm – Дата доступа: 10.11.2021.
2. Официальный сайт Oracle [Электронный ресурс] / Foundation, Inc. https://www.oracle.com/database/database-vault/index.html– Дата доступа: 10.11.2021.
3. Продукты Oracle [Электронный ресурс] [Электронный ресурс] / Foundation, Inc. http://www.interface.ru/home.asp?artId=24678 – Дата доступа: 21.11.2021.
4. Документы Oracle [Электронный ресурс] / Foundation, Inc. http://www.oracle.com/technetwork/database/security/database-vault-ds-12c-1898877.pdf – Дата доступа: 11.11.2021.