Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ

УНИВЕРСИТЕТ»

«Проектирование и разработка баз данных интернет-приложений»

Отчёт по лабораторной работе №2

Реализация базы данных

Выполнил: Гвоздовский К.В.

ФИТ 3 курс 6 группа

Преподаватель: Нистюк О.А.

**Microsoft SQL Server**

**Таблицы**

Таблица Apartments

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничения | Назначение |
| apartment\_id | int | primary key, not null | Идентификатор квартиры |
| day\_cost | money | not null | Цена аренды за день |
| room\_number | int | not null | Номер квартиры |
| house\_number | nvarchar(10) | Максимум 10 символов, not null | Номер дома |
| street | nvarchar(50) | Максимум 50 символов, not null | Название улицы |
| city | nvarchar(30) | Максимум 30 символов, not null | Название города |
| number\_of\_rooms | int | not null | Количество комнат в квартире |
| description | nvarchar(max) |  | Описание квартиры |

Таблица Rents

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничения | Назначение |
| rent\_id | int | primary key | Идентификатор договора об аренде |
| apartment\_id | int | foreign key | Индентификатор квартиры |
| status | nvarchar(10) | Максимум 10 символов, not null | Статус аренды |
| data\_begin | date | not null | Дата начала аренды |
| end\_date | date |  | Дата окончания аренды |
| client\_id\_ | Int | not null, foreign key | Идентификатор клиента |

Таблица Clients

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничения | Назначение |
| client\_id | int | primary key, not null | Идентификатор клиента |
| surname | nvarchar(20) | Максимум 20 символов, not null | Фамилия клиента |
| name | nvarchar(20) | Максимум 20 символов, not null | Имя клиента |
| passport\_number | nvarchar(15) | Максимум 15 символов, not null | Идентификационный номер пасспорта |
| phone\_number | int | not null | Контактный номер клиента |

Таблица Bill\_details

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничения | Назначение |
| bill\_id | int | primary\_key, not null | Идентификатор счёта |
| rent\_id | int | foreign key, not null | Идентификатор договора об аренде |
| bill\_date | date | not null | Дата оплаты |
| total | money | not null | Сумма счёта |

**Функции**

Функция CalculateTotalCost: вычисление общей стоимости аренды на основе цены за день аренды и общего количества дней

|  |
| --- |
| CREATE FUNCTION CalculateTotalCost (@day\_cost money, @num\_days int)  RETURNS money  AS  BEGIN  DECLARE @total\_cost money  SET @total\_cost = @day\_cost \* @num\_days  RETURN @total\_cost  END |

Функция GetClientRentHistory: возвращает историю аренды для определенного клиента

|  |
| --- |
| CREATE FUNCTION GetClientRentHistory (@client\_id int)  RETURNS TABLE  AS  RETURN (  SELECT R.rent\_id, R.data\_begin, R.end\_date, R.status,  A.apartment\_id, A.street, A.city,  A.number\_of\_rooms, A.day\_cost  FROM dbo.Rents R  JOIN dbo.Apartments A ON R.apartment\_id = A.apartment\_id  WHERE R.client\_id = @client\_id  ) |

Функция GetTotalNumberOfApartmentsInCity: возвращает количество квартир в определённом городе

|  |
| --- |
| CREATE FUNCTION GetTotalNumberOfApartmentsInCity (@city nvarchar(30))  RETURNS int  AS  BEGIN  DECLARE @total\_apartments int  SELECT @total\_apartments = COUNT(apartment\_id)  FROM dbo.Apartments  WHERE city = @city  RETURN @total\_apartments  END |

**Процедуры**

InsertNewClient: добавляет нового клиента:

|  |
| --- |
| CREATE PROCEDURE InsertNewClient  @surname nvarchar(20),  @name nvarchar(20),  @passport\_number nvarchar(15),  @phone\_number int  AS  BEGIN  INSERT INTO dbo.Clients (surname, name, passport\_number, phone\_number)  VALUES (@surname, @name, @passport\_number, @phone\_number)  END |

CloseRent: закрывает аренду

|  |
| --- |
| CREATE PROCEDURE CloseRent  @rent\_id int  AS  BEGIN  UPDATE dbo.Rents  SET status = 'Closed'  WHERE rent\_id = @rent\_id  END |

UpdateApartmentDescription: обновляет описание квартиры

|  |
| --- |
| CREATE PROCEDURE UpdateApartmentDescription  @apartment\_id int,  @description nvarchar(max)  AS  BEGIN  UPDATE dbo.Apartments  SET description = @description  WHERE apartment\_id = @apartment\_id  END |

**Представления**

ActiveRentsView: показывает активные аренды с информацией о квартире и клиенте

|  |
| --- |
| CREATE VIEW ActiveRentsView AS  SELECT R.rent\_id, R.data\_begin, R.end\_date, R.status,  A.apartment\_id, A.street, A.city,  C.client\_id, C.surname, C.name  FROM dbo.Rents R  JOIN dbo.Apartments A ON R.apartment\_id = A.apartment\_id  JOIN dbo.Clients C ON R.client\_id = C.client\_id  WHERE R.status = 'Active' |

AvailableApartmentsView: показывает доступные для аренды квартиры

|  |
| --- |
| CREATE VIEW AvailableApartmentsView AS  SELECT A.apartment\_id, A.room\_number, A.house\_number, A.street, A.city, A.number\_of\_rooms, A.description  FROM dbo.Apartments A  LEFT JOIN dbo.Rents R ON A.apartment\_id = R.apartment\_id  WHERE R.rent\_id IS NULL |

OverdueRentsView: отображает просроченные аренды и информацию о них

|  |
| --- |
| CREATE VIEW dbo.OverdueRentsView AS  SELECT R.rent\_id, R.data\_begin, R.end\_date, R.status,  C.client\_id, C.surname, C.name, C.passport\_number, C.phone\_number  FROM dbo.Rents R  JOIN dbo.Clients C ON R.client\_id = C.client\_id  WHERE R.end\_date < GETDATE() AND R.status = 'Active' |

**Индексы**

idx\_apartments\_city\_street: ускоряет поиск квартир по конкретному городу и улице

|  |
| --- |
| CREATE INDEX idx\_apartments\_city\_street ON Apartments(city, street); |

idx\_rents\_apartment\_id: обеспечивает быстрый доступ к данным об арендах на основе идентификатора квартиры

|  |
| --- |
| CREATE INDEX idx\_rents\_apartment\_id ON Rents(apartment\_id); |

idx\_rents\_client\_id: позволяет быстро находить все аренды, связанные с определенным клиентом по его идентификатору

|  |
| --- |
| CREATE INDEX idx\_rents\_client\_id ON Rents(client\_id); |

idx\_bill\_details\_rent\_id: обеспечивает эффективный доступ к данным о счетах на основе идентификатора аренды

|  |
| --- |
| CREATE INDEX idx\_bill\_details\_rent\_id ON Bill\_details(rent\_id); |

**Oracle**

**Таблицы**

Таблица Apartments

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничения | Назначение |
| apartment\_id | int | primary key, not null | Идентификатор квартиры |
| day\_cost | float(2) | not null, 2 знака после запятой | Цена аренды за день |
| room\_number | int | not null | Номер квартиры |
| house\_number | nvarchar2(10) | Максимум 10 символов, not null | Номер дома |
| street | nvarchar2(50) | Максимум 50 символов, not null | Название улицы |
| city | nvarchar(30) | Максимум 30 символов, not null | Название города |
| number\_of\_rooms | int | not null | Количество комнат в квартире |
| description | nvarchar(max) |  | Описание квартиры |

Таблица Rents

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничения | Назначение |
| rent\_id | int | primary key, not null | Идентификатор договора об аренде |
| apartment\_id | int | foreign key, not null | Индентификатор квартиры |
| status | nvarchar2(10) | Максимум 10 символов, not null | Статус аренды |
| data\_begin | date | not null | Дата начала аренды |
| end\_date | date | not null | Дата окончания аренды |
| client\_id\_ | int | not null, foreign key | Идентификатор клиента |

Таблица Clients

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничения | Назначение |
| client\_id | int | primary key, not null | Идентификатор клиента |
| surname | nvarchar2(20) | Максимум 20 символов, not null | Фамилия клиента |
| name | nvarchar2(20) | Максимум 20 символов, not null | Имя клиента |
| passport\_number | nvarchar2(15) | Максимум 15 символов, not null | Идентификационный номер пасспорта |
| phone\_number | int | not null | Контактный номер клиента |

Таблица Bill\_details

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип данных | Ограничения | Назначение |
| bill\_id | int | primary\_key, not null | Идентификатор счёта |
| rent\_id | int | foreign key, not null | Идентификатор договора об аренде |
| bill\_date | date | not null | Дата оплаты |
| total | float(2) | not null, 2 цифры после запятой | Сумма счёта |

**Функции**

Функция CalculateTotalCost: вычисление общей стоимости аренды на основе цены за день аренды и общего количества дней

|  |
| --- |
| CREATE OR REPLACE FUNCTION *CalculateTotalCost* (day\_cost FLOAT(2), num\_days NUMBER) RETURN FLOAT(2) IS  total\_cost NUMBER; BEGIN  total\_cost := day\_cost \* num\_days;  RETURN total\_cost; END *CalculateTotalCost*; |

Функция GetClientRentHistory: возвращает историю аренды для определенного клиента

|  |
| --- |
| CREATE OR REPLACE FUNCTION GetClientRentHistory (client\_id NUMBER) RETURN SYS\_REFCURSOR IS  rent\_history SYS\_REFCURSOR; BEGIN  OPEN rent\_history FOR  SELECT R.rent\_id, R.data\_begin, R.end\_date, R.status,  A.apartment\_id, A.street, A.city,  A.number\_of\_rooms, A.day\_cost  FROM Rents R  JOIN Apartments A ON R.apartment\_id = A.apartment\_id  WHERE R.client\_id = client\_id;  RETURN rent\_history; END GetClientRentHistory; |

Функция GetTotalNumberOfApartmentsInCity: возвращает количество квартир в определённом городе

|  |
| --- |
| CREATE OR REPLACE FUNCTION *GetTotalNumberOfApartmentsInCity* (city VARCHAR2) RETURN NUMBER IS  total\_apartments NUMBER; BEGIN  SELECT *COUNT*(apartment\_id) INTO total\_apartments  FROM Apartments  WHERE city = city;   RETURN total\_apartments; END *GetTotalNumberOfApartmentsInCity*; |

**Процедуры**

InsertNewClient: добавляет нового клиента:

|  |
| --- |
| CREATE OR REPLACE PROCEDURE *InsertNewClient*(  surname NVARCHAR2,  name NVARCHAR2,  passport\_number NVARCHAR2,  phone\_number NUMBER ) AS BEGIN  INSERT INTO Clients (surname, name, passport\_number, phone\_number)  VALUES (surname, name, passport\_number, phone\_number);  COMMIT; END *InsertNewClient*; |

CloseRent: закрывает аренду

|  |
| --- |
| CREATE OR REPLACE PROCEDURE *CloseRent*(  rent\_id NUMBER ) AS BEGIN  UPDATE Rents  SET status = 'Closed'  WHERE rent\_id = rent\_id;  COMMIT; END *CloseRent*; |

UpdateApartmentDescription: обновляет описание квартиры

|  |
| --- |
| CREATE OR REPLACE PROCEDURE *UpdateApartmentDescription*(  apartment\_id NUMBER,  description CLOB ) AS BEGIN  UPDATE Apartments  SET description = description  WHERE apartment\_id = apartment\_id;  COMMIT; END *UpdateApartmentDescription*; |

**Представления**

ActiveRentsView: показывает активные аренды с информацией о квартире и клиенте

|  |
| --- |
| CREATE OR REPLACE VIEW ActiveRentsView AS SELECT R.rent\_id, R.data\_begin, R.end\_date, R.status,  A.apartment\_id, A.street, A.city,  C.client\_id, C.surname, C.name FROM Rents R JOIN Apartments A ON R.apartment\_id = A.apartment\_id JOIN Clients C ON R.client\_id = C.client\_id WHERE R.status = 'Active'; |

AvailableApartmentsView: показывает доступные для аренды квартиры

|  |
| --- |
| CREATE OR REPLACE VIEW AvailableApartmentsView AS SELECT A.apartment\_id, A.room\_number, A.house\_number, A.street, A.city, A.number\_of\_rooms, A.description FROM Apartments A LEFT JOIN Rents R ON A.apartment\_id = R.apartment\_id WHERE R.rent\_id IS NULL; |

OverdueRentsView: отображает просроченные аренды и информацию о них

|  |
| --- |
| CREATE OR REPLACE VIEW OverdueRentsView AS SELECT R.rent\_id, R.data\_begin, R.end\_date, R.status,  C.client\_id, C.surname, C.name, C.passport\_number, C.phone\_number FROM Rents R JOIN Clients C ON R.client\_id = C.client\_id WHERE R.end\_date < *SYSDATE* AND R.status = 'Active'; |

**Индексы**

idx\_apartments\_city\_street: ускоряет поиск квартир по конкретному городу и улице

|  |
| --- |
| CREATE INDEX idx\_apartments\_city\_street ON Apartments(city, street); |

idx\_rents\_apartment\_id: обеспечивает быстрый доступ к данным об арендах на основе идентификатора квартиры

|  |
| --- |
| CREATE INDEX idx\_rents\_apartment\_id ON Rents(apartment\_id); |

idx\_rents\_client\_id: позволяет быстро находить все аренды, связанные с определенным клиентом по его идентификатору

|  |
| --- |
| CREATE INDEX idx\_rents\_client\_id ON Rents(client\_id); |

idx\_bill\_details\_rent\_id: обеспечивает эффективный доступ к данным о счетах на основе идентификатора аренды

|  |
| --- |
| CREATE INDEX idx\_bill\_details\_rent\_id ON Bill\_details(rent\_id); |