

В запроснике «ЗаявкаНаОбслуживание»

SELECT Заявка."СтатусЗаявки",

Клиенты."ФИО",

Заявка."КодКлиента"

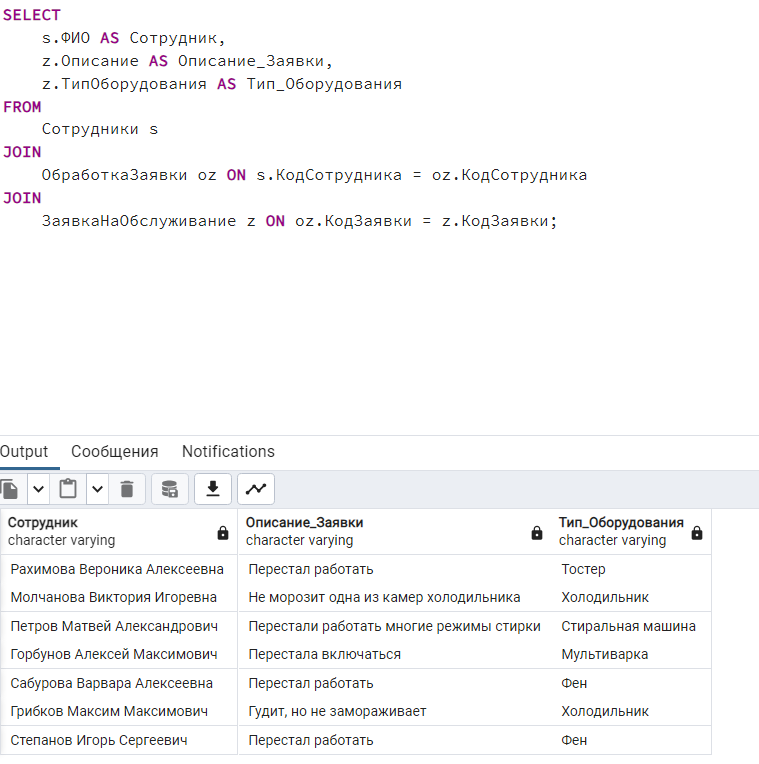
FROM "ЗаявкаНаОбслуживание" AS Заявка

JOIN "Клиенты" AS Клиенты

ON Заявка."КодКлиента" = Клиенты."КодКлиента"

WHERE Заявка."СтатусЗаявки" = 'Готова к выдаче';

Запрос выбирает данные данные из двух таблиц: "ЗаявкаНаОбслуживание" и "Клиенты". Он извлекает три столбца из этих таблиц: "СтатусЗаявки" из таблицы "Заявка", "ФИО" из таблицы "Клиенты" и "КодКлиента" из таблицы "Заявка".  
  
Далее, запрос объединяет эти две таблицы с использованием оператора JOIN по столбцу "КодКлиента", чтобы сопоставить данные клиентов с соответствующими данными заявок на обслуживание.  
  
Затем, в разделе WHERE, указывается условие, что будет выбраны только те строки, у которых значение столбца "СтатусЗаявки" равно 'Готова к выдаче'.  
  
Таким образом, этот запрос выбирает статус заявки, ФИО клиента и его код из таблиц "ЗаявкаНаОбслуживание" и "Клиенты" для всех заявок, которые находятся в статусе 'Готова к выдаче'.



В Запроснике заявканаобс

*SELECT*

*s.ФИО AS Сотрудник,*

*z.Описание AS Описание\_Заявки,*

*z.ТипОборудования AS Тип\_Оборудования*

*FROM*

*Сотрудники s*

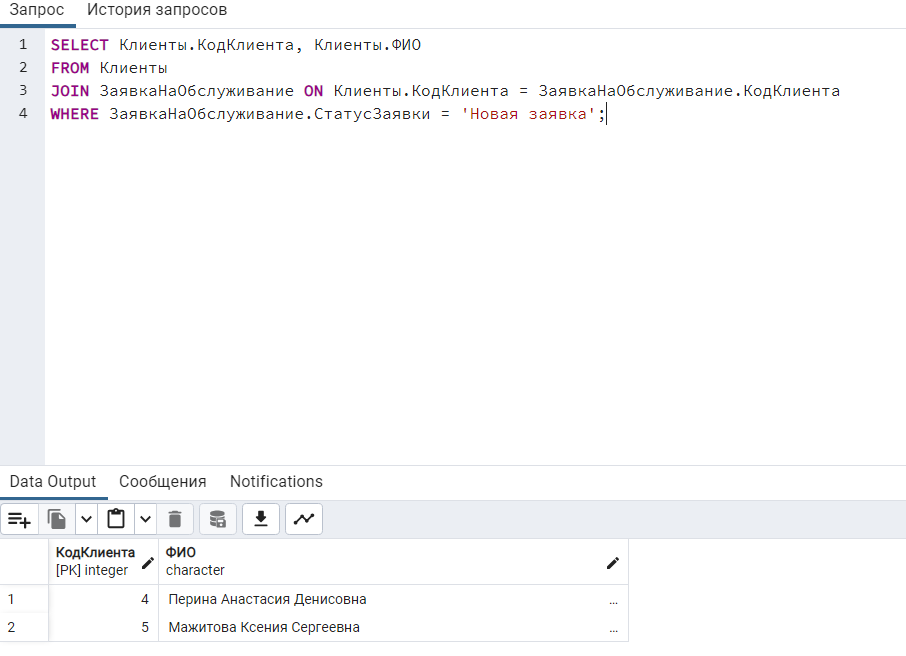
*JOIN*

*ОбработкаЗаявки oz ON s.КодСотрудника = oz.КодСотрудника*

*JOIN*

*ЗаявкаНаОбслуживание z ON oz.КодЗаявки = z.КодЗаявки;*

Этот запрос выбирает данные из таблиц Сотрудники (s), ОбработкаЗаявки (oz) и ЗаявкаНаОбслуживание (z), используя операторы JOIN для соединения этих таблиц. SELECT выбирает три столбца: ФИО сотрудника из таблицы Сотрудники, Описание заявки и тип оборудования из таблицы ЗаявкаНаОбслуживание. По атрибутам, представленным в запросе и использованных операторах JOIN, можно предположить, что запрос выводит информацию о связи между сотрудниками, заявками на обслуживание и типом оборудования.



Делать в запроснике «ЗаявкаНаОбслуживание».

*SELECT Клиенты.КодКлиента, Клиенты.ФИО*

*FROM Клиенты*

*JOIN ЗаявкаНаОбслуживание ON Клиенты.КодКлиента = ЗаявкаНаОбслуживание.КодКлиента*

*WHERE ЗаявкаНаОбслуживание.СтатусЗаявки = 'Новая заявка';*

Этот запрос извлекает КодКлиента и ФИО клиентов, у которых есть новая заявка на обслуживание. Он соединяет таблицы Клиенты и ЗаявкаНаОбслуживание по КодКлиента и выбирает только записи, где статус заявки равен "Новая заявка".

Старт (овал)

Клиент оставляет заявку (прямоугольник)

Регистрация заявки оператором (прямоугольник)

Обработка заявки (прямоугольник)

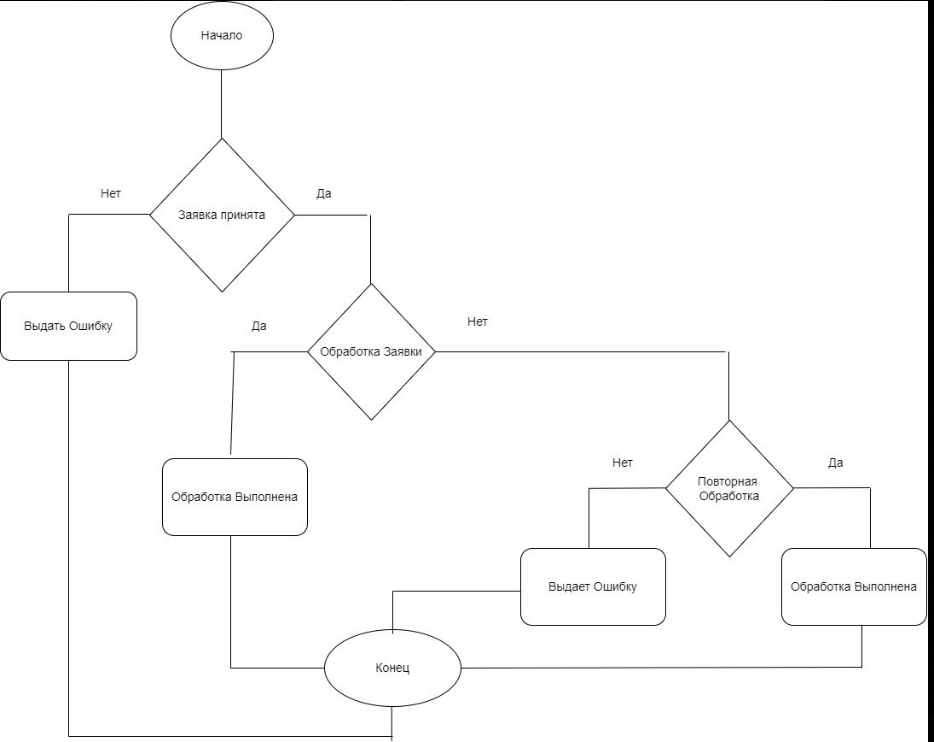
Назначение исполнителя (прямоугольник)

Исполнение заявки (прямоугольник)

Отчётность и информирование (прямоугольник)

Мониторинг и анализ (прямоугольник)

Финиш (овал)



З

* **СоздатьЗаявку**: Создает новую заявку, заполняет её данными и сохраняет в базе данных.
* **ОбработатьЗаявку**: Обрабатывает заявку, присваивает приоритет и назначает исполнителя.
* **ИсполнитьЗаявку**: Выполняет ремонт, изменяет статус заявки на "Выполнена" и сохраняет изменения.
* **ОпределитьПриоритет**: Определяет приоритет заявки в зависимости от типа оборудования.
* **НазначитьИсполнителя**: Назначает исполнителя из справочника сотрудников.
* **ВыполнитьРемонт**: Логика выполнения ремонта для заявки.

О

он выполняет, включают регистрацию движения заявки, проверку уникальности заявки и изменение её статуса. Рассмотрим код более подробно.

**Процедура ОбработкаПроведения**

**Основные шаги:**

1. **Инициализация движения по регистру накопления "ОбработкаЗаявок"**:
   * Создаёт новую запись движения в регистре накопления для регистрации поступления (прихода) новой заявки.
   * Устанавливает параметры движения: тип движения (Приход), период, номер заявки и количество.

Движения.ОбработкаЗаявок.Записывать = Истина;

Движение = Движения.ОбработкаЗаявок.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Приход;

Движение.Период = Дата;

Движение.Заявка = Заявка;

Движение.Количество = 1;

1. **Запрос к регистру накопления для проверки уникальности заявки**:
   * Формирует и выполняет запрос к регистру накопления "ОбработкаЗаявок", чтобы проверить, была ли уже обработана заявка с таким номером.

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ОбработкаЗаявок.Заявка КАК Заявка

|ИЗ

| РегистрНакопления.ОбработкаЗаявок КАК ОбработкаЗаявок

|ГДЕ

| ОбработкаЗаявок.ВидДвижения = &ВидДвижения";

Запрос.УстановитьПараметр("ВидДвижения",ВидДвиженияНакопления.Приход);

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

1. **Проверка уникальности и обработка заявки**:
   * Если заявка с таким номером уже существует, устанавливается флаг отказа и выводится сообщение.
   * Если заявки нет, то вызывается процедура обработки заявки ОбработатьЗаявку.

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Если ВыборкаДетальныеЗаписи.Заявка = Движение.Заявка Тогда

Отказ = Истина;

Сообщить ("Заявка под номер " + Движение.Заявка.Номер + " уже обработана")

КонецЕсли

КонецЦикла;

Если Не Отказ Тогда

ОбработатьЗаявку(Движение.Заявка)

КонецЕсли;

**ОбработатьЗаявку**

**Основные шаги:**

1. **Формирование и выполнение запроса для поиска документа "ЗаявкаНаОбслуживание" по номеру**:
   * Выполняет запрос к документам "ЗаявкаНаОбслуживание" для получения ссылки на документ с указанным номером.

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ЗаявкаНаОбслуживание.Ссылка КАК Ссылка

|ИЗ

| Документ.ЗаявкаНаОбслуживание КАК ЗаявкаНаОбслуживание

|ГДЕ

| ЗаявкаНаОбслуживание.Номер = &Номер";

Запрос.УстановитьПараметр("Номер", Заявка.Номер);

РезультатЗапроса = Запрос.Выполнить();

1. **Изменение статуса заявки**:
   * Получает объект документа "ЗаявкаНаОбслуживание" и изменяет его статус на "В работе".

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

ИскомыйДокумент = ВыборкаДетальныеЗаписи.Ссылка.ПолучитьОбъект();

ИскомыйДокумент.СтатусЗаявки = Перечисления.СтатусыЗаявки.Вработе;

ИскомыйДокумент.Записать()

КонецЦикла;

**Возможные ошибки и проверка условий**

* **Ошибка: Дублирование заявок**: Если заявка уже была обработана, выводится сообщение и устанавливается флаг отказа (Отказ = Истина).

Если ВыборкаДетальныеЗаписи.Заявка = Движение.Заявка Тогда

Отказ = Истина;

Сообщить ("Заявка под номер " + Движение.Заявка.Номер + " уже обработана")

КонецЕсли

* **Проверка условий: Уникальность заявки**: Проверка наличия заявки с таким же номером в регистре накопления.

Если Не Отказ Тогда

ОбработатьЗаявку(Движение.Заявка)

КонецЕсли;

Этот код, в целом, занимается управлением заявками на обслуживание, проверяет их уникальность, обновляет статусы и регистрирует движения в системе.

Р

**ОбработкаПроведения**

1. **Инициализация движения по регистру накопления "ВыполнениеЗаявок"**:

Создаётся новая запись движения в регистре накопления для регистрации поступления (прихода) новой заявки на выполнение.

Устанавливаются параметры движения: тип движения (Приход), период, номер заявки и количество.

Движения.ВыполнениеЗаявок.Записывать = Истина;

Движение = Движения.ВыполнениеЗаявок.Добавить();

Движение.ВидДвижения = ВидДвиженияНакопления.Приход;

Движение.Период = Дата;

Движение.Заявка = Заявка;

Движение.Количество = 1;

**Запрос к регистру накопления для проверки уникальности выполнения заявки**:

* + Формируется и выполняется запрос к регистру накопления "ВыполнениеЗаявок", чтобы проверить, была ли уже выполнена заявка с таким номером.

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ВыполнениеЗаявок.Заявка КАК Заявка

|ИЗ

| РегистрНакопления.ВыполнениеЗаявок КАК ВыполнениеЗаявок

|ГДЕ

| ВыполнениеЗаявок.ВидДвижения = &ВидДвижения";

Запрос.УстановитьПараметр("ВидДвижения", ВидДвиженияНакопления.Приход);

РезультатЗапроса = Запрос.Выполнить();

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

1. **Проверка уникальности выполнения заявки**:
   * Если заявка с таким номером уже выполнена, устанавливается флаг отказа и выводится сообщение.

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Если ВыборкаДетальныеЗаписи.Заявка = Движение.Заявка Тогда

Отказ = Истина;

Сообщить ("Заявка под номер " + Движение.Заявка.Номер + " уже выполнена");

КонецЕсли

КонецЦикла;

**Проверка, обработана ли заявка**:

* + Если заявка не была обработана, устанавливается флаг отказа и выводится сообщение.

Если Не ПроверитьОбработку(Заявка) Тогда

Отказ = Истина;

Сообщить("Заявка под номером " + Движение.Заявка.Номер + " не обработана");

КонецЕсли;

1. **Выполнение заявки**:
   * Если заявка уникальна и обработана, вызывается процедура ВыполнитьЗаявку.

Если Не Отказ Тогда

ВыполнитьЗаявку(Движение.Заявка);

КонецЕсли;

**ВыполнитьЗаявку**

1. **Формирование и выполнение запроса для поиска документа "ЗаявкаНаОбслуживание" по номеру**:
   * Выполняется запрос к документам "ЗаявкаНаОбслуживание" для получения ссылки на документ с указанным номером.

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ЗаявкаНаОбслуживание.Ссылка КАК Ссылка

|ИЗ

| Документ.ЗаявкаНаОбслуживание КАК ЗаявкаНаОбслуживание

|ГДЕ

| ЗаявкаНаОбслуживание.Номер = &Номер";

Запрос.УстановитьПараметр("Номер", Заявка.Номер);

РезультатЗапроса = Запрос.Выполнить();

1. **Изменение статуса заявки на "Выполнено"**:
   * Получается объект документа "ЗаявкаНаОбслуживание" и изменяется его статус на "Выполнено".

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

ИскомыйДокумент = ВыборкаДетальныеЗаписи.Ссылка.ПолучитьОбъект();

ИскомыйДокумент.СтатусЗаявки = Перечисления.СтатусыЗаявки.Выполнено;

ИскомыйДокумент.Записать()

КонецЦикла;

**Функция ПроверитьОбработку**

**Основные шаги:**

1. **Формирование и выполнение запроса для проверки, обработана ли заявка**:
   * Выполняется запрос к документам "ОбработкаЗаявки" для проверки, была ли заявка с указанным номером обработана.

Запрос = Новый Запрос;

Запрос.Текст =

"ВЫБРАТЬ

| ОбработкаЗаявки.Заявка КАК Заявка

|ИЗ

| Документ.ОбработкаЗаявки КАК ОбработкаЗаявки";

РезультатЗапроса = Запрос.Выполнить();

1. **Проверка наличия заявки в обработанных**:
   * Проверяется, была ли заявка найдена среди обработанных заявок.

ВыборкаДетальныеЗаписи = РезультатЗапроса.Выбрать();

ЗаявкаВОбработке = Ложь;

Пока ВыборкаДетальныеЗаписи.Следующий() Цикл

Если ВыборкаДетальныеЗаписи.Заявка = Заявка Тогда

ЗаявкаВОбработке = Истина

КонецЕсли

КонецЦикла;

Возврат ЗаявкаВОбработке;

**Возможные ошибки и проверка условий**

* **Ошибка: Дублирование выполненных заявок**: Если заявка с таким номером уже выполнена, выводится сообщение и устанавливается флаг отказа (Отказ = Истина).

Если ВыборкаДетальныеЗаписи.Заявка = Движение.Заявка Тогда

Отказ = Истина;

Сообщить ("Заявка под номер " + Движение.Заявка.Номер + " уже выполнена");

КонецЕсли

* **Проверка условий: Обработка заявки**: Проверяется, была ли заявка обработана. Если нет, устанавливается флаг отказа.

Если Не ПроверитьОбработку(Заявка) Тогда

Отказ = Истина;

Сообщить("Заявка под номером " + Движение.Заявка.Номер + " не обработана");

КонецЕсли;

управление заявками на обслуживание, проверяет их уникальность, статус обработки, регистрирует выполнение заявок и обновляет статусы заявок на "Выполнено".