**Шаги**

## **Начальные действия**

**Ознакомление с таблицами**

|  |  |  |  |
| --- | --- | --- | --- |
| Works - заказы на проведение исследований | WorkItem - элементы заказа (заказанное исследование) | Analiz - спецификации исследования | Employee- сотрудники |
| ALTER TABLE [dbo].[Works] ADD DEFAULT ((0)) FOR [IS\_Complit]  GO  ALTER TABLE [dbo].[Works] ADD DEFAULT (getdate()) FOR [CREATE\_Date]  GO  ALTER TABLE [dbo].[Works] ADD DEFAULT (getdate()) FOR [Material\_Get\_Date]  GO  ALTER TABLE [dbo].[Works] ADD DEFAULT (getdate()) FOR [Material\_Reg\_Date]  GO  ALTER TABLE [dbo].[Works] ADD DEFAULT ((0)) FOR [Is\_Del] | ALTER TABLE [dbo].[WorkItem] ADD DEFAULT (getdate()) FOR [CREATE\_DATE]  GO  ALTER TABLE [dbo].[WorkItem] ADD DEFAULT ((0)) FOR [Is\_Complit]  GO  ALTER TABLE [dbo].[WorkItem] ADD DEFAULT ((1)) FOR [Is\_Print]  GO  ALTER TABLE [dbo].[WorkItem] ADD DEFAULT ((0)) FOR [Is\_Select]  GO  ALTER TABLE [dbo].[WorkItem] ADD DEFAULT ((1)) FOR [Is\_NormTextPrint] | ALTER TABLE [dbo].[WorkItem] WITH NOCHECK ADD CONSTRAINT [FK\_\_WorkItem\_\_ID\_ANA\_\_1F98B2C1] FOREIGN KEY([ID\_ANALIZ])  REFERENCES [dbo].[Analiz] ([ID\_ANALIZ])  GO | ALTER TABLE [dbo].[Employee] ADD DEFAULT (suser\_sname()) FOR [Login\_Name]  GO  ALTER TABLE [dbo].[Employee] ADD DEFAULT ('') FOR [Name]  GO  ALTER TABLE [dbo].[Employee] ADD DEFAULT ('') FOR [Patronymic]  GO  ALTER TABLE [dbo].[Employee] ADD DEFAULT ('') FOR [Surname]  GO  ALTER TABLE [dbo].[Employee] ADD DEFAULT ((0)) FOR [Archived]  GO  ALTER TABLE [dbo].[Employee] ADD DEFAULT ((0)) FOR [IS\_Role]  GO  ALTER TABLE [dbo].[Employee] ADD DEFAULT ((0)) FOR [Role] |
| CREATE TABLE [dbo].[Works](  [Id\_Work] [int] IDENTITY(1,1) NOT NULL,  [IS\_Complit] [bit] NOT NULL,  [CREATE\_Date] [datetime] NULL,  [Close\_Date] [datetime] NULL,  [Id\_Employee] [int] NULL,  [ID\_ORGANIZATION] [int] NULL,  [Comment] [varchar](255) NULL,  [Print\_Date] [datetime] NULL,  [Org\_Name] [varchar](50) NULL,  [Part\_Name] [varchar](50) NULL,  [Org\_RegN] [int] NULL,  [Material\_Type] [smallint] NULL,  [Material\_Get\_Date] [datetime] NULL,  [Material\_Reg\_Date] [datetime] NULL,  [MaterialNumber] [decimal](8, 2) NULL,  [Material\_Comment] [varchar](255) NULL,  [FIO] [varchar](255) NOT NULL,  [PHONE] [varchar](50) NULL,  [EMAIL] [varchar](255) NULL,  [Is\_Del] [bit] NOT NULL,  [Id\_Employee\_Del] [int] NULL,  [DelDate] [datetime] NULL,  [Price] [decimal](8, 2) NULL,  [ExtRegN] [varchar](255) NULL,  [MedicalHistoryNumber] [varchar](255) NULL,  [DoctorFIO] [varchar](255) NULL,  [DoctorPhone] [varchar](255) NULL,  [OrganizationFax] [varchar](255) NULL,  [OrganizationEmail] [varchar](255) NULL,  [DoctorEmail] [varchar](255) NULL,  [StatusId] [smallint] NULL,  [SendToOrgDate] [datetime] NULL,  [SendToClientDate] [datetime] NULL,  [SendToDoctorDate] [datetime] NULL,  [SendToFax] [datetime] NULL,  [SendToApp] [datetime] NULL,  PRIMARY KEY CLUSTERED  (  [Id\_Work] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY] | CREATE TABLE [dbo].[WorkItem](  [ID\_WORKItem] [int] IDENTITY(1,1) NOT NULL,  [CREATE\_DATE] [datetime] NULL,  [Is\_Complit] [bit] NOT NULL,  [Close\_Date] [datetime] NULL,  [Id\_Employee] [int] NULL,  [ID\_ANALIZ] [int] NULL,  [Id\_Work] [int] NULL,  [Is\_Print] [bit] NOT NULL,  [Is\_Select] [bit] NOT NULL,  [Is\_NormTextPrint] [bit] NULL,  [Price] [decimal](8, 2) NULL,  [Id\_SelectType] [int] NULL,  PRIMARY KEY CLUSTERED  (  [ID\_WORKItem] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY] | CREATE TABLE [dbo].[Analiz](  [ID\_ANALIZ] [int] IDENTITY(1,1) NOT NULL,  [IS\_GROUP] [bit] NULL,  [MATERIAL\_TYPE] [int] NULL,  [CODE\_NAME] [varchar](50) NULL,  [FULL\_NAME] [varchar](255) NULL,  [ID\_ILL] [int] NULL,  [Text\_Norm] [varchar](255) NULL,  [Price] [decimal](8, 2) NULL,  [NormText] [varchar](2048) NULL,  [UnNormText] [varchar](2048) NULL,  PRIMARY KEY CLUSTERED  (  [ID\_ANALIZ] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY] | CREATE TABLE [dbo].[Employee](  [Id\_Employee] [int] IDENTITY(1,1) NOT NULL,  [Login\_Name] [varchar](50) NOT NULL,  [Name] [varchar](50) NOT NULL,  [Patronymic] [varchar](50) NOT NULL,  [Surname] [varchar](50) NOT NULL,  [Email] [varchar](50) NULL,  [Post] [varchar](50) NULL,  [CreateDate] [datetime] NULL,  [UpdateDate] [datetime] NULL,  [EraseDate] [datetime] NULL,  [Archived] [bit] NOT NULL,  [IS\_Role] [bit] NOT NULL,  [Role] [int] NULL,  [FULL\_NAME] AS (([SURNAME]+' ')+[NAME]),  PRIMARY KEY CLUSTERED  (  [Id\_Employee] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY] |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Organization | PrintTemplate | SelectType | TemplateType | WorkStatus |
| CREATE TABLE [dbo].[Organization](  [ID\_ORGANIZATION] [int] IDENTITY(1,1) NOT NULL,  [ORG\_NAME] [varchar](255) NULL,  [TEMPLATE\_FN] [varchar](255) NULL,  [Id\_PrintTemplate] [int] NULL,  [Email] [varchar](255) NULL,  [SecondEmail] [varchar](255) NULL,  [Fax] [varchar](255) NULL,  [SecondFax] [varchar](255) NULL,  PRIMARY KEY CLUSTERED  (  [ID\_ORGANIZATION] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY] | CREATE TABLE [dbo].[PrintTemplate](  [Id\_PrintTemplate] [int] IDENTITY(1,1) NOT NULL,  [TemplateName] [varchar](255) NULL,  [CreateDate] [datetime] NULL,  [Ext] [varchar](10) NULL,  [Comment] [varchar](255) NULL,  [TemplateBody] [image] NULL,  [Id\_TemplateType] [int] NULL,  PRIMARY KEY CLUSTERED  (  [Id\_PrintTemplate] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY] TEXTIMAGE\_ON [PRIMARY] | CREATE TABLE [dbo].[SelectType](  [Id\_SelectType] [int] IDENTITY(1,1) NOT NULL,  [SelectType] [varchar](50) NULL,  PRIMARY KEY CLUSTERED  (  [Id\_SelectType] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY] | CREATE TABLE [dbo].[TemplateType](  [Id\_TemplateType] [int] IDENTITY(1,1) NOT NULL,  [TemlateVal] [varchar](50) NULL,  [Comment] [varchar](255) NULL,  PRIMARY KEY CLUSTERED  (  [Id\_TemplateType] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY] | CREATE TABLE [dbo].[WorkStatus](  [StatusID] [smallint] IDENTITY(1,1) NOT NULL,  [StatusName] [varchar](255) NULL,  CONSTRAINT [PK\_WorkStatus] PRIMARY KEY CLUSTERED  (  [StatusID] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  ) ON [PRIMARY] |

Индексы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CREATE UNIQUE NONCLUSTERED INDEX [XAKLoginName] ON [dbo].[Employee]  (  [Login\_Name] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, IGNORE\_DUP\_KEY = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  GO | CREATE NONCLUSTERED INDEX [XIF1Organization] ON [dbo].[Organization]  (  [Id\_PrintTemplate] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  GO | CREATE NONCLUSTERED INDEX [XIF1PrintTemplate] ON [dbo].[PrintTemplate]  (  [Id\_TemplateType] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  GO | CREATE NONCLUSTERED INDEX [XIF3WorkItem] ON [dbo].[WorkItem]  (  [Id\_Employee] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  GO | CREATE NONCLUSTERED INDEX [XIF4WorkItem] ON [dbo].[WorkItem]  (  [ID\_ANALIZ] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  GO |
| CREATE NONCLUSTERED INDEX [XIF5WorkItem] ON [dbo].[WorkItem]  (  [Id\_Work] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  GO | CREATE NONCLUSTERED INDEX [XIF6WorkItem] ON [dbo].[WorkItem]  (  [Id\_SelectType] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  GO | CREATE NONCLUSTERED INDEX [XIF1Works] ON [dbo].[Works]  (  [Id\_Employee] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  GO | CREATE NONCLUSTERED INDEX [XIF2Works] ON [dbo].[Works]  (  [ID\_ORGANIZATION] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]  GO | CREATE NONCLUSTERED INDEX [XIF3Works] ON [dbo].[Works]  (  [Id\_Employee\_Del] ASC  )WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, SORT\_IN\_TEMPDB = OFF, DROP\_EXISTING = OFF, ONLINE = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY] |

**Ознакомление с функциями**

|  |  |  |  |
| --- | --- | --- | --- |
| F\_EMPLOYEE\_FULLNAME | F\_EMPLOYEE\_GET | F\_WORKITEMS\_COUNT\_BY\_ID\_WORK | F\_WORKS\_LIST |
| CREATE FUNCTION [dbo].[F\_EMPLOYEE\_FULLNAME] (  @ID\_EMPLOYEE INT  )  RETURNS VARCHAR(101)  AS  BEGIN  DECLARE @RESULT VARCHAR(101)  SET @ID\_EMPLOYEE = COALESCE(@ID\_EMPLOYEE, dbo.F\_EMPLOYEE\_GET())  IF @ID\_EMPLOYEE = -1  SET @RESULT = ''  ELSE  SELECT @RESULT = SURNAME + ' ' + UPPER(SUBSTRING(NAME, 1, 1)) + '. ' +  UPPER(SUBSTRING(PATRONYMIC, 1, 1)) + '.' FROM Employee  WHERE ID\_EMPLOYEE = @ID\_EMPLOYEE  SET @RESULT = RTRIM (REPLACE(@RESULT, '. .', ''))    IF @RESULT = ''  SELECT @RESULT = LOGIN\_NAME FROM Employee Where Id\_Employee = @ID\_Employee  RETURN @RESULT  END | CREATE FUNCTION [dbo].[F\_EMPLOYEE\_GET] ()  RETURNS  INT  AS  BEGIN  -- Возвращает идентификатор текщего пользователя  DECLARE  @RESULT INT  SELECT  @RESULT = ID\_EMPLOYEE  FROM  EMPLOYEE  WHERE  LOGIN\_NAME = SYSTEM\_USER  RETURN  @RESULT  END | CREATE FUNCTION [dbo].[F\_WORKITEMS\_COUNT\_BY\_ID\_WORK] (  @id\_work int,  @is\_complit bit  )  RETURNS int  AS  BEGIN  -- количество готовых / не готовых анализов для заказа  declare @result int  select @result = count(\*) from workitem  where id\_work = @id\_work  -- не является групповым  and id\_analiz  not in  (select id\_analiz  from analiz where is\_group = 1)    and is\_complit = @is\_complit  Return @result  END | CREATE FUNCTION [dbo].[F\_WORKS\_LIST] (  )  RETURNS @RESULT TABLE  (  ID\_WORK INT,  CREATE\_Date DATETIME,  MaterialNumber DECIMAL(8,2),  IS\_Complit BIT,  FIO VARCHAR(255),  D\_DATE varchar(10),  WorkItemsNotComplit int,  WorkItemsComplit int,  FULL\_NAME VARCHAR(101),  StatusId smallint,  StatusName VARCHAR(255),  Is\_Print bit  )  AS  -- СПИСОК РАБОТ  begin  insert into @result  SELECT  Works.Id\_Work,  Works.CREATE\_Date,  Works.MaterialNumber,  Works.IS\_Complit,  Works.FIO,  convert(varchar(10), works.CREATE\_Date, 104 ) as D\_DATE,  dbo.F\_WORKITEMS\_COUNT\_BY\_ID\_WORK(works.Id\_Work,0) as WorkItemsNotComplit,  dbo.F\_WORKITEMS\_COUNT\_BY\_ID\_WORK(works.Id\_Work,1) as WorkItemsComplit,  dbo.F\_EMPLOYEE\_FULLNAME(Works.Id\_Employee) as EmployeeFullName,  Works.StatusId,  WorkStatus.StatusName,  case  when (Works.Print\_Date is not null) or  (Works.SendToClientDate is not null) or  (works.SendToDoctorDate is not null) or  (Works.SendToOrgDate is not null) or  (Works.SendToFax is not null)  then 1  else 0  end as Is\_Print  FROM  Works  left outer join WorkStatus on (Works.StatusId = WorkStatus.StatusID)  where  WORKS.IS\_DEL <> 1  order by id\_work desc -- works.MaterialNumber desc  return  end |

Добавление данных (см. скрипт).

## Задачи

### Задача 1-го уровня

|  |  |  |
| --- | --- | --- |
| **Функция** | **Связанные объекты** | **Недочёты и потенциальные проблемы производительности** |
| F\_WORKS\_LIST | Works, F\_WORKITEMS\_COUNT\_BY\_ID\_WORK, F\_EMPLOYEE\_FULLNAME, WorkStatus | * Зависит от несколько других функции, что значит, что не оптимальная структуризация или формирование этих функций также влияет на производительность этой. Также сама необходимость вызывать другие функции может влиять на производительность. * Для поля Is\_Print проверяется множество условий, возможно это можно как-то оптимизировать. * Не для всех таблиц указывается схема (Works и WorkStatus), что может стать проблемой в случае существования/появления одноименных таблиц. * Можно попытаться оптимизировать запросы F\_WORKITEMS\_COUNT\_BY\_ID\_WORK (для id\_work и значения is\_complit), чтобы уменьшить количество раз сканирования данных. |
| Рассмотрим функции связанные с F\_WORKS\_LIST | | |
| F\_WORKITEMS\_COUNT\_BY\_ID\_WORK | Workitem, Analiz | * Можно использовать in (where is\_group=0) вместо not in (where is\_group=1), чтобы не нужно было выполнять запрос целиком. * Нет параметров по умолчанию для @id\_work и в случае недействительного или не существующих данных функция может возвращать ошибку. * Можно получить аналогичный вывод через встроенные функции. |
| F\_EMPLOYEE\_FULLNAME | F\_EMPLOYEE\_GET(), Employee | * Обращение к F\_EMPLOYEE\_GET() излишне, она просто связана с таблицей Employee. Можно сразу обратится к таблице Employee и избавится от дополнительных обращений. * Можно немного переделать значение [FULL\_NAME] в Employee, чтобы получить аналогичный вывод. * Формирование результата через объединение строк и манипуляции с символами вместо использования встроенных функций. * Большое количество символов для результата (VARCHAR (101)), можно немного сократить, чтобы в перспективе занимало меньше памяти. |

### Задача 2-го уровня

|  |  |  |
| --- | --- | --- |
| **Исходная функция** | **Измененная функция** | **Комментарии** |
| CREATE FUNCTION [dbo].[F\_WORKS\_LIST] (  )  RETURNS @RESULT TABLE  (  ID\_WORK INT,  CREATE\_Date DATETIME,  MaterialNumber DECIMAL(8,2),  IS\_Complit BIT,  FIO VARCHAR(255),  D\_DATE varchar(10),  WorkItemsNotComplit int,  WorkItemsComplit int,  FULL\_NAME VARCHAR(101),  StatusId smallint,  StatusName VARCHAR(255),  Is\_Print bit  )  AS  -- СПИСОК РАБОТ  begin  insert into @result  SELECT  Works.Id\_Work,  Works.CREATE\_Date,  Works.MaterialNumber,  Works.IS\_Complit,  Works.FIO,  convert(varchar(10), works.CREATE\_Date, 104 ) as D\_DATE,  dbo.F\_WORKITEMS\_COUNT\_BY\_ID\_WORK(works.Id\_Work,0) as WorkItemsNotComplit,  dbo.F\_WORKITEMS\_COUNT\_BY\_ID\_WORK(works.Id\_Work,1) as WorkItemsComplit,  dbo.F\_EMPLOYEE\_FULLNAME(Works.Id\_Employee) as EmployeeFullName,  Works.StatusId,  WorkStatus.StatusName,  case  when (Works.Print\_Date is not null) or  (Works.SendToClientDate is not null) or  (works.SendToDoctorDate is not null) or  (Works.SendToOrgDate is not null) or  (Works.SendToFax is not null)  then 1  else 0  end as Is\_Print  FROM  Works  left outer join WorkStatus on (Works.StatusId = WorkStatus.StatusID)  where  WORKS.IS\_DEL <> 1  order by id\_work desc -- works.MaterialNumber desc  return  end | CREATE FUNCTION [dbo].[F\_WORKS\_LIST] (  )  RETURNS @RESULT TABLE  (  ID\_WORK INT,  CREATE\_Date DATETIME,  MaterialNumber DECIMAL(8,2),  IS\_Complit BIT,  FIO VARCHAR(255),  D\_DATE varchar(10),  WorkItemsNotComplit int,  WorkItemsComplit int,  FULL\_NAME VARCHAR(101),  StatusId smallint,  StatusName VARCHAR(255),  Is\_Print bit  )  AS  -- СПИСОК РАБОТ  begin  insert into @result  SELECT  Works.Id\_Work,  Works.CREATE\_Date,  Works.MaterialNumber,  Works.IS\_Complit,  Works.FIO,  convert(varchar(10), works.CREATE\_Date, 104 ) as D\_DATE,  COUNT(CASE WHEN WorkItem.is\_complit = 0 THEN 1 ELSE NULL END) as WorkItemsNotComplit,  COUNT(CASE WHEN WorkItem.is\_complit = 1 THEN 1 ELSE NULL END) as WorkItemsComplit,  dbo.F\_EMPLOYEE\_FULLNAME(Works.Id\_Employee) as EmployeeFullName,  Employee.FULL\_NAME as EmployeeFullName,  --RTRIM(REPLACE(Employee.SURNAME + ' ' + UPPER(SUBSTRING(Employee.NAME, 1, 1)) + '. ' + UPPER(SUBSTRING(Employee.PATRONYMIC, 1, 1)) + '.', '. .', '')) AS EmployeeFullName,  Works.StatusId,  WorkStatus.StatusName,  case  when (Works.Print\_Date is not null) or  (Works.SendToClientDate is not null) or  (works.SendToDoctorDate is not null) or  (Works.SendToOrgDate is not null) or  (Works.SendToFax is not null)  then 1  else 0  end as Is\_Print  FROM  Works  left outer join WorkStatus on (Works.StatusId = WorkStatus.StatusID)  left join Employee on (Works.Id\_Employee=Employee.Id\_Employee)  left join WorkItem on (Works.Id\_Work = WorkItem.Id\_Work)  where  WORKS.IS\_DEL <> 1  order by id\_work desc  return  end | * Убрали обращение к другой функции и вместо этого напрямую обращаемся к таблице Employee. * Чтобы вывод был точно, как в оригинальной функции можем использовать закомментированную строку (RTRIM…) со встроенными функциями. * Используем встроенные функции (COUNT) вместо вложенного запроса. |

### Задача 3-го уровня

Если для оптимизации требуется создание новых таблиц, столбцов, триггеров или хранимых процедур (функций), то это может усложнить структуру базы данных и, следовательно, её понимание, обращение и обслуживание. Также большее количество элементов означает, что есть больше мест для возникновения ошибок, например, с неправильным формированием индексов или противоречивыми структурами данных. Помимо этого, увеличение объема БД может привести к проблемам производительности из-за времени, которое будет необходимо для обработки запросов (которые могут быть не оптимальными).

## Сторонние источники:

Для генерации данных сначала использовался немного изменений ответ ChatGPT-3.5-turbo на запрос: «создай генератор тестовых данных для этой базы данных» (с описанием создания таблиц).

Для выполнения задач 1-го и 2-го уровня были также использованы ответы на запросы: «недостатки функции "F\_WORKITEMS\_COUNT\_BY\_ID\_WORK"», «исправь эту функцию» (с указание функции F\_WORKS\_LIST).