1. После развертывания базы необходимо выполнить:

CREATE LOGIN [user\_login1]

WITH PASSWORD = 'password123!'

CREATE LOGIN [user\_login2]

WITH PASSWORD = 'password123!'

-- Создание пользователя в базе данных

USE [test4]

Sp\_dropuser 'user\_login1'

Sp\_dropuser 'user\_login2

'

CREATE USER [user\_login1] FOR LOGIN [user\_login1]

CREATE USER [user\_login2] FOR LOGIN [user\_login2]

-- Добавление пользователя к роли

EXEC sp\_addrolemember @rolename = 'Manager', @membername = 'user\_login1';

EXEC sp\_addrolemember @rolename = 'dictionary', @membername = 'user\_login2';

1. Требуется создать БД для хранения заявок на основе полученных данных: произвести декомпозицию, разделение на сущности, выполнить необходимые связи.

Созданы таблицы:

position

division

department

applicationReason

employe

furnitureOrder

usersLog

status

furnitureOrderHistory

resolutionSolution

resolution

Select \* from sys.tables

1. Требуется загрузить JSON в БД и обеспечить разбор данных с последующим размещением в нормализованном виде в структуре БД.

Для загрузки реализована процедура: furnitureOrder\_Save

Пример вызова:

execute ParseJsonData

@JsonData='

{

"furnitureOrderId": 0,

"applicant": {

"id": "9948BBE4-8464-466C-94A9-8CA214732EDB",

"name": "Василий",

"surname": "Петров",

"department": "Департамент ИТ"

},

"applicationReason": "поломка",

"coordinatingManager": {

"id": "9221F291-3F5B-45A8-888E-43EAD1741FD8",

"name": "Владимир",

"surname": "Александров",

"department": "Департамент хозяйственной деятельности",

"division": "отдел эксплуатации здания",

"position": "заведующий хозяйством"

},

"resolution": "одобрено",

"resolutionPeriod": 4,

"resolutionSolution": "замена оборудования",

"status": "в работе"

}'

1. Следует создать хранимые процедуры для создания/обновления/удаления записей во всех таблицах

Select \* from sys.objects Where type='P'

ParseJsonData

furnitureOrder\_Save

furnitureOrder\_Delete

Generator

applicationReason\_Delete

department\_Delete

division\_Delete

employe\_Delete

position\_Delete

resolution\_Delete

resolutionSolution\_Delete

status\_Delete

status\_Save

resolutionSolution\_Save

resolution\_Save

position\_Save

division\_Save

department\_Save

applicationReason\_Save

employe\_Save

1. Для просмотра статистики использования, нужно создать view, отображающее

информацию о заявках из БД с возможностью выбора отчётного периода.

Select \* from sys.objects Where type='V'

Select \* from furnitureOrderView Where data\_Order between '20240507' and '20240508'

1. Добавить объекты, позволяющие управлять пользователями, вести журнал операций для фиксации всех действий пользователей
2. В процедуры вставлен лог действий пользователя. То, что присылает пользователь сохраняется в таблицу UserLog
3. На таблице furnitureOrder сделаны триггеры, которые сохраняют все изменения в таблице (furnitureOrderHistory)

Select \* from sys.objects Where type='TR'

furnitureOrder\_On\_HistoryDelete

furnitureOrder\_On\_HistoryInsert

furnitureOrder\_On\_HistoryUpdate

1. Если нужно фиксировать все действия пользователя, в том числе создание процедур, таблиц и д, то можно создать триггер базы данных.
2. Добавить объекты, позволяющие управлять пользователями, вести журнал операций для фиксации всех действий пользователей

Созданы логины и пользователи:

‘user\_login1’

‘user\_login2’

Созданы роли :

'Manager',

'dictionary'

Пользователям присвоены роли:

EXEC sp\_addrolemember @rolename = 'Manager', @membername = 'user\_login1';

EXEC sp\_addrolemember @rolename = 'dictionary', @membername = 'user\_login2';

На процедурах выставлены гранты на запуск.

Гранты на DML операции для пользователей убраны, изменения можно вносить только через процедуры.