**Резервное копирование и восстановление данных**

Резервное копирование (РК) – копировароние всей БД\ее части на независимое устройство постоянного хранения данных для обеспечения их сохранности

**При планировании резервного копирования учитывается:**

1. Зависимость бизнес-процессов от данных
2. Надежность среды, в которой функционирует СУБД
3. Стоимость простоя сервера
4. Стоимость простоя приложения

**Схемы РК:**

1. Ориентированная на быстрое восстановление
2. Ориентированная на быстрое создание резервной копии

**Этапы PK MK SQL:**

1. Полное резервное копирование
2. Разностное резервное копирование – PK БД с момента полного РК БД (последнего)
3. РК журнала транзакций (выполненных команд) – позволяет скопировать часть журнала транзакции, повышающуюся с момента любой последней PK
4. PK отдельных файлов, входящих в состав БД
5. РК ФС

**Типичные схемы РК:**

* только полное
* полное и журнал и журнал транзакции – наименьшее влияние на работу сервера в течении недели
* полное, разностное и журнал транзакции
* полное копирование файлов БД, разностное, журнал транзакции

БД -> ПКМ -> задачи -> создать резервную копию…

**Команды резервного копирования**

**Полная РК:**

BACKUP DATABASE ИмяБД

TO DISK = ‘имя файла.bok’

**Разностное копирование:**

BACKUP DATABASE ИмяБД

TO DISK = ‘Имя файла\_diff.bok’

WITH DIFFERENTIAL

BACKUP LOG ИмяБД

TO DISK = ‘имя файла\_log1.bok’

Восстановление – процесс восстановления БД из РК

БД -> ПКМ –> задачи -> Восстановить

Шаги восстановления – тот же порядок, что и у создания

Если РК несколько, то у кажой команды, кроме последней в конце WITH NORECOVERY

Для запуска восстановления – WITH RECOVERY, прервать связи с БД

**Команды:**

RESTORE DATABSE ИмяБД,

FROM DISK = ‘имя файла.bok’

WITH NORECOVERY

WITH RESTRICTED\_USER – запрещает пользователю работать с БД, пока идёт РК (закрывает все активные соединения)

**Создание РК в виде скриптов:**

БД -> ПКМ -> задачи -> сформировать скрипты -> выбор элементов -> (в следущем окне) Дополнительно -> Внести в скрипт DROP и CREATE; типы данных для внесения в скрипт (схема и данны);внести в скрипт триггеры (true)

**При восстановлении данных с помощью скрипта**

SET IDENITY\_INSERT таблица ON – отключение запрета на вставку автоинкрементных значений. После вставки – та же команда с параметром OFF вместо ON.

SET DATEFORMAT mdy

SET DATEFORMAT dmy

DECLARE @date DATETIME

SET DATEFORMAT mdy

DECLARE @date DATETIME = ’10-20-2023’

SELECT @date

**Команды массовой вставки данных**

BULK INSERT – вставляет данные из текстовых файлов

BULK INSERT таблица

FROM ‘полное имя файла’

WITH (пар-ры)

txt/tsv/csv

c:\temp\games.csv

BULK INSERT Games

FROM ‘c:\temp\games.csv’

WITH (FIRSTROW = 2, --в 1ой-заголовки

,FILEFORMAT = ‘csv’

,FIELDQUOTE = ‘”’

,FIELDTERMINATOR = ‘;’

,ROWTERMINATOR = ‘\n’)

Функции для работы со строками :

TRIM(значение)

STRING\_SPLIT (столбец, разделитель) – делит данные в ячейке по разделителю

SELECT value / SELECT TRIM (value)

FROM STRING\_SPLIT(‘#1, #2, #3’, ‘,’)

SELECT GameId, Name, TRIM(value)

FROM Games

CROSS APPLY STRING\_SPLIT (description, ‘.’) –составляет значения строк одной таблицы строками другой

**Указывается псевдоним**

CROSS APPLY STRING\_SPLIT (description, ‘.’) AS t1

CROSS APPLY STRING\_SPLIT (description, ‘;’) AS t2

PIVOT – запрос, позволяющий повернуться результат

Особенность – должны знать заголовки стобцов

Пишется в разделе FROM

SELECT название столбцов-заголовков

FROM таблица AS t

PIVOT

(

Агрегатная функция (столбец)

FOR столбец IN (столбцы-заголовки)

) AS put

SELECT [1], [2], [3]

FROM (SELECT CategoryId, GameId

FROM Games) AS t

PIVOT

(

COUNT(GameId)

FOR CategoryId IN ([1], [2], [3])

) AS put