Что такое первичный ключ? Для чего он нужен?

Первичный ключ это столбец или набор столбцов в таблице, каждое поле или набор полей в которых имеет уникальное значение и позволяет однозначно идентифицировать строку, к которой он принадлежит.

Первичный ключ нужен для ускорения поиска данных в таблице, а также для создания связей с другими таблицами.

Чем отличается первичный ключ от ограничения уникальности?

В первую очередь тем, что первичный ключ по умолчанию имеет ограничение not null, что позволяет всегда однозначно определять строку в таблице, а также тем, что первичный ключ может быть у таблицы только один, а ограничений уникальности может быть сколько угодно. Поэтому несмотря на то, что некоторые диалекты технически позволяют устанавливать связь между таблицами используя столбцы с ограничениями unique not null, для связей все равно стараются использовать только ограничение первичный ключ, которое позволяет определить каждую таблицу однозначно. Первичный ключ также значительно упрощает нормализацию таблиц, что важно при построении реляционной базы данных.

Что такое индекс и для чего он нужен? В каком случае целесообразно использовать индексы?

Индекс это создаваемый объект в базе данных, где содержится указание на хранимые в какой-то таблице данные (некластеризованный индекс) или сами данные (кластеризованный индекс). Индекс значительно повышает скорость поиска данных в таблице. К примеру, без индекса для нахождения данных в таблице с 1000000 строк, пришлось бы по очереди проверять каждую запись, а при наличии индекса это можно сделать за 20 действий.

Недостатками индекса являются: необходимость хранения, индекс занимает место, а также тот факт, что при внесении изменений в таблицу все индексы также модифицируются, что замедляет работу базы данных. Поэтому индексы целесообразно использовать на тех таблицах, которые имеют много записей, к которым часто обращаются, но редко обновляют.

Чем отличается функция от хранимой процедуры?

Пользовательские функции это сохраненный пользователем код, который может вызываться по команде как и любая другая функция, предназначенный в первую очередь для вычисления и возвращающий одно значение.

Пользовательская хранимая процедура это сохраненный пользователем набор инструкций T-SQL, или же ссылка на метод.

Отличия:

1. Функция всегда возвращает одно значение, процедура может как возвращать любое количество значений, так и не возвращать ничего.
2. Функции можно использовать в операторе select, процедуры нельзя использовать в select, where, join и т.д.
3. В функциях можно использовать select, но нельзя использовать insert, update, delete, в процедурах можно использовать select, insert, delete, update и т.д.
4. Внутри функции нельзя вызвать процедуры, а в процедурах можно вызывать как функции, таи другие процедуры.
5. В функциях запрещено использование транзакций, в то время как в процедурах это разрешено.
6. Конструкцию для обработки ошибок try catch нельзя использовать в функциях. Так же как нельзя использовать инструкцию raiserror. В хранимых процедурах можно использовать и конструкцию try catch, и инструкцию raiserror.
7. В функциях нельзя использовать динамический SQL, а в процедурах можно.
8. В функциях можно использовать только входные параметры, а в процедурах можно использовать как входные, так и выходные параметры.

В чем отличие хранимой процедуры от триггера?

Триггер это процедура, которая выполняется автоматически как реакция на событие с данными в существующей таблице.

Процедура отличается от триггера тем, что не может быть вызвана автоматически в ответ на какое-то действие. Она обязательно должна быть вызвана другой программой, которая передаст на сервер имя процедуры для исполнения.