

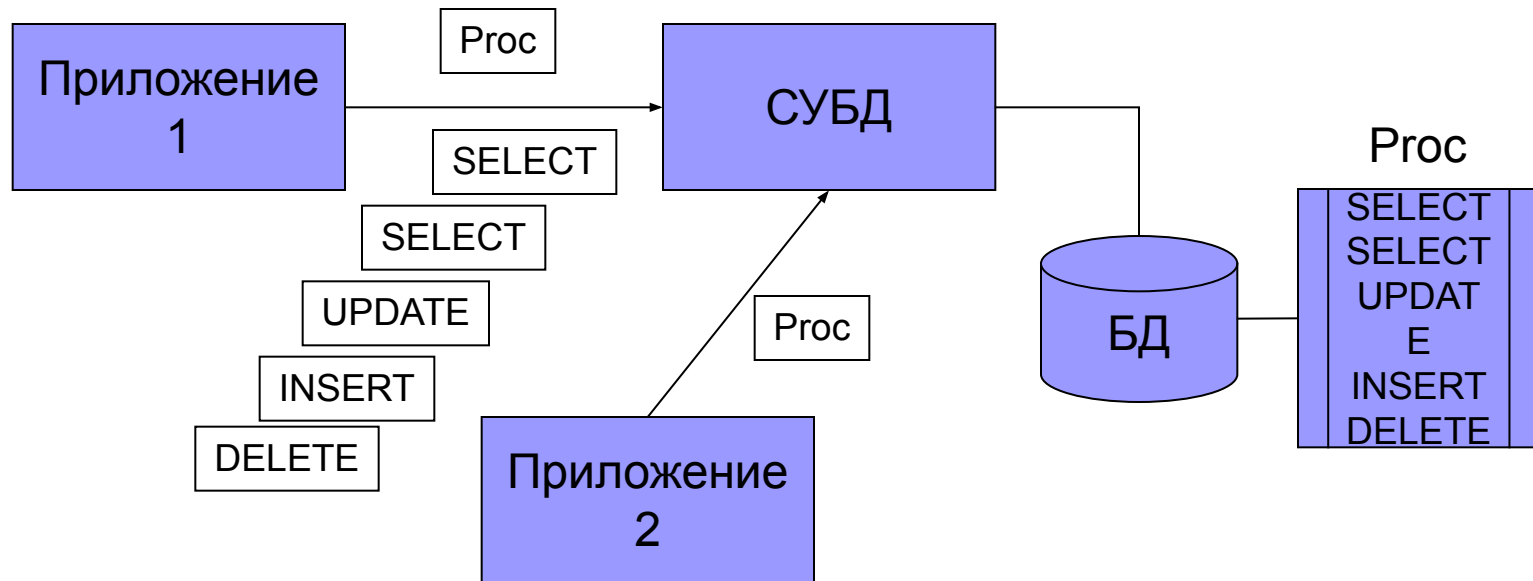


# ОСНОВЫ проектирования баз данных

Хранимые процедуры

# Хранимые процедуры

**Хранимая процедура** (Stored Procedure) – это именованный набор команд языка Transact-SQL, хранящийся на сервере в качестве самостоятельного объекта БД



- SP хранится на сервере (пакет – на клиенте)
- Для вызова используется имя (у пакета нет имени)
- SP могут вызвать друг друга (пакеты не могут)
- Изменения вносятся в одном месте, на сервере (в пакеты – на всех клиентах)
- Реализуется модульность
- Уменьшается объем передаваемой информации по сети

# Типы хранимых процедур

- **Системные ХП** – входят в состав SQL Server, реализуют все действия администрирования сервера, начинаются с sp\_, хранятся в БД master. Контекст выполнения любой
- **Пользовательские ХП (User-Defined SP)** – размещаются в пользовательских БД и выполняются в контексте только одной БД
- **Временные ХП** – существуют только в рамках одного соединения с сервером (временно хранятся в tempDB)

## **Хранимые процедуры обладают следующими преимуществами.**

- Повторное использование кода – после создания хранимой процедуры ее можно вызывать из любых приложений и SQL-запросов.
- Сокращение сетевого трафика – вместо нескольких запросов экономнее послать серверу запрос на выполнение хранимой процедуры и сразу получить ответ.
- Безопасность – действия не приведут к нарушению целостности данных, т.к. для выполнения хранимой процедуры пользователь должен иметь привилегию.
- Простота доступа – хранимые процедуры позволяют инкапсулировать сложный код и оформить его в виде простого вызова.
- Выполнение бизнес-логики – хранимые процедуры позволяют перенести код сохранения целостности БД из прикладной программы на сервер БД. Бизнес-логика в виде хранимых процедур не зависит от языка разработки приложения.

# Создание хранимых процедур

```
CREATE PROCEDURE имя_процедуры  
( [IN | OUT | INOUT] имя_параметра тип[, ... ] )  
[характеристика ...] тело_процедуры
```

- *IN* – данные передаются строго внутрь хранимой процедуры;
- *OUT* – данные передаются строго из хранимой процедуры;
- *INOUT* – значение этого параметра как принимается во внимание внутри процедуры, так и сохраняет свое значение при выходе из нее.

# Создание хранимых процедур

```
mysql> CREATE PROCEDURE Hello_World(<
-> BEGIN
->   SELECT('Hello, world!');
-> END
-> //
Query OK, 0 rows affected (0.00 sec)
```

DELIMITER //

CREATE PROCEDURE  
Hello\_World()

BEGIN

SELECT('Hello, world!');

END

//

```
mysql> CALL Hello_World;
-> //
+-----+
| Hello, world! |
+-----+
| Hello, world! |
+-----+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.00 sec)
```

# Создание хранимых процедур

```
CREATE PROCEDURE P1()  
BEGIN  
DECLARE S VARCHAR(20);  
SELECT weight INTO S FROM Parts ORDER BY Weight  
DESC LIMIT 1;  
SELECT(S);  
END  
//
```

# Управление хранимыми процедурами

- Изменение (кроме имени). Процедура заменяется полностью:  
`ALTER PROCEDURE <имя процедуры>`  
далее как в `CREATE PROCEDURE`
- Изменение имени – при помощи системной хранимой процедуры  
`sp_rename '<старое имя>', '<новое имя>' [, 'object']`  
(она же используется для переименования таблиц и столбцов в таблице) например,  
`exec sp_rename 'GetReader', 'GetAllReaders', 'object'`
- Удаление хранимой процедуры  
`DROP PROCEDURE <имя процедуры>`



# Управляющие конструкции Transact-SQL

- BEGIN...END – блок
- BREAK – выход из цикла
- CONTINUE – переход к началу цикла
- GOTO – безусловный переход
- IF...ELSE – ветвление
- RETURN – возврат из хранимой процедуры
- WHILE – цикл с предусловием
- TRY...CATCH – обработка исключения