

Троицкий АТК - филиал ФГБОУ ВО МГТУ ГА  
Отделение: Программирование  
в компьютерных системах

# *Хранимые процедуры и функции*

Предмет: РИЗУБД

Предназначено для курсантов 4 курса

Выполнил: преподаватель ЦК ПЭВМ, Черевкова О.А.

- **Хранимые процедуры понятие, назначение.**
- **Преимущества и недостатки использования ХП.**
- **Виды ХП. Синтаксис. Создание и вызов. Изменение и удаление.**

# **Хранимые процедуры понятие, назначение**

**Хранимая процедура представляет собой набор SQL-операторов, которые можно сохранять на сервере.**

Часто при выполнении рутинных операций требуется производить последовательность одинаковых запросов.

ХП позволяют объединить последовательность одинаковых запросов и сохранить их на сервере. После этого клиентам достаточно послать один запрос на выполнение ХП.

Хранимые процедуры представляют собой набор команд, состоящий из одного или нескольких операторов SQL или функций и сохраняемый в базе данных в откомпилированном виде.

# Определение

**Хранимая процедура** — объект базы данных, представляющий собой набор SQL-инструкций, который компилируется один раз и хранится на сервере.

# Назначение ХП:

1. Многочисленные клиентские приложения написаны на разных языках или работают на различных платформах, но должны выполнять одинаковые операции с базами данных
2. Пользователи не получают непосредственный доступ к таблицам базы данных и могут выполнять только конкретные хранимые процедуры

3. Упрощение сложных операций за счет инкапсуляции процессов в один блок, простой для выполнения
4. Обеспечение непротиворечивости данных
5. Предотвращение ошибок
6. СУБД тратит меньше времени на обработку команд ХП
7. Язык хранимых процедур



# Преимущества ХП

1. Повторное использование кода
2. Повышается производительность БД
3. Создание библиотек ХП и функций
4. Сокращение сетевого трафика
5. Безопасность
6. Реализация Бизнес Правил (проверка условий, например отклонение заказа в случае недостаточности количества товара)

# Недостатки ХП

1. Использование ХП увеличивает нагрузку на сервер БД
2. Хранимые процедуры сложнее в написании и их подготовка требует большей квалификации и опыта

# Виды ХП

ХП делятся на:

1. Без параметров
2. С параметрами(входными, выходными)

Также ХП делятся на:

1. Процедуры выбора(Возвращающие единственную запись и Несколько записей)
2. Исполняемые (Insert, Delete, Update-процедуры)

Для создания ХП существует два  
оператора:

CREATE PROCEDURE и CREATE FUNCTION

# Синтаксис создания хранимой процедуры

**CREATE PROCEDURE**

**имя\_хранимой\_процедуры ([параметр[, ...]])**

**[характеристика ...] тело\_процедуры**

*параметр:*

**[ IN | OUT | INOUT ] имя\_параметра тип**

**тип: любой допустимый тип данных MySQL**

IN (по умолчанию) – данные передаются строго внутрь ХП, но если такому параметру внутри ХП присваивается новое значение, оно не сохраняется по выходу из нее.

OUT – данные передаются строго из ХП, даже если параметр имеет начальное значение, внутри ХП это значение не принимается во внимание.

INOUT – значение этого параметра как принимается во внимание внутри ХП, так и сохраняет свое значение по выходу из нее.

Скобки обязательны.

## *Характеристика:*

LANGUAGE SQL

| [NOT] DETERMINISTIC

| SQL SECURITY {DEFINER | INVOKER}

| COMMENT 'строка'

LANGUAGE SQL – пока не имеет особого смысла и означает, что ХП написана на языке SQL.

**Deterministic** – процедура (Функция) считается "детерминированной", если она всегда возвращает один и тот же результат для одних и тех же входных аргументов, в противном случае функция является "недетерминированной".

**SQL security** – параметры безопасности процедуры

**Definer** – процедуру будет использовать создатель

**Invoker** – процедуру будет использовать пользователь, вызывающий процедуру оператором CALL

**Comment** – описание ХП, которое отображается операторами SHOW CREATE PROCEDURE и SHOW CREATE FUNCTION



## Пример 1.

```
CREATE PROCEDURE PROC_1()
```

```
BEGIN
```

```
Insert into depart values (1,'экономический');
```

```
Update emp set depno=1;
```

```
END;
```

## Пример 2.

```
CREATE PROCEDURE proc4  
(IN ilnput1 INT, IN ilnput2 INT)  
BEGIN  
SELECT * FROM emp  
WHERE salary between ilnput1 AND ilnput2;  
END;
```

# Вызов ХП

**Процедура вызывается с помощью оператора CALL и может только передавать значения обратно, используя выходные переменные.**

Функции могут возвращать скалярное значение и вызываются из оператора точно так же, как и любые другие функции (то есть, через указание имени функции).

**Хранимые процедуры могут вызывать другие хранимые процедуры.**

**CALL имя\_хранимой\_процедуры(  
[параметр[ , ...]] )**

**Пример:**

CALL simpleproc(@a);

Пробел после имени ХП недопустим.

# Функции

```
CREATE FUNCTION имя_функции ([параметр[, ...]])  
RETURNS тип  
[характеристика ...] тело_процедуры
```

Конструкция RETURNS может быть определена только для FUNCTION. Она используется для указания типа результата функции, при этом в теле функции должен присутствовать оператор RETURN значение.

**На заметку!**

**В настоящее время хранимые функции,  
создаваемые с помощью CREATE FUNCTION,  
не могут содержать ссылки на таблицы.**

## Пример.

```
CREATE FUNCTION hello  
(s CHAR(20)) RETURNS CHAR(50)  
RETURN CONCAT('Hello, ',s, '!') ;
```

## **Вызов функции:**

```
SELECT имя_функции ( [параметр[ , ...]] )
```

```
SELECT hello('world');
```

## **Результат:**

```
Hello, world!
```



# Переменные в MySQL

## Простые переменные

```
DECLARE iVar INT DEFAULT 0;
```

```
SET iVar = 5;
```

```
SELECT * FROM data WHERE id = iVar;
```

```
DECLARE iVar INT DEFAULT 0;
```

```
SELECT COUNT(*) INTO iVar FROM data;
```

## Системные переменные

```
SET @iVar = 5;
```

```
SELECT @iVar;
```

# Изменение ХП и функций

ALTER {PROCEDURE | FUNCTION}

имя\_хранимой\_процедуры [характеристика  
...]

характеристика:

NAME новое имя

| SQL SECURITY {DEFINER | INVOKER}

| COMMENT 'строка'

Данный оператор можно использовать для переименования хранимой процедуры или функции, а также для изменения ее характеристик. В операторе ALTER PROCEDURE или

ALTER FUNCTION разрешается указывать и более одной замены.

# Удаление ХП и функций

```
DROP {PROCEDURE | FUNCTION} [IF EXISTS]
```

```
имя_храняемой_процедуры
```

# Операторы управления потоком(язык ХП)

## Оператор IF

IF (условие\_поиска) THEN оператор(ы)

[ELSEIF условие\_поиска THEN оператор (ы)]

...

[ELSE оператор (ы)]

END IF

# Задание для практической работы

1. Проверить работу всех ХП, записанных на лекции.

На оценку:

1. Создать ХП (выбора) без параметров «Сотрудники отделов, з/п которых выше средней по каждому отделу».
2. Создать ХП (выбора) с параметрами, которая возвращает макс, мин и среднее значение заработной платы сотрудников указанного отдела.
3. Создать исполняемую ХП, возвращающую количество сотрудников, работающих на предприятии.
4. Создать исполняемую ХП с параметрами, которая увеличивает з/п сотрудников на 13% на каждого ребенка.
5. Создать ХП с условием, в зависимости от которого, ХП возвращает все записи одной из трех таблиц.