

Обеспечение целостности данных

Процедурное



Хранимые процедуры

- Хранимые процедуры пишутся на специальном встроенном языке программирования, они могут включать любые операторы SQL, а также включают некоторый набор операторов, управляющих ходом выполнения программ



- Хранимые процедуры являются объектами БД.
- Каждая хранимая процедура компилируется при первом выполнении
- В процессе компиляции строится оптимальный план выполнения процедуры.
- Описание процедуры совместно с планом ее выполнения хранится в системных таблицах БД.



Синтаксис

- CREATE [OR REPLACE]
- (“аргумент” IN | OUT | IN OUT “Тип данных” [,..])
- IS | AS
- “Тело процедуры PL/SQL”



Функция получения ip-адреса

- **create or replace function**
sys.client_ip_address
return varchar2 is
begin
return dbms_standard.client_ip_address;
end;



Пример процедуры

- create or replace procedure
sys.subptxt(name varchar2, subname
varchar2, usr varchar2,
txt in out varchar2) is
- begin
- subptxt2(name, subname, usr, null, null,
txt);
- end;



Триггеры

- Фактически триггер — это специальный вид хранимой процедуры, которую SQL Server вызывает при выполнении операций модификации соответствующих таблиц.
- Триггер автоматически активизируется при выполнении операции, с которой он связан.
- Триггеры связываются с одной или несколькими операциями модификации над одной таблицей.



- В СУБД Oracle определены два типа триггеров:
- триггеры, которые могут быть запущены перед реализацией операции модификации, они называются BEFORE-триггерами,
- и триггеры, которые активизируются после выполнения соответствующей модификации, аналогично триггерам MS SQL Server, — они называются AFTER-триггерами.



Синтаксис (стандарт)

- CREATE TRIGGER <имя_триггера>
- ON <имя_таблицы>
- FOR { [INSERT] [,UPDATE] [, DELETE] }
- AS
- SQL-операторы (Тело триггера)



В Oracle

- CREATE [OR REPLACE] TRIGGER
<имя_триггера> BEFORE | AFTER
- ON <имя_таблицы>
- FOR { [INSERT] [,UPDATE] [, DELETE] }
- FOR EACH ROW
- WHEN (условие)
- AS
- SQL-операторы (Тело триггера)



Пример

- **create or replace trigger** add_author **AFTER**
INSERT OR UPDATE OF C_AUTHOR
- **ON T CLAUSES**
FOR EACH ROW
- **DECLARE**
id_cl int;
aut varchar2(500);
res int;
- **BEGIN**
id_cl := :new.N_ID_CL;
aut := :new.C_AUTHOR;
res := ANALIZ_AUT(id_cl, aut);
END;

Ограничения

- Нельзя использовать в теле триггера операции создания объектов БД (новой БД, новой таблицы, нового индекса, новой хранимой процедуры, нового триггера, новых индексов, новых представлений),
- Нельзя использовать в триггере команду удаления объектов DROP для всех типов базовых объектов БД.
- Нельзя использовать в теле триггера команды изменения базовых объектов ALTER TABLE, ALTER DATABASE.
- Нельзя изменять права доступа к объектам БД, то есть выполнять команду GRANT или REVOKE.
- Нельзя создать триггер для представления (VIEW).
- В отличие от хранимых процедур, триггер не может возвращать никаких значений, он запускается автоматически сервером и не может связаться самостоятельно ни с одним клиентом.

Пример приложения

- Программа отправляет и отслеживает SMS-сообщения (через e-mail шлюз) с сайта оператора сотовой связи.



Информационные потоки

- ТЕКСТ СООБЩЕНИЯ набирает пользователь в форме.
- Сообщение проверяется на наличие «ключевых» слов.
- При наличии этих слов пользователь отправлявший сообщение, попадает в «черный список»
- Периодически всем пользователям «черного списка» рассылается автоматически сообщение-предупреждение.



Сущности

- Пользователи (E-Mail, Telefon)
- Сообщения
- (E-Mail,orig_mobile,dest_mobile,message)
- Черный список (E-Mail, Telefon, count_err)
- Список «ключевых слов»

