Темы

- Получение данных в PL/SQL
- Работа с данными в PL/SQL
- Введение в курсоры SQL

SQL выражения в PL/SQL

Использование команд SQL в блоках PL/SQL позволяет:

- Получить данные из СУБД при помощи команды `SELECT`.
- Внести изменения в СУБД, используя команды DML.
- Управлять транзакциями про помощи команд TCL `COMMIT`, `ROLLBACK`, `SAVEPOINT`.

Выражение SELECTв PL/SQL

Для получения данных из СУБД используется команда `SELECT`.

Синтаксис:

```
SELECT   select_list
INTO     {variable_name[, variable_name]...
         | record_name}
FROM     table
[WHERE   condition];
```

Data Science Academy

Выражение SELECTв PL/SQL

- Обязательно использовать `INTO`.
- Запрос должен возвращать одну запись

```
DECLARE
 v_fname VARCHAR2(25);
BEGIN
 SELECT first_name INTO v_fname
 FROM employees WHERE employee_id=200;
 DBMS_OUTPUT.PUT_LINE(' First Name is : '||v_fname);
END;
/
```

```
anonymous block completed
 First Name is : Jennifer
```

Data Science Academy

Выражение SELECTв PL/SQL: пример

Получить дату приема на работу (`hire_date`) и заработную плату указанного работника (`salary`).

```
DECLARE
 v_emp_hiredate   employees.hire_date%TYPE;
 v_emp_salary     employees.salary%TYPE;
BEGIN
  SELECT   hire_date, salary
  INTO     v_emp_hiredate, v_emp_salary
  FROM     employees
  WHERE    employee_id = 100;
  DBMS_OUTPUT.PUT_LINE ('Hire date is :'|| v_emp_hiredate);
  DBMS_OUTPUT.PUT_LINE ('Salary is :'|| v_emp_salary);
END;
/
```

```
anonymous block completed
Hire date is :17-JUN-03
Salary is :24000
```

Data Science Academy

Получение данных в PL/SQL

Получить суммы заработных плат всех работников указанного департамента.

Пример:

```
DECLARE
   v_sum_sal    NUMBER(10,2);
   v_deptno     NUMBER NOT NULL := 60;
BEGIN
  SELECT  SUM(salary)  -- group function
  INTO v_sum_sal  FROM employees
  WHERE     department_id = v_deptno;
  DBMS_OUTPUT.PUT_LINE ('The sum of salary is ' || v_sum_sal);
END;
```

```
anonymous block completed
The sum of salary is 28800
```

Data Science Academy

Неоднозначность названий

```
DECLARE
  hire_date       employees.hire_date%TYPE;
  sysdate         hire_date%TYPE;
  employee_id     employees.employee_id%TYPE := 176;
BEGIN
  SELECT      hire_date, sysdate
  INTO        hire_date, sysdate
  FROM        employees
  WHERE       employee_id = employee_id;
END;
/
```

```
Error report:
ORA-01422: exact fetch returns more than requested number of rows
ORA-06512: at line 6
01422. 00000 -  "exact fetch returns more than requested number of rows"
*Cause:    The number specified in exact fetch is less than the rows returned.
*Action:   Rewrite the query or change number of rows requested
```

Data Science Academy

Правила наименования

- Используйте правила наименования чтобы избежать неоднозначности в условии `WHERE`.
- Избегайте называть переменные также, как и столбцы в таблице.
- В PL/SQL Названия столбцов имеют больший приоритет.
- Имена локальных переменных и параметров имеют приоритет перед названиями таблиц.
- Названия переменных имеют более высокий приоритет перед названиями функций.

Темы

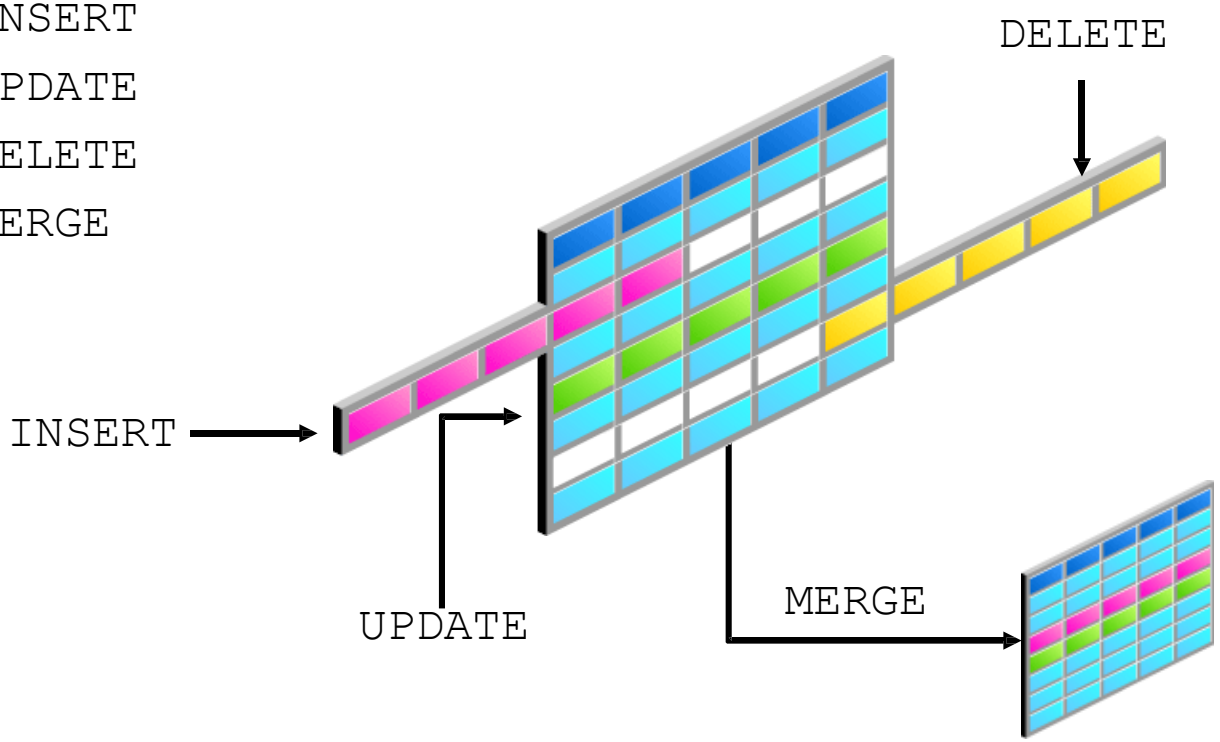
- Получение данных в PL/SQL
- Работа с данными в PL/SQL
- Введение в курсоры SQL

Data Science Academy

Использование PL/SQL для манипуляции данными

Для внесения изменений в базу используются DML команды:

- INSERT
- UPDATE
- DELETE
- MERGE



Добавление данных

Добавить нового сотрудника в таблицу EMPLOYEES.

```
BEGIN
 INSERT INTO employees
  (employee_id, first_name, last_name, email,
   hire_date, job_id, salary)
   VALUES(employees_seq.NEXTVAL, 'Ruth', 'Cores',
   'RCORES',CURRENT_DATE, 'AD_ASST', 4000);
END;
/
```

Обновление данных

Поднять заработную плату всем складским работникам.

```
DECLARE
  sal_increase   employees.salary%TYPE := 800;
BEGIN
  UPDATE      employees
  SET         salary = salary + sal_increase
  WHERE       job_id = 'ST_CLERK';
END;
/
```

```
anonymous block completed
FIRST_NAME           SALARY
-------------------- ----------------------
Julia                4000
Irene                3500
James                3200
Steven               3000
```

...

```
Curtis               3900
Randall              3400
Peter                3300

20 rows selected
```

Удаление данных: пример

Удалить из таблицы `employees` всех сотрудников, которые работают в департаменте с номером 10.

```
DECLARE
  deptno   employees.department_id%TYPE := 10;
BEGIN
  DELETE FROM   employees
  WHERE  department_id = deptno;
END;
/
```

Data Science Academy

Объединение записей

Вставить или обновить все записи в таблице `copy_emp` чтобы привести в соответствие с таблицей `employees`.

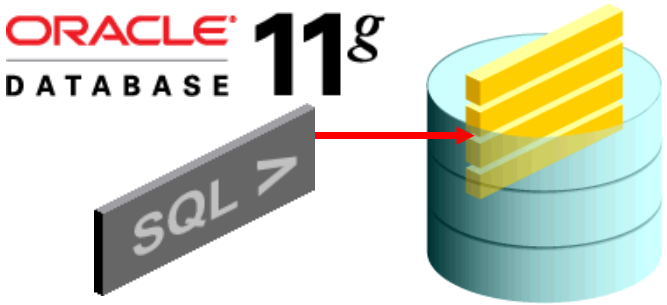
```
BEGIN
MERGE INTO copy_emp c
     USING employees e
     ON (e.employee_id = c.empno)
   WHEN MATCHED THEN
     UPDATE SET
       c.first_name     = e.first_name,
       c.last_name      = e.last_name,
       c.email          = e.email,
       . . .
   WHEN NOT MATCHED THEN
     INSERT VALUES(e.employee_id, e.first_name, e.last_name,
          . . .,e.department_id);
END;
/
```

Темы

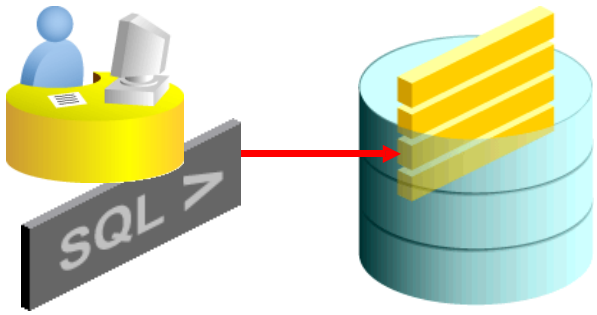
- Получение данных в PL/SQL
- Работа с данными в PL/SQL
- Введение в курсоры SQL

SQL Cursor

- Курсор (cursor) – это указатель на область в памяти, выделенной под обработку пользовательского запроса в Oracle Server. Используется для выполнения команды `SELECT`.
- Существует два типа курсоров: явные и неявные.
  - **Неявный:** Создается сервером СУБД для выполнение SQL запросов и управляется сервером
  - **Явный :** Объявляется и управляется разработчиком



Неявный курсор



Явный курсор

Атрибуты неявных курсоров SQL

Используя атрибуты неявного курсора можно проверить результат выполнения SQL выражений

SQL%FOUND	Boolean значение, `TRUE` означает, что последний SQL запрос обработал хотя бы одну запись
SQL%NOTFOUND	Boolean значение, `TRUE` означает, что последний SQL запрос не обработал ни одной записи
SQL%ROWCOUNT	Целое число, которое соответствует количеству записей, которое обработал последний запрос

Атрибуты неявных курсоров SQL

Пример: удалить записи из таблица `employees`. Вывести информацию о количестве удаленных записей.

```
DECLARE
  v_rows_deleted VARCHAR2(30);
  v_empno employees.employee_id%TYPE := 176;
BEGIN
  DELETE FROM  employees
  WHERE employee_id = v_empno;
  v_rows_deleted := (SQL%ROWCOUNT ||
                        ' row deleted.');
  DBMS_OUTPUT.PUT_LINE (v_rows_deleted);

END;
```