федеральное государственное автономное образовательное учреждение высшего образования

Санкт-Петербургский национальный исследовательский УНИВЕРСИТЕТ информационных технологий, механики и оптики

**Факультет программной инженерии и компьютерной техники**

**ЛАБОРАТОРНАЯ РАБОТА № 1**

**ПО ДИСЦИПЛИНЕ «СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫМИ»**

**Вариант 5212**

Выполнил: Чудаков М.И.

Группа: P3419

Преподаватели: Николаев В.В.

Санкт-Петербург 2019

**Задание**

Используя сведения из представлений словаря данных получить информацию о любой таблице: Номер по порядку, Имя столбца, Атрибуты (в атрибуты столбца включить тип данных, комментарий, ограничение целостности типа NOT NULL).

**Исходный код**

**set** verify off;  
**set** serveroutput **ON**;  
  
**DECLARE** inputString VARCHAR2(128) := **'&tableName'**;  
 *--inputString VARCHAR2(128) := 's225141.point';* tableName VARCHAR2(128) := **''**;  
 schemaName VARCHAR2(128) := **null**;  
 inputStringLength **NUMBER** := 0;  
 tableNameLength **NUMBER** := 0;  
 schemaNameLength **NUMBER** := 0;  
 pointPosition **NUMBER** := 0;  
 noLength **NUMBER** := 3;  
 columnNameLength **NUMBER** := 15;  
 attributesLength **NUMBER** := 60;  
 attributeNameLength **NUMBER** := 15;  
 tableExists **NUMBER** := 0;  
 tableReached **NUMBER** := 0;  
 isCorrectTableName BOOLEAN := **TRUE**;  
 isCorrectSchemaName BOOLEAN := **TRUE**;  
 tableAndSchemaNameRegex VARCHAR2(128) := **'^[A-Za-zА-Яа-я][A-Za-zА-Яа-я0-9\_$#.]\*$'**;  
 columnId ALL\_TAB\_COLUMNS.COLUMN\_ID%**TYPE**;  
 columnName ALL\_TAB\_COLUMNS.COLUMN\_NAME%**TYPE**;  
 columnDataType ALL\_TAB\_COLUMNS.DATA\_TYPE%**TYPE**;  
 columnComment ALL\_COL\_COMMENTS.COMMENTS%**TYPE**;  
 allColumnsCursor SYS\_REFCURSOR;  
 **CURSOR** notNullConstraints **IS  
 SELECT DISTINCT COLUMN\_NAME  
 FROM** ALL\_CONSTRAINTS  
 **INNER JOIN** ALL\_CONS\_COLUMNS  
 **ON** ALL\_CONSTRAINTS.CONSTRAINT\_NAME = ALL\_CONS\_COLUMNS.CONSTRAINT\_NAME  
 **WHERE** ALL\_CONS\_COLUMNS.TABLE\_NAME = tableName  
 **AND** ALL\_CONSTRAINTS.TABLE\_NAME = tableName  
 **AND** ALL\_CONSTRAINTS.CONSTRAINT\_TYPE = **'C'  
 AND** ALL\_CONSTRAINTS.SEARCH\_CONDITION **IS NOT NULL**;  
  
 **PROCEDURE** getColumnByTable(allColumns OUT SYS\_REFCURSOR) **IS  
 BEGIN  
 OPEN** allColumns **FOR  
 SELECT** ALL\_TAB\_COLUMNS.COLUMN\_ID,  
 ALL\_TAB\_COLUMNS.COLUMN\_NAME,  
 ALL\_TAB\_COLUMNS.DATA\_TYPE,  
 ALL\_COL\_COMMENTS.COMMENTS  
 **FROM** ALL\_TAB\_COLUMNS  
 **INNER JOIN** ALL\_COL\_COMMENTS  
 **ON** ALL\_TAB\_COLUMNS.COLUMN\_NAME = ALL\_COL\_COMMENTS.COLUMN\_NAME  
 **WHERE** ALL\_TAB\_COLUMNS.TABLE\_NAME = tableName  
 **AND** ALL\_COL\_COMMENTS.TABLE\_NAME = tableName;  
 **END**;  
  
 **PROCEDURE** getColumnByTableAndSchema(allColumns OUT SYS\_REFCURSOR) **IS  
 BEGIN  
 OPEN** allColumns **FOR  
 SELECT** ALL\_TAB\_COLUMNS.COLUMN\_ID,  
 ALL\_TAB\_COLUMNS.COLUMN\_NAME,  
 ALL\_TAB\_COLUMNS.DATA\_TYPE,  
 ALL\_COL\_COMMENTS.COMMENTS  
 **FROM** ALL\_TAB\_COLUMNS  
 **INNER JOIN** ALL\_COL\_COMMENTS  
 **ON** ALL\_TAB\_COLUMNS.COLUMN\_NAME = ALL\_COL\_COMMENTS.COLUMN\_NAME  
 **WHERE** ALL\_TAB\_COLUMNS.TABLE\_NAME = tableName  
 **AND** ALL\_COL\_COMMENTS.TABLE\_NAME = tableName  
 **AND** ALL\_TAB\_COLUMNS.OWNER = schemaName;  
 **END**;  
  
**BEGIN** pointPosition := INSTR(inputString, **'.'**);  
  
 IF pointPosition != 0 **THEN** tableName := SUBSTR(inputString, pointPosition + 1);  
  
 inputStringLength := **LENGTH**(inputString);  
 tableNameLength := **LENGTH**(tableName);  
 schemaNameLength := inputStringLength - tableNameLength;  
  
 schemaName := SUBSTR(inputString, 1, schemaNameLength - 1);  
 isCorrectSchemaName := REGEXP\_LIKE(schemaName, tableAndSchemaNameRegex);  
  
 **SELECT TABLE\_NAME**, OWNER  
 **INTO** tableName, schemaName  
 **FROM** ALL\_TAB\_COLUMNS  
 **WHERE** *UPPER*(**TABLE\_NAME**) = *UPPER*(tableName)  
 **AND** *UPPER*(OWNER) = *UPPER*(schemaName)  
 **AND** ROWNUM = 1;  
  
 **SELECT** *COUNT*(*\**)  
 **INTO** tableExists  
 **FROM** dba\_tables  
 **WHERE TABLE\_NAME** = tableName  
 **AND** OWNER = schemaName;  
  
 **SELECT** *COUNT*(*\**)  
 **INTO** tableReached  
 **FROM** all\_tables  
 **WHERE TABLE\_NAME** = tableName  
 **AND** OWNER = schemaName;  
 **ELSE  
 SELECT TABLE\_NAME  
 INTO** inputString  
 **FROM** ALL\_TAB\_COLUMNS  
 **WHERE** *UPPER*(**TABLE\_NAME**) = *UPPER*(inputString)  
 **AND** ROWNUM = 1;  
  
 tableName := inputString;  
  
 **SELECT** *COUNT*(*\**)  
 **INTO** tableExists  
 **FROM** dba\_tables  
 **WHERE TABLE\_NAME** = tableName;  
  
 **SELECT** *COUNT*(*\**)  
 **INTO** tableReached  
 **FROM** all\_tables  
 **WHERE TABLE\_NAME** = tableName;  
 **end** if;  
  
 isCorrectTableName := REGEXP\_LIKE(tableName, tableAndSchemaNameRegex);  
  
 IF **NOT** isCorrectTableName **AND NOT** isCorrectSchemaName **THEN** DBMS\_OUTPUT.PUT\_LINE(**'Имя таблицы и имя схемы не корректны'**);  
 ELSIF **NOT** isCorrectTableName **THEN** DBMS\_OUTPUT.PUT\_LINE(**'Имя таблицы не корректно'**);  
 ELSIF **NOT** isCorrectSchemaName **THEN** DBMS\_OUTPUT.PUT\_LINE(**'Имя схемы не корректно'**);  
 ELSIF tableExists = 0 **THEN** DBMS\_OUTPUT.PUT\_LINE(**'Таблица не существует '** || inputString);  
 ELSIF tableReached = 0 **THEN** DBMS\_OUTPUT.PUT\_LINE(**'У вас нет доступа к таблице '** || inputString);  
 **ELSE** IF schemaName **IS NULL THEN  
 SELECT TABLE\_NAME  
 INTO** inputString  
 **FROM** ALL\_TAB\_COLUMNS  
 **WHERE TABLE\_NAME** = tableName  
 **AND** ROWNUM = 1;  
  
 getColumnByTable(allColumnsCursor);  
 **ELSE  
 SELECT TABLE\_NAME**, OWNER  
 **INTO** tableName, schemaName  
 **FROM** ALL\_TAB\_COLUMNS  
 **WHERE TABLE\_NAME** = tableName  
 **AND** OWNER = schemaName  
 **AND** ROWNUM = 1;  
  
 inputString := UTL\_LMS.FORMAT\_MESSAGE(**'%s.%s'**, schemaName, tableName);  
  
 getColumnByTableAndSchema(allColumnsCursor);  
 **end** if;  
  
 DBMS\_OUTPUT.PUT\_LINE(**'Таблица: '** || inputString);  
 DBMS\_OUTPUT.PUT\_LINE(**''**);  
 DBMS\_OUTPUT.PUT\_LINE(RPAD(**'No.'**, noLength) || **' '** ||  
 RPAD(**'Имя столбца'**, columnNameLength) || **' '** ||  
 RPAD(**'Атрибуты'**, attributesLength));  
 DBMS\_OUTPUT.PUT\_LINE(RPAD(**'-'**, noLength, **'-'**) || **' '** ||  
 RPAD(**'-'**, columnNameLength, **'-'**) || **' '** ||  
 RPAD(**'-'**, attributesLength, **'-'**));  
  
 LOOP  
 **FETCH** allColumnsCursor  
 **INTO** columnId, columnName, columnDataType, columnComment;  
 EXIT **WHEN** allColumnsCursor%NOTFOUND;  
  
 DBMS\_OUTPUT.PUT\_LINE(RPAD(columnId, noLength) || **' '** ||  
 RPAD(columnName, columnNameLength) || **' '** ||  
 RPAD(**'Type:'**, attributeNameLength) ||  
 RPAD(columnDataType, attributesLength - attributeNameLength));  
 IF columnComment **IS NOT NULL THEN** DBMS\_OUTPUT.PUT\_LINE(RPAD(**' '**, noLength + columnNameLength + 2) ||  
 RPAD(**'COMMEN:'**, attributeNameLength) ||  
 RPAD(columnComment, attributesLength - attributeNameLength));  
 **end** if;  
  
 **FOR constraint IN** notNullConstraints  
 LOOP  
 IF (**constraint**.COLUMN\_NAME = columnName) **THEN** DBMS\_OUTPUT.PUT\_LINE(RPAD(**' '**, noLength + columnNameLength + 2) ||  
 RPAD(**'Constraint:'**, attributeNameLength) ||  
 RPAD(**'Not null'**, attributesLength - attributeNameLength));  
 **end** if;  
 **end** loop;  
 **end** loop;  
 **end** if;  
**END** ;  
/

**Результат работы**

Таблица: Н\_УЧЕНИКИ

No. Имя столбца Атрибуты

--- --------------- ------------------------------------------------------------

1 ИД Type: NUMBER

COMMEN: Уникальный идентификатор

Constraint: Not null

2 ЧЛВК\_ИД Type: NUMBER

COMMEN: Внешний ключ к таблице Н\_ОБУЧЕНИЯ

Constraint: Not null

3 ПРИЗНАК Type: VARCHAR2

COMMEN: (обучен,отчисл,академ,диплом)

Constraint: Not null

4 СОСТОЯНИЕ Type: VARCHAR2

COMMEN: Состояние (проект, утвержден, отменен)

Constraint: Not null

5 НАЧАЛО Type: DATE

Constraint: Not null

6 КОНЕЦ Type: DATE

Constraint: Not null

7 ПЛАН\_ИД Type: NUMBER

COMMEN: Искусственный первичный уникальный идентифика

Constraint: Not null

8 ГРУППА Type: VARCHAR2

COMMEN: Номер студенческой группы

Constraint: Not null

9 П\_ПРКОК\_ИД Type: NUMBER

COMMEN: Внешний ключ к таблице Н\_ПУНКТЫ\_ПРИКАЗОВ\_ОК

Constraint: Not null

10 ВИД\_ОБУЧ\_ИД Type: NUMBER

COMMEN: Внешний ключ к таблице Н\_ОБУЧЕНИЯ

Constraint: Not null

11 ПРИМЕЧАНИЕ Type: VARCHAR2

COMMEN: Текст примечания

12 КТО\_СОЗДАЛ Type: VARCHAR2

Constraint: Not null

13 КОГДА\_СОЗДАЛ Type: DATE

Constraint: Not null

14 КТО\_ИЗМЕНИЛ Type: VARCHAR2

Constraint: Not null

15 КОГДА\_ИЗМЕНИЛ Type: DATE

Constraint: Not null

16 КОНЕЦ\_ПО\_ПРИКАЗ Type: DATE

COMMEN: конец периода действия строки по приказу

17 ВМЕСТО Type: NUMBER

COMMEN: ссылка на строку, вместо которой введена теку

18 В\_СВЯЗИ\_С Type: NUMBER

COMMEN: ид начисления, вызвавшего изменение текущего

19 ТЕКСТ Type: VARCHAR2

**Вывод**

В ходе выполнения лабораторной работы мною был написан скрипт, который используя сведения из представлений словаря данных выводит информацию о любой таблице по ее имени. В результате мною были изучены новые языковые конструкции и приобретен опыт извлечения информации о структуре БД и объектов внутри нее. Трудности возникли в ходе оптимизации скрипта, обработке ошибок ввода и добавлении удобств пользования скриптом.