# РАБОТА С КОЛЛЕКЦИЯМИ В БАЗЕ ДАННЫХ ORACLE

WORK WITH COLLECTIONS IN ORACLE DATABASE

**УДК 004.023**

**Журавлев Александр Александрович**, магистрант, Уральский Федеральный Университет им. Б. Н. Ельцина, ИРИТ-РтФ, г. Екатеринбург

**Zhuravlev A.A.** SanyaProgrammer2503@gmail.com

**Аннотация**

Роль баз данных для разработчиков приложений огромна. Они позволяют хранить и анализировать различную информацию. Основными задачами базы данных являются: обеспечение хранения необходимой информации, возможность получение конкретной информации, а также обеспечение целостности базы данных. Наиболее интересными случаями применения базы данных являются хранение и использование коллекций. В данной статье представлена работа с коллекциями в базе данных Oracle. В статье использован эмпирический метод исследования, поскольку в качестве основного источника результата взят анализ над данными (коллекциями). В ходе написания статьи получены следующие результаты:

- создана таблица Authors_Experience, на основе таблицы Authors;
- проведена работа с базой данных в соответствии с заданием в разделе «Описание базы данных»;
- создан PL/SQL блок для обработки базы данных;
- сделаны соответствующие выводы по проделанной работе.

**Annotation**

The role of databases for application developers is huge. They allow to store and analyze different information. The main objectives of the database are: ensuring the storage of necessary information, the ability to obtain specific information, as well as ensuring the integrity of the database. The most interesting cases of using the database are the storage and use of collections. This article presents working with collections in an Oracle database. The empirical method of research is used in t he article, since the analysis of data (collections) is taken as the main source of the result. During the writing of the article, the following results were obtained:

- created table Authors_Experience, based on table Authors;
- work with the database was carried out in accordance with the task in the «Description of the database» section;
- created PL/SQL block for database processing;
- relevant conclusion were drawn on t he work done.

**Ключевые слова:** коллекция, база данных, Oracle

**Keywords:** collection, database, Oracle

## Введение

Роль баз данных для разработчиков приложений огромна. Они позволяют хранить и анализировать различную информацию [1].

Наиболее интересными случаями применения базы данных являются хранение и использование коллекций. В данной статье проведена работа с таблицей Authors_Experience, которая создана на основе таблицы Authors. Над данной таблицей выполнены следующие операции:

- вставка строк в таблицу;
- добавление новой строки;
- удаление строки из таблицы;
- обработка таблицы с помощью блока PL/SQL;
- и т. д.

**Цель данной статьи** – создать таблицу Authors_Experience, основанную на таблице Authors, провести работу с созданной БД в соответствии с заданием, представленном в разделе «Описание базы данных».

В качестве **материала исследования** выступает база данных Oracle.

В статье используется **эмпирический метод исследования**, поскольку основной источник результатов – анализ.

## Описание базы данных

*Описание таблицы Authors*

Поля таблицы Authors:

- au_id – уникальный идентификатор (первичный ключ) записи об авторе, представленный в формате номера полиса социального страхования. Формат поля имеет вид: XXX-XX-XXXX, где X десятичная цифра от 0 до 9, - – знак «минус» в номере полиса. Поле является первичным ключом таблицы, имеет тип VARCHAR2(11) и должно соответствовать (проверяться сервером) заданному формату;
- au_lname – фамилия автора, тип VARCHAR2(40). Не допускает отсутствия значения (NOT NULL);
- au_fname – имя автора, тип VARCHAR2(20). Не допускает отсутствия значения (NOT NULL);
- phone – номер телефона автора, тип CHAR(12), NOT NULL, имеет умалчиваемое значение ‘UNKNOWN’;
- address – адрес, тип VARCHAR2(20), может быть не задан (NULL);
- state – штат, в котором проживает автор. Задается двухсимвольной аббревиатурой (например, CA), может быть не задан (NULL);
- zip – почтовый индекс из 5 десятичных цифр. Формат поля имеет вид: XXXXX, где X десятичная цифра от 0 до 9. В БД должно контролироваться соответствие данных заданному формату. Допускает отсутствие значения данного в поле;

- contract – числовое (NUMBER) поле, означающее наличие или отсутствие контракта с автором. Не допускает отсутствия значения (NOT NULL).

*Задание*

1. Создать в БД средствами SQL таблицу Authors_Experience из таблицы (или скрипта таблицы) Authors. Добавить в таблицу Authors_Experience столбец, содержащий коллекцию (вложенную таблицу) мест работы (название организации и количество лет) автора. Элемент коллекции имеет тип Varchar2(120).
2. Создание и обработка коллекции в SQL:

- вставка трех строк в главную таблицу;
- вывод главной таблицы с коллекцией, представленной списком элементов в поле;
- вывод главной таблицы с коллекцией, представленной столбцом (используя функцию TABLE());
- добавление в коллекцию, находящуюся во второй строке нового элемента (используя функцию TABLE());
- удаление второго элемента из коллекции, находящейся в первой строке;
- проверка добавления и удаления элементов путем вывода главной таблицы с коллекцией, представленной столбцом;
- вывод строк главной таблицы, содержащих в своих коллекциях элемент, заданный в условии запроса.

3. Создание блока PL\SQL, в котором:

- коллекция из второй строки главной таблицы читается в переменную блока;
- в коллекцию, находящуюся в переменной добавляется новый элемент,
- проверяется содержимое коллекции, путем поэлементарного вывода процедурой dbms_output.put_line(...);

- из коллекции удаляется третий элемент;
- измененная коллекция, находящаяся в переменной, заменяет коллекцию во второй строке главной таблицы.

## Работа с коллекциями

*Создание таблицы Authors*

Код таблицы Authors представлен ниже.

```
--Создание таблицы Авторы
CREATE TABLE Authors (
      au_id VARCHAR2(11) NOT NULL,
      au_lname VARCHAR2(40) NOT NULL,
      au_fname VARCHAR2(20) NOT NULL,
      phone CHAR(12) DEFAULT 'UNKNOWN' NOT NULL,
      address VARCHAR2 (40) NULL,
      city VARCHAR2(20) NULL,
      state VARCHAR2(2) NULL,
      zip VARCHAR2(5) NULL,
      contract NUMBER(1) NOT NULL,
      CONSTRAINT pk_au_id PRIMARY KEY (au_id),
      CONSTRAINT cf_au_id CHECK(regexp_like(au_id, '[0-9]{3}-[0-9]{2}-
      [0-9]{4}')),
      CONSTRAINT cf_zip CHECK(regexp_like(zip, '[0-9]{5}'))
);
```

*Создание нового типа и добавление вложенной таблицы*

Для создания нового типа данных использовалась команда CREATE OT REPLACE TYPE. Для изменения структуры таблицы использовалась команда ALTER TABLE (в нашем случае, добавление вложенного таблицы) [2]. Код создания нового типа и добавление вложенной таблицы представлены ниже.

```
--Создание типа Место работы
CREATE OR REPLACE TYPE Working_place AS OBJECT(
name VARCHAR2(120), years_worked INT);

--Создание коллекции типа Место работы
CREATE OR REPLACE TYPE Working_place_collection IS TABLE of
Working_place;

--Копирование таблицы Authors
CREATE TABLE Authors_Experience AS SELECT * FROM Authors;
```

```
--Добавление вложенной таблицы
ALTER TABLE Authors_Experience ADD(
worked_at Working_place_collection)
nested table worked_at STORE AS worked_at;
```

*Вставка строк в таблицу*

Вставка строк в таблицу осуществлялось с помощью команды INSERT INTO [3]. Код вставки данных:

```
--Вставка трех строк в таблицу
INSERT INTO Authors_Experience(au_id, au_lname, au_fname, phone, contract, worked_at) VALUES( '111-11-1111', 'Перумов', 'Ник', '11111111111', 1,
      Working_place_collection(
      Working_place('Дом печати № 1', 1),
      Working_place ('Дом печати № 2', 2)
     )
);

INSERT INTO Authors_Experience(au_id, au_lname, au_fname, phone, contract, worked_at) VALUES(
'222-22-2222', 'Лукьяненко', 'Сергей', '22222222222', 1,
      Working_place_collection(
      Working_place('Дом печати № 1', 2),
      Working_place ('Дом печати № 2', 1)
     )
);
INSERT INTO Authors_Experience(au_id, au_lname, au_fname, phone, contract, worked_at) VALUES(
'333-33-3333", 'Журавлев', 'Александр', '33333333333', 1,
      Working_place_collection(
      Working_place('Дом печати № 1', 1),
     )
);
```

*Вывод таблицы*

Запрос главной таблицы с коллекцией, представленной списком элементов в поле [4]:

```
--Вывод таблицы с коллекцией, представленной списком элементов в поле
SELECT worked_at FROM Authors_Experience;
```

Результат запроса представлен на рисунке 1.

```
WORKED_AT
--------------------------------------------------------------------------------------------------------------

WORKING_PLACE_COLLECTION(WORKING_PLACE('Дом печати № 1', 1), WORKING_PLACE('Дом печати № 2', 2))
WORKING_PLACE_COLLECTION(WORKING_PLACE('Дом печати № 1', 2), WORKING_PLACE('Дом печати № 2', 1))
WORKING_PLACE_COLLECTION(WORKING_PLACE('Дом печати № 1', 1))
```

Рисунок 1 – Результат запроса, представленный списком

Запрос главной таблицы с коллекцией, представленной столбцом.

```
--Вывод таблицы с коллекцией, представленной столбцом
SELECT AE.au_id, WA.* FROM Authors Experience AE, TABLE(AE.worked_at)
WA;
```

Результат запроса представлен на рисунке 2.

```
AU_ID       NAME
----------- ------------------
111-11-1111 Дом печати № 1
111-11-1111 Дом печати № 2
222-22-2222 Дом печати № 1
222-22-2222 Дом печати № 2
333-33-3333 Дом печати № 1
```

Рисунок 2 – Результат запроса, представленный столбцом

*Добавление элемента в таблицу*

Добавление элемента в коллекцию [5]:

```
--Добавление в коллекцию во вторую строку нового элемента
INSERT INTO TABLE (SELECT worked_at FROM Authors_Experience WHERE
au_id = '222-22-2222')
VALUES (
      Working_place('Дом печати № 3', 3)
);
```

Состояние таблицы после добавления в коллекцию элемента приведено на рисунке 3.

```
AU_ID       NAME                                                
```

*Удаление элемента из коллекции*

Удаление элемента из коллекции в соответствие с задание из раздела «Описание базы данных»:

```
--Удаление второго элемента из первой строки
DELETE FROM TABLE (SELECT worked_at FROM Authors Experience WHERE
au_id = '111-11-1111')
WHERE name = 'Дом печати № 2';
```

Состояние таблицы после удаления элемента из коллекции приведено на рисунке 4.

| AU_ID | NAME | YEARS_WORKED |
|---|---|---|
| 111-11-1111 | Дом печати № 1 | 1 |
| 222-22-2222 | Дом печати № 1 | 2 |
| 222-22-2222 | Дом печати № 2 | 1 |
| 222-22-2222 | Дом печати № 3 | 3 |
| 333-33-3333 | Дом печати № 1 | 1 |

Рисунок 4 – Состояние таблицы после удаления элемента из коллекции

Ниже представлен запрос, позволяющий вывести строки таблицы, содержащие в своих коллекциях элемент в условии запроса [6].

```
-- Вывод строк таблицы, содержащих в своих коллекциях элемент,
заданный в условии запроса
SELECT AE.au_fname, AE.au_lname FROM Authors_Experience AE, TABLE
(AE.worked_at) WA
WHERE WA.name = 'Дом печати № 1";
```

Результат вывода представлен на рисунке 5.

| AU_FNAME | AU_LNAME |
|---|---|
| Ник | Перумов |
| Сергей | Лукьяненко |
| Александр | Журавлев |

Рисунок 5 – Результат запроса

*Выполнение PL/SQL блока*

Ниже представлено состояние второй строки до выполнения PL\SQL блока (рисунок 6). Данный результат получен с помощью запроса:

```
SELECT WA.* FROM Authors_Experience AE, TABLE(AE.worked_at) WA
WHERE AE.au_id = '222-22-2222';
```

| NAME | YEARS_WORKED |
|---|---|
| Дом печати № 1 | 2 |
| Дом печати № 2 | 1 |
| Дом печати № 3 | 3 |

Рисунок 6– Состояние строки до выполнения PL\SQL блока

Далее представлен PL\SQL блок в соответствии с заданием, приведенным в разделе «Описание базы данных» [7].

```
DECLARE
      Working_places Working_place_collection :=
      Working_place_collection();
      CURSOR Authors_cursor IS
            SELECT WA.* FROM Authors_Experience AE, TABLE (AE.worked_at)
            WA
            WHERE AE.au_id = '222-22-2222';
BEGIN
      --Копирование существующих элементов
      FOR val IN Authors_cursor
      LOOP
            Working_places.extend();
            Working_places(Working_places.count):=Working_place(val.name
            , val.years_worked);
      END LOOP;

      --Добавление нового элемента
      Working_places.extend();
      Working_places(Working_places.count) := Working_place('Дом печати
      № 4', 4);

      --Проверка на наличие
      FOR ind in 1 .. Working_places.count()
      LOOP
            dbms_output.put_line(Working_places(ind).name);
      END LOOP;

      --Удаление 3-го элемента
      Working_places.delete(3);

      --Замена значения
      UPDATE Authors_Experience
      SET worked_at = Working_places
      WHERE au_id = '222-22-2222';
END;
```

Состояние строки после выполнения PL\SQL блока представлено на рисунке 7.

| NAME | YEARS_WORKED |
|---|---|
| Дом печати № 1 | 2 |
| Дом печати № 2 | 1 |
| Дом печати № 4 | 4 |
| Дом печати № 1 | 2 |

Рисунок 7 – Состояние второй строки после выполнения PL\SQL блока

## Заключение

В данной статье проведена работа с коллекциями в базе данных Oracle. В качестве объекта для работы представлена таблица Authors_Experience, которая создана на основе таблицы Authors.

*Результаты исследования*

Над таблицей Authors_Experience выполнены следующие операции:

- вставка строк в таблицу;
- добавление новой строки;
- удаление строки из таблицы;
- обработка таблицы с помощью блока PL/SQL;
- и пр.

## Литература

1. Медведев Ю. С., Пиотровский Д. Л. Мониторинг производительности базы данных Oracle // Вестник АГУ. 2018. № 4. С. 210-213.
2. Бьюли Алан. Изучаем SQL. Издательство: «Символ-плюс», 2016. 309 С.
3. Аллен Дж. Тейлор. SQL для чайников. Издательство: «Диалектика», 2014. 416 С.
4. Бен Форта. SQL за 10 минут. Издательство: «Вильямс», 2014. 288 С.
5. Рон Хардман, Майкл МакЛафлин. Oracle Database PL/SQL. Рекомендации эксперта. Издательство: «Лори», 2014. 450 С.
6. Линн Бейли. Изучаем SQL. Издательство: «Питер», 2012. 592 С.
7. Билл Карвин. Программирование баз данных SQL. Типичные ошибки и их устранение. Издательство: «Рид Групп», 2012. 336 С.

**Literature**

1. Medvedev U. S., Piotrovsky D. L. Monitoring the performance of the Oracle database // Vestnik ASU. 2018. No. 4.P. 210-213.
2. Byulie Alan. Learn SQL. Publisher: "Symbol-plus", 2016. 309 P.
3. Allen J. Taylor. SQL for dummies. Publisher: "Dialectics", 2014. 416 P.
4. Ben Fort. SQL in 10 minutes. Publisher: Williams, 2014. 288 P.
5. Ron Hardman, Michael McLaughlin. Oracle Database PL / SQL. Expert advice. Publisher: Lori, 2014. 450 P.
6. Lynn Bailey. Learn SQL. Publisher: Peter, 2012. 592 P.
7. Bill Carvin. SQL database programming. Typical errors and their elimination. Publisher: Reed Group, 2012. 336 P.