

УДК 004.6

## СОЗДАНИЕ И ИСПОЛЬЗОВАНИЕ ОБЪЕКТОВ В БАЗЕ ДАННЫХ ORACLE

**Журавлев А.А., Аксенов К.А.**

*Уральский федеральный университет им. Б.Н. Ельцина, ИРИТ РтФ, Екатеринбург,  
e-mail: SanyaProgrammer2503@gmail.com, k.a.aksenov@urfu.ru*

Базы данных играют огромную роль в работе разработчика приложений. Они необходимы для хранения и анализа различной информации. Отличительной чертой базы данных от других хранилищ типа архивов является то, что информация в ней обязательно подвергается обработке электронными носителями с последующей систематизацией, которая подчинена определенным правилам. Основными задачами базы данных являются: обеспечение хранения необходимой информации, возможность получения конкретной информации, а также обеспечение целостности базы данных. Наиболее интересными случаями применения базы данных являются хранение и использование объектов. В данной статье представлено создание и использование объектов в базе данных Oracle на примере геометрической фигуры (треугольника). В статье использован эмпирический метод исследования, поскольку в качестве основного источника результата взят анализ над данными (объектами). В ходе написания статьи получены следующие результаты: с помощью базы данных Oracle построена новая таблица с полем-объектом, для хранения треугольников; создана отдельная (вторая) таблица со строками-объектами и заполнена объектами из первой таблицы; создан столбец-ссылка на объекты дополнительной таблицы; сделаны соответствующие выводы о проделанной работе.

**Ключевые слова:** объекты, базы данных, создание, использование, Oracle

## CREATING AND USING OBJECTS IN ORACLE DATABASE

**Zhuravlev A.A., Aksenov K.A.**

*Ural Federal University named after B.N. Yeltsin, IRIT RTF, Yekaterinburg,  
e-mail: SanyaProgrammer2503@gmail.com, k.a.aksenov@urfu.ru*

Databases play a huge role in the work of the application developer. It is necessary for storing and analyzing different information. A distinguishing feature of the database from other repositories such as archives is that the information in it is necessarily processed by electronic media with subsequent systematization, which is subject to certain rules. 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 database application is storing objects. This article presents the creation and use of objects in an Oracle database using the example of a geometric figure (triangle). The article uses the empirical research method, since the analysis of data (objects) is taken as the main source of the result. During the writing of the article, the following results were obtained: using the Oracle database, a new table with an object field was build for storing triangles; a separate (second) table was created with rows-objects and filled with objects from the first table; created a column link to the objects of the additional table; appropriate conclusions were made about the work done.

**Keywords:** objects, database, creating, using, Oracle

Базы данных играют огромную роль в работе разработчика приложений. Они необходимы для хранения и анализа различной информации [1–3].

Наиболее интересными случаями применения базы данных являются хранение и использование объектов. В данной статье будет представлено создание и использование объектов в базе данных Oracle на примере геометрической фигуры (треугольника). Для данного объекта будут выполнены следующие операции:

- построение таблицы для хранения типа Треугольник;
- создание дополнительной таблицы для хранения таблицы, хранящей объекты типа Треугольник;
- создание столбца-ссылки на объекты дополнительной таблицы.

Цель данной статьи – создать базу данных, хранящую информацию о рассматриваемом типе данных (треугольнике), провести работу с созданной БД в соответствии

с заданием, представленным в разделе «Описание проектируемой базы данных».

### Материалы и методы исследования

В качестве материала исследования выступает база данных Oracle.

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

### Результаты исследования и их обсуждение

#### *Описание проектируемой базы данных*

Треугольник в трехмерном пространстве с осями X, Y, Z, заданный координатами вершин. Для вершины (точки в пространстве) определяется объектный тип с тремя числовыми атрибутами – координатами. Для треугольников задается свой тип с тремя атрибутами – вершинами и методом, вычисляющим наибольшее расстояние точки, находящейся на треугольнике от плоскости X-O-Y.

*Задание*

1. Построить в БД новую таблицу с полем-объектом, для хранения геометрических фигур. Столбцы таблицы содержат: идентификатор фигуры – целое (первичный ключ), название фигуры – строка до 30 символов, объект – геометрическая фигура из варианта задания. Загрузить 3-4 строки таблицы данными. С помощью запросов вывести атрибуты и результаты выполнения методов для каждого объекта таблицы.

2. Создать отдельную (вторую) таблицу со строками-объектами и заполнить ее объектами из первой таблицы. Вывести атрибуты и результаты выполнения методов для каждого объекта созданной таблицы.

3. В первую таблицу добавить столбец-ссылку на объекты второй таблицы и занести в его некоторые поля ссылку на соответствующий объект второй таблицы. Пользуясь ссылками, вывести значение любого атрибута и результат выполнения метода для соответствующего объекта второй таблицы.

*Создание типов Точка и Треугольник*

Для работы с описанной базой данных необходимо создание двух типов объекта: Точка и Треугольник. Ниже представлено создание типа Точка (POINT, хранит координаты) и Треугольник (TRIANGLE, хранит вершины типа Точка), имеющего метод `max_distance_from_plane` (максимальное расстояние от плоскости XOY) [4–6].

```
--Создание типа Точка
CREATE OR REPLACE TYPE POINT AS OBJECT (
  x FLOAT,
  y FLOAT,
  z BLOAT
);

--Создание типа Треугольник
CREATE OR REPLACE TYPE TRIANGLE AS OBJECT (
  A POINT,
  B POINT,
  C POINT,
  --Функция для определения максимального расстояния от треугольника до
  плоскости XOY
  MEMBER FUNCTION max_distance_from_plane(t TRIANGLE) RETURN FLOAT
);

/
CREATE OR REPLACE TYPE BODY TRIANGLE AS MEMBER FUNCTION max_distance_from_
plane(t TRIANGLE) RETURN FLOAT IS
BEGIN
  IF (ABS(t.A.z) >= ABS(t.B.z) AND ABS(t.A.z) >= ABS(t.C.z)) THEN
    RETURN ABS(t.A.z);
  ELSE IF (ABS(t.B.z) >= ABS(t.A.z) AND ABS(t.B.z) >= ABS(t.C.z)) THEN
    RETURN ABS(t.B.z);
  ELSE
    RETURN ABS(t.C.z);
  END IF;
END IF;
END;

END;
```

Наибольшим расстоянием от плоскости X-O-Y до треугольника является вершина треугольника с наибольшим по модулю значением координаты z.

*Использование объектов**Построение таблицы для хранения геометрических фигур*

Для хранения данных о треугольниках создана таблица Figures. Данная таблица содержит следующие атрибуты:

- id (ID треугольника) имеет тип Number (число), является первичным ключом таблицы;
  - name (название треугольника) имеет тип VARCHAR2 (строка);
  - object (хранит данные о вершинах треугольника) имеет тип Triangle (треугольник).
- Ниже представлено создание таблицы Figures (с помощью команды CREATE TABLE), хранящей данные о геометрических фигурах [7–9].

```
--Создание таблицы Фигуры
CREATE TABLE FIGURES (
  id NUMBER(10) NOT NULL,
  name VARCHAR2 (30) NOT NULL,
  object TRIANGLE NOT NULL,
  CONSTRAINT pk_id PRIMARY KEY (id)
);
```

Вставка в таблицу Figures 3 строк с помощью команды INSERT INTO представлена ниже.

```
--Вставка строк в таблицу
INSERT INTO FIGURES(id, name, object)
SELECT 1, 'Треугольник № 1', TRIANGLE(POINT(1, 1, 1), POINT(1, 3, 1), POINT(1, 2, 3)) FROM DUAL
UNION ALL
SELECT 2, 'Треугольник № 2', TRIANGLE(POINT(2, 2, 2), POINT(2, 6, 2), POINT(2, 4, 6)) FROM DUAL
UNION ALL
SELECT 3, 'Треугольник № 3', TRIANGLE(POINT(3, 3, 3), POINT(3, 9, 3), POINT(3, 6, 9)) FROM DUAL;
```

Выведем значения атрибутов и вычисленное для каждого треугольника максимальное расстояние до плоскости XOY.

```
SELECT
  f.id "ID", f.name "Название фигуры",
  f.object.A "Вершина A", f.object.B "Вершина B", f.object.C "Вершина C",
  f.object.max_distance_from_plane(f.object) "Максимальное расстояние от плоскости XOY"
FROM FIGURES f;
```

Результат запроса изображен на рис. 1.

Название фигуры	ID	Вершина A	Вершина B	Вершина C	Максимальное расстояние от плоскости XOY
Треугольник № 1	1	[SYSTEM.POINT]	[SYSTEM.POINT]	[SYSTEM.POINT]	3
Треугольник № 2	2	[SYSTEM.POINT]	[SYSTEM.POINT]	[SYSTEM.POINT]	6
Треугольник № 3	3	[SYSTEM.POINT]	[SYSTEM.POINT]	[SYSTEM.POINT]	9

Рис. 1. Результат запроса

#### Создание дополнительной таблицы

Создание (команда CREATE TABLE) дополнительной таблицы NewFigures (данная таблица имеет такие же атрибуты (id, name и object)), содержащей объекты из первой таблицы, представлено ниже [10].

```
--Создание дополнительной таблицы
CREATE TABLE NEWFIGURES OF TRIANGLE;
```

Заполнение (команда INSERT INTO) таблицы NewFigures выполнено с помощью кода, представленного ниже.

```
--Заполнение таблицы
INSERT INTO NEWFIGURES SELECT f.object FROM FIGURES f;
SELECT
  f.A "Вершина A", f.B "Вершина B", f.C "Вершина C",
  f.max_distance_from_plane (TRIANGLE(f.A, f.B, f.C)) "Максимальное расстояние от плоскости XOY"
FROM NEWFIGURES f;
```

Данные созданной таблицы представлены на рис. 2.

	Вершина А	Вершина В	Вершина С	Максимальное расстояние от плоскости XOY
1	[SYSTEM.POINT]	[SYSTEM.POINT]	[SYSTEM.POINT]	3
2	[SYSTEM.POINT]	[SYSTEM.POINT]	[SYSTEM.POINT]	6
3	[SYSTEM.POINT]	[SYSTEM.POINT]	[SYSTEM.POINT]	9

Рис. 2. Данные таблицы

### Создание столбца-ссылки

Добавление столбца-ссылки на объекты дополнительной таблицы (с помощью команда ALTER TABLE и UPDATE) представлено ниже [11].

--Добавление ссылки на дополнительный столбец

```
ALTER TABLE FIGURES ADD (
    newfigures_ref REF TRIANGLE
);
```

--Обновление таблицы

```
UPDATE FIGURES f SET f.newfigures_ref = (
    SELECT REF(newf) FROM NEWFIGURES newf
    WHERE f.object.A.x = newf.A.x AND f.object.A.y = newf.A.y AND f.object.A.z = newf.A.z
    AND f.object.B.x = newf.B.x AND f.object.B.y = newf.B.y AND f.object.B.z = newf.B.z
    AND f.object.C.x = newf.C.x AND f.object.C.y = newf.C.y AND f.object.C.z = newf.C.z
);
```

Запрос данных через функцию deref() представлен ниже.

```
SELECT
    f.id "ID", f.name "Название фигуры",
    deref(f.newfigures_ref).A "Вершина А",
    deref (f.newfigures_ref).B "Вершина В",
    deref (f.newfigures_ref).C "Вершина С",
    deref (f.newfigures_ref).max_distance_from_plane(TRIANGLE(
    deref (f.newfigures_ref).A, deref(f.newfigures_ref).B, deref(f.newfigures_ref) .C))
    "Максимальное расстояние от плоскости XOY"
FROM FIGURES f;
```

Результат запроса изображен на рис. 3.

ID	Название фигуры	Вершина А	Вершина В	Вершина С	Максимальное расстояние от плоскости XOY
1	1 Треугольник № 1	[SYSTEM.POINT]	[SYSTEM.POINT]	[SYSTEM.POINT]	3
2	2 Треугольник № 2	[SYSTEM.POINT]	[SYSTEM.POINT]	[SYSTEM.POINT]	6
3	3 Треугольник № 3	[SYSTEM.POINT]	[SYSTEM.POINT]	[SYSTEM.POINT]	9

Рис. 3. Результат запроса

В данной статье представлено создание и использование объектов в базе данных Oracle. В качестве хранимого объекта выступает треугольник. Для работы с базой данных, в первую очередь, были созданы типы данных Точка (хранит координаты) и Треугольник (хранит вершины треугольника и метод, определяющий максимальное расстояние от плоскости X-O-Y до треугольника).

На основе созданных типов данных получены следующие результаты.

- Определена таблица Figures (содержит атрибуты id (ID треугольника, также

является первичным ключом таблицы), name (название треугольника) и object (содержит данные о вершинах треугольника)), хранящая данные о треугольниках.

- Создана таблица NewFigures, хранящая объекты из первой таблицы.

- Определен столбец-ссылка на объекты таблицы NewFigures. Выполнен запрос данных через функции. deref().

### Заключение

Основное отличие объектов от обычных данных заключается в том, что объекты содержат в себе несколько полей. Однако

создание и использование объектов практически не отличается от создания и использования обычной информации в базах данных, поскольку в одном и другом случае используются одинаковые команды (такие как CREATE, ALTER и пр.), что и представлено в данной статье на примере типа данных Треугольник.

#### Список литературы

1. Медведев Ю.С., Пиотровский Д.Л. Мониторинг производительности базы данных Oracle // Вестник АГУ. 2018. № 4. С. 210–213.

2. Объекты базы данных Oracle. [Электронный ресурс]. URL: [https://life-prog.ru/2\\_34406\\_ob-ekti-bazi-dannih-Oracle.html](https://life-prog.ru/2_34406_ob-ekti-bazi-dannih-Oracle.html) (дата обращения: 29.07.2020).

3. Объекты в Oracle. [Электронный ресурс]. URL: <http://citforum.ru/database/oracle/prz/prosto.shtml> (дата обращения: 29.07.2020).

4. Бьюли Алан. Изучаем SQL. СПб., М.: «Символ-плюс», 2016. 309 с.

5. Аллен Дж. Тейлор. SQL для чайников. М.: «Диалектика», 2014. 416 с.

6. Бен Форта. SQL за 10 минут. М.: «Вильямс», 2014. 288 с.

7. Рон Хардман, Майкл МакЛафлин. Oracle Database PL/SQL. Рекомендации эксперта. М.: «Лори», 2014. 450 с.

8. Линн Бейли. Изучаем SQL. СПб.: «Питер», 2012. 592 с.

9. Билл Карвин. Программирование баз данных SQL. Типичные ошибки и их устранение. М.: «Рид Групп», 2012. 336 с.

10. Том Кайт. Издательство: Эффективное проектирование приложений Oracle. М.: «Лори», 2006. 657 с.

11. Дж. Льюис. Oracle. Основы стоимостной оптимизации. СПб.: Питер, 2006. 528 с.