


Oracle как объектно-реляционная СУБД



Выбор точки зрения – это первичный акт культуры.

Х. Ортега-и-Гассет

Корпоративные системы баз данных © М.Л. Цымблер

Содержание

- Реляционные, объектно-реляционные и объектно-ориентированные СУБД
- Поддержка пользовательских типов данных в СУБД Oracle
- Поддержка технологии XML в СУБД Oracle

Корпоративные системы баз данных © М.Л. Цымблер 2

Объектно-реляционные СУБД

Характеристика \ Тип СУБД	Язык запросов	Сложность данных	Типы данных
Реляционные	SQL	Простые	Стандартные
Объектно-реляционные	SQL + OQL	Сложные	Стандартные + пользовательские
Объектно-ориентированные	OQL	Сложные	Стандартные + пользовательские

Корпоративные системы баз данных © М.Л. Цымблер 3

Терминология

- *Объект* объединяет в себе структуру данных и операции над этой структурой данных. Структура данных – *свойства* объекта. Операции – *методы* для анализа и изменения свойств.
- *Экземпляр* представляет значение объекта.
 - *Персистентный* экземпляр – хранится постоянно (в базе данных как объект схемы).
 - *Транзитный* экземпляр – существует только в оперативной памяти.

Корпоративные системы баз данных

© М.Л. Цымблер

4

Объектный тип

- *Объектный тип* служит абстракцией сущности предметной области.
- Составляющие объектного типа:
 - *имя* для идентификации в схеме;
 - *атрибуты* для моделирования структуры и состояния сущности – могут иметь встроенный или объектный тип;
 - *методы* для реализации операций над сущностью – могут быть реализованы на PL/SQL, C++, Ada, Java.

Корпоративные системы баз данных

© М.Л. Цымблер

5

Пример: определение объектного типа

```

create type TAddress as object (
  Country char(3),
  City char(20),
  ZipCode char(8), ...);
create type TPerson as object (
  Name char(50),
  Birth date,
  Address TAddress,
  member function Age return number);
create type body TPerson is
member function Age return number is
begin
  return round(months_between(sysdate, Birth)/12);
end;
end;
```

Корпоративные системы баз данных

© М.Л. Цымблер

6

Методы-конструкторы

- *Конструктор* – предопределенный метод объектного типа для создания экземпляров данного типа.

```
declare
  JohnDoe TPerson;
  JohnAddress TAddress;
  JohnAge number;
begin
  JohnDoe := TPerson('John Doe', 22-Apr-05, TAddress('USA', 'NY',
'12345'));
  JohnAddress := JohnDoe.Address;
  JohnAge := JohnDoe.Age();
end;
```

Методы сравнения экземпляров

- Пользователь может определить один из (но не оба одновременно) следующих *методов сравнения* экземпляров объектного типа :

- *map-метод* – сопоставляет одному экземпляру число;
- *order-метод* – выдает результат сравнения двух экземпляров (0, >0 или <0).

```
create type TPerson as object (
  ... ,
  map member function MapPerson
  return number);
create type body TPerson is
map member function MapPerson
  return number is
begin
  return Age();
end;
end;
```

```
create type TPerson as object (
  ... ,
  order member function MapPerson
  return number);
create type body TPerson is
order member function OrderPerson
  (x in TPerson) return number is
begin
  return Age() - x.Age();
end;
end;
```

Объектные типы в таблицах

- Поле реляционной таблицы может иметь объектный тип (поле – *объект-столбец*).

```
declare
  JohnDoe TPerson;
create table emp (
  empno number primary key,
  person TPerson);
JohnDoe := TPerson('John Doe', 22-Apr-05, TAddress('USA', 'NY',
'12345'));
insert into emp values (emp_seq.nextval, JohnDoe);
insert into emp values (emp_seq.nextval,
  TPerson('James Bond', 22-Apr-05, TAddress('ENG', 'London', '54321')));
select empno, person.Name, person.Age() from emp
order by person.Age() desc;
```

Объектные таблицы

- Запись реляционной таблицы может иметь объектный тип. В этом случае ее можно рассматривать
 - как реляционную таблицу из нескольких столбцов (их столько, сколько полей в объектном типе)
 - как таблицу из одного столбца, содержащего *объекты-строки*

```
create table PersonTab of TPerson;
```

```
insert into emp values (
  TPerson('James Bond', 22-Apr-05, TAddress('ENG', 'London', '54321'));
insert into emp values ('John Doe', 22-Apr-05, 'USA', 'NY', '12345');
```

```
select * from PersonTab p order by p.Age() desc;
select value(p) from PersonTab p order by value(p) desc;
```

Корпоративные системы баз данных

© М.Л. Цымблер

10

Идентификаторы объектов

- Объект-строка имеет скрытый назначаемый системой атрибут *ОИД* – *уникальный идентификатор* экземпляра.
- ОИД представляет собой 16-байтовое целое число. ОИД используется при построении ссылок на экземпляры объектных типов.
- Пользователь может специфицировать ОИД объекта-строки как первичный ключ соответствующей реляционной таблицы.

```
create table PersonTab of TPerson (Name primary key
  object id primary key;
```

Корпоративные системы баз данных

© М.Л. Цымблер

11

Ссылки на объектный тип

- Встроенный тип REF используется для организации ссылок на экземпляры объектов-строк.

```
declare
  PJohnDoe ref TPerson;
```

```
select ref(p) into PJohnDoe from PersonTab p
where p.Name='John Doe';
update PersonTab p set p.Address.City='LA'
where ref(p)=PJohnDoe;
```

Корпоративные системы баз данных

© М.Л. Цымблер

12

Раскрытие ссылок

```

create type TProject as object (
  ProjNo number,
  Name char(50),
  Budget number,
  MgrRef ref TPerson);

create table ProjectTab of TProject;

insert into ProjectTab values (1, 'Mission impossible', 0, PJohnDoe);

select ProjName, Budget, deref(MgrRef) from ProjectTab
where deref(MgrRef).Age() < 40;

update ProjectTab set MgrRef=PJohnDoe where Budget > 100000;

```

Корпоративные системы баз данных

© М.Л. Цымблер

13

Объектные типы-таблицы

- Запись реляционной таблицы может иметь поле – *вложенную таблицу*.

```

create type TProjectTab as table of TProject;
create table Dept (
  DeptNo number primary key,
  Name char(50),
  Projects TProjectTab) nested table Projects store as ProjectTab;
-- Удаление вложенной таблицы
insert into Dept values (1, 'Sales', NULL);
-- Создание вложенной таблицы
update Dept set Projects=TProjectTab() where DeptNo=1;
-- Вставка записей во вложенную таблицу
insert into table (select Projects from Dept where DeptNo=1)
values (1, 'Mission Impossible', 0, PJohnDoe);
-- Обновление записей вложенной таблицы
update Dept set Projects= TProjectTab(TProject(1, 'Mission Is Possible', 100000,
PJamesBond)) where DeptNo=1;

```

Корпоративные системы баз данных

© М.Л. Цымблер

14

Объектные типы-массивы

- Oracle поддерживает объектные типы-массивы.

```

create type TBox as varray(3) of number;
create table Goods (
  id number primary key,
  name char(20),
  Sizes TBox not null);

insert into Goods values (1, 'Juice', TBox(10, 5, 20));

select id, name, Sizes(1) as Length, Sizes(2) as Width, Sizes(3) as Height,
Sizes(1)*Sizes(2)*Sizes(3) as Volume
from Goods;

```

Корпоративные системы баз данных

© М.Л. Цымблер

15

Целостность данных

- Висячие ссылки
- Циклические ссылки
- Ограничение области действия ссылок
- NULL-объекты
- Указание значений по умолчанию
- Ограничения целостности
- Триггеры

Висячие ссылки

```
create type TProject as object (
  ProjNo number,
  Name char(50),
  Budget number,
  MgrRef ref TPerson);

create table ProjectTab of TProject;

select ProjName, Budget, deref(MgrRef) from ProjectTab
where MgrRef is not dangling;

update ProjectTab set MgrRef=PJohnDoe where MgrRef is dangling;
```

Циклические ссылки

```
create type TDept; -- Незавершенное определение типа

create type TEmp as object (
  name char(30),
  dept ref TDept,
  mgr ref TEmp);

create type TEmpList as table of TEmp;

create type TDept as object (
  name char(30),
  mgr ref TEmp,
  staff TEmpList);
```

Ограничение области действия ссылок

- Область действия ссылки на объект-столбец может быть ограничена объектами-строками указанных таблиц (при этом типы объекта-столбца и объекта-строки должны совпадать).

```
create type TProject as object (
  ProjNo number,
  Name char(50),
  Budget number,
  MgrRef ref TPerson scope is PersonTab);
```

```
alter table ProjectTab
  add (scope for MgrRef is EmpTab);
```

NULL-объекты

```
create type TContact as object (
  name varchar2(30),
  phone varchar2(20) );
create table daycontacts (
  contact TContact,
  day date );
```

-- Выделяется место для размещения экземпляра

```
insert into daycontacts values (
  TContact(NULL, NULL), '22.04.2005' );
```

-- Место для размещения экземпляра НЕ выделяется

```
insert into daycontacts values (
  NULL, '22.04.2005' );
```

Указание значений по умолчанию

```
create TPeople as table of TPerson;
```

```
create table Department (
  deptno char(5) primary key,
  name char(20),
  mgr TPerson default
    TPerson(0,'John Doe',null),
  emps TPeople default
    TPeople(
      TPerson(1,'John Smith',null),
      TPerson(7, 'James Bond', null)) )
nested table emps store as empstab;
```

Ограничения целостности

- Ограничения целостности могут быть заданы только при непосредственном создании таблицы (не при создании пользовательского типа).

```
create table emps (
  emp TPerson,
  mgr ref TPerson,
  constraint pk_name emp.name primary key,
  constraint fk_mgr foreign key (mgr) references emps (emp),
  constraint unq_arph unique
    (emp.phonenum.area, contact.phonenum.phone),
  constraint nn_address check (emp.address is not null) );
```

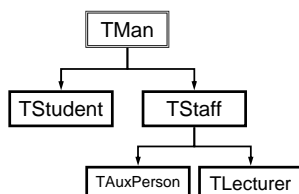
Триггеры

- Можно создать триггер объектной таблицы (так же, как триггер реляционной таблицы).
- Нельзя создать триггер вложенной таблицы.

Подтипы и супертипы (наследование)

- В СУБД Oracle 9.2 поддерживается простое (не множественное) наследование пользовательских типов.

```
create type TMan as object (
  name char(50),
  gender char(1) ) not final;
create type TStudent under TMan
as object ( squad char(20) ) not final;
create type TStaff under TMan
as object ( salary number,
  office char(3) ) not final;
create type TLecturer under TStaff
as object ( degree char(10),
  dept char(50) ) not final;
create type TAuxPerson under TStaff
as object ( job char(20) ) not final;
```



Подтипы и супертипы (наследование)

```
select supertype_name, subtype_name
from user_types
order by supertype_name,
subtype_name ;
```

SUPERTYPE_NAME	SUBTYPE_NAME
NULL	TMAN
TMAN	TSTAFF
TMAN	TSTUDENT
TSTAFF	TAUXPERSON
TSTAFF	TLECTURER

```

graph TD
    TMan --> TStudent
    TMan --> TStaff
    TStaff --> TAuxPerson
    TStaff --> TLECTURER
    
```

Корпоративные системы баз данных © М.Л. Цымблер 25

Подтипы и супертипы (наследование)

```
create type TCheck as object (
    when date;
    person TMan );
create table chkpoint of TCheck;

insert into chkpoint values ( sysdate,
    TMan('John Doe', 'M') );
insert into chkpoint values ( sysdate,
    TStudent('Иванов И.И.', 'M', 'МП-401') );
insert into chkpoint values ( sysdate,
    TStaff('Сидорова М.М.', 'Ж', 1500, '8') );
insert into chkpoint values ( sysdate,
    TLECTURER('Петрова А.П.', 'Ж', 3500, '8', 'ктн', 'каф. АСУ') );
insert into chkpoint values ( sysdate,
    TAuxPerson ('James Bond', 'M', 7000, '007', 'Cleaner') );
```

Корпоративные системы баз данных © М.Л. Цымблер 26

Подтипы и супертипы (наследование)

```
select * from chkpoint;
```

NAME	GENDER
James Bond	M
John Doe	M
Иванов И.И.	M
Петрова А.П.	Ж
Сидорова М.М.	Ж

```
select * from chkpoint
where person is of (TStudent);
```

NAME	GENDER	SQUAD
Иванов И.И.	M	МП-401

```
select * from chkpoint
where person is of (TStaff);
```

NAME	GENDER	SALARY	OFFICE
James Bond	M	7000	007
Петрова А.П.	Ж	3500	8
Сидорова М.М.	Ж	1500	8

```
select * from chkpoint
where person is of (only TStaff);
```

NAME	GENDER	SALARY	OFFICE
Сидорова М.М.	Ж	1500	8

Корпоративные системы баз данных © М.Л. Цымблер 27

Пользовательские типы в приложениях баз данных

- *Утилита Pro*C/C++* – генератор определений структур/классов C/C++, адекватных пользовательским типам, для последующего использования в приложениях на C/C++ (вызовы встроенного SQL).
- *Утилита JPublisher* – генератор определений классов Java, адекватных пользовательским типам, для последующего использования в приложениях на Java (вызовы встроенного SQL).

Объектные представления

- *Объектное представление (object view)* – виртуальная объектная таблица на базе реляционной либо объектной таблицы. Используется для конвертирования данных из реляционных таблиц в объектные и освоения объектно-реляционных техник работы с базой данных.
- Объектные представления не-обновляемы; их обновление реализуется с помощью триггеров INSTEAD OF.

```
create table emp (
  empno number primary key,
  name char(50),
  birth date,
  salary number);
create type TEmp (
  EmpNo number primary key,
  Name char(50),
  Birth date,
  Salary number);
```

```
create view HighlyPaidEmp of TEmp
with object OID (EmpNo) as
select * from emp
where Salary>1500;
```

Минусы ОР-возможностей

- Отсутствие полной поддержки ОО-концепций (нет виртуальных методов).
- Сложный синтаксис ОР-запросов.
- Сложность внесения изменений в объектную схему данных.

Пример. Для добавления атрибута в объектный тип-строку объектной таблицы нужно (1) сохранить данные во временной таблице, (2) удалить таблицу, (3) переопределить объектный тип, (4) создать таблицу заново, (5) поместить в нее данные из временной таблицы.

Пример ОР-системы (самостоятельно)

- Автоматизация работы отдела поставок: реляционный и объектно-реляционный подход – см. Oracle Application Developer's Guide, глава User-Defined Datatypes.

```
create table Customer_reltab (
  CustNo number not null,
  CustName char(200) not null,
  Street char(200) not null,
  City char(200) not null,
  State char(2) not null,
  Zip char(20) not null,
  Phone1 char(20),
  Phone2 char(20),
  Phone3 char(20),
  primary key (CustNo) );
...

create type PhoneList_vartyp as
  varray(10) of char(20) ;
create type Address_objtyp as object (
  Street char(200),
  City char(200),
  State char(2),
  Zip char(20) );
create type customer_objtyp as object (
  custno number, custname char(200),
  address_obj address_objtyp,
  phonenumber var phoneList_vartyp,
  order member function cmpcustorders
    (x in customer_objtyp) return integer );
```

Корпоративные системы баз данных

© М.Л. Цымблер

31

Поддержка технологии XML

- В СУБД Oracle 9.0 введен объектный тип данных XMLTYPE.

```
create table books (
  bookno number primary key,
  description XMLTYPE);
insert into books values (
  100, XMLTYPE('
<book>
  <title> Язык программирования Си </title>
  <author> Б. Керниган </author>
  <author> Д. Ритчи </author>
  <pages>271 </pages>
</book>
'));
select bookno, b.description from books b;
select bookno, b.description.XMLDATA FROM books b;
```

Корпоративные системы баз данных

© М.Л. Цымблер

32

Поддержка технологии XML

```
insert into books values (
  101, XMLTYPE('
<?xml version="1.0"?>
<!DOCTYPE book [
  <!ELEMENT book (title, author*, pages)>
  <!ELEMENT title (#PCDATA)>
  <!ELEMENT author (#PCDATA)>
  <!ELEMENT pages (#PCDATA)>
]>
<book>
  <title> Незнайка на Луне </title>
  <author> Н. Носов </author>
  <pages> 103 </pages>
</book>
'));
```

Корпоративные системы баз данных

© М.Л. Цымблер

33
