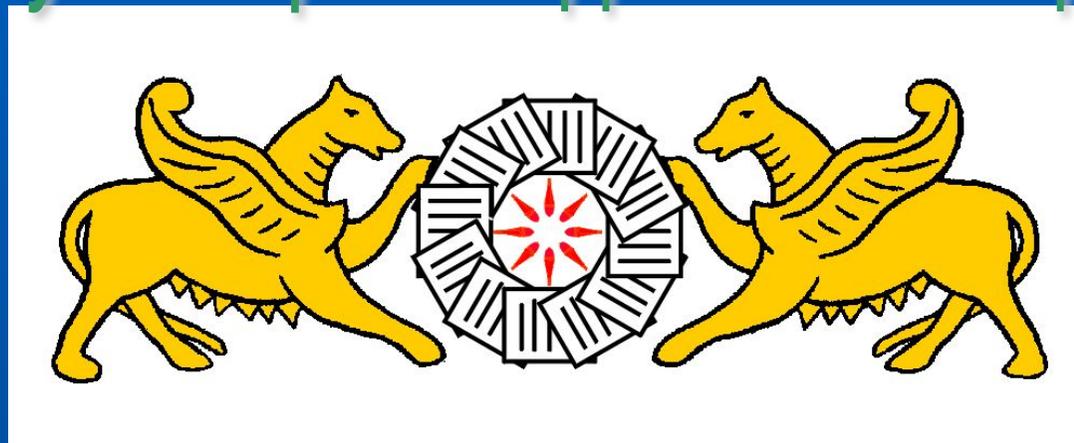


Научно-производственный центр



ИНТЕЛТЕК ПЛЮС

<http://www.inteltec.ru>

Тема доклада

Внутренняя организация ОСУБД на примере Versant, Poet, ODB-Jupiter

Области применения ОСУБД

- САПР, Case-средства
- Автоматизация производственных процессов
- Автоматизация управления предприятием
- Геоинформационные, картографические системы
- Телекоммуникации
- Internet приложения

ОСУБД **Versant**

Компания Versant представляет программные продукты:

- **Versant Developer Suite** - сервер ОСУБД + комплект разработчика;
- **Versant enJin** - сервер объектно-ориентированного промежуточного уровня, ориентированный на обработку Java-объектов.

Основной акцент делается на разработку приложений средствами Java, хотя при этом обеспечивается работа с ОСУБД на языках C, C++, Smalltalk.

<http://www.versant.com>

ОСУБД Poet

POET Software представляет следующие продукты:

- **Poet Database** - сервер хранилища данных;
- **Poet Object Server** - диспетчер объектов ОСУБД;
- **SQL Object Factory** - сервер объектно-ориентированного промежуточного уровня, обеспечивающий прозрачную работу пользователей как с собственным сервером ОСУБД, так и с инородными для нее СУБД типа Oracle, DB2, MS SQL.

Значительное внимание уделяется на разработку Java-приложений.

<http://www.poet.com>, <http://support.poet.de>

ОСУБД **ODB-Jupiter**

Компания НПЦ «ИНТЕЛТЕК ПЛЮС» представляет ОСУБД **ODB-Jupiter**. В состав которой входят компоненты:

- сервер хранилища объектов;
- сервер-диспетчер объектов, также реализующий функции объектно-ориентированного промежуточного уровня

В настоящее время на основе ОСУБД ODB-Jupiter создан коммерческий продукт - информационно-поисковая система ODB-Text.

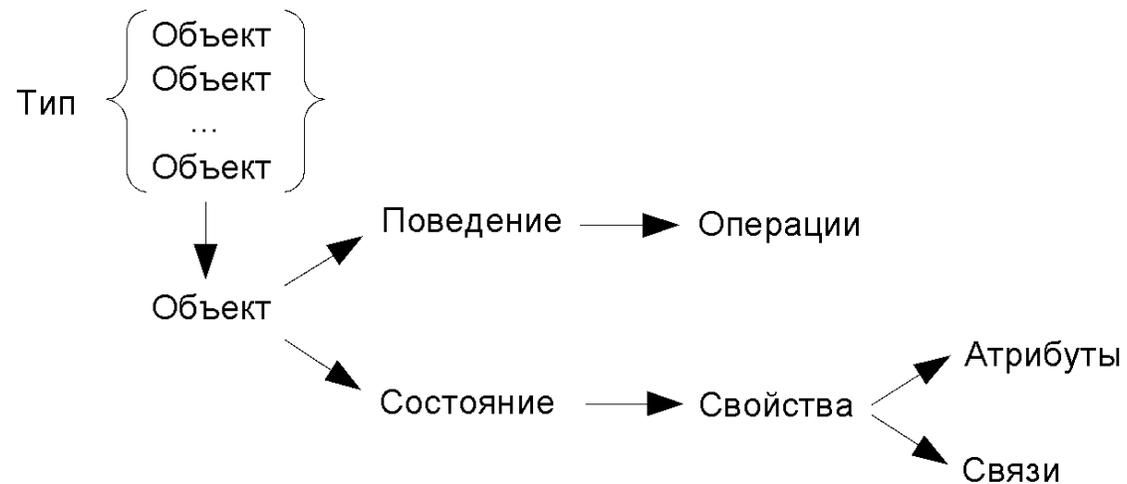
<http://www.inteltec.ru>

Особенности построения ОСУБД. Модель данных.

Требования ODMG 3.0

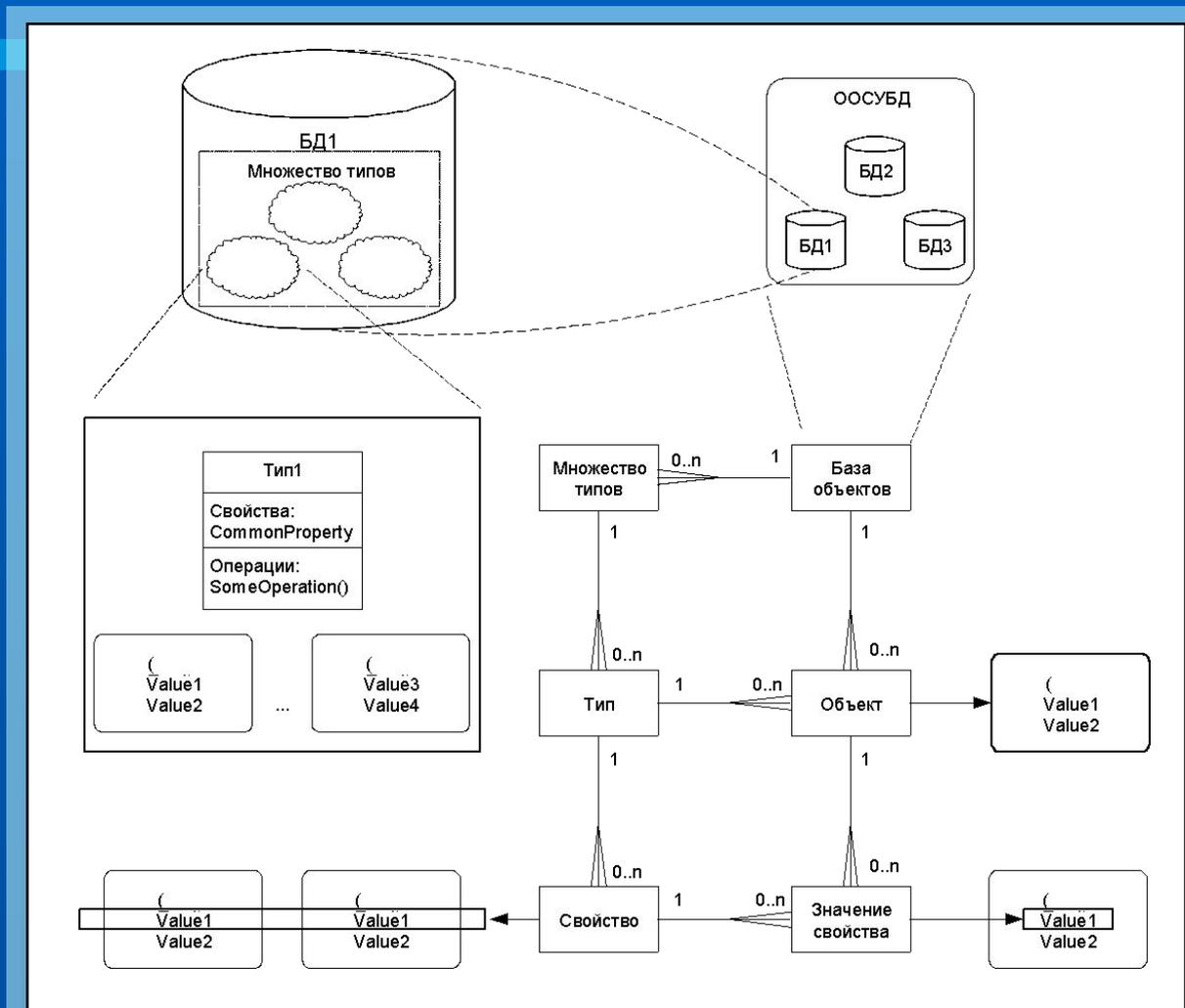
- Объектная модель, предписывающая формат представление данных, их однозначную идентифицируемость, методы типизации данных и построения иерархий типов;
- Язык запросов OQL для ОСУБД;
- Набор служебных функций для организации доступа к базе данных, получения схемы данных, работы с транзакциями;
- Библиотеки программиста для работы с ОСУБД для языков Java, C++, Smalltalk

Особенности построения ОСУБД. Модель данных.



Основные элементы объектной модели

Особенности построения ОСУБД. Модель данных.



Идентификаторы объектов ОСУБД.

ОСУБД Versant

OID = { Storage ID : Object ID }

Storage ID - 4 байта

Object ID - 8 байт

ОСУБД Poet

OID = { Class ID : Object ID }

Class ID, Object ID - 4 байта

ОСУБД ODB-Jupiter

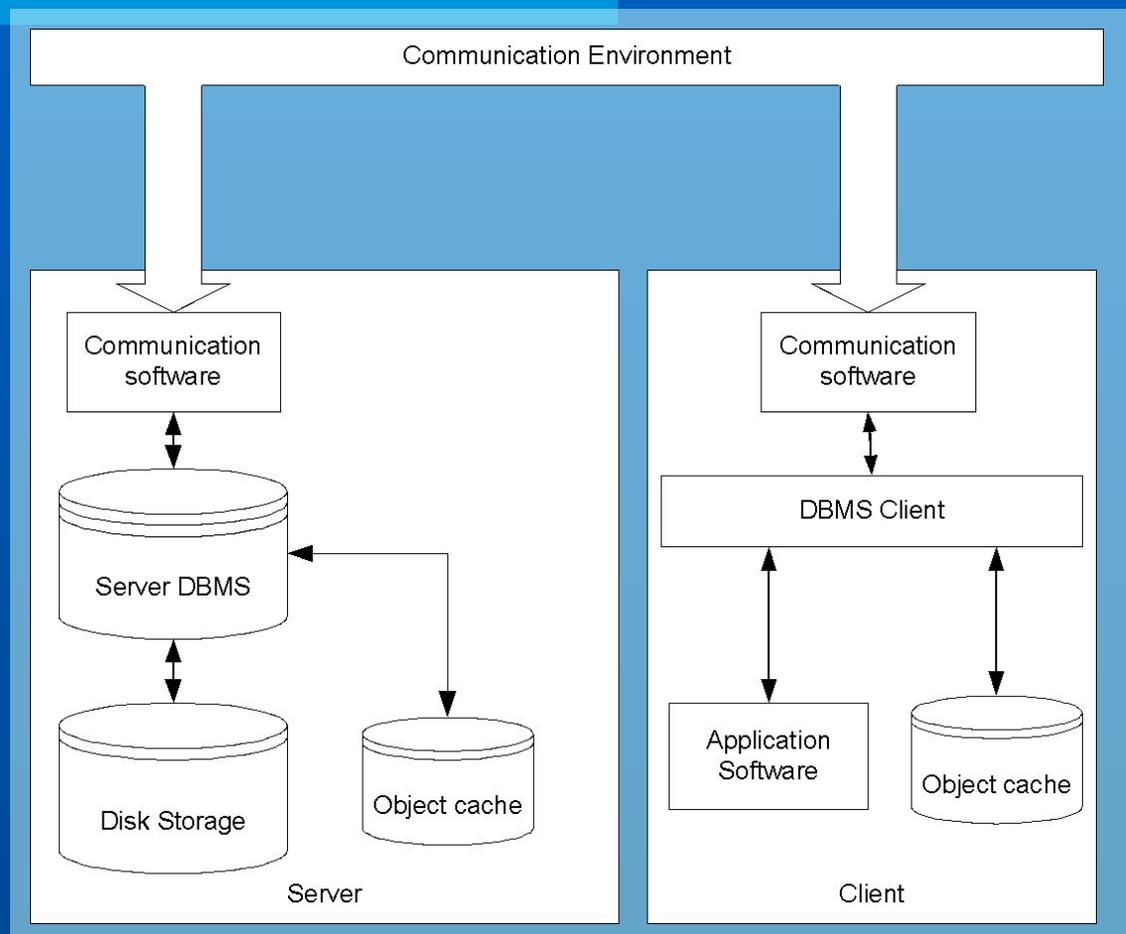
OID = { Storage ID : Class ID : Object ID }

Storage ID - 4 байта

Object ID - 8 байт

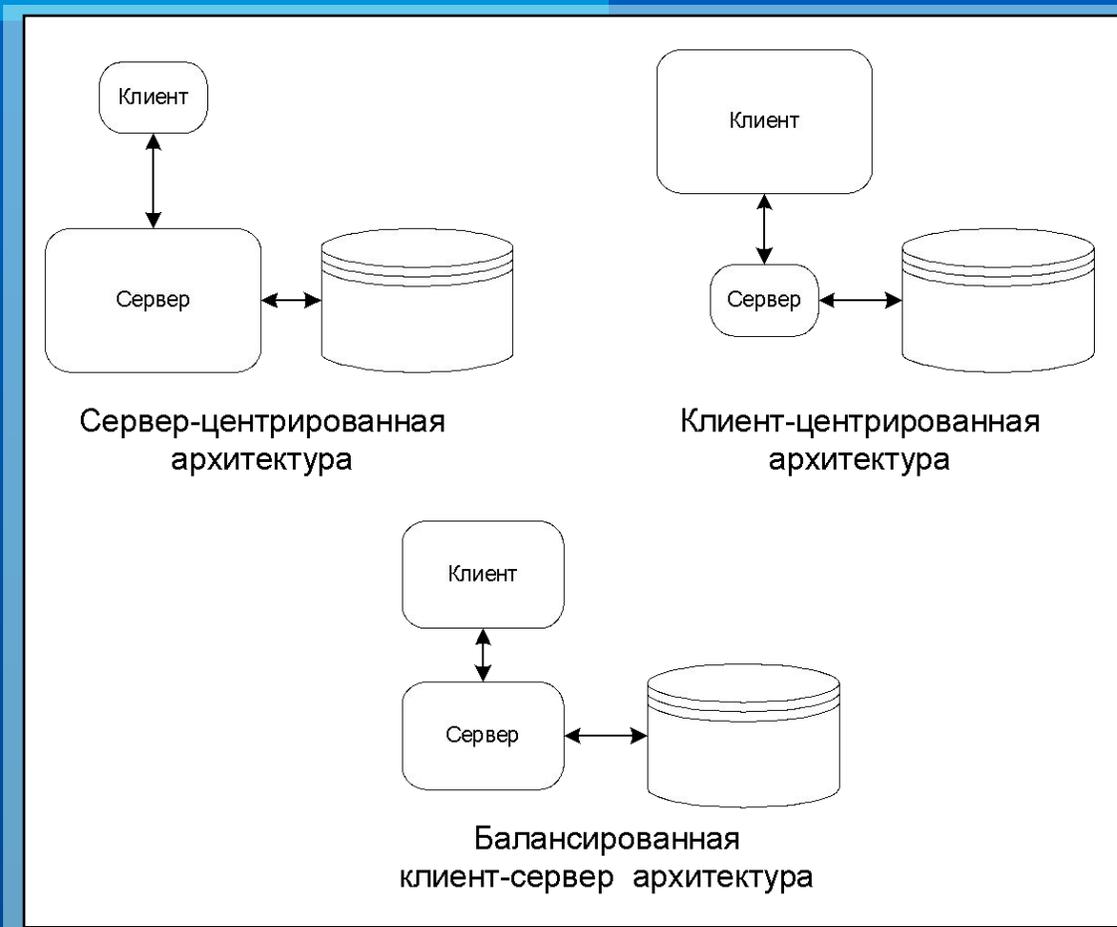
Class ID - строка без ограничения длины

Особенности построения ОСУБД. Методы организации клиент-серверного взаимодействия.



Клиент-серверная архитектура ОСУБД

Особенности построения ОСУБД. Методы организации клиент-серверного взаимодействия.



Клиент-серверные архитектуры СУБД

Организация хранилищ данных

-  системное хранилище используется для хранения системы классов. Системное хранилище создается на этапе создания базы данных и содержит информацию о классах, о наличии и месторасположении пользовательских хранилищах;
-  пользовательское хранилище для хранения пользовательских объектов;
-  служебное хранилище, функция которого - хранение временной информации, например о заблокированных объектах, об активных транзакциях, различного вида списки запросов пользователей и т.д.

Организация хранилищ данных в ОСУБД **Versant**

Классификация по типу базы данных:

- 1) Групповые базы данных;
- 2) Персональные базы данных.

Классификация по типу функциональному назначению:

- системный раздел (system volume) - создается автоматически при формировании базы данных;
- раздел данных (data volumes) - может быть добавлен для увеличения доступного объема базы;
- раздел логического протокола (logical log volume) и раздел физического протокола (physical log volume) предназначены для хранения служебной информации о транзакциях, блокировках для обеспечения возможности отката транзакций и восстановления базы.

Организация хранилищ данных в ОСУБД

ОДВ-Jupiter

-  системное хранилище, которое содержит описатели схемы данных и ряд служебных объектов;
-  хранилище безопасности, хранящее информацию о группах, пользователях и их прав доступа;
-  пользовательские хранилища для хранения объектов пользователей.

Организация хранилищ данных в ОСУБД

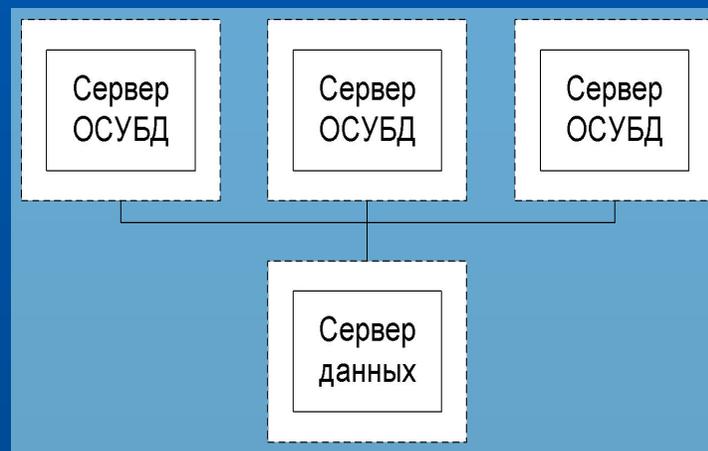
ODB-Jupiter

Варианты организации взаимодействия серверов ОСУБД и сервера данных

•Единая ЭВМ



•Различные ЭВМ



Некоторые функции ОСУБД

- ведение схемы данных
- ведение версий объектов;
- транзакции и блокировки;
- длинные транзакции и блокировки;
- разделение доступа к объектам.

Ведение схемы данных ОСУБД

Схема данных является основой функционирования ОСУБД, содержит описание хранимых типов, их иерархии и взаимосвязи.

Изменение схемы данных ОСУБД, как правило, не требует изменения хранимых данных.

ОСУБД предусматривают динамическое изменение схемы данных.

Ведение схемы данных ОСУБД

ODB-Jupiter

- выполнение функций, предусмотренных ODMG;
- возможность ведения группы алиасов атрибута;
- возможность работы с «анонимными» объектами.

Некоторые характеристики ОСУБД на примере **ODV-Jupiter**

- количество типов в схеме данных: до 2^{16}
- количество атрибутов типа: до 2^{16}
- количество объектов в одном хранилище: до 2^{50}
- количество объектов одного типа и хранилища: до 2^{64}
- объем данных, содержащихся в объекте: до 2^{64}
- количество хранилищ: до 2^{32}

Ведение версий объектов



Управление версиями в ОСУБД Versant

Разделение доступа к объектам

ОСУБД Poet, ОСУБД Versant - необходима самостоятельная реализация механизма разделения доступа к объектам.

ОСУБД ODB-Jupiter - реализует разделение доступа в соответствии с классом «B3» Orange book и классом «5» ГОСТ Р 51241-98.

Транзакции и блокировки

Транзакции:

- короткие;
- длинные;
- вложенные;
- параллельные.

Блокировки:

- короткие;
- продолжительные;
- оптимистические.

Язык запросов ОСУБД

Группа ODMG для связи с ОСУБД разработала язык OQL:

- Язык опирается на объектную модель ODMG;
- основная конструкция языка - это конструкция `SELECT...FROM...WHERE...GROUP BY...HAVING...ORDER BY...`;
- результат запросов представляет собой множество объектов. Сложные взаимосвязи объектов могут напрямую указываться в запросе, что является одной из отличительных особенностей OQL от языка SQL;
- язык является функциональным, операторы могут объединяться в выражение в качестве операндов;
- может использоваться в качестве языка программирования;
- возможно увеличение его функциональности.

Язык запросов ОСУБД

Группы операторов OQL:

- операторы сравнения - операторы типа равно, не равно, больше, меньше и их комбинации;
- строковые операции, позволяющие указывать требование на полное совпадение, частичное совпадение или на соответствие некоторой последовательности;
- операции с классами, конструкторы стандартных типов set, bag, list, array(..), struct(..), а также конструкторы типов пользователя;
- операции с множествами объектов типа пересечения, объединения, дополнения и др.;
- предикаты объединения типа AND, OR, NOT, используемые для указания условия на поиск данных;
- простые выражения типа сложения, вычитания, умножения, деления, остаток от деления;

Язык запросов ОСУБД

Пример запроса в OQL:

```
select distinct struct( name: x.name, hps:(select y
                                from x.subordinates as y
                                where y.salary>100000))
from Employees x
```

Тип результата:

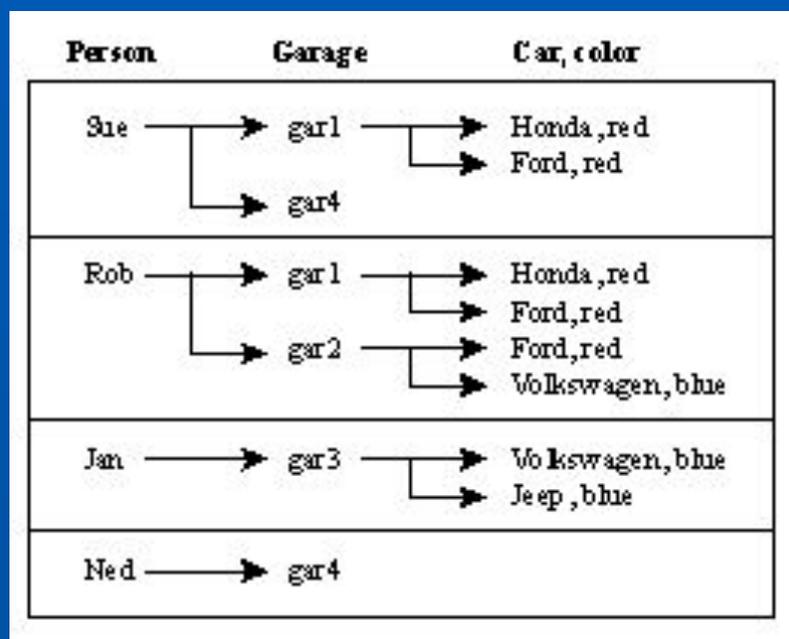
```
set<struct (name:string, hps:bag<Employee>) >
```

Язык запросов ОСУБД. Расширение языка **OQL** в СУБД **Versant**:

- В функциональных библиотеках ОСУБД добавлены операции с классами, позволяющие указать, возвращать ли объекты непосредственно базового класса, или только объекты классов потомков;
- возможно управление логическими путями в запросах;
- введены дополнительные конструкции SQL;
- введено понятие «виртуальный атрибут».

Язык запросов ОСУБД. Расширение языка **OQL** в СУБД **Versant**:

Управление логическими путями в запросах



car color == «red», O("exists")

Sue - Одна ссылка со значением «red».

Rob - Одна ссылка со значением «red».

car color == «red», O_ALL_PATHS

Sue - Две завершённые ссылки, все ссылки имеют значение «red».

car color == «red», O_EMPTY_TRUE

Sue - Одна ссылка со значением «red».

Rob - Одна ссылка со значением «red».

Ned - Нет завершённых ссылок.

Язык запросов ОСУБД. Расширение языка **OQL** в СУБД **Versant**:

Конструкции Versant Query Language(VQL):

COMMIT	Commit transaction.
DELETE	Delete objects.
INSERT	Insert objects.
QUIT	Terminate application.
ROLLBACK	Roll back transaction.
SELECT	Find objects.
UPDATE	Update objects.

Язык запросов ОСУБД. Расширение языка **OQL** в СУБД **Versant**:

Виртуальные атрибуты ОСУБД Versant:

Пример: выборка книг на немецком языке, имеющих названия, начинающиеся с букв определенного диапазона.

```
PClassObject<Book>::Object().select(NULL, FALSE,  
PPredicate(PAttribute("/national de_DE utf8 Book::title") > "U" &&  
PAttribute("/national de_DE utf8 Book::title") == "V"));
```

Результат:

книги, имеющие названия, начинающиеся с букв диапазона ["U", "V"], то есть "U", "U" и "V".

Язык запросов ОСУБД. Поиск текстовых данных

- ОСУБД Versant: поддержка национальных кодировок;
- ОСУБД Poet: хранение текста в формате Unicode, система Verity® для поиска с учетом морфологии языка, по лексикографической или фонетической близости;
- ОСУБД ODB-Jupiter: изначально ориентирована на поиск текста на русском языке. Содержит специальный язык запроса с возможностью указания параметров запроса, позволяющий вводить запрос на естественном русском языке.

Язык запросов ОСУБД. Поиск текстовых данных

Язык запросов ОСУБД ODB-Jupiter включает конструкции, позволяющие обрабатывать текстовые данные, в частности, на русском языке:

Название операции	Формат записи	Комментарий
Точное соответствие	xxxxxx	Поиск объектов, значение которых точно соответствует указанному
Пересечение	xxxxx yyyyy	Поиск только тех объектов, значения которых как xxxxx, yyyyy
Объединение	xxxxx, yyyyy	Поиск всех объектов, значения которых как xxxxx, yyyyy
Отрицание	-xxxx	Из результата поиска будут удалены объекты, у которых имеются значения xxxxx
Пусто	<	Поиск объектов, значения которых - NULL
Не пусто	-<	Поиск объектов, значения которых - не NULL
Приоритет	(xxxx, yyy) zzzz	Поиск объектов, значения которых как (xxxx zzzz), yyy zzzz
Расстояние между словами	xxxx \ n / yyy	Поиск объектов, у которых слова xxxx yyy встречаются на расстоянии не более n слов
Маскирование окончания	xxxx*	Поиск всех объектов, у которых встречаются значения, начинающиеся с xxxx

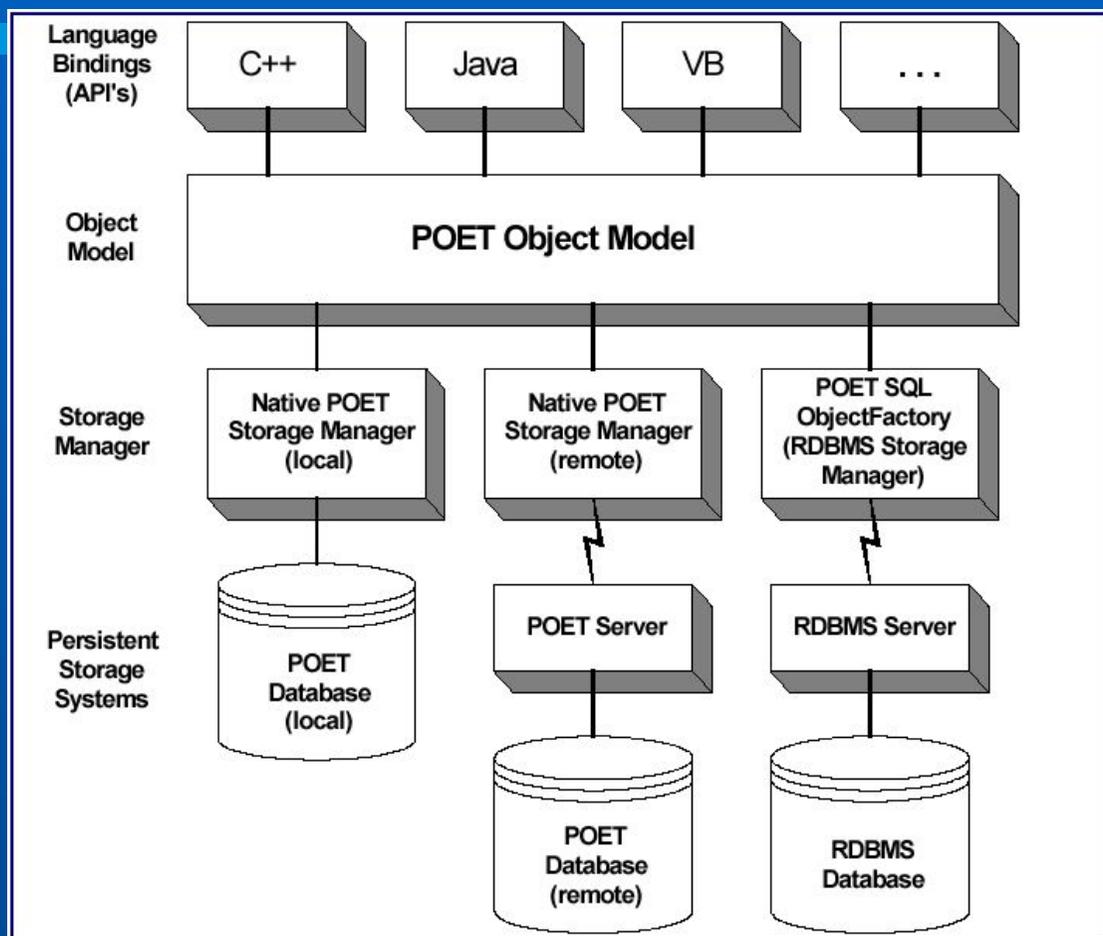
Язык запросов ОСУБД. Тенденции развития

- ◆ Унифицированный язык запросов СУБД;
- ◆ Типовой набор функций, выполняемых СУБД (select, insert, update);
- ◆ Типовой набор сервисных функций, выполняемых СУБД (sin, cos, ln);
- ◆ Расширяемость за счет возможности создания пользовательских функций;
- ◆ Поддержка расширений базового языка SQL за счет специализированных дополнений:
 - полнотекстовый поиск текстовых данных;
 - поиск не текстовых данных;
 - преобразование форматов данных;
- ◆ Разделение прав доступа к данным к данным.

Промежуточный объектно-ориентированный слой

- **Использование данных из СУБД сторонних производителей;**
- **расширение функциональных возможностей внешних СУБД за счет эмуляции отсутствующих функций;**
- **обеспечение «прозрачного» интерфейса доступа к данным для пользователей ОСУБД;**
- **оптимальное использование внешних баз данных для хранения классификаторов и т.д.;**
- **использование таких СУБД как DB2, Oracle.**

Промежуточный объектно-ориентированный слой **Poet** **SQL Object Factory**



Варианты организации внутреннего взаимодействия ОСУБД Poet

Отображение объектной схемы данных на реляционную структуру

Описание класса на языке C++:

```
persistent class Person
{
    PtString name;
    int age;
    PtDate birthDate;
};
```

Результат преобразования в запрос на языке определения данных РСУБД:

```
CREATE TABLE
    SCHEMA.PERSONC101V0 (
-- Identity columns
    OID NUMBER (38) not null,
    CID NUMBER (38) not null,
-- Columns for class members
    NAME0V0 VARCHAR2 (50),
    AGE1V0 NUMBER (38),
    BIRTHDATE2V0 DATE,
    PRIMARY KEY ( OID ) );
```

Отображение объектной схемы данных на реляционную структуру

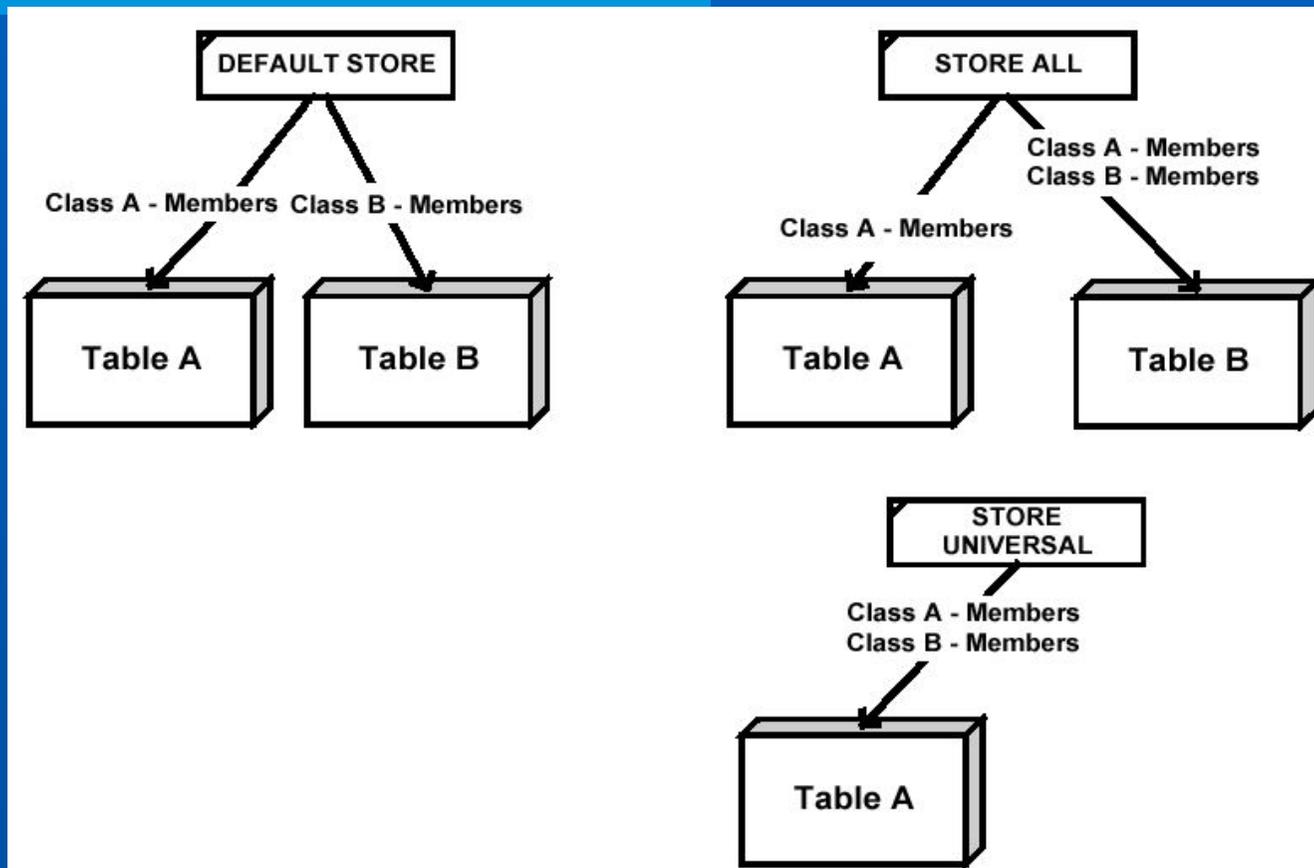
Пример C++ POET Object Model:

```
persistent class Car
{
  PtString make;
  PtString model;
  int year;
};
persistent class RichPerson
{
  PtString name;
  lset<Car *> fleet;
  lset<PtString> villas;
};
```

Результат преобразования в запрос DDL СУБД Oracle:

```
CREATE TABLE SCHEMA.RICHPERSONC102V0
( OID NUMBER (38) not null, CID NUMBER (38) not null,
  NAME0V0 VARCHAR2 (50), PRIMARY KEY ( OID )
);
CREATE TABLE SCHEMA.FLEETC102P0V0
( OID NUMBER (38) not null,
  CID NUMBER (38) not null,
  EID NUMBER (38) not null,
  -- An Object ID points to the actual persistent object.
  CAROID0V0 NUMBER (38), CARCID1V0 NUMBER (38),
  PRIMARY KEY ( OID, EID )
);
CREATE TABLE SCHEMA.VILLASC102P1V0
( OID NUMBER (38) not null, CID NUMBER (38) not null,
  EID NUMBER (38) not null,
  DATA0V0 VARCHAR2 (50), -- PtString elements
  PRIMARY KEY ( OID, EID )
```

Варианты реализации наследования при отображении на реляционную структуру



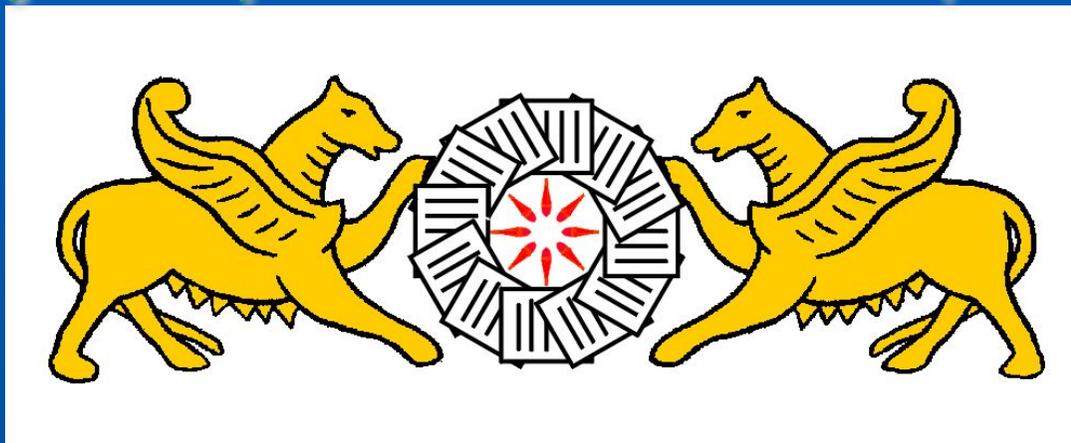
Варианты организации наследования в Poet SQL Object Factory.

Заключение

Тенденции развития ОСУБД:

- унификация языка запросов;
- формализация методов работы с объектами и управления работой ОСУБД;
- построение разделяемых компонентных систем;
- создание объектных надстроек, позволяющих интегрировать разнородные СУБД в рамках единого интерфейса программиста

Научно-производственный центр



ИНТЕЛТЕК ПЛЮС

<http://www.inteltec.ru>

Язык запросов ОСУБД

Допустимые имена атрибутов

```
struct Date {  
  int day;  
  Base b;  
};  
  
struct Base {  
  int id;  
  int code;  
};
```

Допустимые:

```
Date::day  
Date::b.Base::id  
Date::b.Base::code
```

```
struct Date {  
  int day;  
  Base *b;  
};  
  
struct Base {  
  int id;  
  int code;  
};
```

Допустимые:

```
Date::day  
Date::b->Base::id  
Date::b->Base::code
```

Язык запросов ОСУБД

Определение именованных запросов в OQL:

```
define age(x) as
    select p.age
    from Persons p
    where p.name=x;
```

Определение типов в OQL и варианты создания их объектов:

```
typedef set<integer> vectint;
```

```
vectint(select distinct age from Persons where name="John")
```

```
vectint(set(1,3,10))
```

Язык запросов ОСУБД

Обработка элементов пронумерованных коллекций в OQL:

```
list(a,b,c,d) [1:3]
```

```
element( select x from Courses x
         where x.name="OQL" and x.number="101").requires [0:2]
```

```
first(element( select x from Courses x
              where x.name="OQL" and x.number="101").requires)
```

```
list(1,2)+list(2,3) -> list(1,2,2,3)
```

```
Dict["OQL"]
```

Язык запросов ОСУБД

Операции с группами в OQL:

`bag(2,2,3,3,3) union bag(2,3,3,3) -> bag(2,2,3,3,3, 2,3,3,3)`

`bag(2,2,3,3,3) intersect bag(2,3,3,3) -> bag(2,3,3,3)`

`bag(2,2,3,3,3) except bag(2,3,3,3) -> bag(2)`

Язык запросов ОСУБД

Операции преобразования в OQL:

```
element(select x from Professors x where x.name="Jemand")
```

```
listtoset(list(1,2,3,2)) -> set(1,2,3)
```

```
distinct(list(1,4,2,3,2,4,1)) -> list(1,4,2,3)
```

```
flatten(list(set(1,2,3),set(3,4,5,6),set(7,8))) ->  
set(1,2,3,4,5,6,7,8)
```