

СЕТЕВАЯ БАЗА ДАННЫХ, ПОСТРОЕННАЯ ПО ОБЪЕКТНО-АТРИБУТНОМУ ПРИНЦИПУ^{1,2}

Салибекян Сергей Михайлович, к.т.н., доцент, Московский институт электроники и математики Национального исследовательского университета «Высшая школа экономики», Москва,
ssalibekyan@hse.ru

Белоусов Алексей Юрьевич, студент, Московский государственный университет технологий и управления имени Г.К. Разумовского, Москва, 3ret@rambler.ru

Введение

Уже на протяжении 30 лет наибольшей популярностью пользуются реляционные базы данных (БД) [1]. Однако сейчас реляционная модель перестала удовлетворять потребностям времени: ограниченность в средствах описания онтологии не позволяет применять ее в интеллектуальных системах, трудности масштабирования затрудняют реализацию подобных СУБД на параллельных и распределенных вычислительных системах. В ответ на создавшийся кризис СУБД [2] было предложено множество путей выхода из него – объектно-ориентированный (ОО) подход к БД, древовидная БД, сетевая модель, движение NoSQL (Amazon, Google). Однако ни один из них, кроме ОО-модели, которой также присуща масса недостатков [3], не сумел составить достаточно серьезной конкуренции реляционной модели. В статье мы предложим свой способ построения БД, основанный на объектно-атрибутном (ОА) подходе к организации вычислительного процесса и структур данных [4]. Для его иллюстрации покажем, каким образом можно реализовывать основные типы связей между сущностями ОО- и реляционной моделей БД, и приведем пример реализации небольшой БД.

1. Реализация основных типов сущностей и видов связей между сущностями в ОА-БД

Основой для проводимого анализа выберем UML-диаграмму [6], с помощью которой можно задать основные сущности ОО-модели: класс, атрибут класса, операции класса; а также следующие типы связей: зависимость, обобщение, множественное наследование, ассоциация, агрегация, композиция. Не следует забывать и о понятиях «кратность отношения» и «роль» (рис. 1). Теперь опишем методы реализации каждого типа отношений в ОА-БД. Мы будем применять ОА-язык – специализированный язык для описания ОА-графов [4].

Начнем с описания класса в ОА-БД. Итак, класс (шаблон, по которому создается объект) представляет собой набор полей и методов. В ОА-системе объект представляется в виде информационной капсулы (ИК); каждому полю объекта соответствует информационная пара (ИП), входящая в эту ИК. Имена атрибутов (индексу атрибута сопоставляется уникальная мнемоника) будут являться атрибутами ИП, в нагрузке ИП будет помещаться значение поля класса. Методы в объекте – это ссылка на подпрограмму. В ОА-системе программа представляет собой последовательность ИП (милликоманд). Эта последовательность может передаваться на функциональные устройства (ФУ), которые под управлением потока милликоманд производят определенные вычисления, преобразование данных и ввод/вывод. Опишем на ОА-языке класс на рис. 1 а: Человек{Пол=М
ДатаРожд=07.02.1970
ФамилияИмя=ИвановИван
ВыдатьВозраст= ВыдатьВозрастПрог
СохранитьТекущийДоход= СохранитьТекущДоходПрог
ВыдатьОбщийДоход=

¹ Исследование осуществлено в рамках Программы фундаментальных исследований НИУ ВШЭ в 2014 году

² Статья рекомендована к опубликованию в журнале "Информационные технологии"

ВыдатьОбщийДоходПрог}, где в нагрузках последних трех ИП находятся указатели на информационные капсулы (ИК) программ.



Рис. 1 – Основные типы связей в ОО-модели БД

Связь-обобщение (подклассы) реализуется с помощью добавления ИП в ИК описания объекта. Атрибут добавленной ИП идентифицирует подкласс, в нагрузке помещается указатель на ОА-граф, описывающий подкласс. Например, для рис. 1 б связь задается так: ЧеловекИзУниверситета{Студент= СтудентУк {...} Преподаватель=ПреподавательУк {...}}, где вместо многоточия помещается описание объекта, относящегося к подклассам Студент и Преподаватель; «Студент» и «Преподаватель» – мнемоники атрибутов. Мнемоники «СтудентУк» и «ПреподавательУк» – это указатели на капсулу с описанием студента и преподавателя соответственно. Наследование в ОА-БД осуществляется с помощью копирования ОА-графа описания объекта и добавления в корневую ИК ИП, описывающих новые поля и методы. Например, на рис. 1. б. представлено множественное наследование: класс «Студент-преподаватель» наследует свойства классов «Студент» и «Преподаватель». На ОА-языке такое наследование будет описано так: СтудентПреподаватель{СтудентУк ПреподавательУк}. Если в описании капсул в ОА-языке присутствует мнемоники указателя на ИК (но не в нагрузке ИП), то данная ИК (и весь ОА-граф, в который она входит) копируется в формируемую ИК. Таким образом, в ИК под именем «СтудентПреподаватель», будут скопированы все ИП из капсул «Студент» и «Преподаватель».

Связи-ассоциации задаются с помощью ИП, добавленной в ИК описания объекта: в нагрузке ИП находится указатель на ИК с описанием объекта, с которым происходит ассоциация, в атрибуте – мнемоника роли. В ОА-подходе связь один к одному задается с помощью двух ИП, ссылающихся на ИК описаний объектов. Например, между объектом класса «Человек» и «Университет» (рис. 1 в) существует связь, в которой «человек» выступает в роли «работник», а «Университет» в роли «Работодатель». На ОА-языке связь описывается следующим образом: Человек{ ... } Студент=Университет

Работник=Университет ... } Университет{ ... **Наниматель**= Человек **Обучающий**=Человек ... } Человек={**Студент**=Университет **Работник**=Университет}. Человек может работать в нескольких университетах, а университет нанимать несколько человек на работу (связь парной кратности), тогда ОА-граф будет выглядеть следующим образом: Человек1{... **Работник**=Университет1 **Работник**=Университет2 ...} Человек2{... **Работник**=Университет1 **Работник**=Университет2 ...} Университет1{... **Нанимает**= Человек1 **Нанимает**= Человек2 ...} Университет2{...**Нанимает**= Человек1 **Нанимает**= Человек 2...}. Как видно, название роли в данном случае выступает как имя атрибута.

Композицию (неотъемлемая часть объекта) можно реализовать, например, так: составить список атрибутов, в нагрузке которых имеется указатель на объекты, неотъемлемые от своего родителя. Тогда, если в удаляемой ИК встречаются ИП с такими атрибутами, то и капсулы, чей указатель хранится в нагрузке, также удаляются.

Сложные отношения и отношения-классы реализуются с помощью капсулы. Например, необходимо описать отношение «действие» (такая задача возникает в задаче составления БД, исходя из текста на естественном языке), в котором участвует достаточно много объектов: субъект (тот, кто производит действие), объект (то, на что направлено действие), посредник (например, инструмент, которым действие производят) и т.д. Например, такое отношение может выглядеть так: Действие{**Субъект**= {...} **Объект**= {...} **Посредник**= {...} **Адресат**= {...} **ОбразДействия**= {...}...}.

В заключение следует отметить, что ОА-подход к построению БД реализует все типы связей, существующие в ОО- и реляционном подходах.

2. Пример реализации БД по ОА-принципу

Для иллюстрации предложенного принципа реализуем БД, описанную с помощью UML-нотации, предложенной в [7]. На рис. 3 на ОА-языке («//» – знак комментария) представлено описание БД, формат которой задан на рис. 2. Компилятор ОА-языка входит в состав ОА-среды программирования и моделирования, которая позволяет моделировать вычислительный процесс в ОА-системе.

На листинге (рис. 3) приводится ОА-программа инициализации ОА-БД. В первой строке ОА-программы осуществляется создание функционального устройства (ФУ), обрабатывающего ОА-граф. Далее происходит инициализация мнемоник атрибутов и констант (знак «#» обозначает инициализацию константы, изолированная мнемоника – инициализация атрибута). Милликоманда (ИП, адресованная для ФУ) с мнемоникой «DB.Set» задает операцию «установить указатель на ОА-граф для ФУ «DB» – после знака «==», идет описание ОА-графа на ОА-языке. Знак «>» обозначает начало новой записи в списке (список организуется из сущностей «Position», этот список «сшивает» все записи в единую БД). После инициализации с помощью ФУ «DB» можно производить модификацию ОА-графа и поиск информации в нем. ОА-граф, созданный в среде ОА-программирования и моделирования, представлен на рис. 4 (для вывода ОА-графа в среде ОА-программирования и моделирования реализован специальный программный компонент). На рисунке в фигурных скобках помещаются индексы информационных капсул (ИК): т.к. ОА-граф представляет собой граф произвольной топологии, то на экране отображается только его остов, а по индексам можно отследить все связи между вершинами, не вошедшими в остов, – после знака «==» (обозначение нагрузки ИП) указывается индекс той ИК, на которую указывается ссылка в нагрузке (например, «ID_14»).

3. Преимущества и недостатки ОА-подхода к организации БД

ОА-подход обеспечивает такие ключевые качества БД как: описание любой онтологии, масштабируемость вычислений, целостность БД, эргономичность (запросы вводятся на

естественном языке) [8], высокая скорость поиска в БД (для ОА-БД разработана методика поиска с применением индексации ИК).

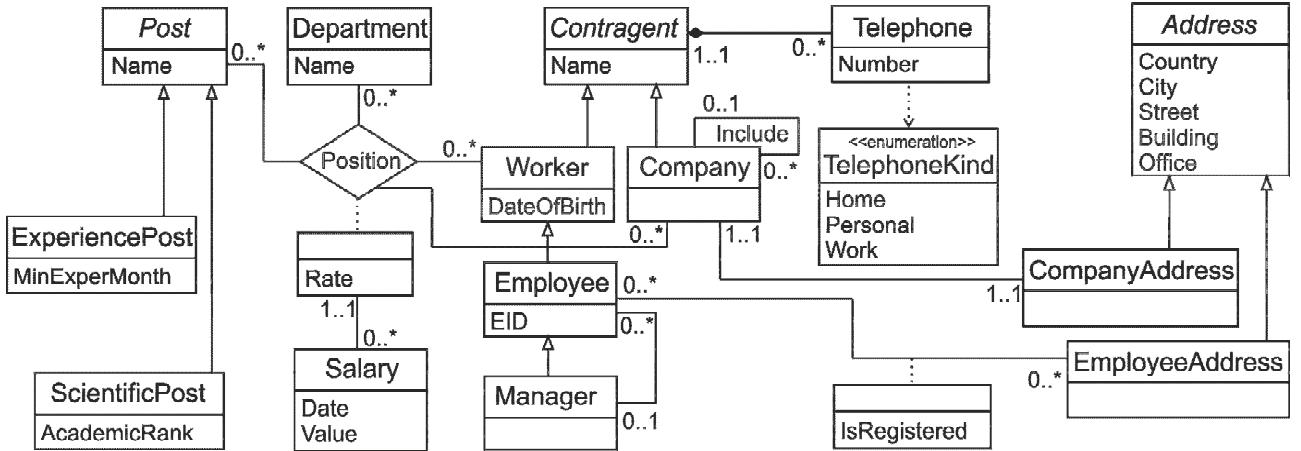


Рис. 2 – Структура тестовой базы данных

```

NewFU={Mnemo="DB" FUType=FUGraph}
// Инициализация атрибутов
Department Post
ExperiencePost MinExperMonth
ScientificRank AcademicRank
Position Rate Salary Date
Value Worker DateOfBirth
Company Contragent Employee
EID Manager Include Telephone
Number TelephoneKind
Home#1 Personal#2 Work#3
Name Adress Country
City Street Building
Office CompanyAdress
EmployeeAdress IsRegistered

DB.Set=
>Position1{ // Position1 -метка ИК
  Post={Name="Full professor"
  ExperiencePost={MinExperMonth=12}}
  Department={
    Name="Computer science"
  Department={
    Name="Applied mathematics"
  Salary={Date="1.10.2014"
  Value=4000}

Worker={
  DateOfBirth="13.04.1968"
  Telephone={Number=1234567
  TelephoneKind=Personal}
  Telephone={Number=7654321
  TelephoneKind=Home}
}
Employee=Employee1{// Метка
EID=12876 Manager=null
EmployeeAdress=
  {IsRegistred=true
  Adress={Country="USA"
  City="New York" Street="Street
  20"
  Building=10 Office=1}}
Company=
  >{Name="IBM"
  Telephone={Number=1111111
  TelephoneKind=Personal}
  Telephone={Number=2222222
  TelephoneKind=Work}
  CompanyAdress={Country="USA"
  City="New York" Street="Street
  10" Building=10 Office=1}
  }}
>Position2{...// Вторая запись БД
  
```

Рис. 3 – Листинг ОА-БД с форматом данных, представленным на рис. 2.

ОА-БД относится к классу сетевых (графовых), когда описание онтологии представляет собой множество вершин (семов), обозначающих объекты и дуги, задающие связи между объектами. Сетевая БД не имеет ограничений на топологию, и поэтому потенциально имеет возможность описывать любую онтологию. Этим она качественно отличается от реляционной и ОО (фреймовой) моделей: в первой из них практически невозможно описать онтологию с большим числом типов связей (для каждого типа необходимо создавать собственное отношение (таблицу), что очень громоздко), у второго существует ограничение на топологию (граф типа «дерево»).

Масштабируемость ОА-БД достигается благодаря применению управления вычислениями с помощью потока данных (dataflow) [9]. Dataflow позволяет «отвязать» адресацию данных от оперативной памяти: доступ к данным производится через

функциональные устройства (ФУ), которые являются устройствами виртуальными, т.е. жестко не привязанными к аппаратуре вычислительных узлов. Совокупность ФУ формирует так называемый ОА-образ – виртуальную систему, производящую вычисления. Затем ОА-образ «накладывается» на конкретную вычислительную систему (ВС) путем распределения ФУ по вычислительным узлам ВС и настройки самих ФУ и маршрутизаторов, ответственных за пересылку информационных пар (ИП) между вычислительными узлами. При изменении конфигурации ВС необходимо только произвести перераспределение ФУ между вычислительными узлами и произвести новые настройки [4].

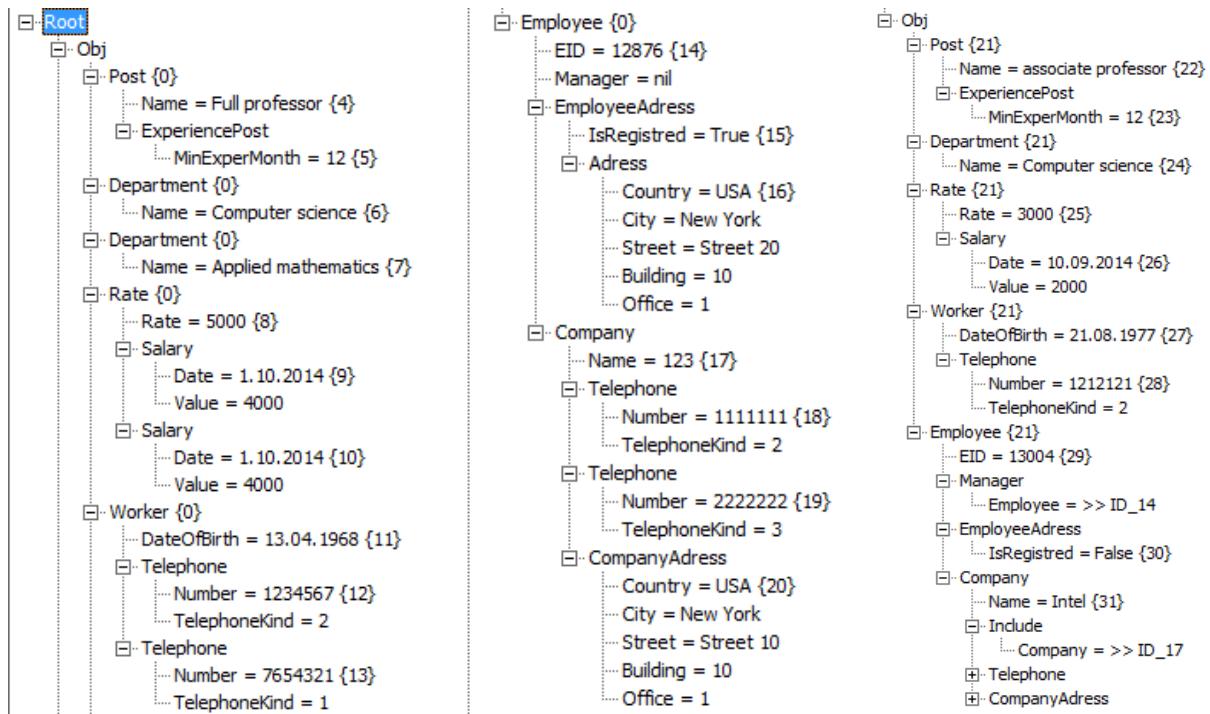


Рис. 4 – БД в виде ОА-графа

Целостность ОА-БД обеспечивается благодаря формату и технологии обработки ОА-графа. Так, модификация ОА-БД заключается в добавлении или удалении из ИК, составляющих ОА-граф, некоторых ИП. Допустим (рис. 5), в ИК имеется ИП с атрибутом «1», затем мы добавили туда же ИП с атрибутом «2». И теперь уже можно производить поиск данных либо по атрибуту «1», либо по атрибуту «2» независимо: если мы ищем данные с атрибутом «1», то атрибут «2» мы игнорируем, и наоборот. Таким образом, с помощью анализа атрибутов ИП можно в одном ОА-графе выделять несколько подграфов... Т.е. мы можем менять структуру БД без опасения за нарушения ее целостности. В результате, отпадает необходимость в рефакторинге, без чего практически нельзя обойтись при серьезной модификации ОО-БД.

Эргономичность обеспечивается благодаря тому, что запросы к ОА-БД выполняются на естественном языке: запрос преобразуется в ОА-граф запроса и далее производится определение, является ли запрос подграфом ОА-графа БД. О технологии преобразования текста не естественном языке в ОА-граф рассказывается в [8]. Ответы на запросы так же могут выводиться на естественном языке.

Подобно ОО-подходу, ОА-БД обеспечивает широкие возможности абстракции данных. Например, ОА-граф можно выстроить по топологии «дерево»: тогда ИК высшего уровня, как бы, инкапсулирует информационную структуру под ней (в ИК помещаются основные характеристики нижележащего уровня). Если ОА-граф имеет свободную топологию, то в нем можно выделить подграфы (например, по принципу описания какого-либо объекта, множества объектов или явления) и включить в ОА-граф ИК, описывающие основные характеристики таких подграфов.

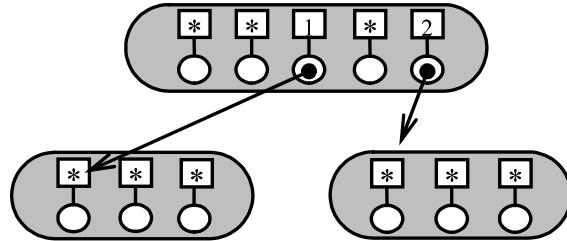


Рис. 5 – Обеспечение целостности ОА-БД при ее модификации

В ОА-БД имеется возможность производить достаточно быстрый поиск информации, благодаря индексации ИК. Как уже говорилось ранее, поиск в ОА-БД заключается в отыскании подграфа (подграфов) в ОА-графе БД, совпадающих с графом-запросом [8]. Процесс поиска упрощается благодаря тому, что вершины ОА-графа как бы помечены с помощью ИП, расположенных в ИК. Поэтому можно предложить следующую методику поиска подграфа, который состоит из нескольких этапов. Первый – индексация всех вершин ОА-графа БД и запроса. Вторая – составление двух списков атрибутов ИП: первый – атрибутов, встречающихся в ОА-графе БД; второй – в ОА-графе запроса. Если какие-то атрибуты из запроса не присутствуют в БД, то поиск прекращается и пользователю выдается сообщение об отрицательном результате. Иначе в графе БД выделяются все ИК, где встречаются атрибуты из графа-запроса. Третий – группировка найденных в ОА-графе ИП по капсулам: если найдены ИП, у которых имеется одинаковый индекс ИК (т.е. они находятся в одной капсуле), то они объединяются в группу. Четвертый – сопоставление групп: если группа не совпадает ни с одной группой из графа-запроса, то она удаляется. Если в результате все группы были удалены или количество групп стало меньше, чем в графе-запросе; то поиск завершается с отрицательным результатом. Пятый – сопоставление оставшихся групп с ИК графа-запроса. Для ускорения поиска нужных атрибутов можно использовать хеширование, которое значительно повышает скорость расчетов.

В заключении следует сказать, что теоретически ОА-СУБД имеет достаточно хорошие перспективы, т.к. обладает множеством положительных качеств СУБД. Однако следует отметить и ее недостатки: во-первых, чрезмерный расход оперативной памяти (в ОА-БД для каждой связи выделяются две ИП, т.к. дуги ОА-графа должны быть двунаправленными); во-вторых, благодаря возможности добавления новых структур данных без нарушения старых в ОА-БД могут возникнуть «сопли» – структуры данных, которые уже никому не нужны и о которых разработчики уже давно забыли, но которые занимают оперативную память и замедляют поиск нужной информации.

Литература

1. Дейт, К. Дж. Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом "Вильямс", 2005
2. Леонид Черняк. Смутное время СУБД // Открытые системы, №2,2012 URL: <http://www.osp.ru/os/2012/02/13014107/>
3. Gabriel, R. Objects Have Failed: Notes for a Debate. (retrieved 17 May 2009). <http://www.dreamsongs.com/Files/ObjectsHaveFailed.pdf>
4. Салибекян С.М, Панфилов П.Б. ОА-архитектура построения и моделирования распределенных систем автоматизации // Автоматизация в промышленности. №11, 2011
5. Сайт, посвященный работам в области ОА-архитектуры. URL: dataflow.miem.edu.ru
6. Новиков Ф.А., Иванов Д.Ю. Моделирование на UML. Теория, практика, видеокурс. – СПб.: Профессиональная литература, Наука и Техника, 2010. – 640 с.: ил. + цв. Вклейки (+ 2 DVD)
7. Олейник П.П. Унифицированная модель тестирования инструментов разработки объектно-ориентированных приложений // Объектные системы – 2014 (Зимняя сессия): материал VII Международной научно-практической конференции (Ростов-на-Дону, 10-12 декабря 2014 г.) / Под общ. ред. П.П. Олейника. – Ростов-на-Дону: ШИ (ф) ЮРГПУ (НПИ) им. М.И. Платова, 2014. С. 23-32.

8. Салибекян С.М., Халькина С.Б., Тиновицкий К.Д. Объектно-атрибутный подход для семантического анализа естественного языка. // Объектные системы – 2014: материал VI Международной научно-практической конференции (Ростов-на-Дону, 10-12 мая 2014 г.) / Под общ. ред. П.П. Олейника. – Ростов-на-Дону: ШИ ЮРГТУ (НПИ), 2014. – С. 80-86
9. Jurij Silk, Borut Robic and Theo Ungerer «Asynchrony in parallel computing: From dataflow to multithreading» Institut Jozef Stefan, Technical Report CDS-97-4, September 1997.

УДК 004.43:378.09

УСТРОЙСТВО ОБЪЕКТНО-ОРИЕНТИРОВАННЫХ БАЗ ДАННЫХ

Мясникова Нелли Александровна, доцент, Южно-Российский государственный политехнический университет (Новочеркасский политехнический институт) имени М.И. Платова, Россия,
Новочеркасск, mnela@list.ru

Курин Николай Дмитриевич, студент, Южно-Российский государственный политехнический университет (Новочеркасский политехнический институт) имени М.И. Платова, Россия,
Новочеркасск, kurin.kolyan@yandex.ru

Первая информация об объектно-ориентированных базах данных (ООБД) появилась в середине 80-х годов. Возникновение ООБД было обусловлено, в первую очередь, потребностями их практического использования: необходимостью разработки все более сложных информационных прикладных систем, для которых технологии предшествующих систем баз данных были не очень удобны. В наше время ООБД используются практически во всех областях информационных технологий, начиная от операционных систем и заканчивая системами управления различных предприятий.

Главная особенность объектно-ориентированных баз данных в том, что данные внутри таких баз данных моделируются в виде объектов, их атрибутов, методов и классов. ООБД, в основном применяются в случаях, когда требуется высокая производительность при обработке данных, имеющих сложную структуру [4].

В статье представлены устройство и способ создания простейшей ООБД. В качестве предметной области использован магазин игрушек.

Объектно-ориентированные базы данных реализуются в объектно-ориентированных системах управления базами данных (ООСУБД). У этой системы есть некоторые требования, которые необходимо учесть при проектировании. В манифесте ООБД выделены обязательные характеристики, которым должна отвечать любая объектно-ориентированная база данных. Их выбор основан на двух основных критериях: система должна быть объектно-ориентированной и представлять собой базу данных. Основными характеристиками являются [1]:

1. Поддержка индивидуальности объектов.
2. Поддержка инкапсуляции.
3. Поддержка типов и классов.
4. Поддержка наследования типов от классов и их предков.

Рассмотрим подробнее каждую из выделенных характеристик.

Поддержка индивидуальности объектов означает, что каждый объект должен иметь свой уникальный идентификатор, который не должен зависеть от значений атрибутов этого объекта.

Поддержка инкапсуляции достигается за счет того, что пользователь обладает правами доступа только к некоторым методам, реализующим интерфейс, а данные и внутренняя реализация методов скрыты внутри объектов базы данных.

Поддержка типов и классов. Требуется, чтобы в ООБД поддерживалось хотя бы небольшое различие между типами и классами. Термин «тип» более соответствует понятию абстрактного типа данных. Информация о типе данных может использоваться для проверки