InterSystems CACHÉ
Make Applications Faster

Работа с существующими глобалами через объекты и SQL

Вадим Федоров
InterSystems Corporation

Содержание

Обзор стратегий хранения

CacheStorage

CacheSQLStorage

CustomStorage

Пример CacheSQLStorage

InterSystems CACHÉ
Make Applications Faster

Сравнение стратегий хранений

	CacheStorage	CacheSQLStorage	CustomStorage
SQL	✓	✓	
Objects	✓	✓	

✓ Обеспечивается Caché

 Реализуется разработчиком

InterSystems CACHÉ
Make Applications Faster

Выбираем стратегию хранения

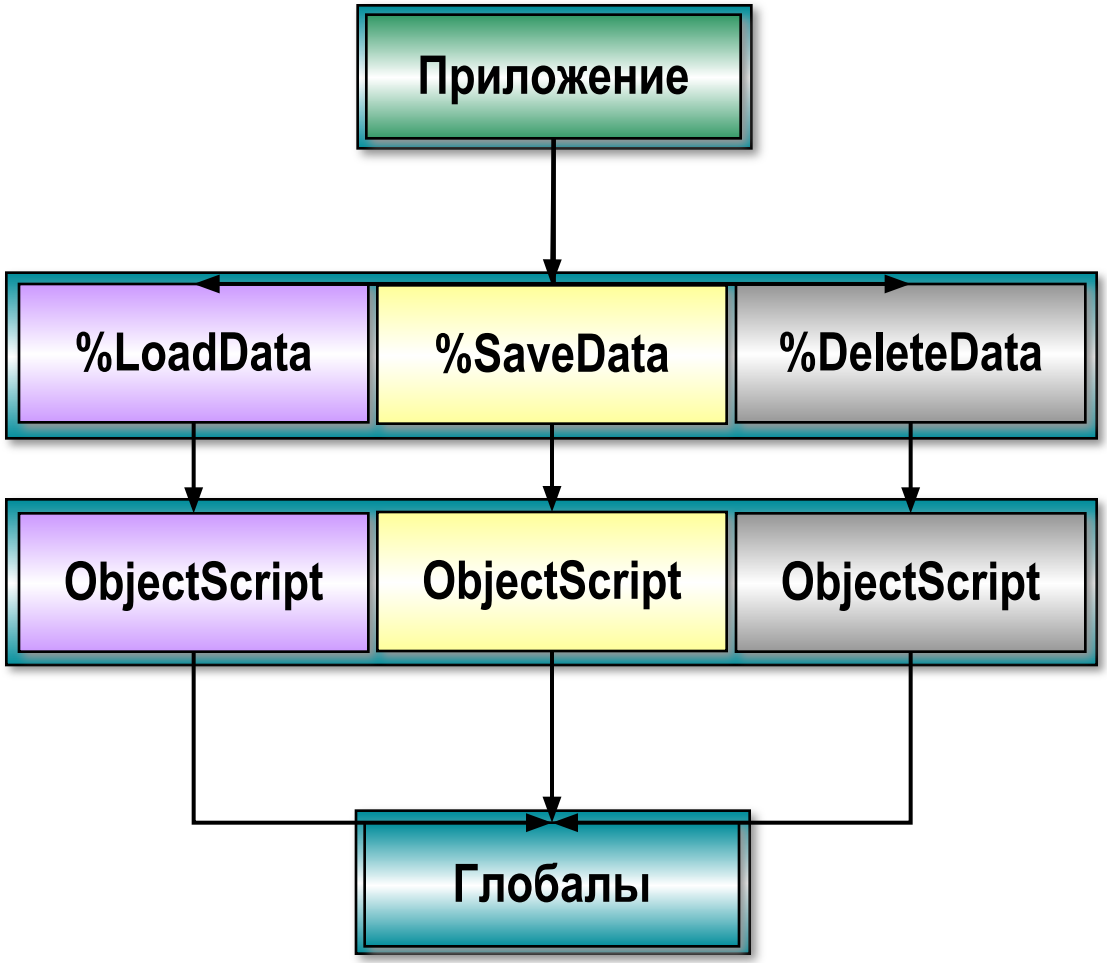
- CacheStorage идеально подходит для новых приложений
- CacheSQLStorage применяется, когда с существующими глобалами можно и нужно работать с помощью SQL
- CustomStorage используется тогда, когда нельзя работать с глобалами через SQL и нужно реализовывать собственную сложную логику для обеспечения объектного доступа

Содержание

- Обзор стратегий хранения
- CacheStorage
- CacheSQLStorage
- CustomStorage
- Пример CacheSQLStorage

InterSystems CACHÉ
Make Applications Faster

Обзор CacheStorage



Объектное API

Объектная реализация

InterSystems CACHÉ
Make Applications Faster

Информация о CacheStorage

- CacheStorage генерирует глобалы, в которых используется $ListBuild
- Уникальный идентификатор (IDKey / PrimaryKey) может автоматически сгенерирован Cache или задан разработчиком вручную
  - Это влияет на структуру глобалов
- Можно управлять хранением свойств классов в глобалах
  - Это влияет на структуру значений глобалов

Содержание

Обзор стратегий хранения

CacheStorage

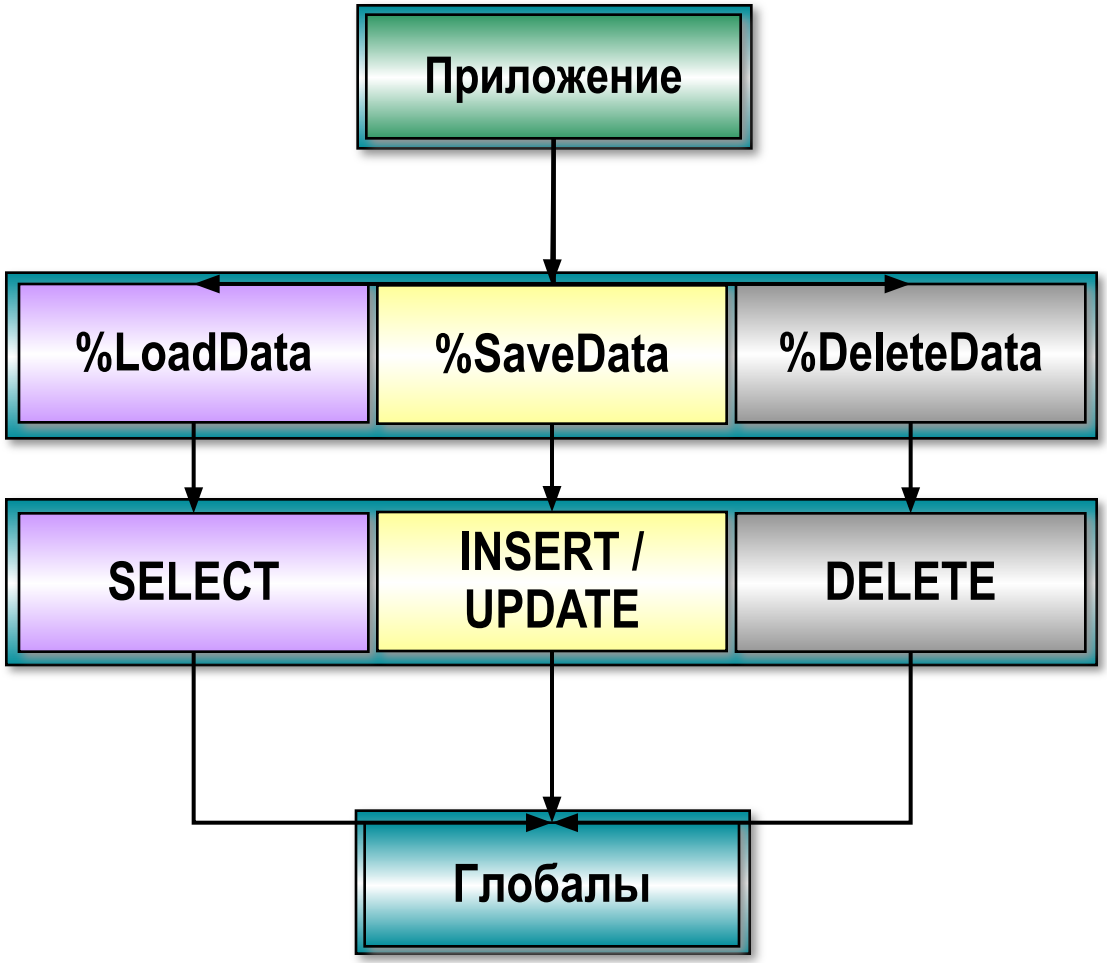
CacheSQLStorage

CustomStorage

Пример CacheSQLStorage

InterSystems CACHÉ
Make Applications Faster

Обзор CacheSQLStorage



Объектное API

SQL Реализация

InterSystems CACHÉ
Make Applications Faster

Обзор CacheSQLStorage

- Создайте Persistent-класс
- Добавьте свойства в класс
- Определите свойство (свойства), которое будет идентификатором класса (IDKey / Primary Key)
- Создайте стратегию хранения выставив соответствие между свойствами класса и данными глобала

Особенности CacheSQLStorage

- При работе через объекты будут вызываться триггеры (при использовании CacheStorage они не вызываются)!
- Поддерживается ссылка и Parent-Child отношение для связи классов

Создание CacheSQLStorage

- CacheSQLStorage обычно создается:
  - Программистом
  - Программой конвертации из F-DBMS
  - Программой конвертации из KB-SQL

Обеспечение SQL-доступа

- Не ко всем структурам глобалов можно настроить CacheSQLStorage так чтобы обеспечить полный SQL-доступ (Read/Write/Delete)
- Некоторые структуры подходят только для чтения через SQL (SELECT)
- Чтобы обеспечитьобновление иногда необходима дополнительная работа, кроме установки Mapping

Виды карт

- Различают следующие виды CacheSQLStorage карт:
  - Данные (MasterMap): Должны быть определены все поля
  - Индексы: Должны быть определены некоторые поля
    - Full (по умолчанию): Все данные попадают в индексы
    - Conditional: Данные попадают в индексы, если выполняется
    - Nonnull: Нулевые значения (Null values) не попадают в индексы

ID и Primary Key

- IDKey индексы определяют уникальные идентификаторы для объектов
- Primary Key индексы определяют уникальные идентификаторы для SQL
- IDKey и Primary Key индексы обычно строятся по одним и тем же полям

Индексы глобалов (Subscripts)

- Индексы (subscripts) карт обычно эквивалентны индексам глобалов
- Индексы карт используется для формирования кода, перебирающего записи таблицы.
- Поддерживаются следующие типы индексов (subscripts) :

  Sub: Основанный на «стандартном» индексе глобалов

  Piece: Основанный на определенной позиции (используется разделитель)

  Global: Основанный на данных, хранящихся в нескольких глобалах

  Other: Основанный на пользовательском коде

InterSystems CACHÉ
Make Applications Faster

RowID

- Существует тесная связь между IDKey и RowID для карт данных, но не для карт индексов
- A RowID используется для уникальной идентификации данных в глобале, на основе индекса глобала, определенного в карте
- Например, для глобала:

^Person(PersonID,“Cars”,CarID)=“Make^Model^Year”

2 поля будут нашими RowID:

PersonID: хранится на первом уровне {L1}

CarID: хранится на третьем уровне {L3}

Карты данных

- Когда определены индексы глобала, нужно определить хранение свойств класса в глобале
- Можно определить дополнительные узлы индекса глобала (только литералы)
- Можно использовать $Piece или $ListBuild

Подробности редактирования карт

- Map Name: Имя карты.
- Map Type: Данные или Индексы (Data or Index)
- Global Name: Имя глобала (^…) или локального массива
- Node Structure: Структура узла: $Piece или $List
- Population Type: Тип заполнения
- Population %: Оценка предполагаемого количества рядов в индексе
- Condition: Выражение определяющее условие. Например, {Name}‘=“”
- Conditional Fields: Поля, по которым будет проверяться условие
- Conditional with hostvars: Булева значение, которое влияет на использование индекса кэшированными запросами
- Row Reference: Позволяет программистам переопределить сгенерированный RowID

Подробности редактирования индексов глобалов

- Access Type: Тип доступа. Sub, Piece, Global или Other
- Delimiter: Разделитель. Используется, если тип доступа Piece
- Expression: Выражение. Обычно {поле}, “string” или число, или определенная позиция
- Loop Init Value: Не включаемое значение, используемое для генерации кода обхода
- Start Value: Включаемое значение, используемое в сгенерированном коде обхода
- Stop Value: Значение, при котором обход останавливается
- Stop Expression: Выражение, при котором обход останавливается, например, {L1}>200
- Data Access: Доступ к данным. Переопределяет контекст текущего выражения для вычисления значения текущего уровня доступа (Override the context of the current access-level’s value expression)
- Next Code: Используется программистом для переопределения генерируемого кода обхода
- Invalid Conditions: Выражения, используемые для исключения рядов из карты. Например, {L1}<1
- Access Variables: Переменные, используемые программистом, для обеспечения обеспечения уникальности имен

InterSystems CACHÉ
Make Applications Faster

Подробности редактирования RowID

- RowID: Позиция поля в спецификации RowID
- Field: Имя поля, состовляющего часть RowID
- Expression: Уровень внутри определения индексов глобала (subscript definition). Например, {L2} или {L6}

Подробности редактирования данных

- Field: Имя поля
- Node: Дополнительный индекс глобала (только литерал), где находится поле
- Piece: Позиция в строке $Piece
- Delimiter: Разделитель. Например, “^” или $c(1)

Содержание

Обзор стратегий хранения

CacheStorage

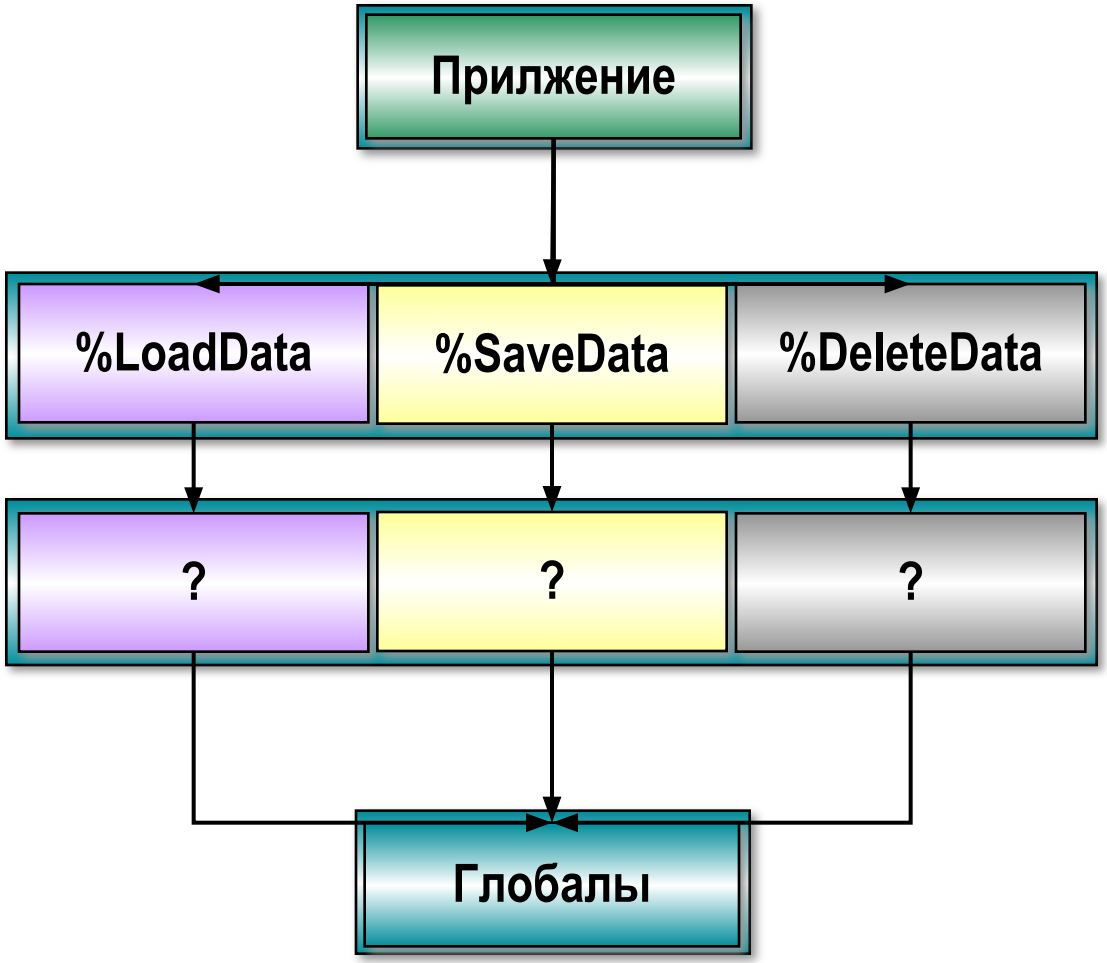
CacheSQLStorage

CustomStorage

Пример CacheSQLStorage

InterSystems CACHÉ
Make Applications Faster

Обзор CustomStorage



Объектные API

Собственная реализация

InterSystems CACHÉ
Make Applications Faster

Обзор CustomStorage

- Создайте Persistent-класс
- Добавьте свойства в класс
- Определите свойство (свойства), которое будет идентификатором класса (IDKey / Primary Key)
- Создайте стратегию хранения выставив соответствие между свойствами класса и данными глобала
- Реализуйте код доступа к объектам: %LoadData, %SaveData, %DeleteData

CustomStorage и SQL

- Для того чтобы использовать SQL с CustomStorage, **необходимо** определить специальный параметр класса:

  Parameter SQLENABLED = 1;

- Mapping the SQL portion with CustomStorage is identical to the methods used for CacheSQLStorage

CustomStorage и объекты

- Для того чтобы использовать объекты с CustomStorage, **необходимо** выполнить следующее:
  - Реализовать %LoadData, %SaveData, %DeleteData
  - В Вашем коде Вы должны управлять :
    - ID объектов на диске и в памяти (с помощью метода %IdSet())
    - Переменными экземпляров свойств (имена свойств имеют первые символы “i%”)
    - Concurrency
    - Уникальностью данных
    - Ограничениями, накладываемыми внешних ключей (Foreign key constraints)

%LoadData

- Код, реализованный в %LoadData(), будет выполняться каждый раз, когда загружается объект, обычно после вызова %Open() и %OpenId()
- Пример %LoadData:

```
Method %LoadData(id As %Library.String) As %Library.Status
    {
        Set i%SSN = id
        Set i%Name = $Piece(^P(id),“^”,1)
        Set i%DOB = $Piece(^P(id),“^”,2)

        Quit $$$OK
    }
```

%SaveData

- Код, реализованный в %SaveData(), будет выполняться каждый раз, когда сохраняется объект, в результате вызова метода %Save()
- Пример %SaveData:

```
Method %SaveData(id As %Library.String) As %Library.Status
    {
        Lock ^P(id):5 If '$Test Quit $$$ERROR($$$LockFailedToAcquireExclusive)

        Set id = i%SSN
        Do ..%IdSet(id)

        Set $Piece(^P(id),"^",1) = i%Name
        Set $Piece(^P(id),"^",2) = i%DOB

        Quit $$$OK
    }
```

InterSystems CACHÉ
Make Applications Faster

%DeleteData

- Код, реализованный в %DeleteData, будет выполняться каждый раз, когда объект будет удаляться, в результате вызова %Delete() или %DeleteId()

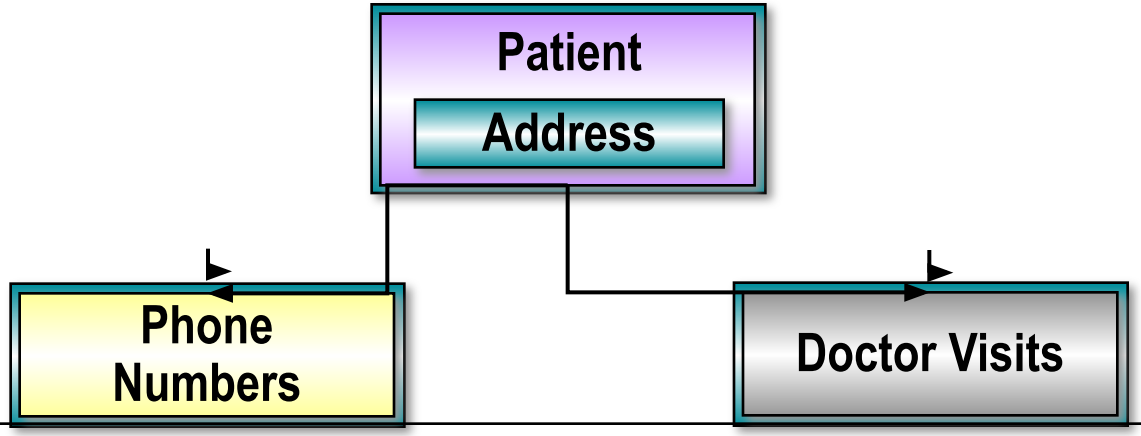
- Пример %DeleteData:

```
Method %DeleteData(id As %String, concurrency as %Integer) As %Status
    {
        Lock ^P(id):5 If '$Test Quit $$$ERROR($$$LockFailedToAcquireExclusive)

        Kill ^P(id)

        Quit $$$OK
    }
```

Содержание

- Обзор стратегий хранения
- CacheStorage
- CacheSQLStorage
- CustomStorage
- Пример CacheSQLStorage

Пример модели данных



- Есть два отношения Parent-Children:
  - Пациент может иметь несколько номеров телефона
  - Пациент может посещать доктора несколько раз
- Удаление пациента удаляет его номера телефонов и визиты к врачу

Пример структуры данных глобала

^P(SSN) = “Name^DOB^Phone1~Phone2~...~PhoneN^Company”

^P(SSN,“Address”) = “City^PostalCode^Country”

^P(SSN,“Visits”,VisitDate,VisitTime) = “Symptom^Payment”

^P(“211-22-1222”) = “Smith,John^39873^718-317-3312~917-225-2213^AT&T”

^P(“211-22-1222”,“Address”) = “New York^10312^USA”

^P(“211-22-1222”,“Visits”,58809,43200) = “Cough^15.00”

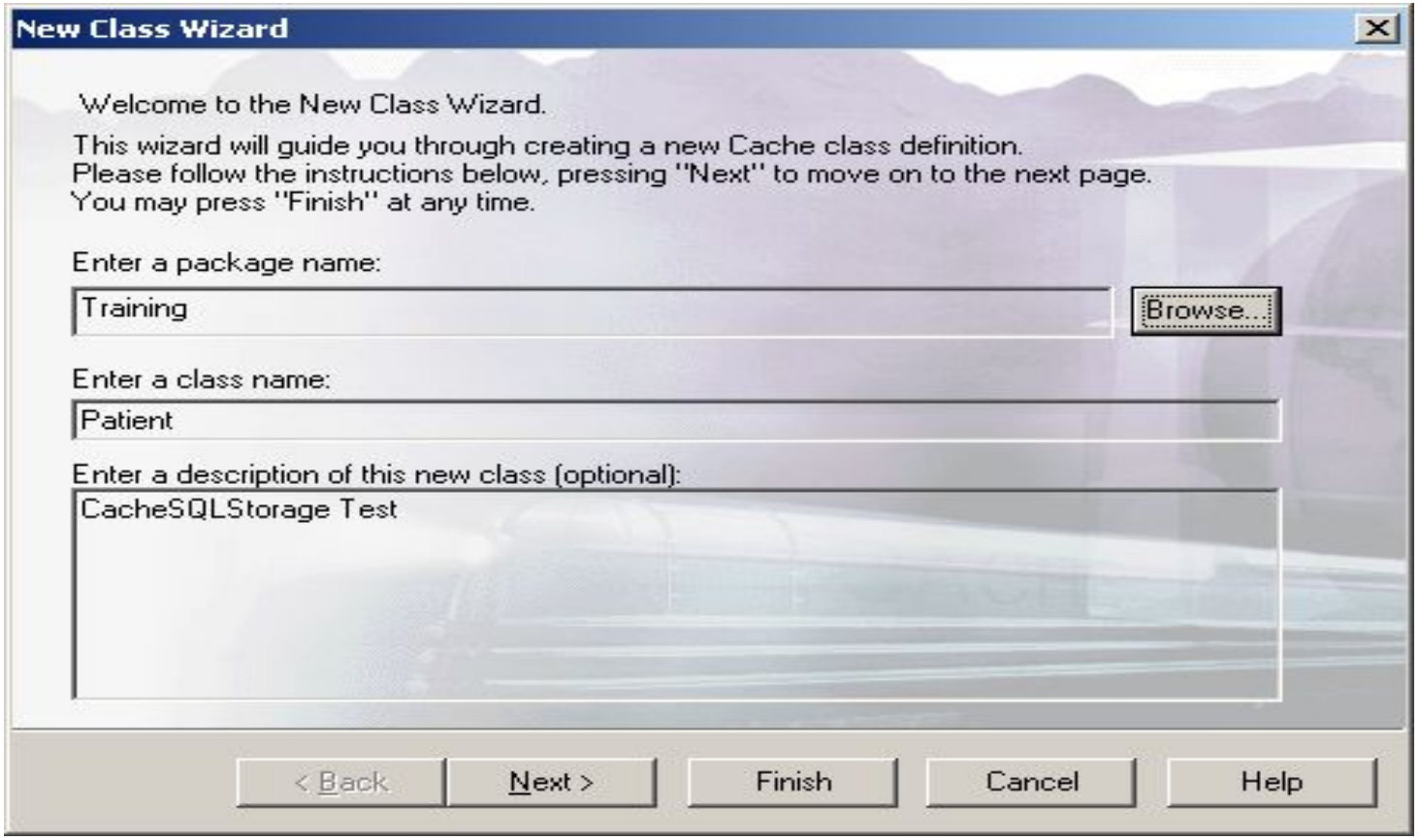
^P(“211-22-1222”,“Visits”,58820,57900) = “Sore Throat^50.00”

Пример структуры индексов глобала

^PI(Name,SSN) = “”

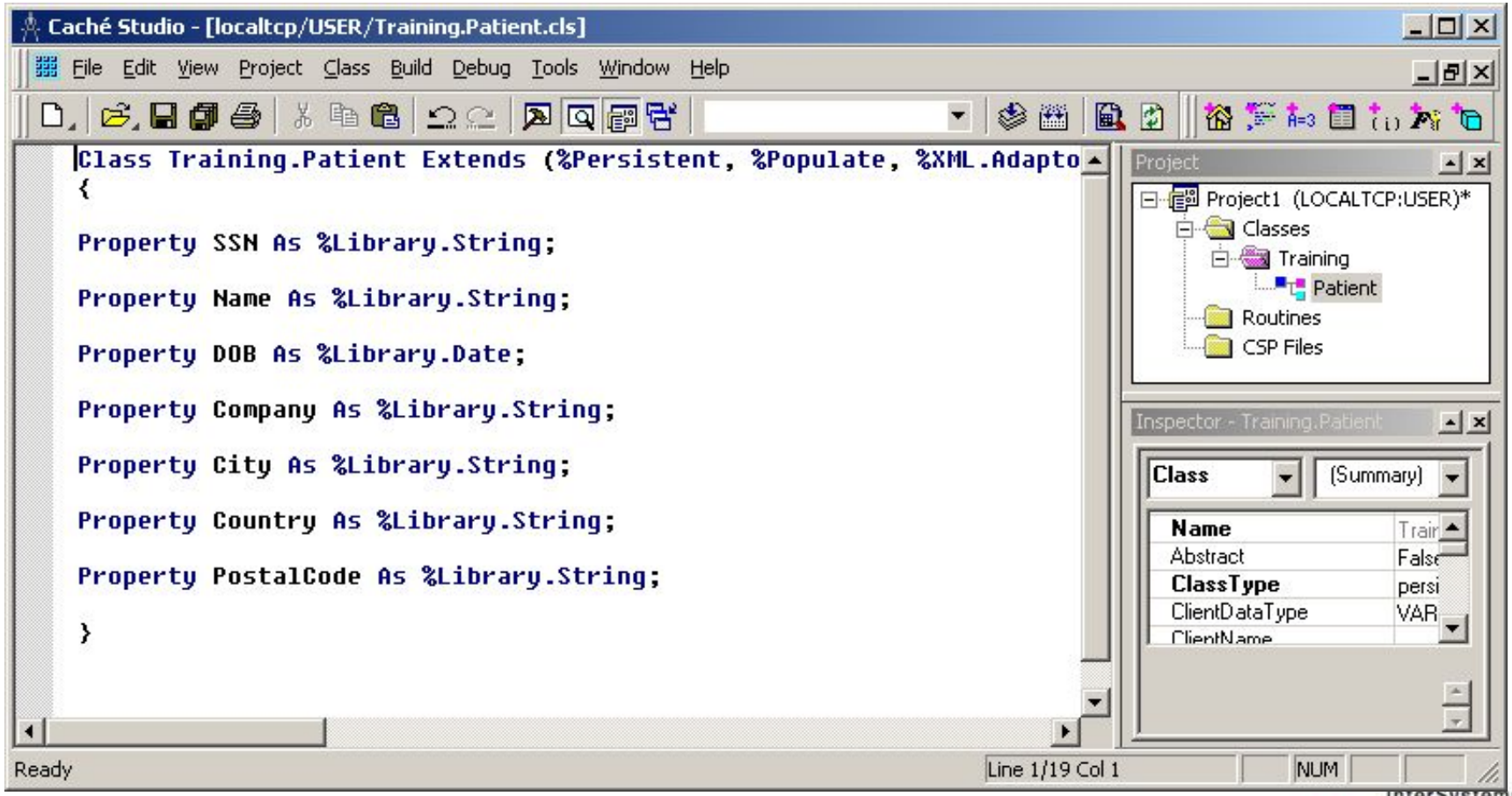
^PI(“Smith,John”,“211-22-1222”) = “”

Создаем Persistent-класс



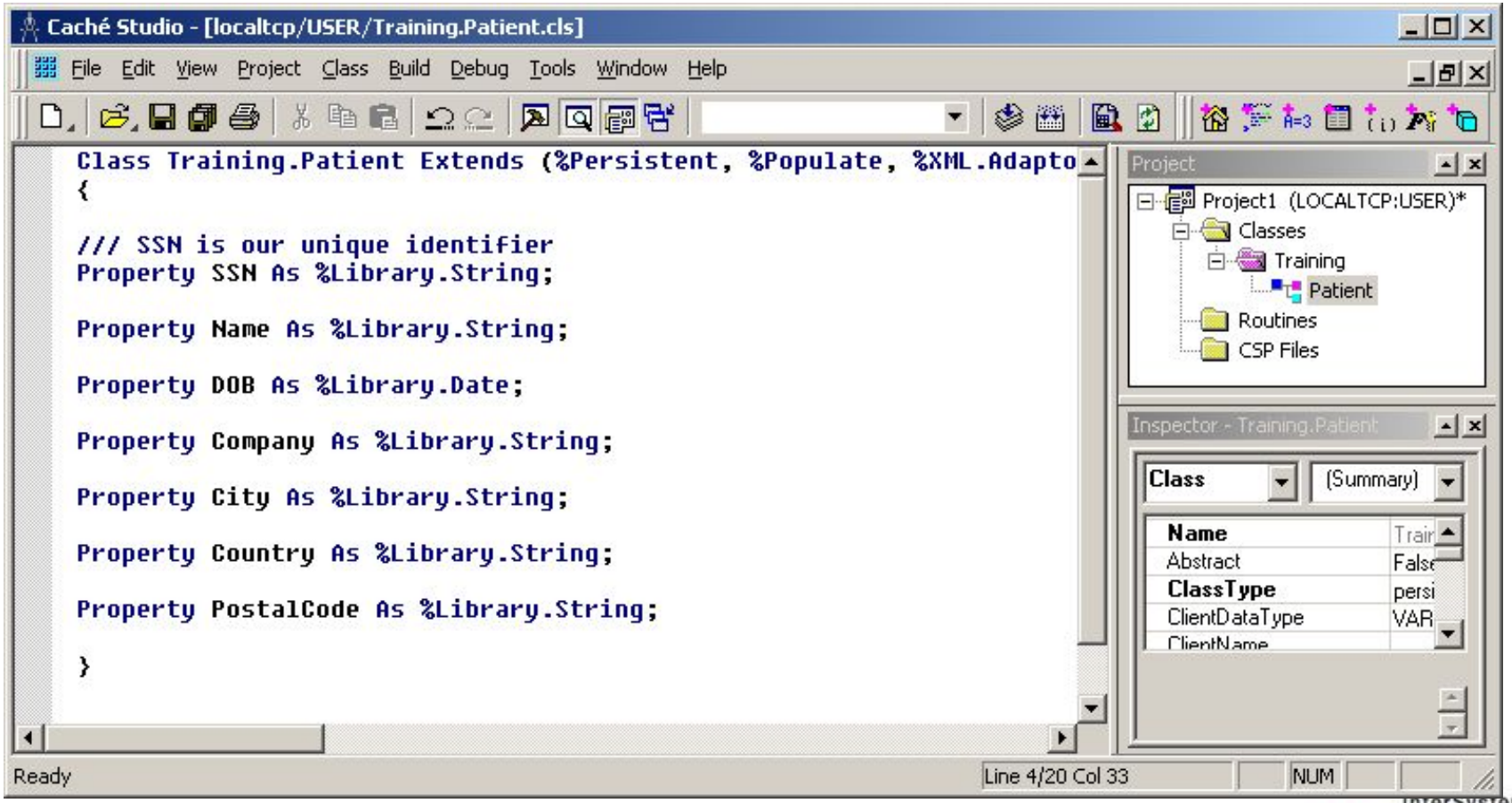
InterSystems CACHÉ
Make Applications Faster

Добавляем свойства



InterSystems CACHÉ
Make Applications Faster

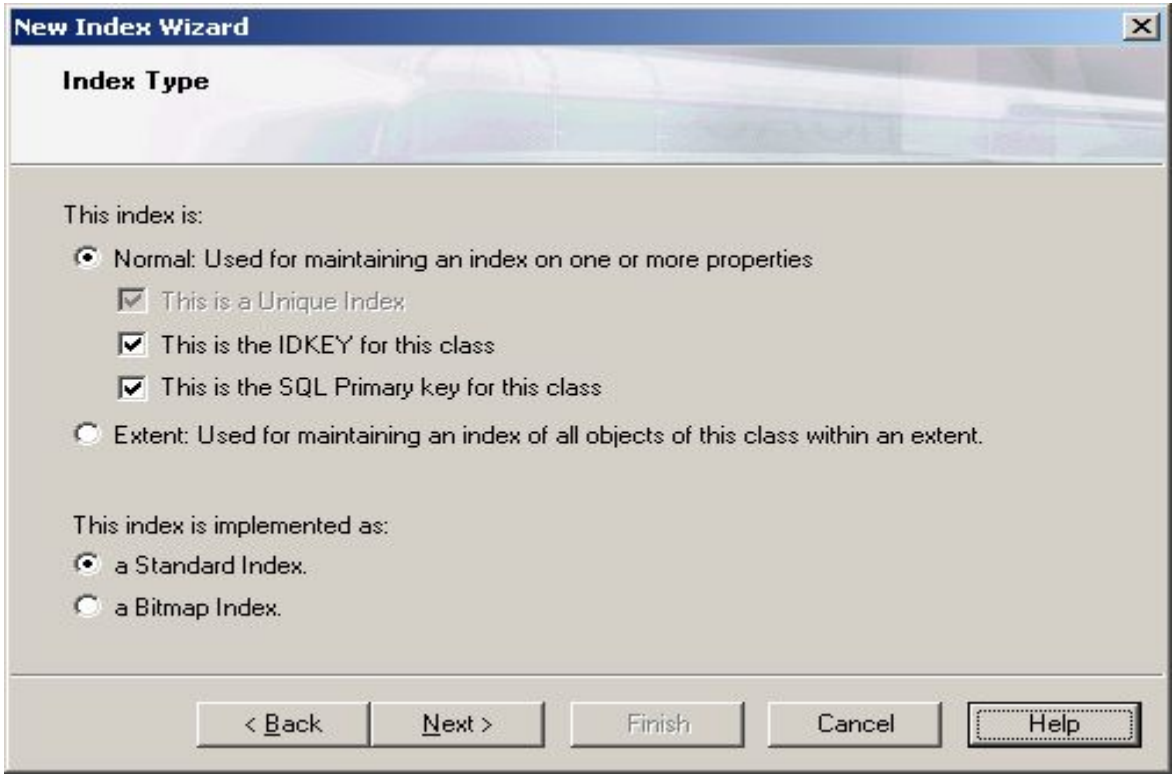
Выбираем уникальный идентификатор



- Базируется на одном поле: SSN

InterSystems CACHÉ
Make Applications Faster

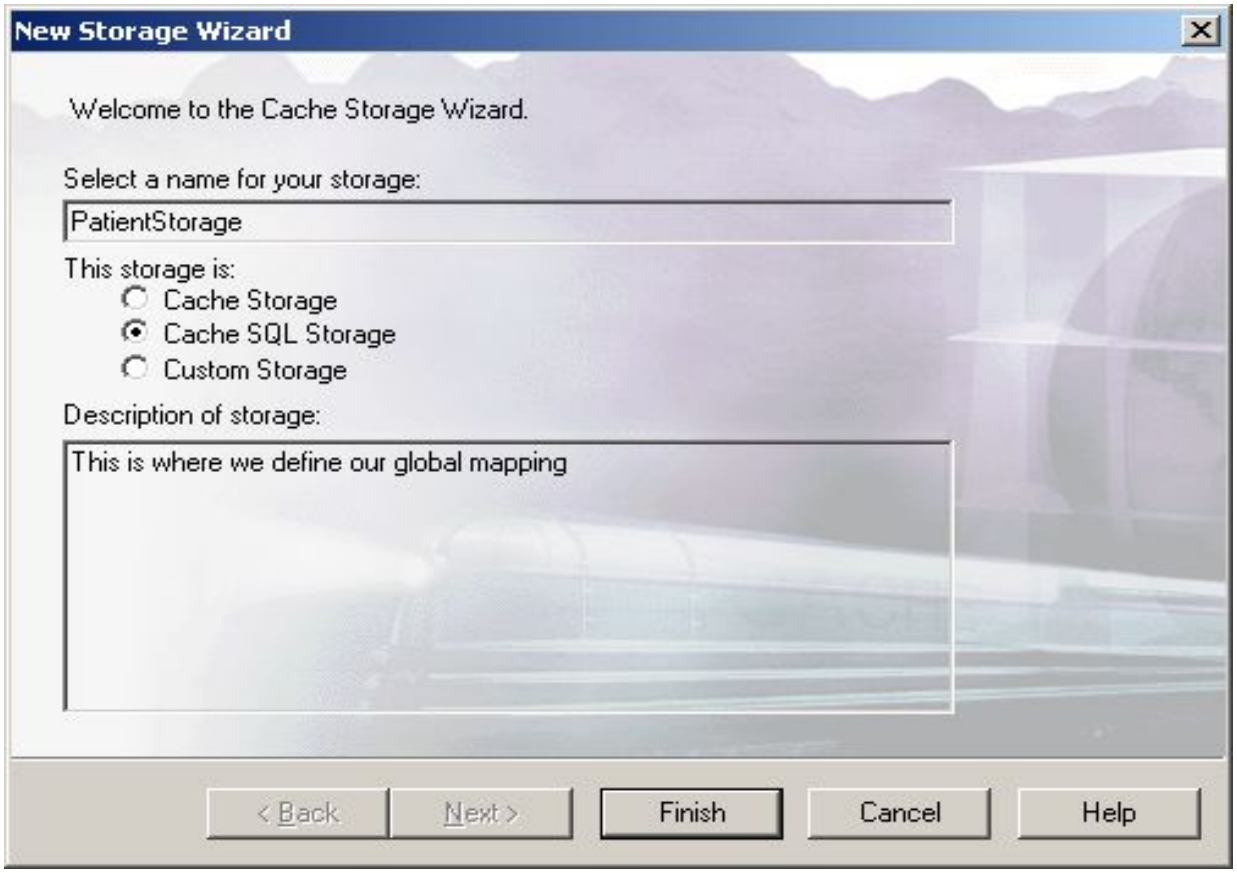
Определяем ID / Primary Key индекс



- Основан на свойстве SSN
- Не изменяйте collation индекса

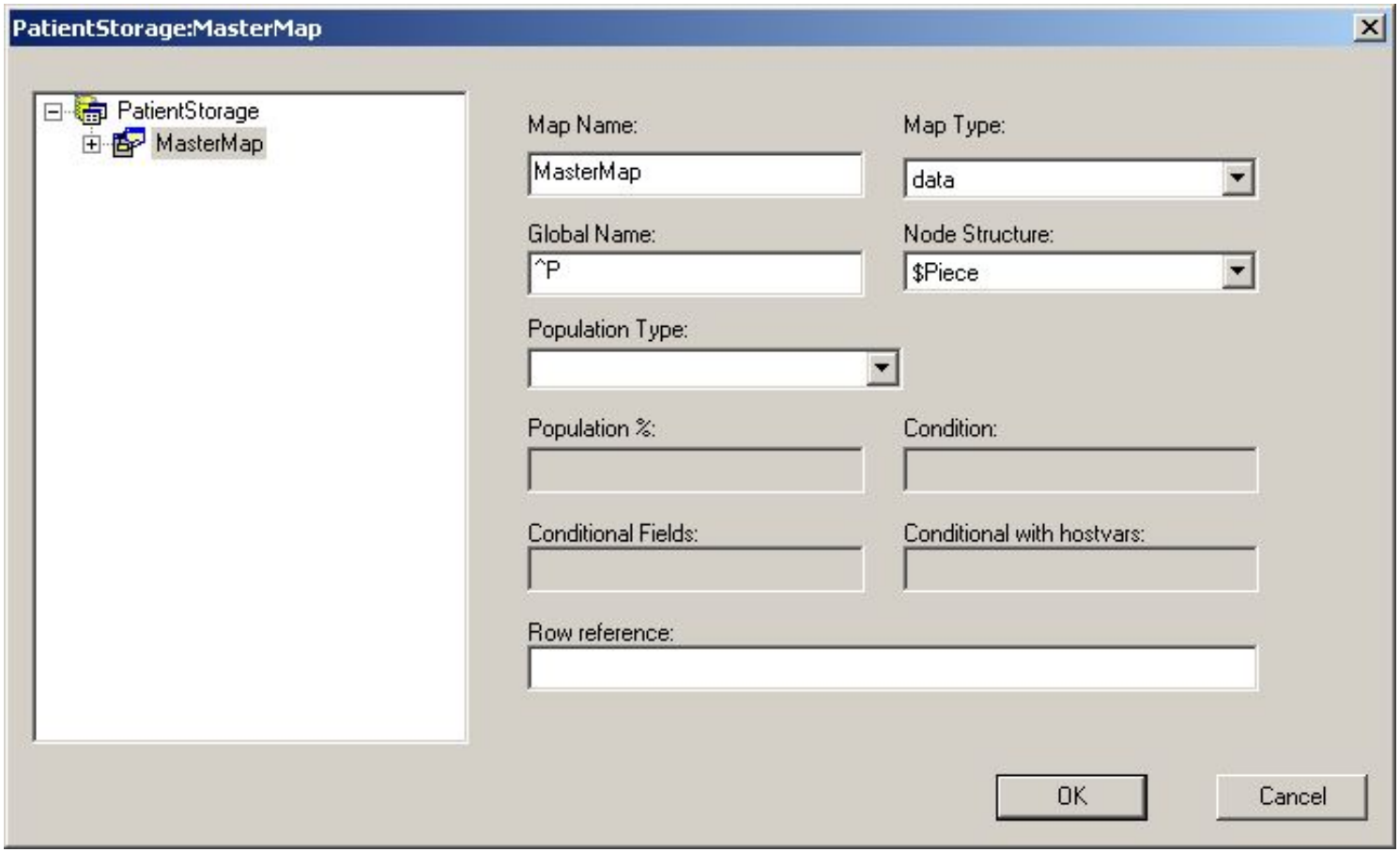
InterSystems CACHÉ
Make Applications Faster

Создаем Storage



InterSystems CACHÉ
Make Applications Faster

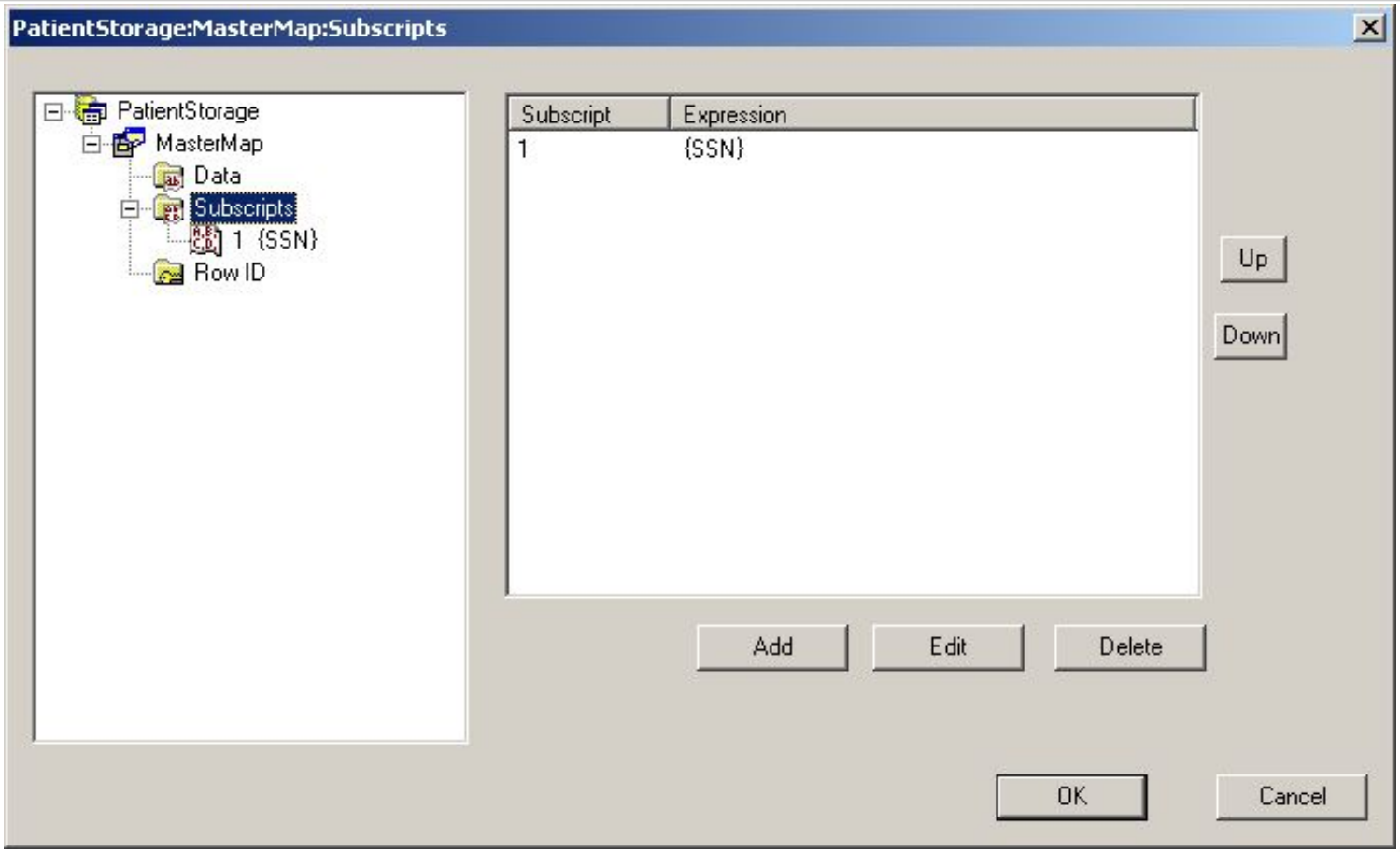
Создаем карту данных



- Имя карты не может содержать символ «пробел»

InterSystems CACHÉ
Make Applications Faster

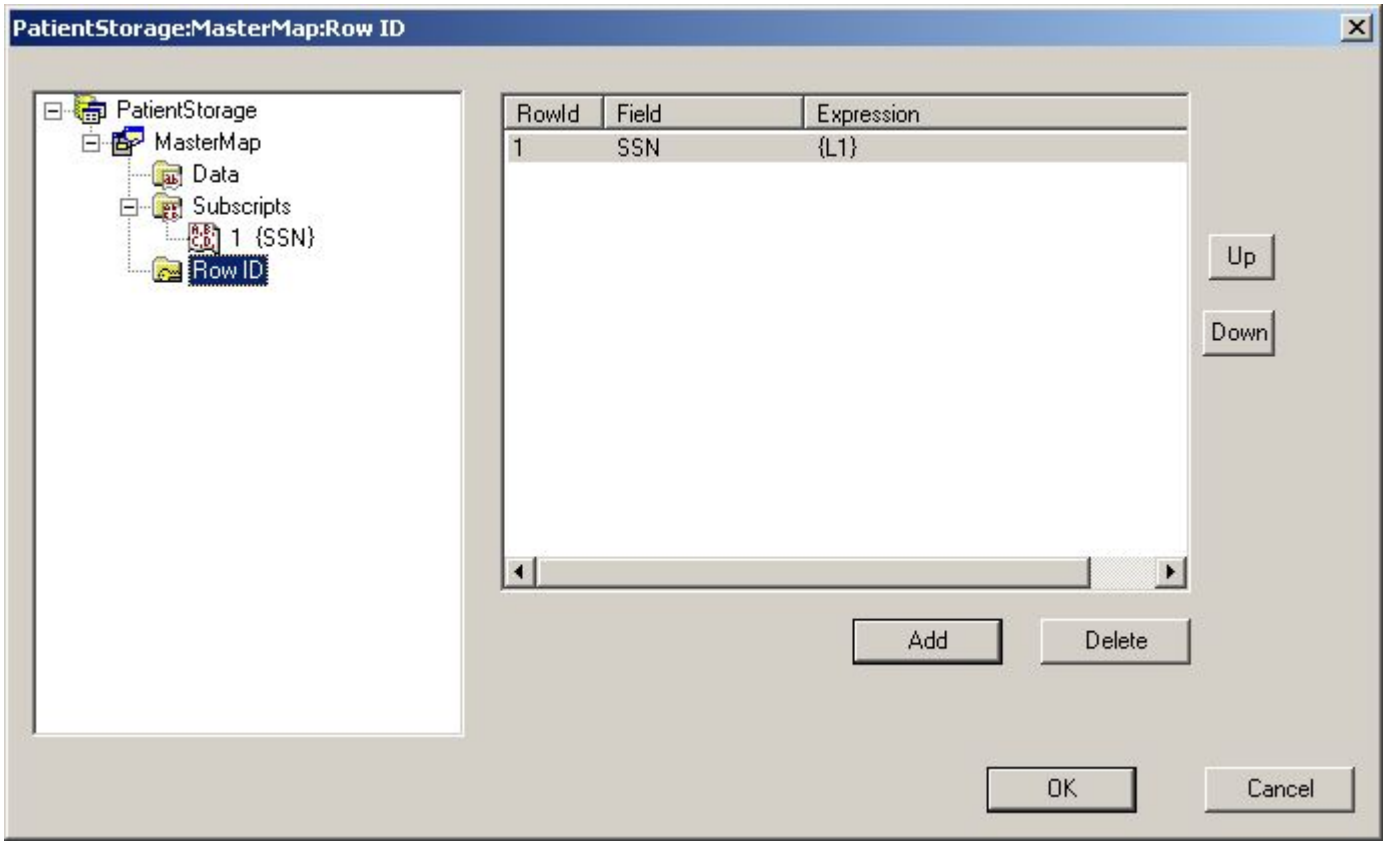
Определяем индексы глобала



- Первый уровень индекса глобала - SSN

InterSystems CACHÉ
Make Applications Faster

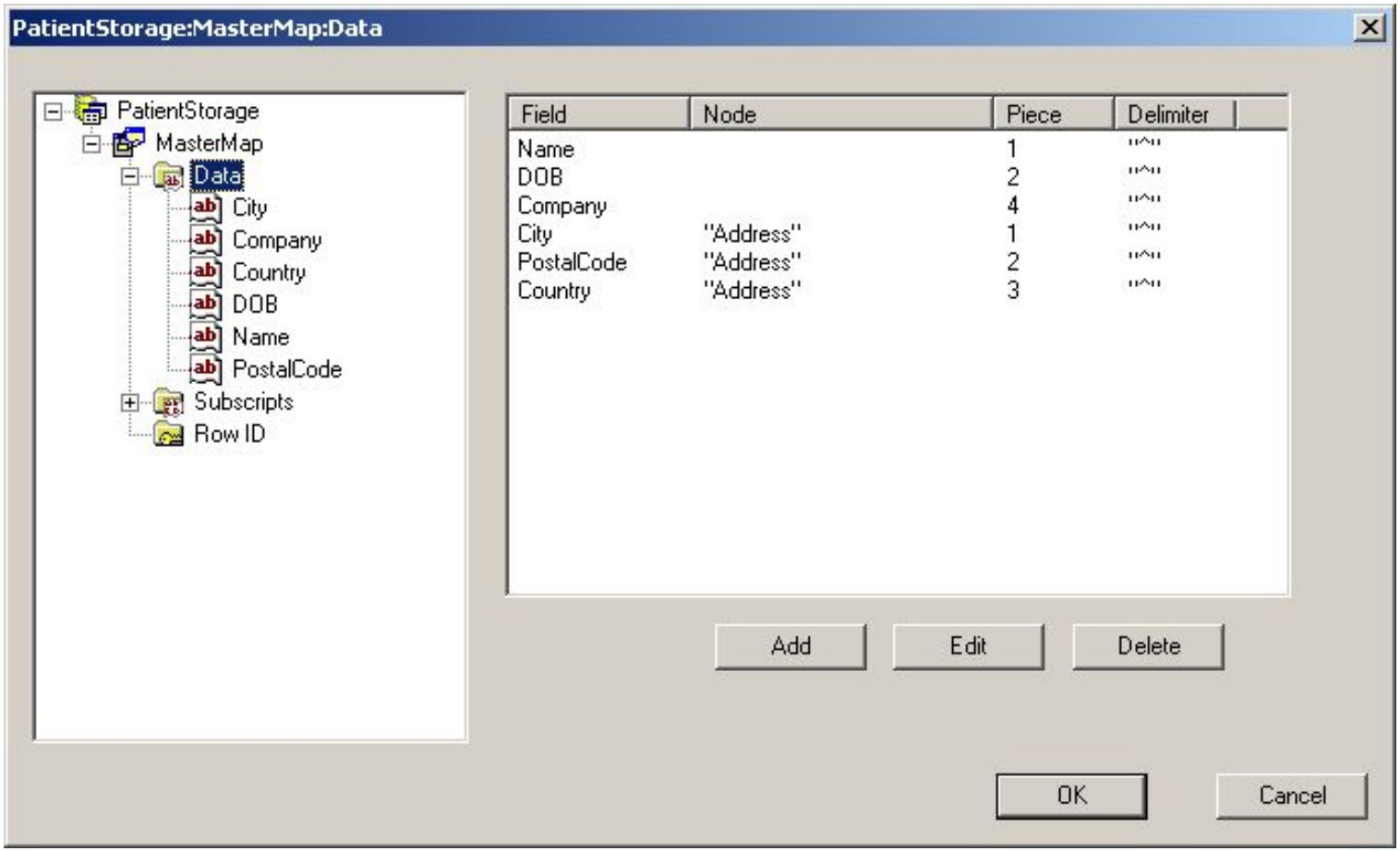
Определяем Row ID



- Первый Row ID 1 основан на SSN, которое хранится в первом уровне индекса глобала

InterSystems CACHÉ
Make Applications Faster

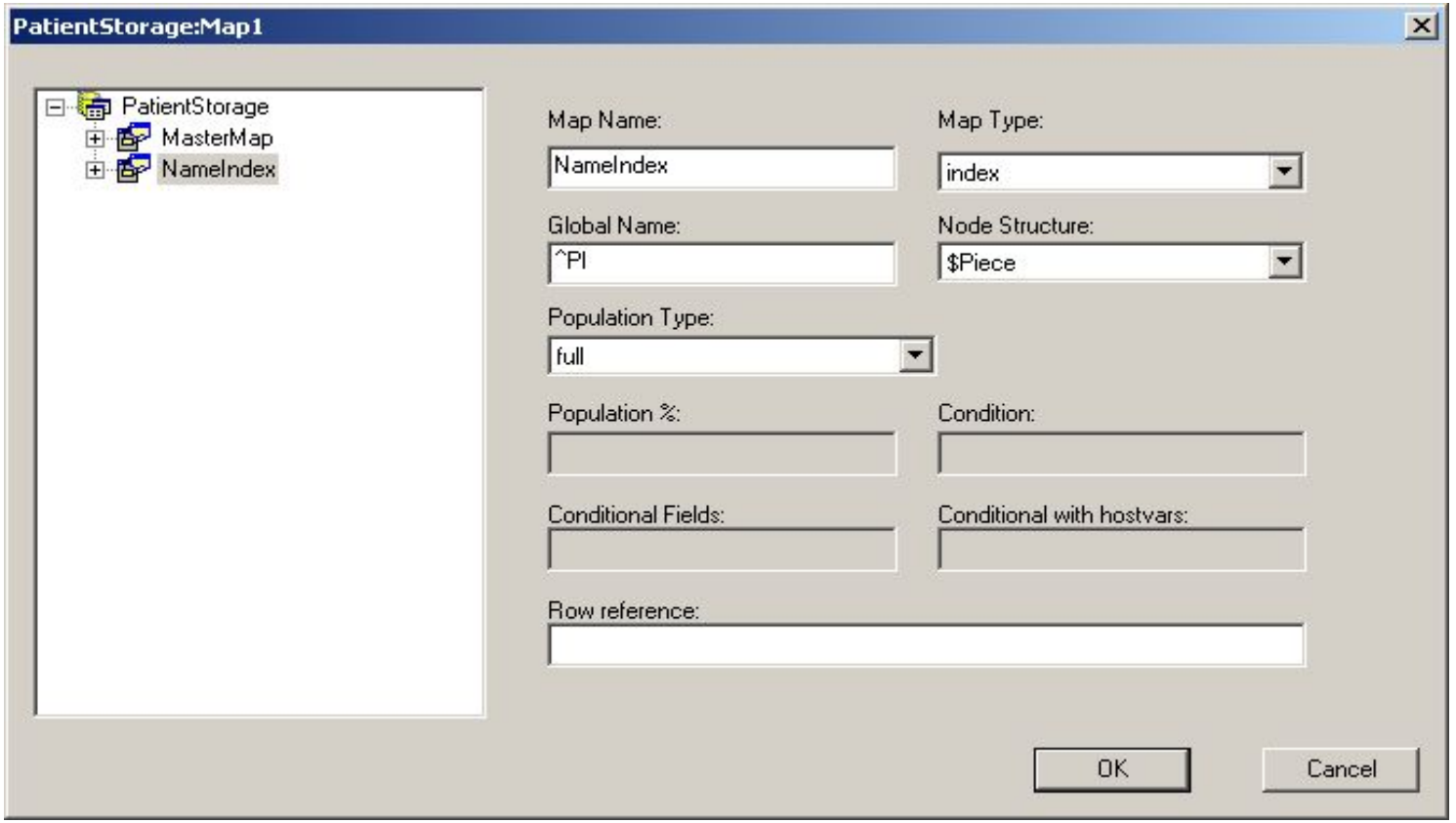
Определяем свойства



- Введите разделитель и дополнительную информацию

InterSystems CACHÉ
Make Applications Faster

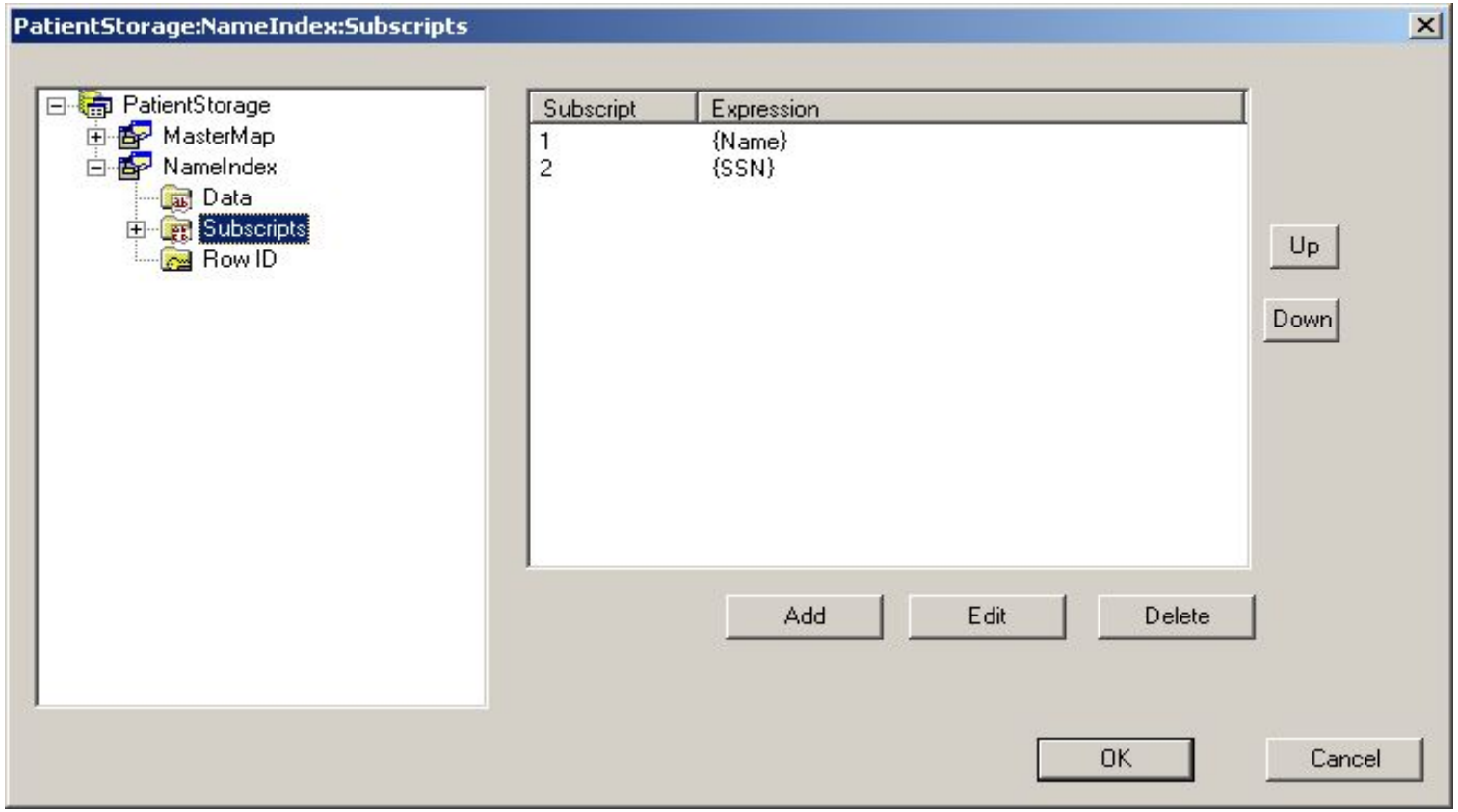
Срздаем карту индексов



- Выберите тип заполнения ‘full’

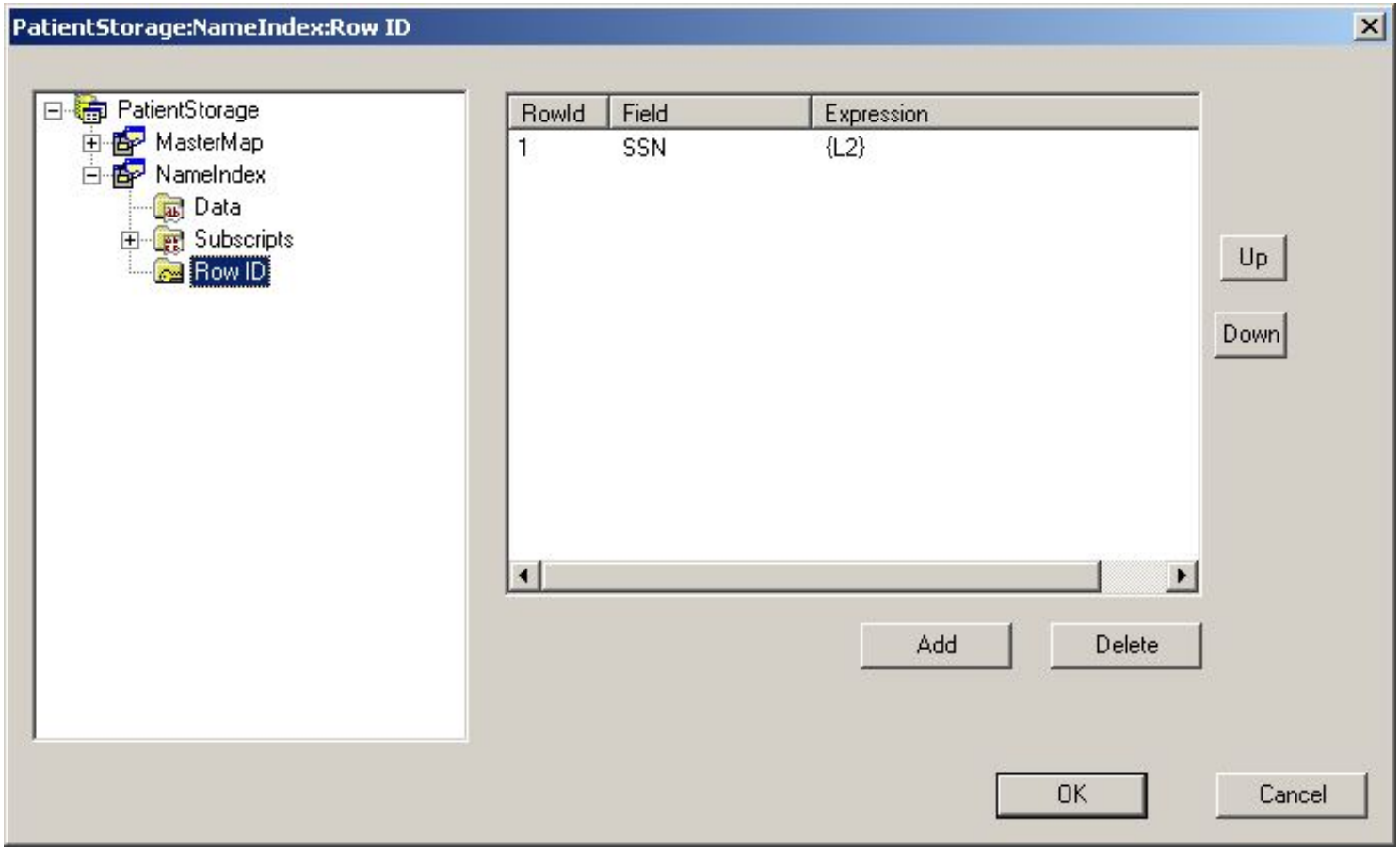
InterSystems CACHÉ
Make Applications Faster

Определяем индексы глобала индексов



InterSystems CACHÉ
Make Applications Faster

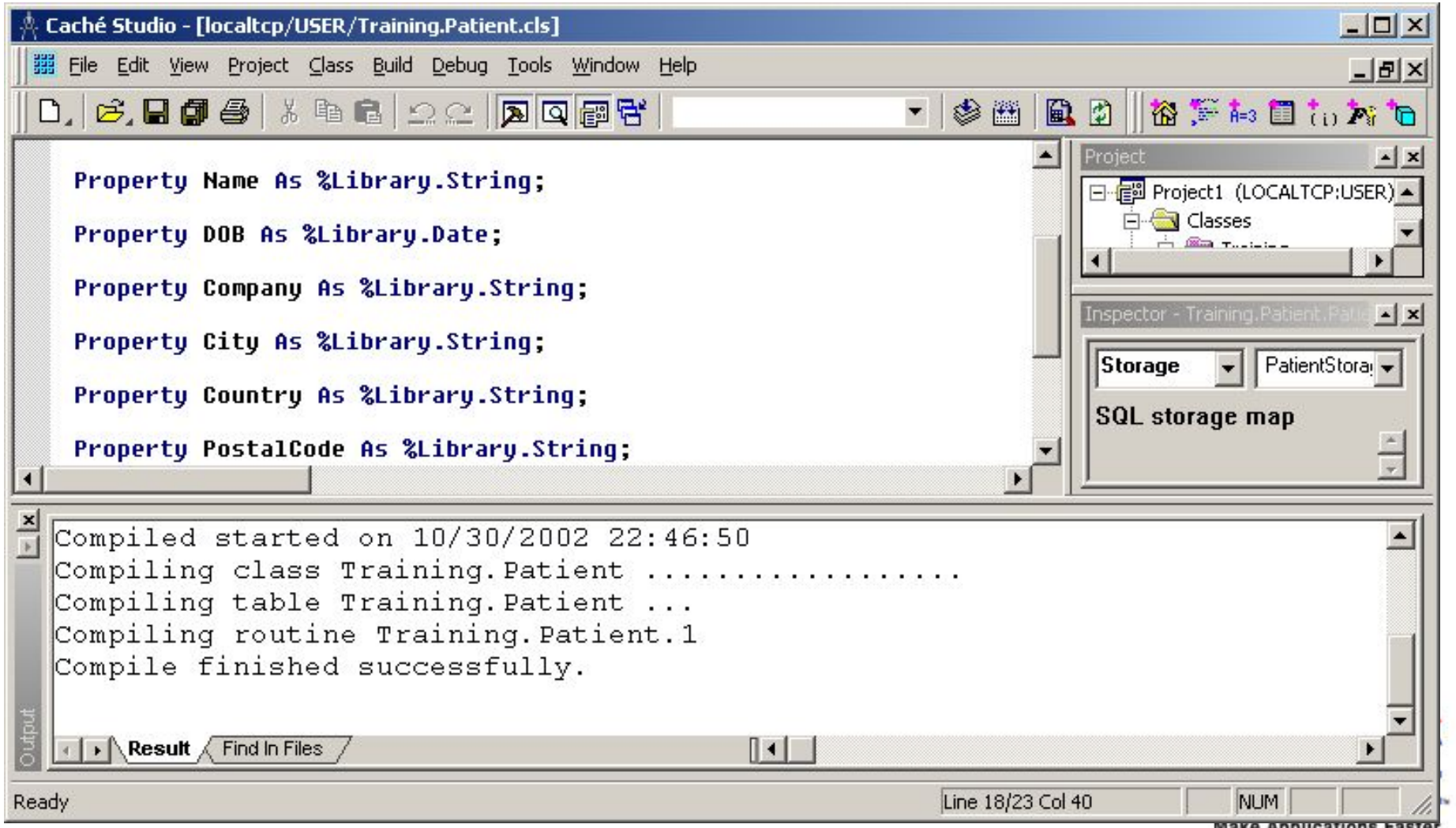
Определяем Row ID индекса



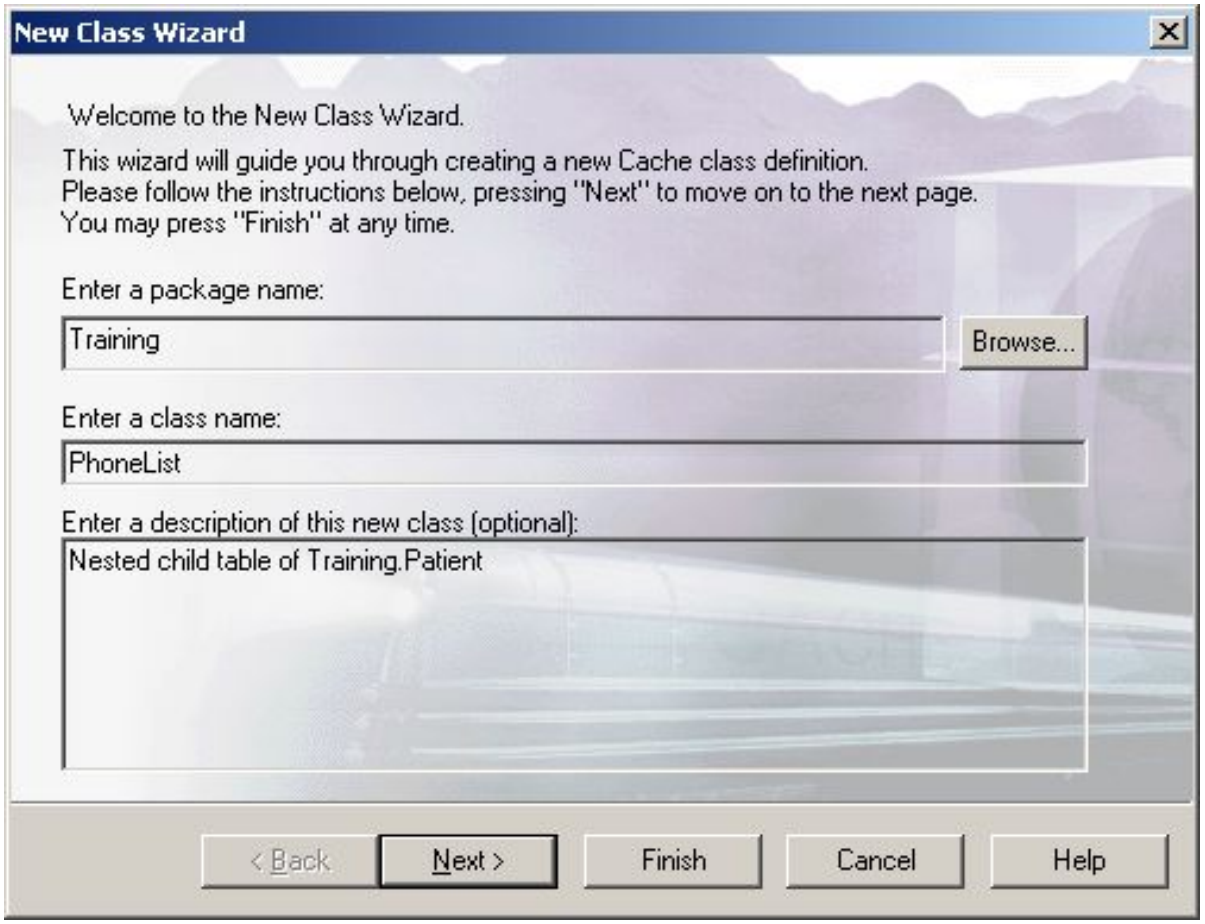
- Первый Row ID основан на SSN, который хранится в уровне 2 индекса глобала

InterSystems CACHÉ
Make Applications Faster

Сохраняем и компилируем класс



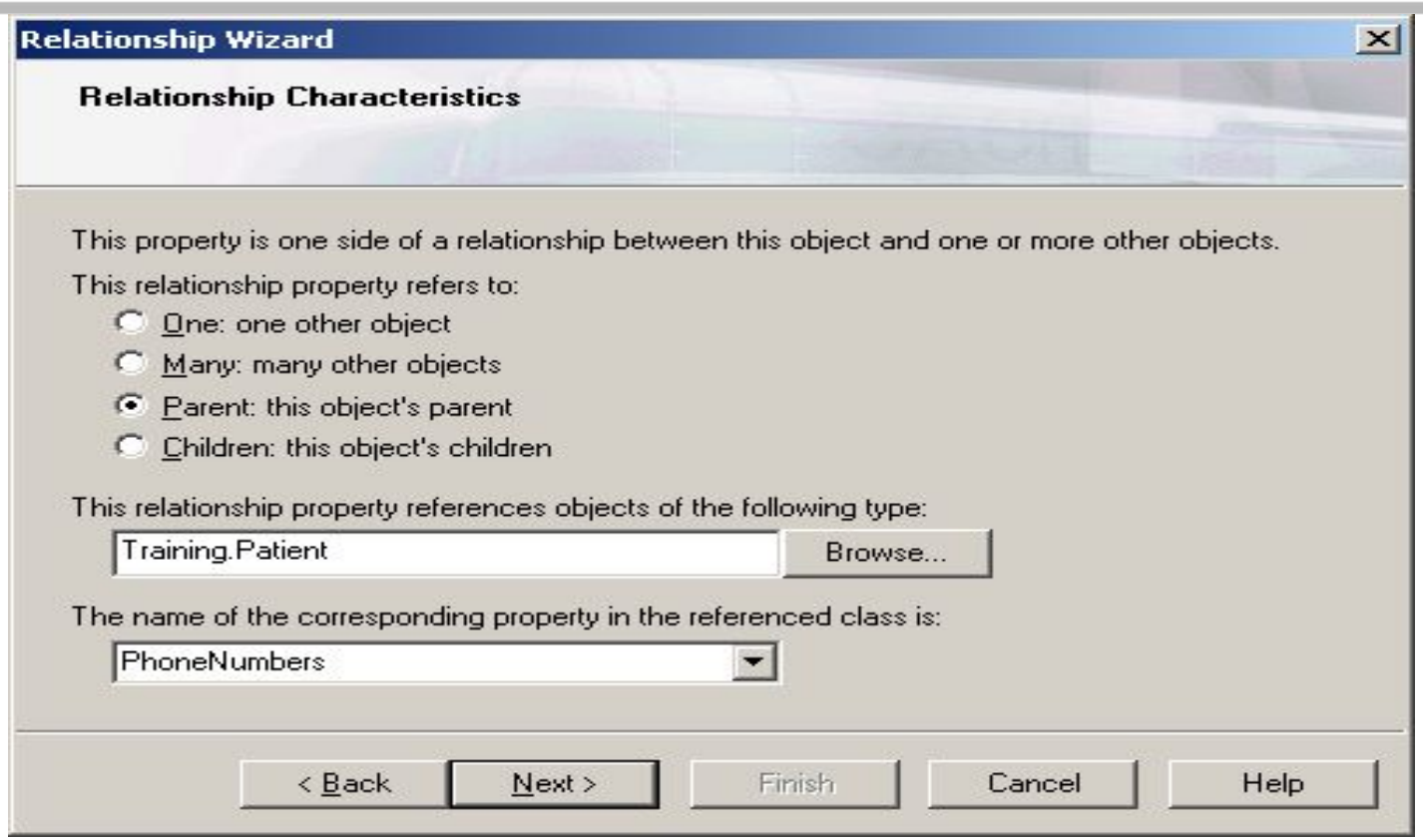
Создаем дочернюю таблицу PhoneList



- Этот класс тоже Persistent

InterSystems CACHÉ
Make Applications Faster

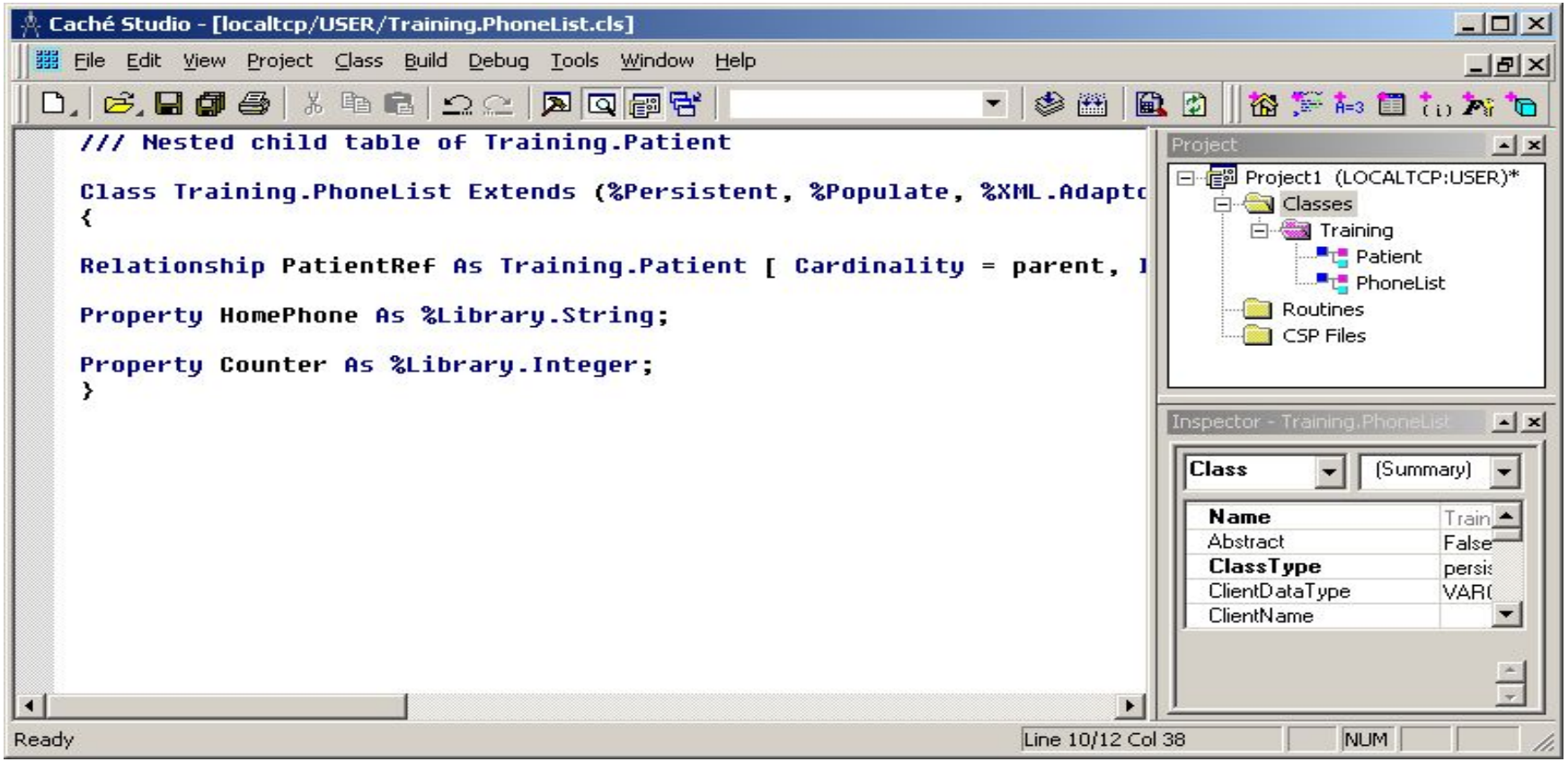
Создаем отношение Parent-Child



- Отношения (Relationship) – специальный класс свойств
- Кроме определения свойства в этом классе (PatientRef), Вы должны определить другую сторону отношения (PhoneNumbers)

InterSystems CACHÉ
Make Applications Faster

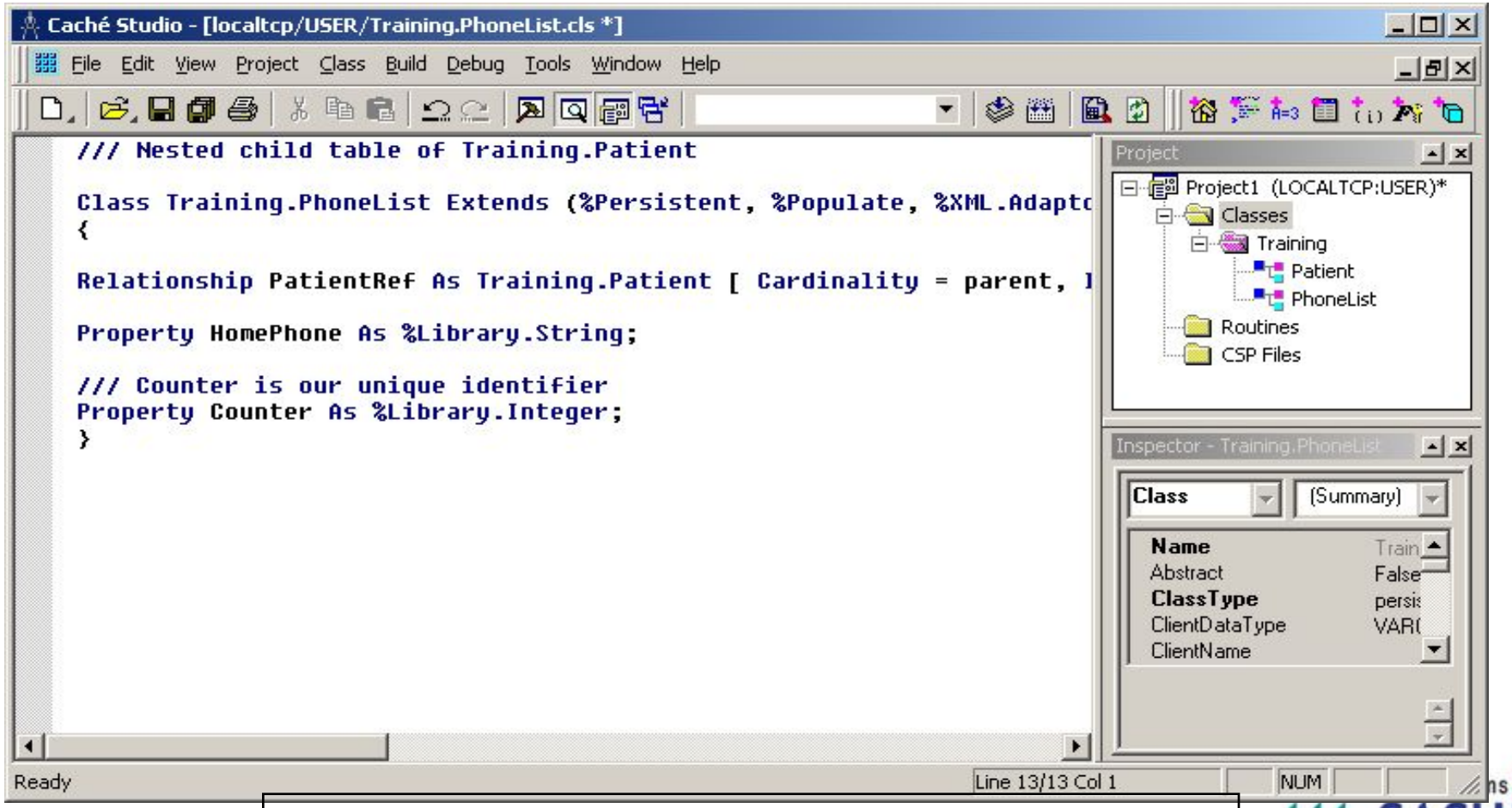
Добавляем остальные свойства



- Кроме свойства, Вы должны определить свойство для представления позиции во встроенной разделенной (Counter)

InterSystems CACHÉ
Make Applications Faster

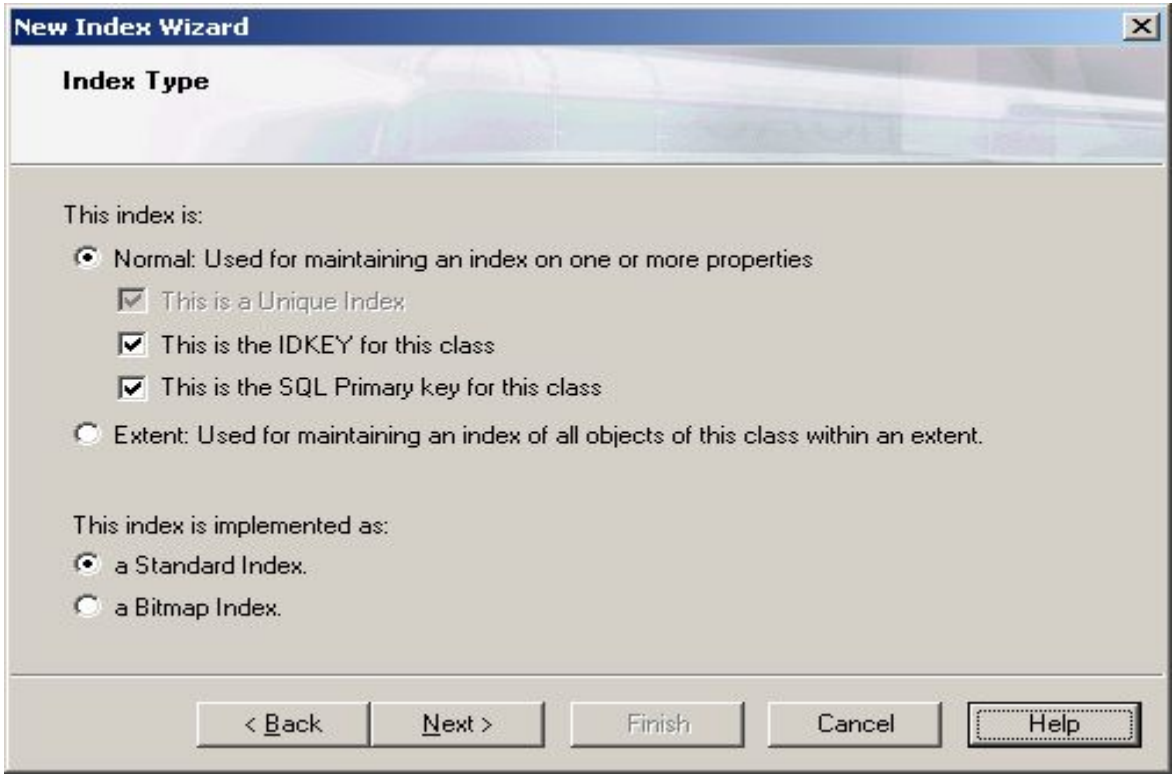
Выбираем уникальный идентификатор



- Наш идентификатор будет строится на поле Counter

Make Applications Faster

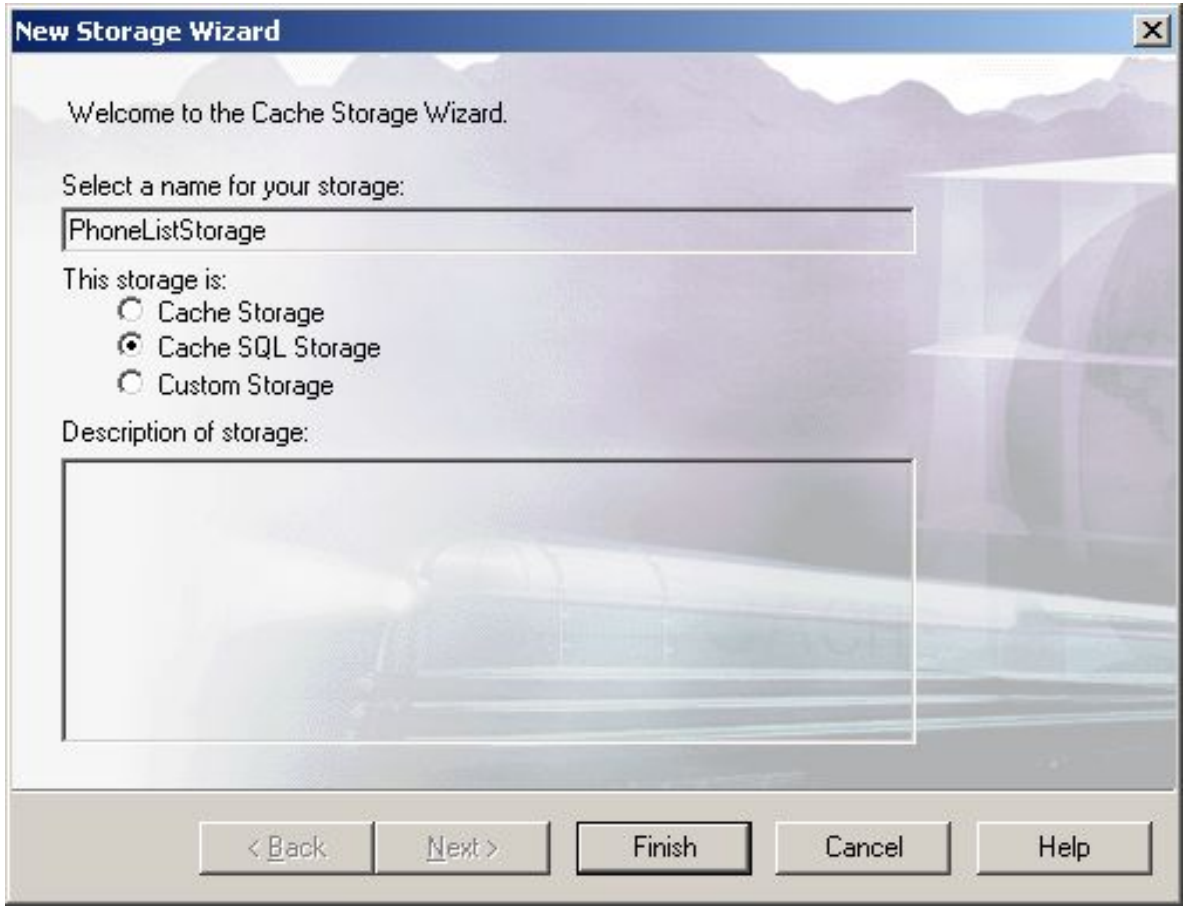
Определяем индекс ID / Primary Key



- Задайте индекс по свойству Counter
- Не модифицируйте collation
- Свойство PatientRef неявно часть IDKey / Primary Key

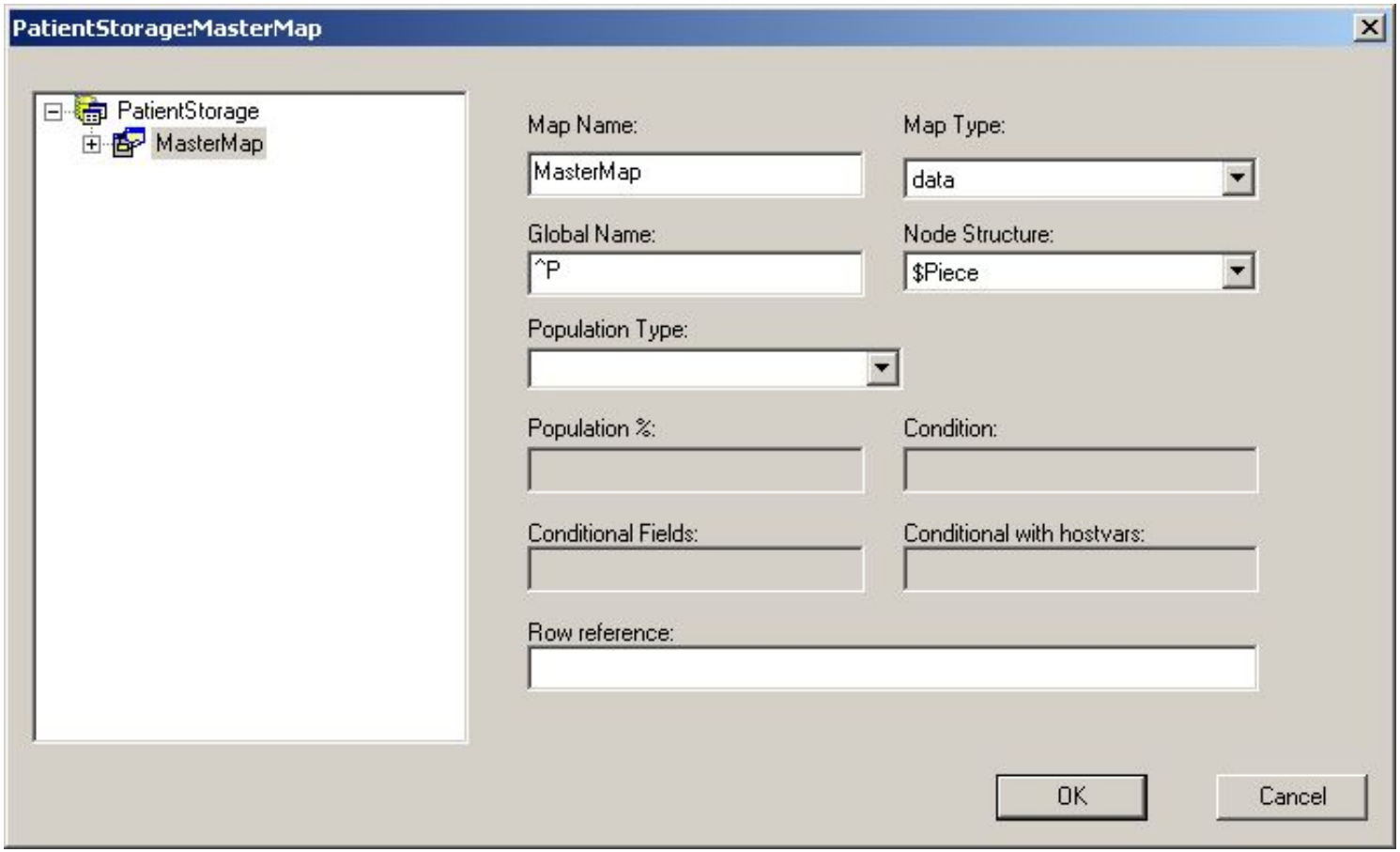
InterSystems CACHÉ
Make Applications Faster

Создаем Storage



InterSystems CACHÉ
Make Applications Faster

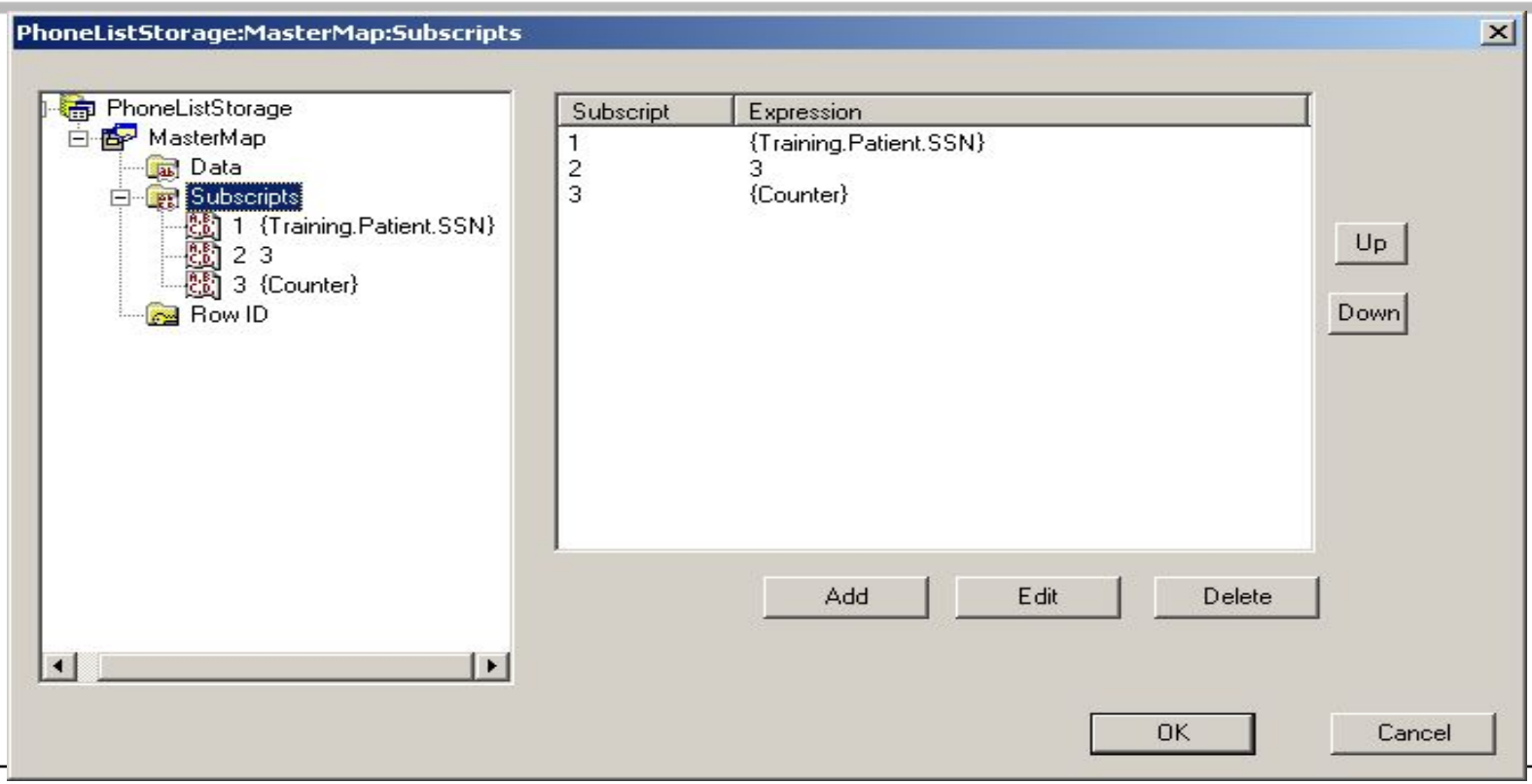
Создаем карту данных



- Имя карты не может содержать символ «пробел»

InterSystems CACHÉ
Make Applications Faster

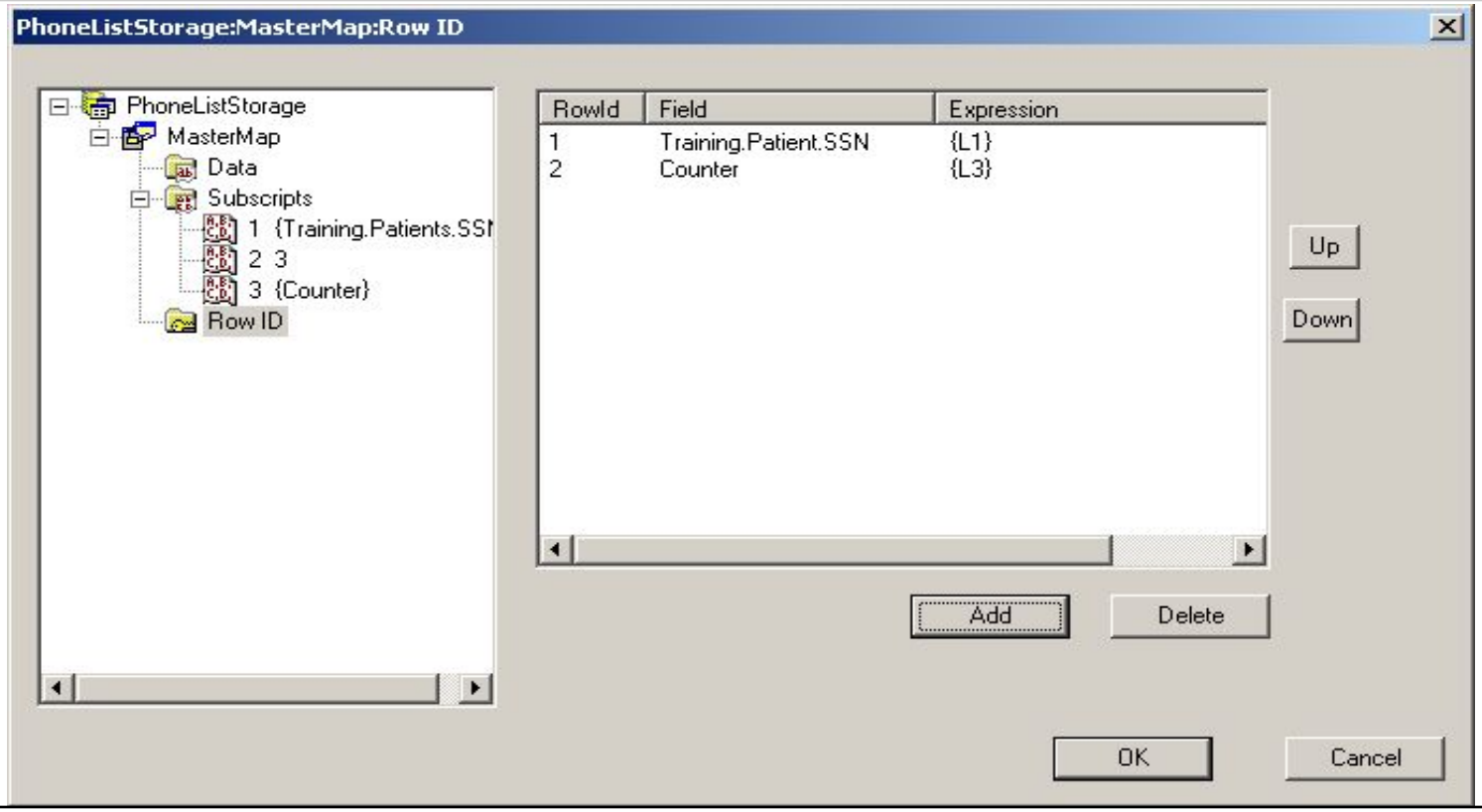
Определяем индексы глобала



- Первый уровень индекса глобала основан на Training.Patient.SSN
- Второй уровень индекса глобала основан на “^” разделителе, используя третью позицию
- Третий уровень индекса глобала основан на “~” разделителе, используя свойство Counter для позиции

InterSystems CACHÉ
Make Applications Faster

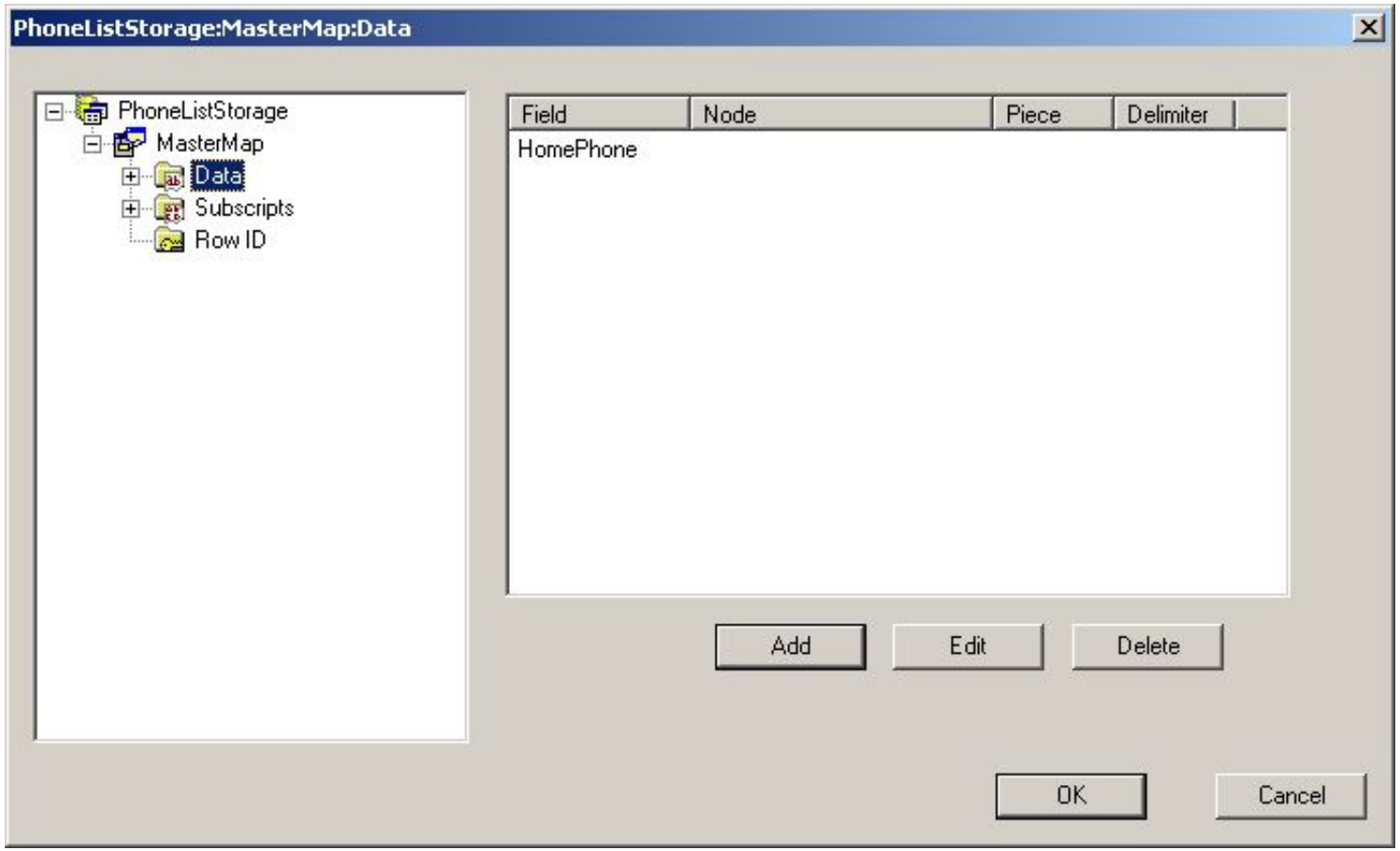
Определяем Row ID



- Row ID 1 основан на Training.Patient.SSN, которое хранится на первом уровне индекса глобала
- Row ID 2 основан на Counter, которое хранится на первом уровне индекса глобала

InterSystems CACHE
Make Applications Faster

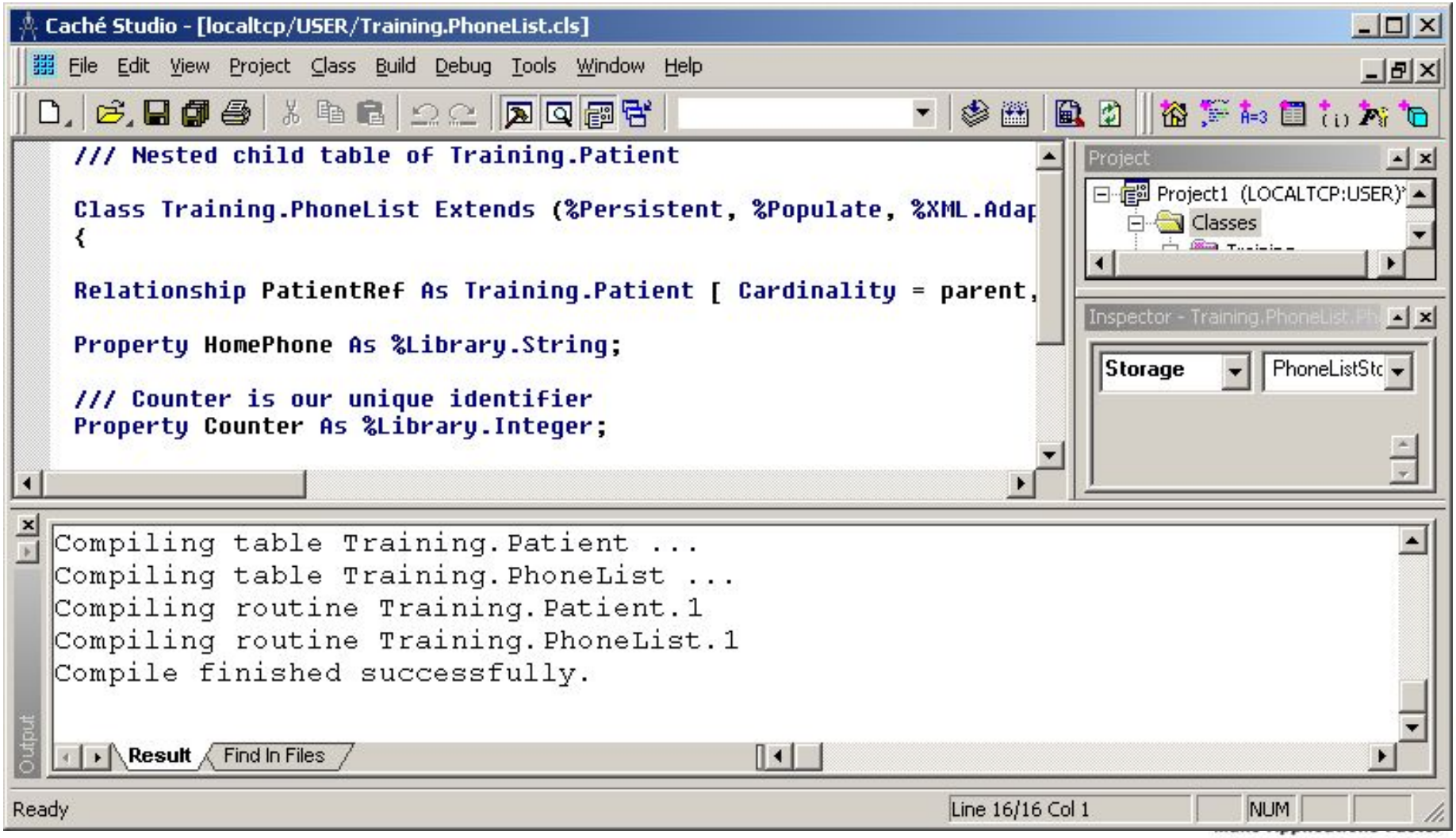
Определяем свойства



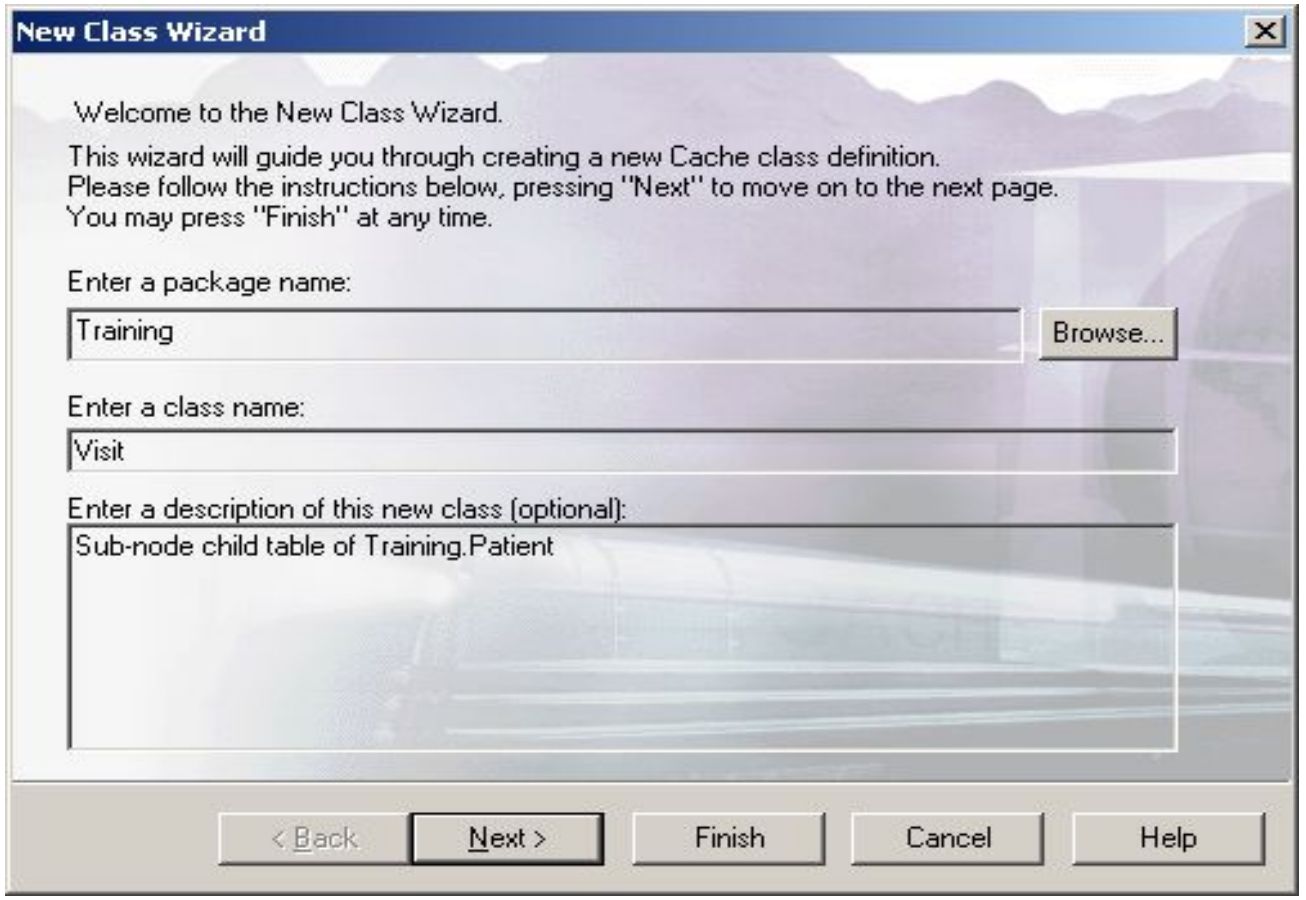
- Внесите только свойство HomePhone

InterSystems CACHÉ
Make Applications Faster

Сохраняем и компилируем класс



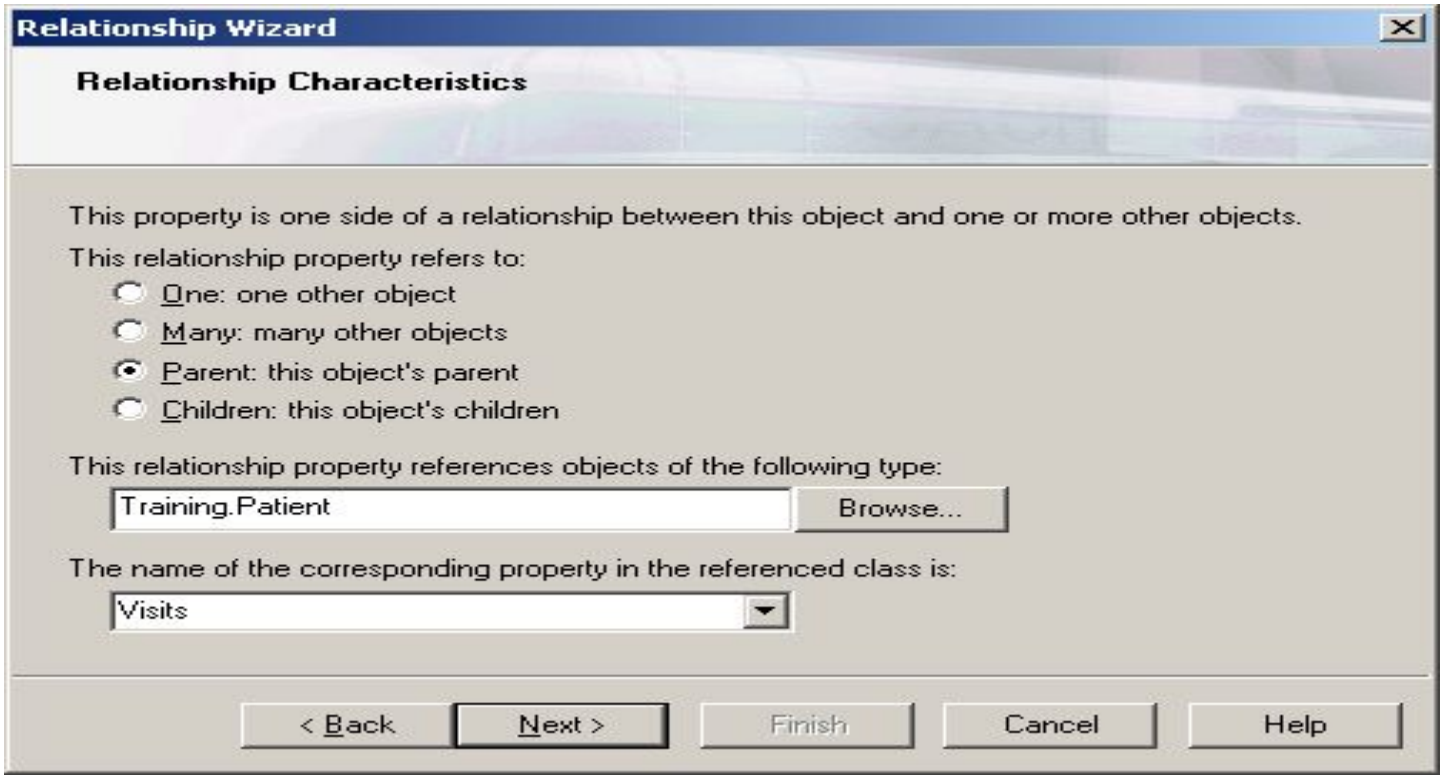
Создаем дочернюю таблицу Visit



- Этот класс тоже Persistent

InterSystems CACHÉ
Make Applications Faster

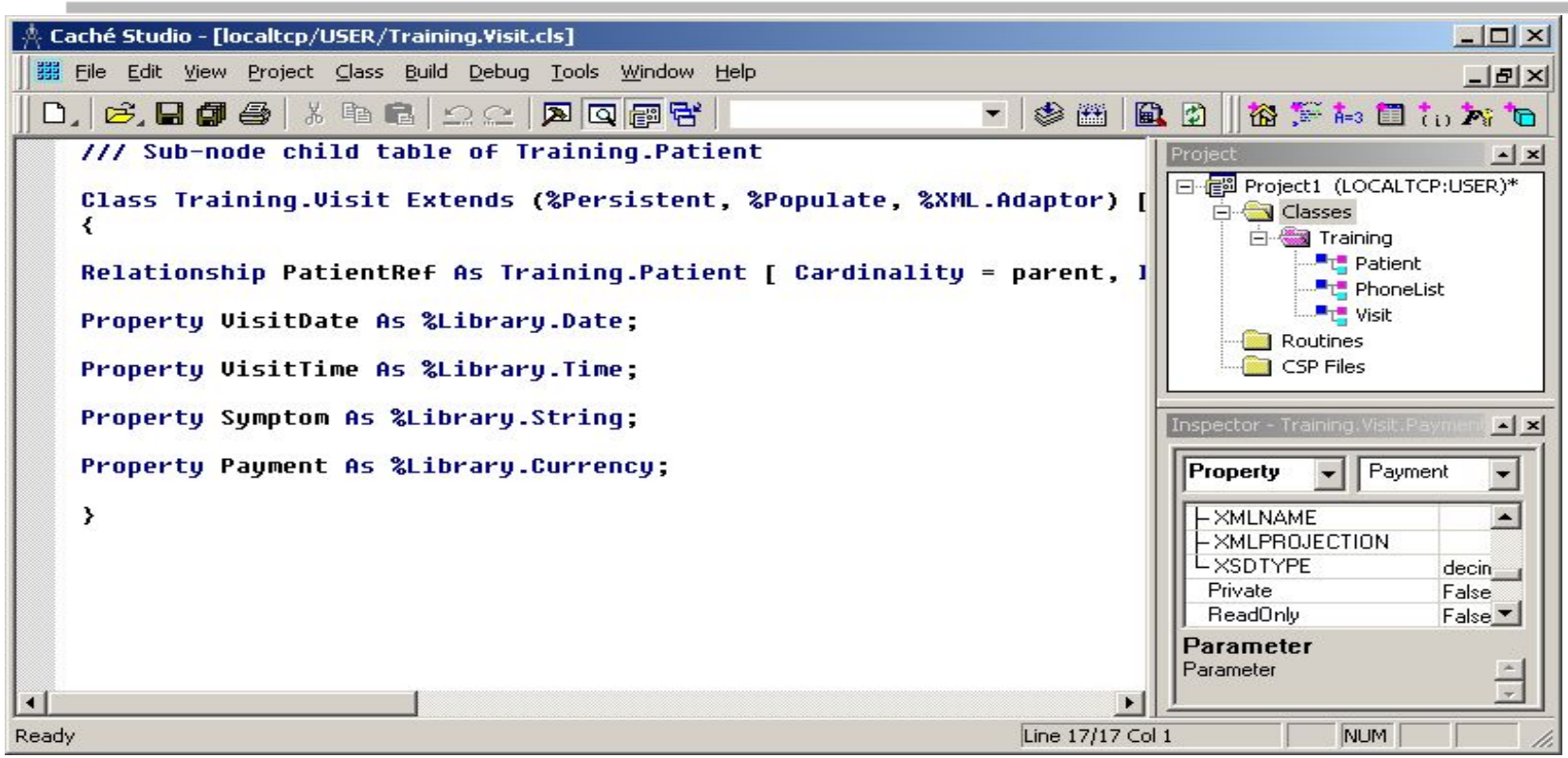
Создаем отношение Parent-Child



- Кроме определения свойства в этом классе (PatientRef), Вы должны определить другую сторону отношения (Visits)

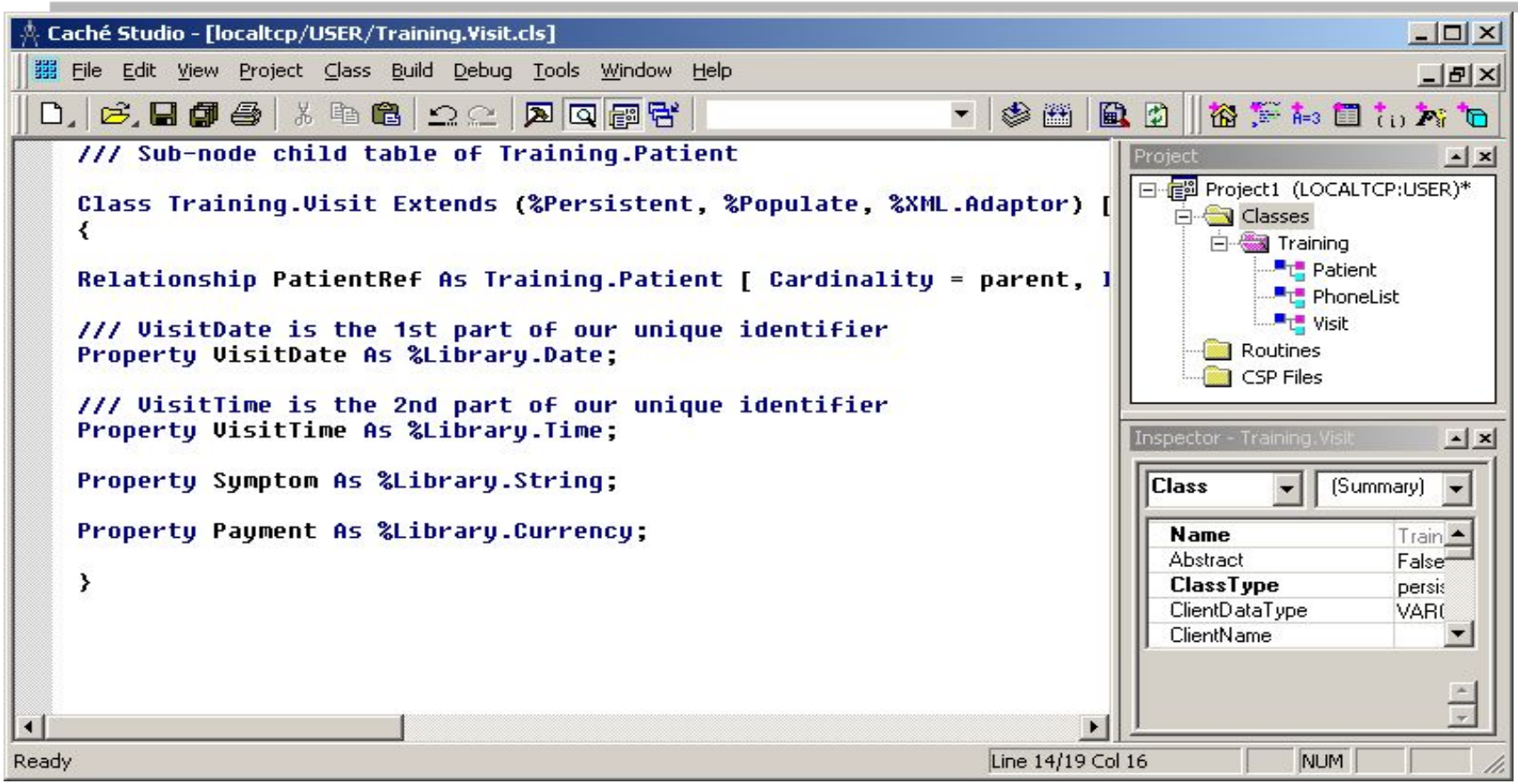
InterSystems CACHÉ
Make Applications Faster

Добавляем остальные свойства



InterSystems CACHÉ
Make Applications Faster

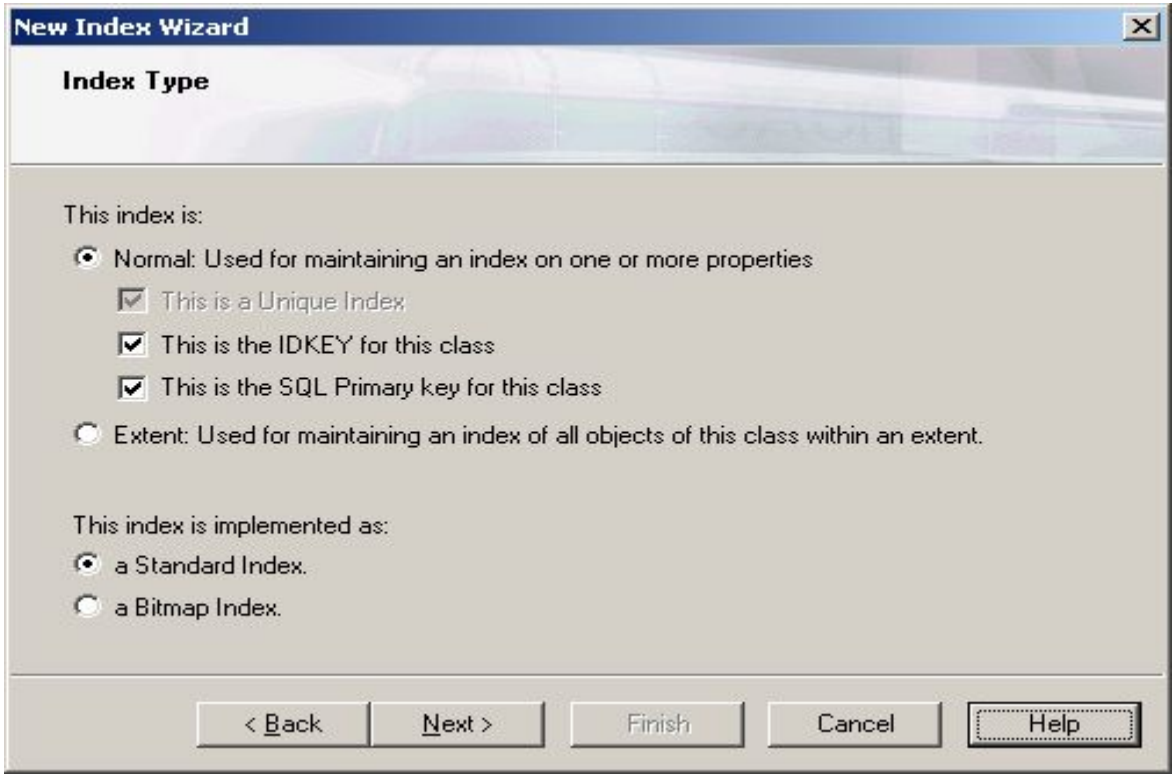
Выбираем уникальный идентификатор



- В этот раз идентификатор будет строиться по 2 полям: VisitDate и VisitTime

InterSystems CACHÉ
Make Applications Faster

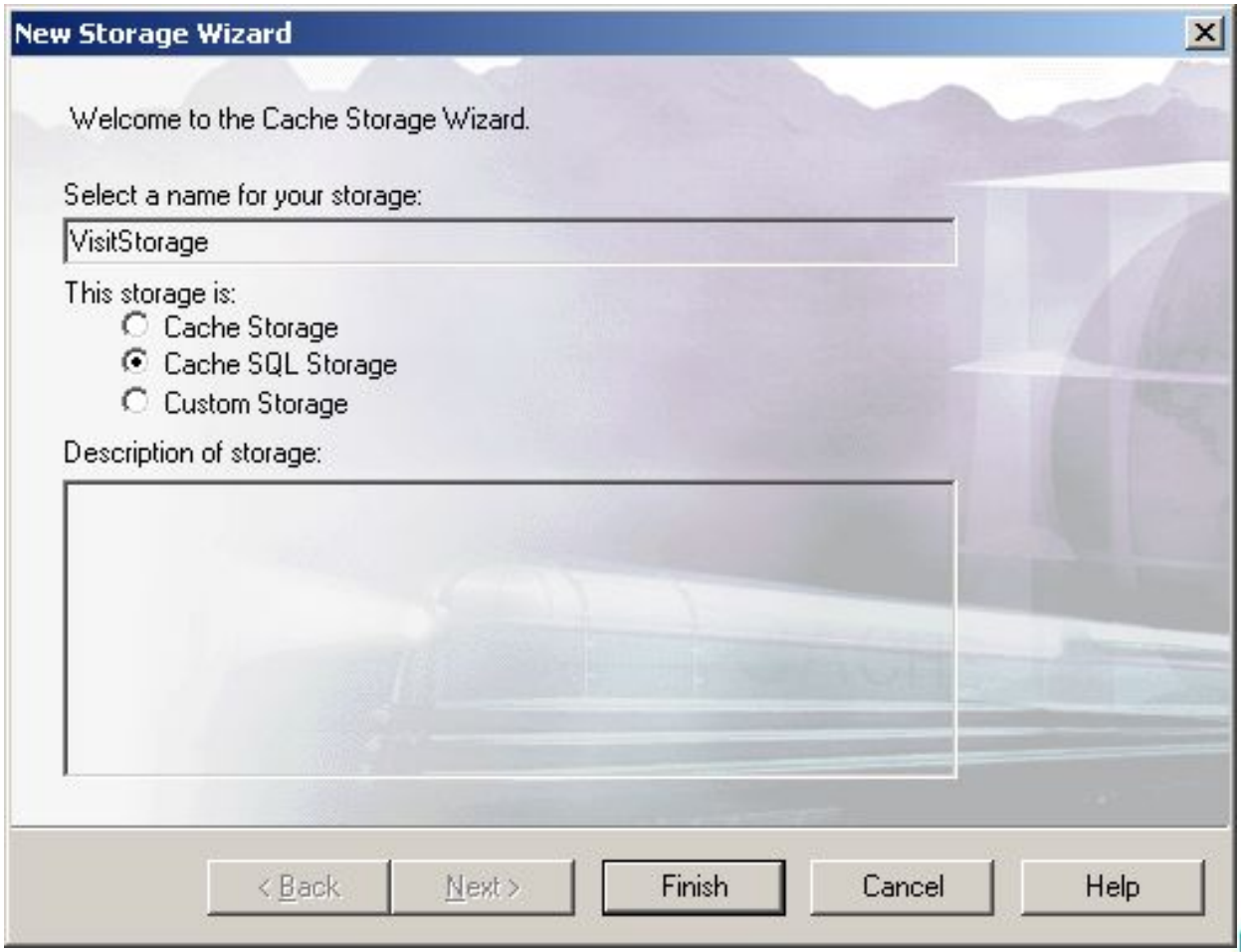
Определяем индекс ID / Primary Key



- Постройте индекс по свойствам VisitDate и VisitTime
- Не изменяйте collation
- Свойство PatientRef неявно часть IDKey / Primary Key

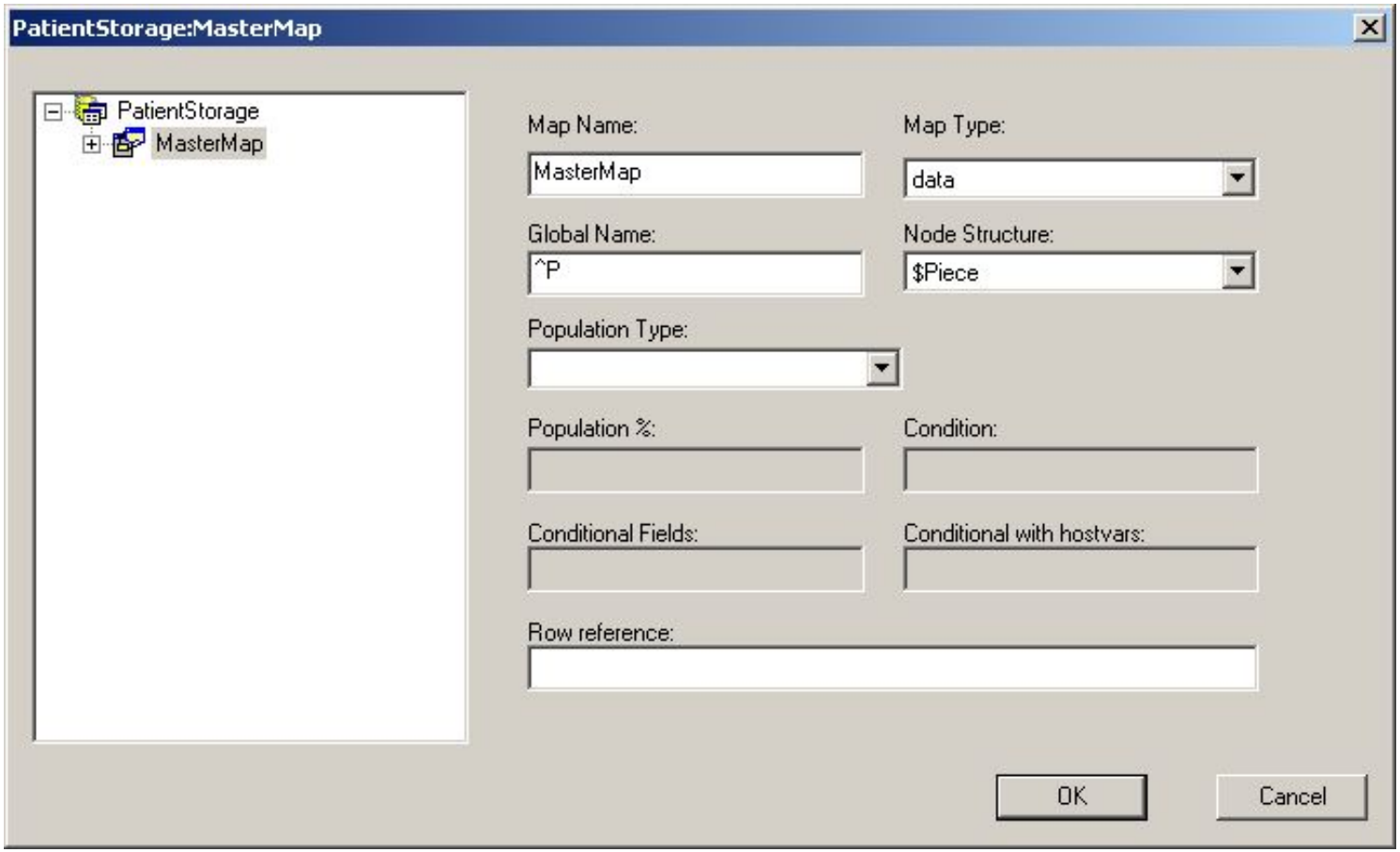
InterSystems CACHÉ
Make Applications Faster

Создаем Storage



InterSystems CACHÉ
Make Applications Faster

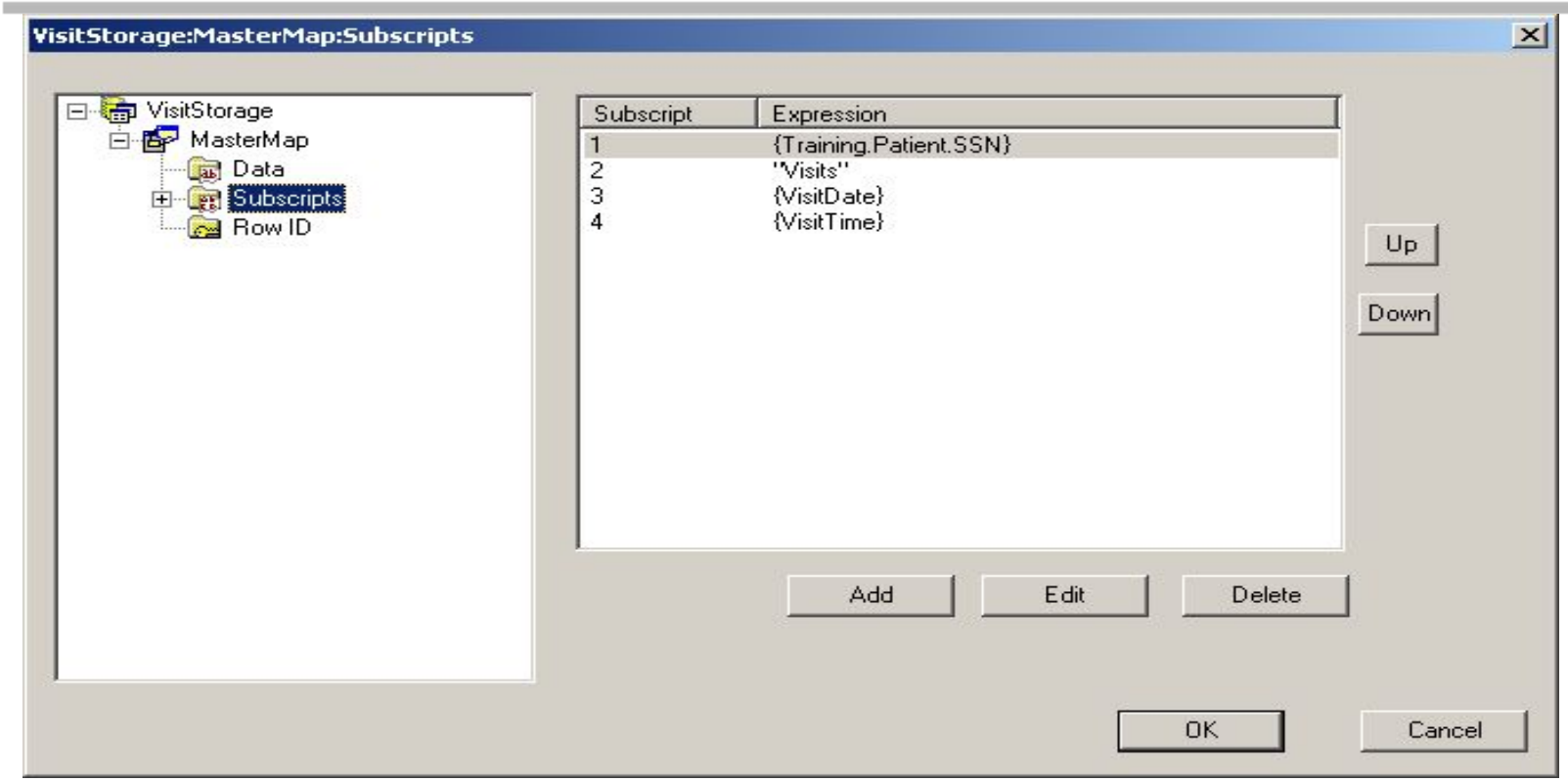
Создаем карту данных



- Имя карты не может содержать символ «пробел»

InterSystems CACHÉ
Make Applications Faster

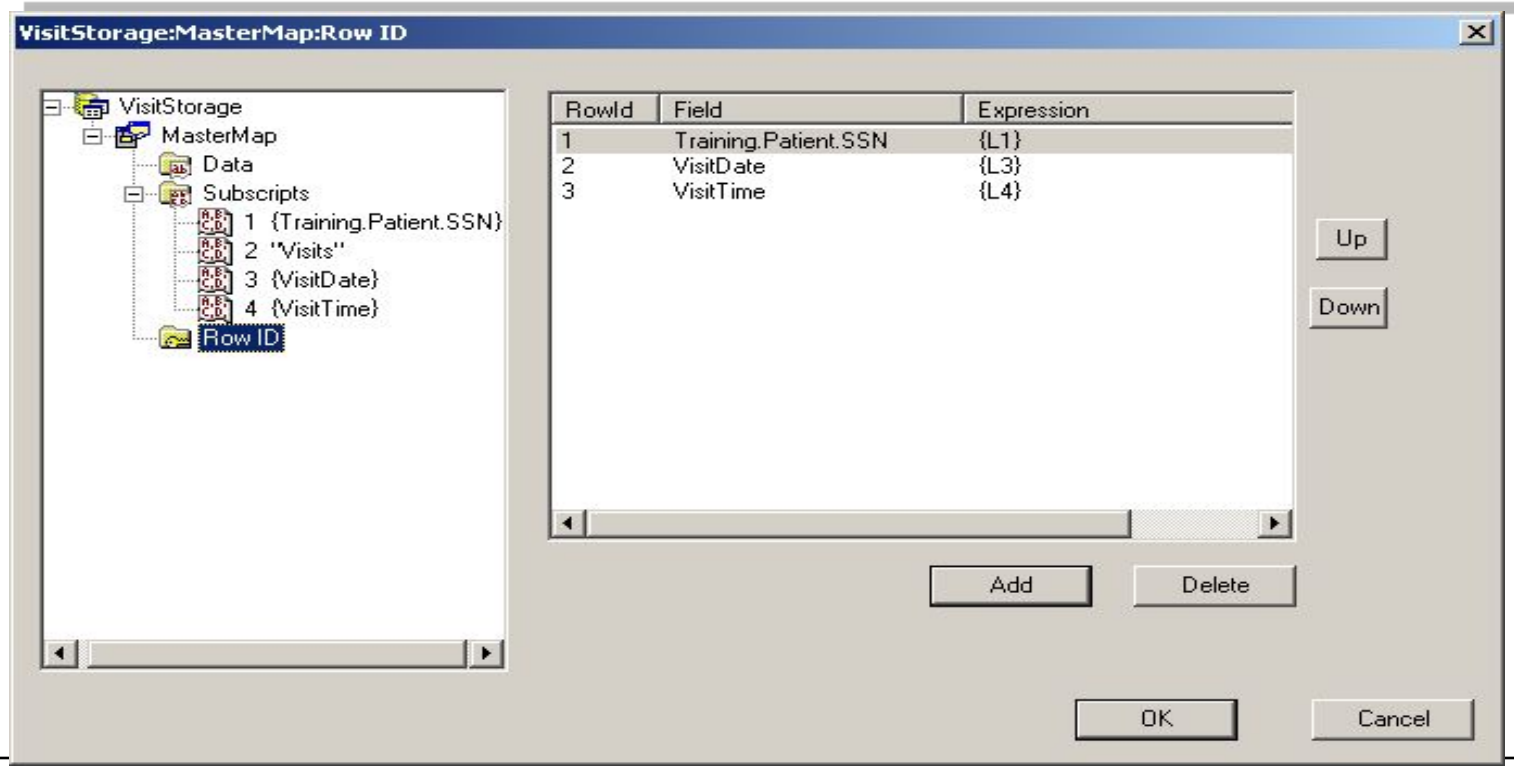
Определяем индексы глобала



- Первый уровень индекса глобала основан на Training.Patient.SSN
- Второй уровень индекса глобала основан на литерале “Visits”
- Третий уровень индекса глобала основан на VisitDate
- Четвертый уровень индекса глобала основан на VisitTime

InterSystems CACHÉ
Make Applications Faster

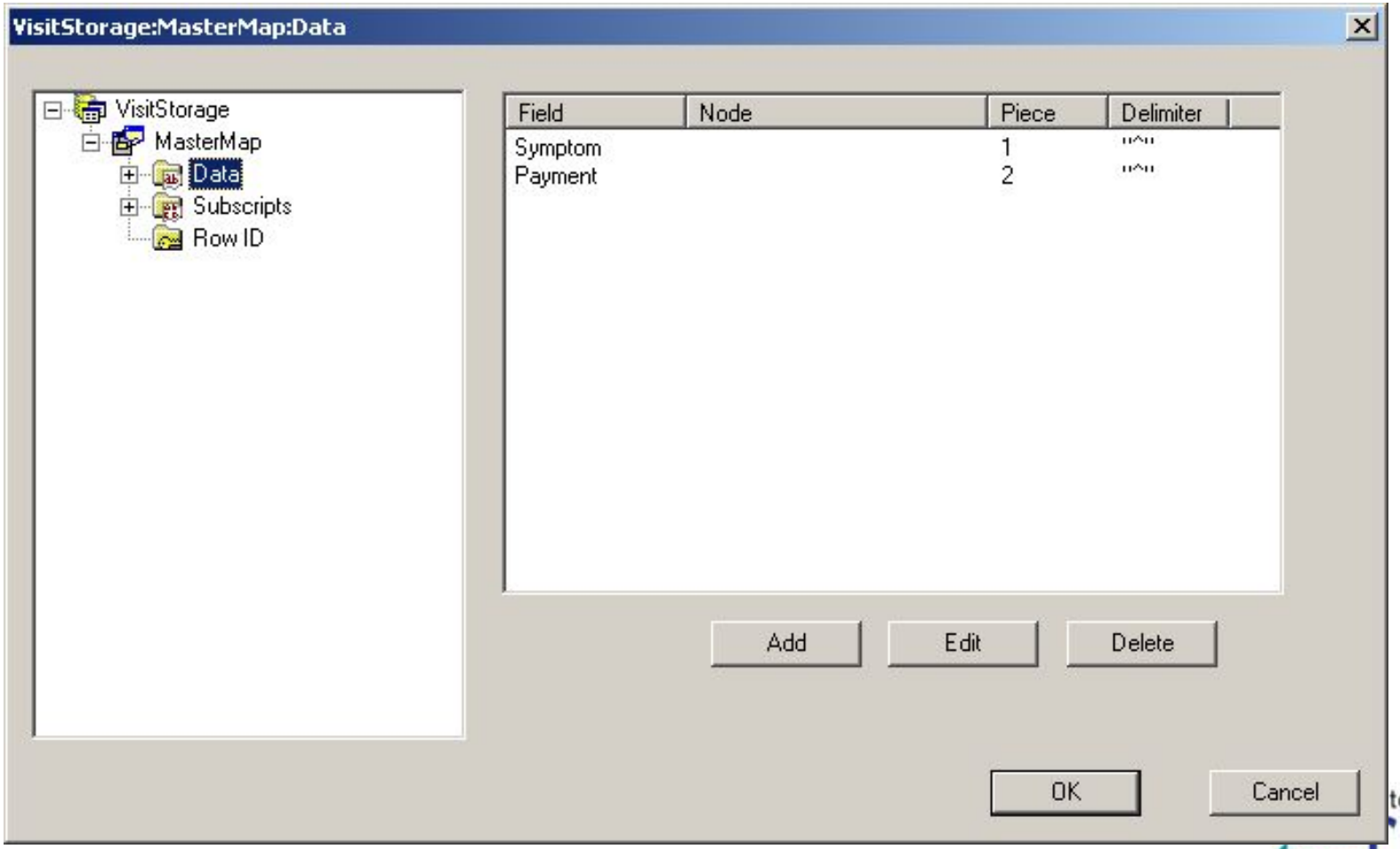
Определяем Row ID



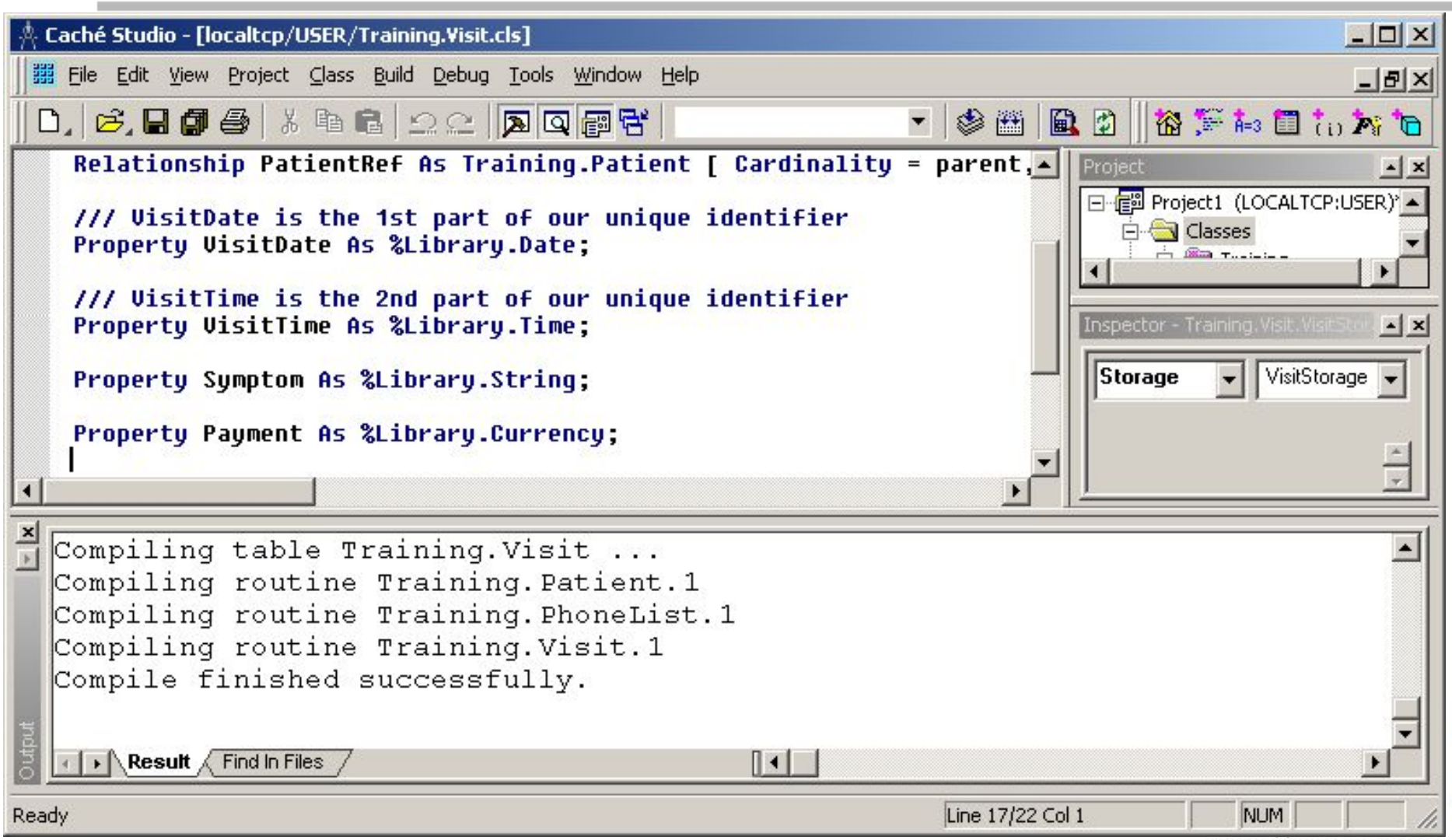
- Row ID 1 основан на свойстве Training.Patient.SSN, которое хранится на первом уровне индекса глобала
- Row ID 2 основан на свойстве VisitDate, которое хранится на третьем уровне индекса глобала
- Row ID основан на свойстве VisitTime, которое хранится на четвертом уровне индекса глобала

InterSystems CACHE Make Applications Faster

Определяем свойства



Сохраняем и компилируем класс



InterSystems CACHÉ
Make Applications Faster

Работа с существующими глобалами через объекты и SQL

Вадим Федоров
InterSystems Corporation