

Лекція 4-2020_2021

- На попередній лекції
- Три частини реляційної моделі
 - Структурна частина
 - Ключі, посилання, зв'язки
- Цілісна частина
- Етапи проектування
- Приклад проектування

ХНУ ім.В.Н.Каразіна, ФКН,
Лазурик В.М.

1

На прошлой лекции

- NewSQL – совмещают в себе преимущества NoSQL (масштабируемость – scalability, и доступность – availability, и транзакционные требования РБД, обеспечивающих надежность и целостность данных.

Архитектура централизованных БД с сетевым доступом:

- Телеобработка. Один компьютер с единственным процессором соединен с несколькими терминалами. Все ПО на нем.
- Файловый сервер. Пользовательские приложения и сама СУБД на отдельных рабочих станциях, файловый сервер для размещения нужных файлов.
- Клиент-сервер. Двухуровневая модель: сервер БД и приложения БД на рабочих станциях. Между ними промежуточное ПО. Интранет. Трехуровневая модель: браузер, сервер сети+сервер приложений, сервер БД. Интернет.

ХНУ ім.В.Н.Каразіна, ФКН,
Лазурик В.М.

3

На прошлой лекции

Модели данных. Продолжение.

- ООБД оперируют объектами, между данными и функциями их обработки устанавливаются взаимосвязи подобные соответствующим средствам в объектно-ориентированных языках программирования.
- ОРСУБД – реляционная СУБД, поддерживающая некоторые технологии, реализующие объектно-ориентированный подход.
- NoSQL – «не только SQL» (Not Only SQL). NoSQL БД обеспечивают масштабируемость и доступность, жертвуя атомарностью и согласованностью данных. В NoSQL нет заранее заданных схем документов, отсутствует поддержка целостности данных, отсутствует механизм транзакций. Модели: «ключ-значение», колоночные, документоориентированные, графовые.

ХНУ ім.В.Н.Каразіна, ФКН,
Лазурик В.М.

2

Реляционная модель данных

Данные хранятся в двумерных таблицах. Требования:

- Все строки таблицы должны быть уникальны. Уникальность достигается за счет наличия ключей.
 - Все строки таблицы должны иметь одну и ту же структуру, т.е. одно и то же количество столбцов с соответственно совпадающими именами.
 - Имена столбцов таблицы должны быть различны, а данные в одном столбце должны быть однотипными.
 - Значения атрибутов должны быть атомарными (скалярными), т.е. отношения не могут иметь в качестве компонент другие отношения или массивы.
 - Порядок следования строк в таблице несущественен, так как влияет лишь на скорость доступа к строке.
- Порядок следования столбцов в таблице несущественен.

ХНУ ім.В.Н.Каразіна, ФКН,
Лазурик В.М.

4

Три части реляционной модели данных

Трактовка реляционной модели данных – Кристофер Дейт. Реляционная модель состоит из трех частей:

структурной, целостной и манипуляционной.

1. **Структурная часть**: единственной родовой структурой данных, используемой в реляционных БД, является нормализованное n-арное отношение.

Определяются понятия *доменов*, *атрибутов*, *кортежей*, *заголовка*, *тела* и *переменной* отношения.

2. Манипуляционная часть – *два основных механизма манипулирования реляционными БД (реляционная алгебра и реляционное исчисление)*.

3. Целостная часть – целостность сущности и целостность ссылок.

5

Структурная часть реляционной модели

Домен T – набор допустимых значений для одного или нескольких атрибутов.

Отношение СЛУЖАЩИЕ {Табельный номер, Фамилия, Зарплата, Отдел}. Домен Фамилия определен на типе данных символьные строки. Но первой буквой не может быть мягкий или твердый знак.

Данные считаются сравнимыми, если они относятся к одному и тому же домену.

Номер отдела – целочисленный (2 цифры),

Табельный номер – целочисленный (5 цифр). Разные домены, их сравнение бессмысленно.

Несколько атрибутов – один домен (фамилия, имя).

7

Структурная часть реляционной модели

Сущность – объект любой природы, данные о котором хранятся в БД.

Отношение r – двумерная таблица, содержащая информацию об объектах одного типа, об одной сущности.

Атрибут A – свойство, характеризующее сущность. Это столбец или поле.

Тип данных – тип значений элементов таблицы. Значения данных, хранимые в реляционных БД (РБД) – являются типизированными (символьные, числовые точные, числовые приближенные, специализированные, темпоральные).

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

6

Структурная часть реляционной модели

Заголовок или схема отношения Hr – конечное множество упорядоченных пар вида $\langle A, T \rangle$. Это заголовки столбцов + домен. Например: $\langle \text{Фамилия, ДоменФамилия} \rangle$, $\langle \text{Отдел, ДоменОтдел} \rangle$

Кортежем Tr отношения **r** называется множество упорядоченных триплетов вида $\langle A, T, v \rangle$, где v – допустимое значение типа данных или домена. Кортеж – строка отношения. Например: $\langle \text{Фамилия, ДоменФамилия, Иванов} \rangle$, $\langle \text{Отдел, ДоменОтдел, 11} \rangle$.

Телом Vr отношения **r** называется произвольное множество кортежей **Tr**.

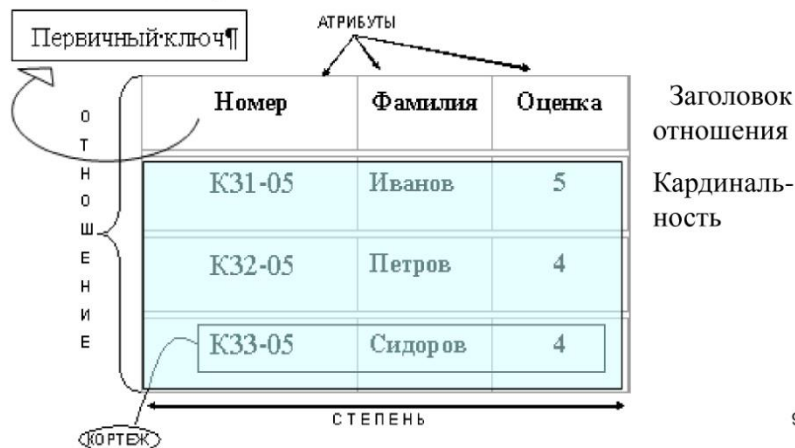
Значением Vr отношения **r** называется пара множеств **Hr** и **Vr**.

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

8

Структурная часть реляционной модели

Переменной VARr называется именованный контейнер, который может содержать любое допустимое значение **Vr**. VARr – это имя таблицы.



9

Ключи

Первичный ключ, внешний ключ.

Ключи используют для:

1. Обеспечения уникальности записей
2. Упорядочивания записей
3. Ускорения работы с таблицами
4. Организации связывания таблиц.

У внешнего ключа дочерней таблицы тип данных такой же как у первичного ключа главной (родительской) таблицы.

Структурная часть реляционной модели

Степень отношения (арность, мощность заголовка) – количество атрибутов отношения.

Кардинальность – количество кортежей отношения.

Первичный ключ – главный ключевой элемент, однозначно идентифицирующий строку в таблице.

Внешний ключ – ключевой элемент подчиненной (внешней, дочерней) таблицы, значение которого совпадает со значением первичного ключа главной (родительской) таблицы.

Связь – способ, которым информация в одной таблице связана с другой.

Схема реляционной базы данных – набор пар <имя VARr, Nr>, (имя переменной отношения + заголовок).

Первичный ключ

Первичный ключ называют ключом отношения, ключевым атрибутом. Ключи бывают **простые** и **сложные** (составные).

Простой ключ – поле любого типа. Например, в таблице ФИРМЫ название фирмы может быть первичным ключом, если нет фирм с одинаковым названием.

Простой ключ может быть **автоинкрементным**, когда уникальность ключа определяется СУБД. Например, в MySQL, таблицы типа MyIsam и InnoDB – это поле целого типа, в Access это поле - счетчик, тип поля – длинное целое.

Первичный ключ

MyIsam и InnoDB в MySQL, счетчик в Access.

<u>Фирм_Ключ</u>	<u>НазвМагазина</u>
------------------	---------------------

1	Рим
2	Будмен
3	Дигма

Уничтожаем запись с Дигма. Вставляем новое название фирмы (Класс), получим:

<u>Фирм_Ключ</u>	<u>НазвМагазина</u>
------------------	---------------------

1	Рим
2	Будмен
4	Класс

Ключи естественные и суррогатные

Суррогатный ключ – это идентификатор строки, который не имеет никакого отношения к атрибутам данной строки. Он делает всю строку уникальной.

Пример суррогатного ключа: integer identity или GIUD.

До сих пор идут дебаты о целесообразности применения натуральных или суррогатных ключей.

1. Integer autoincrement (SERIAL) – целочисленная последовательность. Если односерверная БД – проблем нет. Если данные добавляются на разных серверах, а потом сливаются, например в репликации, возникает проблема совмещения. Приходится выделять диапазон номеров для каждого сервера и отслеживать эту систему.

Составной ключ

Составной ключ может состоять из нескольких атрибутов.

БД “Рецепты”. Есть несколько форм, нумерация форм 1, 2, 3, и т.д.

На каждой форме есть кнопки, нумерация кнопок на каждой форме 1, 2, 3, и т.д. Если хотим создать таблицу названий кнопок, нам нужно однозначно определить кнопку. Для такой таблицы выбран составной ключ, состоящий из двух полей: номер формы и номер кнопки. Сочетание этих двух полей будет уникальным.

Ключи естественные и суррогатные

GUID (Globally Unique Identifier) — статистически уникальный 128-битный идентификатор. Его главная особенность — уникальность, которая позволяет создавать расширяемые сервисы и приложения без опасения конфликтов, вызванных совпадением идентификаторов. Общее количество уникальных ключей настолько велико (2128 или $3,4028 \times 10^{38}$), что вероятность того, что в мире будут независимо сгенерированы два совпадающих ключа, крайне мала.

В тексте GUID записывается в виде строки из тридцати двух шестнадцатеричных цифр, разбитой на группы дефисами и опционально окружённой фигурными скобками: 6F9619FF-8B86-D011-B42D-00CF4FC964FF

Ключи естественные и суррогатные

GUID плюсы:

1. Уникальный по каждой таблице, БД, серверу.
2. Позволяет легко слияние записей из различных БД.
3. Позволяет простое распространение баз данных на нескольких серверах

GUID минусы:

1. Колоссальные объемы (в 4 раза больше, чем 4 байт).
2. Могут быть серьезные последствия для производительности и хранения.
3. Тяжело и громоздко отлаживать.
4. Сгенерированные GUID должны быть частично последовательными, чтобы обеспечить производительность и включить использование кластеризованных индексов

Связи

Связь – это способ, которым таблицы связываются между собой. Между таблицами могут устанавливаться бинарные (2 таблицы), тернарные (3 таблицы) и, в общем случае, n-арные связи.

При связывании **двух** таблиц выделяют основную (родительскую) и подчиненную (дочернюю) таблицы. Суть связывания состоит в установлении соответствия полей связи основной и дополнительной таблиц.

Внешний ключ (Foreign key) в дочерней таблице – ссылка на первичный ключ родительской таблицы. У внешнего ключа дочерней таблицы тип данных такой же как у первичного ключа главной (родительской) таблицы.

Ключи естественные и суррогатные

Практические правила проектирования ключей:

1. Натуральный ключ составной – не годится.
2. Данные, которые будут в натуральном ключе, имеют хоть малейшую вероятность изменения – не годится.
3. Натуральный ключ громоздкий, и негативно влияет на производительность – не годится.
4. Суррогатный ключ можно превратить в используемый в бизнес-процессах идентификатор.

В реальной жизни крайне мало случаев, где лучше использовать натуральный ключ.

Связи

Между двумя таблицами могут устанавливаться следующие основные виды связи:

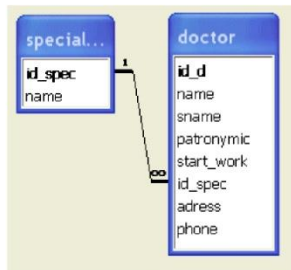
Один к одному (1 : 1); **Один ко многим** (1 : M);

Многие к одному (M : 1); **Многие ко многим** (M : N).

Связь (1:1) – вторая таблица является продолжением первой. Значения первичных ключей в этих таблицах одинаково.

Связь (1:M) – одной записи основной таблицы соответствует несколько записей дочерней таблицы.

Связь (M : 1) – нескольким записям основной таблицы ставится в соответствие одна запись дополнительной таблицы.



Связи

special...	doctor : таблица
id_spec	id_d
name	name
	sname
	patronymic
	start_work
	id_spec
	adress
	phone
1 терапевт	1 Петр Петров Петрович 12.12.2000 1 адрес1 7778866
2 кардиолог	2 Иван Иванов Иванович 30.09.1998 2 адрес2 4569842
3 хирург	3 Степан Степанов Степанович 25.05.2006 3 адрес3 7894561
4 окулист	4 Геннадий Геннадиев Геннадиевич 20.10.1976 4 адрес4 1538975
5 лор	5 Василий Васильев Васильевич 10.10.2005 5 адрес5 1122334
нетчик)	чик)

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

21

Связи

Связь многие ко многим (М : N) возникает, когда нескольким записям основной таблицы ставится в соответствие несколько записей дочерней таблицы.

Связь многие ко многим в реляционных БД не поддерживается. Эта связь между таблицами реализуется за счет введения промежуточной таблицы, которая дает возможность свести связь многие ко многим к связям один ко многим или многие к одному.

Промежуточная таблица называется ассоциативной сущностью.

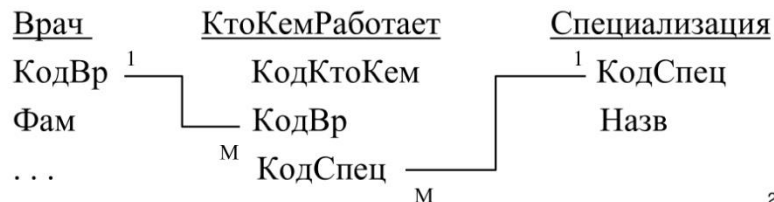
ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

22

Связи

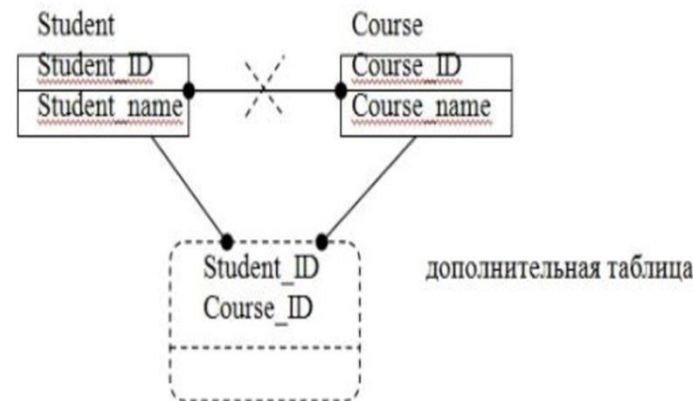
Связь многие ко многим (М:М) в реляционных БД не поддерживается. Эта связь между таблицами реализуется за счет введения промежуточной таблицы, которая дает возможность свести связь многие ко многим к связям один ко многим или многие к одному.

Пусть врач может иметь несколько специализаций и каждая специализация может принадлежать многим врачам



23

Тернарная связь

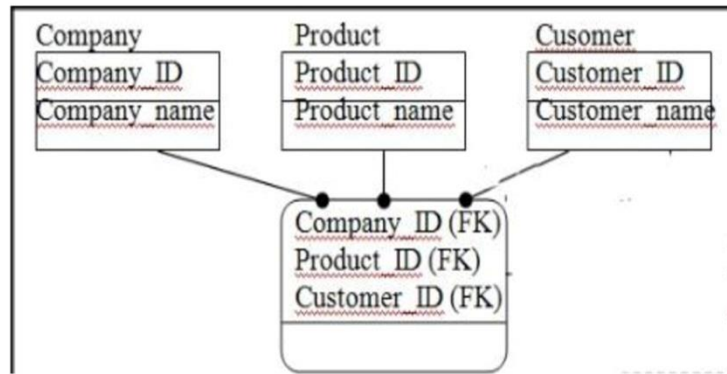


Две родительские таблицы и одна подчиненная.

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

24

Энарная сувязь



Три родительские таблицы и одна подчиненная.

Целостная часть

- целостность сущности,
- целостность ссылок.

Целостность сущности – у любого отношения должен существовать первичный ключ, значение первичного ключа не может быть неопределенным.

Неопределенное значение – **NULL**.

NULL не принадлежит никакому типу данных и может присутствовать среди значений любого атрибута, определенного на любом типе данных. Целостность сущности – первичный ключ не может иметь NULL значение.

Рекурсивная связь (унарная)

Sotrudniki : таблица		
	Имя поля	Тип данных
1	id_sotr	Счетчик
2	FIO	Текстовый
3	dolg	Текстовый
4	podch	Числовой

Sotrudniki : таблица				
	id_sotr	FIO	dolg	podch
	1	Степанов Б.И.	зав.отделом	
	2	Иванов В.А.	рук.группы разработчиков	
	3	Кулик В.П.	вед.программист	
	4	Колесова Н.В.	млад.программист	
	5	Иванченко Д.С.	ст.лаборант	
	6	Ивановская В.Д.	рук.группы сопровождения	
	7	Куликов В.П.	вед.программист	
	8	Колесов В.В.	млад.программист	
	9	Иваненко Г.С.	ст.лаборант	
► Счетчик)				

Связь таблицы сама с собой.

Целостная часть

Целостность по ссылкам или целостность внешнего ключа – для каждого значения внешнего ключа должен найтись кортеж с таким же значением первичного ключа, либо значение внешнего ключа должно быть полностью неопределенным.

Если в таблице «doctor» для врача указана специализация, то в таблице «specialization» обязательно должна существовать запись с таким номером первичного ключа.

Реализация целостной части

Целостность сущности – СУБД гарантирует отсутствие кортежей с **одним и тем же значением** первичного ключа и **запрещает неопределенные значения** первичного ключа.

Целостность по ссылкам – три подхода:

1. **Запрет удаления кортежа**, для которого существуют ссылки. Нельзя удалить специализацию, пока есть ссылки на нее у врачей.
2. **Удаление кортежа** в родительской таблице, в дочерней во всех ссылающихся кортежах **внешний ключ неопределенный**. Если удалили ЛОР, то у всех записей, ссылающихся на ЛОР, значение внешнего ключа становится NULL.

ХНУ ім.В.Н.Каразіна, ФКН,
Лазурик В.М.

29

Этапы проектирования

Методология проектирования - структурированный подход, предусматривающий использование специализированных процедур, технических приемов, инструментов, документации и нацеленный на поддержку и упрощение процесса проектирования.

В предлагаемой методологии проектирования баз данных весь процесс разработки разделяется на три основные фазы: **концептуальное**, **логическое** и **физическое** проектирование.

ХНУ ім.В.Н.Каразіна, ФКН,
Лазурик В.М.

31

Реализация целостной части

3. **Каскадное удаление** – при удалении кортежа из родительской таблицы, в дочерней автоматически удаляются все ссылающиеся кортежи. Если удалили специализацию ЛОР, автоматически удаляются все записи из таблицы ВРАЧИ, ссылающиеся на ЛОР.

Развитые СУБД позволяют настроить режим реализации целостной части.

ХНУ ім.В.Н.Каразіна, ФКН,
Лазурик В.М.

30

Концептуальное проектирование

Процедура конструирования **конкретных моделей данных**, которые определяются задачами, решаемыми базой данных. Результат – концептуальная модель данных предприятия, независимая от любых деталей реализации.

Этап 1. Создание локальных концептуальных моделей данных исходя из требований каждого типа пользователей БД.

Этап 2. Определение типов сущностей.

Этап 3. Определение типов связей.

Этап 4. Определение атрибутов и связей.

Этап 5. Определение доменов атрибутов.

Этап 6. Определение первичных ключей.

Этап 7. Создание диаграммы "сущность-связь".

Этап 8. Обсуждение модели с конечными пользователями.

32

Логическое проектирование

Конструирование информационной модели предприятия на основе существующих конкретных моделей данных.

Результат – преобразовании отдельных концептуальных моделей в единую логическую модель данных предприятия

Этап 1. На базе локальных концептуальных моделей – создание логических локальных моделей (а) Устранение особенностей, несовместимых с реляционной моделью; б) Определение и нормализация отношений; в) Определение транзакций, проверка моделей на обеспечение целостности данных .

Этап 2. Слияние локальных логических моделей данных в единую глобальную модель данных, проверка ее.

Этап 3. Проверка возможностей расширения модели в будущем.

Этап 4. Обсуждение глобальной логической модели данных с пользователями.

Техническое задание «Интернет провайдеры»

Жителям города подключен Интернет. Известно, имеет ли человек подключенный Интернет, каким пакетом и услугами какого провайдера пользуется.

Требуется:

1. Спроектировать таблицы БД, заполнить.
2. Запрос «Жители без Интернета».
3. Запрос «Жители, у кого указанный провайдер/пакет».
4. Запрос «Жители без Интернета в промежутки времени».
5. Средства для просмотра жителей, которые пользуются услугами Интернет провайдеров, по их адресам.

Физическое проектирование

Перенесение логической модели БД в среду целевой СУБД.

Этап 1. Перенос глобальной логической модели данных в среду целевой СУБД.

Этап 2. Проектирование основных таблиц в среде целевой СУБД.

Этап 3. Реализация **бизнес-правил** предприятия в целевой СУБД.

Этап 4. Разработка пользовательских представлений (View).

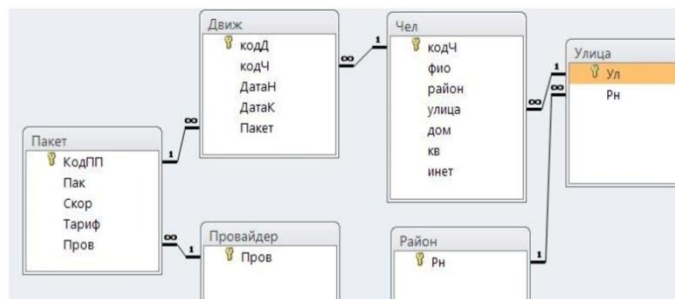
Этап 5. Разработка механизмов защиты БД и авторизации пользователей.

Уточнение технического задания

Обсуждение с заказчиком.

1. В список жителей попадают только те люди, которые заключают с провайдером контракт.
2. Человек может переходить с пакета на пакет, менять провайдеров, отключать Интернет вообще.
3. Важны сведения о подключениях к конкретному провайдеру на текущий момент.
4. Важны сведения о тех клиентах, которые заключили контракт с провайдером, но в указанном диапазоне дат его услугами не пользуются. Для анализа ситуации.

Проектирование. Вариант 1.



Шаг 1. Выделение базовых родительских сущностей и их свойств: «Провайдер», «Район»

Шаг 2. Зависимые, дочерние сущности, их свойства: Бинарные связи: «Пакет», «Улица», «Чел». В таблице «Чел» присутствует логическое поле «Инет». Тернарная связь. «Движ» имеет две родительские таблицы «Пакет» и «Чел».

Простая БД. Простые запросы. Сложные формы.

37

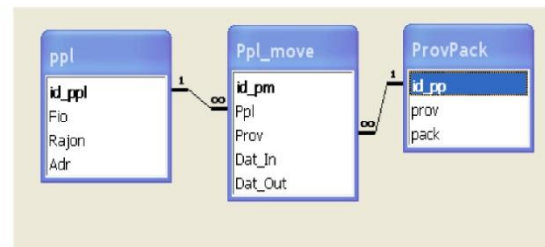
Чел						
кодЧ	фио	район	улица	дом	кв	инет
1	Иванов	Фрунзенский	Велозаводская	1	12	<input checked="" type="checkbox"/>
2	Петров	Фрунзенский	Корчагинцев	5	23	<input checked="" type="checkbox"/>
3	Сидоров	Московский	Московский проспект	126	120	<input checked="" type="checkbox"/>
4	Егоров	Московский	Светлая	9	10	<input type="checkbox"/>
5	Саркисян	Московский	Уборевича	5	25	<input type="checkbox"/>
6	Никакущий	Дзержинский	Сумская	1	1	<input type="checkbox"/>

Пакет				
КодПП	Пров	Пак	Скор	Тариф
1	Воля	Минимальный	1	120
2	Воля	Максимальный	5	250
3	УкрНет	100+	100	11
4	МаксНет	32-	23	24
5	МаксНет	Общий	55	55
6	Воля	100+	75	11
7	МаксНет	100++	99	1

Движ				
кодД	кодЧ	ДатаН	ДатаК	Пакет
2	2	01.09.2014	11.09.2014	3
3	3	01.01.2014	31.12.2014	1
7	1	30.08.2014		4
8	4	10.10.2012	25.09.2014	6
10	2	20.09.2014		7

39

Проектирование. Вариант 2.



Шаг 1. Выделение базовых родительских сущностей и их свойств: «ProvPack», «Ppl». Для района, улицы, провайдера и пакетов используются таблицы-справочники с текстовыми полями на этапе заполнения пользовательских форм.

Логического поля «Есть ли Инет», как характеристики человека нет. Если отсутствует Dat_Out – человек работает с Инетом.

Сложные запросы с подзапросами.

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

38

Простые формы

Главная

Район

Улица

Новый житель

Установить/убрать Инет

Жители без Инета

Провайдер+Пакет --> Жители

Улица

Район:

Улица:

Велозаводская

Московский проспект

Светлая

Уборевича

Вызов формы из главной – макрос «Открыть форму».

При выборе из списка названия района, следующее поле со списком содержит только те улицы, которые входят в этот район.

ХНУ ім.В.Н Каразіна, ФКН,
Лазурик В.М.

40