



Лабораторная работа №2.

Проектирование информационной модели  
методом декомпозиции

# Содержание

1. Разработка словесного описания всех атрибутов электронной системы библиотеки и связей между ними.
2. Вывод ФЗ между атрибутами универсального отношения
3. Преобразование структуры отношения в 1НФ
4. Схема функциональных зависимостей
5. Получение минимального покрытия
6. Список всех ФЗ минимального покрытия
7. Список потенциальных ключей и детерминантов
8. Кандидаты среди ФЗ для осуществления проекции
9. Определение ФЗ для проекции
10. Декомпозиция отношения
11. Дальнейшая декомпозиция
12. Окончательный список нормализованных отношений проектного набора базы данных
13. Проверка списка нормализованных отношений проекта базы данных на избыточность

## Формулировка задания

Разработать электронную библиотечную систему для отслеживания движения книг, которая содержит следующую информацию: номер читательского билета студента, фамилия студента, код факультета, где учится студент, телефон факультета, шифр выданной книги, название книги, фамилию первого автора, дату выхода, город издательства, наименование издательства, количество страниц, количество экземпляров книги, наличие книги, дата выдачи книги, дата возврата книги, дата заказа книги.

### 1. Разработка словесного описания всех атрибутов электронной системы библиотеки и связей между ними.

Пример выполнения.

*Таблица 2.1. Набор атрибутов предметной области*

SNum	Номер читательского билета студента. Целое значение, уникальное для каждого студента университета.
SName	Фамилия студента. Каждый студент имеет только одну фамилию, но не исключено, что одну фамилию носят несколько студентов.
BCode	Шифр книги в библиотеке. Сочетание букв и цифр, характеризующее книгу и место ее хранения – уникален.
title	Название книги. Каждая книга имеет одно название. Разные книги могут называться одинаково. Текст
date	Дата выхода книги. Уникальна для данной книги. Разные книги могут иметь одну дату выхода. Дата/время
volume	Количество страниц в книге. Для каждой книги есть одно количество страниц. У разных книг может быть одинаковое число страниц.
copies	Количество экземпляров книги в библиотеке. Уникальное число для одной книги, может быть одинаковым для разных книг.
available	Наличие книги (логическая)
AName	Фамилия первого автора. Каждый автор имеет только одну фамилию, но не исключено, что одну фамилию носят несколько авторов. Текст.
PName	Название издательства. Каждое издательство имеет только одно название, но не исключено, что одно название носят несколько издательств. Строка.

Продолжение табл 2.1.

town	Город издательства. В одном городе может быть несколько издательств, одно издательство может располагаться в нескольких городах.
FCode	Код факультета, где обучается студент. Уникальный.
FPhone	Телефон факультета, где обучается студент.
issue_date	Дата выдачи книги. Несколько книг могут быть выданы в один день.
return_date	Дата возврата книги. Разные книги могут быть выданы на один срок.
order date	Дата заказа книги

## 2. Вывод ФЗ между атрибутами универсального отношения

На основе анализа описаний атрибутов вывести ФЗ между атрибутами универсального отношения.

Пример выполнения.

- Номера читательских билетов студентов являются уникальными. Если известен номер читательского билета SNum, с ним может быть связана только одна фамилия SName: SNum -> SName. Обратное неверно. ФЗ SName -> SNum неправильная, т.к. несколько студентов могут иметь одну фамилию.
- С номером читательского билета SNum может быть связан код только одного факультета FCode: SNum-> FCode. Обратное неверно. ФЗ FCode -> SNum неправильная, т.к. предполагается, что студент учится на одном факультете.
- С номером читательского билета SNum может быть связан телефон FPhone только того факультета, на котором учится студент: SNum-> FPhone. Обратное неверно. ФЗ FPhone -> SNum неправильная, т.к. предполагается, что студент учится на одном факультете.
- FCode и FPhone связаны взаимной ФЗ, предполагая, что на каждом факультете есть только один телефон, по которому могут звонить библиотечные работники.
- Дата выдачи книги issue\_date и дата ее возврата return\_date функционально зависят от номера студента, который ее получил, и шифра книги BCode : SNum, BCode -> issue\_date, SNum, BCode -> return\_date. Обратное неверно, так как эти же

даты могут соответствовать нескольким студентам и нескольким книгам.

- Дата заказа книги `order_date` функционально зависит от номера студента, который ее заказал, и шифра книги `BCode` : `SNum, BCode -> order_date`. Обратное неверно, так как эта же даты могут соответствовать нескольким студентам и нескольким книгам.
- Все атрибуты описания книги зависят от шифра книги, т.е. `BCode -> AName, title, town, PName, date, volume, copies, available, .`

### 3. Преобразование структуры отношения в 1НФ

Для того чтобы таблица, приведенная могла считаться отношением реляционной базы данных, ее надо преобразовать так, чтобы она не содержала групп повторения, т.е. чтобы каждый элемент кортежа имел атомарное значение (отношение в 1НФ). Обычно для этого используется простая вставка. В результате в отношении появляется большой объем избыточных данных. Но таблица будет представлять собой экземпляр *корректного* отношения. Его называют *универсальным отношением* проектируемой БД. В универсальное отношение включаются все представляющие интерес атрибуты, и оно может содержать все данные, которые предполагается размещать в БД. Универсальное отношение может служить отправной точкой при проектировании БД.

Пример выполнения

Назовем универсальное отношение ДВИЖЕНИЕ\_КНИГ:

```
BOOKS_MOVE(BCode, SNum, AName, title, town, PName, date, volume, copies, available, SName, FCode, FPhone, issue_date, return_date, order_date)
```

### 4. Схема функциональных зависимостей

Схема строится на полном списке ФЗ.

Пример выполнения

Запишите полный список ФЗ.

`SNum -> SName`

`SNum-> FCode`

`FCode <-> FPhone`

SNum, BCode -> issue\_date  
 SNum, BCode -> return\_date  
 SNum, BCode -> order\_date  
 BCode -> AName, title, town, PName, date, volume, copies, available

Введем еще два уникальных атрибута:

A\_ID – идентификатор автора, A\_ID -> AName;

P\_ID – идентификатор издательства, P\_ID -> town, PName.

Эти идентификаторы будут использоваться для корректного ввода информации из таблиц-справочников.

Схема ФЗ будет иметь вид как на рис. 4.1.

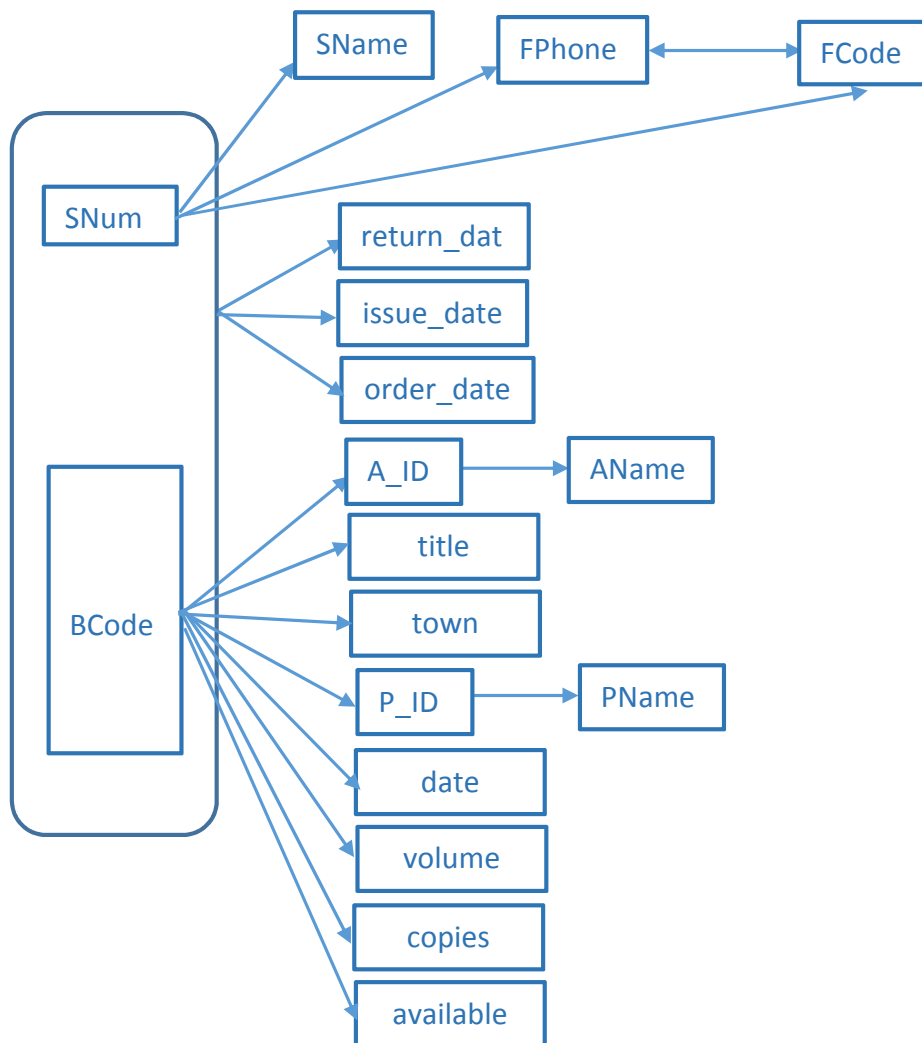


Рис. 4.1. Схема функциональных зависимостей

## 5. Получение минимального покрытия

Минимальное покрытие получается путем удаления избыточных функциональных зависимостей. К последним относятся в частности транзитивные ФЗ.

Пример выполнения

FCode зависит от SNum, FPhone зависит от FCode, тогда SNum → FPhone транзитивная, и ее можно удалить. Результат – на рис. 5.1.

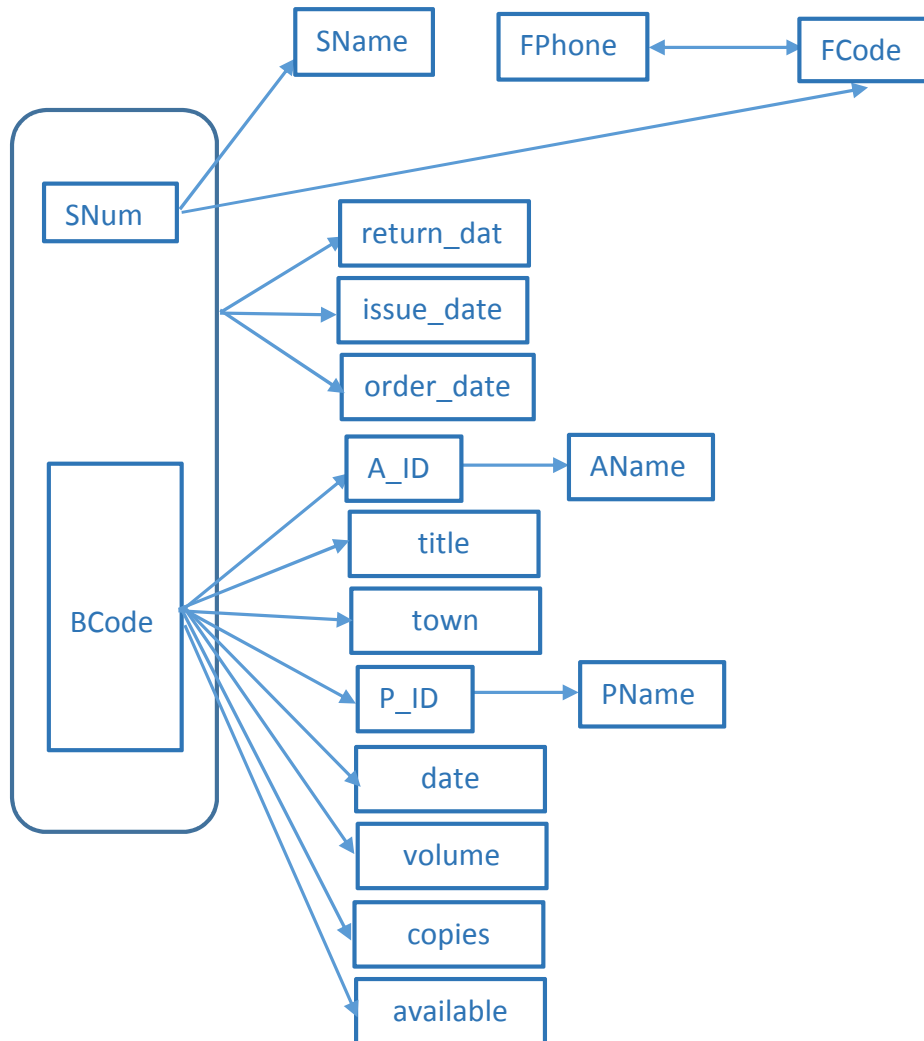


Рис. 5.1. Минимальное покрытие

## 6. Список всех ФЗ минимального покрытия

Выпишите список всех ФЗ минимального покрытия.

Пример выполнения

Исходя из схемы ФЗ минимального покрытия (рис. 4.2), список всех функциональных зависимостей будет иметь вид как в табл. 2.2.

Таблица 2.2. Минимальное покрытие

<SNum>	->	<SName>
<SNum>	->	<FCode>
<FCode>	->	<FPhone>
<FPhone>	->	<FCode>
<SNum, BCode>	->	<issue_date>, <return_date>, <order_date>
<BCode>	->	<A_ID>, <title>, <town>, <P_ID>, <date>, <volume>, <copies>, <available>
<A_ID>	->	<AName>
<P_ID>	->	<PName>

## 7. Список потенциальных ключей и детерминантов

Выпишите список потенциальных ключей и детерминантов.

Пример выполнения

Таблица 2.3. Список потенциальных ключей и детерминантов

Потенциальные ключи	Детерминанты
< SNum, BCode >	< SNum, BCode >
	<SNum>
	<FCode>
	<FPhone>
	<BCode >
	<A ID>
	<P ID>

## 8. Кандидаты среди ФЗ для осуществления проекции

Кандидатами для последующей проекции являются функциональные зависимости, где детерминант не является потенциальным ключом.

Пример выполнения

Рассмотрев детерминанты и потенциальные ключи отношения, видим, что имеются следующие детерминанты, не являющиеся возможными ключами (табл. 2.4).



<SNum>	->	<SName>
<SNum>	->	<FCode>
<FCode>	->	<FPhone>
<FPhone>	->	<FCode>
<BCode>	->	<A_ID>, <title>, <town>, <P_ID>, <date>, <volume>, <copies>, <available>
<A_ID>	->	<AName>
<P_ID>	->	<PName>

Таблица 2.4. Список потенциальных ключей и детерминантов

## 9. Определение ФЗ для проекции

Простым правилом для выбора ФЗ для проекции может служить поиск цепочки ФЗ вида:

$$A \rightarrow B \rightarrow C$$

с последующим использованием для проекции крайней правой зависимости.

Пример выполнения

Из табл. 2.4 выберем цепочку ФЗ:

SNum->FCode->FPhone

Тогда для проекции выбираем FCode->FPhone.

## 10. Декомпозиция отношения

Используя известный алгоритм, произведите декомпозицию отношения.

Пример выполнения

Исходное отношение:

BOOKS\_MOVE (BCode, SNum, AName, title, town, PName, date, volume, copies, available, SName, FCode, FPhone, issue\_date, return\_date, order\_date)

Результат декомпозиции – отношения R1 и R2:

R1(BCode, SNum, AName, title, town, PName, date, volume, copies, available, SName, FCode, issue\_date, return\_date, order\_date)

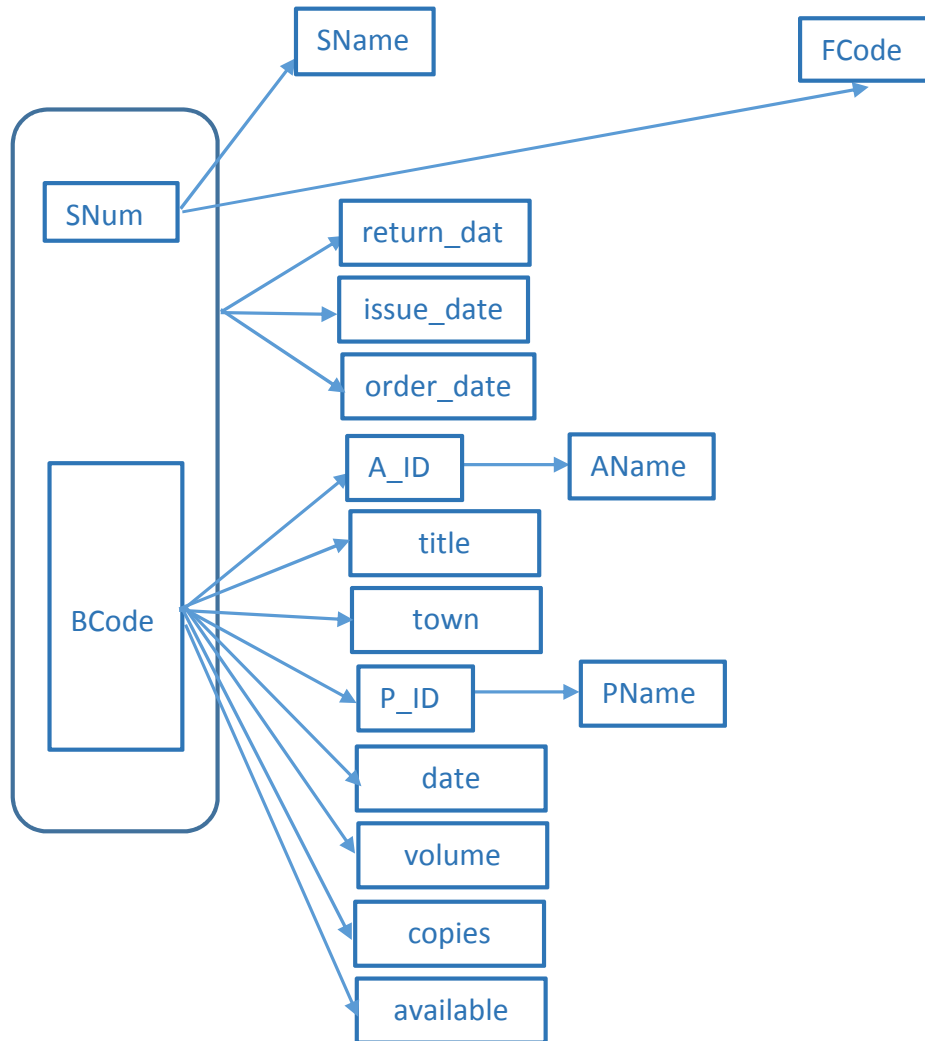


Рис. 10.1. Схема ФЗ отношения R1

Таблица 2.5. Потенциальные ключи и детерминанты отношения R1

Потенциальные ключи	Детерминанты
< SNum, BCode >	< SNum, BCode >
	<SNum>
	<BCode >
	<A_ID>
	<P_ID>

R2 (FCode, FPhone)



Рис. 10.2. Схема ФЗ отношения R2

Таблица 2.5. Потенциальные ключи и детерминанты отношения R2

Потенциальные ключи	Детерминанты
< <u>FCode</u> >	<FCode >
< <u>FPhone</u> >	<FPhone>

Отношение R2 (FCode, FPhone) находится в НФБК, так как все его детерминанты являются потенциальными ключами. Поэтому отношение R2 не нуждается в дальнейшей декомпозиции.

**Поясните - почему отношение R1 не находится не только в НФБК, но и в 2НФ и нуждается в дальнейшей декомпозиции?**

## 11. Дальнейшая декомпозиция

Произведите дальнейшую декомпозицию отношения R1 (рис. 10.1, табл. 2.5).

Пример выполнения

Результат декомпозиции – отношения R3, R4, R5:

R3 (SNum, BCode, issue\_date, return\_date, order\_date)

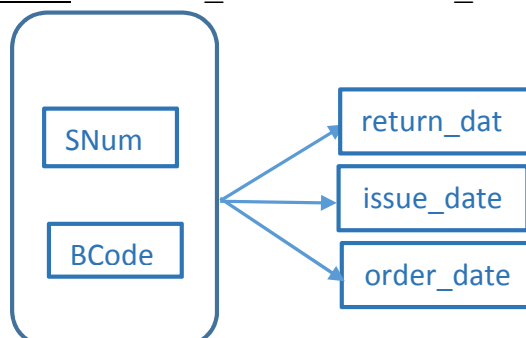


Рис. 11.1. Схема ФЗ отношения R3

Таблица 2.6. Потенциальные ключи и детерминанты отношения R3

Потенциальные ключи	Детерминанты
---------------------	--------------

< <u>SNum</u> , <u>BCode</u> >	< SNum, BCode >
--------------------------------	-----------------

R4 (<SNum>, <SName>, <FCode>)

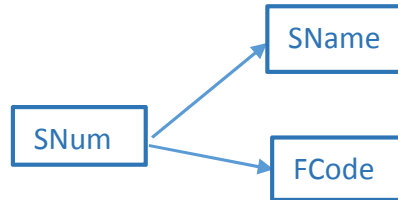


Рис. 11.2. Схема ФЗ отношения R4

Таблица 2.7. Потенциальные ключи и детерминанты отношения R4

Потенциальные ключи	Детерминанты
< <u>SNum</u> >	<SNum>

R5 (<BCode>, <A\_ID>, <AName>, <title>, <town>, <P\_ID>, <PName>, <date>, <volume>, <copies>, <available>)

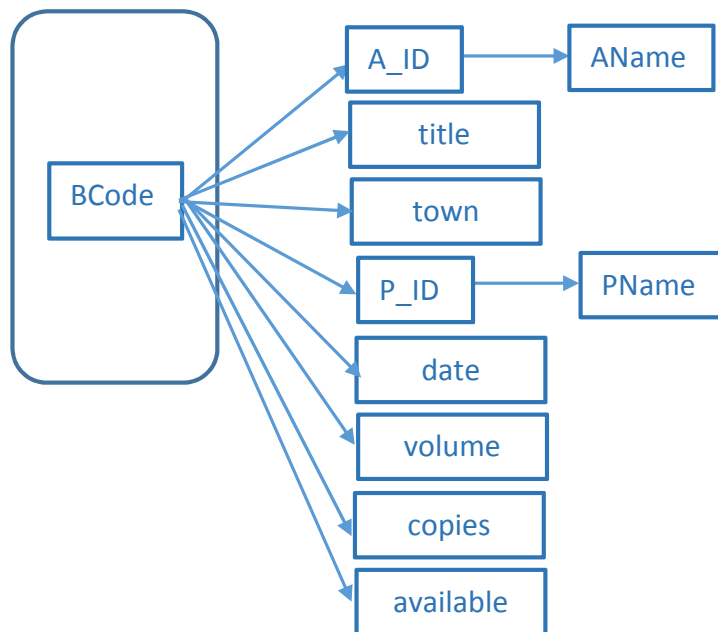


Рис. 11.3. Схема ФЗ отношения R5

Таблица 2.8. Потенциальные ключи и детерминанты отношения R5

Потенциальные ключи	Детерминанты
< <u>BCode</u> >	< <u>BCode</u> >

	<A_ID>
	<P_ID>

Произведите дальнейшую декомпозицию отношения R5. Необходимость этого видна из рис. 11.3 и табл. 2.8).

Результат декомпозиции R5 – отношения R6, R7, R8:

R6(<BCode>, <A\_ID>, <title>, <town>, <P\_ID>, <date>, <volume>, <copies>, <available>)

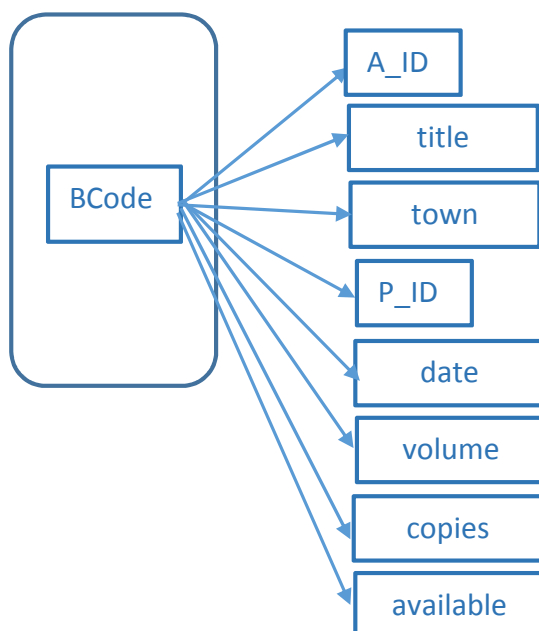


Рис. 11.4. Схема ФЗ отношения R6

Таблица 2.9. Потенциальные ключи и детерминанты отношения R6

Потенциальные ключи	Детерминанты
<BCode>	<BCode>

R7(A\_ID, AName)



Рис. 11.5. Схема ФЗ отношения R7

Таблица 2.10. Потенциальные ключи и детерминанты отношения R7

Потенциальные ключи	Детерминанты
<A_ID>	<A_ID>

R8(P\_ID, PName)



Рис. 11.6. Схема ФЗ отношения R8

Таблица 2.11. Потенциальные ключи и детерминанты отношения R8

Потенциальные ключи	Детерминанты
<P_ID>	<P_ID>

## 12. Окончательный список нормализованных отношений проектного набора базы данных

Пример выполнения

R3(SNum, BCode, issue\_date, return\_date, order\_date)

R2(FCode, FPhone)

R4(<SNum>, <SName>, <FCode>)

R6(<BCode>, <A\_ID>, <title>, <town>, <P\_ID>, <date>, <volume>, <copies>, <available>)

R7(A\_ID, AName)

R8(P\_ID, PName)

Разобъем отношение R3 по смыслу на два отношения: ВЫДАЧА и БРОНЬ:

ISSUE(SNum, BCode, issue\_date, return\_date)

**RESERV(SNum, BCode, order\_date)**

Присвоим отношениями смысловые наименования:

Таблица 2.12. Проектный набор нормализованных отношений

Русское имя	Отношение
-------------	-----------

СТУДЕНТ	STUDENT ( <u>SNum</u> , SName, FCode)
КНИГА	BOOK ( <u>BCode</u> , title, date, volume, copies, available, A ID, P ID)
ВЫДАЧА	ISSUE ( <u>SNum</u> , <u>BCode</u> , issue_date, return_date)
БРОНЬ	RESERV ( <u>SNum</u> , <u>BCode</u> , order_date)
АВТОР	AUTHOR (A ID, AName)
ИЗДАТЕЛЬСТВО	PUBLISH ( <u>P ID</u> , town, PName)
ФАКУЛЬТЕТ	DEPART ( <u>FCode</u> , FPhone)

### 13. Проверка списка нормализованных отношений проекта базы данных на избыточность

Пример выполнения

Табл. 2.13. Список ФЗ всех полученных отношений

ФУНКЦИОНАЛЬНАЯ ЗАВИСИМОСТЬ			ОТНОШЕНИЕ
<SNum>	->	<SName>	STUDENT
<SNum>	->	<FCode>	STUDENT
<BCode>	->	title, date, volume, copies, available, A ID, P ID	BOOK
<SNum, BCode >	->	issue_date, return_date	ISSUE
<SNum, BCode >	->	order_date	RESERV
<A_ID>	->	<AName>	AUTHOR
<P_ID>	->	<PName>	PUBLISH
<FCode>	->	<FPhone>	DEPART

- a) Одна и та же ФЗ не появляется более чем в одном отношении. Кроме того, набор ФЗ, полученный в результате проектирования, в точности совпадает с набором из минимального покрытия, полученного перед началом декомпозиции (см. п.6)
- b) нет избыточного отношении все атрибуты которого могут быть найдены в другом отношении проектного набора;
- c) нет избыточного отношении все атрибуты которого могут быть найдены в отношении, которое может быть получено из других отношений проектного набора с помощью одной или нескольких операций объединения.