

**Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное  
учреждение высшего профессионального образования  
«Севастопольский государственный университет»**

## **МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

к лабораторной работе

**“Исследование методов рефакторинга структуры и качества данных”**  
по дисциплине

**«Рефакторинг программного обеспечения»**

для студентов, обучающихся по направлению  
**09.03.02 “Информационные системы и технологии”**  
очной и заочной форм обучения

**Севастополь  
2018**

## 1. ЦЕЛЬ РАБОТЫ

Исследование основных методов рефакторинга структуры баз данных и качества хранимых данных. Приобретение навыков переработки баз данных с целью упрощения доступа к данным.

## 2. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Рефакторингом базы данных называется простое изменение в схеме базы данных, способствующее улучшению проекта базы данных и вместе с тем обеспечивающее сохранение функциональной и информационной семантики базы данных; иными словами, проведение операций рефакторинга не должно приводить к добавлению новых функциональных средств или нарушению работы существующих, а также не должно иметь своим следствием добавление новых данных или изменение смысла существующих данных.

Признаками необходимости рефакторинга структуры базы данных являются наличие:

- **многоцелевых столбцов** — использование столбца в нескольких целях увеличивает вероятность, что существует дополнительный код, который служит для обеспечения использования исходных данных “по назначению”, часто путем проверки значений в одном или нескольких других столбцах. В качестве примера можно привести столбец, в котором хранятся даты рождения клиентов и даты приема на работу сотрудников;
- **многоцелевых таблиц** — если какая-либо таблица используется для хранения данных о сущностях разных типов, то ее появление в базе является следствием упущения в проекте. В качестве примера можно назвать общую таблицу Customer, которая используется для хранения информации и о физических лицах, и о корпорациях. Недостатком подобного подхода является то, что для представления информации о физических лицах и

корпорациях используются разные структуры данных: для физических лиц необходимо хранить сведения о фамилии, имени и отчестве, а для корпораций — юридическое название. В подобной общей таблице Customer неизбежно появляются столбцы с NULL-значениями, относящимися к одним типам клиентов, но не к другим;

- **таблиц со слишком большим количеством столбцов** — если в таблице имеется много столбцов, это можно рассматривать как признак отсутствия слитности в структуре таблицы; **в этой таблице представлены данные, относящиеся к нескольким разным сущностям**. Предположим, что в таблице Customer находятся столбцы, предназначенные для хранения трех разных адресов (адреса поставки, адреса выставления счетов и адреса, применяемого в течение сезона) и нескольких номеров телефонов (домашний телефон, рабочий телефон, сотовый телефон). По-видимому, должна быть проведена нормализация этой структуры путем добавления таблиц PhoneNumber и Address;
- **многозначных столбцов** — в которых **в различных позициях представлено несколько разных фрагментов информации**. Например, как многозначный рассматривается столбец с идентификатором клиента, в котором первые четыре цифры идентификатора клиента обозначают головное отделение компании этого клиента, поскольку для выявления дополнительной информации приходится выполнять синтаксический анализ значений из этого столбца (допустим, чтобы узнать идентификатор головного отделения).

## 2.1. Рефакторинг структуры

### 2.1.1. Методы рефакторинга в рамках одной таблицы

- Разбиение столбца.

Эта операция применяется для разбиения столбца на один или несколько столбцов, находящихся в одной и той же таблице. Следует отметить, что если один или несколько новых столбцов должны появиться в другой таблице, то сначала необходимо применить операцию “Разбиение столбца”, а затем “Перемещение столбца”.

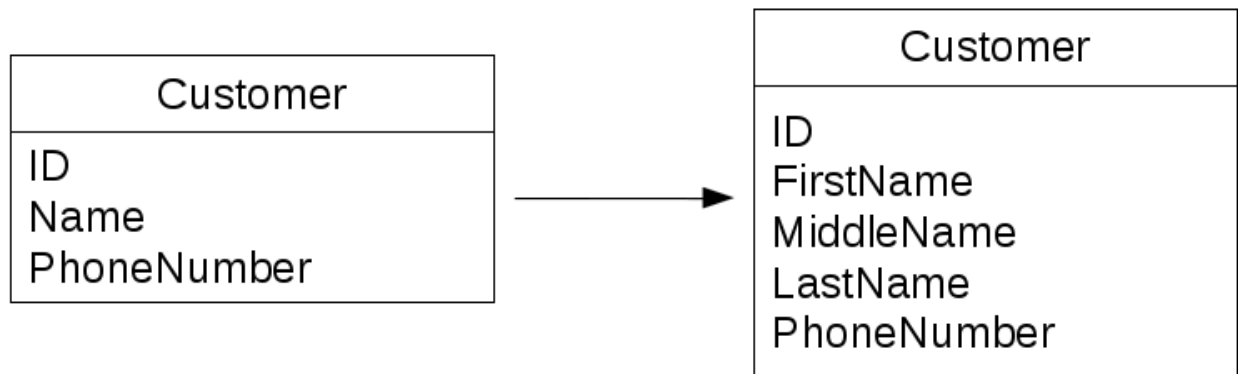


Рисунок 2.1 — разбиение столбца «Name» на составные части

- Перемещение столбца.

Эта операция обеспечивает перемещение столбца таблицы вместе со всеми данными в другую существующую таблицу.

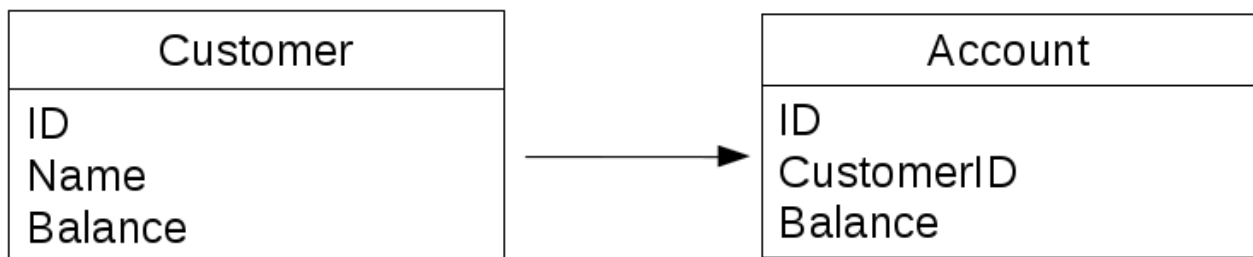


Рисунок 2.2 — перемещение столбца «Balance» в другую таблицу

- Удаление столбца.

В ходе анализа структуры приложения может оказаться, что столбец более не используется. В таком случае его удаляют.

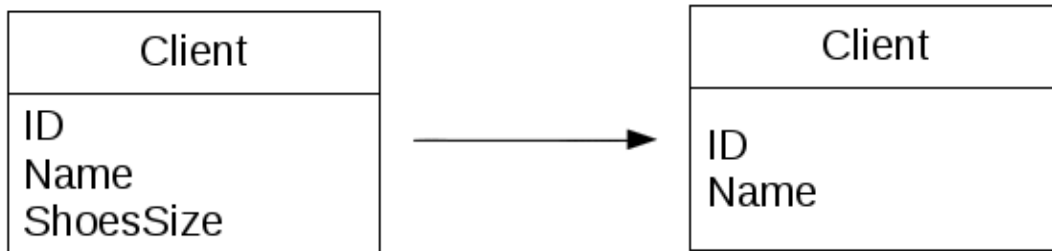


Рисунок 2.3 — удаление неиспользуемого столбца

- Переименование столбца.

Данная операция применяется, как правило, после разбиения многоцелевого столбца с целью улучшения понимания и доступа к данным таблицы, либо при смене СУБД, т. к. имя столбца может являться зарезервированным словом.

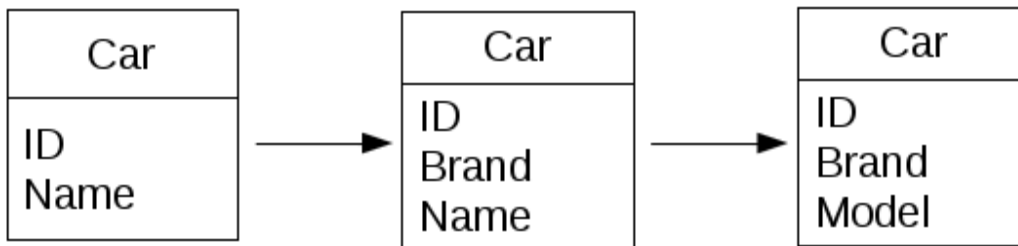


Рисунок 2.4 — разбиение и переименование столбца

- Слияние столбцов.

Рассматриваемая операция применяется в случае обнаружения идентичных столбцов, полученных в ходе рефакторинга, либо в случае предусмотренной чрезмерной гибкости, которая оказалась ненужной в ходе эксплуатации.

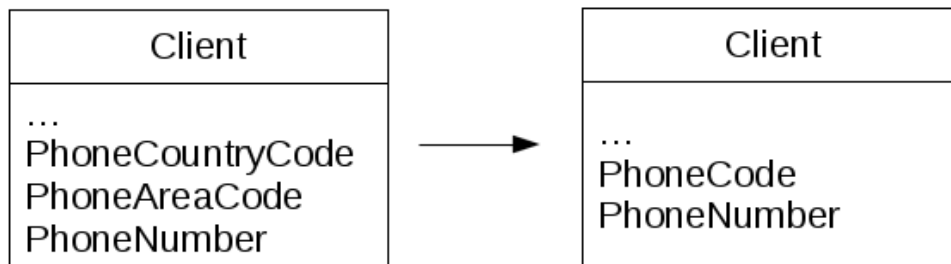
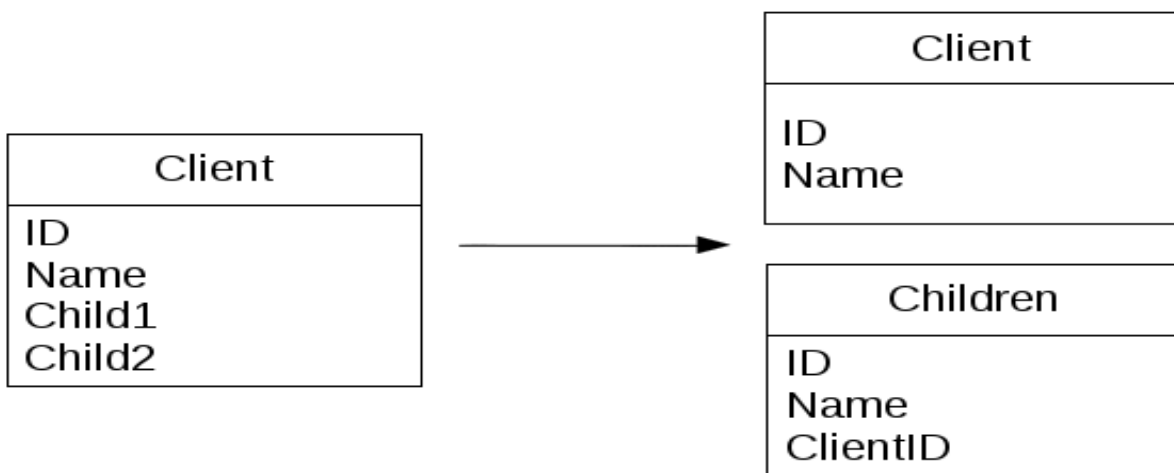


Рисунок 2.5 — слияние столбцов

### 2.1.2. Методы рефакторинга таблиц

- Разбиение таблицы.

Эта операция позволяет выполнить разбиение существующей таблицы по вертикали (например, по столбцам) на одну или несколько таблиц. Следует отметить, что если местом назначения столбцов, полученных в результате разбиения, фактически становится существующая таблица, то в

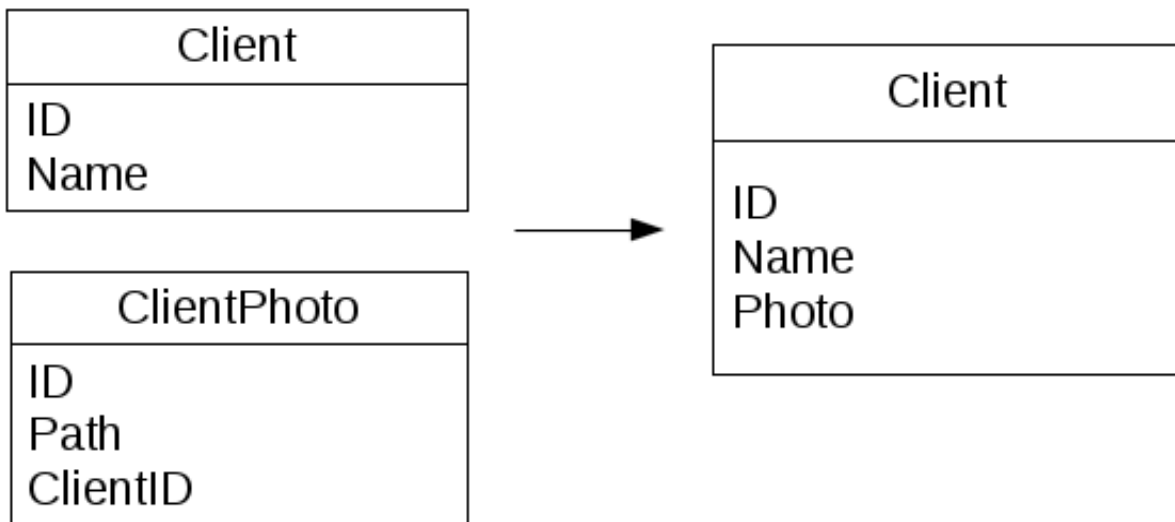


действительности применяется операция “Перемещение столбца”. Рисунок

2.6 — разбиение многоцелевой таблицы

- Слияние таблиц.

Данная операция применяется в случае обнаружения логически связанных таблиц, полученных в ходе рефакторинга, в случае обнаружения дубликатов таблиц, полученных в ходе эксплуатации, либо в случае предусмотренной чрезмерной гибкости, которая оказалась ненужной в ходе работы программного продукта.



- Рисунок 2.7 — слияние таблиц

Операция применяется для того, чтобы пояснить назначение таблицы и общую цель ее использования во всей схеме базы данных.

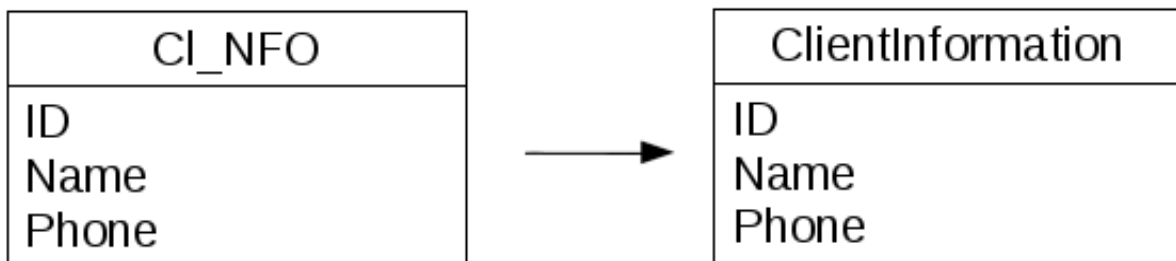


Рисунок 2.8 — переименование таблицы

- Удаление таблиц.

Операция применяется, если таблица больше не требуется и (или) не используется. Такая ситуация возникает после того, как таблица заменяется другим аналогичным источником данных, например другой таблицей или представлением, или же после того, как этот конкретный источник данных становится больше ненужным.

## 2.2. Рефакторинг качества данных

- Применение стандартных кодовых обозначений

Эта операция предусматривает применение стандартного набора кодовых обозначений к отдельному столбцу для обеспечения соответствия хранящихся в нем данных значениям аналогичных столбцов, хранящихся в другом месте в базе данных. То есть, для унификации с целью соответствия единообразной обработке в программном коде или следованию корпоративным стандартам.

Cities					
ID	...	Country			
752	...	CAN			
755	...	USA			

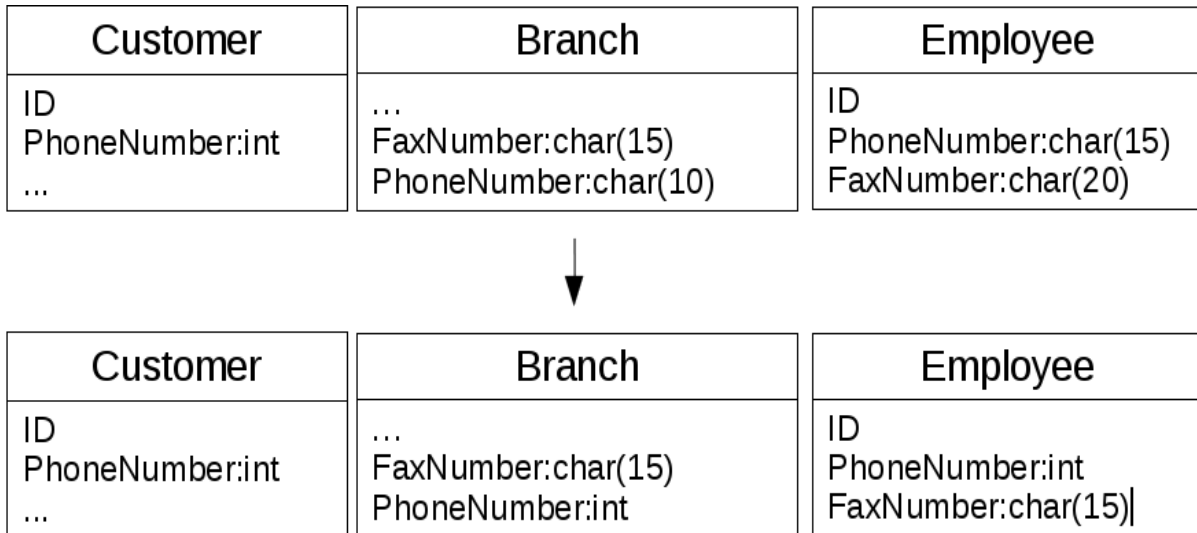
→

Cities					
ID	...	Country			
752	...	CA			
755	...	US			

Рисунок 2.9 — приведения данных к стандарту

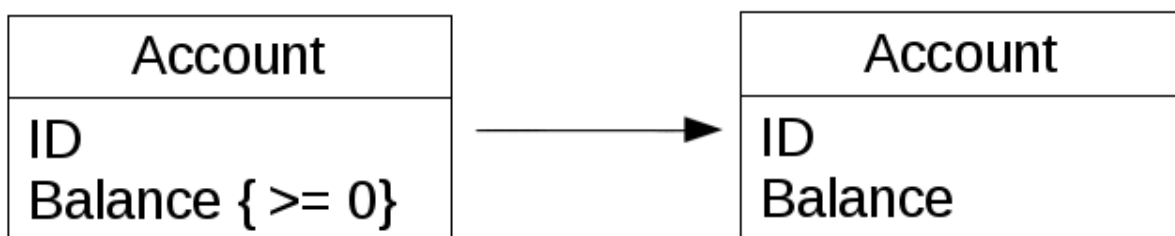
- Применение **стандартного типа**

Данная операция применяется, как правило, для упрощения программного кода обработки данных из базы. Также после применения операции данного типа не возникнет проблем с определением внешних ключей на уровне БД, так как типы уже приведены.



- Рисунок 2.10 — приведения типа данных к стандарту **Уничтожение ограничения столбца**

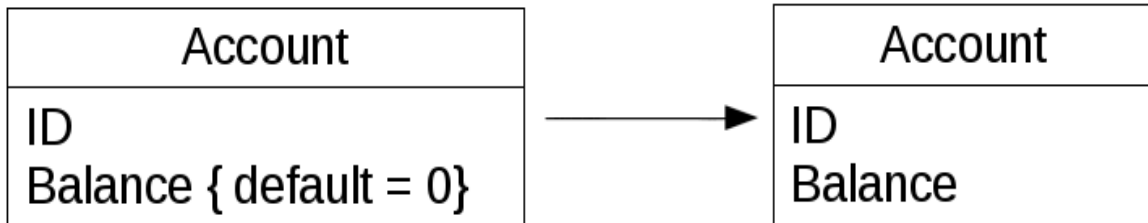
Наиболее распространенной причиной, по которой может потребоваться применить данную операцию, является то, что рассматриваемое ограничение становится больше не применимым из-за изменений в бизнес-правилах.



- Рисунок 2.11 — уничтожение ограничений для столбца «Balance»  
Введение ограничения столбца  
Операция, обратная, рассмотренной выше. Также применяется при изменении бизнес-логики.

- Уничтожение значения, заданного по умолчанию

По тем же причинам, что «Уничтожение ограничений столбца» выполняется рассматриваемая операция.



- Рисунок 2.12 — уничтожение значения по умолчанию столбца «Balance» Введение значения, заданного по умолчанию

Данная операция является обратной операции, рассмотренной выше. Область применения — приведение таблицы к корпоративному стандарту или (и) изменившиеся в ходе работы программного продукта требования.

- Уничтожение столбца, не допускающего NULL-значений

Необходимость в применении операции “Уничтожение столбца, не допускающего NULL-значений” может быть обусловлена двумя основными причинами. Во первых, сущность может создаваться в одном приложении, но при этом значение одному из ее не будет присваиваться сразу же, а в другом приложении соответствующая строка обновляется в дальнейшем. Во вторых, может возникнуть необходимость, чтобы какой-то конкретный столбец мог принимать NULL-значения. Например, если в одном из приложений невозможно предоставить значение для рассматриваемого столбца в связи с тем, что это приложение проходит через какую операцию рефакторинга, то может потребоваться отменить ограничение, касающееся запрета применения NULL значений.

- Замена кодового обозначения типа флажками свойств

Эта операция предназначена для замены столбца с кодовыми обозначениями отдельными флажками свойств, обычно реализуемыми как булевы столбцы, в той же таблице. В результате упрощается выборка и отпадает необходимость в дополнительной обработке данных в программном коде.

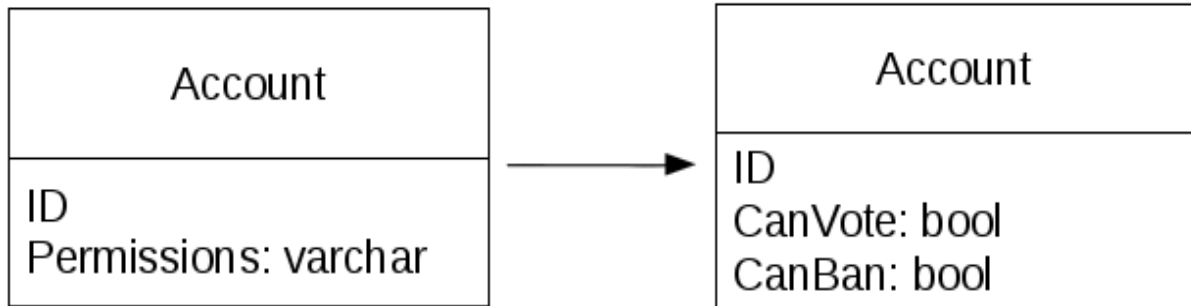


Рисунок 2.13 — замена сложного столбца на столбцы-флажки

### 3. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

3.1. Изучить теоретические сведения, относящиеся к рефакторингу баз данных.

3.2. Выполнить анализ предоставленных по варианту таблиц БД.

3.3. Выполнить рефакторинг заданных по варианту таблиц БД.

3.4. Оформить отчёт, содержащий подробное описание каждого модифицированного фрагмента базы данных и описание использованного метода рефакторинга.

### 4. СОДЕРЖАНИЕ ОТЧЕТА

4.1. Цель работы.

4.2. Постановка задачи.

4.3. Анализ первоначального варианта программного кода.

4.4. Результаты рефакторинга.

4.5. Выводы по работе.

## **5. КОНТОЛЬНЫЕ ВОПРОСЫ**

5.1. Назовите причины необходимости рефакторинга базы данных.

5.2. Какие основные предпосылки для проведения рефакторнига структуры и качества базы данных?

5.3. Назовите основные методы рефакторинга в рамках одной таблицы.

5.4. Назовите основные методы рефакторинга на уровне нескольких таблиц.

5.5. Какие существуют методы рефакторинга качества данных?

## ПРИЛОЖЕНИЕ А — Варианты задний

Таблица 1.

№ П/П	Метод рефакторинга
1	Разбиение столбца или Слияние столбца
2	Перемещение столбца
3	Удаление столбца
4	Переименование столбца
5	Разбиение таблицы
6	Удаление таблицы
7	Переименование таблицы
8	Применение стандартного типа
9	Уничтожение или введение ограничений столбца
10	Уничтожение или введение значения, заданного по умолчанию
11	Уничтожение столбца, не допускающего NULL-значений
12	Замена кодового обозначения типа флажками свойств

Таблица 2.

№ Варианта	Таблица	Методы рефакторинга
1.	private	1, 4, 5, 6, 9, 11
2.	name	2, 3, 5, 7, 10, 12
3.	private	1, 3, 5, 6, 8, 9
4.	name	2, 4, 5, 7, 9, 10
5.	private	1, 2, 5, 6, 10, 11
6.	name	1, 4, 5, 7, 8, 12
7.	private	1, 4, 5, 6, 9, 11
8.	name	1, 2, 5, 7, 11, 12
9.	private	2, 4, 5, 6, 10, 11
10.	name	1, 4, 5, 7, 8, 12

## ПРИЛОЖЕНИЕ Б — SQL-команды создания таблиц, заданных по варианту

```
--
-- Структура таблицы `name`
--
CREATE TABLE IF NOT EXISTS `name` (
  `TimeStmp` timestamp NULL DEFAULT NULL ON UPDATE CURRENT_TIMESTAMP,
  `Number` int(10) NOT NULL,
  `Surname` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `Name` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `WifeMobile` varchar(30) DEFAULT NULL,
  `Email` varchar(30) DEFAULT NULL,
  `FatherName` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `smnpassno` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `SMNPASSPO` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `CVLPASSNO` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `CvliISSD` date DEFAULT NULL,
  `CvliVLD` date DEFAULT NULL,
  `CVLPASSLET` varchar(10) CHARACTER SET utf8 DEFAULT NULL,
  `CVLPASSPO` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `TravelPassSerial` varchar(30) DEFAULT NULL,
  `TravelPassNo` varchar(30) DEFAULT NULL,
  `TravelPassIssue` date DEFAULT NULL,
  `TravelPassVLD` date DEFAULT NULL,
  `Readyness` date DEFAULT NULL,
  `readyprem` varchar(30) CHARACTER SET utf8 DEFAULT NULL,
  `Remarks` varchar(1500) CHARACTER SET utf8 DEFAULT NULL,
  `evaluation` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `additionalAI` text CHARACTER SET utf8,
  `English` int(10) DEFAULT NULL,
  `Licence` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `PhoneSeamen` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `Photo` mediumblob,
  `NATIONALIT` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `POB` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `HAIRS` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `EYES` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `HEIGHT` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `WEIGHT` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `flat` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `APPART` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `STREET` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `CITY` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `COUNTRY` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `RELATION` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `WIFE` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `YearWife` int(10) DEFAULT NULL,
  `Father` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `YearFather` int(10) DEFAULT NULL,
  `Mother` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `YearMother` int(10) DEFAULT NULL,
  `Maiden` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `CHLD1` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `YearChld1` date DEFAULT NULL,
  `CHLD2` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `YearChld2` date DEFAULT NULL,
```

```

`CHLD3` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
`YearChld3` date DEFAULT NULL,
`PrivateREMARKS` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
`Shoes` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
`Color` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
`Overall` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
`StreetR` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
`CityR` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
`CountryR` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
`EDUCATION` varchar(255) DEFAULT NULL,
`education2` varchar(255) DEFAULT NULL,
`dat_educ2` varchar(50) DEFAULT NULL,
`CERTNO` varchar(255) DEFAULT NULL,
`GRADE` varchar(255) DEFAULT NULL,
`FIRSTAIDNO` varchar(255) DEFAULT NULL,
`USA VISA VLD` varchar(50) DEFAULT NULL,
`password` varchar(50) DEFAULT NULL,
`login` varchar(50) DEFAULT NULL
) ENGINE=InnoDB AUTO_INCREMENT=33350 DEFAULT CHARSET=utf8

```

```

--
-- Структура таблицы `private`
--

```

```

CREATE TABLE IF NOT EXISTS `private` (
  `number` int(10) DEFAULT NULL,
  `Phone` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `Skype` varchar(50) DEFAULT NULL,
  `Mobile` varchar(30) DEFAULT NULL,
  `MobileViberStatus` tinyint(1) DEFAULT NULL,
  `MobileWhatsAppStatus` tinyint(1) DEFAULT NULL,
  `MobileTelegramStatus` tinyint(1) DEFAULT NULL,
  `AltMobile` varchar(30) DEFAULT NULL,
  `NATIONALIT` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `POB` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `HAIRS` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `EYES` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `HEIGHT` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `WEIGHT` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `flat` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `APPART` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `STREET` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `CITY` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `COUNTRY` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `NEXTKIN` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `RELATION` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `WIFE` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `year_wife` int(10) DEFAULT NULL,
  `FATHER` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `year_father` int(10) DEFAULT NULL,
  `MOTHER` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `year_mother` int(10) DEFAULT NULL,
  `maiden` varchar(50) CHARACTER SET utf8 DEFAULT NULL,
  `CHLD1` varchar(255) CHARACTER SET utf8 DEFAULT NULL,
  `year_child1` datetime DEFAULT NULL,
  `CHLD2` varchar(255) CHARACTER SET utf8 DEFAULT NULL,

```

```
`year_child2` datetime DEFAULT NULL,  
`CHLD3` varchar(255) CHARACTER SET utf8 DEFAULT NULL,  
`year_child3` datetime DEFAULT NULL,  
`REMARKS` varchar(255) CHARACTER SET utf8 DEFAULT NULL,  
`KOLD` int(10) DEFAULT NULL,  
`Shoes` varchar(50) CHARACTER SET utf8 DEFAULT NULL,  
`Color` varchar(50) CHARACTER SET utf8 DEFAULT NULL,  
`Overall` varchar(50) CHARACTER SET utf8 DEFAULT NULL,  
`Улица` varchar(50) CHARACTER SET utf8 DEFAULT NULL,  
`Город` varchar(50) CHARACTER SET utf8 DEFAULT NULL,  
`Страна` varchar(50) CHARACTER SET utf8 DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8
```