**3 МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ**

**КОНТРОЛЬНОЙ РАБОТЫ**

Контрольная работа заключается в проверке знаний по темам «Нор-

мализация отношений и язык SQL».

**3.1 Варианты контрольной работы**

**Вариант 2**

***Задание 1***

Заполните значения атрибутов отношения «Счета клиентов», учиты-

вая, что один клиент может иметь несколько счетов как в одном, так

и в нескольких банках, при этом номера счетов в разных банках могут сов-

падать. Выявите первичный ключ и все возможные зависимости, нормали-

зуйте отношение по 3НФ.

7

Счета клиентов (Код клиента; ФИО клиента; Код банка; Наименова-

ние банка; № счета; ФИО управляющего банком).

***Задание 2***

Реализуйте SQL-запросы на создание таблиц, созданных в ходе нор-

мализации отношения «Счета клиентов».

***Задание 3***

Реализуйте SQL-запросы на добавление записей в созданные таблицы.

**3.2 Порядок выполнения контрольной работы**

**Выполнение задания 1**

Нормализация предложенного отношения должна быть проведена

с учетом правил нормализации отношений, изложенных в разделе 4.1 учеб-

ного пособия [1].

В отношениях должны быть определены первичные ключи и внеш-

ние ключи (для результирующих отношений), все возможные зависимости

между атрибутами отношений. ***Суррогатные первичные ключи допуска-***

***ется создавать в нормализованных отношениях только в том случае,***

***если отсутствует возможность явного определения первичного ключа.***

При нормализации отношений необходимо руководствоваться сле-

дующими основными определениями.

Отношение R находится во второй нормальной форме (2НФ) тогда

и только тогда, когда отношение находится в первой нормальной форме

и каждый его неключевой атрибут полностью зависит от первичного клю-

ча. Или, что тоже справедливо, отношение, находящееся во второй нор-

мальной форме, не содержит атрибутов, зависящих от части ключа.

15

Отношение R находится в третьей нормальной форме (3НФ) в том

и только в том случае, если находится в 2НФ и каждый неключевой атри-

бут нетранзитивно зависит от первичного ключа, т. е. среди атрибутов от-

ношения нет атрибутов, транзитивно зависящих от ключа (среди его не-

ключевых атрибутов нет зависящих от другого неключевого атрибута).

Нормализации подвергаются отношения, каждое из которых содер-

жит характеристики объектов предметной области, при этом на каждом сле-

дующем шаге проектирования (нормализации) с помощью декомпозиции от-

ношений достигается такой набор схем отношений, что каждая следующая

нормальная форма обладает лучшими свойствами, чем предыдущая.

Для выявления аномалий и коллизий необходимо заполнить значения

атрибутов исходного и результирующих отношений. Обратите внимание, что

каждому нормализованному отношению должно быть присвоено имя.

**Выполнение заданий 2 и 3**

***Формирование запросов на создание таблиц***

Перед созданием SQL-запросов рекомендуется повторить раздел 5

учебного пособия.

При создании запросов необходимо изучить синтаксис команды

CREATE TABLE, с помощью которой создается новая таблица и которая

используется для описания полей и индексов в таблице.

Для каждого поля необходимо определить размер и тип данных.

В следующем примере представлено создание двумя запросами но-

вых таблиц «Студент» и «Задолженность\_за\_обучение» с внешним клю-

чом «Код\_студента», связанным с полем «Код\_студента», в таблице «Сту-

дент».

CREATE TABLE Студент

(Код\_студента AUTOINCREMENT PRIMARY KEY,

Номер\_зачетной\_книжки INTEGER, ФИО\_студента TEXT(50),

16

Место\_рождения TEXT (50));

CREATE TABLE Задолженность\_за\_обучение

(Код\_задолженности AUTOINCREMENT PRIMARY KEY,

Код\_студента INTEGER, Сумма\_задолженности MONEY,

CONSTRAINT f1\_i FOREIGN KEY (Код\_студента)

REFERENCES Студент (Код\_студента)

***Формирование запросов на добавление данных***

Добавление записей в таблицы необходимо осуществить с помощью

команды INSERT INTO. Для этого необходимо изучить синтаксис данной

команды.

В следующем примере с помощью запроса добавляется новая запись

в таблицу «Студент»:

INSERT INTO Студент (Номер\_зачетной\_книжки,

ФИО\_студента, Место\_рождения, Дата\_рождения )

VALUES (201454321, 'Иванов Иван Петрович', 'г. Томск',

'12.02.1996');

Все запросы необходимо выполнить в какой-либо СУБД (например,

в СУБД MS Access). Результат выполнения запросов следует представить

в виде скриншотов либо в виде mdb-файла (для СУБД MS\_\_Access).