

**Филиал федерального государственного бюджетного образовательного учреждения  
высшего образования  
«Национальный исследовательский университет «МЭИ»  
в г. Смоленске**

**МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ ДЛЯ ОБЕСПЕЧЕНИЯ  
ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА**

---

Направление подготовки: **09.03.03 «Прикладная информатика»**

Профиль: **«Прикладная информатика в топливно-энергетическом комплексе»**

Уровень высшего образования: **бакалавриат**


Нормативный срок обучения: **4 года**

Форма обучения: **очная**

Год набора: **2023**


Направление подготовки 09.03.03 «Прикладная информатика»  
Профиль «Прикладная информатика в топливно-энергетическом комплексе»  
Методическое обеспечение РПД Б1.В.ДВ.02.02 «Корпоративные информационные системы»

**Методические материалы составил:**

канд. экон. наук, доцент кафедры  
информационных технологий в экономике и управлении  О.В. Булыгина

«20» января 2023 г.

**Заведующий кафедрой информационных технологий в экономике и управлении:**

  
\_\_\_\_\_ д-р техн. наук, профессор М.И. Дли  
подпись \_\_\_\_\_ ФИО

«08» февраля 2023 г.

## МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ЛАБОРАТОРНЫХ РАБОТ ПО ДИСЦИПЛИНЕ

### ЛАБОРАТОРНАЯ РАБОТА 1 «Проектирование и создание базы данных в СУБД MySQL»

**Цель работы:** приобрести навыки построения логической и физической модели данных предметной области, а также их практической реализации в СУБД MySQL.

#### Задание 1.

Для заданной предметной области в соответствии со стандартом IDEF1X построить логическую и физическую модели данных.

#### Пример выполнения работы

В качестве примера рассмотрим базу данных *book* Интернет-магазина, торгующего компьютерной литературой, которая содержит:

- тематические каталоги, по которым сгруппированы книги;
- предлагаемые книги (название, автор, год издания, цена, имеющееся на складе количество);
- зарегистрированные покупатели (имя, отчество, фамилия, телефон, адрес электронной почты, статус – авторизованный, неавторизованный, заблокированный, активный с хорошей кредитной историей);
- покупки, совершенные в магазине (время совершения покупки, число приобретенных экземпляров книги).

Логическая модель данных предметной области в стандарте IDEF1X представлена на рис. 1. Выделены сущности *КАТАЛОГ*, *КНИГА*, *КЛИЕНТ*, *ЗАКАЗ*, между которыми установлены неидентифицирующие связи мощностью один-ко-многим, определенные спецификой предметной области.

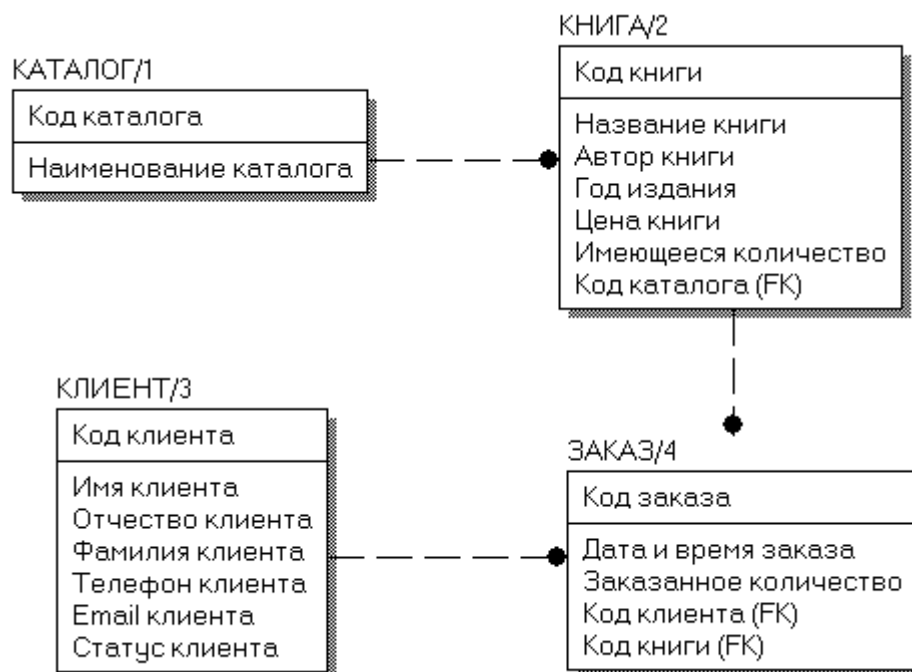


Рис. 1. Логическая модель данных предметной области

Физическая модель данных предметной области в стандарте IDEF1X для СУБД MySQL представлена на рис. 2.

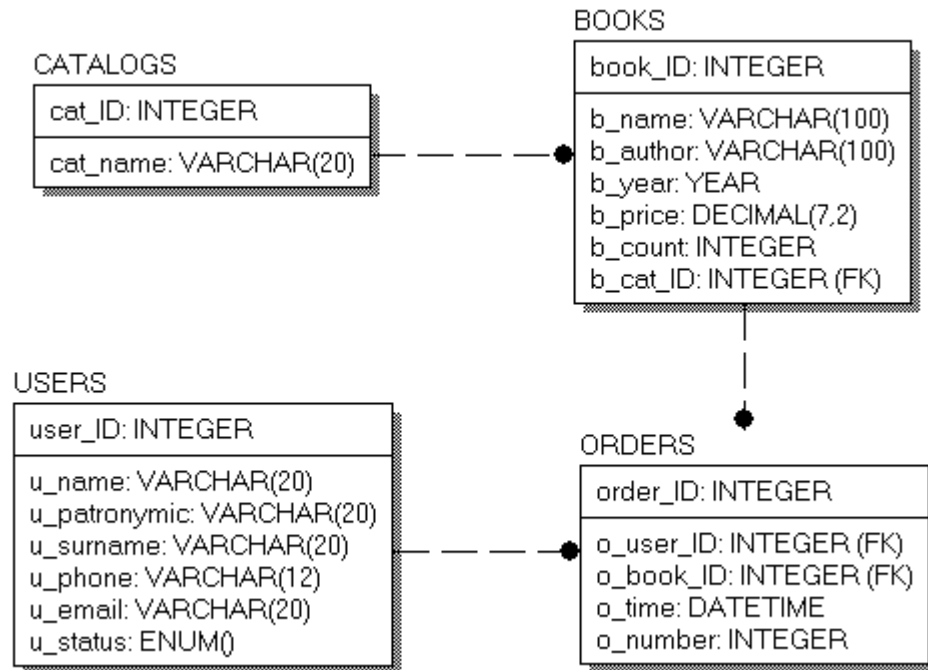


Рис. 2. Физическая модель предметной области

База данных *book* состоит из четырех таблиц:

- *catalogs* – список торговых каталогов;
- *books* – список предлагаемых книг;
- *users* – список зарегистрированных пользователей магазина;
- *orders* – список заказов (осуществленных сделок).

Таблица *catalogs* состоит из двух полей:

- *cat ID* – уникальный код каталога;
- *cat name* – имя каталога.

Оба поля должны быть снабжены атрибутом *NOT NULL*, поскольку неопределенное значение для них недопустимо.

Таблица *books* состоит из семи полей:

- *book ID* – уникальный код книги;
- *b name* – название книги;
- *b author* – автор книги;
- *b year* – год издания;
- *b price* – цена книги;
- *b count* – количество книг на складе;
- *b\_cat ID* – код каталога из таблицы *catalogs*.

Цена книги *b\_price* и количество экземпляров на складе *b\_count* могут иметь атрибут *NULL*. На момент доставки часто неизвестны количество товара и его цена, но отразить факт наличия товара в прайс-листе необходимо.

Поле *b\_cat ID* устанавливает связь между таблицами *catalogs* и *books*. Это поле должно быть объявлено как внешний ключ (FK) с правилом каскадного удаления и обновления. Обновление таблицы *catalogs* вызовет автоматическое обновление таблицы *books*.

Удаление каталога в таблице *catalogs* приведет к автоматическому удалению всех записей в таблице *books*, соответствующих каталогу.

Таблица *users* состоит из семи полей:

- *user ID* – уникальный код покупателя;
- *u\_name* – имя покупателя;
- *u\_patronymic* – отчество покупателя;
- *u\_surname* – фамилия покупателя;
- *u\_phone* – телефон покупателя (если имеется);
- *u\_email* – e-mail покупателя (если имеется);
- *u\_status* – статус покупателя.

Статус покупателя представлен полем типа *ENUM*, которое может принимать одно из четырех значений:

- *active* – авторизованный покупатель, который может осуществлять покупки через Интернет;
- *passive* – неавторизованный покупатель (значение по умолчанию), который осуществил процедуру регистрации, но не подтвердил ее и пока не может осуществлять покупки через Интернет, однако ему доступны каталоги для просмотра;
- *lock* – заблокированный покупатель, не может осуществлять покупки и просматривать каталоги магазина;
- *gold* – активный покупатель с хорошей кредитной историей, которому предоставляется скидка при следующих покупках в магазине.

Поля *u\_phone* и *u\_email* могут быть снабжены атрибутом *NULL*. Остальные поля должны получить атрибут *NOT NULL*.

Таблица *orders* включает пять полей:

- *order ID* – уникальный номер сделки;
- *o\_user ID* – номер пользователя из таблицы *users*;
- *o\_book ID* – номер товарной позиции из таблицы *books*;
- *o\_time* – время совершения сделки;
- *o number* – число приобретенных товаров.

Поля таблицы *orders* должны быть снабжены атрибутом *NOT NULL*, т. к. при совершении покупки вся информация должна быть занесена в таблицу.

В таблице *orders* устанавливается связь с таблицами *users* (за счет поля *o\_user ID*) и *books* (за счет поля *o\_book ID*). Эти поля объявлены как внешние ключи (FK) с правилом каскадного удаления и обновления. Обновление таблиц *users* и *books* приведет к автоматическому обновлению таблицы *orders*. Удаление любого пользователя в таблице *users* приведет к автоматическому удалению всех записей в таблице *orders*, соответствующих этому пользователю.

## Задание 2.

Средствами СУБД MySQL практически реализовать разработанную физическую модель данных для заданной предметной области.

### Пример выполнения работы

Операторы создания БД *book* имеют следующий вид:

```
DROP DATABASE IF EXISTS book;  
CREATE DATABASE book;  
USE book;  
CREATE TABLE catalogs (  
    cat_ID int(6) NOT NULL AUTO_INCREMENT,  
    cat_name varchar(20) NOT NULL,  
    PRIMARY KEY (cat_ID)
```

```
) TYPE=InnoDB;  
CREATE TABLE books (  
    book_ID int(6) NOT NULL AUTO_INCREMENT,  
    b_name varchar(100) NOT NULL,  
    b_author varchar(100) NOT NULL,  
    b_year year NOT NULL,  
    b_price decimal(7,2) NULL default '0.00',  
    b_count int(6) NULL default '0',  
    b_cat_ID int(6) NOT NULL default '0',  
    PRIMARY KEY (book_ID),  
    FOREIGN KEY (b_cat_ID) REFERENCES catalogs(cat_ID) ON DELETE  
CASCADE ON UPDATE CASCADE  
) TYPE=InnoDB;  
CREATE TABLE users (  
    user_ID int(6) NOT NULL AUTO_INCREMENT,  
    u_name varchar(20) NOT NULL,  
    u_patronymic varchar(20) NOT NULL,  
    u_surname varchar(20) NOT NULL,  
    u_phone varchar(12) NULL,  
    u_email varchar(20) NULL,  
    u_status ENUM ('active', 'passive', 'lock', 'gold') default 'passive',  
    PRIMARY KEY (user_ID)  
) TYPE=InnoDB;  
CREATE TABLE orders (  
    order_ID int(6) NOT NULL AUTO_INCREMENT,  
    o_user_ID int NOT NULL,  
    o_book_ID int NOT NULL,  
    o_time datetime NOT NULL default '0000-00-00 00:00:00',  
    o_number int(6) NOT NULL default '0',  
    PRIMARY KEY (order_ID),  
    FOREIGN KEY (o_book_ID) REFERENCES books(book_ID) ON DELETE  
CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY (o_user_ID) REFERENCES users(user_ID) ON DELETE  
CASCADE ON UPDATE CASCADE  
) TYPE=InnoDB;
```

### **Задание 3.**

Средствами MySQL заполнить согласованными данными таблицы БД.

#### **Пример выполнения работы**

Операторы заполнения БД *book* имеют следующий вид.

```
USE book;  
SET CHARACTER SET cp1251;  
DELETE FROM catalogs;  
INSERT INTO catalogs VALUES (1, 'Программирование');  
INSERT INTO catalogs VALUES (2, 'Интернет');  
INSERT INTO catalogs VALUES (3, 'Базы данных');  
INSERT INTO catalogs VALUES (4, 'Сети');  
INSERT INTO catalogs VALUES (5, 'Мультимедиа');  
DELETE FROM books;  
INSERT INTO books VALUES (1, 'JavaScript в кармане', 'Рева О.Н.', 2008, 42.00, 10, 1);
```

```
INSERT INTO books VALUES (2,'Visual FoxPro 9.0','Клепинин В.Б.', 2007, 660.00, 2, 1);
INSERT INTO books VALUES (3,'C++ Как он есть','Тимофеев В.В.',2009, 218.00, 4, 1);
INSERT INTO books VALUES (4,'Создание приложений с помощью С#','Фаронов В.В.', 2008,
169.00, 1, 1);
INSERT INTO books VALUES (5,'Delphi. Народные советы','Шкрыль А.А.',2007,243.00,6,1);
INSERT INTO books VALUES (6,'Delphi. Полное руководство','Сухарев М.',2008,500.00,6,1);
INSERT INTO books VALUES (7,'Профессиональное программирование на PHP', 'Шлос-
снейгл Дж.', 2006, 309.00, 5, 1);
INSERT INTO books VALUES (8,'Совершенный код','Макконнелл С.', 2007, 771.00, 1, 1);
INSERT INTO books VALUES (9,'Практика программирования','Керниган Б.', 2004, 214.00,
12, 1);
INSERT INTO books VALUES (10,'Принципы маршрутизации в Internet','Хелеби С.', 2001,
428.00, 4, 2);
INSERT INTO books VALUES (11,'Поиск в Internet','Гусев В.С.',2004,107.00,2,2);
INSERT INTO books VALUES (12,'Web-конструирование','Дуванов А.А.', 2003, 177.00, 6, 2);
INSERT INTO books VALUES (13,'Самоучитель Интернет','Константинов Ю.П.', 2009,
121.00, 4, 2);
INSERT INTO books VALUES (14,'Популярные интернет-браузеры','Маринин С.А.', 2007,
82.00, 6, 2);
INSERT INTO books VALUES (15,'Общение в Интернете','Экслер А.', 2006, 85.00, 5, 2);
INSERT INTO books VALUES (16,'Базы данных','Малыхина М.П.', 2006, 326.00, 2, 3);
INSERT INTO books VALUES (17,'Базы данных. Разработка приложений','Рудикова Л.В.',
2006, 189.00, 6, 3);
INSERT INTO books VALUES (18,'Раскрытие тайн SQL','Оппель Э.', 2007, 200.00, 3, 3);
INSERT INTO books VALUES (19,'Практикум по Access','Золотова С.И.', 2007, 87.00, 6, 3);
INSERT INTO books VALUES (20,'Компьютерные сети','Танненбаум Э.', 2007, 630.00, 6, 4);
INSERT INTO books VALUES (21,'Сети. Поиск неисправностей','Бигелу С.', 2005, 434.00, 4,
4);
INSERT INTO books VALUES (22,'Безопасность сетей','Брегг Р.', 2006, 462.00, 5, 4);
INSERT INTO books VALUES (23,'Анализ и диагностика компьютерных сетей', 'Хогдал
Дж.', 2001, 344.00, 3, 4);
INSERT INTO books VALUES (24,'Локальные вычислительные сети', 'Епанешников А.' ,
2005, 82.00, 8, 4);
INSERT INTO books VALUES (25,'Цифровая фотография','Надеждин Н.', 2004, 149.00,
20,5);
INSERT INTO books VALUES (26,'Музыкальный компьютер для гитариста', 'Петелин
Р.Ю.', 2004, 217.00, 15, 5);
INSERT INTO books VALUES (27,'Видео на ПК','Федорова А.',2003,231.00,10,5);
INSERT INTO books VALUES (28,'Мультипликация во Flash','Киркпатрик Г.', 2006, 211.00,
20, 5);
INSERT INTO books VALUES (29,'Запись CD и DVD','Гультияев А.К.', 2003, 167.00, 12, 5);
INSERT INTO books VALUES (30,'Запись и обработка звука на компьютере', 'Лоянич А.А.',
2008, 51.00, 8, 5);
DELETE FROM users;
INSERT INTO users VALUES (1,'Александр','Валерьевич','Иванов','58-98-78',
'ivanov@email.ru', 'active');
INSERT INTO users VALUES (2,'Сергей','Иванович','Лосев','90-57-77', 'losev@email.ru', 'pas-
sive');
INSERT INTO users VALUES (3,'Игорь','Николаевич','Симонов','95-66-61', 'simonov@email.ru',
'active');
```

```
INSERT INTO users VALUES (4,'Максим','Петрович','Кузнецов',NULL, 'kuznetsov@email.ru',
'active');
INSERT INTO users VALUES (5,'Анатолий','Юрьевич','Петров', NULL, NULL, 'lock');
INSERT INTO users VALUES (6,'Александр','Александрович','Корнеев','89-78-36',
'korneev@email.ru', 'gold');
DELETE FROM orders;
INSERT INTO orders VALUES (1,3,8,'2009-01-04 10:39:38',1);
INSERT INTO orders VALUES (2,6,10,'2009-02-10 09:40:29',2);
INSERT INTO orders VALUES (3,1,20,'2009-02-18 13:41:05',4);
INSERT INTO orders VALUES (4,4,20,'2009-03-10 18:20:00',1);
INSERT INTO orders VALUES (5,3,20,'2009-03-17 19:15:36',1);
```

**Отчет по итогам выполнения лабораторной работы** должен содержать описание предметной области, логическую и физическую модель данных, подробное описание таблиц, а также код создания базы данных.

## ЛАБОРАТОРНАЯ РАБОТА 2 «Создание сложных запросов на выборку»

**Цель работы:** приобрести навыки создания простых и сложных запросов на выборку данных в СУБД MySQL.

### Задание 1.

Для заданной предметной области построить два простых запроса на выборку с использованием операторов и функций MySQL.

#### Пример выполнения работы

1. Создадим простой запрос на выборку к таблице *books*, который выводит максимальную и минимальную цены товарных позиций, присваивая им соответственно псевдонимы *maximum* и *minimum*:

```
mysql> SELECT MAX(b_price) AS maximum, MIN(b_price) AS minimum
-> FROM books;
+-----+-----+
| maximum | minimum |
+-----+-----+
| 771.00  | 42.00   |
+-----+-----+
1 row in set (0.00 sec)
```

2. Создадим простой запрос на выборку к таблице *books*, который выводит количество записей, соответствующих каждому из уникальных значений *b\_cat\_ID*. Для этого используем функцию *COUNT()* выражением *GROUP BY*:

```
mysql> SELECT b_cat_ID, COUNT(b_cat_ID) FROM books
-> GROUP BY b_cat_ID ORDER BY b_cat_ID;
+-----+-----+
| b_cat_ID | COUNT(b_cat_ID) |
+-----+-----+
| 1        | 9                |
| 2        | 6                |
| 3        | 4                |
| 4        | 5                |
| 5        | 6                |
+-----+-----+
5 rows in set (0.00 sec)
```

### Задание 2.

Для заданной предметной области построить многотабличный запрос на выборку с использованием объединения.

#### Пример выполнения работы

Создадим многотабличный запрос на выборку, который выводит фамилии, имена и отчества покупателей магазина, сделавших менее двух покупок:

```
mysql> SELECT users.u_surname,users.u_name,users.u_patronymic,
-> COUNT(orders.order_ID) AS total
-> FROM users LEFT JOIN orders ON users.user_ID=orders.o_user_ID
-> GROUP BY users.user_ID
-> HAVING total<2
-> ORDER BY total DESC;
```

u_surname	u_name	u_patronymic	total
Иванов	Александр	Валерьевич	1
Корнеев	Александр	Александрович	1
Кузнецов	Максим	Петрович	1
Петров	Анатолий	Юрьевич	0
Лосев	Сергей	Иванович	0

5 rows in set (0.02 sec)

### Задание 3.

Для заданной предметной области построить запрос на выборку, содержащий вложенный запрос.

#### Пример выполнения работы

Создадим запрос на выборку с вложенным запросом, выводящим перечень книг, которые не заказывались покупателями:

```
mysql> SELECT book_ID, b_name, b_price FROM books
-> WHERE NOT EXISTS (SELECT * FROM orders
-> WHERE orders.o_book_ID=books.book_ID);
```

book_ID	b_name	b_price
1	JavaScript в кармане	42.00
2	Visual FoxPro 9.0	660.00
3	C++ Как он есть	218.00
4	Создание приложений с помощью С#	169.00
5	Delphi. Народные советы	243.00
6	Delphi. Полное руководство	500.00
7	Профессиональное программирование на PHP	309.00
9	Практика программирования	214.00
11	Поиск в Internet	107.00
12	Web-конструирование	177.00
13	Самочитель Интернет	121.00
14	Популярные интернет-браузеры	82.00
15	Общение в Интернете	85.00
16	Базы данных	326.00
17	Базы данных. Разработка приложений	189.00
18	Раскрытие тайн SQL	200.00
19	Практикум по Access	87.00
21	Сети. Поиск неисправностей	434.00
22	Безопасность сетей	462.00
23	Анализ и диагностика компьютерных сетей	344.00
24	Локальные вычислительные сети	82.00
25	Цифровая фотография	149.00
26	Музыкальный компьютер для гитариста	217.00
27	Видео на ПК	231.00
28	Мультипликация во Flash	211.00
29	Запись CD и DVD	167.00
30	Запись и обработка звука на компьютере	51.00

27 rows in set (0.03 sec)

Отчет по итогам выполнения лабораторной работы должен содержать код запросов и результаты их выполнения.

## ЛАБОРАТОРНАЯ РАБОТА 3 «Хранимые процедуры и триггеры»

**Цель работы:** приобрести навыки создания хранимых процедур и триггеров.

### Задание 1.

Для заданной предметной области построить написать две хранимые процедуры и включить их в БД.

#### Пример выполнения работы

1. Создадим хранимую процедуру, которая выводит число заказов покупателя по вводимому в качестве параметра процедуры коду покупателя.

```
mysql> CREATE PROCEDURE num<OUT total INT, IN user_kod INT>
-> BEGIN
-> SELECT COUNT(*) INTO total FROM orders WHERE o_user_ID=user_kod;
-> END
-> //
Query OK, 0 rows affected (0.00 sec)
```

Параметр *total* является выходным, его значение равно числу заказов покупателя, код которого записывается во входной параметр *user\_kod*. Процедура считает все строки, где код клиента совпадает с параметром *user\_kod*.

До вызова процедуры присваиваем параметру процедуры значение кода клиента. Затем вызываем процедуру оператором *CALL*. Для вывода результата можно воспользоваться оператором *SELECT*.

```
mysql> SET @user_kod=3//
Query OK, 0 rows affected (0.00 sec)

mysql> CALL num(@total,@user_kod)//
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT @total,@user_kod//
+-----+-----+
| @total | @user_kod |
+-----+-----+
| 2      | 3         |
+-----+-----+
1 row in set (0.00 sec)
```

2. Создадим хранимую процедуру, которая записывает в новую таблицу *fevral* все заказы, сделанные в феврале 2009 г. Предварительно необходимо создать новую пустую таблицу *fevral* со структурой, аналогичной структуре таблицы *orders*.

```
mysql> CREATE TABLE fevral<
-> f_order_ID int(6) NOT NULL,
-> f_o_user_ID int NOT NULL,
-> f_o_book_ID int NOT NULL,
-> f_o_time datetime NOT NULL,
-> f_o_number int(6) NOT NULL,
-> PRIMARY KEY (f_order_ID)
-> >TYPE=InnoDB//
Query OK, 0 rows affected, 1 warning (0.08 sec)
```

Хранимая процедура *ord\_fevr ( )* использует курсор *curf*, который в цикле читает данные из таблицы *orders* и добавляет их в таблицу *fevral*.

### Задание 2.

Для заданной предметной области построить написать два триггера для разных таблиц базы данных.

### Пример выполнения работы

1. Создадим триггер, который при оформлении нового заказа (при добавлении новой записи в таблицу *orders*) будет увеличивать на 1 значение пользовательской переменной *@tot*.

```
mysql> delimiter //
mysql> CREATE TRIGGER sub_count AFTER INSERT ON orders
-> FOR EACH ROW
-> BEGIN
-> SET @tot =@tot+1;
-> END//
Query OK, 0 rows affected (0.02 sec)

mysql> SELECT @tot //
+-----+
| @tot |
+-----+
| NULL |
+-----+
1 row in set (0.00 sec)
```

Для корректной работы триггера необходимо, чтобы пользовательская переменная *@tot* имела значение, отличное от *NULL*, т. к. операция сложения с *NULL* также приводит к *NULL*.

```
mysql> SET @tot=5//
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO orders VALUES (NULL,1.5,NOW(),10)//
Query OK, 1 row affected (0.00 sec)

mysql> SELECT @tot//
+-----+
| @tot |
+-----+
| 6    |
+-----+
1 row in set (0.00 sec)
```

2. Создадим триггер, который при добавлении новых покупателей преобразует имена и отчества покупателей в инициалы.

```
mysql> CREATE TRIGGER restrict_user BEFORE INSERT ON users
-> FOR EACH ROW
-> BEGIN
-> SET NEW.u_name = LEFT(NEW.u_name,1);
-> SET NEW.u_patronymic = LEFT(NEW.u_patronymic,1);
-> END//
Query OK, 0 rows affected (0.06 sec)

mysql> INSERT INTO users VALUES (NULL, 'Светлана', 'Петровна', 'Титова',
-> '83-89-00', NULL, 'active')//
Query OK, 1 row affected (0.03 sec)

mysql> SELECT u_surname, u_name, u_patronymic FROM users
-> WHERE userID = LAST_INSERT_ID()//
+-----+-----+-----+
| u_surname | u_name | u_patronymic |
+-----+-----+-----+
| Титова   | С      | П            |
+-----+-----+-----+
1 row in set (0.00 sec)

mysql> CREATE PROCEDURE ord_fevr()
-> BEGIN
-> DECLARE id int;
-> DECLARE _end int DEFAULT 0;
-> DECLARE userID int;
-> DECLARE bookID int;
-> DECLARE tim datetime;
-> DECLARE num int;
-> DECLARE curf CURSOR FOR SELECT * FROM orders
-> WHERE o_time BETWEEN '2009.02.01' AND '2009.02.28';
-> DECLARE CONTINUE HANDLER FOR NOT FOUND SET _end=1;
-> OPEN curf;
-> wet: LOOP
-> FETCH curf INTO id,userID,bookID,tim,num;
-> IF _end THEN LEAVE wet;
-> END IF;
-> INSERT INTO fevral VALUES(id,userID,bookID,tim,num);
-> END LOOP wet;
-> CLOSE curf;
-> END
-> //
Query OK, 0 rows affected (0.03 sec)
```

Вызов процедуры осуществляется оператором *CALL*. Для просмотра результата выполнения процедуры используем полную выборку из таблицы *fevral*.

```
mysql> CALL ord_fevr//
Query OK, 1 row affected (0.05 sec)

mysql> SELECT * FROM fevral//
+-----+-----+-----+-----+-----+
| f_order_ID | f_o_user_ID | f_o_book_ID | f_o_time           | f_o_number |
+-----+-----+-----+-----+-----+
| 2          | 6          | 10         | 2009-02-10 09:40:29 | 2         |
| 3          | 1          | 20         | 2009-02-18 13:41:05 | 4         |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

Отчет по итогам выполнения лабораторной работы должен содержать код хранимых процедур и триггеров, а также результаты их выполнения.

## ЛАБОРАТОРНАЯ РАБОТА 4 «Транзакции и представления»

**Цель работы:** приобрести навыки создания транзакций и представлений.

### Задание 1.

Для заданной предметной области создать транзакцию, произвести ее откат и фиксацию.

#### Пример выполнения работы

Для выполнения задания объединим несколько операций по добавлению в таблицу *catalogs* новых каталогов, а затем произведем откат транзакции, т. е. отмену произведенных действий. Отключаем режим автоматического завершения, добавляем новые записи и проверяем, добавились записи или нет.

```
mysql> SET AUTOCOMMIT=0;
Query OK, 0 rows affected (0.00 sec)

mysql> INSERT INTO catalogs VALUES(NULL,'Аппаратура');
Query OK, 1 row affected (0.06 sec)

mysql> INSERT INTO catalogs VALUES(NULL,'Безопасность');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM catalogs;
+-----+-----+
| cat_ID | cat_name |
+-----+-----+
|      1 | Программирование |
|      2 | Интернет |
|      3 | Базы данных |
|      4 | Сети |
|      5 | Мультимедиа |
|      6 | Аппаратура |
|      7 | Безопасность |
+-----+-----+
7 rows in set (0.02 sec)
```

Откатываем транзакцию оператором *ROLLBACK* (изменения не сохранились).

```
mysql> ROLLBACK;
Query OK, 0 rows affected (0.03 sec)

mysql> SELECT * FROM catalogs;
+-----+-----+
| cat_ID | cat_name |
+-----+-----+
|      1 | Программирование |
|      2 | Интернет |
|      3 | Базы данных |
|      4 | Сети |
|      5 | Мультимедиа |
+-----+-----+
5 rows in set (0.00 sec)
```

Воспроизведем транзакцию и сохраним действия оператором *COMMIT*.

```
mysql> INSERT INTO catalogs VALUES(NULL,'Аппаратура');
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO catalogs VALUES(NULL,'Безопасность');
Query OK, 1 row affected (0.00 sec)

mysql> SELECT * FROM catalogs;
+-----+-----+
| cat_ID | cat_name |
+-----+-----+
|      1 | Программирование |
|      2 | Интернет |
|      3 | Базы данных |
|      4 | Сети |
|      5 | Мультимедиа |
|      8 | Аппаратура |
|      9 | Безопасность |
+-----+-----+
7 rows in set (0.00 sec)
```

```
mysql> COMMIT;
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT * FROM catalogs;
+-----+-----+
| cat_ID | cat_name |
+-----+-----+
|      1 | Программирование |
|      2 | Интернет |
|      3 | Базы данных |
|      4 | Сети |
|      5 | Мультимедиа |
|      8 | Аппаратура |
|      9 | Безопасность |
+-----+-----+
7 rows in set (0.00 sec)
```

## Задание 2.

Для заданной предметной области создать два представления в БД и сформировать запрос к одному из представлений.

### Пример выполнения работы

1. Создадим вертикальное представление *list user*, которое будет отображать фамилию и инициалы покупателей, скрывая другие поля.

```
mysql> CREATE OR REPLACE VIEW list_user
-> AS SELECT CONCAT(u_surname, " ",
-> SUBSTRING(u_name,1,1), "- ",
-> SUBSTRING(u_patronymic,1,1), ".") AS name
-> FROM users ORDER BY name;
Query OK, 0 rows affected (0.08 sec)

mysql> SELECT * FROM list_user;
+-----+
| name |
+-----+
| Кузнецов М.П. |
| Корнеев А.А. |
| Петров А.Ю. |
| Иванов А.В. |
| Лосев С.И. |
| Симонов И.Н. |
+-----+
6 rows in set (0.02 sec)
```

2. Создадим представление *price* с общей стоимостью книг в каждом каталоге.

```
mysql> CREATE VIEW price
-> AS SELECT b_catID, SUM(b_price*b_count) AS price
-> FROM books
-> GROUP BY b_catID
-> ORDER BY price;
Query OK, 0 rows affected (0.08 sec)

mysql> SELECT * FROM price;
+-----+-----+
| b_catID | price |
+-----+-----+
|      3 | 2908.00 |
|      2 | 4389.00 |
|      4 | 9514.00 |
|      1 | 12123.00 |
|      5 | 15177.00 |
+-----+-----+
5 rows in set (0.01 sec)
```

Сформируем запрос к таблице *catalogs* и представлению *price*.

```
mysql> SELECT catalogs.cat_name, price.price
-> FROM price, catalogs
-> WHERE price.b_catID = catalogs.catID
-> ORDER BY catalogs.catID;
+-----+-----+
| cat_name | price |
+-----+-----+
| Программирование | 12123.00 |
| Интернет | 4389.00 |
| Базы данных | 2908.00 |
| Сети | 9514.00 |
| Мультимедиа | 15177.00 |
+-----+-----+
5 rows in set (0.02 sec)
```

Сформируем запрос к представлению *price* (получение минимального и максимального значений стоимости книг в каталогах и общей стоимости всех книг).

Направление подготовки 09.03.03 «Прикладная информатика»

Профиль «Прикладная информатика в топливно-энергетическом комплексе»

Методическое обеспечение РПД Б1.В.ДВ.02.02 «Корпоративные информационные системы»

```
mysql> SELECT MIN(price), MAX(price), SUM(price) FROM price;
+-----+-----+-----+
| MIN(price) | MAX(price) | SUM(price) |
+-----+-----+-----+
|    2908.00 |    15177.00 |    44111.00 |
+-----+-----+-----+
1 row in set (0.02 sec)
```

**Отчет по итогам выполнения лабораторной работы** должен содержать код транзакций и представлений, а также результаты их выполнения.