

Самостоятельная работа 4

Инвариантная часть

Задание 4.1: Проектирование с учетом ЗНФ

1. Требуется разработать реляционную БД, таблицы которой соответствующую ЗНФ. (Группированную таблицу привести к набору отношений, соответствующих всем ограничениям ЗНФ)

- Customer

CustomerNum(PK)	CustomerName	CustomerPhone	CustomerAddress
1	Иванов И.И	(812)-111-11-11	СПб, ул. Политехническая, 29
2	Петров П.П	(812)-222-22-22	СПб, ул. Политехническая, 21

- Product

CatalogNum(PK)	Product	Price
101	Рюкзак городской. Модель 1.	2000,00
102	Рюкзак городской. Модель 2.	4000,00
201	Накидка на рюкзак. Модель 2.	400,00

- Order

OrderNum(PK)	CatalogNum(FK)	Quantity	CustomerNum(FK)
1000	101	1	1
1000	201	1	1
1001	101	2	2
1002	102	1	2

1. Укажите первичные ключи Первичные ключи указаны в таблицах (PK)

2. Определите, какие атрибуты являются обязательными (NOT NULL)

- Customer - **NOT NULL**
 - {CustomerNum(PK), CustomerName}
- Product - **NOT NULL**
 - {CatalogNum, Product, Price}
- Order - **NOT NULL**
 - {OrderNum(PK), CatalogNum, Quantity, CustomerNum}

3. Определите связи между отношениями.

- Customer ~ Order – Один ко многим
- Product ~ Order - Один ко многим

4. Установите внешние ключи

- Ключи установлены в таблице

Написать простые запросы на языке SQL, поддержание нормализации отношений.

1. Добавьте в таблицу Book несколько книг одного автора.

```
insert into book(Author, Title, Publisher, BookYear) VALUES('Ganz',  
                    'Berserk', 'AAA', 2000), ('Ganz', 'Naruto', 'BBB', 2010)
```

```
mysql> select * from book;  
Empty set (0.02 sec)  
  
mysql> insert into book(Author, Title, Publisher, BookYear) VALUES('Ganz', 'Berserk', 'AAA', 2000), ('Ganz', 'Naruto', 'BBB', 2010)  
-> ;  
Query OK, 2 rows affected (0.01 sec)  
Records: 2 Duplicates: 0 Warnings: 0  
  
mysql> select * from book;  
+----+-----+-----+-----+-----+  
| BookID | Author | Title  | Publisher | BookYear |  
+----+-----+-----+-----+-----+  
| 7      | Ganz   | Berserk | AAA       | 2000     |  
| 8      | Ganz   | Naruto  | BBB       | 2010     |  
+----+-----+-----+-----+-----+  
2 rows in set (0.00 sec)  
  
mysql>
```

4399e27681eed73f6b4bfc5814b66bd3.png

2. В конструкторе создайте запрос, выводящий список авторов книг из таблицы Book (нам нужны только авторы, без названий книг и т.д.). Выполните запрос и обратите внимание на то, что если автор написал несколько книжек, то (В результатах запроса он будет фигурировать тоже несколько раз.

```
select author from book;
```

```
mysql> select author from book;  
+-----+  
| author |  
+-----+  
| Ganz   |  
| Ganz   |  
+-----+  
2 rows in set (0.00 sec)
```

7cff42fb6b8eb2c9cf4bdeac3fed396a.png

3. Создайте запрос, который выводит список авторов без повторений в алфавитном порядке.

```
select DISTINCT author from book;
```

```
mysql> select DISTINCT author from book;  
+-----+  
| author |  
+-----+  
| Ganz   |  
+-----+  
1 row in set (0.02 sec)
```

b29d14d4fbabb4c9de9b582c343820d9.png

4. Напишите запрос, выводящий информацию о книгах из таблицы BoоK (3 порядке от самых старых к самым новым (все поля таблицы BoоK должны быть отображены). Книги с одинаковым годом издания упорядочивайте (3 списке по названию в алфавитном порядке).

```
select * from book ORDER BY bookyear, title;
```

```
mysql> select * from book ORDER BY bookyear, title;
+-----+-----+-----+-----+-----+
| BookID | Author | Title   | Publisher | BookYear |
+-----+-----+-----+-----+-----+
|      7 | Ganz   | Berserk | AAA       | 2000     |
|      8 | Ganz   | Naruto  | BBB       | 2010     |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

e505bcff0ddb59afe1e9abade9390496.png