

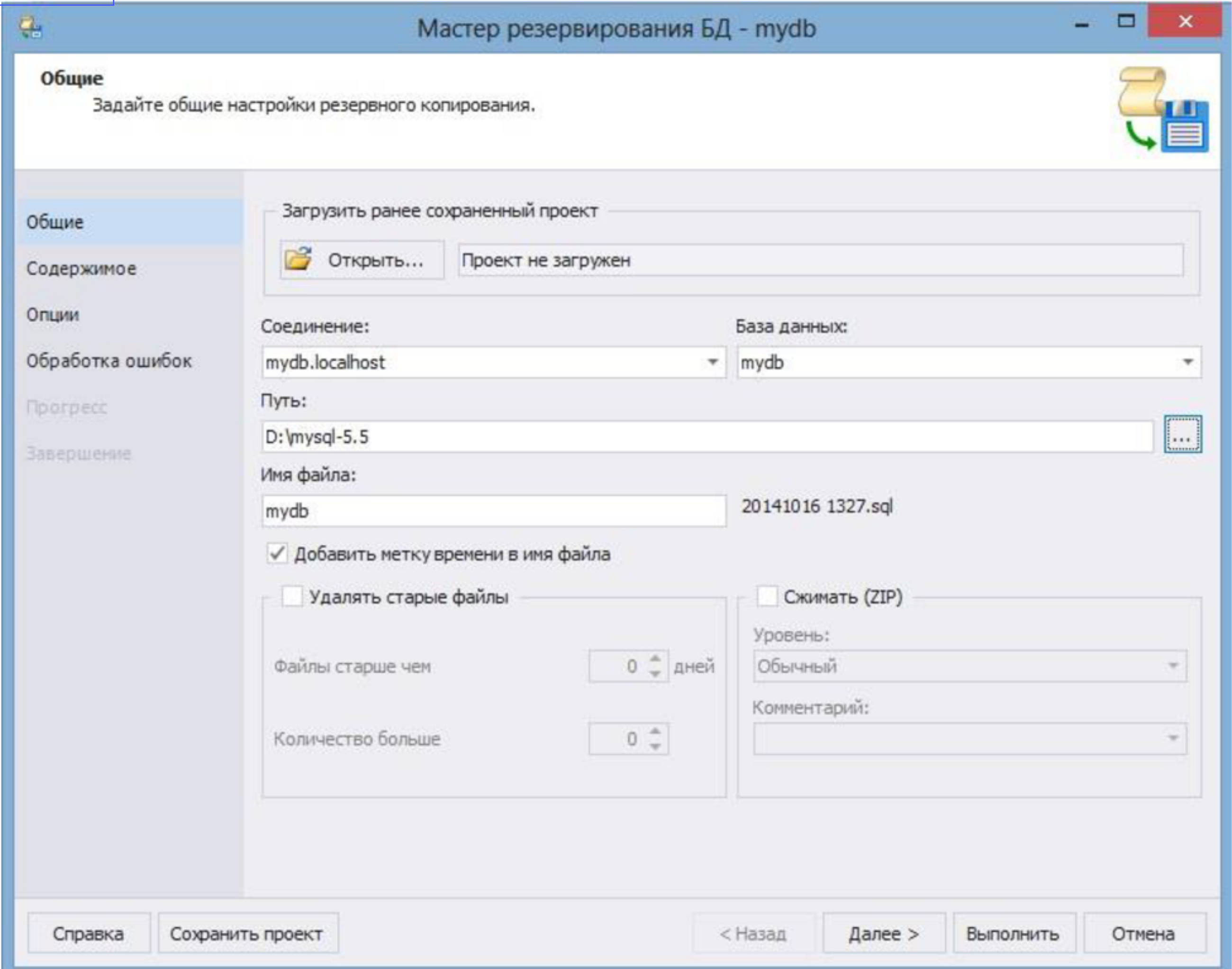


Рисунок 16.3 – Этап 4

5. Создаем пользователя для репликации «RepUser» с паролем «reppass»(рисунок 16.4) и даем ему привилегию Replication Slave, рисунок 16.5.

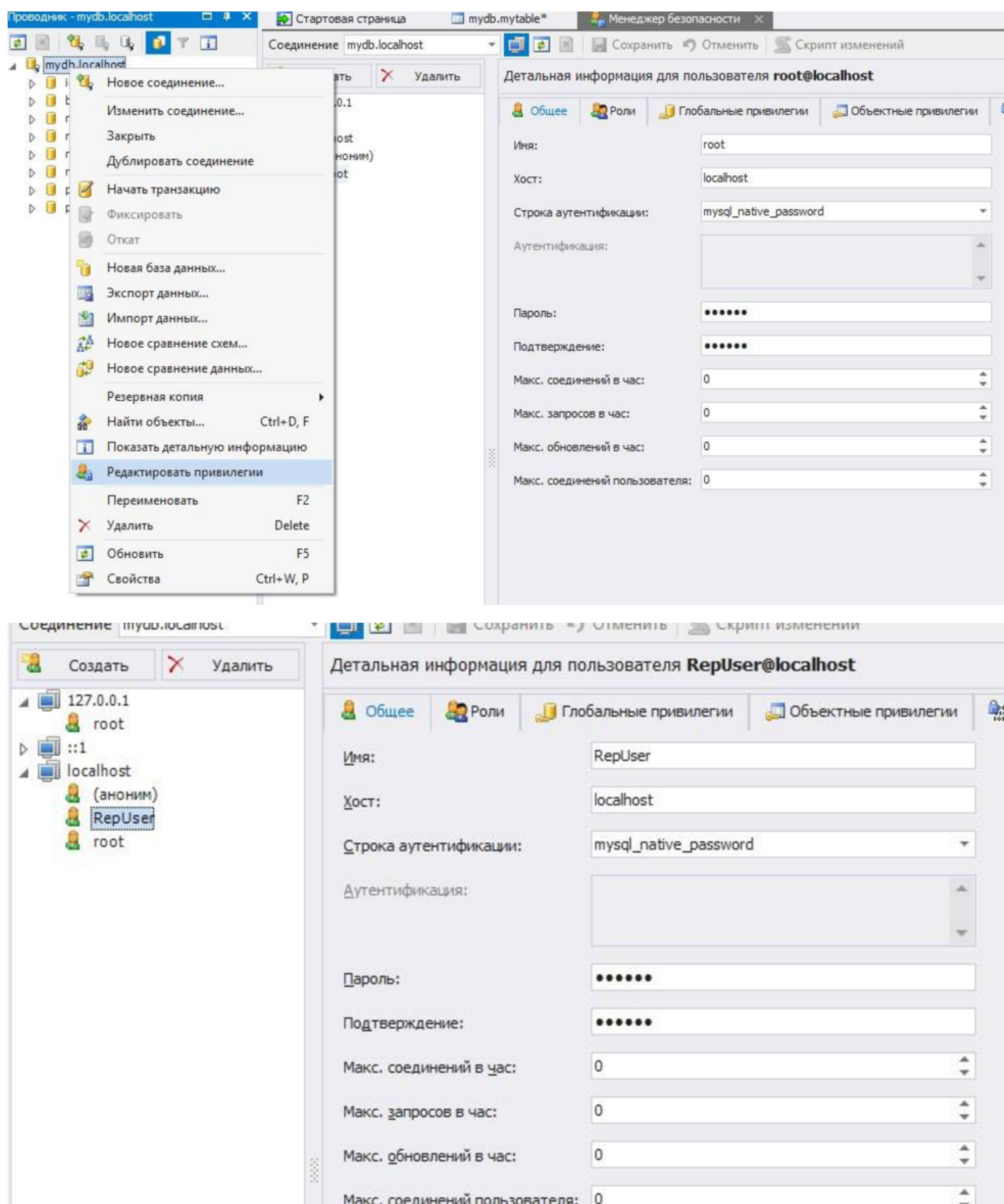


Рисунок 16.4 – Создание пользователя с паролем

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

Детальная информация для пользователя RepUser@localhost	
Общее	Роли
Глобальные привилегии	Объектные привилегии
SSL	
Название	☐ Выбранные
Alter	☐
Alter Routine	☐
Create	☐
Create Routine	☐
Create Tablespace	☐
Create Temporary Tables	☐
Create User	☐
Create View	☐
Delete	☐
Drop	☐
Event	☐
Execute	☐
File	☐
Index	☐
Insert	☐
Lock Tables	☐
Process	☐
References	☐
Reload	☐
Replication Client	☐
Replication Slave	☒
Select	☐
Show Databases	☐
Show View	☐
Shutdown	☐
Super	☐
Trigger	☐
Update	☐

Рисунок 16.5 – Этап 5

- Останавливаем головной сервер.
- Запускаем подчиненный сервер и подключаемся к нему. Запускаем файл StartMySQL, рисунок 16.6.

Поскольку Мы пока не меняли настройки порта подчиненного сервера, то подключаться можно с теми же параметрами что и к главному. То есть к порту 3306.

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

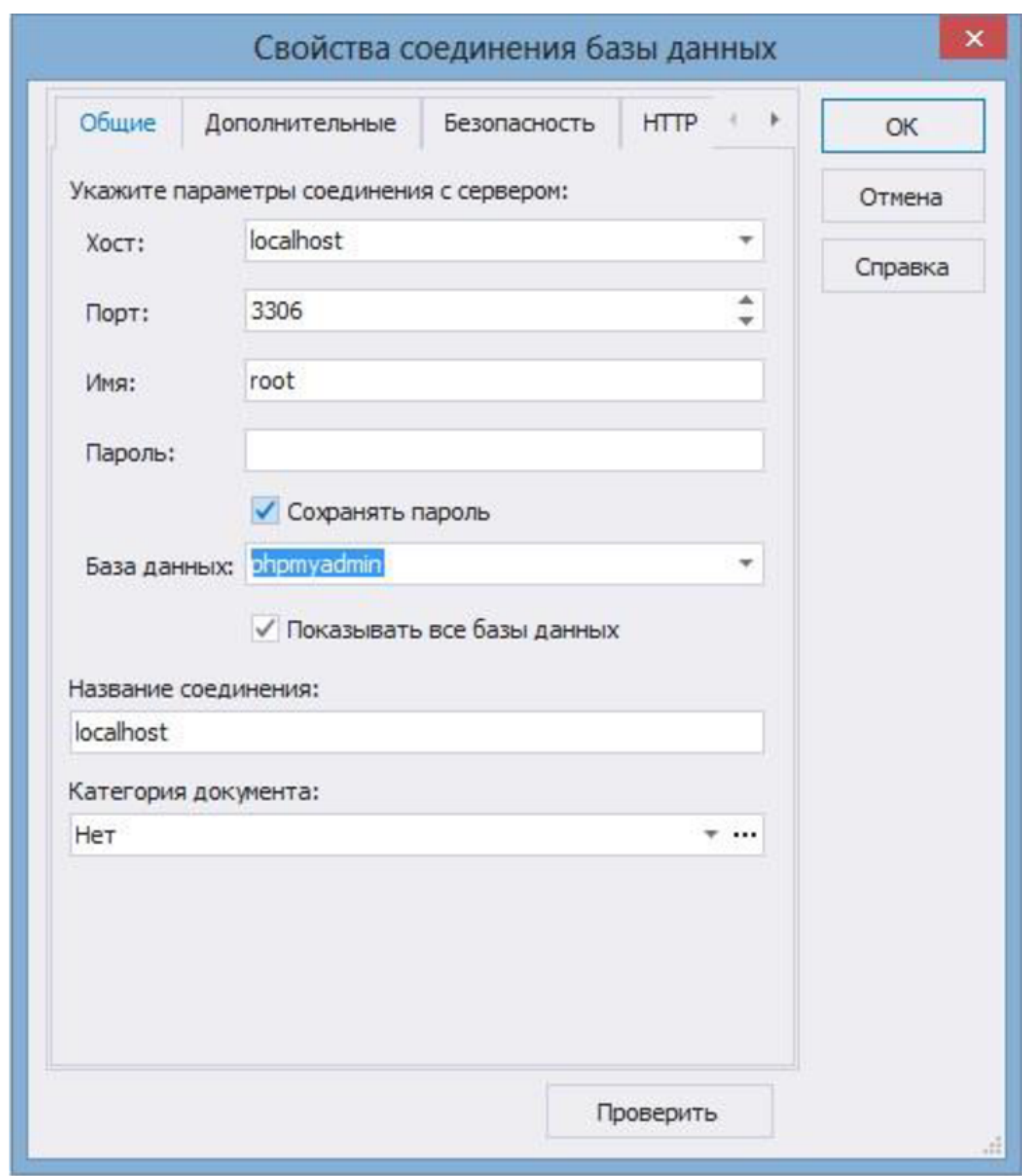
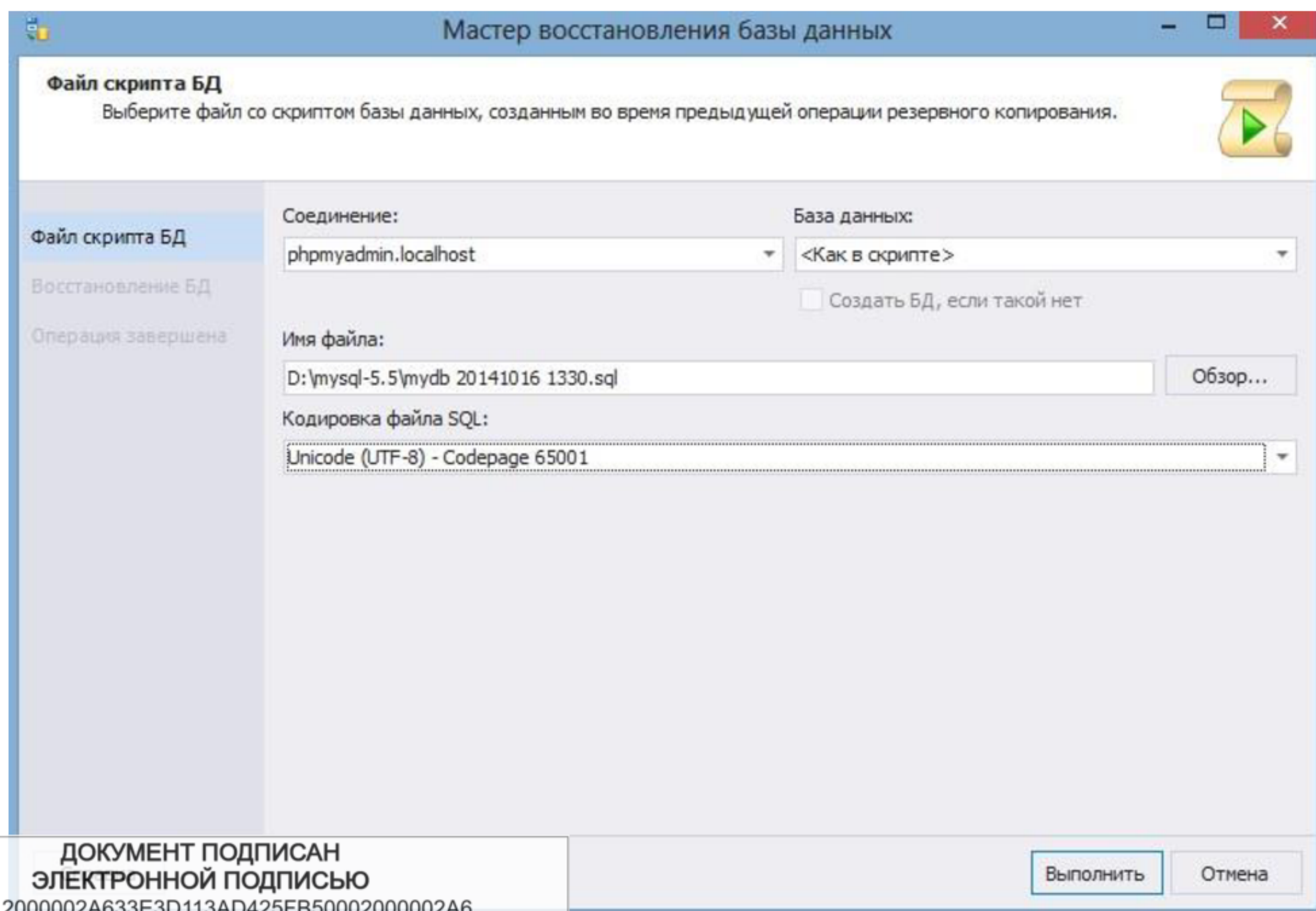
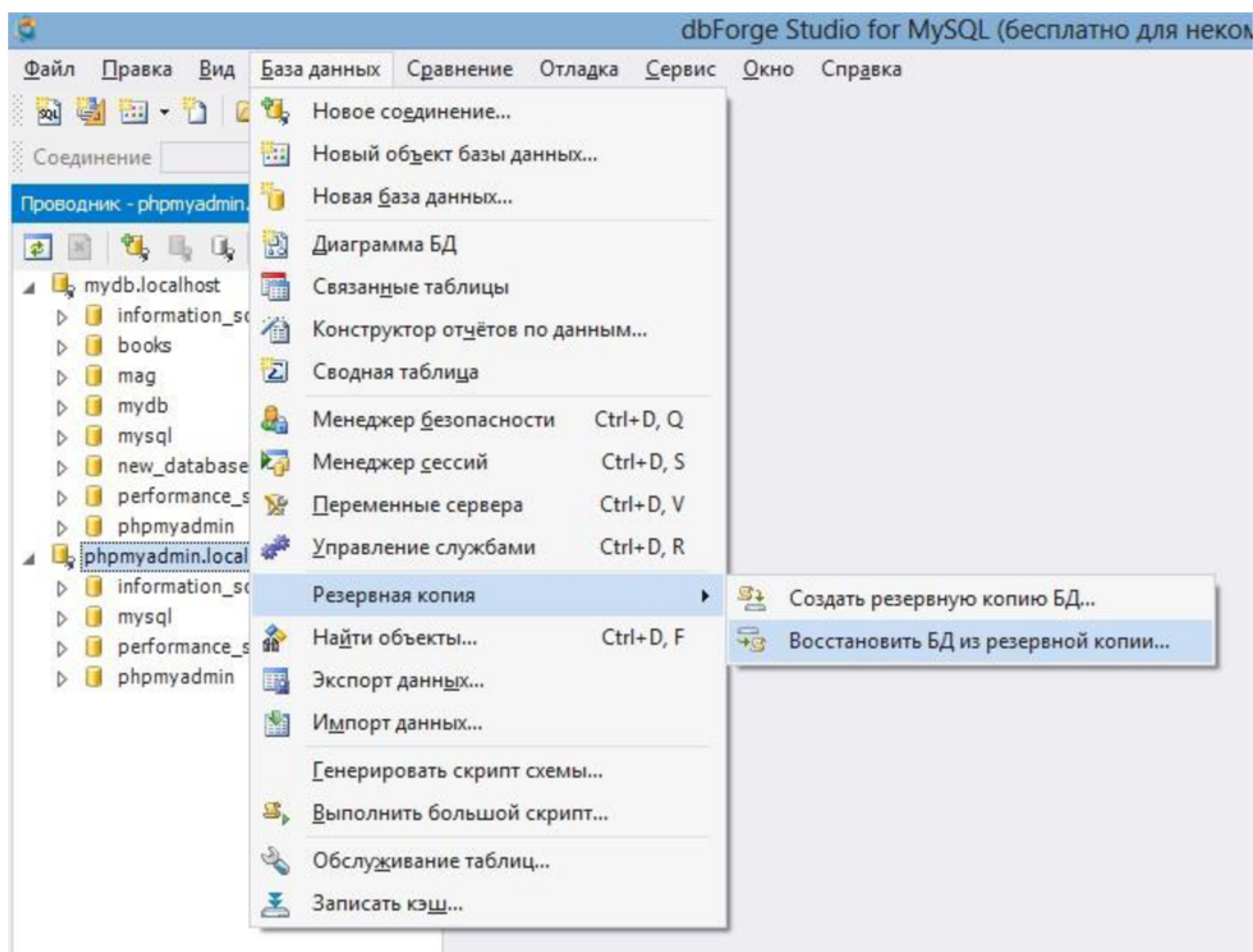


Рисунок 16.6 – Этап 7

8. Восстанавливаем базу данных «mydb» из архива, рисунок 16.7.



**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

9. Останавливаем подчиненный сервер (StopMySQL).
10. Настраиваем головной сервер.

Строка `server-id = 1` уже присутствует в конфигурационном файле. А строку `log-bin=mysql-bin` надо раскомментировать. Файл «my.ini» раздел `[mysqld]`

`server-id = 1 log-bin=mysql-bin`

С этого момента головной сервер будет записывать все изменения в базах данных, создание новых баз данных, создание пользователей.

11. Настраиваем подчиненный сервер. Изменяем порт на 3307, например, в двух местах и изменяем идентификатор сервера на 2, например.

Файл «my.ini»

`[client] port = 3307`

`[mysqld]`

`port = 3307`

`server-id = 2`

12. Запускаем головной сервер

13. Запускаем подчиненный сервер

14. Запускаем командную строку и переходим в папку bin подчиненного сервера

15. Запускаем из командной строки утилиту `mysql.exe` с параметрами порта (3307), хоста (127.0.0.1), и пользователя (root). Таким образом, мы подключимся к подчиненному серверу, рисунок 16.8. `mysql.exe --host=127.0.0.1 --port=3307 --user=root`

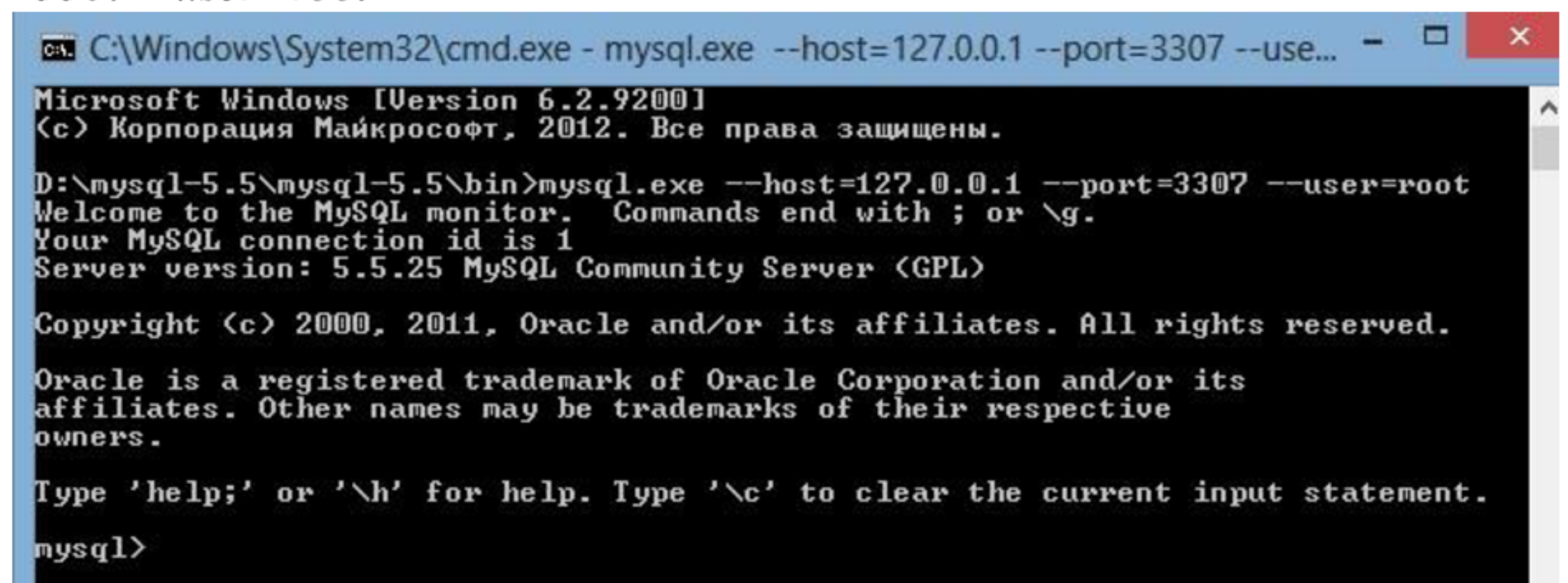
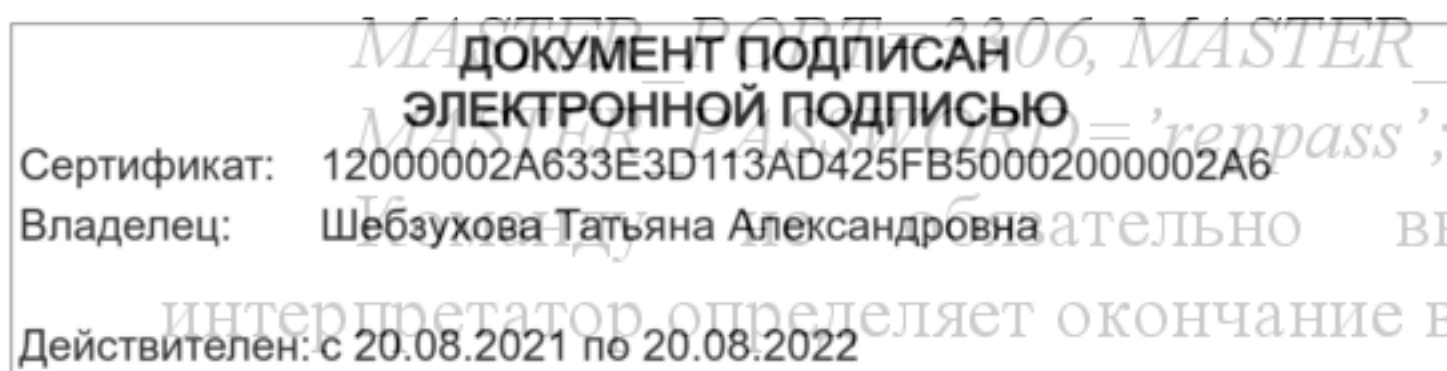


Рисунок 16.8 – Этап 15

16. Настраиваем репликацию. Нам нужно прописать имя головного сервера, порт, пользователя и пароль. Для этого настраиваем параметры MASTER -MASTER_HOST, MASTER_PORT, MASTER_USER и MASTER_PASSWORD. Для этого вводим команду:

`CHANGE MASTER TO MASTER_HOST='127.0.0.1',
MASTER_PORT=3306, MASTER_USER='RepUser',`



MASTER_PASSWORD='reppass';
Вводить одной строкой. Командный интерпретатор определяет окончание ввода по символу «;», рисунок 16.9.


```
mysql> change master to master_host='127.0.0.1',master_port=3306, master_user='RepUser', master_password='reppass';
Query OK, 0 rows affected (0.24 sec)

mysql>
```

Рисунок 16.9 – Этап 16

17. Запускаем репликацию командой

START SLAVE;

18. Проверяем состояние репликации (рисунок 16.10)

SHOW SLAVE STATUS \G

```
mysql> show slave status \G
***** 1. row *****
Slave_IO_State: Connecting to master
Master_Host: 127.0.0.1
Master_User: RepUser
Master_Port: 3306
Connect_Retry: 60
Master_Log_File:
Read_Master_Log_Pos: 4
Relay_Log_File: L220-onit-04-relay-bin.000001
Relay_Log_Pos: 4
Relay_Master_Log_File:
Slave_IO_Running: Connecting
Slave_SQL_Running: Yes
Replicate_Do_DB:
Replicate_Ignore_DB:
Replicate_Do_Table:
Replicate_Ignore_Table:
Replicate_Wild_Do_Table:
Replicate_Wild_Ignore_Table:
Last_Errno: 0
Last_Error:
Skip_Counter: 0
Exec_Master_Log_Pos: 0
Relay_Log_Space: 107
Until_Condition: None
Until_Log_File:
Until_Log_Pos: 0
Master_SSL_Allowed: No
Master_SSL_CA_File:
Master_SSL_CA_Path:
Master_SSL_Cert:
Master_SSL_Cipher:
Master_SSL_Key:
Seconds_Behind_Master: NULL
Master_SSL_Verify_Server_Cert: No
Last_IO_Errno: 1045
Last_IO_Error: error connecting to master 'RepUser@127.0.0.1:3306' - retry-time: 60 retries: 86400
Last_SQL_Errno: 0
Last_SQL_Error:
Replicate_Ignore_Server_Ids:
Master_Server_Id: 0
1 row in set (0.00 sec)

mysql> _
```

Рисунок 16.10 – Этап 18

19. Смотрим список баз данных, рисунок 16.11.

SHOW DATABASES;

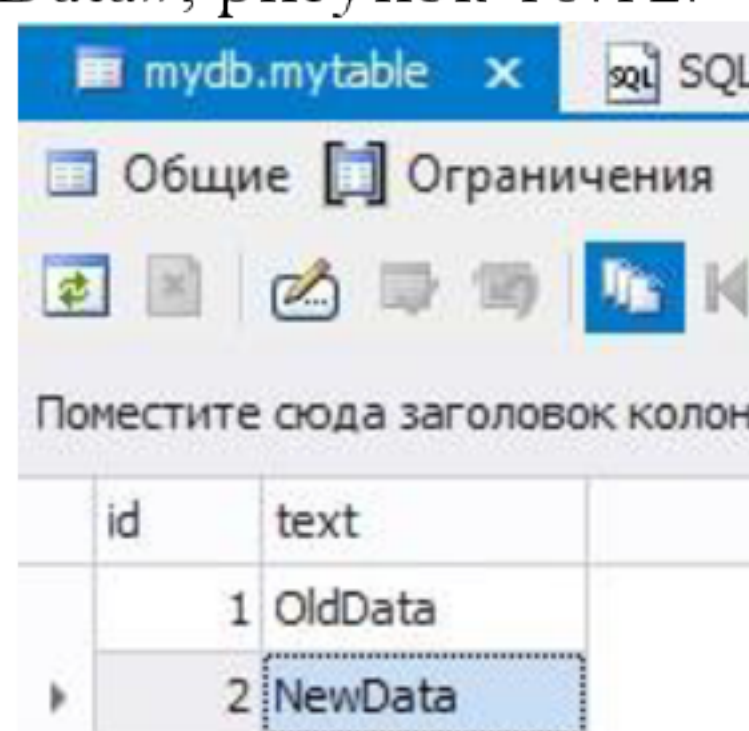
```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mydb      |
| mysql     |
| performance_schema |
| phpmyadmin |
+-----+
5 rows in set (0.00 sec)

mysql> _
```

Рисунок 16.11 – Этап 19

20. Не закрывая консоль подключаемся к головному серверу и создаем в головном сервере новую базу данных «newdb»

21. Добавляем в старую базу данных «mydb» в таблицу «mytable» новую строку, например «NewData», рисунок 16.12.



id	text
1	OldData
2	NewData

Рисунок 16.12 – Этап 21

22. В консоли, которую мы не закрыли, проверяем наличие новой базы данных. Для этого смотрим список баз данных, рисунок 16.13.

SHOW DATABASES;

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mydb      |
| mysql     |
| new_db    |
| performance_schema |
| phpmyadmin |
+-----+
```

Рисунок 16.13 – Этап 22

Как мы видим новая база данных «newdb» появилась в списке 23. Проверяем наличие новой строки в старой базе данных «mydb». Сначала подключаемся к базе, рисунок 16.14.

USE mydb;

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

```
mysql> Use mydb;
Database changed
mysql>
```

Рисунок 16.14 – Этап 23, подключение к БД

Выводим список строк, рисунок 16.15:

*SELECT * FROM mytable*

```
mysql> Use mydb;
Database changed
mysql> select * from mytable
-> ;
+----+-----+
| id | text  |
+----+-----+
| 1  |OldData|
| 2  |NewData|
+----+-----+
2 rows in set (0.00 sec)
```

Рисунок 16.15 – Этап 23

Как мы видим, на подчиненном сервере в таблице появилась новая строка.

24. Выход из командного интерпретатора mysql.exe осуществляется командой \q

\q

25. Выход из командного интерпретатора mysql.exe осуществляется командой exit

Exit

26. Оба сервера можно остановить. Теперь их можно запускать и останавливать в произвольном порядке. Головной сервер при любых изменениях в базе данных будет записывать все изменения в специальный файл, а подчиненный сервер при удачном подключении к главному выполнит все эти изменения у себя.

Обратить внимание. Для функционирования схемы необходимо не только скопировать базы данных на новый сервер, но и скопировать пользователей с их привилегиями. Вновь созданные пользователи будут копироваться на подчиненный сервер автоматически. Например, пользователь RepUser, которого мы создали ДО включения логгирования изменений на главном сервере, на подчиненном сервере отсутствует.

Задание

Организовать репликацию Вашей базы данных, разработанной в предыдущих лабораторных работах. Проверить ее работоспособность.

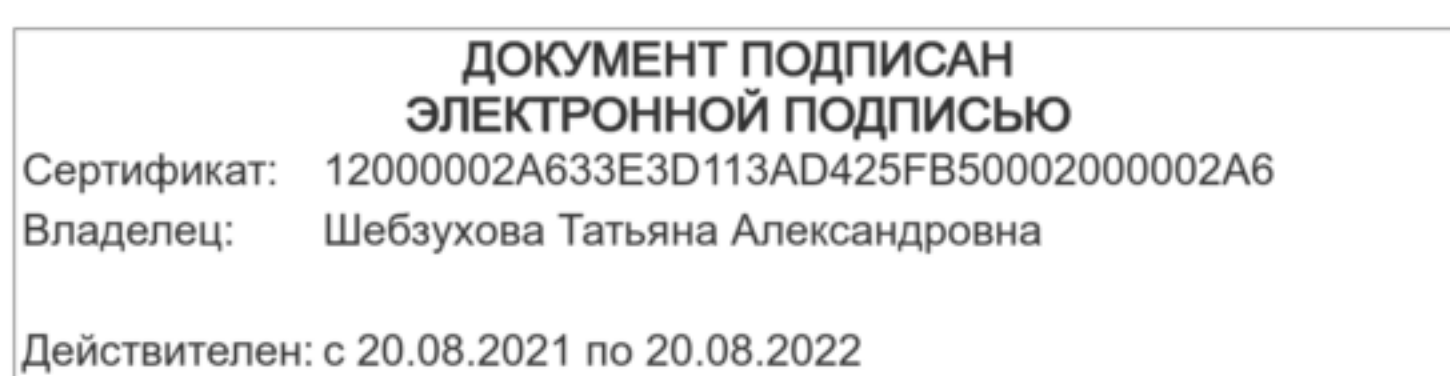
Содержание отчета и его форма

Отчет по лабораторной работе должен состоять из:

- 1) названия лабораторной работы;
- 2) описание примеров репликации.

Вопросы для защиты работы:

1. Каким образом осуществляется репликация?



Лабораторная работа №17. SQL-команды, относящиеся к репликации

Цель работы: научиться управлять репликациями в MySQL.

Теоретическое обоснование

Управление репликацией производится командами SQL. Ниже приводится краткое описание команд:

Команда	Описание
SLAVE START	Запускает поток подчиненного сервера (подчиненный сервер)
SLAVE STOP	Завершает поток подчиненного сервера (подчиненный сервер)
SET SQL_LOG_BIN=0	Блокирует ведение записей в журналах обновлений, если пользователь имеет привилегию SUPER. В противном случае ничего не выполняет (головной сервер)
SET SQL_LOG_BIN=1	Отменяет блокировку ведения записей в журналах обновлений, если пользователь имеет привилегию SUPER. В противном случае ничего не выполняет (головной сервер)
SET SQL_SLAVE_SKIP_COUNTER=n	Пропускает последующие n событий на головном сервере. Опция допустима, если поток подчиненного сервера не запущен, в противном случае будет выдана ошибка. Полезна для восстановления после сбоев репликации.
RESET MASTER	Удаляет все двоичные журналы, перечисленные в индексном файле, и делает индексный файл двоичных журналов пустым. Для версий ниже 3.23.26 используйте команду FLUSH SLAVE (головной сервер)
RESET SLAVE	Заставляет подчиненный сервер "забыть" свою точку положения репликации в журналах головного сервера. Для версий ниже 3.23.26 эта команда называется FLUSH SLAVE (подчиненный сервер)

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

LOAD TABLE tblname FROM MASTER	Загружает копию таблицы из головного на подчиненный сервер. Используется главным образом для отладки команды LOAD DATA FROM MASTER, но некоторые "пользователи-
-----------------------------------	---

	гурманы" могут найти ей и другие применения. Если вы относите себя к числу обычных, не отягощенных хакерскими амбициями пользователей, данную опцию применять не стоит (подчиненный сервер).
LOAD DATA FROM MASTER	Присутствует в версиях выше 4.0.0. Создает образ головного сервера и копирует его на подчиненный сервер. Обновляет значения MASTER_LOG_FILE и MASTER_LO G_POS таким образом, чтобы подчиненный сервер начинал репликацию из конкретной позиции. Будет обрабатывать ограничения таблиц и баз данных, указанные в опциях replicate-*. При этом, пока происходит создание образа, могут использоваться лишь таблицы MyISAM и требуется глобальная блокировка чтения на головном сервере. В будущем планируется обеспечить работу этой команды с таблицами InnoDB и устранить необходимость глобальной блокировки чтения при помощи интерактивного резервного копирования, не требующего блокировки.
CHANGE MASTER TO master_def_list	Заменяет параметры головного сервера значениям и, заданным в списке master_def_list, и перезапускает поток подчиненного сервера. mast er_def_list – это список с разделителем-запятой, содержащий значения master_def, где master_def – одно из следующих значений: MASTER_HOST, MASTER_USER, MASTER_PASSWORD, MASTER_PORT, MASTER_CONNECT_RETRY, MASTER_LOG_FILE, MASTER_LOG_POS. Например:

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ Сертификат: 12000002A633E3D113AD425FB50002000002A6 Владелец: Шебзухова Татьяна Александровна Действителен: с 20.08.2021 по 20.08.2022
--

	<pre>CHANGE MASTER TO MASTER_HOST='master2.mycompany.com', MASTER_USER='replication', MASTER_PASSWORD='big3cret', MASTER_PORT=3306, MASTER_LOG_FILE='master2-bin.001', MASTER_LOG_POS=4;</pre>
	Следует указывать только те значения, которые подлежат изменению. Не указанные значения останутся неизменными, за исключением тех случаев, когда изменяется хост или порт. В этом

	случае подчиненный сервер считает, что поскольку изменяется хост или порт, головной сервер становится другим. Следовательно, старые значения и точки положения в журнале будут автоматически заменены на значение пустой строки и 0 соответственно (начальные значения). Обратите внимание: если подчиненный сервер перезапускается, он сохраняет "память" о своем последнем головном сервере. Если это нежелательно, можно перед перезапуском удалить файл 'master.info' - тогда подчиненный сервер будет считывать информацию о своем головном сервере из файла 'my.cnf' или из командной строки. Эта команда используется для настройки подчиненного сервера при наличии образа головного сервера, а также записей из журнала и сдвига головного сервера, которые соответствуют образу. Можно выполнить команду
	<pre>CHANGE MASTER TO MASTER_LOG_FILE='log_name_on_master', MASTER_LOG_POS=log_offset_on_master</pre>
	на подчиненном сервере после восстановления образа (подчиненный сервер)
SHOW MASTER STATUS ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ Сертификат: 12000002A633E3D113AD425FB50002000002A6 Владелец: Шебзухова Татьяна Александровна	Выводит информацию о состоянии головного сервера, исходя из информации в журналах (головной сервер)

STATUS
 Действителен: с 20.08.2021 по 20.08.2022

SHOW SLAVE HOSTS	Присутствует в версии 4.0.0 и выше. Выводит список подчиненных серверов, связанных в текущее время с головным сервером (подчиненный сервер)
SHOW SLAVE STATUS	Выводит информацию о состоянии существенных параметров потока подчиненного сервера (головной сервер)
SHOW MASTER LOGS	Присутствует только начиная с версии 3.23.28. Выводит список двоичных журналов головного сервера. Эту команду следует использовать перед вызовом команды PURGE MASTER LOGS TO для определения того, какие из журналов можно удалить (головной сервер)
SHOW BINLOG EVENTS [IN 'logname'] [FROM pos] [LIMIT [offset,] rows	

] @tab Показывает события в двоичном журнале обновлений. Преимущественно применяется для тестирования/отладки, но может также использоваться и для обычных клиентов, по какой-либо причине нуждающихся в чтении содержимого двоичных журналов (головной сервер).	
---	--

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

<pre> SHOW NEW MASTER FOR SLAVE WITH MASTER_LOG_FILE='logfile' AND MASTER_LOG_POS=pos AND MASTER_LOG_SEQ=log_seq AND MASTER_SERVER_ID=server_id </pre>	Эта команда используется, когда подчиненному серверу, связанному с головным сервером, который, возможно, является "мертвым" или недоступным, нужно отключить репликации на другом подчиненном сервере, связанном с тем же головным сервером. Команда возвратит пересчитанные координаты репликации, и вывод этой команды может использоваться в последующей команде CHANGE MASTER TO. Обычным пользователям данная команда, как правило, никогда не понадобится: она главным образом служит для внутреннего использования в отказобезопасном репликационном коде. В будущем возможны изменения синтаксиса опции, если будет найден более интуитивно понятный способ описания этой операции.
<pre> PURGE MASTER LOGS TO 'logname' </pre>	Присутствует начиная с версии 3.23.28. Удаляет все журналы репликации, которые перечислены в индексном файле журналов до передаваемого журнала, и удаляет их из индексного файла журналов. Таким образом передаваемый журнал становится первым в индексном файле журналов. Пример:
	<pre> PURGE MASTER LOGS TO 'mysql-bin.010' </pre>
	Эта команда не выполнит никаких действий и возвратит ошибку, если имеется активный подчиненный сервер, который в текущее время читает данные из одного из журналов, который должен быть удален. Однако если имеется бездействующий подчиненный сервер и

	происходит удаление одного из журналов, который он хочет прочитать, то после того, как подчиненный сервер "поднимется", он станет неспособным к репликации. Команда может быть безопасно выполнена на подчиненных серверах во время процесса репликации - не нужно останавливать процесс. Сначала необходимо проверить все подчиненные серверы при помощи команды SHOW SLAVE STATUS, чтобы увидеть, какой журнал используется, затем вывести список журналов головного сервера при помощи команды SHOW MASTER LOGS, найти самый ранний журнал среди всех подчиненных серверов (если все подчиненные серверы получили последние обновления, это будет последний журнал в списке), сделать резервные копии всех журналов, которые должны быть удалены (необязательно), и очистить все до целевого журнала.
--	---

Задание

Изучить назначение команд, относящихся к репликации БД.

Содержание отчета и его форма

Отчет по лабораторной работе должен состоять из:

- 1) названия лабораторной работы;
- 2) описание примеров использования команд репликации.

Вопросы для защиты работы:

1. Какие команды используются для репликации данных?

Лабораторная работа №18. Проблемы и распространённые ошибки MySQL

Цель работы: научиться работать с СУБДMySQL в режиме клиентсервер.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ	
Сертификат:	12000002A633E3D113AD425FB50002000002A6
Владелец:	Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022	

Теоретическое обоснование

Как определить, чем вызваны проблемы? При возникновении проблемы прежде всего следует обнаружить ее источник - программу или элемент оборудования.

Если присутствует один из следующих симптомов, то проблема, скорее всего, связана с аппаратным обеспечением (с памятью, материнской платой, процессором или жестким диском) либо с ядром:

- ☐ Не работает клавиатура. Обычно ее работоспособность можно проверить по реакции на нажатие Caps Lock. Если индикатор Caps Lock не меняется, то клавиатуру необходимо заменить (прежде чем это сделать, следует попробовать перезагрузить компьютер и проверить все кабели, подключенные к клавиатуре).
- ☐ Не перемещается курсор мыши.
- ☐ Машина не отвечает на ping-запросы удаленной машины.
- ☐ Различные не связанные между собой программы не работают, как надо.
- ☐ Система неожиданно перезагрузилась (дефектная программа пользовательского уровня **никогда** не должна быть способна вызвать отказ системы).

В этом случае необходимо начать с проверки всех кабелей и запуска диагностических средств для проверки аппаратуры.

Следует также проверить, нет ли патчей, обновлений, сервисных пакетов (service pack) для используемой операционной системы, при помощи которых вы, возможно, могли бы решить проблемы. Кроме того, следует удостовериться, что у вас установлены достаточно свежие версии библиотек (таких как glibc). Для раннего обнаружения проблем хорошо использовать машину с ECC-памятью.

В случае блокировки клавиатуры положение можно исправить, если войти на свою машину с другой машины и выполнить на своей машине `kbd_mode -a`.

Исследуйте свой системный журнальный файл ('/var/log/messages' или т.п.) на предмет причин возникающих проблем. Если есть основания полагать, что проблема - в MySQL, то следует также изучить журнальные файлы MySQL.

Если вы считаете, что аппаратные проблемы отсутствуют, то следует попробовать обнаружить вызывающую проблемы программу. Попробуйте с помощью `top`, `ps`, `taskmanager` или подобной программы проверить, какая программа забирает все ресурсы процессора или блокирует машину.

Проверьте с помощью `top`, `df` или подобной программы, нет ли нехватки памяти, дискового пространства, дескрипторов для открытия файлов или каких-либо других критических ресурсов.

Если проблема связана с бесконтрольным процессом, то всегда можно попробовать уничтожить его. Если он не хочет уничтожаться, то, вероятно, существует ошибка в операционной системе.

Если вы проверили все возможные причины вы сделали вывод, что источник проблемы именно MySQL-сервер или клиент, то следует сделать отчет об ошибке для команды поддержки. В отчете об ошибке постарайтесь дать очень подробное описание поведения системы и свое мнение

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

по поводу происходящего. Следует также объяснить, почему вы считаете, что проблемы вызывает именно MySQL. Примите во внимание все ситуации, описанные в данном разделе. Опишите все проблемы в точности так, как они наблюдаются при исследовании системы. При помещении в отчет для всего вывода программ и/или их сообщений об ошибках и/или подобной информации из журнальных файлов используйте метод "вырезать и вставить". При обращении к группе поддержки MySQL необходимо

детально описать, какая именно программа не работает, и какие симптомы вы наблюдали! Нам доводилось получать много отчетов об ошибках, где просто утверждалось, что "система не работает", - такие отчеты не давали никакой информации о характере возможной проблемы.

Если программа сбоит, то всегда полезно выяснить:

Не вызвала ли данная программа ошибки сегментации (core dump)?

Не забирает ли программа все ресурсы процессора? Проверьте с помощью `top`. Дайте программе немного поработать - возможно, она занимается сложными вычислениями.

Если проблемы вызваны именно сервером `mysqld`, то можно ли выполнить `mysqladmin -u root ping` или `mysqladmin -u root processlist`?

Что сообщает клиентская программа (попробуйте поработать, например, с `mysql`) при попытке соединиться с MySQL? Происходит ли заклинивание клиента? Выдает ли программа какой-нибудь вывод?

Коды ошибки представлены в таблице 18.1. Какой код из них вы получите, зависит от ОС.

Таблица 18.1 – Коды ошибок

Код ошибки	Описание
CR_SERVER_GONE_ERROR	Клиент не может послать запрос серверу.
CR_SERVER_LOST	Клиент не получил ошибки при передаче запроса серверу, но он не получил также полного ответа (или хоть какого-то ответа) на запрос.

Ошибка будет также выдана, если кто-нибудь уничтожит выполняющийся поток посредством `kill` номер потока.

Проверить, что MySQL на ходу, можно, запустив `mysqladmin version` и изучив время работы (`uptime`). Если проблема в аварийном завершении `mysqld`, то необходимо сосредоточиться на поиске причины аварии. В этом случае следует сначала проверить, не будет ли уничтожен MySQL снова при повторном задании запроса.

Этот документ также выдаться при посылке серверу неверного или большого пакета, то сервер предполагает, что с клиентом что-то не так, и выдаст ошибку. Если необходимо выполнять объемные запросы

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен с 20.08.2021 по 20.08.2022

(например, при работе с большими столбцами типа BLOB), можно увеличить предельный размер запроса, запустив `mysqld` с опцией `O max_allowed_packet=#` (по умолчанию 1 Мб). Дополнительная память выделяется по требованию, так что `mysqld` будет использовать больше памяти только в случае, когда выдан большой запрос или когда `mysqld` должен вернуть большую строку результата.

Задание

Изучить распространённые ошибки и методы их решения.

Содержание отчета и его форма

Отчет по лабораторной работе должен состоять из:

- 1) названия лабораторной работы;
- 2) описание проблем и распространенных ошибок MySQL.

Вопросы для защиты работы:

1. Какие проблемы могут возникнуть при работе с MySQL?

Список рекомендуемой литературы

а) основная литература:

1. Агальцов В.П. Базы данных. В 2 книгах. Книга 2 Распределенные и удаленные базы данных. Учебник. М.:ИД ФОРУМ НИЦ Инфра-М, 2013. – 2013 (электронная библиотечная система znanium.com)
2. Справочное руководство по MySQL. [Электронный ресурс]: <http://www.php.su/mysql/manual/>
3. <http://www.intuit.ru>
4. <http://www.citforum.ru>
5. <http://guardmag.com>

б) дополнительная литература:

6. PHP, MySQL, XML: программирование для Интернета.
Автор: Бенкен Е.С. Издательство: БХВ-Петербург, 2007 г. 314 с.
7. Максимов, Е.М. Базы данных в системах управления производственными процессами: учебное пособие [Текст] / Максимов Е.М., Бахтадзе Н.Н.: Издательство Московского государственного открытого университета, 2011 г. 160 с.
8. Туманов, В.Е. Проектирование хранилищ данных для систем бизнес-аналитики: учебное пособие [Текст] / Туманов В.Е.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2010 г.
9. Чекмарев, Ю.В. Локальные вычислительные сети: Учебное пособие [Текст] / Ю.В. Чомаев. ЛМК Пресс, 2009 г.

10. Рудин, П.П. Практика написания хорошего кода, П.

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**Федеральное государственное автономное образовательное учреждение высшего
образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

Пятигорский институт (филиал) СКФУ

Управление данными

Методические указания к выполнению курсовой работы

Направление подготовки 09.03.02 «Информационные системы и технологии»

Квалификация (степень) выпускника – бакалавр

Реализуется в 5,6 семестрах

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**

Сертификат: 12000002A633E3D113AD425FB50002000002A6

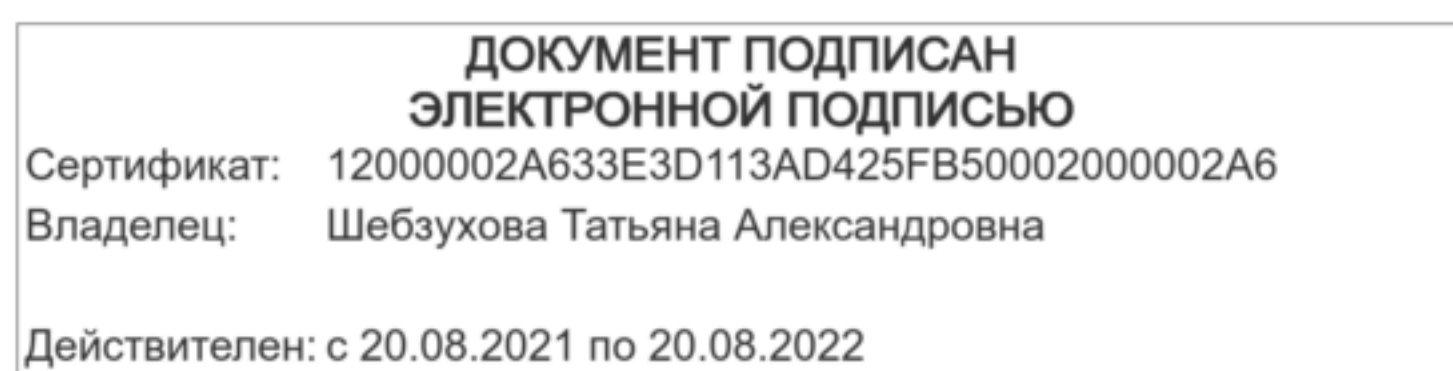
Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

Пятигорск, 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	131
1. ЦЕЛЬ И ЗАДАЧИ	132
2. ФОРМУЛИРОВКА ЗАДАНИЯ	133
3. СТРУКТУРА РАБОТЫ	147
4. ОБЩИЕ ТРЕБОВАНИЯ К НАПИСАНИЮ И ОФОРМЛЕНИЮ РАБОТЫ.....	171
5. ПОСЛЕДОВАТЕЛЬНОСТЬ ВЫПОЛНЕНИЯ ЗАДАНИЯ	176
6. КРИТЕРИИ ОЦЕНИВАНИЯ ПРОЕКТА	177
7. ПОРЯДОК ЗАЩИТЫ ПРОЕКТА	177
СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ.....	178
ПРИЛОЖЕНИЕ 1	179
ПРИЛОЖЕНИЕ 2	181
ПРИЛОЖЕНИЕ 3	182



ВВЕДЕНИЕ

«История исследований систем баз данных – это, по сути, история развития приложений, достигших исключительной производительности и оказавших потрясающее воздействие на экономику. Достижения в исследованиях баз данных стали основой фундаментальных разработок коммуникационных систем, транспорта и логистики, финансового менеджмента, систем с базами знаний, а также большого количества гражданских и военных приложений. Они послужили фундаментом значительного прогресса в ведущих областях науки – от информатики до биологии». Эта цитата материалов международного семинара по системам баз данных отражает исключительную важность их применения, с одной стороны, как основы практически любой информационной системы, с другой стороны, как катализатора научных достижений не только в области программного обеспечения, но и во многих других областях науки.

Поэтому изучение дисциплины «Управление данными» бакалаврами направления 09.03.02 «Информационные системы и технологии» является одним из важнейших этапов их специальной подготовки, а также подготовки к последующему изучению курсов «Технологии обработки информации», «Корпоративные информационные системы», «Предметно-ориентированные информационные системы», «Методы и средства проектирования информационных систем и технологий», «Управление данными в сетях», а также при подготовке выпускной квалификационной работы.

Курсовое проектирование по дисциплине «Управление данными» осуществляется в течение всего семестра, в котором изучается данная дисциплина, и предполагает самостоятельное проектирование студентом базы данных (БД) в среде одной из промышленных систем управления базами данных (СУБД).

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

1. ЦЕЛЬ И ЗАДАЧИ

Курсовая работа является самостоятельной учебно-исследовательской работой студента и представляет собой логически завершенное и оформленное научное исследование. Цель курсового проектирования – формирование у студента навыков научно-исследовательской работы, повышение уровня его профессиональной (теоретической и практической) подготовки, углубление знаний по учебной дисциплине, развитие интереса и навыков самостоятельной работы с научной и справочной литературой.

Целью его выполнения курсовой работы является закрепление, углубление и обобщение теоретических знаний и практических навыков, выработка навыков самостоятельной работы по разработке баз данных и информационных систем. Для реализации этой цели тематикой и требованиями к выполнению курсовой работы предусматривается выполнение основных этапов проектирования и создания баз данных, включая формулировку цели и задач проектирования, изучение предметной области, моделирование данных, разработку средств обеспечения защиты данных от несанкционированного доступа, проектирование интерфейса и выходных форм.

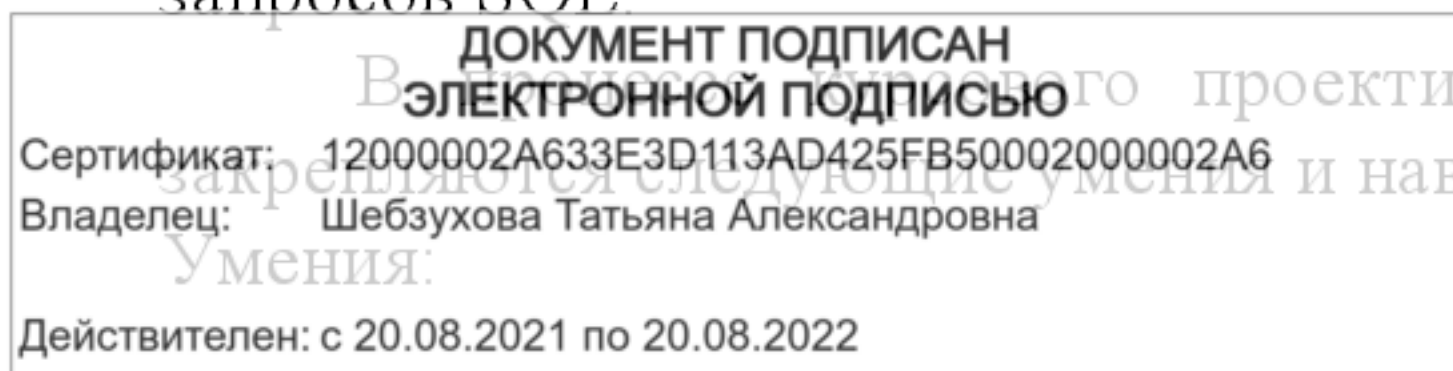
Выполнение курсовой работы имеет целью расширение знаний студентов, обучение методам теоретического анализа явлений и закономерностей науки, отработку навыков самостоятельного применения теоретических знаний к комплексному решению профессиональных задач, использования справочной литературы, методов математической обработки экспериментальных данных, компьютерных технологий.

Задание к курсовой работы по дисциплине «Управление данными» содержит теоретическую часть и практическое задание. При выполнении теоретического задания студентом должны решаться следующие задачи:

- приобретение новых теоретических знаний в соответствии с темой работы и заданием руководителя;
- умение систематизировать, обобщать и логично излагать концепции, альтернативные точки зрения по исследуемой проблеме;
- развитие учебно-исследовательских и методических навыков, необходимых для системного научного анализа изучаемого явления;
- совершенствование профессиональной подготовки.

При выполнении практического задания решаются задачи:

- приобретение практических навыков обследования предметной области, концептуального, логического и физического проектирования базы данных;
- освоение средств поддержания целостности БД;
- изучение языка запросов SQL.



- анализировать информационную среду предметной области и устанавливать структурное представление и взаимосвязи с другими компонентами информационного пространства, информационные потоки, систематизировать документооборот; существующий рынок аппаратного и программного обеспечения, классифицировать существующие информационные системы, определять направления их развития;
- формулировать запросы к базам данных на естественном языке, с помощью абстрактных реляционных языков и на языке SQL;
- разрабатывать информационно-логическую, функциональную и объектно-ориентированную модели информационной системы, модели данных информационных систем.

Навыки:

- работы с программно-техническими средствами проектирования БД в интерактивном режиме;
- методами и средствами представления данных о предметной области.

2. ФОРМУЛИРОВКА ЗАДАНИЯ

Курсовая работа выполняется на общую тему: «Разработка приложения для управления данными в базе данных (для заданной предметной области) средствами Visual Studio»

Задание на курсовую работу состоит из двух частей: теоретической и практической.

Теоретическая часть

1. Анализ тенденций развития отдельных направлений управления данными.

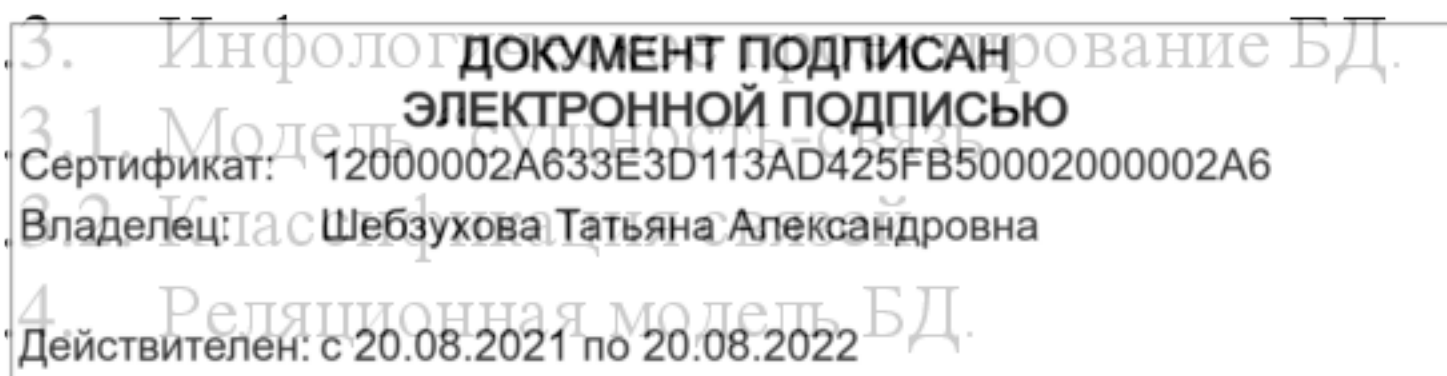
Практическая часть

- 2.1. Разработка базы данных
- 2.2. Разработка приложения для управления данными
 - 2.2.1. Добавление записей
 - 2.2.2. Удаление записей
 - 2.2.3. Редактирование записей
- 2.3. Разработка экранных форм и отчетов

2. Разработка локальной реляционной базы данных.

- 2.1. Обследование предметной области.
- 2.2. Концептуальное проектирование.
 - 2.2.1. Перечень сущностей (обосновать список).
 - 2.2.2. Перечень атрибутов.

- 2.3. Информационное проектирование БД.
 - 2.3.1. Модель логической БД.
 - 2.3.2. Классификация БД.
- 2.4. Реляционная модель БД.



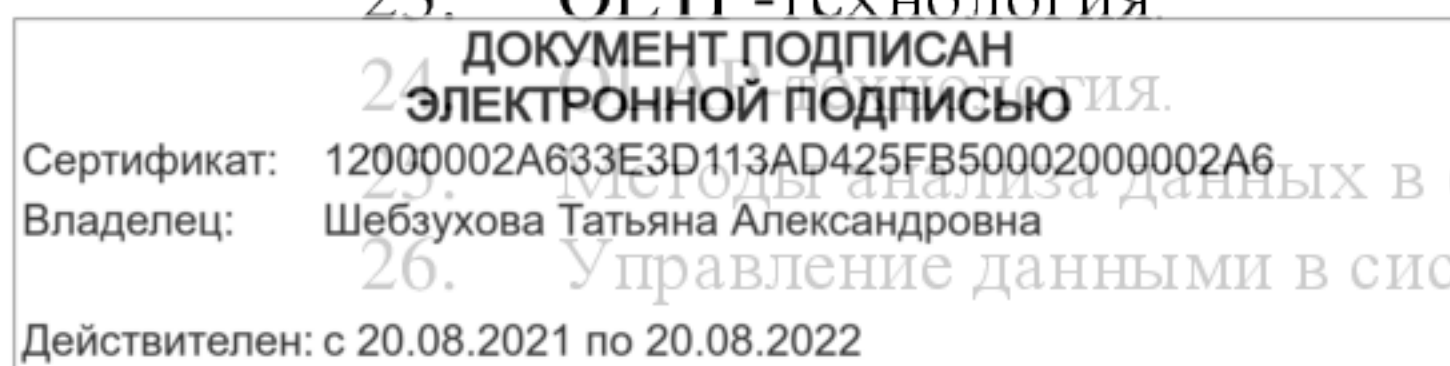
- 2.4.1. Функциональные зависимости между атрибутами.
- 2.4.2. Выбор ключей.
- 2.4.3. Нормализация отношений.
- 2.5. Даталогическое проектирование БД.
- 2.5.1. Состав таблиц БД.
- 2.5.2. Средства поддержания целостности.
- 2.6. Запросы к БД.
- 2.7. Разработка механизмов защиты данных от несанкционированного доступа.
- 2.8. Требования к техническому обеспечению.
- 2.9. Инструкция по использованию БД.
- 2.9.1. Вызов программы.
- 2.9.2. Экранные формы.
- 2.9.3. Описание отчетов.

2.1. Тематика теоретических заданий на курсовой проект

Теоретическое задание состоит в анализе тенденций развития отдельных направлений управления данными.

Примерные темы вариантов для выполнения теоретического задания

1. Стандартизация языка SQL. Стандарт SQL/MM (Multimedia).
2. Темпоральные базы данных.
3. Стандартизация языка SQL. SQL/XML
4. Дедуктивные базы данных.
5. Стандартизация языка SQL. SQL/JRT (Java Routines and Types).
6. Обзор Российского рынка СУБД.
7. Разработка новых архитектур СУБД.
8. Средства управления доступом.
9. Администрирование БД.
10. Гипертекстовые БД.
11. Объектно-ориентированные БД.
12. Коммерческие БД.
13. Средства управления транзакциями.
14. Распределенная обработка данных.
15. Параллельная обработка данных.
16. Сжатие информации в БД.
17. Фрактальные методы сжатия информации.
18. Файловые и бесфайловые методы организации данных.
19. Тенденции развития методов организации данных.
20. Оптимизация запросов.
21. Тенденции построения файловых систем.
22. Странично-сегментная организации данных.
23. OLTP-технология.



24. Методы анализа данных в фактографических системах.

25. Управление данными в системах Data Grids.

27. Облачные вычисления (Cloud computing).
28. NoSQL СУБД.
29. Перспективы развития СУБД.
30. Арендная модель инфраструктурных платформ.
31. Проблемы стандартизации СУБД.
34. СУБД ведущих фирм-производителей.
35. Языковые средства управления данными.
36. Глубинный анализ данных.
37. Архитектура хранилищ данных.
38. Методы распределенной обработки данных.
39. Быстродействие и включение измерения времени в СУБД (In memory DB).
40. Эволюция технологий управления данными.

2.2 Предметные области для выполнения практической части курсового проекта

Практическая часть курсовой работы представляет собой разработку реляционной базы данных для заданной предметной области, а также приложения для управления данными в этой БД. Примеры предметных областей и атрибуты их характеризующие приведены ниже.

Вариант 1 БД Ресторана

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Склад (Код ингредиента, Наименование ингредиента, Дата выпуска, Объем, Срок годности, Стоимость, Поставщик)[10 записей]. Меню (Код блюда, Наименование блюда, Код ингредиента 1, Объем ингредиента 1, Код ингредиента 2, Объем ингредиента 2, Код ингредиента 3, Объем ингредиента 3, Стоимость, Время приготовления)[10 записей]. Заказ (Дата, Время, ФИО заказчика, Телефон, Код блюда 1, Код блюда 2, Код блюда 3, Стоимость, Отметка о выполнении, Код сотрудника)[10 записей].
Запросы:	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности"). Меню (Связывает таблицы "Меню" и "Склад" по полям "Код ингредиента", "Код ингредиента 1", "Код ингредиента 2" и "Код ингредиента 3"). Заказ (Связывает таблицы "Заказ", "Меню" и "Сотрудники" по полям "Код блюда", "Код блюда 1", "Код блюда 2", "Код блюда 3" и "Код сотрудника").

Вариант 2. БД Банка

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей].
-----------------	---

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

	Вклады (Код вклада, Наименование вклада, Минимальный срок вклада, Минимальная сумма вклада, Код валюты, Процентная ставка, Дополнительные условия)[5 записей]. Валюта (Код валюты, Наименование, Обменный курс)[3 записи]. Вкладчики (ФИО вкладчика, Адрес, Телефон, Паспортные данные, Дата вклада, Дата возврата, Код вклада, Сумма вклада, Сумма возврата, Отметка о возврате вклада, Код сотрудника)[10 записей].
Запросы:	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности"). Вклады (Связывает таблицы "Вклады" и "Валюта" по полю "Код валюты"). Вкладчики (Связывает таблицы "Вкладчики", "Вклады" и "Сотрудники" по полям "Код вклада" и "Код сотрудника").

Вариант 3. БД Больницы.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Лекарства (Код лекарства, Наименование, Показания, Противопоказания, Упаковка, Стоимость)[5 записей]. Болезни (Код болезни, Наименование, Симптомы, Продолжительность, Последствия, Код лекарства 1, Код лекарства 2, Код лекарства 3)[10 записей]. Пациенты (ФИО пациента, Возраст, Пол, Адрес, Телефон, Дата обращения, Код болезни, Код сотрудника, Результат лечения)[10 записей].
Запросы:	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности"). Болезни (Связывает таблицы "Болезни" и "Лекарства" по полю "Код лекарства", "Код лекарства 1", "Код лекарства 2" и "Код лекарства 3"). Пациенты (Связывает таблицы "Пациенты", "Болезни" и "Сотрудники" по полям "Код болезни" и "Код сотрудника").

Вариант 4. БД Гостиницы.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Номера (Код номера, Наименование, Вместимость, Описание, Стоимость, Код сотрудника)[5 записей]. Услуги (Код услуги, Наименование, Описание, Стоимость)[5 записей]. Клиенты (ФИО, Паспортные данные, Дата заселения, Дата выезда, Код номера, Код услуги 1, Код услуги 2, Код услуги 3, Стоимость, Код сотрудника)
Запросы:	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности"). Связывает таблицы "Сотрудники" и "Номера" по полю "Код

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

	Клиенты (Связывает таблицы "Клиенты", "Номера", "Услуги" и "Сотрудники" по полям "Код номера", "Код услуги", "Код услуги 1", "Код услуги 2", "Код услуги 3" и "Код сотрудника").
--	--

Вариант 5. БД МВД.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности, Код звания)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Звания (Код звания, Наименование, Надбавка, Обязанности, Требования)[5 записей]. Виды преступлений (Код вида преступления, Наименование, Статья, Наказание, Срок)[5 записей]. Преступники (Номер дела, ФИО, Дата рождения, Пол, Адрес, Код вида преступления, Код пострадавшего, Состояние, Код сотрудника)[10 записей]. Пострадавшие (Код пострадавшего, ФИО, Дата рождения, Пол, Адрес)[5 записей].
Запросы:	Отдел кадров (Связывает таблицы "Сотрудники", "Должности" и "Звания" по полям "Код должности" и "Код звания"). Преступники (Связывает таблицы "Преступники", "Виды преступлений", "Пострадавшие" и "Сотрудники" по полям "Код вида преступления", "Код пострадавшего" и "Код сотрудника").

Вариант 6. БД Аэропорта.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Самолёты (Код самолёта, Марка, Вместимость, Грузоподъемность, Код типа, Технические характеристики, Дата выпуска, Налётано часов, Дата последнего ремонта, Код сотрудника)[5 записей]. Типы самолётов (Код типа, Наименование, Назначение, Ограничения). Экипажи (Код экипажа, Налётано часов, Код сотрудника 1, Код сотрудника 2, Код сотрудника 3)[5 записей]. Рейсы (Код рейса, Дата, Время, Откуда, Куда, Код экипажа, Код самолёта, Время полёта)[5 записей]. Билеты (ФИО пассажира, Паспортные данные, Место, Код рейса, Цена)
Запросы:	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности"). Самолёты (Связывает таблицы "Самолёты", "Типы самолётов" и "Сотрудники" по полям "Код типа" и "Код сотрудника") Экипажи (Связывает таблицы "Экипажи" и "Сотрудники" по полям "Код сотрудника" "Код сотрудника 1", "Код сотрудника 2" и "Код сотрудника 3") Рейсы (Связывает таблицы "Рейсы", "Самолёты" и "Экипажи" по полям "Код экипажа" и "Код самолёта") Билеты (Связывает таблицы "Билеты" и "Рейсы" по полю "Код рейса")

Вариант 7 БД Видео проката.

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ Сертификат: 12000002A633E3D113AD425FB50002000002A6 Владелец: Шебзухова Татьяна Александровна Действителен: с 20.08.2021 по 20.08.2022		Таблицы: Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей].
---	--	--

	Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Жанры (Код жанра, Наименование жанра, Описание)[5 записей]. Кассеты (Код кассеты, Наименование фильма, Год создание, Производитель, Страна, Главный актёр, Дата записи, Код жанра, Цена)[10 записей]. Клиенты (ФИО, Адрес, Телефон, Паспортные данные, Дата взятия, Дата возврата, Отметка об оплате, Отметка о возврате, Код кассеты 1, Код кассеты 2, Код кассеты 3, Код сотрудника)[10 записей].
Запросы:	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности"). Кассеты (Связывает таблицы "Кассеты" и "Жанры" по полю "Код жанра"). Кассеты на руках (Связывает таблицы "Клиенты", "Кассеты" и "Сотрудники" по полям "Код кассеты", "Код кассеты 1", "Код кассеты 2", "Код кассеты 3" и "Код сотрудника").

Вариант 8. БД Библиотеки.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Издательства (Код издательства, Наименование, Город, Адрес)[5 записей]. Жанры (Код жанра, Наименование, Описание) [5 записей]. Книги (Код книги, Наименование, Автор, Код издательства, Год издания, Код жанра) [10 записей]. Читатели (Код читателя, ФИО, Дата рождения, Пол, Адрес, Телефон, Паспортные данные) [10 записей]. Выданные книги (Код книги, Код читателя, Дата выдачи, Дата возврата, Отметка о возврате, Код сотрудника) [10 записей].
Запросы:	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности"). Каталог (Связывает таблицы "Книги", "Издательства" и "Жанры" по полям "Код издательства" и "Код жанра"). Книги на руках (Связывает таблицы "Выданные книги", "Книги", "Читатели" и "Сотрудники" по полям "Код книги", "Код читателя" и "Код сотрудника")

Вариант 9. БД Радиостанции.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Исполнители (Код исполнителя, Наименование, Описание)[5 записей]. Жанры (Код жанра, Наименование, Описание)[5 записей]. Записи (Код записи, Наименование, Код исполнителя, Альбом, Год, Код жанра, Дата записи, Длительность, Рейтинг)[10 записей]. График работы (Дата, Код сотрудника, Время 1, Код записи 1, Время 2, Код записи 2, Код записи 3)[10 записей].
-----------------	---

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ Сертификат: 12000002A633E3D113AD425FB50002000002A6 Владелец: Шебзухова Татьяна Александровна Действителен: с 20.08.2021 по 20.08.2022		Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности").
---	--	---

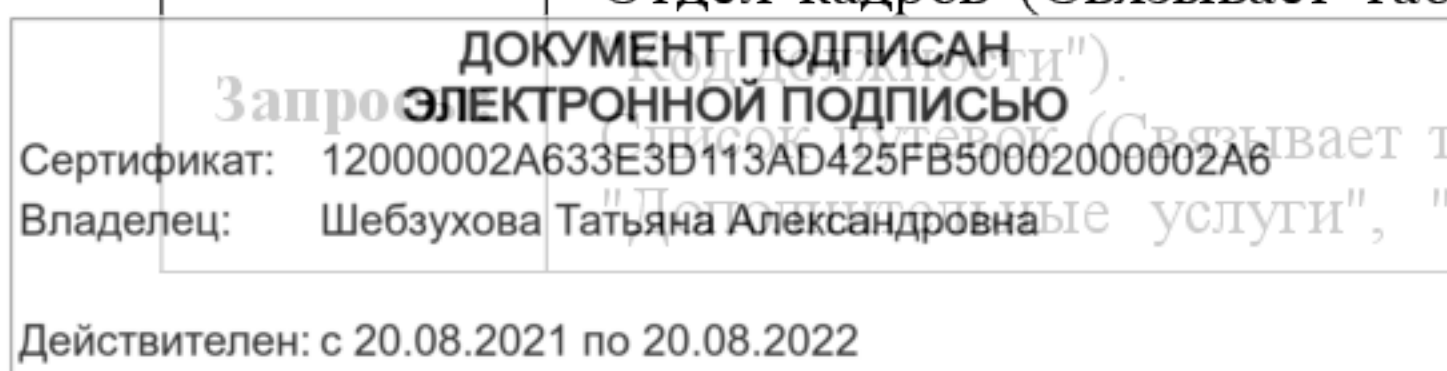
	Музыкальный архив (Связывает таблицы "Записи", "Исполнители" и "Жанры" по полям "Код исполнителя" и "Код жанра"). Сетка вещания (Связывает таблицы "График работы", "Сотрудники" и "Записи" по полям "Код сотрудника", "Код записи", "Код записи 1", "Код записи 2" и "Код записи 3").
--	---

Вариант 10. БД Таксопарка.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Марки (Код марки, Наименование, Технические характеристики, Стоимость, Специфика)[5 записей]. Тарифы (Код тарифа, Наименование, Описание, Стоимость)[5 записей]. Дополнительные услуги (Код услуги, Наименование, Описание услуги, Стоимость)[5 записей]. Автомобили (Код автомобиля, Код марки, Регистрационный номер, Номер кузова, Номер двигателя, Год выпуска, Пробег, Код сотрудника-шофёра, Дата последнего ТО, Код сотрудника-механика, Специальные отметки)[10 записей]. Вызовы (Дата, Время, Телефон, Откуда, Куда, Код тарифа, Код услуги, Код автомобиля, Код сотрудника-оператора)[10 записей].
Запросы:	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности"). Автопарк (Связывает таблицы "Автомобили", "Марки" и "Сотрудники" по полю "Код марки" и "Код сотрудника"). Список вызовов (Связывает таблицы "Вызовы", "Тарифы", "Услуги", "Автомобили" и "Сотрудники" по полю "Код тарифа", "Код услуги", "Код автомобиля" и "Код сотрудника-диспетчера").

Вариант 11. БД Туристического агентства.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Виды отдыха (Код вида, Наименование, Описание, Ограничения)[5 записей]. Отели (Код отеля, Наименование, Страна, Город, Адрес, Телефон, Количество звёзд, Контактное лицо)[10 записей]. Дополнительные услуги (Код услуги, Наименование, Описание, Цена) [5 записей]. Клиенты (Код клиента, ФИО, Дата рождения, Пол, Адрес, Телефон, Паспортные данные)[5 записей]. Путёвки (Дата начала, Дата окончания, Продолжительность, Код отеля, Код вида, Код услуги 1, Код услуги 2, Код услуги 3, Код клиента, Код сотрудника, Отметка о бронировании, Отметка об оплате)[10 записей].
	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности").
Запросы:	Связывает таблицы "Путёвки", "Отели", "Виды отдыха", "Дополнительные услуги", "Клиенты" и "Сотрудники" по полям "Код услуги 1", "Код услуги 2", "Код услуги 3", "Код клиента", "Код сотрудника", "Отметка о бронировании", "Отметка об оплате".



	отеля", "Код вида", "Код услуги", "Код услуги 1", "Код услуги 2", "Код услуги 3", "Код клиента" и "Код сотрудника").
--	--

Вариант 12. БД Страховой компании.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Риски (Код риска, Наименование, Описание, Средняя вероятность)[5 записей]. Виды полисов (Код вида полиса, Наименование, Описание, Условия, Код риска 1, Код риска 2, Код риска 3)[5 записей]. Группы клиентов (Код группы, Наименование, Описание)[5 записей]. Клиенты (Код клиента, ФИО, Дата рождения, Пол, Адрес, Телефон, Паспортные данные, Код группы)[10 записей]. Полисы (Номер полиса, Дата начала, Дата окончания, Стоимость, Сумма выплаты, Код вида полиса, Отметка о выплате, Отметка об окончании, Код клиента, Код сотрудника)[10 записей].
Запросы:	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности"). Риски полисов (Связывает таблицы "Виды полисов" и "Риски" по полям "Код риска", "Код риска 1", "Код риска 2", "Код риска 3"). Список клиентов (Связывает таблицы "Клиенты" и "Группы клиентов" по полю "Код группы"). Список полисов (Связывает таблицы "Полисы", "Виды полисов", "Клиенты" и "Сотрудники" по полям "Код вида полиса", "Код клиента" и "Код сотрудника").

Вариант 13. БД Брачного агентства.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Знаки зодиака (Код знака, Наименование, Описание)[5 записей]. Отношения (Код отношения, Наименование, Описание)[5 записей]. Национальности (Код национальности, Наименование, Замечания)[5 записей]. Дополнительные услуги (Код услуги, Наименование, Описание, Цена)[5 записей]. Клиенты (Код клиента, ФИО, Пол, Дата рождения, Возраст, Рост, Вес, Количество детей, Семейное положение, Вредные привычки, Хобби, Описание, Код знака, Код отношения, Код национальности, Адрес, Телефон, Паспортные данные, Информация о партнере)[10 записей]. Услуги (Код клиента, Дата, Код услуги 1, Код услуги 2, Код услуги 3, Стоимость, Код сотрудника)[10 записей].
	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности").

Запрос: Список клиентов (Связывает таблицы "Клиенты", "Знаки зодиака", "Отношения" по полям "Код знака", "Код отношения" и "Код национальности").

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

	Список услуг (Связывает таблицы "Услуги", "Клиенты", "Дополнительные услуги" и "Сотрудники" по полям "Код клиента", "Код услуги", "Код услуги 1", "Код услуги 2", "Код услуги 3" и "Код сотрудника").
--	---

Вариант 14. БД Сервис центра.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Запчасти (Код запчасти, Наименование, Функции, Цена)[5 записей]. Ремонтируемые модели (Код модели, Наименование, Тип, Производитель, Технические характеристики, Особенности)[5 записей]. Виды неисправностей (Код вида, Код модели, Описание, Симптомы, Методы ремонта, Код запчасти 1, Код запчасти 2, Код запчасти 3, Цена работы)[5 записей]. Обслуживаемые магазины (Код магазина, Наименование, Адрес, Телефон)[5 записей]. Заказы (Дата заказа, Дата возврата, ФИО заказчика, Серийный номер, Код вида неисправности, Код магазина, Отметка о гарантии, Срок гарантии ремонта, Цена, Код сотрудника)[10 записей].
Запросы:	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности"). Список неисправностей (Связывает таблицы "Виды неисправностей", "Ремонтируемые модели" и "Запчасти" по полям "Код модели", "Код запчасти", "Код запчасти 1", "Код запчасти 2", "Код запчасти 3"). Список заказов (Связывает таблицы "Заказы", "Виды неисправностей", "Обслуживаемые магазины" и "Сотрудники" по полям "Код вида неисправности", "Код магазина" и "Код сотрудника").

Вариант 15. БД Школы.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Ученики (ФИО, Дата рождения, Пол, Адрес, ФИО отца, ФИО матери, Код класса, Дополнительная информация) [10 записей]. Классы (Код класса, Код сотрудника-классного руководителя, Код вида, Количество учеников, Буква, Год обучения, Год создания)[5 записей]. Виды классов (Код вида, Наименование, Описание)[5 записей]. Предметы (Код предмета, Наименование, Описание, Код сотрудника-учителя)[10 записей]. Расписание (Дата, День недели, Код класса, Код предмета, Время начала, Время окончания)[10 записей].
Запросы:	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности"). Список учеников (Связывает таблицы "Ученики" и "Классы" по полю "Код класса"). Список классов (Связывает таблицы "Классы", "Виды классов" и "Ученики" по полям "Код класса", "Код вида" и "Код сотрудника"). Список предметов (Связывает таблицы "Предметы" и "Сотрудники" по полю "Код сотрудника").

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

	Расписание занятий (Связывает таблицы "Расписание", "Классы" и "Предметы" по полям "Код класса" и "Код предмета").
--	--

Вариант 16. БД Транспортной компании.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Виды автомобилей (Код вида автомобиля, Наименование, Описание)[5 записей]. Марки автомобилей (Код марки, Наименование, Технические характеристики, Описание) [5 записей]. Виды грузов (Код вида груза, Наименование, Код вида автомобиля для транспортировки, Описание)[5 записей]. Грузы (Код груза, Наименование, Код вида груза, Срок годности, Особенности)[5 записей]. Автомобили (Код автомобиля, Код марки, Код вида автомобиля, Регистрационный номер, Номер кузова, номер двигателя, Год выпуска, Код сотрудника-водителя, Дата последнего ТО, Код сотрудника-механика)[5 записей]. Рейсы (Код автомобиля, Заказчик, Откуда, Куда, Дата отправления, Дата прибытия, Код груза, Цена, Отметка об оплате, Отметка о возвращении, Код сотрудника)[10 записей].
Запросы:	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности"). Транспортировка (Связывает таблицы "Виды грузов" и "Виды автомобилей" по полю "Код вида автомобиля"). Перевозимые грузы (Связывает таблицы "Грузы" и "Виды грузов" по полю "Код вида груза"). Автопарк (Связывает таблицы "Автомобили", "Марки автомобилей", "Виды автомобилей" и "Сотрудники" по полям "Код марки", "Код вида автомобиля" и "Код сотрудника"). Заказы (Связывает таблицы "Рейсы", "Автомобили", "Грузы" и "Сотрудники" по полям "Код автомобиля", "Код груза" и "Код сотрудника").

Вариант 17. БД Проката автомобилей.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Марки автомобилей (Код марки, Наименование, Технические характеристики, Описание) [5 записей]. Дополнительные услуги (Код услуги, Наименование, Описание, Цена)[5 записей]. Автомобили (Код автомобиля, Код марки, Регистрационный номер, Номер кузова, Номер двигателя, Год выпуска, Пробег, Цена автомобиля, Цена дня проката, Дата последнего ТО, Код сотрудника-механика, Специальные отметки, Отметка о возврате)[10 записей]. Клиенты (Код клиента, ФИО, Пол, Дата рождения, Адрес, Телефон, Паспортные данные)[5 записей].
-----------------	---

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

	Прокат (Дата выдачи, Срок проката, Дата возврата, Код автомобиля, Код клиента, Код услуги 1, Код услуги 2, Код услуги 3, Цена проката, Отметка об оплате, Код сотрудника)[10 записей].
--	--

Вариант 18. БД Оптового склада.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Товары (Код товара, Код типа, Производитель, Наименование, Условия хранения, Упаковка, Срок годности) [10 записей]. Типы товаров (Код типа, Наименование, Описание, Особенности) [5 записей]. Поставщики (Код поставщика, Наименование, Адрес, Телефон, Код поставляемого товара 1, Код поставляемого товара 2, Код поставляемого товара 3) [5 записей]. Заказчики (Код заказчика, Наименование, Адрес, Телефон, Код потребляемого товара 1, Код потребляемого товара 2, Код потребляемого товара 3) [5 записей]. Склад (Дата поступления, Дата заказа, Дата отправки, Код товара, Код поставщика, Код заказчика, Способ доставки, Объем, Цена, Код сотрудника) [10 записей].
Запросы:	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности"). Список товаров (Связывает таблицы "Товары" и "Типы товаров" по полю "Код типа"). Список поставщиков (Связывает таблицы "Поставщики" и "Товары" по полям "Код товара", "Код поставляемого товара 1", "Код поставляемого товара 2" и "Код поставляемого товара 3"). Список заказчиков (Связывает таблицы "Заказчики" и "Товары" по полям "Код товара", "Код потребляемого товара 1", "Код потребляемого товара 2" и "Код потребляемого товара 3"). Заказы (Связывает таблицы "Склад", "Товары", "Поставщики", "Заказчики" и "Сотрудники" по полям "Код товара", "Код поставщика", "Код заказчика" и "Код сотрудника").

Вариант 19. БД Строительной компании.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Виды работ (Код вида, Наименование, Описание, Цена работы, Код материала 1, Код материала 2, Код материала 3)[5 записей]. Материалы (Код материала, Наименование, Упаковка, Описание, Цена) [5 записей]. Бригады (Код бригады, Код сотрудника 1, Код сотрудника 2, Код сотрудника 3) [5 записей]. Заказчики (Код заказчика, ФИО, Адрес, Телефон, Паспортные данные)[5 записей].
-----------------	---

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ Сертификат: 12000002A633E3D113AD425FB50002000002A6 Владелец: Шебзухова Татьяна Александровна Действителен: с 20.08.2021 по 20.08.2022	
---	--

Заказы (Код заказчика, Код вида работ, Код бригады, Стоимость, Дата начала, Дата окончания, Отметка о завершении, Об оплате, Код сотрудника) [10 записей].

Запросы:	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности"). Список работ (Связывает таблицы "Виды работ" и "Материалы" по полям "Код материала", "Код материала 1", "Код материала 2" и "Код материала 3"). Список бригад (Связывает таблицы "Бригады" и "Сотрудники" по полям "Код сотрудника", "Код сотрудника 1", "Код сотрудника 2" и "Код сотрудника 3"). Список заказов (Связывает таблицы "Заказы", "Виды работ", "Бригады" и "Сотрудники" по полям "Код вида", "Код бригады" и "Код сотрудника").
-----------------	--

Вариант 20. БД Риэлтерской фирмы.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Виды услуг (Код вида услуги, Наименование, Описание, Цена)[5 записей]. Виды квартир (Код вида, Наименование, Описание)[5 записей]. Продавцы (Код продавца, ФИО, Пол, Дата рождения, Адрес проживания, Телефон, Паспортные данные, Код вида квартиры, Адрес квартиры, Количество комнат, Площадь, Отметка о раздельном санузле, Отметка о наличии телефона, Цена, Дополнительная информация)[10 записей]. Покупатели (Код покупателя, ФИО, Пол, Дата рождения, Адрес проживания, Телефон, Паспортные данные, Код вида квартиры, Количество комнат, Площадь, Отметка о раздельном санузле, Отметка о наличии телефона, Цена, Дополнительные пожелания)[10 записей]. Договоры (Дата заключения, Код продавца, Код покупателя, Сумма сделки, Стоимость услуг, Код вида услуги, Код сотрудника)[10 записей].
Запросы:	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности"). Продажа (Связывает таблицы "Продавцы" и "Виды квартир" по полю "Код вида квартиры"). Покупка (Связывает таблицы "Покупатели" и "Виды квартир" по полю "Код вида квартиры"). Заключённые договора (Связывает таблицы "Договоры", "Продавцы", "Покупатели", "Услуги" и "Сотрудники" по полям "Код продавца", "Код покупателя", "Код услуги" и "Код сотрудника").

Вариант 21. БД Рекламного агентства.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Виды рекламы (Код вида, Наименование, Описание) [5 записей]. Дополнительные услуги (Код услуги, Наименование, Описание, Стоимость) [5 записей]. Места расположения (Код места, Наименование, Расположение, Код вида, Описание, Стоимость) [10 записей]. Заказы (Код заказа, Дата начала, Дата окончания, Код заказчика, Код места, Код услуги 1, Код услуги 2, Код услуги 3, Стоимость, Отметка об оплате, Код сотрудника) [10 записей].
-----------------	--

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
 Сертификат: 12000002A633E3D113AD425FB50002000002A6
 Владелец: Шебзухова Татьяна Александровна
 Действителен: с 20.08.2021 по 20.08.2022

	Дата регистрации, Цвет, Технический осмотр, Дата технического осмотра, Описание, Код сотрудника)[15 записей]. Автомобили в угоне (Дата угона, Дата обращения, Код автомобиля, Код водителя, Обстоятельства угона, Отметка об нахождении, Дата нахождения, Код сотрудника)[5 записей].
Запросы:	Отдел кадров (Связывает таблицы "Сотрудники", "Должности" и "Звания" по полям "Код должности" и "Код звания"). Список автомобилей (Связывает таблицы "Автомобили", "Марки автомобилей", "Водители" и "Сотрудники" по полям "Код марки", "Код водителя" и "Код сотрудника"). Список угонов (Связывает таблицы "Автомобили в угоне", "Автомобили" и "Водители" по полям "Код автомобиля" и "Код водителя").

Вариант 24. БД Кинотеатра.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Жанры (Код жанра, Наименование, Описание)[5 записей]. Фильмы (Код фильма, Наименование, Код жанра, Длительность, Фирма производитель, Страна производитель, Актёры, Возрастные ограничения, Описание)[10 записей]. Репертуар (Код сеанса, Дата, Время начала, Время окончания, Цена билета)[10 записей]. Места (Код сеанса, Номер места, Занятость, Код сотрудника)[15 записей].
Запросы:	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности"). Список фильмов (Связывает таблицы "Фильмы" и "Жанры" по полю "Код жанра"). Билеты (Связывает таблицы "Места", "Репертуар" и "Сотрудники" по полям "Код сеанса" и "Код сотрудника").

Вариант 25 БД Автосалона.

Таблицы:	Сотрудники (Код сотрудника, ФИО, Возраст, Пол, Адрес, Телефон, Паспортные данные, Код должности)[10 записей]. Должности (Код должности, Наименование должности, Оклад, Обязанности, Требования)[5 записей]. Производители (Код производителя, Наименование, Страна, Адрес, Описание, Код сотрудника)[5 записей]. Дополнительное оборудование (Код оборудования, Наименование, Характеристики, Цена)[5 записей]. Тип кузова (Код типа кузова, Название, Описание)[5 записей]. Автомобили (Код автомобиля, Марка, Код производителя, Код типа кузова, Дата производства, Цвет, Номер кузова, Номер двигателя, Характеристики, Код оборудования 1, Код оборудования 2, Код оборудования 3, Цена, Код сотрудника)[10 записей]. Заказчики (ФИО, Адрес, Телефон, Паспортные данные, Код автомобиля, Дата продажи, Отметка о выполнении, Отметка об оплате, Код сотрудника)[10 записей].
-----------------	--

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

Запросы:	Отдел кадров (Связывает таблицы "Сотрудники" и "Должности" по полю "Код должности"). Каталог автомобилей (Связывает таблицы "Автомобили", "Производители", "Тип кузова", "Дополнительное оборудование" и "Сотрудники" по полям "Код производителя", "Код типа кузова", "Код оборудования", "Код оборудования 1", "Код оборудования 2", "Код оборудования 3" и "Код сотрудника"). Список заказов (Связывает таблицы "Заказчики", "Автомобили" и "Сотрудники" по полям "Код автомобиля" и "Код сотрудника").
-----------------	---

3. СТРУКТУРА РАБОТЫ

По своей структуре курсовая работа по дисциплине «Управление данными» состоит из введения, теоретической и практической частей, заключения, приложений и списка использованной литературы.

Примерный план курсового проекта:

ВВЕДЕНИЕ

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

- 1.1 Анализ тенденций развития отдельных направлений управления данными по вариантам).

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

- 2.1 Обследование предметной области.
- 2.2 Техническое задание на разработку базы данных
- 2.3 Концептуальное проектирование.
 - 2.3.1 Перечень сущностей (обосновать список).
 - 2.3.2 Перечень атрибутов.
- 2.4 Инфологическое проектирование БД.
 - 2.4.1 Модель "сущность-связь".
 - 2.4.2 Классификация связей.
- 2.5. Реляционная модель БД.
 - 2.5.1 Функциональные зависимости между атрибутами.
 - 2.5.2 Выбор ключей.
 - 2.5.3 Нормализация отношений.
- 2.6 Даталогическое проектирование БД.
 - 2.6.1 Состав таблиц БД.
 - 2.6.2 Средства поддержания целостности.
- 2.7 Запросы к БД.
- 2.8 Разработка механизмов защиты данных от несанкционированного доступа.
- 2.9 Требования к техническому обеспечению.
- 2.10 Инструкция по использованию БД.
 - 2.10.1 Вызов программы.

10.2 Экраны
10.3 Отчеты

**ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ**
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

ПРИЛОЖЕНИЯ

3.1 Введение к курсовому проекту

Во введении к курсовому проекту обосновывается актуальность темы, формулируется цель, а также приводятся краткие сведения о содержании пояснительной записки с разбивкой по разделам. Во введении необходимо отразить следующую информацию: наименование предприятия (истинное или вымышленное), для которого разрабатывается база данных (БД), наименование предметной области, назначение разработки БД, место разрабатываемой БД в общей системе управления предприятием, требования заказчика к разрабатываемой БД, чья точка зрения используется при проектировании, перечисление пользователей БД, права пользователей, описание (перечисление) общетехнических и общесистемных программных средств. Примерный объем введения – не более 3 страниц машинописного текста.

3.2 Теоретическая часть курсового проекта

Теоретическая часть курсового проекта по дисциплине «Управление данными» содержит аналитический обзор тенденций развития систем управления данными, составленный на основе анализа научноисследовательской литературы и ресурсов Internet.

3.3 Практическая часть курсового проекта

В практическую часть курсового проекта по дисциплине «Управление данными» включены девять подразделов:

- обследование предметной области;
- концептуальное проектирование – выделение сущностей и атрибутов;
- инфологическое проектирование БД;
- разработка реляционной модель БД;
- даталогическое проектирование БД;
- запросы к БД;
- разработка механизмов защиты данных от несанкционированного доступа;
- требования к техническому обеспечению; - инструкция по использованию БД.

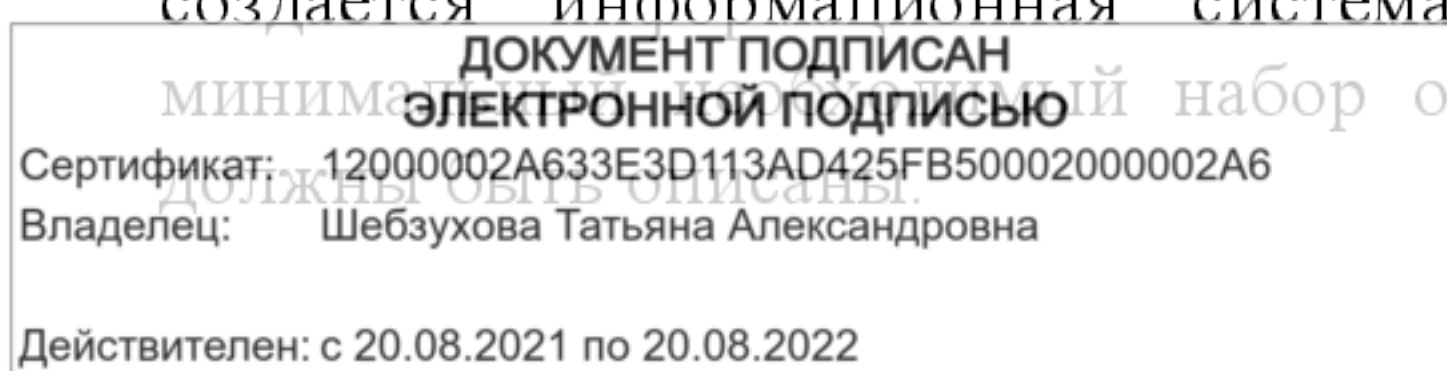
3.3.1 Обследование предметной области

В этом разделе необходимо указать область применения базы данных, то есть провести анализ предметной области, определив ее объекты и связи между ними.

При выборе состава и структуры предметной области возможны два подхода: функциональный и предметный.

Функциональный подход реализует принцип движения «от задач» и применяется, когда определен комплекс задач, для обслуживания которых создается информационная система. В этом случае можно выделить

минимальный набор объектов предметной области, которые



В предметном подходе объекты предметной области определяются с таким расчетом, чтобы их можно было использовать при решении множества разнообразных, заранее не определенных задач.

При необходимости можно разработать словарь терминов предметной области. Здесь же необходимо указать источники информации, которые были использованы при анализе предметной области и информационных потребностей пользователей; перечислить бизнес-процессы, для поддержки которых разрабатывается БД. А также провести анализ входных и выходных документов.

Обследование предметной области в процессе проектирования реляционной базы данных служит основой для разработки технического задания.

Пример. Компьютерная фирма ООО «Сатурн+» оказывает услуги по прокату художественных фильмов на CD- и DVD-носителях. Необходимо спроектировать информационную подсистему «Прокат», содержащую в себе реляционную базу данных для учета выдачи носителей посетителям.

Пользователями подсистемы в данной фирме являются менеджер первого звена, менеджер по работе с клиентами, работник склада и бухгалтер. За аренду фильма начисляется плата, за просрочку возврата – пеня. Новые фильмы фирма приобретает у оптовых поставщиков, заключая с ними долговременные контракты. Менеджер по работе с клиентами и работник склада регулярно готовят отчеты для руководства за определенный период времени об изменении спроса, о поступлении новых лент, о ценах на оптовом рынке, о финансовых показателях.

В функции менеджера по работе с клиентами входит контроль за возвратом носителей, не допуская их выдачу тем, кто просрочил срок аренды.

Функциональный анализ предметной области, лежащей в основе информационной подсистемы «Прокат», позволяет выделить 4 основные функции, которые будет выполнять данная система (рисунок 2.1). Каждый блок функций соответствует одному из пользователей и будет выполняться на его рабочем месте.

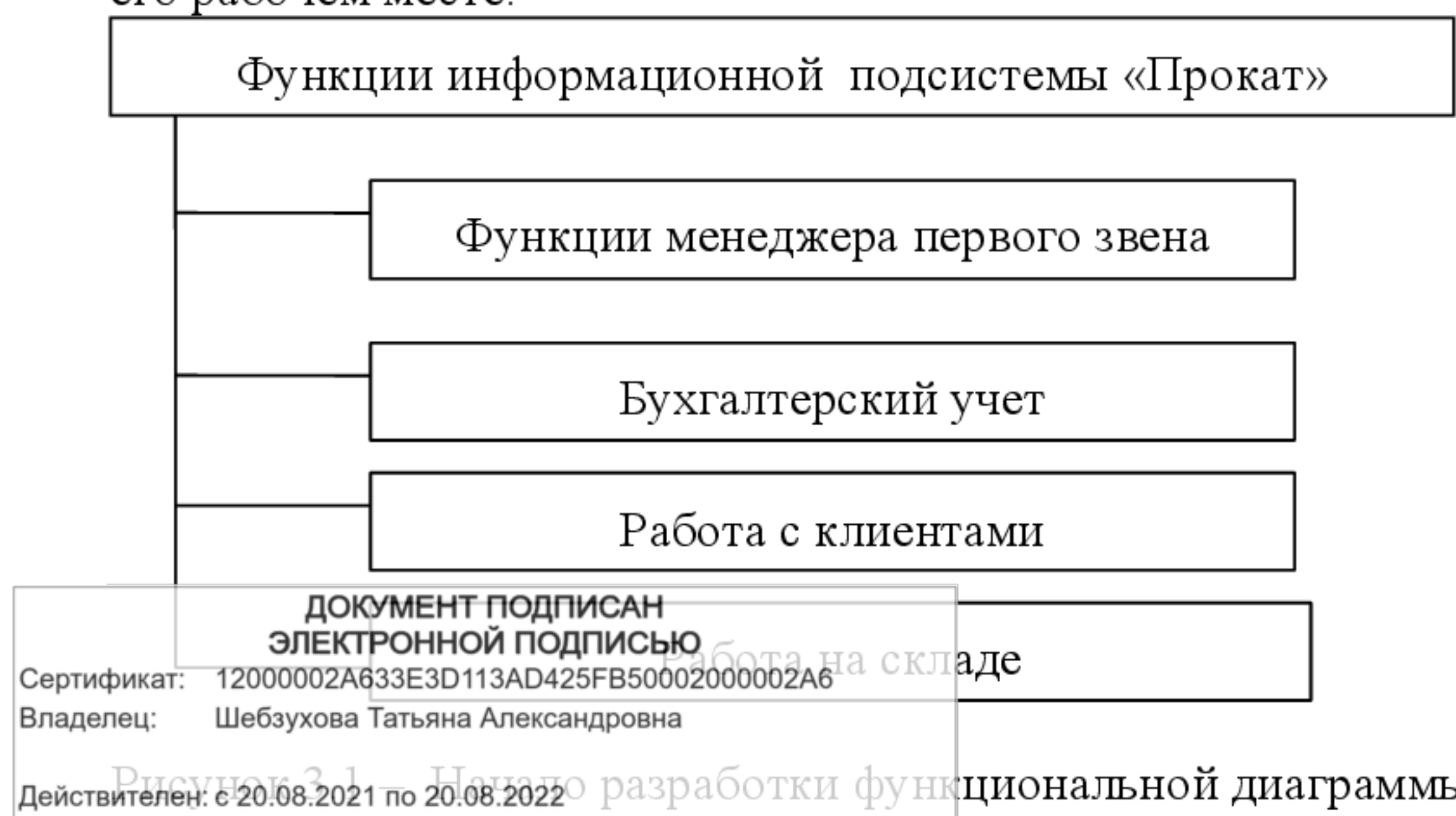


Рисунок 2.1 – Начало разработки функциональной диаграммы

На следующем этапе выполняется декомпозиция информационной подсистемы, то есть проводится анализ каждой функции. При выполнении процесса декомпозиции каждая из функций разбивается на более мелкие и составляется список тех документов, которые обрабатывает пользователь, разобьем функцию на более мелкие. На рисунке 2.2 представлена декомпозиция функции «Работа с клиентами». В нижней половине каждого блока перечислены документы, которые нужны для выполнения данной функции. Родительский блок содержит также должность ответственного за функцию лица.

На основании предложенных функциональных диаграмм можно разработать техническое задание на проектирование базы данных, являющейся частью информационной подсистемы «Прокат».

База данных «Прокат» разрабатывается на основании рекомендации руководителя организации для автоматизации деятельности компании ООО «Сатурн+».

База данных «Прокат» предназначена для учета выдаваемых клиентам товаров, платежей и формирования всей необходимой документации и отчетов. Для ускорения обслуживания, работа может вестись с применением технологий штрихового кодирования и членских карточек. База данных должна иметь развитую систему разделения доступа сотрудников к различным функциям и отчетам.

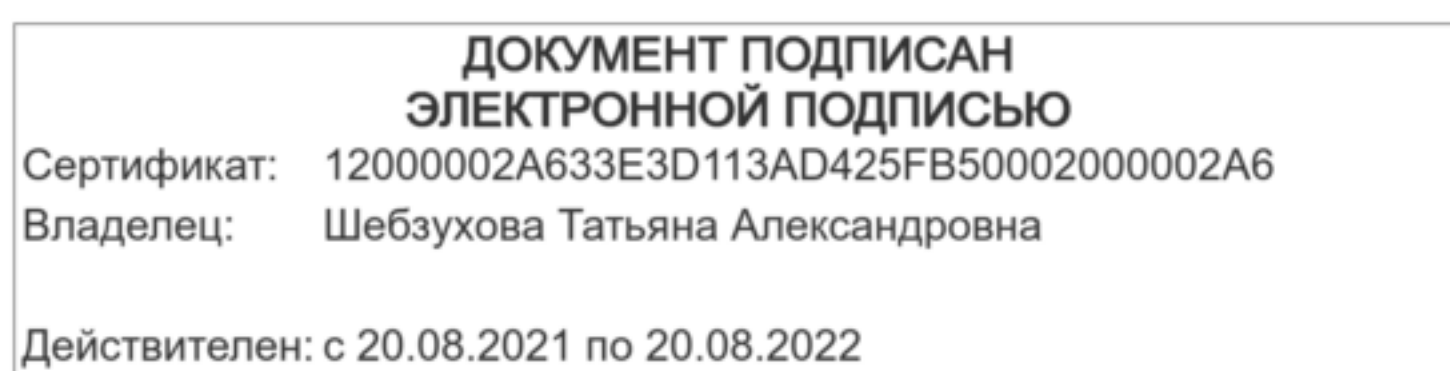




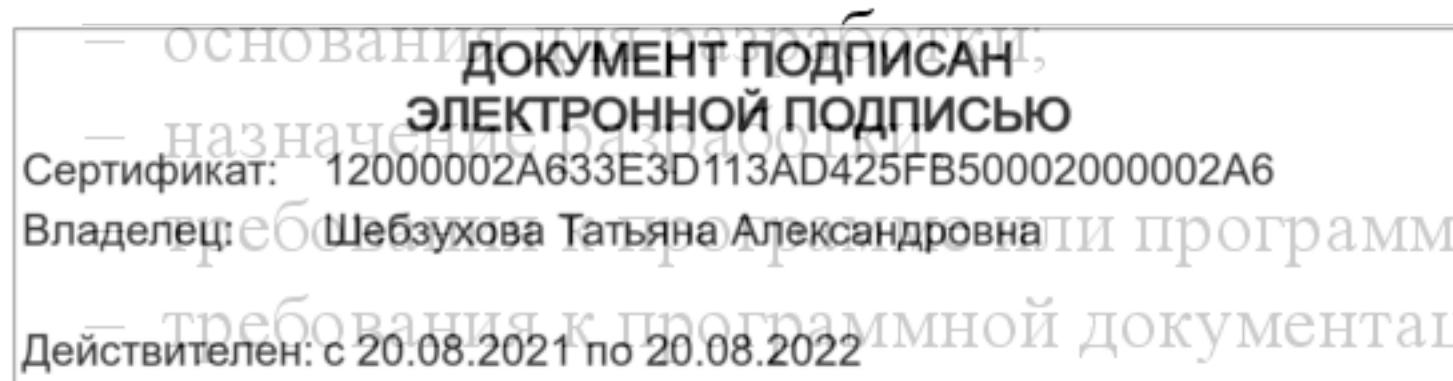
Рисунок 3.2 – Детализация фрагмента диаграммы для функции «Работа с клиентами»

На основании предложенных функциональных диаграмм можно разработать техническое задание на проектирование базы данных, являющейся частью информационной подсистемы «Прокат».

3.3.2 Разработка технического задания

На техническое задание существует стандарт ГОСТ 19.201-78 «Техническое задание. Требования к содержанию и оформлению». В соответствии с этим стандартом техническое задание должно содержать следующие разделы:

– введение;



ному изделию;
и;

- технико-экономические показатели;
- стадии и этапы разработки;
- порядок контроля и приемки.

При необходимости допускается в техническое задание включать приложения.

Пример технического задания. Требуется разработать техническое задание на разработку базы данных «Прокат».

1. ВВЕДЕНИЕ

Настоящее техническое задание распространяется на разработку базы данных «Прокат».

База данных «Прокат» предназначена для учета платежей и выдаваемых клиентам товаров, формирования всей необходимой документации и отчетов. Для ускорения обслуживания работа может вестись с применением технологий штрихового кодирования и членских карточек. База данных должна иметь развитую систему разделения доступа сотрудников к различным функциям и отчетам.

Разрабатываемая программа позволит автоматизировать работу с клиентами компьютерной фирмы ООО «Сатурн+».

2. ОСНОВАНИЕ ДЛЯ РАЗРАБОТКИ

Программа разрабатывается на основе учебного плана кафедры «Информационные системы и технологии», утвержденного на заседании кафедры «__»____201_г., протокол № _____. База данных «Прокат» разрабатывается на основании рекомендации руководителя организации ООО «Сатурн+» для автоматизации деятельности компании.

3. НАЗНАЧЕНИЕ

Основным назначением базы данных является помощь сотрудникам ООО «Сатурн+» при выполнении функций:

- регистрация клиентов;
- ввод товаров;
- автоматическое определение клиентов и товаров;
- быстрый поиск товаров по названию, жанру, ценовой категории, стране, режиссеру...;
- прием оплаты от клиентов, договора с клиентом и т.п.;
- составление различных отчетов.

4. ТРЕБОВАНИЯ К ПРОГРАММЕ ИЛИ ПРОГРАММНОМУ ИЗДЕЛИЮ

4.1. Требования к функциональным характеристикам

4.1.1. Программа должна обеспечивать возможность выполнения следующих функций:

- инициализация системы, которая включает ввод информации о клиентах фирмы, имеющихся носителях и т.п.;
- ввод и корректировка текущей информации о деятельности фирмы;
- электронная подписью информации о деятельности фирмы;
- получение сведений о текущем состоянии деятельности фирмы.

4.1.2. Исходные данные:

ДОКУМЕНТ ПОДПИСАН	
ЭЛЕКТРОННОЙ ПОДПИСЬЮ	
Сертификат:	12000002A633E3D113AD425FB50002000002A6
Владелец:	Шебзухова Татьяна Александровна
4.1.2. Исходные данные:	
Действителен: с 20.08.2021 по 20.08.2022	

- информация о клиентах фирмы;
- каталог имеющихся носителей;
- сотрудники фирмы;
- договора с поставщиками;
- договора с клиентами.

4.1.3. Результаты:

- информация о клиентах фирмы, имеющих задолженности по аренде носителей;
- информация о клиентах фирмы, не имеющих задолженности по аренде носителей;
- информация о поставке новых носителей;
- поиск товаров по названию, жанру, ценовой категории, стране, режиссеру

4.2. Требования к надежности

4.2.1.Предусмотреть контроль вводимой информации.

4.2.2.Предусмотреть блокировку некорректных действий пользователя при работе с системой.

4.2.3.Обеспечить целостность хранимой информации.

4.2.4. Обеспечить разграничение доступа к функциям системы для сотрудников пункта проката.

4.3. Требования к составу и параметрам технических средств

4.3.1.Система должна работать на IBM совместимых персональных компьютерах.

4.3.2.Минимальная конфигурация:

- ☐ тип процессора - Pentium и выше;
- ☐ объем оперативного запоминающего устройства - 32 Мб и более.

4.4. Требования к информационной и программной совместимости

Система должна работать под управлением семейства операционных систем Win 32 (Windows 95, Windows 98, Windows 2000, Windows NT и т. п.).

5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

5.1.Разрабатываемые программные модули должны быть самодокументированы, т. е. тексты программ должны содержать все необходимые комментарии.

5.2.Программная система должна включать справочную информацию о работе и подсказки пользователю.

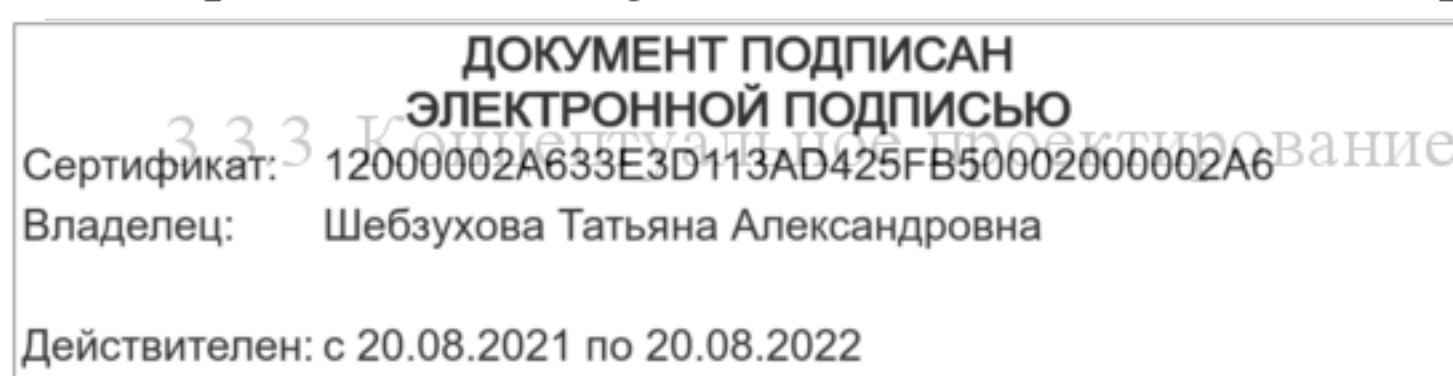
5.3.В состав сопровождающей документации должны входить:

5.3.1.Пояснительная записка, содержащая описание разработки.

5.3.2. Руководство пользователя.

5.3.3.Схема структурная базы данных.

5.3.4. Формы интерфейса пользователя. База данных должна иметь простой и интуитивно-понятный, настраиваемый интерфейс пользователя.



В процессе концептуального проектирования базы данных производится выбор информационных объектов, составляется перечень сущностей и атрибутов.

Для выделения информационных объектов в проектируемой базе данных необходимо соблюдать следующие условия:

1. Выявить документы, использующиеся фирмой в своей деятельности, и их реквизиты (поля).
2. Определить функциональную зависимость между реквизитами для каждого документа в отдельности.
3. Выбрать все зависимые реквизиты и указать для каждого из них все его ключевые (один или несколько), т.е. те, от которых он зависит функционально полно.
4. Сгруппировать реквизиты, зависимые от одинаковых ключевых реквизитов. Полученные группы и будут являться информационным объектом.

Совокупность полей – реквизитов выделенного объекта должна отвечать требованиям нормализации, в том числе:

- информационный объект должен содержать уникальный ключ, который может быть как простым, так и составным;
- все остальные (описательные или дополнительные) реквизиты должны быть независимыми друг от друга;
- все реквизиты, входящие в составной ключ, также должны быть независимыми;
- каждый описательный реквизит должен функционально полно зависеть от ключа, т.е. каждому значению ключа должно соответствовать только одно значение описательного реквизита;
- при составном ключе описательный реквизит должен зависеть целиком от всей совокупности реквизитов, образующих ключ;
- каждый описательный реквизит не должен зависеть от ключа транзитивно, через другой промежуточный реквизит.

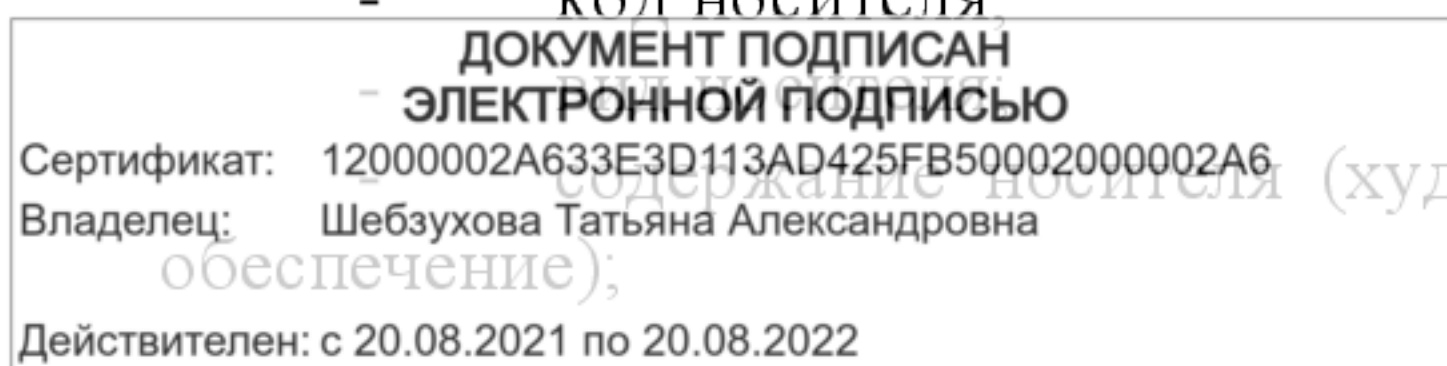
Проведенный анализ предметной области «Прокат» позволяет выделить следующие сущности: «Клиент», «Носитель», «Заказ» и «Сотрудник».

Сущность «Клиент» может быть описана следующим перечнем атрибутов:

- код клиента,
- фамилия, имя, отчество клиента;
- паспортные данные;
- адрес;
- телефон;
- категория

Сущность «Носители» характеризуется атрибутами:

- код носителя;



(художественный фильм или программное

- наименование;
- аннотация;
- обложка;
- стоимость проката.

Сущность «Сотрудник» можно описать с помощью атрибутов:

- код сотрудника;
- фамилия, имя, отчество сотрудника;
- личный пароль.

Центральной в проектируемой базе данных «Прокат» является сущность «Заказ», которая может быть описана с помощью атрибутов:

- код заказа;
- дата заказа;
- код клиента;
- код носителя;
- код сотрудника;
- дата возврата;
- стоимость услуги.

Первичными (уникальными) ключами для сущностей, входящих в базу данных «Прокат» являются:

- сущность «Клиент» – код клиента;
- сущность «Носитель» – код носителя;
- сущность «Сотрудник» – код сотрудника;
- сущность «Заказ» – код заказа.

Для сущности «Заказ» уникальные ключи «код клиента», «код носителя» и «код сотрудника» являются вторичными, и с их помощью осуществляется связь между всеми выбранными сущностями.

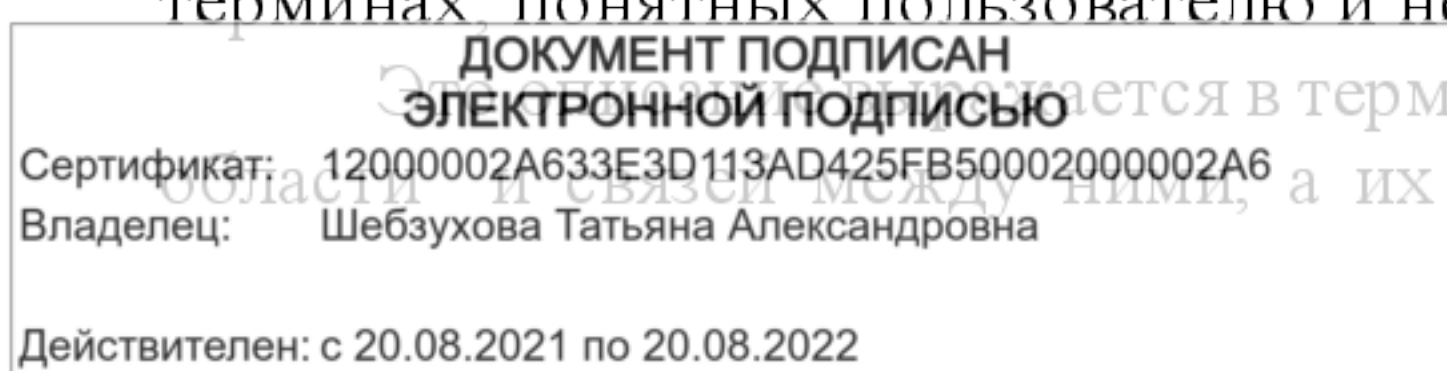
3.3.4 Инфологическое проектирование

В данном подразделе курсового проекта приводится алгоритм разработки ER-диаграммы (диаграммы «сущность – связь») разработанной модели предметной области, рассматривается классификация бинарных связей между сущностями, описывается моделирование связи «один-многим». Рассматривается перечень атрибутов, описывающих, идентифицирующих или моделирующих свойства сущностей.

Основными задачами инфологического проектирования при разработке реляционных баз данных являются описание предметной области и формирование взгляда на предметную область с точки зрения пользователей базы данных.

Инфологическая модель предметной области включает:

- описание ее структуры и динамики;
- характер информационных потребностей пользователей в терминах понятных пользователю и не зависящих от реализации баз данных.



целостности и процессов, которые приводят к переходу предметной области из одного состояния в другое.

Основными подходами к созданию инфологической модели предметной области являются: функциональный подход к проектированию базы данных; предметный подход к проектированию базы данных; проектирование с использованием метода «сущность-связь»

Функциональный подход к проектированию базы данных. Этот метод реализует принцип «от задач» и применяется для обслуживания информационных потребностей тогда, когда известны функции некоторой группы лиц и/или комплекса задач, для которых создаётся рассматриваемая база данных.

Предметный подход к проектированию базы данных. Предметный подход к проектированию базы данных применяется в тех случаях, когда у разработчиков есть чёткое представление о самой предметной области и о том, какую именно информацию они хотели бы хранить в базе данных, а структура запросов не определена или определена не полностью. Тогда основное внимание должно быть уделено всестороннему обследованию предметной области и наиболее адекватному её отображению в базе данных с учётом самого широкого спектра информационных запросов к ней.

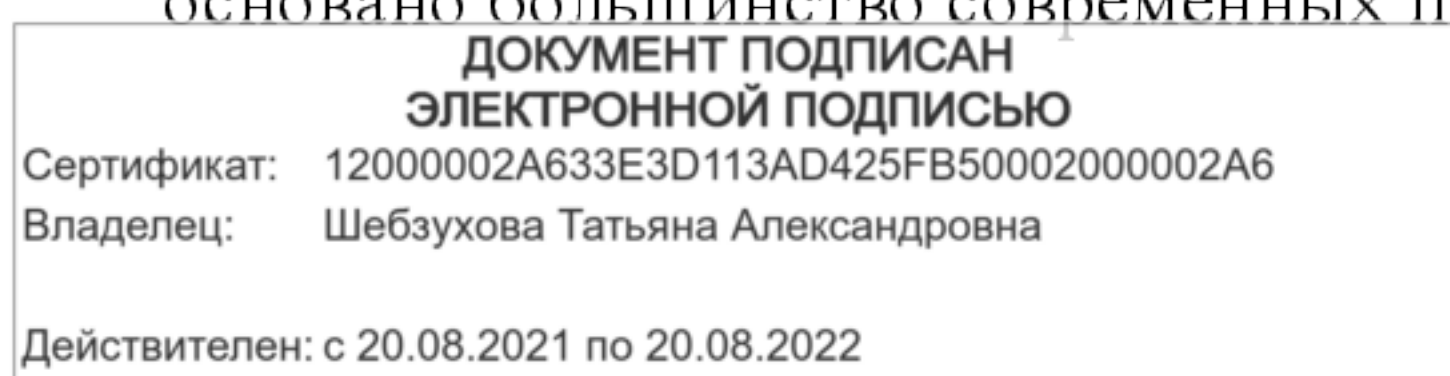
Проектирование с использованием метода «сущность-связь». Метод «сущность-связь» (entity-relation, ER-method) является комбинацией двух предыдущих и обладает достоинствами обоих.

Проектировщик разбивает предметную область на ряд локальных областей, каждая из которых включает в себя информацию, достаточную для обеспечения запросов отдельной группы будущих пользователей или решения отдельной задачи. Каждое локальное представление моделируется отдельно, затем они объединяются.

Диаграмма «сущность-связь» - Entity-Relationship Diagram в англоязычной литературе обозначается термином «ERD». ERD – это графическое представление структуры предметной области в терминах модели ER (сущностей-связей). На диаграммах именованные прямоугольники обозначают множества сущностей, а именованные ромбы – множества связей сущностей. Дуги, связывающие прямоугольники и ромбы могут быть ориентированными.

На рисунке 3.3 показана диаграмма «сущность-связь» (ERD) с тремя сущностями: «Клиент», «Сотрудник» и «Заказ». Для получения диаграммы «сущность-связь» для всей базы данных «Прокат» необходимо добавить еще одну сущность «Носитель».

Впервые модель «сущность-связь» (ERD) была предложена Петером Пин-Шен Ченом в 1976 году. На использовании различных разновидностей ERD основано большинство современных подходов к проектированию баз данных.



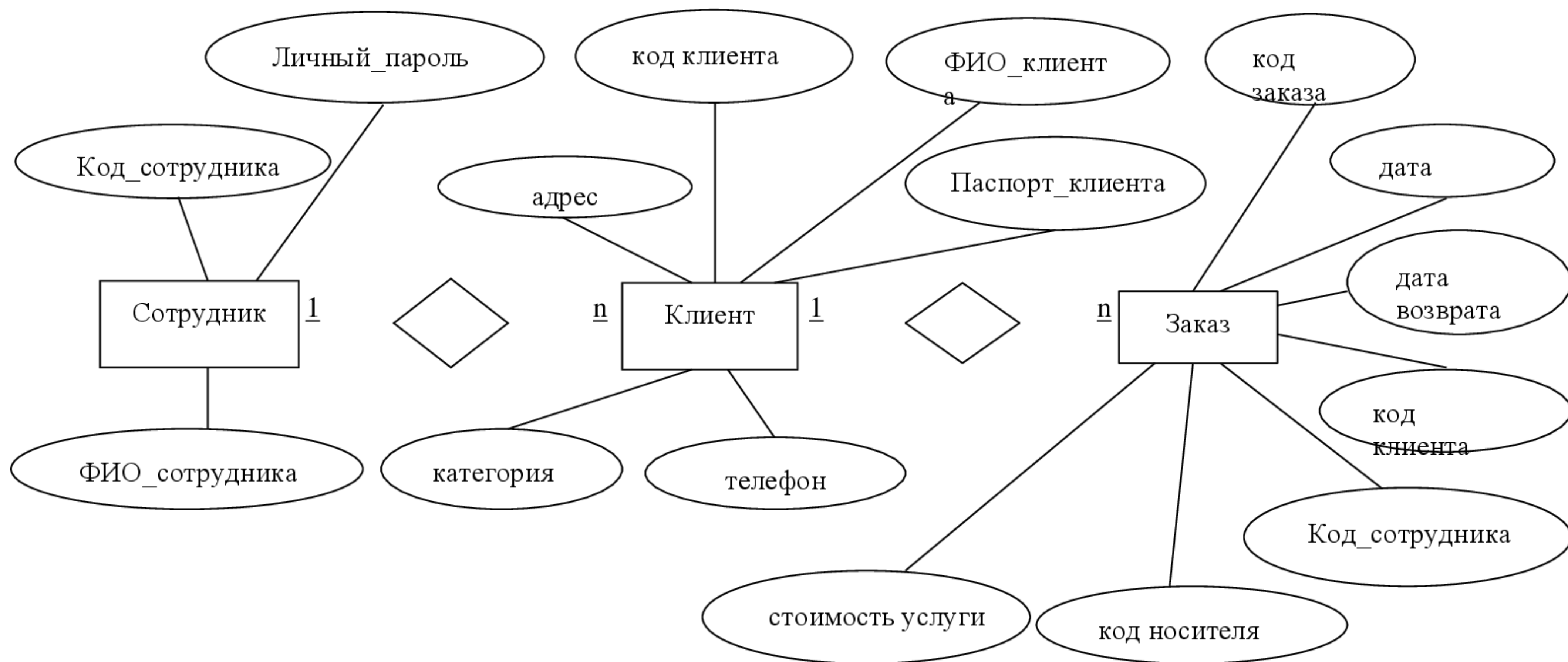


Рисунок 3.3 – ERD с тремя сущностями

Когда в фирму приходит новый клиент, он проходит процедуру регистрации, после чего его заносят в базу данных клиентов проката и вручают ему членскую карточку. На этой членской карточке присутствует уникальный идентификационный номер клиента. С помощью этого номера оператор видео проката быстро идентифицирует клиента и получает полную картину о нем.

Все товары также пронумерованы. Каждому товару присваивается уникальный номер. Этикетка с этим кодом наклеивается на каждый товар, который выставляется на стеллаж. Клиент может взять любой товар, рассмотреть его и, если товар ему понравился, подойти с ним к продавцу. По коду товара оператор очень быстро находит товар в программе, получает от клиента деньги и отдает ему товар. Таким образом, вся процедура обслуживания клиента занимает несколько секунд.

При построении инфологической модели на основе метода «сущность-связь» выделяются три основных типа связей между сущностями:

- «один – к – одному»,
- «один – ко – многим»,
- «многие – ко – многим».

На основании проведенного концептуального проектирования можно сделать вывод о том, что между всеми сущностями в проектируемой базе «Прокат» данных связь между объектами осуществляется по типу «один – ко – многим».

Так один сотрудник может обслуживать много клиентов, но при этом клиент может сделать много заказов.

3.3.5 Реляционная модель БД

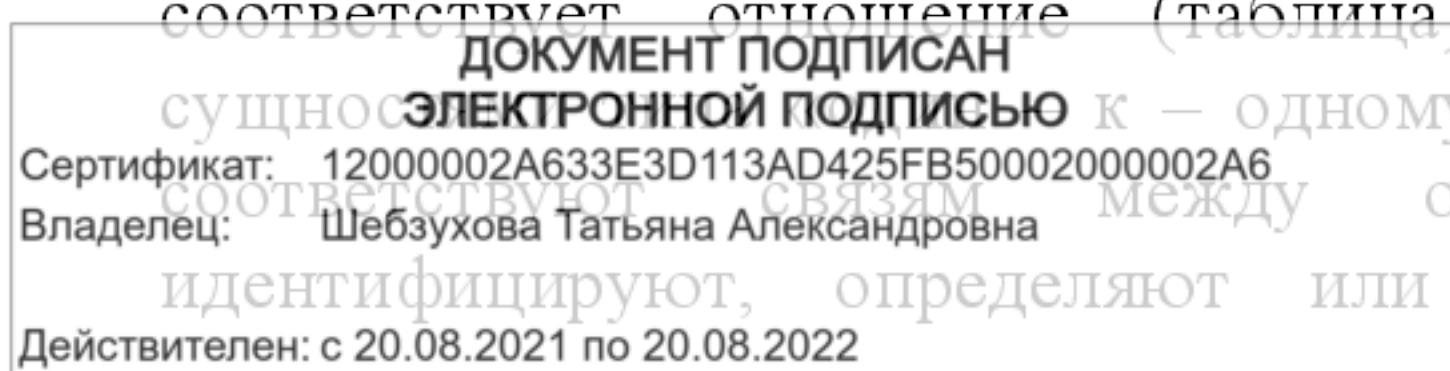
При переходе от концептуальной модели предметной области к схеме БД (например, реляционной) используются три различных подхода, которые подробно описаны в списке использованных источников.

При первом подходе преобразование осуществляется вручную.

Второй подход основан на автоматизированной компиляции концептуальной модели предметной области в схему БД, в результате которой создается реляционная база данных в третьей нормальной форме. Наиболее популярным программным продуктом для автоматизированного проектирования БД является CASE-средство ERwin фирмы Platinum.

Третий подход основан на применении СУБД, использующих семантические модели данных.

В рамках курсового проекта по дисциплине «Управление данными» используется первый подход. При этом сущности модели «сущность-связь» соответствуют отношения (таблица) реляционной БД. Связи между сущностями (1:1) и «один – ко – многим (1:M)» отношениями. Атрибуты, которые идентифицируют, определяют или моделируют сущности в модели



«сущность-связь» соответствуют атрибутам отношений. Сложнее моделируется связь типа «М : N» или «многие-ко-многим». Такой тип связи может быть представлен вспомогательным отношением и двумя функциональными связями 1:M и M:1. Рассмотрим пример, в котором имеются 2 сущности: «Преподаватель» и «Дисциплина». Если преподаватель ведет занятия по нескольким дисциплинам, а занятия по одной и той же дисциплине могут вести разные преподаватели, то связь между сущностями «Преподаватель» и «Дисциплина» соответствует типу «многие-ко-многим». Представление связи типа «М : N» показано на рисунке 2.4. Сущности «Преподаватель» и «Дисциплина» преобразуются в отношения. Связь типа «М : N» преобразуется в дополнительное отношение.

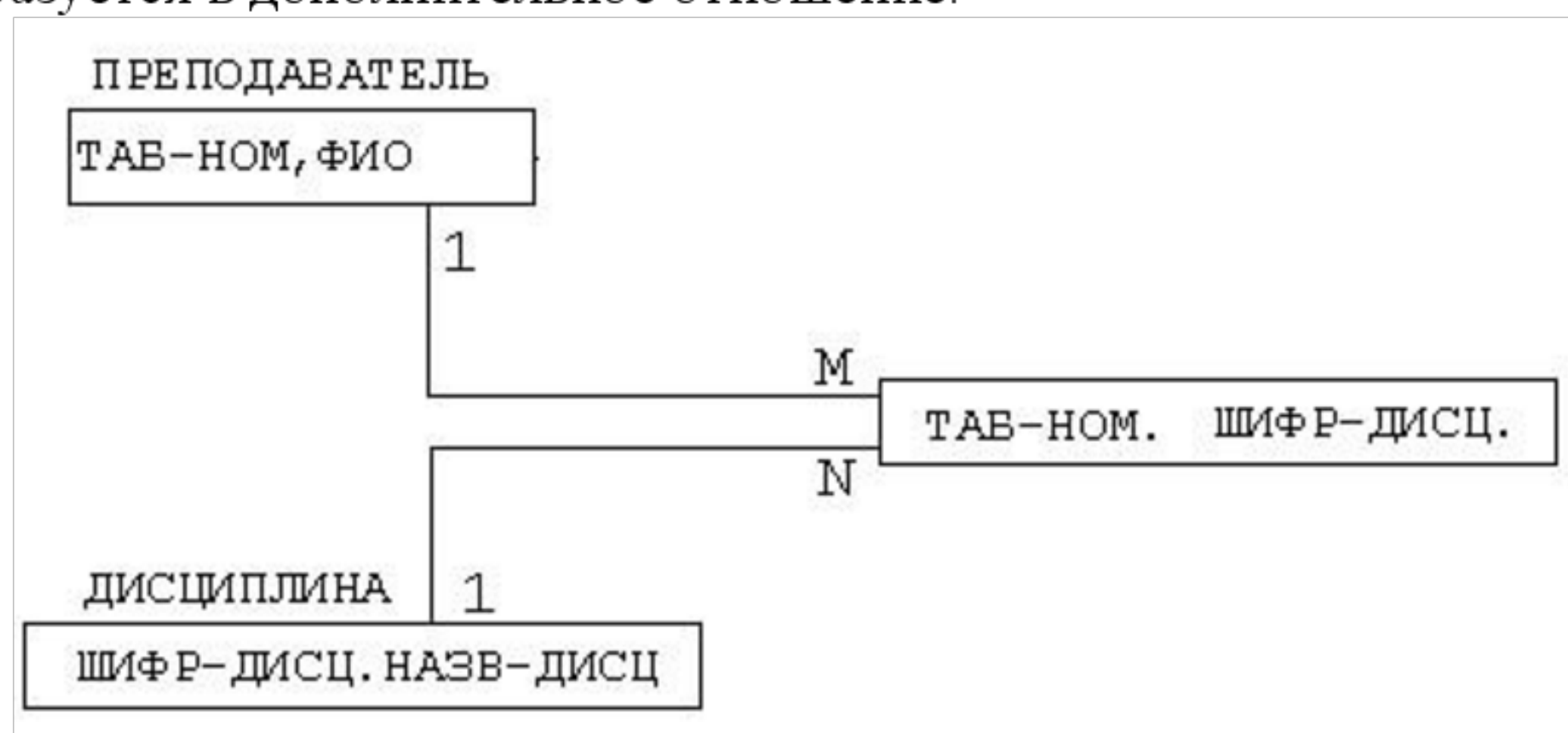


Рисунок 3.4 - Представление связи типа «М : N»

Нормализация баз данных основана на функциональных зависимостях между атрибутами отношений. Функциональную зависимость атрибута Y от X обозначают с помощью записи: $X \rightarrow Y$. На практике используются первая, вторая и третья нормальные формы. В рамках курсового проекта необходимо разработать реляционную базу данных в третьей нормальной форме.

Выполнение требования первой нормальной формы необходимо для того, чтобы работали SQL-запросы. Схема отношения R находится в первой нормальной форме, когда все атрибуты атомарны (неделимы). Обычно имя столбца, состоящее из нескольких слов, записывают одним словом, например, код_заказа, дата_заказа, код_носителя, так как пробел в имени столбца будет нарушать работу запросов. Последующая нормализация, связанная с ограничением функциональных зависимостей в схеме отношения, необходима для уменьшения нежелательных характеристик БД по отношению к включению, удалению, модификации.

Вторая нормальная форма запрещает включать в отношение частичные функциональные зависимости. Если в отношении все ключи простые (т.е. состоят из одного атрибута), то отношение, которое находится в первой

нормальной форме, находится одновременно и во второй нормальной форме. Третья нормальная форма необходима, если первичный ключ отношения является составным (т.е. состоит из нескольких

атрибутов). Схема отношения R находится во второй нормальной форме, если она соответствует первой нормальной форме, и каждый атрибут, не входящий в первичный ключ в этом отношении, полностью зависит от ключа этого отношения R. Атрибут A частично зависит от K, если K есть составной ключ отношения R, $K_1 \sqsubset K$, и одновременно A зависит от ключа, т.е. $K \sqsubset A$ и от части ключа, т.е. $K_1 \sqsubset A$.

Третья нормальная форма запрещает функциональные зависимости непервичных (т.е. не входящих в первичный ключ) атрибутов от других непервичных атрибутов. Схема отношения R находится в третьей нормальной форме, если она соответствует второй нормальной форме и каждый атрибут, не являющийся первичным в этом отношении, нетранзитивно зависит от каждого возможного ключа.

Атрибут A называется транзитивно зависимым от K в отношении R на множестве функциональных зависимостей F, если существует атрибут Y в этом же отношении $Y \sqsubset R$, такой что $K \sqsubset Y$, $Y \sqsubset A$ (при этом K не зависит от Y). Чтобы привести схему отношения к третьей нормальной форме обычно производят декомпозицию отношения R на несколько отношений, чаще на два, таким образом, чтобы ни в одном из отношений не было транзитивных зависимостей. В приведенном примере зависимость $K \sqsubset A$ является транзитивной, поэтому при декомпозиции эти атрибуты (K и A) должны попасть в разные отношения.

Рассмотрим схему отношения «Заказ» из базы данных «Прокат». В схеме отношения Заказ (Код_заказа, дата_заказа, код_клиента, код_носителя, код_сотрудника, дата_возврата, стоимость_услуги) все атрибуты атомарны. Это позволяет сделать вывод о том, что требования первой нормальной формы выполняются.

Проведем анализ функциональных зависимостей между атрибутами в отношении «Заказ» из базы данных «Прокат». В таблице 2.1 показана зависимость всех атрибутов от атрибута «Код_заказа». Первичным ключом в отношении «Заказ» может быть только атрибут «Код_заказа», так как именно от этого атрибута функционально зависят все остальные атрибуты. Так как первичный ключ «Код_заказа» отношения «Заказ» является простым, то из этого следует, что в отношении нет частичных функциональных зависимостей, поэтому схема отношения находится во второй нормальной форме. Необходимо далее провести анализ предметной области с целью выявления и других функциональных зависимостей, если они есть. Если при анализе схемы отношения не выявлена зависимость непервичных атрибутов от других непервичных атрибутов, то можно сделать вывод об отсутствии транзитивных зависимостей в отношении «Заказ» и о том, что требования третьей нормальной формы выполняются.

Наименование атрибутов	Функциональные зависимости
Код_заказа	←
дата_заказа	←
код_клиента	←
код_носителя	←
код_сотрудника	←
дата_возврата	←
стоимость_услуги	←

При разработке реляционной базы данных необходимо проанализировать функциональные зависимости между атрибутами всех отношений и выбрать идентифицирующие атрибуты, которые в реляционной модели данных используются в качестве первичных ключей.

После этого необходимо нормализовать отношения, исключив частичные и транзитивные функциональные зависимости. Проверить соответствие схем отношений требованиям второй и третьей нормальной формы.

Логическая связь отношений в реляционной БД является иерархической. Одно из отношений является основным (родительским или PARENT TABLE), а другое – подчиненным (дочерним или CHILD TABLE).

Для поддержания связей оба отношения должны содержать одинаковые наборы атрибутов (ключи), по которым они связаны. В основном отношении это первичный ключ (PRIMARY KEY), который однозначно определяет кортеж основного отношения. В подчиненном отношении этот же набор атрибутов называют внешним ключом отношения (FOREIGN KEY). В подчиненном отношении множество кортежей, определяемых внешним ключом, связано с единственным кортежем основного отношения. Условие связи отношений:

PARENT TABLE. PRIMARY KEY = CHILD TABLE. FOREIGN KEY

3.3.6 Даталогическая модель БД

3.3.6.1. Состав таблиц

При разработке даталогической модели реляционной базы данных необходимо определить состав таблиц. Затем для каждого поля таблицы указать размер поля (в количестве символов), тип данных. Для первичных ключей необходимо ввести запрет неопределенных значений. Для остальных полей возможность запрета неопределенных значений определяется семантикой предметной области. Пример описания состава таблицы приведен в таблице 3.2.

Таблица 3.2 – Состав таблицы «Клиент»

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ Наименование: Таблица 3.2 – Состав таблицы «Клиент»		Тип поля	Размер поля	Допустимость неопределенных значений
Сертификат: 12000002A633E3D113AD425FB50002000002A6				
Владелец: Шебзухова Татьяна Александровна				
Действителен: с 20.08.2021 по 20.08.2022				

Код_клиента	Integer		NOT NULL
ФИО_клиента	Character	30	
Паспортные_данные	Character	20	
адрес	Character	20	
телефон	Character	15	
категория	Character	15	

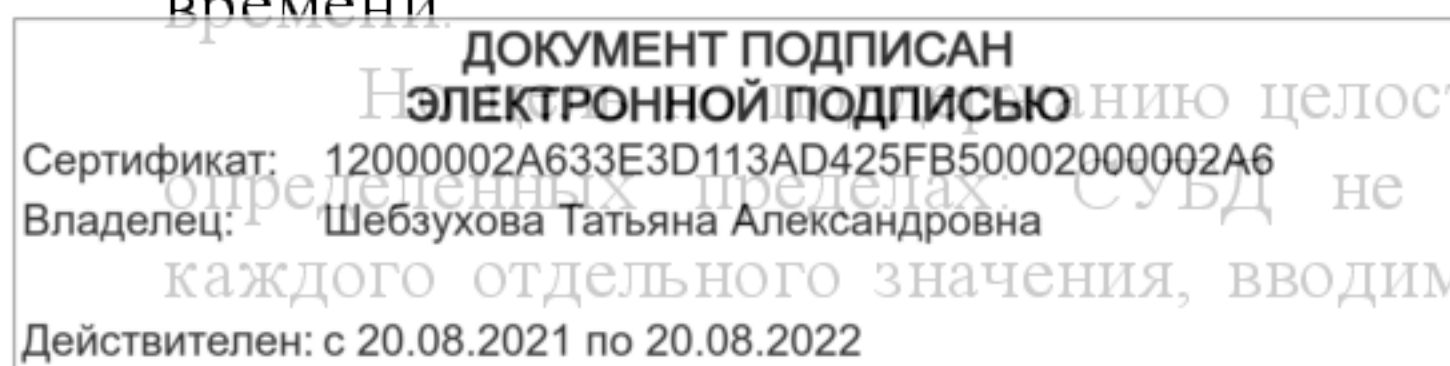
Для создания первичных и внешних ключей отношений используются индексы или индексные выражения, подробно описанные в методических указаниях к лабораторным работам. После определения состава отношений (таблиц) и описания их полей (по аналогии с таблицей 3.2) необходимо установить постоянные отношения между таблицами, т.е. связи по ключам. Для этого необходимо выбрать родительские (управляющие) таблицы, на первичные ключи которых ссылаются другие (дочерние) таблицы.

3.3.6.2 Средства поддержания целостности

Важнейшим вопросом, возникающим при проектировании и эксплуатации систем обработки данных, является обеспечение целостности данных. Нарушение целостности может быть связано не только с ошибками оператора при вводе данных, но и со сбоями в работе системы. Отказ в работе системы может возникнуть, когда одни операции были выполнены, а другие, связанные с ними определенными ограничениями операции, не успели выполняться. Например, удалена запись о сотруднике, а не удалены относящиеся к ней записи о его детях и т. п. Нарушение целостности возникает при вводе или корректировке данных, а также при выполнении реляционных операций (проекция, соединение).

Решение проблемы целостности заключается в обеспечении в любой момент времени правильности данных в базе данных, для этого используют правила проверки достоверности данных гарантирующие, что недействительные данные не попадут в таблицы БД. Целостность данных обеспечивается набором специальных предложений, называемых ограничениями целостности. Ограничения целостности, присущие той или иной предметной области, должны быть выявлены при обследовании и зафиксированы в инфологической модели. Для того чтобы обеспечить целостность при выполнении реляционных операций, необходимо, чтобы соблюдались определенные правила при их выполнении, а также чтобы файлы были правильно спроектированы.

Целостность (от англ. integrity – нетронутость, неприкосновенность, сохранность, целостность) – это правильность данных в любой момент времени.



Нарушение целостности может быть достигнута лишь в определенных пределах. СУБД не может контролировать правильность каждого отдельного значения, вводимого в базу данных. Например, нельзя

обнаружить, что вводимое значение 5 (представляющее номер дня недели) в действительности должно быть равно 3. Поддержание целостности базы данных может рассматриваться как защита данных от неверных и случайных действий персонала или разрушений. Ограничения целостности определяются в большинстве случаев особенностями предметной области, хотя могут отражать и чисто информационные характеристики. Например, в проектируемой базе данных «Прокат» примером одного из ограничений может служить запрет на ввод будущей даты.

Ограничения целостности могут относиться к разным информационным объектам базы данных, в том числе атрибутам (полям), кортежам (строкам, записям), отношениям (таблицам, файлам), связям между файлами и так далее. Современные СУБД имеют ряд средств для поддержания целостности баз данных.

Выделяют три группы правил поддержания целостности:

- Целостность по сущностям.
- Целостность по ссылкам.
- Целостность, определяемая пользователем.

Целостность по сущностям состоит в следующем: Не допускается, чтобы первичный ключ сущности принимал неопределенное значение. Неопределенные значения соответствуют пустым клеткам таблиц. Запрет неопределенных значений обозначают как: «NOT NULL». Ключевое слово NULL может обозначать в базах данных различные понятия: неопределенное значение, ложь (в логическом поле), нуль (в числовом поле). При попытке ввода новой строки с неопределенным первичным ключом операция ввода отвергается средствами СУБД.

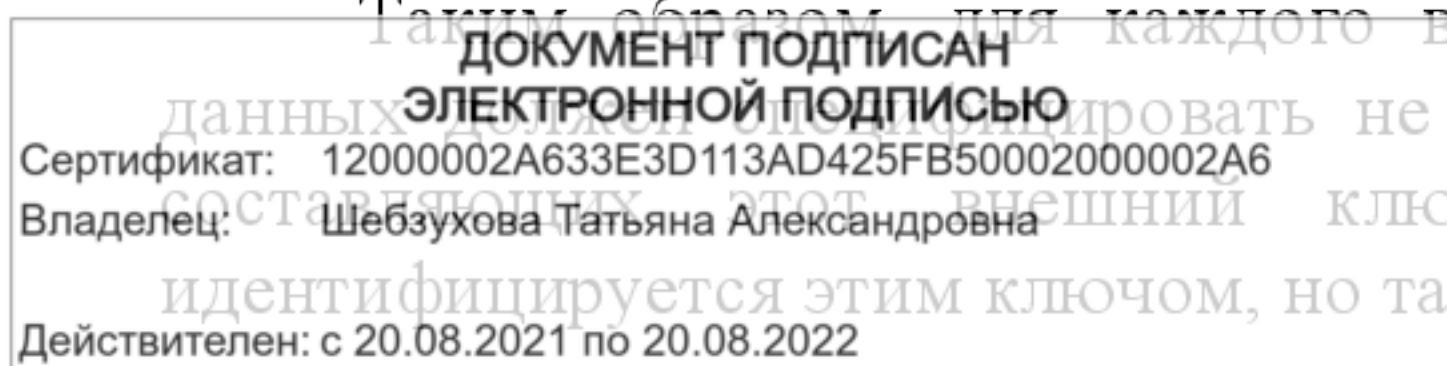
Первичный ключ должен быть уникален. При попытке ввода двух разных строк с одинаковыми первичными ключами операция ввода отвергается средствами СУБД. Таким образом, целостность по сущностям, которая ограничивает возможные значения первичного ключа, поддерживается автоматически средствами СУБД.

Внешний ключ может принимать неопределенные значения. СУБД не отвергает ввод строк с неопределенными значениями внешнего ключа. В логически связанных таблицах значение внешнего ключа дочерней таблицы должно быть либо:

- равно значению первичного ключа родительской таблицы;
- полностью неопределенным.

Целостность по ссылкам определяет механизм каскадирования изменений первичного ключа отношения, при котором согласовано изменяются логически связанные с первичным внешние ключи, или механизм, который отвергает эти изменения.

Таким образом, для каждого внешнего ключа проектировщик базы данных должен указать, как каскадировать не только поле или комбинацию полей, составляющих этот внешний ключ, и основную таблицу, которая идентифицируется этим ключом, но также и ответы на вопросы:



- что должно происходить при попытке ОБНОВЛЕНИЯ первичного ключа родительской таблицы, на которую ссылается некоторый внешний ключ?

- что должно происходить при попытке УДАЛЕНИЯ первичного ключа родительской таблицы, на которую ссылается некоторый внешний ключ?

На эти вопросы можно дать три ответа (по выбору проектировщика):

1. КАСКАДИРОВАНИЕ. Операция обновления (или удаления) «каскадируется» с тем, чтобы обновить (удалить) также и внешний ключ в дочерней таблице. (На лабораторных работах для каскадирования обновления (и удаления) используется оператор CASCADE).

2. ОГРАНИЧЕНИЕ. Обновляются (или удаляются) первичные ключи лишь тех записей, на которые нет ссылок в дочерних таблицах. При этом используется оператор RESTRICT.

3. УСТАНОВЛЕНИЕ неопределенных значений (NULL-значений). Внешний ключ устанавливается в неопределенное значение, а затем обновляется первичный ключ родительской таблицы.

Целостность, определяемая пользователем. Для любой конкретной базы данных существует ряд дополнительных специфических правил, которые относятся к ней одной и определяются разработчиком. Чаще всего контролируется:

- уникальность тех или иных атрибутов,
- диапазон значений (обычно используется для числовых полей, различают открытые и закрытые диапазоны; первые фиксируют значение только одной из границ, а вторые фиксируют обе границы; например, экзаменационная оценка от 2 до 5;
- принадлежность набору значений или *задание домена* (пол "М" или "Ж").

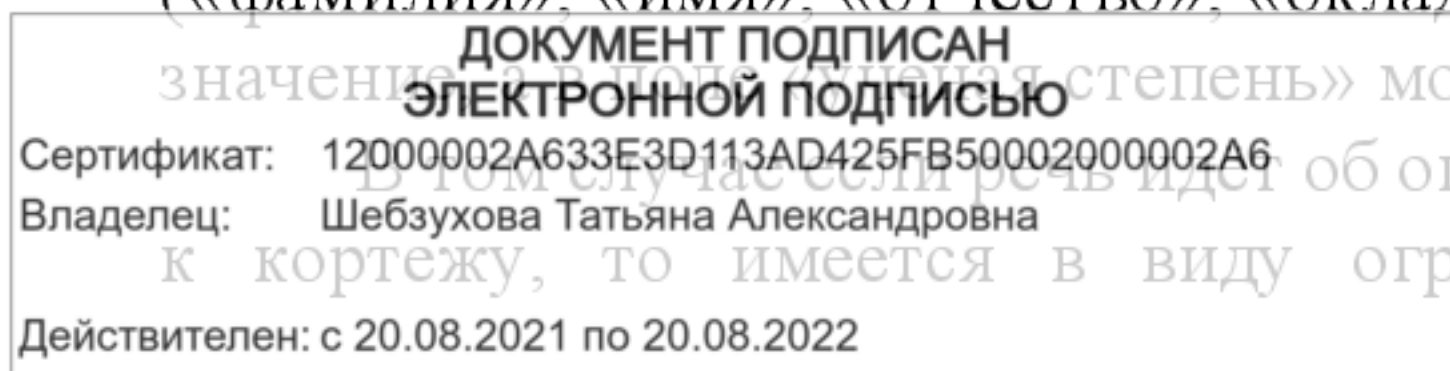
Ограничения целостности – это утверждения о допустимых значениях отдельных информационных единиц и связях между ними.

Ограничение целостности устанавливает правила на уровне баз данных, определяя набор проверок для таблиц системы. Эти проверки автоматически выполняются всякий раз, когда вызывается оператор вставки, модификации или удаления данных в таблице. Если какие-либо ограничения нарушены, операторы отменяются.

Для полей в таблицах, входящих в структуру базы данных, в основном используются следующие виды ограничений.

1. *Тип и формат поля.*
2. *Признак непустого поля.* Характеризует недопустимость пустого значения поля в БД. Например, поля со сведениями о сотрудниках («фамилия», «имя», «отчество», «оклад») обязательно должны иметь какое-то значение.

В таблицах, содержащих ограничения целостности, относящихся к кортежу, то имеется в виду ограничение на значение всей строки,



рассматриваемой как единое целое, или ограничения на соотношения значений отдельных полей в пределах одной строки. Естественным ограничением является требование уникальности каждой строки таблицы.

По определению в реляционном отношении не может быть одинаковых кортежей. Однако не все реляционные СУБД обеспечивают автоматическое соблюдение этого ограничения. Пример ограничения на соотношения полей внутри одного кортежа: значение поля «дата возврата» не должно превышать «дату заказа».

Если встроенные возможности СУБД не позволяют контролировать подобные ограничения целостности, то надо написать универсальную программу (создать процедуру), позволяющую это делать.

Видом ограничения является также *запрет на обновление*. Он может относиться как к отдельному полю, так и к записи или целой таблице. В некоторых СУБД существует запрет на корректировку ключевого поля.

Запрет на обновление может быть связан с технологией обработки данных или со спецификой предметной области. Например, если описывается объект «Личность», то его атрибуты дата и место рождения являются постоянными и не могут меняться. Для соответствующих полей в БД рекомендуется задавать запрет на обновление. При этом запрет на обновление может относиться как к отдельному полю, так и ко всей записи и или файлу.

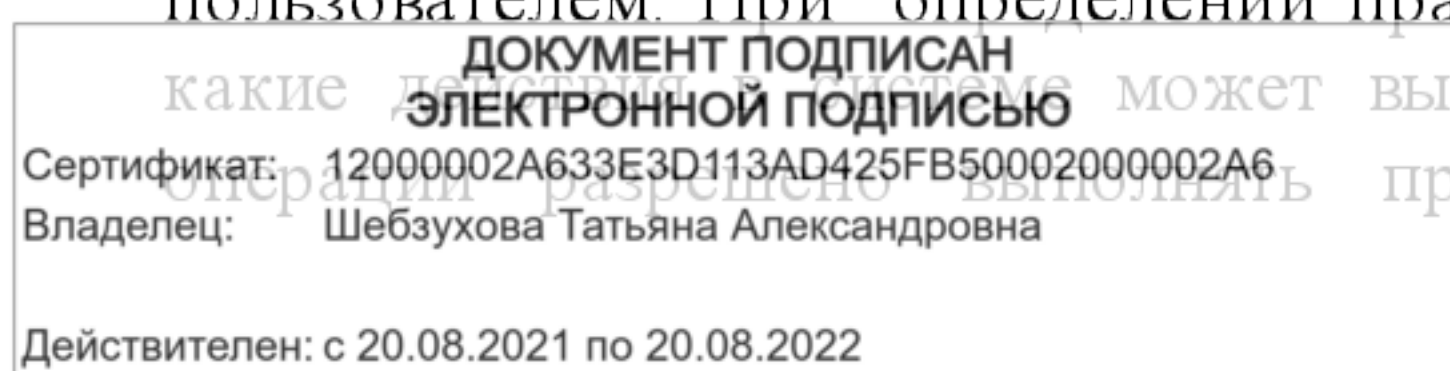
3.3.7 Запросы к базе данных

В шестом подразделе практической части курсового проекта по дисциплине «Управление данными» следует привести не менее 10 запросов всех типов, реализуемых средствами СУБД и средствами языка SQL. Составить запросы с коррелированными и некоррелированными подзапросами. Предварительно они должны быть сформулированы на естественном языке и выражены в терминах реляционной алгебры или реляционного исчисления. Примеры возможных запросов к базе данных и способы их реализации приведены в приложении 3.

3.3.8 Разработка механизмов защиты данных от несанкционированного доступа

При разработке механизмов защиты данных в разрабатываемой базе необходимо проанализировать состав обслуживающего персонала, который будет работать с ней. Рассмотреть привилегии, предоставляемые пользователям для работы с базой данных, таблицами, представлениями.

Целью управления доступом к объектам базы данных является ограничение действий, проводимых зарегистрированным в системе пользователем. При определении прав доступа администратор БД решает, какие действия пользователь может выполнять в системе, а также какие операции разрешено выполнять приложениям, запущенным от имени



пользователя. Управление доступом предназначено для предотвращения действий пользователя, которые способны нанести вред базе данных.

Под управлением доступом понимается возможность субъекта (сущность, определяющая пользователя при работе в системе) проводить действия над объектом (как базы данных целиком, так и ее компонентов). Различаются три метода управления доступом: дискреционный, обязательный, ролевой.

Дискреционный контроль предполагает, что владелец объекта сам может указывать, кто имеет доступ к объекту и вид этого доступа.

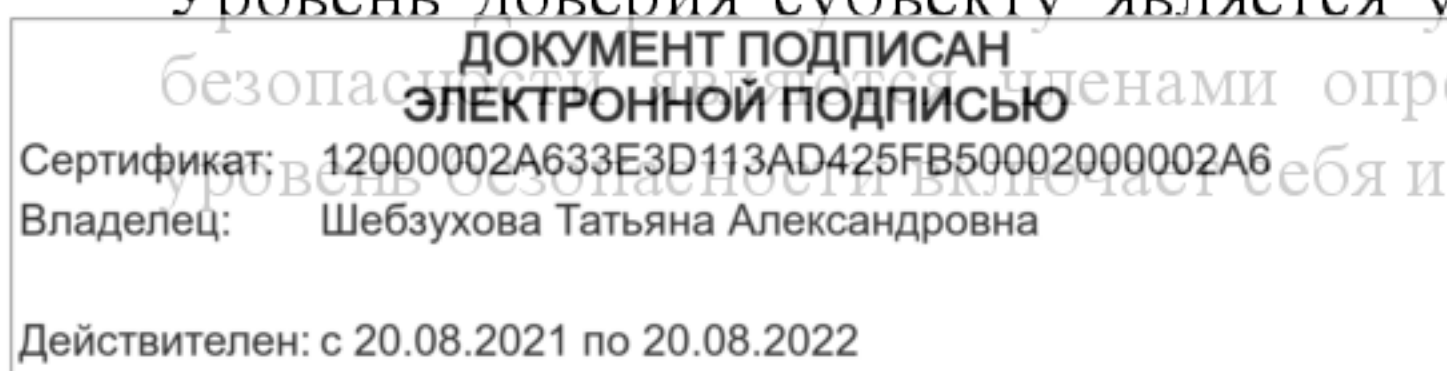
Для управления доступа к объектам дискреционный контроль доступа использует идентификационную информацию объекта и список доступа, который содержит субъекты и ассоциированные с ними типы доступа. Во время запроса доступа к объекту, система производит поиск субъекта в списке прав доступа объекта, и, в случае, если субъект там присутствует и разрешённый тип доступа включает требуемый тип, разрешает доступ. В противном случае запрещает доступ.

Благодаря своей гибкости, дискреционный контроль доступа широко используется. Одним из значительных недостатков данного вида контроля является отсутствие полной гарантии того, что информация не станет доступна другим субъектам, которые не имеют к ней доступа. Причина этого кроется в том, что субъект, имеющий право чтения информации может без уведомления владельца объекта передать её другим субъектам, не имеющим такого права. Дискреционная модель контроля доступа не накладывает ограничений на дальнейшее распространение информации после того, как субъект её получил.

Помимо этого, к недостаткам можно отнести ещё одну особенность дискреционной модели контроля доступа: объекты в системе принадлежат субъектам, которые настраивают доступ к ним для других. Но на практике, в большинстве случаев, данные в системе принадлежат всей системе, а не отдельным субъектам. Информационная система является наиболее распространённым примером.

Различают закрытую и открытую систему дискреционного контроля. Под закрытой системой понимается такая система, в которой изначально объект никому не доступен и список разрешений описывается в списке прав доступа, а открытая – та, в которой к объектам все имеют полный доступ, а в списке доступа описывается список ограничений.

Обязательный контроль доступа осуществляет управление доступом, опираясь на элементы самой системы. Каждому из элементов системы назначается уровень безопасности, который описывает важность этого объекта и ущерб, причинённый при разглашении информации в этом объекте. Уровень доверия субъекту является уровнем его безопасности. Все уровни безопасности образуют иерархию, то есть каждый уровень включает в себя и уровни, находящиеся ниже.



Доступ субъекта к структурному элементу базы данных (таблице, запросу и т. д.), а также самой базе данных предоставляется в случае выполнения определённого условия отношения между уровнями безопасности субъекта и объекта. Доступ на выполнение операции даётся при выполнении определённых условий. Доступ на чтение даётся в случае, если уровень безопасности субъекта включает в себя уровень безопасности объекта. Доступ на запись даётся в случае, если уровень безопасности субъекта включается в уровень безопасности объекта. При выполнении этих условий гарантируется, что данные высокоуровневых объектов не попадут в низкоуровневые объекты.

Однако в такой модели существуют два момента, ставящие под вопрос её непротиворечивость.

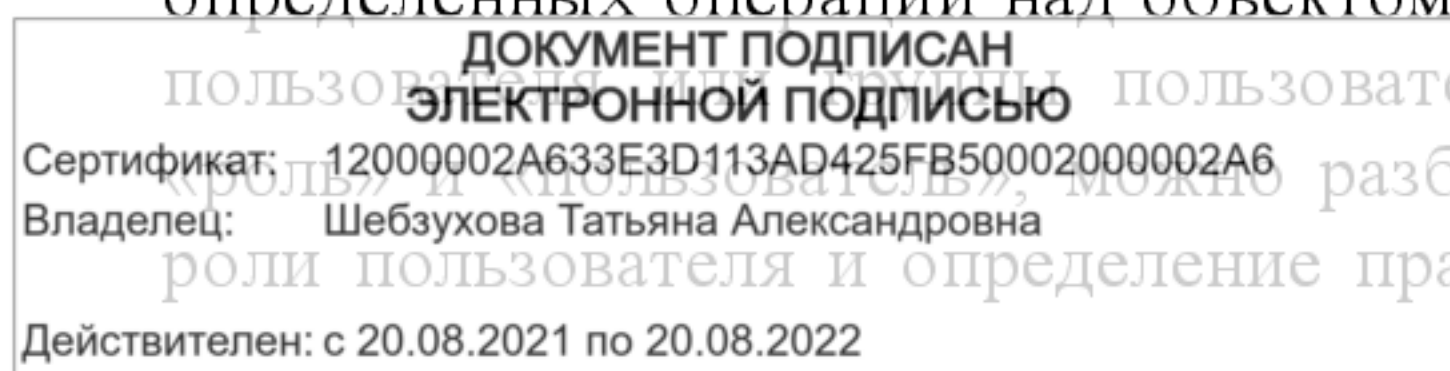
1. Пользователь нижнего уровня может записывать информацию в объекты верхних уровней. Он может переписать существующий объект своим собственным и информация будет потеряна. Такой недостаток можно устранить, запретив запись на более верхние уровни. При такой схеме доступ на чтение будет даваться, если уровень безопасности субъекта включается в себя уровень безопасности объекта, а доступ на чтение будет даваться в случае, если уровень безопасности субъекта будет равняться уровню безопасности объекта.

2. Пользователи с более высоким уровнем не могут изменять объекты с более низким уровнем. Этот недостаток можно устранить, позволив пользователю при доступе к объектам выступать от имени субъектов с различными уровнями.

Часто, кроме обеспечения безопасности, также требуется обеспечение её достоверности. Чем выше уровень доверия объекта, тем выше его достоверность. Чем выше уровень безопасности субъекта, тем достовернее информация, которую он вносит в систему. Для такой модели описанные выше правила должны быть изменены: доступ на запись даётся в случае, если уровень безопасности субъекта включает в себя уровень безопасности объекта, а доступ на чтение даётся в случае, если уровень безопасности субъекта включается в уровень безопасности объекта. То есть критерии просто поменялись местами.

Вместе с использованием уровней безопасности можно использовать категории. В таком случае, кроме уровня безопасности, каждому объекту и субъекту можно назначить список категорий, к которым он относится. Категории объекта используются для описания тех областей, в которых он используется. Категории субъекта описывают области, в которых он работает. Такая система позволит детальнее управлять доступом в системе.

В классических моделях разграничения доступа права на выполнение определённых операций над объектом должны быть прописаны для каждого пользователя. Благодаря разделению понятий «роль» и «пользователь», можно разбить задачу на две части: определение роли пользователя и определение прав доступа к объекту для роли. Такой



подход позволяет упростить процесс администрирования, так как при изменении области ответственности пользователя, требуется лишь убрать у него старые роли и назначить новые, соответствующие его текущим обязанностям. Эта же процедура потребует массу усилий при переназначении новых прав, если права доступа определялись напрямую между пользователями и объектами.

Посредством построения иерархии ролей, систему ролей можно настроить так, чтобы она точнее отображала реальные бизнес процессы. Каждая роль, вместе со своими привилегиями, может ещё наследовать привилегии других ролей. Это тоже упростит администрирование системы

Ролевая модель позволяет пользователю регистрироваться в системе с наименьшей ролью для выполнения требуемых задач. Пользователям, у которых множество ролей, не всегда требуются все привилегии для выполнения определённой задачи.

Принцип наименьшей привилегии обеспечивает достоверность данных в системе. Необходимо давать пользователю только те разрешённые привилегии, которые ему нужны для выполнения определённой задачи. Для этого следует выяснить цель задачи, привилегии, необходимые для её выполнения и ограничить этим набором привилегии пользователя. Запрещение привилегий, не требуемых для выполнения текущей задачи, не позволит обойти политику безопасности системы.

Разделение обязанностей также является одним из важных принципов в системе управления доступом. Не редки ситуации, когда ряд определённых действий не может быть выполнен одним и тем же человеком для предотвращения мошенничеств, например, операции создания и подтверждения платежа. Система ролевого управления доступом позволит решить эту задачу без особых затруднений.

Модель состоит из сущностей пользователя, роли и привилегии, где пользователь – человек либо программа запущенная им, роль – вид деятельности человека в организации, а привилегия – разрешение доступа к объектам системы. У пользователя может быть несколько ролей, а одна роль может принадлежать нескольким пользователям. Также и несколько привилегий могут принадлежать одной роли, а несколько ролей – иметь одну привилегию.

Сессия, которую активизирует пользователь для выполнений задачи, позволяет отнести пользователя к множеству ролей. В этот момент система определяет роли и привилегии, требующиеся пользователю для выполнения задачи, и запрещает все остальные.

В проектируемой базе данных «Прокат» можно рассматривать следующие механизмы доступа как к базе данных целиком, так к ее компонентам:

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ
Сертификат: 12000002A633E3D113AD425FB50002000002A6
Владелец: Шебзухова Татьяна Александровна
Действителен: с 20.08.2021 по 20.08.2022

- парольная защита (ролевая защита).

Пользователями базы данных «Прокат» могут быть менеджер первого звена, менеджер по работе с клиентами и администратор базы данных.

Из всех перечисленных пользователей наиболее широким спектром полномочий обладает администратор, который может выполнять все операции с базой данных и ее компонентами.

3.3.9 Требования к техническому обеспечению

В этом разделе следует сформулировать требования к системе БД, например, определить быстродействие системы, требования к аппаратному обеспечению, объему оперативной памяти и жесткого диска.

При описании технического обеспечения базы данных необходимо соблюдать следующие требования:

- к видам технических средств, в том числе к видам комплексов технических средств, программно-технических комплексов и других комплектующих изделий, допустимых к использованию;
- к функциональным, конструктивным и эксплуатационным характеристикам средств технического обеспечения системы.

Техническое обеспечение системы должно максимально и наиболее эффективным образом использовать существующие в фирме средства компьютерной и вычислительной техники.

Для эффективного функционирования базы данных «Прокат» необходимы персональные компьютеры, являющиеся рабочими местами менеджера первого звена, менеджеров по работе с клиентами и администратора, которые могут обладать следующими техническими характеристиками:

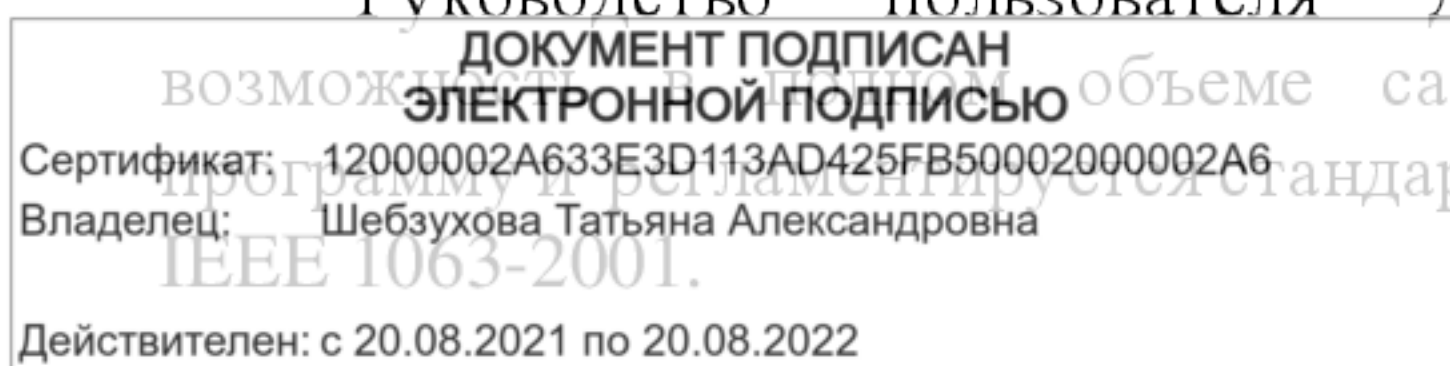
- процессор – Intel Core i3 380M;
- объем оперативной памяти – 6 Гб;
 - жесткий диск объемом не менее 500 Гб;
 - дисковая подсистема – 40 Гб;
 - устройство чтения компакт-дисков (DVD-ROM);
- сетевой адаптер – 100 Мбит.

3.3.10 Инструкция по использованию базы данных

При разработке инструкции для пользователя базы данных необходимо описать, как производится вызов программы, и как характеризуются ее основные функции. В этом разделе приводятся справочные сведения о разработанной базе данных, описание экранных форм и отчетов.

Руководство пользователя — один из основных программных документов. Основная задача документа состоит в том, чтобы обеспечить пользователям возможность самостоятельно решать все основные задачи, на которые нацелена программа.

Руководство пользователя должно обеспечить пользователям возможность самостоятельно освоить и применять



программы и регламенты ее использования стандартами ГОСТ 34.201-89, РД 50-34.69890,

Руководство пользователя содержит полное описание программы с точки зрения ее целевого применения. В руководстве пользователя обязательно должны быть описаны:

- назначение базы данных;
- основные задачи и возможности;
- способ отражения предметной области в базе данных;
- пользовательский интерфейс базы данных;
- порядок решения основных пользовательских задач;
- все функции программы и порядок их применения;
- пользовательская настройка программы;
- проблемы при использовании и способы их решения.

При документировании небольших программ в руководство пользователя часто включают инструкции по установке, настройке, администрированию, обновлению и прочему обслуживанию программы.

В зависимости от особенностей программы и целевой аудитории руководство пользователя по способу изложения материала может приближаться к учебнику или, наоборот, к справочнику. Порядок изложения материала в руководстве пользователя определяется *пользовательской перспективой* программы.

Если программа представляет собой инструмент, позволяющий решать практические задачи из некоторого конечного набора, то в руководстве приводят типовые процедуры решения каждой из них. Каждое из этих действий можно разложить на последовательные элементарные шаги, во всяком случае, для типичных ситуаций. Руководство пользователя, построенное по принципу пользовательских задач, напоминает учебник, хотя, как правило, лишено присущего учебникам методического аппарата: проверочных заданий, вопросов, упражнений.

Если программа представляет собой среду, в пределах которой пользователь может решать задачи, поставленные им самостоятельно, руководство пользователя должно быть ближе к справочнику.

Если программа является инструментом, с помощью которого пользователь контролирует состояние того или иного объекта, например, промышленной установки, то руководство пользователя строится по принципу таблицы: сообщение программы - реакция или возможные реакции пользователя.

Если пользователь применяет программу для решения задач в нетривиальных предметных областях, в руководство пользователя настоятельно рекомендуется включить концептуальный раздел. В нем должен быть описан реализованный в программе способ представления объектов реального мира, чтобы пользователь хорошо понимал, с какими из них и на

каком уровне он может работать.

ДОКУМЕНТ ПОДПИСАН

ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 12000002A633E3D113AD425FB50002000002A6

Владелец: Щербухова Татьяна Александровна

Действителен: с 20.08.2021 по 20.08.2022

структура руководства будет в основном описываемой программы. Обычно структура руководства пользователя содержит следующие разделы: