

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное
учреждение
высшего профессионального образования
Национальный исследовательский университет
«Высшая школа экономики»
Московский институт электроники и математики
Национального исследовательского университета
«Высшая школа экономики»

Домашняя работа по курсу «Базы данных»
на тему:
«База данных производства фруктовой продукции».

Выполнил:
студент группы
БИВ-135
Чибисов А.П.
Проверила:
к.т.н, доцент
Карпова И.П.

Москва, 2016

Оглавление

1. Инфологическое проектирование.....	3
1.1. Анализ предметной области	3
1.2. Анализ информационных задач и круга пользователей системы.....	4
2. Определение требований к операционной обстановке	4
3. Выбор системы управления базой данных и других программных средств	5
4. Логическое проектирование базы данных	6
4.1. Преобразование ER-диаграммы в схему реляционной базы данных.....	6
4.2. Составление реляционных отношений.....	6
4.3. Нормализация реляционных отношений.....	9
4.4. Определение дополнительных ограничений целостности	13
4.5. Описание групп пользователей и прав доступа.....	14
5. Реализация проекта базы данных	14
5.1. Создание таблиц.....	14
5.2. Создание представлений (готовых запросов)	17
5.3. Создание функции.....	20
5.4. Создание процедуры.....	21
5.5. Создание триггеров.....	23
5.6. Создание индексов.....	26
5.7. Разработка стратегии резервного копирования	26
6. Список литературы	26

1. Инфологическое проектирование

1.1. Анализ предметной области

Особенности предметной области:

- Каждый поставщик может поставлять несколько поставок фруктов, у одной поставки должен быть только один поставщик.
- Одна поставка может входить в состав нескольких продукций, у одной продукции может быть в составе несколько поставок фруктов.
- Один отдел может производить несколько продукций, одна продукция может быть произведена только одним отделом.
- Каждый сотрудник работает в определенном отделе, в одном отделе может быть несколько сотрудников.
- Каждый отдел специализируется на выпуске определенного вида продукции (пр.: джем, варенье и т.п.).
- Зарплата сотрудника зависит только от занимаемой им должности.

Сущности предметной области:

- 1) **Поставщики.** Атрибуты: ид, название, адрес, телефон, контактное лицо.
- 2) **Поставки продуктов.** Атрибуты: ид, название, дата сбора, дата поставки, срок годности, вес, цена, страна происхождения.
- 3) **Продукция.** Атрибуты: ид, название, тип упаковки, дата производства, срок годности, масса брутто.
- 4) **Отделы.** Атрибуты: ид, название, телефон, специализация.
- 5) **Сотрудники.** Атрибуты: ид, ФИО, дата рождения, пол, паспортные данные, СНИЛС, ИНН, адрес, телефон, должность, заработная плата, образование.



Рис. 1. ER–диаграмма «Производство фруктовой продукции»

1.2. Анализ информационных задач и круга пользователей системы

Группы пользователей и их примерные задачи и запросы:

1. Руководители организации:

- придумывание новой продукции
- создание отделов
- создание/изменение должностей и их зарплат
- заключение договоров с поставщиками
- просмотр полных сведений о производстве

2. Руководители отделов с производством:

- изменение информации о продукции
- просмотр сведений о поставках
- просмотр сведений о сотрудниках своего отдела
- просмотр сведений о начальниках других отделов

3. Менеджеры производства:

- изменение единиц измерения
- изменение информации о составе
- изменение списка типов упаковок

4. Рабочие отдела с производством:

- просмотр данных о других людях этого отдела

5. Сотрудники отдела кадров:

- прием/увольнение сотрудников
- изменение данных о сотрудниках

6. Бухгалтеры:

- получение ведомости на выплату зарплаты
- получение информации о закупленных поставках

7. Отдел закупок:

- ведение отчетности по поставкам.

2. Определение требований к операционной обстановке

На основе результатов анализа предметной области рассчитаем приблизительный объём памяти, требуемый для хранения данных.

Ориентировочно:

- на производстве будет существовать примерно 15 отделов на которых будут трудиться примерно 1 000 сотрудников;
- будет выпускаться около 100 000 продукции в месяц;
- поставщиков будет порядка 1000 фирм, которые будут поставлять приблизительно 100 000 поставок в месяц;

- для связи поставок и поставщиков будет храниться информации о составе продукции примерно $100\,000 * 5$, где 5 – примерное максимальное вхождение фруктов в состав.

Наименование	Примерный объем одной записи, КБ	Количество записей, шт	Общий объем, КБ
Поставщики	0,2	1 000	200
Поставки фруктов	0,25	100 000	25 000
Сотрудники	1	1 000	1 000
Отделы	0,1	15	1,5
Продукция	0,5	100 000	50 000
Состав	0,1	500 000	50 000
Всего			126201,5

Тогда объем памяти для хранения данных примерно составит:

$$M_d = 2 * 126,3 = 252,6 \text{ МБ}$$

Коэффициент 2 введён для того, чтобы учесть необходимость выделения памяти под дополнительные структуры (например, индексы). Объем памяти будет увеличиваться ежемесячно примерно на столько же при условии сохранения показателей выпуска.

Для хорошего функционирования нашей базы данных необходим немаленький объем памяти, растущий ежемесячно, это необходимо учитывать, при выборе сервера базы данных.

3. Выбор системы управления базой данных и других программных средств

Анализ информационных задач показывает, что для реализации требуемых функций подходят почти все СУБД для ПЭВМ (MS Access, Firebird, MySQL и др.). Все они поддерживают реляционную модель данных и предоставляют разнообразные возможности для работы с данными.

Для проектирования базы данных производства фруктовой продукции нами была выбрана свободная реляционная система управления базами данных MySQL, так как изучению именно этой системы была часть курса «Базы данных».

4. Логическое проектирование базы данных

4.1. Преобразование ER-диаграммы в схему реляционной базы данных

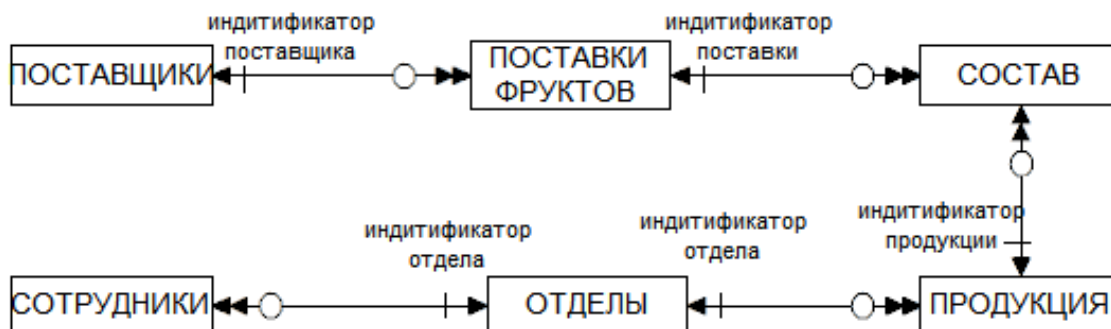


Рис. 2. Схема РБД, полученная из ER-диаграммы производства фруктовой продукции

4.2. Составление реляционных отношений

Таблица 1. Поставщики (Providers)

Содержание поля	Имя поля	Тип, длина	Примечания
id	id_p	number(3)	ПК
название	name_p	varchar(30)	обязательное поле
адрес	address_p	varchar(100)	обязательное поле
телефон	phone_p	varchar(10)	обязательное поле
контактное лицо	contname_p	varchar(30)	

Таблица 2. Поставки фруктов (PostavkiFruits)

Содержание поля	Имя поля	Тип, длина	Примечания
id	id_pf	number(3)	ПК
название	name_pf	varchar(30)	обязательное поле
дата сбора	datesbor_pf	date	обязательное поле
срок годности	srok_pf	date	обязательное поле
вес	weigh_pf	number(10,6)	обязательное поле, >0
единица измерения	edizm_pf	varchar(20)	обязательное поле
стоимость	price_pf	number(10,2)	обязательное поле, значение по умолчанию - 0

страна происхождения	country_pf	varchar(50)	обязательное поле
поставщик	postavshik_pf	number(3)	Внк к Providers
дата поступления	datepost_pf	date	обязательное поле, больше или равна даты сбора

Таблица 3. Состав (Sostav)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
id	id_s	number(3)	ПК
продукция	prod_s	number(3)	Внк к Production
фрукт	fruct_s	number(3)	Внк к PostavkiFruits
кол-во(масса)	kol_s	number(10,6)	обязательное поле, >0
единица измерения	edizm_f	varchar(20)	обязательное поле

Таблица 4. Продукция (Production)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
id	id_prod	number(3)	ПК
название	name_prod	varchar(50)	обязательное поле
тип упаковки	type_prod	varchar(30)	обязательное поле
дата производства	date_prod	date	обязательное поле
срок годности	srok_prod	date	обязательное поле, больше даты производства
произведено отделом	otd_prod	number(3)	Внк к Otdels
масса брутто	mas_prod	number(10,6)	обязательное поле, >0
единица измерения	edizm_prod	varchar(20)	обязательное поле

Таблица 5. Отделы (Otdels)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
id	id_o	number(3)	ПК
название	name_o	varchar(30)	обязательное поле
телефон	phone_o	varchar(10)	обязательное поле
специализация	spec_o	varchar(20)	обязательное поле

Таблица 6. Работники (Employees)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
id	id_e	number(3)	ПК
ФИО	name_e	varchar(65)	обязательное поле
дата рождения	born_e	date	обязательное поле
пол	sex_e	char(1)	обязательное поле
паспортные данные	passport_e	varchar(50)	обязательное поле
ИНН	inn_e	char(12)	обязательное поле
СНИЛС	snils_e	char(14)	обязательное поле
адрес	address_e	varchar(100)	обязательное поле
телефон	phone_e	varchar(10)	
отдел	otdel_e	number(3)	Внк к Otdels
должность	dolzn_e	varchar(30)	обязательное поле
заработная плата	sal_e	number(15)	обязательное поле
образование	obraz_e	varchar(30)	обязательное поле

4.3. Нормализация реляционных отношений

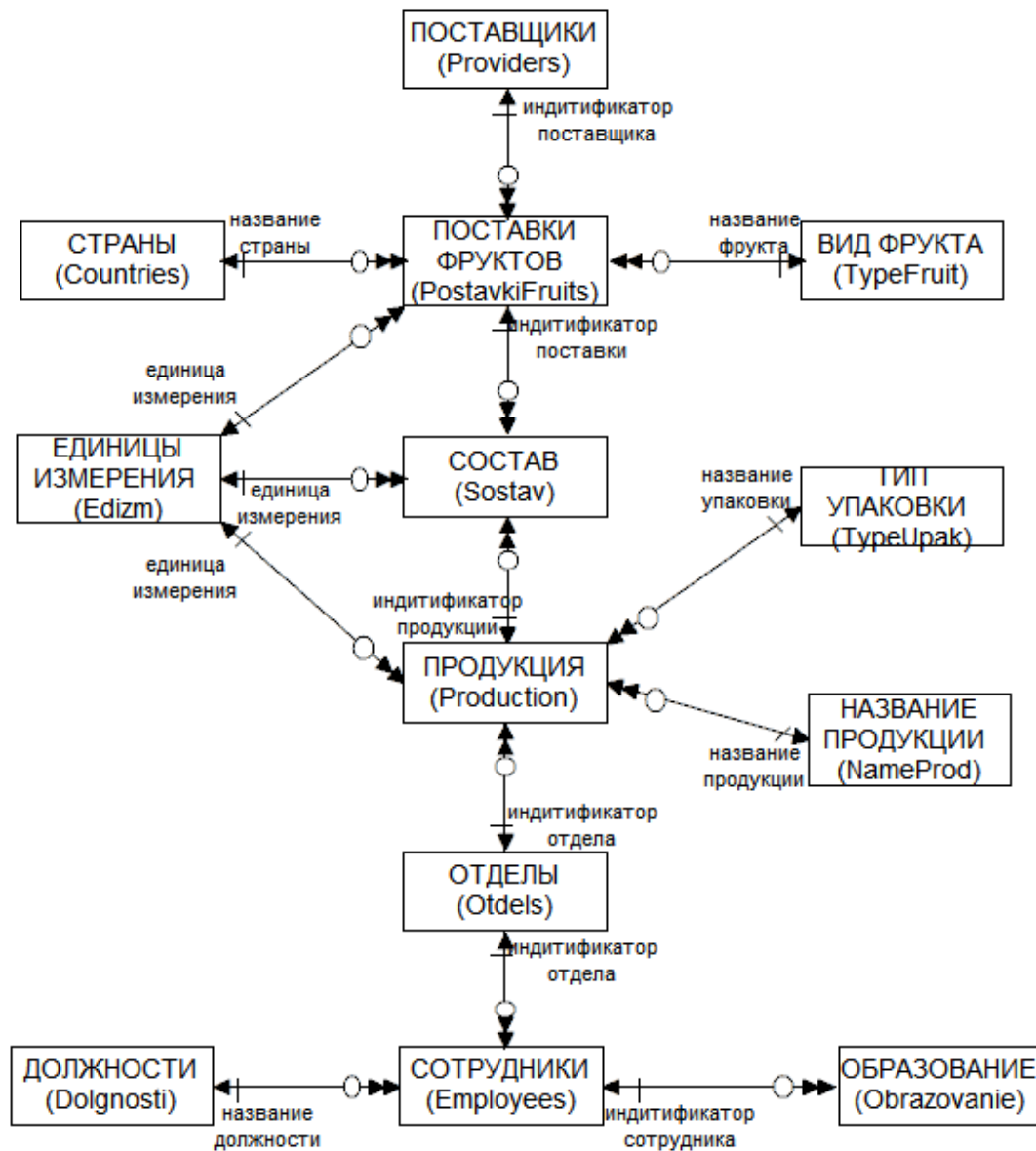


Рис. 3. Окончательная схема БД производства фруктовой продукции

Таблица 1.1 Поставщики (Providers)

Содержание поля	Имя поля	Тип, длина	Примечания
id	id_p	number(3)	ПК
название	name_p	varchar(30)	обязательное поле
адрес	address_p	varchar(100)	обязательное поле
телефон	phone_p	varchar(10)	обязательное поле
контактное лицо	contname_p	varchar(30)	

Таблица 2.1 Поставки фруктов (PostavkiFruits)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
id	id_pf	number(3)	ПК
название	name_pf	varchar(30)	Внк к TypeFruit
дата сбора	datesbor_pf	date	обязательное поле
срок годности	srok_pf	date	обязательное поле, больше даты сбора
вес	weigh_pf	number(10,6)	обязательное поле, >0
единица измерения	edizm_pf	varchar(20)	Внк к Edizm
стоимость	price_pf	number(10,2)	обязательное поле, значение по умолчанию - 0
страна происхождения	country_pf	varchar(50)	Внк к Countries
поставщик	postavshik_pf	number(3)	Внк к Providers
дата поступления	datepost_pf	date	обязательное поле, больше или равна даты сбора

Таблица 2.2. Виды фруктов (TypeFruit)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Название фрукта	name_f	varchar(30)	ПК

Таблица 2.3. Страны (Countries)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Название страны	name_count	varchar(50)	ПК

Таблица 2.4. Единицы измерения(Edizm)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
единица измерения	name_ed	varchar(20)	ПК
система мер	sistmer_ed	varchar(20)	обязательное поле
коэффициент перевода в кг	kg_ed	number(10,6)	обязательно поле, >0

Таблица 3.1 Состав (Sostav)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
id	id_s	number(3)	ПК
продукция	prod_s	number(3)	Внк к Producion
фрукт	fruct_s	number(3)	Внк к PostavkiFruits
кол-во(масса)	mas_s	number(10,6)	Обязательное поле, >0
единица измерения	edizm_s	varchar(20)	Внк к Edizm

Таблица 4.1 Продукция (Production)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
id	id_prod	number(3)	ПК
название	name_prod	varchar(50)	Внк к NameProd
тип упаковки	type_prod	varchar(30)	Внк к TypesUpak
дата производства	date_prod	date	обязательное поле
срок годности	srok_prod	date	обязательное поле, больше даты производства
произведено отделом	otd_prod	number(3)	Внк к Otdels
масса брутто	mas_prod	number(10,6)	обязательное поле, >0
единица измерения	edizm_prod	varchar(20)	Внк к Edizm

Таблица 4.2. Названия продукции (NameProd)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
название продукции	name_pr	varchar(50)	ПК

Таблица 4.3. Типы упаковки (TypeUpak)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
название упаковки	name_t	varchar(30)	ПК
вес упаковки	weigh_t	number(10,6)	обязательное поле, >0

Таблица 5.1 Отделы (Otdels)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
id	id_o	number(3)	ПК
название	name_o	varchar(30)	обязательное поле
телефон	phone_o	varchar(10)	обязательное поле
специализация	spec_o	varchar(20)	обязательное поле

Таблица 6.1 Работники (Employees)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
id	id_e	number(3)	ПК
фамилия	fname_e	varchar(25)	обязательное поле
имя, отчество	lname_e	varchar(40)	обязательное поле
дата рождения	born_e	date	обязательное поле
пол	sex_e	char(1)	обязательное поле
серия и номер паспорта	sernum_e	char(10)	обязательное уникальное поле
когда выдан паспорт	date_e	date	обязательное поле
кем выдан паспорт	given_e	varchar(50)	обязательное поле
ИНН	inn_e	char(12)	обязательное уникальное поле
СНИЛС	snils_e	char(14)	обязательное уникальное поле
адрес	address_e	varchar(100)	обязательное поле
телефон	phone_e	varchar(10)	
отдел	otdel_e	number(3)	Внк к Otdels
должность	dolzn_e	varchar(30)	Внк к Dolgnosti

Таблица 6.2. Должности (Dolgnosti)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
название должности	name_dol	varchar(30)	ПК
заработная плата	sal_dol	number(10,2)	обязательное поле, > 6250 руб.

Таблица 6.3. Образование (Obrazovanie)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
id работника	id_obr	number(3)	Внк к Employees
тип образования	type_obr	number(15)	обязательное поле
специальность	spec_obr	varchar(30)	
дата окончания учебного заведения	date_obr	date	обязательное поле
номер диплома/аттестата	numdip_obr	number(15)	

4.4. Определение дополнительных ограничений целостности

1. Атрибут пол (*sex_e*) может принимать одно из двух значений: 'ж','м'.
2. Поля серия и номер паспорта, ИНН (*sernum_e*, *inn_e*, *snils_e*) соответственно могут состоять только из цифр.
3. Атрибут тип образования (*type_obr*) может принимать одно из следующих значений: 'начальное', 'среднее', 'средне-специальное', 'высшее'.
4. Атрибут масса упаковки (*typeupak.weight_t*) должен указываться в кг.
5. Масса нетто продукции(*mas_prod*) должна быть равна суммарной массе($\sum mas_s$) входящих поставок плюс масса упаковки(*weight_t*). Все величины должны суммироваться и складываться в кг.

4.5. Описание групп пользователей и прав доступа

Таблицы	Группы пользователей (роли)					
	Руководители организации	Руководители отделов с производством	Менеджеры производства	Сотрудни ки отд. кадров	Бухгалтеры	Отдел закупок
Поставщики	SIUD					S
Поставки фруктов	S	S	S		S	SIUD
Виды фруктов	S					SIUD
Страны	S					SIUD
Единицы измерения	S	S	SIUD		S	S
Состав	S	S	SUID		S	
Продукция	S	SUID	S		S	
Название продукции	SIUD	S				
Типы упаковки	S	S	SUID			
Отдел	SIUD	S	S	S	S	
Работники	S			SIUD	S	
Должности	SIUD				S	
Образование	S			SIUD	S	

Где:

S – чтение данных (select);

I – добавление данных (insert);

U – модификация данных (update);

D – удаление данных(delete).

5. Реализация проекта базы данных

5.1. Создание таблиц

1. Поставщики (Providers):

```
create table providers
(
  id_p numeric(3) primary key not null,
  name_p varchar(30) not null,
  address_p varchar(100) not null,
  phone_p varchar(10) not null,
  contname_p varchar(30)
);
```

2. Поставки фруктов (PostavkiFruits):

```
create table postavkifruits
(
  id_pf numeric(3) primary key not null,
  name_pf varchar(30) references typefruit,
  datesbor_pf date not null,
  srok_pf date not null,
  weigth_pf numeric(10,6) not null check,
  edizm_pf varchar(20) references edizm,
  price_pf numeric(10,2) not null,
  country_pf varchar(50) references countries,
  postavshik_pf numeric(3) references providers,
  datepost_pf date not null
);
```

3. Виды фруктов (TypeFruit):

```
create table typefruit
(
  name_f varchar(30) primary key not null
);
```

4. Страны (Countries):

```
create table countries
(
  name_count varchar(50) primary key not null
);
```

5. Единицы измерения (Edizm):

```
create table edizm
(
  name_ed varchar(20) primary key not null,
  sistmer_ed varchar(20) not null,
  kg_ed numeric(10,6) not null
);
```

6. Состав (Sostav):

```
create table sostav
(
  id_s numeric(3) primary key not null,
  prod_s numeric(3) references production,
  fruct_s numeric(3) references postavkifruits,
  mas_s numeric(10,6) not null,
  edizm_s varchar(20) references edizm
);
```

7. Продукция (Production):

```
create table production
```

```
(
id_prod numeric(3) primary key not null,
name_prod varchar(50) references nameprod,
type_prod varchar(30) references typesupak,
date_prod date not null,
srok_prod date not null,
otd_prod numeric(3) references otdels,
mas_prod numeric(10,6) not null,
edizm_prod varchar(20) references edizm
);
```

8. Названия продукции (NameProd):

```
create table nameprod
(
name_pr varchar(50) primary key not null
);
```

9. Типы упаковки (TypeUpak):

```
create table typeupak
(
name_t varchar(30) primary key not null,
weigh_t numeric(10,6) not null
);
```

10. Отделы (Otdels):

```
create table otdels
(
id_o numeric(3) primary key not null,
name_o varchar(30) not null,
phone_o varchar(10) not null,
spec_o varchar(20) not null
);
```

11. Работники (Employees):

```
create table employees
(
id_e numeric(3) primary key not null,
fname_e varchar(25) not null,
lname_e varchar(40) not null,
born_e date not null,
sex_e char(1) not null,
sernum_e char(10) not null unique,
date_e date not null,
given_e varchar(50) not null,
inn_e char(12) not null unique,
snils_e char(14) not null unique,
address_e varchar(100) not null,
```



```

phone_e varchar(10),
otdel_e numeric(3) references ottdels,
dolgn_e varchar(30) references dolgnosti,
login_e varchar(15)
);

```

12. Должности (Dolgnosti):

```

create table dolgnosti
(
name_dol varchar(30) primary key not null,
sal_dol numeric(10,2) not null
);

```

13. Образование (Obrazovanie):

```

create table obrazovanie
(
id_obr numeric(3) references employees,
type_obr varchar(15) not null,
spec_obr varchar(30),
date_obr date not null,
numdip_obr numeric(15)
);

```

5.2. Создание представлений (готовых запросов)

1. Данные о участниках отдела, где работает сотрудник:

```

create or replace view my_kollegi( Familiya, Name, Dolgnost, Tel)
as
    select e1.fname_e, e1.lname_e, e1.dolgn_e, e1.phone_e
    from employees e1
    where exists (select *
                  from employees e
                  where e.otdel_e = e1.otdel_e and e.login_e =
user());

```

2. Данные о других начальниках отделов, для начальника отдела:

```

create or replace view other_nach( Num_otdel, Familiya, Name, Tel)
as
    select e1.otdel_e, e1.fname_e, e1.lname_e,
e1.phone_e/*исключить некоторые поля*/
    from employees e1
    where exists (select *
                  from employees e
                  where e.otdel_e <> e1.otdel_e and e.login_e = user()
                  and e.dolgn_e = e1.dolgn_e);

```

3. Поставщики, от которых не было поставок фруктов:

```

create or replace view no_postavok(ID, NAME_Provider) as
    select p.id_p, p.name_p from providers p

```

```

where not exists (select * from postavkifruits pf
                  where p.id_p = pf.postavshik_pf);

```

4. Продукция, в состав которой входит несколько фруктов и их количество:

```

create or replace view no_single_prod(ID, NAME_Postavki, kol_vo)
as
select prod.id_prod, prod.name_prod, (select count(*)
                                     from sostav s
                                     where prod.id_prod = s.prod_s) as cnt
from production prod
having cnt > 1;

```

5. Количество продукции по категории и названию:

```

create or replace view cnt_production(kateg, name, kol_vo) as
select o.spec_o, prod.name_prod, count(*)
from production prod, otdels o
where prod.otd_prod = o.id_o
group by o.spec_o, prod.name_prod;

```

6. Статистика по странам стоимости поставок фруктов:

```

create or replace view sum_postavki(country, cost) as
select pf.country_pf, sum(pf.price_pf)
from postavkifruits pf
group by pf.country_pf;

```

7. Поставщики, поставляющие фрукты из разных стран:

```

create or replace view many_countries(ID, Name, Phone,
kolvo_country) as
select p.id_p, p.name_p, p.phone_p,
       (select count(distinct pf.country_pf)
        from postavkifruits pf
        where p.id_p = pf.postavshik_pf) as cnt_country
from providers p
having cnt_country > 1;

```

8. Сколько дней проходит с момента поставки фруктов до производства продукции:

```

create or replace view dneysr (ID_prod, avg_day) as
select prod.id_prod, avg(datediff(prod.date_prod, spf.d))
from production prod, (select
                        s.prod_s as pr, pf.datepost_pf as d
                        from sostav s, postavkifruits pf
                        where
                        s.fruct_s = pf.id_pf
                        ) as spf
where spf.pr = prod.id_prod
group by prod.id_prod;

```

9. Самые дешёвые поставщики за 1 кг одной и той же поставки фруктов:

```

create or replace view cheaper_postavki (id_postav, name,
postavshik, min_price) as
    select pf0.id_pf, pf0.name_pf, pf0.postavshik_pf, mn.min_pf
    from postavkifruits pf0, (
        select pf.name_pf as name, min(pf.price_pf /
to_kg(pf.weigth_pf, pf.edizm_pf)) as min_pf
        from postavkifruits pf
        group by pf.name_pf) as mn
    where mn.name = pf0.name_pf
    and pf0.price_pf / to_kg(pf0.weigth_pf, pf0.edizm_pf) -
mn.min_pf <= 0.000001;

```

10. Остаток поставок (неиспользуемые для производства поставки):

```

create or replace view ostatki_postavok(ID, Name, Weight) as
    select p0.id_pf, p0.name_pf, to_kg(p0.weigth_pf, p0.edizm_pf)
    from postavkifruits p0
    where not exists (
        select *
        from sostav s
        where s.fruct_s = p0.id_pf
    )
    union
        select p0.id_pf, p0.name_pf, to_kg(p0.weigth_pf,
p0.edizm_pf) - p1.sum_
        from postavkifruits p0,
        (select p.id_pf as id_pf, sum(to_kg(s.mas_s,
s.edizm_s)) as sum_
        from sostav s, postavkifruits p
        where s.fruct_s = p.id_pf
        group by p.id_pf
        ) as p1
    where p0.id_pf = p1.id_pf
    and p1.sum_ < to_kg(p0.weigth_pf, p0.edizm_pf);

```

Примечание: в представлениях 9-10 используется функция to_kg.

Права доступа к представлениям:

Представления	Группы пользователей (роли)						
	Руководители орган.	Руководители отделов с производ-ом	Менеджеры производ-ва	Рабочие отдела с производ-ом	Сотрудники отд. кадров	Бухгалтеры	Отдел закупок
Данные об участниках отдела, где работает сотрудник My_kollegi	S	S	S	S	S	S	S
Данные о других начальниках отделов, для начальника отдела Other_nach		S					
Поставщики, от которых не было поставок фруктов no_postavok	S		S				
Продукция, в состав которой входит несколько фруктов и их кол-во no_single_prod	S	S	S				
Количество продукции по категории и названию cnt_production	S	S	S				
Статистика по странам стоимости поставок фруктов sum_postavki	S	S	S				S
Поставщики, поставляющие фрукты из разных стран many_countries	S		S			S	S
Сколько дней проходит с момента поставки фруктов до производства продукции dneysr	S		S				
Самые дешевые поставщики за 1 кг одной и той же поставки фруктов cheaper_postavki	S	S	S				S
Остаток поставок ostatki_postavok	S	S	S		S	S	S

5.3. Создание функции

Функция перевода из одних единиц измерения в кг:

```
delimiter //
```

```
create function to_kg (weight_ numeric(10,6), ed varchar(20))
```

```
returns numeric(10,6)
```

```
begin
```

```
    declare perem numeric(10,6) ;
```

```
    set perem = -1;
```

```
    select edizm.kg_ed into perem from edizm where ed =
```

```
edizm.name_ed;
```

```

        if (perem = -1) then
            signal sqlstate '45000' set
            message_text = 'нет такой ед.изм';
        end if;
    return perem*weight_;
end; //
delimiter ;

```

5.4. Создание процедуры

Создание этикетки для всех продуктов с указанием даты по текущую дату, названия, состава, массы брутто, массы нетто:

```

delimiter //
create procedure etiketka(d date)
begin
    declare mas numeric(10,6) default 0;
    declare cnt numeric(5) default 0;
    declare kat varchar(20);
    declare etik text;
    declare l_eof bool default false;
    declare otd_prod1 numeric(3);
    declare name_prod1 varchar(50);
    declare id_prod1 numeric(3);
    declare mas_prod1 numeric(10,6);
    declare date_prod1 date;
    declare edizm_prod1 varchar(20);
    declare crs1 cursor for select otd_prod, name_prod, id_prod,
mas_prod, date_prod, edizm_prod
                                from production
                                where date_prod between d and now();
    declare continue handler for not found set    l_eof = true;
    open crs1;
    l1: loop
        set etik = '';
        fetch crs1 into otd_prod1, name_prod1, id_prod1,
mas_prod1, date_prod1, edizm_prod1;
        select o.spec_o into kat from otdels o where o.id_o =
otd_prod1;
        set etik = concat_ws(' ',etik,'==Etiketka== ', kat ,
name_prod1);
        set etik = concat_ws('\n',etik, '==Sostav:==');
        set mas = 0;
        if l_eof then
            leave l1;
        end if;
    end if;
end;

```

```

        set cnt = cnt + 1;
    begin
        declare weigh_pfl numeric(10,6);
        declare edizm_sl varchar(20);
        declare name_pfl varchar(30);
        declare crs2 cursor for select weigh_pf,
edizm_s, name_pf
            from sostav s, postavkifruits pf
            where id_prod1 = s.prod_s and pf.id_pf =
s.fruct_s;

        open crs2;
        l2: loop
            fetch crs2 into weigh_pfl, edizm_sl,
name_pfl;

            if l_eof then

                leave l2;
            end if;
            set mas = mas + to_kg(weigh_pfl,
edizm_sl);

            set etik = concat_ws('\n',etik,
name_pfl);

        end loop l2;
        close crs2;
        set l_eof = false;
    end;
    set mas_prod1 = to_kg(mas_prod1,edizm_prod1);
    set etik = concat_ws('\n',etik, 'Massa netto kg:',
mas_prod1 - mas);
    set etik = concat_ws('\n',etik, 'Massa brytto
kg:',mas_prod1);
    set etik = concat_ws('\n',etik, 'Goden
do:',date_prod1);
    select etik;
end loop l1;
set etik = concat_ws('_', 'After date: ', d, 'have', cnt,
'production');
select etik;
close crs1;
end; //
delimiter ;

```

5.5. Создание триггеров

1. Проверка наличия уже такого состава на тот же продукт и поставку:

```
delimiter //
create trigger sost
    before
    insert
    on sostav
    for each row
begin
    DECLARE    cnt1 numeric;
    DECLARE    cnt2 numeric;
    DECLARE    cnt3 numeric;
    select count(*) into cnt1
    from postavkifruits as pf
    where new.fruct_s = pf.id_pf;
    select count(*) into cnt2
    from production as p
    where new.prod_s = p.id_prod;
    select count(*) into cnt3
    from sostav as s
    where new.fruct_s = s.fruct_s and new.prod_s = s.prod_s;
    if not(cnt1 > 0 and cnt2 > 0 and cnt3 = 0) then
        signal sqlstate '45000' set
        message_text = 'такая запись уже есть (обновляйте значение
массы)';
    end if;
end; //
delimiter ;
```

2. Проверка корректности ввода данных работников:

```
delimiter //
create trigger proverka before insert on employees
for each row
begin
    if((year(curdate()) - year(new.born_e) -
    (date_format(curdate(), '00-%m-%d') <
date_format(new.born_e, '00-%m-%d')) < 16) then
        signal sqlstate '45000' set
        message_text = 'Человеку меньше 16 лет';
    end if;
    if ((year(new.date_e) - year(new.born_e) -
    (date_format(new.date_e, '00-%m-%d') <
date_format(new.born_e, '00-%m-%d')) < 14) then
        signal sqlstate '45000' set
```

```

        message_text = 'Неверная дата выдачи паспорта';
    end if;
    set new.sex_e := lower(new.sex_e);
    set new.fname_e := upper(new.fname_e);
    set new.lname_e := upper(new.lname_e);
end; //
delimiter ;

```

3. Проверка корректности ввода данных поставок фруктов:

```

delimiter //
create trigger prov_postavki before insert on postavkifruits
for each row
begin
    if((datesbor_pf >= now()) or (srok_pf < datesbor_pf) or
(datepost_pf <= datesbor_pf and datepost_pf >= now())) then
        signal sqlstate '45000' set
        message_text = 'Неверно введена дата';
    end if;
    if(weighth_pf <= 0) then
        signal sqlstate '45000' set
        message_text = 'Неверно введен вес';
    end if;
    if(price_pf < 0) then
        signal sqlstate '45000' set
        message_text = 'Неверно введена цена';
    end if;
end; //
delimiter ;

```

4. Проверка корректности ввода данных единиц измерения:

```

delimiter //
create trigger prov_edizm before insert on edizm
for each row
begin
    if(kg_ed <= 0) then
        signal sqlstate '45000' set
        message_text = 'Неверно введен коэффициент перевода в кг';
    end if;
end; //
delimiter ;

```

5. Проверка корректности ввода данных состава:

```

delimiter //
create trigger prov_sostav before insert on sostav
for each row
begin

```



```

        if(mas_s <= 0) then
            signal sqlstate '45000' set
            message_text = 'Неверно введена масса';
        end if;
    end; //
delimiter ;

```

6. Проверка корректности ввода данных типы упаковки:

```

delimiter //
create trigger prov_typeupak before insert on typeupak
for each row
begin
    if(weigh_t <= 0) then
        signal sqlstate '45000' set
        message_text = 'Неверно введен вес';
    end if;
end; //
delimiter ;

```

7. Проверка корректности ввода данных продукции:

```

delimiter //
create trigger prov_product before insert on production
for each row
begin
    if((date_prod > now()) or (srok_prod <= date_prod)) then
        signal sqlstate '45000' set
        message_text = 'Неверно введена дата';
    end if;
    if(mas_prod <= 0) then
        signal sqlstate '45000' set
        message_text = 'Неверно введена масса';
    end if;
end; //
delimiter ;

```

8. Проверка корректности ввода данных должностей:

```

delimiter //
create trigger prov_dolgnost before insert on dolgnosti
for each row
begin
    if(sal_dol<6250) then
        signal sqlstate '45000' set
        message_text = 'Неверно введена заработная плата';
    end if;
end; //
delimiter ;

```

5.6. Создание индексов

Для повышения эффективности работы с данными нами не создавались индексы, кроме автоматических индексов первичных ключей и уникальных полей таблиц. Другие индексы создаются администратором базы данных после анализа запросов пользователей.

5.7. Разработка стратегии резервного копирования

Интенсивность обновления разработанной базы данных производства фруктовой продукции довольно низкая, поэтому для обеспечения сохранности вполне достаточно проводить полное резервное копирование базы данных раз в день. Для разработанной базы данных необходимо держать сервер включенным круглосуточно, так как менеджеры производства должны иметь круглосуточный доступ к базе данных, ведь производство идёт без остановки.

6. Список литературы

1. Карпова И.П. «Проектирование реляционных баз данных: Метод. указания к курсовому проектированию по курсу «Базы данных». – М., МИЭМ, 2010.
2. Карпова И.П. «Учебное пособие по курсу «Базы данных». – М., РИО МГИЭМ, 2009.
3. Справочное руководство по MySQL, издание 306, 2010.