

**МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА
И ПРОДОВОЛЬСТВИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ
ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

Кафедра автоматизированных
систем управления производством

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к лабораторным работам по курсу

**"АВТОМАТИЗИРОВАННЫЕ СИСТЕМЫ
УПРАВЛЕНИЯ ПРЕДПРИЯТИЕМ" (АСУП)**

часть II

для студентов специальности:

25.01.07. "Экономика и управление на предприятии",
специализации 25.01.07.02 эи "Экономическая информатика"

М И Н С К, 2003г.

Издание второе, дополненное и переработанное.

Методические указания к лабораторным работам по курсу "Автоматизированные системы управления предприятием" для студентов специальности "Экономическая информатика" рассмотрены на заседании методической комиссии агроэнергетического факультета и рекомендованы к изданию на ротапинтере БАТУ.

Протокол № 6 от 20 декабря 2002г.

Составители : к.т.н., доцент Гируцкий И.И.;
ассистент Музыченко Е.Н.,
ассистент Гагаков Ю.В.

Содержание

Лабораторная работа №1 Применение систем управления базами данных при автоматизации производства	3
Лабораторная работа №2 Построение и анализ математической модели по обоснованию и информатизации производства молока	8
Лабораторная работа №3 Построение и анализ статистической модели откорма свиней.....	16
Лабораторная работа №4 Анализ производства птицы по двухмерной модели.....	23
Приложение 1. Реляционные базы данных.....	29
1. Базовые понятия реляционных баз данных	29
2. Фундаментальные свойства отношений	31
3. Отношения между таблицами.	32
4. Реляционная модель данных	33
5. Проектирование реляционных БД.....	36
6. Структура реляционных баз данных	40
7. Таблицы	41
8. Запросы	46
9. Построитель выражений.....	57
10. Создание запросов других типов.....	59
11. SQL-запросы.....	61
ПРИЛОЖЕНИЕ 2. Молочно-товарная ферма	63
1. Математическая модель продуктивности коров.	63
2. Оптимизация производства молока	65
3. Способы кормления животных	66
ПРИЛОЖЕНИЕ 3. Откорм свиней.....	67
1. Математическая модель откорма свиней.....	67
2. Критерий оптимальности производства свинины.....	68

3. Способы кормления	69
Приложение 4. Производство птицы	70
1. Особенность производства птицы.....	70
2. Микроклимат птичника	73
3. Параметры микроклимата.....	73
4. Математическая модель продуктивности и производства птицы	77

Лабораторная работа №1
ПРИМЕНЕНИЕ СИСТЕМ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ ПРИ
АВТОМАТИЗАЦИИ ПРОИЗВОДСТВА

1. Цель работы: Изучение основных принципов работы в СУБД Microsoft Access. Создать базу данных сотрудников. Научиться создавать запросы для анализа и обработки информации.

2. Краткие теоретические сведения

2.1. АСУ предприятия не обходиться без СУБД, которая является основой АСУ. Данные, хранимые в БД предприятия, так же имеют ценность, как и продукция предприятия. Поэтому можно сказать, что СУБД современного предприятия является его неотъемлемой частью.

2.2. Современная СУБД предприятия, построенная по новым информационным технологиям, позволяет служащим получать оперативно производственно необходимую информацию. Кроме оперативности информация должна быть правильной, по возможности краткой и информативной. Таких показателей качества информации можно добиться, используя реляционные БД и язык структурированных запросов (SQL).

3. Программа работы

3.1. Ознакомиться с теоретическими сведениями (Приложение 1).

3.2. Ознакомитесь с принципами создания реляционных баз данных (конспект лекций) и создание SQL запросов Microsoft Access (приложение 2).

Создайте базу данных сотрудников. В имеющийся шаблон базы данных добавьте объект "Сотрудники" и установите отношения в базе данных между объектами.

3.2.1. Открыть в MS Access базу данных "Лабораторная №1". Добавить в базу данных таблицу "Служащий", наименование и тип полей которой представлены в табл.1.

Таблица.1. Объект "Служащий "

Наименование поля	Тип данных	Размер
Код работника	Счетчик	Длинное целое
ФИО	Текстовый	50
Дата рождения	Дата/время	Краткий формат даты
Код пола	Числовой	Длинное целое
Код семейного положения	Числовой	Длинное целое
Стаж работы	Числовой	Длинное целое
Код разряда	Числовой	Длинное целое
Код наличия детей	Числовой	Длинное целое

Как сделать. Выберите объекты базы данных – таблицы. Создайте таблицу в режиме конструктора. Ввести поля таблицы и установить тип данных полей согласно табл.1. (Размер поля нужно указать в свойствах поля, а не в описании). Для обеспечения целостности информации поля [ФИО], [Дата рождения] и [Стаж работы] сделайте обязательными. Поле [Код работника] установите ключевым. Сохраните таблицу под именем "Служащие".

3.2.2. В базе данных кроме созданного Вами объекта "Служащие" имеются объекты "Наличие детей", "Пол", Разряд" и "Семейное положение", которые позволяют описать объект "Служащие". Установите связи между таблицей "Служащий" и таблицами "Наличие детей", "Пол", Разряд" и "Семейное положение" используя, мастер подстановок в конструкторе таблицы или схему базы данных (рис.1.).

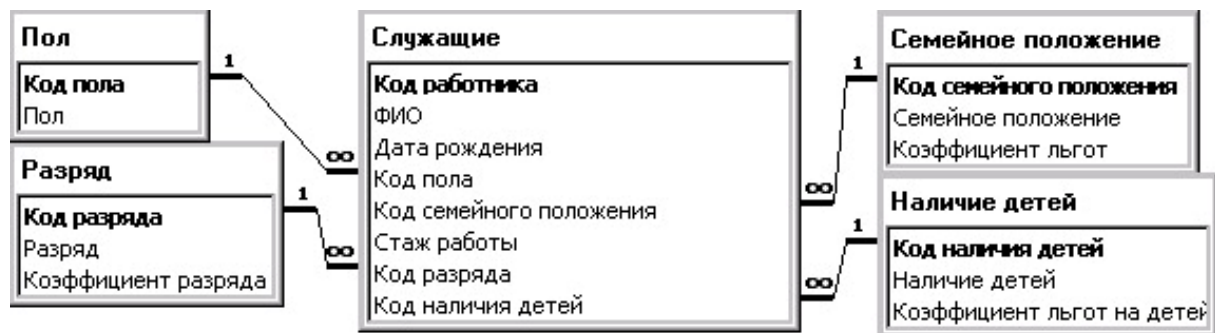


Рис.1. Схема базы данных

Как сделать. Откройте таблицу "Служащие" с помощью конструктора таблиц. В типе данных поля [Код пола] выберите тип – мастер подстановок. Выберите, что значения будут использоваться из таблицы. Нажать кнопку "Далее". Выберите для подстановки данных таблицу "Пол". Нажать кнопку "Далее". Из доступных полей выбрать поле [Пол], соответствующее описанию объекта "Служащие". Переведите в выбранные поля кнопкой ">". Нажмите кнопку "Далее" два раза и кнопку готово. Повторите такую операцию с полями [Код семейного по-

ложения], [Код разряда], [Код наличия детей] таблицы "Служащие" выбрав для подстановки соответствующие объекты. Откройте схему данных (появится изображение схемы данных похожее на рис.1.). Если не видны все пять таблиц нажмите кнопки "Отобразить прямые связи" и "Отобразить все связи". Если опять нет пяти таблиц то Вы не установили с помощью мастера подстановок связи, тогда добавьте недостающие таблицы в схему данных и установите связи как на рис.1. Измените параметры связей. Для этого правой кнопкой мыши щелкните по связи, выберите изменить связь и установите птичку "Обеспечение целостности данных". Это нужно для того чтобы при вводе нового служащего были обязательно введены пол, семейное положение, стаж работы и наличие детей, иначе служащий не будет введен.

3.2.3. Занесите десять записей в таблицу "Служащие" по данным, которые дает преподаватель.

Как сделать. Откройте таблицу "Служащие" в режиме просмотра и редактирования, и внесите десять записей.

3.3. Создать запросы на получения информации.

3.3.1. Создайте запрос "Все данные по служащим". В запросе отобразите все данные по служащим (ФИО, дата рождения, пол, семейное положение, стаж работы, разряд, наличие детей). Установить в запросе сортировку по полю [ФИО].

Как сделать. Выберите объекты базы данных – запросы. Создайте запрос в режиме конструктора. В диалоговом окне выберите все таблицы (Кнопки "Shift" и "↓") для добавления. Нажмите кнопки "Добавить" и "Закреть". Перетащите поля [ФИО], [Дата рождения], [Стаж работы] из таблицы "Служащие" в поля запроса, а поля [Пол], [Семейное положение], [Разряд], [Наличие детей] из таблиц "Пол", "Разряд", "Семейное положение", "Наличие детей". Для лучшего просмотра в поле [ФИО] запроса в строке сортировка установить сортировку "по возрастанию". Сохраните запрос под именем "Все данные по служащим". Проверьте результат, запустив запрос (переключите вид в режим таблицы или кнопку "Запуск")

3.3.2. Создайте запрос "Все данные по служащим со стажем работы больше 10 лет".

Как сделать. Чтобы быстрее сделать работу, скопируйте запрос "Все данные по служащим" в буфер обмена (сделать активным запрос и нажать кнопку "Копировать") и вставьте из буфера под именем. Откройте запрос "Все данные по служащим со стажем работы больше 10 лет" с помощью конструктора запросов. В

строке по полю запроса [Стаж работы] ввести условие отбора >10. Проверьте результат, запустив запрос.

3.4. Создайте запрос на обновление стажа служащих.

Как сделать. Чтобы быстрее сделать работу, скопируйте запрос "Все данные по служащим" в буфер обмена и вставьте из буфера под именем "Изменение стажа". Откройте запрос "Изменение стажа" с помощью конструктора запросов. Установите тип запроса на обновление. В строке "Обновить:" по полю "Стаж работы" запишите $=[\text{Стаж работы}]+1$. Это значит что при запуске запроса стаж работников увеличится на 1 год. Сохраните и закройте запрос. Запустите сначала запрос "Все данные по служащим со стажем работы больше 10 лет" и посмотрите, какой стаж работников. Закройте запрос. Запустите запрос "Изменение стажа". Повторно запустите запрос и проверьте изменение стажа служащих.

3.5. Создать запрос на вычисление зарплаты.

3.5.1. Создать запрос с расчетом зарплаты.

Как сделать. Чтобы быстрее сделать работу, скопируйте запрос "Все данные по служащим" в буфер обмена и вставьте из буфера под именем "Зарплата служащих". Откройте запрос "Зарплата служащих" с помощью конструктора запросов. В конструкторе запросов к запросу добавьте поля [Коэффициент разряда], [Коэффициент льгот], [Коэффициент льгот на детей]. Сохраните запрос, чтобы изменения вступили в силу и поля с коэффициентами появились в построителе выражений. В свободном поле запроса введите выражение **Зарплата: $100 * [\text{Коэффициент разряда}] + 50 * [\text{Коэффициент льгот на детей}] * [\text{Коэффициент льгот}]$** . Выражение можно ввести вручную или использовать построитель выражений для удобства. Сохраните запрос. Уберите птички в строке "Вывод на экран" на полях [Коэффициент разряда], [Коэффициент льгот], [Коэффициент льгот на детей] запроса – эти поля не нужны для просмотра. Правой кнопкой мыши щелкните по полю зарплата выберите "Свойства", затем "Формат поля" и установите денежный формат. Это для лучшего отображения. Сохраните и запустите запрос.

3.5.2. Создайте запрос для подсчета суммы зарплаты.

Как сделать. Создайте запрос с помощью конструктора запросов и добавьте запрос "Зарплата служащих". Перенесите в запрос поле [Зарплата]. Включите групповые операции. По полю зарплата в строке "Групповые операции" включите вычисление суммы (Вместо группировки выберите Sum). Сохраните запрос под именем "Итого по зарплате". Запустите и закройте зарос.

4. Содержание отчета

- 4.1. Название и цель выполнения работы.
- 4.2. Вывод о роли баз данных в АСУП.

5. Контрольные вопросы

- 5.1. Сущность реляционных баз данных (РБД).
- 5.2. Из чего состоят базы данных, и в каком виде храниться информация?
- 5.3. Из чего состоят таблицы, и по каким принципам они создаются?
- 5.4. Отношение один ко многим между таблицами. Рассказать на примере.
- 5.5. Отношение многие ко многим между таблицами. Рассказать на примере.
- 5.6. Типы данных хранимых в таблицах, их значение для правильного построения базы.
- 5.7. Нормализация. Примеры нормализации в лабораторной работе.
- 5.8. Что такое SQL?
- 5.9. Значение SQL запросов в СУБД.
- 5.10. Типы запросов их применение.
- 5.11. Математические возможности Access.
- 5.12. Значение СУБД в организации управления предприятием.

6. Литература

"Access 97", Виктор Пасько – К.: Издательская группа ВАУ 1997г.

Лабораторная работа №2
ПОСТРОЕНИЕ И АНАЛИЗ МАТЕМАТИЧЕСКОЙ МОДЕЛИ ПО
ОБОСНОВАНИЮ И ИНФОРМАТИЗАЦИИ ПРОИЗВОДСТВА МОЛОКА

1. Цель работы: построение и анализ экономико-математической модели доения молочных коров при групповом и индивидуальном дозировании корма.

2. Краткие теоретические сведения

2.1. В нашей стране в основном применяется групповое кормление. Это когда группе животных выдается общая доза корма, и в среднем все животные получают одинаковое количество корма. Такой способ кормления является простым и не требует применения сложной техники. А в развитых странах применяется индивидуальное кормление. Для применения индивидуального кормления необходима система идентификации животных, индивидуального учета надоенного молока в сутки и специальных кормушек. Все это приводит к большим, по сравнению с групповым способом кормления, эксплуатационным затратам, но снижает потери кормов.

3. Программа работы

3.1. Ознакомиться с теоретическими сведениями (Приложение 2).

3.2. Разбиение стада коров на группы. Для этого необходимо открыть базу данных "Лабораторная работа №2". Исходными данными служат таблицы "Продуктивность" и "Цены". В таблице "Продуктивность" даны потенциальные годовые удои по 1000 коровам.

3.2.1. Стадо коров лучше разбить по продуктивности коров – объединив коров с близкой продуктивностью в отдельную группу. Тысячу коров в стаде необходимо разбить на пять групп с одинаковым количеством голов. Информацией о принадлежности коровы в группе является номер группы по умолчанию равный единице. Необходимо разбить стадо коров.

Как сделать. Создайте с помощью конструктора запросов запрос, добавьте в запрос таблицы "Продуктивность коров". В поле запроса перенесите поля [Продуктивность], [Номер группы]. Измените тип запроса на обновление. В строке условие отбора по полю продуктивность поставьте условие отбора первой группы (200 голов). Например:

>0 And <=2405

Переключите вид запроса в режим таблицы. Если запрос выдаст 200 коров, то условие отбора правильное. Если коров больше – необходимо уменьшить правую границу диапазона продуктивности. Если коров меньше 200 необходимо увеличить правую границу диапазона. Когда Вы подберете (переключая вид запроса и изменяя правую границу) необходимое количество коров в группе, переключите запрос в режим конструктора, по полю [Номер группы] в строке "Обновление:" поставьте 1. Запустите запрос (кнопка красный восклицательный знак). На вопрос что будет обновлено 200 записей ответе "Да". Первая группа определена. Для создания следующей группы измените диапазон отбора коров. На место левой границы поставьте правую, а на место правой число соответствующее еще большей продуктивности. Например:

2405> and <=3300

Проверьте количество животных, в новой группе переключив вид запроса в режиме таблица. Когда Вы подберете необходимое количество коров во второй группе, переключите запрос в режим конструктора, по полю [Номер группы] в строке "Обновление:" поставьте 2 и запустите запрос (кнопка красный восклицательный знак). Вторая группа определена. Таким же образом создайте остальные 3 группы. Сохраните запрос под именем "Разбиение на группы".

3.2.2. Исследование эффективности индивидуального и группового кормления по двум критериям оптимальности.

Для исследования группового кормления необходимо рассчитать среднюю продуктивность по группе.

Как сделать. Создайте с помощью конструктора запросов запрос, добавьте в запрос таблицы "Продуктивность коров". В поле запроса перенесите поля [Продуктивность], [Номер группы]. В поле [Продуктивность] введите псевдоним:

Уд пот ср: Продуктивность

Включите групповые операции. По полю [Продуктивность] в строке "Групповые операции" поставьте Avg – вычисление среднего, а по полю [Номер группы] – Группировка. Это значит, что будет подсчитана средняя продуктивность коров по каждой группе. Сохраните запрос под именем "Средняя продуктивность коров".

3.2.3. Для исследования индивидуального и группового кормления необходимо рассчитать параметры модели удоя коров, экономические показатели для каждой коровы. Исследование проводится по двум критериям оптимальности.

Как сделать. Создайте с помощью конструктора запрос, добавьте в запрос таблицы "Продуктивность", "Цены", "Средняя продуктивность коров". Объедините поля таблицы "Продуктивность" и запроса "Средняя продуктивность коров". Для этого перенесите поле [Номер группы] таблицы "Продуктивность" на поле [Номер группы] запроса "Средняя продуктивность коров". В поле запроса перенесите поля [Код коровы], [Продуктивность], [Уд пот ср], [Цена молока] и [Цена корма]. Сохраните запрос под именем "Данные по стаду".

3.2.4. Определение $D_{з\text{пот}}$ потенциальной дозы - параметра модели производится по формуле 2 приложения 2. Из формулы выражается $D_{з\text{пот}}$, $D_{з\text{под}}$ принимается равной 1400 к. ед. в год.

$$D_{з\text{пот}} = \frac{U_d}{1,875} + 1400 \quad (1)$$

Как сделать. В свободном поле запроса "Данные по стаду" введите формулу вычисления $D_{з\text{пот}}$ потенциальной дозы:

$$D_{з\text{ пот}}:[Продуктивность]/1,875+1400$$

Примечание. Знак дробной части числа устанавливается в системных настройках. В основном это запятая, но может быть точка. При несоответствии знака дробной части MS Access выдаст сообщение об ошибке.

После набора выражения в конструкторе нужно нажать кнопку "Ок" и сохранить запрос – это избавит от пустых строк в вашем выражении и при вводе новых выражений в конструкторе будут доступны поля введенных Вами выражений.

3.2.5. Средняя потенциальная доза по группе рассчитывается по той же формуле (1), но берется среднее значение удоя.

Как сделать. В свободном поле запроса "Данные по стаду" введите формулу вычисления $D_{з\text{пот ср}}$ средней потенциальной дозы:

$$D_{з\text{ пот ср}}:[Уд пот ср]/1,875+1400$$

3.2.6. Определение $D_{з\text{опт 1 инд}}$ оптимальной дозы по критерию максимум удоя при минимуме затрат корма для индивидуального кормления. Откуда берется эта оптимальная доза и как вычисляется (6), рассказано в приложении 2.

Как сделать. В запросе "Данные по стаду" в новом поле введите формулу вычисления $D_{з\text{опт 1 инд}}$ оптимальной дозы по критерию максимум удоя при минимуме затрат корма при индивидуальном кормлении:

$$D_{з\text{ опт 1 инд}}:\text{sq}([\text{Дз пот}]^2-0,75*([\text{Дз пот}]-1400)^2)$$

3.2.7. Определение $D_{з\text{опт 1 групп}}$ оптимальной дозы по критерию максимум удоя при минимуме затрат корма для группового кормления.

Как сделать. В запросе "Данные по стаду" в новом поле введите формулу вычисления $D_{\text{ОПТ 1 групп}}$ оптимальной дозы по критерию максимум удоя при минимуме затрат корма при индивидуальном кормлении:

$$D_{\text{з опт 1 групп}}: \text{sqrt}([D_{\text{з пот ср}}]^2 - 0,75 * ([D_{\text{з пот ср}}] - 1400)^2)$$

3.2.8. Определение $D_{\text{ОПТ 2 инд}}$ оптимальной дозы по критерию максимум прибыли для индивидуального кормления. Откуда берется эта оптимальная доза и как вычисляется (8), рассказано в приложении 2.

Как сделать. В запросе "Данные по стаду" в новом поле введите формулу вычисления $D_{\text{ОПТ 2 инд}}$ оптимальной дозы по критерию максимум прибыли при индивидуальном кормлении:

$$D_{\text{з опт 2 инд}}: [D_{\text{з пот}}] - 0,5 * [\text{Цена корма}] / [\text{Цена молока}] * ([D_{\text{з пот}}] - 1400)$$

3.2.9. Определение $D_{\text{ОПТ 2 групп}}$ оптимальной дозы по критерию максимум прибыли для группового кормления.

Как сделать. В запросе "Данные по стаду" в новом поле введите формулу вычисления $D_{\text{ОПТ 2 групп}}$ оптимальной дозы по критерию максимум прибыли при индивидуальном кормлении:

$$D_{\text{з опт 2 групп}}: [D_{\text{з пот ср}}] - 0,5 * [\text{Цена корма}] / [\text{Цена молока}] * ([D_{\text{з пот ср}}] - 1400)$$

3.2.10. Определение оптимального удоя по критерию максимум удоя при минимуме затрат корма. Оптимальный удой получается, когда корове дают оптимальную дозу. Зависимость удоя от дозы приведена в (1) приложении 2. Для расчета берётся участок с параболической зависимостью (третья строка). Коэффициент K_2 определяется по формуле 4 приложения 2.

$$U_{\text{д ОПТ}} = U_{\text{д ПОТ}} - \frac{2,5}{D_{\text{з ПОТ}} - D_{\text{з ПОД}}} \cdot (D_{\text{з ПОТ}} - D_{\text{з ОПТ}})^2 \quad (2)$$

Для индивидуального кормления значения в (2) берутся индивидуальные, а для группового кормления – средние.

Как сделать. В запросе "Данные по стаду" в новом поле введите формулу вычисления $U_{\text{д ОПТ 1 инд}}$ оптимального удоя (2) по критерию максимум удоя при минимуме затрат корма при индивидуальном кормлении:

$$U_{\text{д опт 1 инд}}: ([\text{Продуктивность}] - 2,5 / ([D_{\text{з пот}}] - 1400)) * ([D_{\text{з пот}}] - [D_{\text{з опт 1 инд}}])^2$$

В новом поле запроса введите формулу вычисления Уд опт 1 групп оптимального удоя по критерию максимум удоя при минимуме затрат корма при групповом кормлении:

$$\text{Уд опт 1 групп:}([\text{Уд пот ср}]-2,5/([\text{Дз пот ср}]-1400)*([\text{Дз пот ср}]-[\text{Дз опт 1 групп}])^2)$$

В новом поле введите формулу вычисления Уд опт 2 инд оптимального удоя по критерию максимум прибыли при индивидуальном кормлении:

$$\text{Уд опт 2 инд:}([\text{Продуктивность}]-2,5/([\text{Дз пот}]-1400)*([\text{Дз пот}]-[\text{Дз опт 2 инд}])^2)$$

В новом поле запроса введите формулу вычисления Уд опт 2 групп оптимального удоя по критерию максимум прибыли при групповом кормлении:

$$\text{Уд опт 2 групп:}([\text{Уд пот ср}]-2,5/([\text{Дз пот ср}]-1400)*([\text{Дз пот ср}]-[\text{Дз опт 2 групп}])^2)$$

3.2.11. Определение прибыли (П.2.(7)).

Как сделать. В запросе "Данные по стаду" в новом поле введите формулу вычисления прибыли по критерию максимум удоя при минимуме затрат корма при индивидуальном кормлении:

$$\text{Пр 1 инд:}[\text{Уд опт 1 инд}]*[\text{Цена молока}]-[\text{Дз опт 1 инд}]*[\text{Цена корма}]^2,5$$

В новом поле запроса введите формулу вычисления прибыли по критерию максимум удоя при минимуме затрат корма при групповом кормлении:

$$\text{Пр 1 групп:}[\text{Уд опт 1 групп}]*[\text{Цена молока}]-[\text{Дз опт 1 групп}]*[\text{Цена корма}]^2,5$$

В новом поле введите формулу вычисления прибыли по критерию максимум прибыли при индивидуальном кормлении:

$$\text{Пр 2 инд:}[\text{Уд опт 2 инд}]*[\text{Цена молока}]-[\text{Дз опт 2 инд}]*[\text{Цена корма}]^2,5$$

В новом поле запроса введите формулу вычисления прибыли по критерию максимум прибыли при групповом кормлении:

$$\text{Пр 2 групп:}[\text{Уд опт 2 групп}]*[\text{Цена молока}]-[\text{Дз опт 2 групп}]*[\text{Цена корма}]^2,5$$

Сохраните и закройте запрос.

3.3. Рассчитайте итоговые значения по стаду.

Как сделать. С помощью конструктора запросов создайте новый запрос, добавьте в него запрос "Данные по стаду", перенесите поля оптимальных доз, оптимальных удоев, прибылей, сохраните запрос с именем "Итого по стаду". Включите групповые операции. В строке запроса "Групповая операция" по полям оптимальных доз, оптимальных удоев и прибыли поставьте вместо "Группировка" функцию вычисления суммы Sum. Сохраните запрос. В новых полях вычислите разности по полям с разными способами кормления по критерию максимум удоя при минимуме затрат корма

Разн по Дз опт 1:[Sum-Дз опт 1 инд]-[Sum-Дз опт 1 групп]

Разн по Уд опт 1:[Sum-Уд опт 1 инд]-[Sum-Уд опт 1 групп]

Разн по Пр опт 1:[Sum-Пр 1 инд]-[Sum-Пр 1 групп]

и по критерию максимум прибыли

Разн по Дз опт 2:[Sum-Дз опт 2 инд]-[Sum-Дз опт 2 групп]

Разн по Уд опт 2:[Sum-Уд опт 2 инд]-[Sum-Уд опт 2 групп]

Разн по Пр опт 2:[Sum-Пр 2 инд]-[Sum-Пр 2 групп]

В столбцах запроса с введенными выражениями в строке "Групповая операция" поставьте вместо "Группировка" – "Выражение". Сохраните и закройте запрос.

3.4. Рассчитать итоговые значения по группам.

Как сделать. Скопируйте запрос "Итого по стаду" и вставьте под именем "Итого по группам". Добавьте в запрос поле [Номер группы]. В строке "Групповая операция" оставьте "Группировка". Добавьте в запрос поле [Уд пот ср]. В строке "Групповая операция" оставьте "Группировка". Теперь итоговые значения будут группироваться по группам. Сохраните и закройте запрос.

3.5. Анализ результатов.

Как сделать. Запишите результаты расчетов в табл.1. с запросов "Итого по группам" и "Итого по стаду". Запишите ценны по корму и молоку. Сравните показатели, полученные по разным критериям и разным способам кормления. Проанализируйте изменение прибыли по группам. Выводы запишите в отчет.

Таблице.1. Результаты исследований.

№ Группы	Удой средний потенциальный по группе, л. в год	Тип кормления						Разность		
		индивидуальное			групповое			по кормам, кг. к.ед.	по удо-ям, л. в год	по при-бы-ли, тыс. руб.
		доза оптимальная, кг. к.ед.	удой оптимальный, л. в год	при-быль, тыс. руб.	доза оптимальная, кг. к.ед.	удой оптимальный, л. в год	при-быль, тыс. руб.			
1	2	3	4	5	6	7	8	9	10	11
По критерию максимум удоя при минимуме затрат корма										
1										
2										
3										
4										
5										
ито-го										
По критерию максимум прибыли										
1										
2										
3										
4										
5										
ито-го										

4. Содержание отчета

4.1. Название и цель выполнения работы.

4.2. Выводы и результаты расчетов.

5. Контрольные вопросы

5.1. Чем отличается критерий оптимальности по максимуму удоя от критерия по максимуму прибыли?

5.2. Объяснить сущность каждого критерия оптимальности?

5.3. Чем отличается групповое кормление от индивидуального по технологии кормления?

5.4. Достоинства и недостатки каждого вида кормления.

5.5. Почему по критерию максимум удоя удои выше, чем по критерию максимум прибыли?

- 5.6. Почему по критерию максимальной прибыли удой меньше, а прибыль больше, чем по критерию максимального удоя?
- 5.7. Объясните откуда "потери" при групповом кормлении?
- 5.8. Способ снижения потерь при групповом кормлении?
- 5.9. От чего зависит оптимальный удой по критерию максимум удоя при минимуме затрат корма?
- 5.10. От чего зависит оптимальный удой по критерию максимума прибыли?
- 5.11. Как влияет цена молока и корма на оптимальные удои по разным критериям оптимальности?
- 5.12. Как влияет потенциальный удой на оптимальный удой по разным критериям оптимальности?

6. Литература

- 6.1. "Access 97", Виктор Пасько – К.: Издательская группа ВАН 1997г.
- 6.2. Интенсификация и информатизация молочной отрасли в хозяйствах РБ (рекомендации). – Мн. БелНИИ ЭИ АПК, 1997 г., 78с.
- 6.3. Агрокомплекс Голландии М.: ВНИИ ТЭИ агропром. 1992 г., 136.
- 6.4. Тимошенко В.Н. Проблемы и перспективы интенсификации производства молока. НТИ и рынок №5, 1996 г.
- 6.5. Л. Фредериксен. Расчет зимнего корма для молочного скота. – М. новый агроном. 1927 г. 206 с.

Лабораторная работа №3 ПОСТРОЕНИЕ И АНАЛИЗ СТАТИСТИЧЕСКОЙ МОДЕЛИ ОТКОРМА СВИНЕЙ

1. Цель работы: Определить параметры модели. Построить и проанализировать модели откорма свиней на свиноферме. Исследовать прирост свиней в динамике.

2. Краткие теоретические сведения

2.1. Успех анализа процесса откорма свиней зависит от правильного содержания животных. Насколько "правильно" определяется адекватностью наших представлений о животном, о технологическом процессе реальным процессам. Наши представления выражаются в виде формализованной модели, связывающей выходные параметры с параметрами кормления и содержания животных. В качестве параметров мы будем рассматривать затраты корма на единицу привеса, продолжительность откорма, прибыль или рентабельность откорма. В основе модели лежит зависимость текущих суточных привесов животного от расхода кормов при определенных условиях содержания. Модель должна учитывать поддерживающую дозу корма и зону насыщения привесов.

3. Программа работы

3.1. Ознакомиться с теоретическими сведениями (Приложение 3).

3.2. По исходным данным определите параметры математической модели прироста свиней.

3.2.1. С помощью запросов вычислите коэффициент a модели откорма свиней (5) приложения 2, коэффициент k приравняйте двум (на практике коэффициенты подбираются).

Как сделать. Открыть базу данных "Лабораторная №3". Создать запрос в режиме конструктора запросов, добавить таблицу "Привесы". Перенести в запрос все поля таблицы "Привесы" (поле с символом "*"). Сохранить запрос под именем "40 день". Перенести в запрос поля [День], [Доза]. Поставить условие отбора по полю [День] =40. В свободном поле с помощью построителя введите выражение

$$\text{Пср:}([П1]+[П2]+[П3]+[П4]+[П5]+[П6]+[П7]+[П8]+[П9]+[П10]+[П11]+[П12]+[П13]+[П14]+[П15]+[П16]+[П17]+[П18]+[П19]+[П20]+[П21]+[П22]+[П23]+[П24]+[П25]).$$

Удалите поле [Привесы.*] в запросе. Сохраните и закройте запрос. Скопируйте запрос "40 день" в буфер обмена и вставьте под именем "120 день". Откройте

запрос "120 день" в режиме конструктора и изменить условие отбора по полю =120. Сохраните и закройте запрос. Создать запрос в режиме конструктора запросов, добавить запросы "40 день" и "120 день". Перенести в запрос поля [Доза], [Пср] из двух запросов. Сохраните запрос под именем "Коэффициент а". В свободном поле с помощью построителя введите выражение вычисления коэффициента *a* по формуле (5) приложение 2.

$$a: ([120 \text{ день}].[Пср]-[40 \text{ день}].[Пср])/([120 \text{ день}].[Доза]-[40 \text{ день}].[Доза])^2$$

Сохраните и закройте запрос. Параметры модели определены.

3.3. Выкормите три поросенка 100 дней. Первый поросенок будет кормиться автоматической системой с целью получения максимальных привесов. Второй поросенок будет кормиться автоматической системой с целью получения максимальных привесов при минимуме затрат корма. Третьего поросенка будет кормить работник, то есть ручным способом. Доза кормления при ручном способе будет определяться раз в десять дней и в течении десяти дней будет оставаться постоянной. Точность определения дозы при ручном кормлении соответствуем десятым килограмма.

3.3.1. Подготовьте запросы для вычисления и добавления новых значений.

Как сделать. С помощью конструктора создать таблицу с полями, представленными в табл.1. В этой таблице будут записываться результаты вычислений. Таблица нужна для хранения промежуточных значений, при вычислении рекуррентных значений привесов (среднесуточные привесы зависят от текущей массы).

Таблица.1. Поля таблицы "Привесы 1"

Имя поля	Тип данных	Размер поля
День	Числовой	Длинное целое
Доза А1	Числовой	Двойное с плавающей точкой *
Привес А1	Числовой	Двойное с плавающей точкой *
Масса А1	Числовой	Двойное с плавающей точкой *
Прибыль А1	Числовой	Двойное с плавающей точкой *
Доза А2	Числовой	Двойное с плавающей точкой *
Привес А2	Числовой	Двойное с плавающей точкой *
Масса А2	Числовой	Двойное с плавающей точкой *
Прибыль А2	Числовой	Двойное с плавающей точкой *
Доза Р	Числовой	Двойное с плавающей точкой *
Привес Р	Числовой	Двойное с плавающей точкой *
Масса Р	Числовой	Двойное с плавающей точкой *
Прибыль Р	Числовой	Двойное с плавающей точкой *

* - в Access 97 размер поля называется "С плавающей точкой (8 байт)"

Сохраните таблицу под именем "Привесы 1". Зайдите в таблицу и введите исходные данные представленные в табл.2. Для разных кормлений одинаковые исходные данные и "поросята" одинаковые.

Таблица.2. Исходные данные таблицы "Привесы 1"

Поля	Значение
День	1
Доза А1	2
Привес А1	0,2
Прибыль А1	0
Масса А1	40
Доза А2	2
Привес А2	0,2
Масса А2	40
Прибыль А2	0
Доза Р	2
Привес Р	0,2
Масса Р	40
Прибыль Р	0

Сохраните и закройте таблицу. Теперь необходимо создать четыре запроса для вычисления и добавления новых значений. Первый запрос определяет последний день кормления. Создайте запрос в режиме конструктора запросов, добавьте таблицу "Привесы 1". В свободном поле набрать выражение

Последний День: Max([День]).

Сохраните запрос под именем "Последний день" и закройте. Второй запрос вычисляет параметры модели среднесуточных привесов: доза поддерживающая, доза потенциальная и потенциальный привес. Создайте с помощью конструктора запросов запрос, добавьте таблицу "Привесы 1" и запрос "Последний день". Установите связь между полем [День] таблицы "Привесы 1" и полем [Последний День] запроса "Последний день", перетащив поле [Последний День] на поле [День]. Правой кнопкой мыши щелкнуть по связи и изменить параметры объединения, поставив радиокнопку "Объединение ВСЕХ записей из "Последний день" и только тех записей из "Привесы 1", в которых связанные поля совпадают". Это необходимо для того, чтобы вычисления были за последний день. Перенесите в запрос поля [День], [Доза], [Привес], [Масса]. Сохраните запрос под именем "Вычисление доз". Введите выражения вычисления потенциальной дозы в свободном поле запроса по (3) П.2. для автоматического кормления по максимуму привесов

Дпот А1: -0,004*(120-[Масса А1])^1,5+5.

в новом поле запроса - поддерживающей дозы (2) П.2.

$$\text{Дпот A1: } -0,004*(120-[\text{Масса A1}])^{1,3+2}.$$

в новом поле запроса - поддерживающей дозы потенциального привеса по (5)
П.2.

$$\text{Ппот A1: } -4\text{E-}8*(200-[\text{Масса A1}])^{3,2+0,8}.$$

Для убыстрения работы формулы можно копировать через буфер или столбцы целиком.

И по другим способам кормления:

$$\text{Дпот A2: } -0,004*(120-[\text{Масса A2}])^{1,5+5}.$$

$$\text{Дпод A2: } -0,004*(120-[\text{Масса A2}])^{1,3+2}.$$

$$\text{Ппот A2: } -4\text{E-}8*(200-[\text{Масса A2}])^{3,2+0,8}.$$

$$\text{Дпот P: } -0,004*(120-[\text{Масса P}])^{1,5+5}.$$

$$\text{Дпод P: } -0,004*(120-[\text{Масса P}])^{1,3+2}.$$

$$\text{Ппот P: } -4\text{E-}8*(200-[\text{Масса P}])^{3,2+0,8}.$$

Сохраните и закройте запрос. Третий запрос будет вычислять новые значения дня, привесов масс и доз кормления. Создать запрос в режиме конструктора запросов, добавить запросы "Коэффициент а", "Вычисление доз" и таблицу "Цены". Перенесите в запрос поле [День]. Сохраните запрос под именем "Вычисление новых значений". Введите выражения для вычисления нового дня кормления

$$\text{Новый день:}[\text{День}]+1$$

вычисления новой дозы в свободном поле при автоматическом кормлении по критерию максимум привесов

$$\text{НДоза A1: } [\text{Дпот A1}]$$

вычисления нового привеса

$$\text{НПривес A1: } [\text{Ппот A1}]$$

вычисления новой массы

$$\text{НМасса A1: } [\text{Ппот A1}]+[\text{Масса A1}]$$

вычисления прибыли

$$\text{НПрибыль A1: } [\text{НПривес A1}]*[\text{Цена свинины}]-[\text{Ндоза A1}]*[\text{Цена корма}]$$

Также для остальных способов кормления

$$\text{НДоза A2: } \text{Sqr}([\text{Дпот A2}]^2-[\text{Ппот A2}]/[\text{a}])$$

НПривес А2: $\text{If}([\text{НДоза А2}] > [\text{Дпот А2}]; [\text{Ппот А2}]; -[\text{a}] * ([\text{Дпот А2}] - [\text{НДоза А2}]^2 + [\text{Ппот А2}])$

НМасса А2: $[\text{Масса А2}] + [\text{НПривес А2}]$

НПрибыль А2: $[\text{НПривес А2}] * [\text{Цена свинины}] - [\text{Ндоза А2}] * [\text{Цена корма}]$

НДоза Р: [Новая доза]

НПривес Р: $\text{If}([\text{НДоза Р}] > [\text{Дпот Р}]; [\text{Ппот Р}]; -[\text{a}] * ([\text{Дпот Р}] - [\text{НДоза Р}]^2 + [\text{Ппот Р}])$

НМасса Р: $[\text{Масса Р}] + [\text{НПривес Р}]$

НПрибыль Р: $[\text{НПривес Р}] * [\text{Цена свинины}] - [\text{НДоза Р}] * [\text{Цена корма}]$

Сохраните и закройте запрос. Создайте четвертый запрос на добавление к таблице "Привесы 1" новых вычислений. Для этого создайте запрос в режиме конструктора. Добавьте в запрос таблицу "Привесы 1" и запрос "Вычисление новых значений". Установите связь между полем [День] таблицы "Привесы 1" и полем [День] запроса "Вычисление новых значений", перетаскив поле [День] с запроса "Вычисление новых значений" на поле [День] таблицы "Привесы 1". Правой кнопкой мыши щелкните по линии связи и измените параметры объединения, поставив радиокнопку "Объединение ВСЕХ записей из "Вычисление новых значений" и только тех записей из "Привесы 1", в которых связанные поля совпадают". Перенесите в запрос поля [НДень], [Ндоза А1], [НПривес А1], [Нмасса А1], [НПрибыль А1], [Ндоза А2], [НПривес А2], [Нмасса А2], [НПрибыль А2], [Ндоза Р], [НПривес Р], [Нмасса Р], [НПрибыль Р]. Сохраните запрос под именем "Добавление новых значений". Измените тип запроса на запрос на добавление и укажите что добавление будет производиться к таблице "Привесы 1". В строке запроса добавление укажите поля таблицы "Привесы 1" [День], [Доза А1], [Привес А1], [Масса А1], [Прибыль А1], [Доза А2], [Привес А2], [Масса А2], [Прибыль А2], [Доза Р], [Привес Р], [Масса Р], [Прибыль Р] соответственно вычисляемым полям (Например поле [НДоза А1] будет записываться в поле [Доза А1]). Сохраните и закройте запрос.

3.4. Выкормите поросят 100 дней.

Как сделать. Перед кормлением определите дозу кормления ручным способом. Для этого запустите запрос "Вычисление новых значений". На запрос ввести новую дозу введите 3. Сравните привесы по Вашей дозе и по автоматическому кормлению. Выберите дозу для получения максимальных привесов, но с учетом ручного кормления округлите до десятых. Выкормите трех поросят десять дней, для чего запустите запрос "Добавление новых значений" 10 раз каждый раз вво-

для одну и ту же дозу ручного кормления. После десяти дней проверьте, как выросли поросята по таблице "Привесы 1". Снова определите дозу ручного кормления по запросу "Вычисление новых значений". И кормите поросенка еще 10 дней. И так до ста дней.

3.5. Сравнение результатов, полученные в результате моделирования.

3.5.1. Подсчитайте сумму доз корма, сумму прибыли и конечную массу поросят.

Как сделать. Создайте запрос с помощью конструктора запросов и добавьте в него таблицу "Привесы 1". Перенесите в запрос поля [Доза А1], [Прибыль А1], [Масса А1], [Доза А2], [Прибыль А2], [Масса А2], [Доза Р], [Прибыль Р], [Масса Р]. Включите групповые операции. По полям [Доза А1], [Прибыль А1], [Доза А2], [Прибыль А2], [Доза Р], [Прибыль Р] в строке групповые операции поставьте Sum. А по полям [Масса А1], [Масса А2], [Масса Р] – Last (определения последнего значения).

3.5.2. Перепишите результаты в отчет. Напишите вывод по проделанной работе и рекомендации по способу кормления.

4. Содержание отчета

4.1. Название и цель выполнения работы.

4.2. Вывод и рекомендации.

5. Контрольные вопросы

5.1. Объясните структуру себестоимости производства свинины.

5.2. Понятие алгоритма функционирования и алгоритма управления.

5.3. Назовите алгоритм функционирования и управления для производства свинины.

5.4. Какой критерий производства свинины предпочтительнее?

5.5. Объясните зависимость привеса от суточного расхода кормов.

5.6. От чего и как зависят потенциальная и поддерживающая дозы?

5.7. От чего зависит потенциальный привес животного?

5.8. Пути повышения производства свинины.

5.9. Пути повышения продуктивности свиней.

5.10. Роль БД в АСУ производства свинины.

5.11. От чего зависит прибыльность производства свинины?

5.12. Какой критерий оптимальности выращивания свиней – максимум привесов или максимум привесов при минимуме затрат будет более выгоден и почему?

5.13. Почему ручной способ кормления с целью получения максимума привесов будет менее выгоден, чем автоматическое кормление по критерию максимум привесов?

5.14. Почему ручной способ кормления с целью получения максимума привесов будет менее выгоден, чем автоматическое кормление по критерию максимум привесов при минимуме затрат корма?

6. Литература

6.1. "Access 97", Виктор Пасько – К.: Издательская группа ВАН 1997г.

6.2. Мороз Ю.Д., Ширшова В.В. Эффективность автоматизации свиноводческих комплексов. – Мн.: Ураджай, 1992 г.

Лабораторная работа №4

АНАЛИЗ ПРОИЗВОДСТВА ПТИЦЫ ПО ДВУХМЕРНОЙ МОДЕЛИ

1. Цель работы: Определить влияние отдельных факторов на продуктивность птицы. Проанализировать влияние параметров производства птицы на прибыль по двухмерной экономико-математической модели.

2. Краткие теоретические сведения

2.1. Перед управляющим предприятия ставится задача – выбрать из множества факторов несколько наиболее влияющих на результат производства, так как управление является затратным, и управлять всеми факторами невыгодно. Для этого проводится эксперимент и анализируются полученные результаты. Степень влияния факторов в лабораторной работе предлагается определить с помощью коэффициентов корреляции. Чем больше коэффициент корреляции, тем больше зависимость целевого параметра от фактора. С инженерной точки зрения незначительными факторами следует считать факторы с коэффициентом корреляции меньше 0,05, то есть фактор можно расценивать как 5% погрешность. С экономической точки зрения следует определять не только степень влияния факторов, но и стоимость управления данным параметром.

2.2. При построении сложных систем управления, в каких взаимовлияние факторов имеет нелинейный характер, выявить влияние отдельного фактора на результирующий параметр с помощью корреляционного анализа сложно. Так, например, на привесы влияют только факторы, оказывающие воздействие на биологический объект – доза корма и параметры микроклимата. На прибыль будут влиять факторы, оказывающие воздействие на биологический объект, факторы изменяющие эти воздействия и цены. В этом случае больше подходит имитационное моделирование процесса производства.

2.3. Оптимальная среда для производства птицы будет определяться условием оптимальности. В данной работе параметром среды производства птицы считается температура внутри помещения. Если условие оптимальности – максимум привесов, то эта температура должна соответствовать температуре, при которой птица дает максимальные привесы. Эта оптимальная температура не должна зависеть от системы создающей условия производства. Например, от КПД отопительно-вентиляционной системы (ОВС). Если условие оптимальности – максимум прибыли, то КПД ОВС по этому критерию оптимальности будет влиять на оптимальную температуру.

3. Программа работы

3.1. Ознакомьтесь с теоретическими сведениями (Приложение 4).

3.2. Определите коэффициенты корреляции привеса от дозы корма и параметров микроклимата внутри производственного здания: температуры, влажности и концентрации CO₂.

Как сделать. Откройте базу данных "Лабораторная работа №9". Сначала создайте запрос для вычисления средних значений и дисперсий. Создайте запрос с помощью конструктора запросов и добавьте в запрос таблицу "Данные". Перенесите в запрос поля [Доза корма], [Температура], [Влажность], [Концентрация CO₂], [Привес]. Сохраните запрос под именем "Средние значения и дисперсии". По полю [Доза корма] введите (не стирая названия поля) псевдоним **Ср Доза**, чтобы получилось выражение

Ср Доза: Доза корма

Включите групповые операции и по полю дозы в строке групповая операция выберите функцию Avg. В свободном поле введите выражение вычисления дисперсии

Дис Доза: VarP([Доза корма])

Для быстроты работы созданные поля лучше скопировать и отредактировать.

В строке групповая операция по этому полю выберите Выражение. Повторите аналогичные операции по отношению к полям [Температура], [Влажность], [Концентрация CO₂], [Привес] так чтобы поля запроса были соответственно полям рис.1. Сохраните и закройте запрос. Потом создайте запрос на вычисление промежуточных коэффициентов. Создайте запрос с помощью конструктора и добавьте в запрос таблицу "Данные" и запрос "Средние значения и дисперсии". Включите групповые операции. В свободном поле введите выражение

К Доза: ([Доза корма]-[Ср Доза])*([Привес]-[Ср Пр]).

В строке "Групповая операция" выберите Avg. Повторите аналогичные операции по отношению к полям [Температура], [Влажность], [Концентрация CO₂], [Привес] так чтобы поля были как на рис.1. Сохраните запрос под именем "Промежуточные коэффициенты" и закройте. Создайте запрос на вычисление коэффициентов корреляции. Создайте с помощью конструктора запрос и добавьте в запрос запросы "Средние значения и дисперсии" и "Промежуточные коэффициенты".



Рис.1. Структура запрос для вычисления коэффициентов корреляции

В свободном поле запроса введите выражение для вычисления коэффициента корреляции

$$\text{Кор Доза: } [К \text{ Доза}] / \text{Sqr}([Дис \text{ доза}] * [Дис \text{ Пр}])$$

Повторите аналогичные операции по отношению к полям [Температура], [Влажность], [Концентрация CO₂], [Привес] так чтобы поля были как на рис.1. Сохраните запрос под именем " Коэффициенты корреляции" и закройте.

3.3. Напишите вывод о влиянии факторов на привес, какие параметры следует использовать для контроля, какие – для управления. Почему коэффициент корреляции привеса от привеса равен единице.

3.4. Проанализируйте влияние факторов производства на экономические показатели по двухмерной экономической модели продуктивности птицы.

Как сделать. Запустите Excel. Откройте файл "Лабораторная работа №9". Установите исходные данные, приведенные в табл.1.

Таблица 1. Исходные параметры.

Средняя масса одной птицы, г	Количество птицы, шт	КПД ОВС %	Температура наружного воздуха, С	Коэффициент эффективности топлива	Цена энергии, тыс.руб/кВт*ч.	Цена продукции, тыс.руб/т
1000	8000	80	5	0.99	2000	1000

3.4.1. Определите влияние средней массы птицы на максимальные привесы, прибыль, оптимальные температуры по критерию максимум привесов и максимум прибыли.

Как сделать. Изменяя среднюю массу птицы в диапазоне от 10 г. до 2000 г. (в таблице Excel, в ячейке зеленого цвета) запишите значения максимальных привесов и прибыли (красные ячейки). Зависимости в виде графика зарисуйте в отчет.

При изменении массы определите влияние на оптимальную температуру по критерия максимум прибыли и максимум привесов.

Как сделать. Изменяя среднюю массу птицы в диапазоне в том же диапазоне, запишите приблизительные значения температур соответствующих максимуму привесов (график "Привесы") и максимуму прибыли (график "Прибыль").

Напишите вывод.

3.4.2. Определите влияние температуры наружного воздуха (четыре значения) на максимальные привесы, прибыль, оптимальные температуры по критерию максимум привесов и максимум прибыли. Зависимость максимальной прибыли и привесов от температуры наружного воздуха в виде графика зарисуйте в отчет. Напишите вывод.

3.4.3. Изменяя КПД ОВС (60%, 70%, 80%, 90%) определите зависимости максимального прироста и прибыли от КПД ОВС. Зависимость максимальной прибыли от КПД ОВС в виде графика зарисуйте в отчет. Напишите вывод.

3.4.4. Изменяя вид топлива, определите зависимости максимального прироста и прибыли от вида топлива.

Как сделать. Необходимо изменять коэффициент эффективности и цену топлива.

Зависимость максимальной прибыли от вида топлива в виде графика зарисуйте в отчет. Напишите вывод.

3.4.5. Изменяя цену на энергию, определите зависимости максимального прироста и прибыли от цены на энергию. Зависимость максимальной прибыли от цены на энергию в виде графика зарисуйте в отчет. Напишите вывод.

3.2.8. Изменяя цену на продукцию (три значения), определите зависимости максимального прироста и прибыли от цены на продукцию. Зависимость максимальной прибыли от цены на продукцию в виде графика зарисуйте в отчет. Напишите вывод.

4. Содержание отчета

4.1. Название и цель работы.

4.2. Выводы и графики.

5. Контрольные вопросы

5.1. Почему математические модели, описывающие большинство технологических процессов сельского хозяйства должны быть технико-биолого-экономическими?

5.2. Перечислите параметры, от которых зависит оптимальная температура воздуха по критерию максимум продуктивности в помещении для выращивания птицы.

5.3. Перечислите параметры, от которых зависит оптимальная температура воздуха по максимум прибыли в помещении для выращивания птицы.

5.4. Почему критерием оптимальности вышеописанной системы следует считать получаемую прибыль? Доказать, используя графики, полученные при выполнении работы.

5.5. Как оптимальная температура воздуха в помещении зависит от массы птицы?

5.6. Как оптимальная температура зависит от температуры наружного воздуха?

5.7. Как на прибыльность откорма влияют вид и цена топлива?

5.8. Как и почему на оптимальную температуру влияет КПД ОВС?

5.9. Как привесы зависят от массы птицы?

5.10. Как прибыль зависит от массы птицы?

5.11. Как прибыль и привесы зависят от температуры наружного воздуха?

5.12. Как прибыль и привесы зависят от КПД ОВС?

6. Литература

6.1. "Access 97", Виктор Пасько – К.: Издательская группа ВАУ 1997г.

ПРИЛОЖЕНИЕ 1. РЕЛЯЦИОННЫЕ БАЗЫ ДАННЫХ

1. Базовые понятия реляционных баз данных

Основными понятиями реляционных баз данных являются тип данных, домен, атрибут, кортеж, первичный ключ и отношение.

1.1. Тип данных

Понятие *тип данных* в реляционной модели данных полностью адекватно понятию типа данных в языках программирования. Обычно в современных реляционных БД допускается хранение символьных, числовых данных, битовых строк, специализированных числовых данных (таких как "деньги"), а также специальных "темпоральных" данных (дата, время, временной интервал). Достаточно активно развивается подход к расширению возможностей реляционных систем абстрактными типами данных. Тип данных связан со способом хранения информации, и смыслом хранимой информации (нормализация первого уровня)

1.2. Домен, атрибуты и кортежи

Понятие *домена* более специфично для баз данных, хотя и имеет некоторые аналогии с подтипами в некоторых языках программирования. Наиболее правильной интуитивной трактовкой понятия домена является понимание домена как допустимого потенциального множества атрибутов объединенных одним смысловым понятием. Например, домен "Имена" определен на базовом типе строк символов, но в число его значений могут входить только те строки, которые могут изображать имя (в частности, такие строки не могут начинаться с мягкого знака).

Следует отметить также семантическую нагрузку понятия домена: данные считаются сравнимыми только в том случае, когда они относятся к одному домену.

Атрибуты – элементы домена, имеющие каждый свой тип данных, и описывающие домен как объект с определенными свойствами.

Кортеж – набор значений атрибутов. Кортеж является одним из элементов информации объекта базы данных, хранящемся в домене.

В простейшем понимании доменом является множество полей одной таблицы. Поля таблицы – атрибуты. Каждое поле таблицы описывает одно из свойств объекта хранимого в таблице. Кортеж – запись в таблице. Множество записей одной таблицы – картежами.

1.3. Схема отношения, схема базы данных

Схема отношения - это именованное множество пар {имя атрибута, имя домена}. Если все атрибуты одного отношения определены на разных доменах, осмысленно использовать для именованния атрибутов имена соответствующих доменов (не забывая, конечно, о том, что это является всего лишь удобным способом именованния и не устраняет различия между понятиями домена и атрибута).

Схема БД (в структурном смысле) - это набор именованных схем отношений.

1.4. Кортеж, отношение

Кортеж, соответствующий данной схеме отношения, - это множество пар {имя атрибута, значение}, которое содержит одно вхождение каждого имени атрибута, принадлежащего схеме отношения. "Значение" является допустимым значением домена данного атрибута. Попросту говоря, кортеж - это набор именованных значений заданного типа.

На самом деле, понятие схемы отношения ближе всего к понятию структурного типа данных в языках программирования. Было бы вполне логично разрешать отдельно определять схему отношения, а затем одно или несколько отношений с данной схемой.

Однако в реляционных базах данных это не принято. Имя схемы отношения в таких базах данных всегда совпадает с именем соответствующего отношения-экземпляра. В классических реляционных базах данных после определения схемы базы данных изменяются только отношения-экземпляры. В них могут появляться новые и удаляться или модифицироваться существующие кортежи. Однако во многих реализациях допускается и изменение схемы базы данных: определение новых и изменение существующих схем отношения. Это принято называть *эволюцией схемы базы данных*.

Обычным житейским представлением отношения является таблица, заголовком которой является схема отношения, а строками - кортежи отношения-экземпляра; в этом случае имена атрибутов именуют столбцы этой таблицы. Поэтому иногда говорят "столбец таблицы", имея в виду "атрибут отношения". Когда мы перейдем к рассмотрению практических вопросов организации реляционных баз данных и средств управления, мы будем использовать эту житейскую терминологию. Этой терминологии придерживаются в большинстве коммерческих реляционных СУБД.

Реляционная база данных - это набор отношений, имена которых совпа-

дают с именами схем отношений в схеме БД.

2. Фундаментальные свойства отношений

Остановимся теперь на некоторых важных свойствах отношений, которые следуют из приведенных ранее определений:

2.1. Отсутствие кортежей-дубликатов

То свойство, что отношения не содержат кортежей-дубликатов, следует из определения отношения как множества кортежей. В классической теории множеств по определению каждое множество состоит из различных элементов.

Из этого свойства вытекает наличие у каждого отношения так называемого первичного ключа - набора атрибутов, значения которых однозначно определяют кортеж отношения. Для каждого отношения, по крайней мере, полный набор его атрибутов обладает этим свойством. Однако при формальном определении первичного ключа требуется обеспечение его "минимальности", т.е. в набор атрибутов первичного ключа не должны входить такие атрибуты, которые можно отбросить без ущерба для основного свойства - однозначно определять кортеж. Понятие *первичного ключа* является исключительно важным в связи с понятием целостности баз данных.

2.2. Отсутствие упорядоченности кортежей

Свойство отсутствия упорядоченности кортежей отношения также является следствием определения отношения-экземпляра как множества кортежей. Отсутствие требования к поддержанию порядка на множестве кортежей отношения дает дополнительную гибкость СУБД при хранении баз данных во внешней памяти и при выполнении запросов к базе данных. Это не противоречит тому, что при формулировании запроса к БД, например, на языке SQL можно потребовать сортировки результирующей таблицы в соответствии со значениями некоторых столбцов. Такой результат, вообще говоря, не отношение, а некоторый упорядоченный список кортежей.

2.3. Отсутствие упорядоченности атрибутов

Атрибуты отношений не упорядочены, поскольку по определению схема отношения есть множество пар {имя атрибута, имя домена}. Для ссылки на значение атрибута в кортеже отношения всегда используется имя атрибута. Это свойство теоретически позволяет, например, модифицировать схемы существующих отношений не только путем добавления новых атрибутов, но и путем удаления существующих атрибутов. Однако в большинстве существующих

систем такая возможность не допускается, и хотя упорядоченность набора атрибутов отношения явно не требуется, часто в качестве неявного порядка атрибутов используется их порядок в линейной форме определения схемы отношения.

2.4. Атомарность значений атрибутов

Значения всех атрибутов являются атомарными. Это следует из определения домена как потенциального множества значений простого типа данных, т.е. среди значений домена не могут содержаться множества значений (отношения). Принято говорить, что в реляционных базах данных допускаются только нормализованные отношения или отношения, представленные в первой нормальной форме.

3. Отношения между таблицами.

Связь между таблицами устанавливает отношения между совпадающими значениями в ключевых полях, обычно между полями, имеющими одинаковые имена в обеих таблицах. В большинстве случаев с ключевым полем одной таблицы, являющимся уникальным идентификатором каждой записи, связывается внешний ключ другой таблицы.

3.1. Отношение «один-ко-многим»

Отношение «один-ко-многим» является наиболее часто используемым типом связи между таблицами. В отношении «один-ко-многим» каждой записи в таблице А могут соответствовать несколько записей в таблице В, но запись в таблице В не может иметь более одной соответствующей ей записи в таблице А.

3.2. Отношение «многие-ко-многим»

При отношении «многие-ко-многим» одной записи в таблице А могут соответствовать несколько записей в таблице В, а одной записи в таблице В несколько записей в таблице А. Этот тип связи возможен только с помощью третьей (связующей) таблицы, первичный ключ которой состоит из двух полей, которые являются внешними ключами таблиц А и В. Отношение «многие-ко-многим» по сути дела представляет собой два отношения «один-ко-многим» с третьей таблицей. Например, отношение «многие-ко-многим» между таблицами «Семейное положение» и «Пол» определяется путем создания двух отношений «один-ко-многим» с таблицей «Служащие».

3.3. Отношение «один-к-одному»

При отношении «один-к-одному» запись в таблице А может иметь не более одной связанной записи в таблице В и наоборот. Отношения этого типа используются не очень часто, поскольку большая часть сведений, связанных таким образом, может быть помещена в одну таблицу. Отношение «один-к-одному» может использоваться для разделения очень широких таблиц, для отделения части таблицы по соображениям защиты, а также для сохранения сведений.

3.4. Определение связей между таблицами

Связь между таблицами определяется путем добавления связываемых таблиц в окно «Схема данных» с последующим перетаскиванием ключевого поля из одной таблицы в другую. Также можно определять связи с помощью клавиатуры. Тип отношения в создаваемой MS Access связи зависит от способа определения связываемых полей.

- Отношение «один-ко-многим» создается в том случае, когда только одно из полей является ключевым или имеет уникальный индекс.
- Отношение «один-к-одному» создается в том случае, когда оба связываемых поля являются ключевыми или имеют уникальные индексы.
- Отношение «многие-ко-многим» фактически является двумя отношениями «один-ко-многим» с третьей таблицей, первичный ключ которой состоит из полей — внешних ключей двух других таблиц.

***Примечание.** Если перетащить поле, не являющееся ключевым и не имеющее уникального индекса, на другое поле, которое также не является ключевым и не имеет уникального индекса, создается неопределенное отношение. В запросах, содержащих таблицы с неопределенным отношением, MS Access по умолчанию отображает линию объединения между таблицами, но условия целостности данных при этом не накладываются и нет гарантии уникальности записей в любой из таблиц.*

4. Реляционная модель данных

Когда в предыдущих разделах мы говорили об основных понятиях реляционных баз данных, мы не опирались на какую-либо конкретную реализацию. Эти рассуждения в равной степени относились к любой системе, при построении которой использовался реляционный подход.

Другими словами, мы использовали понятия так называемой реляционной модели данных. Модель данных описывает некоторый набор родовых понятий

и признаков, которыми должны обладать все конкретные СУБД и управляемые ими базы данных, если они основываются на этой модели. Наличие модели данных позволяет сравнивать конкретные реализации, используя один общий язык.

Хотя понятие модели данных является общим, и можно говорить о иерархической, сетевой, некоторой семантической и т.д. моделях данных, нужно отметить, что это понятие было введено в обиход применительно к реляционным системам и наиболее эффективно используется именно в этом контексте. Попытки прямолинейного применения аналогичных моделей к дореляционным организациям показывают, что реляционная модель слишком "велика" для них, а для постреляционных организаций она оказывается "мала".

4.1. Общая характеристика

Наиболее распространенная трактовка реляционной модели данных, по-видимому, принадлежит Дейту¹, который воспроизводит ее (с различными уточнениями) практически во всех своих книгах. Согласно Дейту реляционная модель состоит из трех частей, описывающих разные аспекты реляционного подхода: структурной части, манипуляционной части и целостной части.

В структурной части модели фиксируется, что единственной структурой данных, используемой в реляционных БД, является нормализованное n -арное отношение. По сути дела, в предыдущих двух разделах этой лекции мы рассматривали именно понятия и свойства структурной составляющей реляционной модели.

В манипуляционной части модели утверждаются два фундаментальных механизма манипулирования реляционными БД - реляционная алгебра и реляционное исчисление. Первый механизм базируется в основном на классической теории множеств (с некоторыми уточнениями), а второй - на классическом логическом аппарате исчисления предикатов первого порядка. Основной функцией манипуляционной части реляционной модели является обеспечение меры реляционности любого конкретного языка реляционных БД: язык называется реляционным, если он обладает не меньшей выразительностью и мощностью, чем реляционная алгебра или реляционное исчисление.

4.2. Целостность сущности и ссылок

¹ К. Дейт "Введение в системы баз данных" (Наука, 1980) и "Руководство по реляционной СУБД DB2" (Финансы и статистика, 1988),

Наконец, в целостной части реляционной модели данных фиксируются два базовых требования целостности, которые должны поддерживаться в любой реляционной СУБД. Первое требование называется *требованием целостности сущностей*. Объекту или сущности реального мира в реляционных БД соответствуют кортежи отношений. Конкретно требование состоит в том, что любой кортеж любого отношения отличим от любого другого кортежа этого отношения, т.е. другими словами, любое отношение должно обладать первичным ключом. Как мы видели в предыдущем разделе, это требование автоматически удовлетворяется, если в системе не нарушаются базовые свойства отношений.

Второе требование называется *требованием целостности по ссылкам* и является несколько более сложным. Очевидно, что при соблюдении нормализованности отношений сложные сущности реального мира представляются в реляционной БД в виде нескольких кортежей нескольких отношений. Например, представим, что нам требуется представить в реляционной базе данных сущность ОТДЕЛ с атрибутами ОТД_НОМЕР (номер отдела), ОТД_КОЛ (количество сотрудников) и ОТД_СОТР (набор сотрудников отдела). Для каждого сотрудника нужно хранить СОТР_НОМЕР (номер сотрудника), СОТР_ИМЯ (имя сотрудника) и СОТР_ЗАРП (заработная плата сотрудника). Как мы вскоре увидим, при правильном проектировании соответствующей БД в ней появятся два отношения: ОТДЕЛЫ (ОТД_НОМЕР, ОТД_КОЛ) (первичный ключ - ОТД_НОМЕР) и СОТРУДНИКИ (СОТР_НОМЕР, СОТР_ИМЯ, СОТР_ЗАРП, СОТР_ОТД_НОМ) (первичный ключ - СОТР_НОМЕР).

Как видно, атрибут СОТР_ОТД_НОМ появляется в отношении СОТРУДНИКИ не потому, что номер отдела является собственным свойством сотрудника, а лишь для того, чтобы иметь возможность восстановить при необходимости полную сущность ОТДЕЛ. Значение атрибута СОТР_ОТД_НОМ в любом кортеже отношения СОТРУДНИКИ должно соответствовать значению атрибута ОТД_НОМ в некотором кортеже отношения ОТДЕЛЫ. Атрибут такого рода называется *внешним ключом*, поскольку его значения однозначно характеризуют сущности, представленные кортежами некоторого другого отношения (т.е. задают значения их первичного ключа). Говорят, что отношение, в котором определен внешний ключ, ссылается на соответствующее отношение, в котором такой же атрибут является первичным ключом.

Требование целостности по ссылкам, или требование внешнего ключа состоит в том, что для каждого значения внешнего ключа, появляющегося в ссы-

лающемся отношении, в отношении, на которое ведет ссылка, должен найтись кортеж с таким же значением первичного ключа, либо значение внешнего ключа должно быть неопределенным (т.е. ни на что не указывать). Для нашего примера это означает, что если для сотрудника указан номер отдела, то этот отдел должен существовать.

Ограничения целостности сущности и по ссылкам должны поддерживаться СУБД. Для соблюдения целостности сущности достаточно гарантировать отсутствие в любом отношении кортежей с одним и тем же значением первичного ключа. При обновлении ссылающегося отношения (вставке новых кортежей или модификации значения внешнего ключа в существующих кортежах) достаточно следить за тем, чтобы не появлялись некорректные значения внешнего ключа. Но как быть при удалении кортежа из отношения, на которое ведет ссылка?

Здесь существуют три подхода, каждый из которых поддерживает целостность по ссылкам. Первый подход заключается в том, что запрещается производить удаление кортежа, на который существуют ссылки (т.е. сначала нужно либо удалить ссылающиеся кортежи, либо соответствующим образом изменить значения их внешнего ключа). При втором подходе при удалении кортежа, на который имеются ссылки, во всех ссылающихся кортежах значение внешнего ключа автоматически становится неопределенным. Наконец, третий подход (каскадное удаление) состоит в том, что при удалении кортежа из отношения, на которое ведет ссылка, из ссылающегося отношения автоматически удаляются все ссылающиеся кортежи.

В развитых реляционных СУБД обычно можно выбрать способ поддержания целостности по ссылкам для каждой отдельной ситуации определения внешнего ключа. Конечно, для принятия такого решения необходимо анализировать требования конкретной прикладной области.

5. Проектирование реляционных БД

При проектировании базы данных решаются две основных проблемы:

- Каким образом отобразить объекты предметной области в абстрактные объекты модели данных, чтобы это отображение не противоречило семантике предметной области и было по возможности лучшим (эффективным, удобным и т.д.)? Часто эту проблему называют проблемой логического проектирования баз данных.

- Как обеспечить эффективность выполнения запросов к базе данных, т.е. каким образом, имея в виду особенности конкретной СУБД, расположить данные во внешней памяти, создание каких дополнительных структур (например, индексов) потребовать и т.д.? Эту проблему называют проблемой физического проектирования баз данных.

В случае реляционных баз данных трудно представить какие-либо общие рецепты по части физического проектирования. Здесь слишком много зависит от используемой СУБД. Логическое построение реляционных баз данных одинаковы при использовании любой реляционной СУБД.

Более того, мы не будем касаться очень важного аспекта проектирования - определения ограничений целостности (за исключением ограничения первичного ключа). Дело в том, что при использовании СУБД с развитыми механизмами ограничений целостности (например, SQL-ориентированных систем) трудно предложить какой-либо общий подход к определению ограничений целостности.

Так что будем считать, что проблема проектирования реляционной базы данных состоит в обоснованном принятии решений о том:

- из каких отношений должна состоять БД и
- какие атрибуты должны быть у этих отношений

5.1. Проектирование реляционных баз данных с использованием нормализации

Нормализация процесс упорядочивания данных.

Процесс проектирования представляет собой процесс нормализации схем отношений, причем каждая следующая нормальная форма обладает свойствами лучшими, чем предыдущая.

Каждой нормальной форме соответствует некоторый определенный набор ограничений, и отношение находится в некоторой нормальной форме, если удовлетворяет свойственному ей набору ограничений. Примером набора ограничений является ограничение первой нормальной формы - значения всех атрибутов отношения атомарны (**поля таблицы имеют определенных тип данных**). Поскольку требование первой нормальной формы является базовым требованием классической реляционной модели данных, считается, что исходный набор отношений уже соответствует этому требованию.

Вторая нормальная норма - *единственным ключом отношения является*

первичный ключ. Таблица находится во второй нормальной форме в том и только в том случае, когда находится в первой нормальной форме, и каждое не ключевое поле таблицы полностью зависит от первичного ключа.

Третья нормальная форма - **единственным ключом отношения является первичный ключ.** Таблица находится в третьей нормальной форме в том и только в том случае, если находится во второй нормальной форме и каждое не ключевое поле не транзитивно зависит от первичного ключа.

Примечание. Транзитивная зависимость – зависимость между таблицами *A* и *B*, *A* и *B* и нет зависимости между таблицами *B* и *B*.

MS Access имеет встроенный инструмент нормализации. Если в базе данных MS Access имеется таблица, в одном или нескольких полях которой содержатся повторяющиеся сведения, можно воспользоваться мастером анализа таблиц для разбиения этих данных на связанные таблицы с целью повышения эффективности хранения данных. Этот процесс называют нормализацией таблиц.

В меню **Сервис** выберите команду **Анализ** и подкоманду **Таблица**.

Следуйте инструкциям **мастера анализа таблиц**.

При использовании мастера пользователь имеет возможность самостоятельно определить создаваемые таблицы или позволить мастеру провести нормализацию таблиц. После определения создаваемых таблиц мастер помогает пользователю согласовать данные, несогласованно повторяющиеся в исходной таблице. На последнем этапе можно создать запрос, позволяющий просмотреть все данные из разделенных таблиц в одной таблице, которая будет аналогична исходной.

5.2. Семантическое моделирование данных

Широкое распространение реляционных СУБД и их использование в самых разнообразных приложениях показывает, что реляционная модель данных достаточна для моделирования предметных областей. Однако проектирование реляционной базы данных в терминах отношений на основе кратко рассмотренного нами механизма нормализации часто представляет собой очень сложный и неудобный для проектировщика процесс.

При этом проявляется ограниченность реляционной модели данных в следующих аспектах:

- Модель не предоставляет достаточных средств для представления смысла данных. Семантика реальной предметной области должна независимым

от модели способом представляться в голове проектировщика. В частности, это относится к упоминавшейся нами проблеме представления ограничений целостности.

- Для многих приложений трудно моделировать предметную область на основе плоских таблиц. В ряде случаев на самой начальной стадии проектирования проектировщику приходится производить насилие над собой, чтобы описать предметную область в виде одной (возможно, даже ненормализованной) таблицы.
- Хотя весь процесс проектирования происходит на основе учета зависимостей, реляционная модель не предоставляет каких-либо средств для представления этих зависимостей.
- Несмотря на то, что процесс проектирования начинается с выделения некоторых существенных для приложения объектов предметной области ("сущностей") и выявления связей между этими сущностями, реляционная модель данных не предлагает какого-либо аппарата для разделения сущностей и связей.

5.3. Семантические модели данных

Потребности проектировщиков баз данных в более удобных и мощных средствах моделирования предметной области вызвали к жизни направление семантических моделей данных. При том, что любая развитая семантическая модель данных, как и реляционная модель, включает структурную, манипуляционную и целостную части, главным назначением семантических моделей является обеспечение возможности выражения семантики данных.

Прежде, чем мы коротко рассмотрим особенности одной из распространенных семантических моделей, остановимся на их возможных применениях.

Наиболее часто на практике семантическое моделирование используется на первой стадии проектирования базы данных. При этом в терминах семантической модели производится концептуальная схема базы данных, которая затем вручную концептуальная схема преобразуется к реляционной (или какой-либо другой) схеме. Этот процесс выполняется под управлением методик, в которых достаточно четко оговорены все этапы такого преобразования.

Менее часто реализуется автоматизированная компиляция концептуальной схемы в реляционную. При этом известны два подхода: на основе явного представления концептуальной схемы как исходной информации для компилятора и

построения интегрированных систем проектирования с автоматизированным созданием концептуальной схемы на основе интервью с экспертами предметной области. И в том, и в другом случае в результате производится реляционная схема базы данных в третьей нормальной форме (более точно следовало бы сказать, что автору неизвестны системы, обеспечивающие более высокий уровень нормализации).

Наконец, третья возможность, которая еще не вышла (или только выходит) за пределы исследовательских и экспериментальных проектов, - это работа с базой данных в семантической модели, т.е. СУБД, основанные на семантических моделях данных. При этом снова рассматриваются два варианта: обеспечение пользовательского интерфейса на основе семантической модели данных с автоматическим отображением конструкций в реляционную модель данных (это задача примерно такого же уровня сложности, как автоматическая компиляция концептуальной схемы базы данных в реляционную схему) и прямая реализация СУБД, основанная на какой-либо семантической модели данных. Наиболее близко ко второму подходу находятся современные объектно-ориентированные СУБД, модели данных которых по многим параметрам близки к семантическим моделям (хотя в некоторых аспектах они более мощны, а в некоторых - более слабы).

6. Стрoение реляционных баз данных

Реляционные базы данных состоят из таблиц и связей между ними.

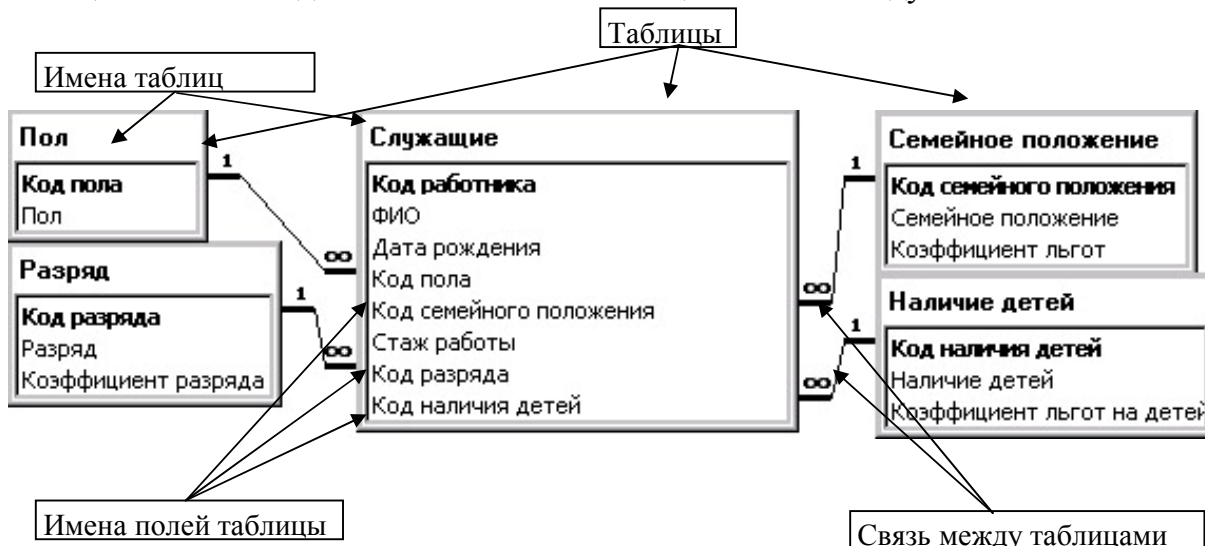


Рис.1. Таблицы и связи в реляционных базах данных.

Таблица состоит из полей и записей. Каждое поле имеет свой тип данных.

ФИО	Дата рождения	Пол	Стаж работы	Разряд
Иванов В.В.	01.10.1949	Мужской	40	VII
Петров И.И.	01.12.1951	Мужской	35	VI
			0	

Запись: 3 из 3

Рис.2. Таблицы.

7. Таблицы

В Access **таблицей** является совокупность данных, объединенных общей темой. Для каждой темы отводится отдельная таблица, что позволяет избежать повторения сохраняемых данных. Это положительно сказывается на эффективности работы с базой данных и уменьшает вероятность возникновения ошибок ввода.

Структуру создаваемых таблиц следует тщательно продумывать, чтобы впоследствии избежать изменений, требующих много времени. Существует несколько способов построения таблиц: на листе данных, в окне конструктора таблиц и с помощью мастера таблиц.

Каждая таблица состоит из записей и полей. Количество полей в записи, и их тип определяются в процессе конструирования таблицы. *Например, на этапе составления таблицы адресов необходимо создать поля для фамилий, адресов, почтовых индексов, названий населенных пунктов и т.п.*

Понятие **тип данных** в реляционной модели данных полностью адекватно понятию типа данных в языках программирования и соответствует способу хранения данных. Обычно в современных реляционных БД допускается хранение символьных, числовых данных, битовых строк, числовых данных специального формата (таких как "деньги"), а также специальных "временных" данных (дата, время, временной интервал). См. табл. 1.

В "Конструкторе таблиц" тип данных поля таблицы выбирается в колонке

"Тип данных".

Таблица 1. Типы данных полей используемые Access.

Тип данных	Описание
Текстовый	Поля этого типа содержат текст, объем которого не должен превышать 255 символов (длина поля устанавливается с помощью параметра Field Size)
Поле МЕМО	В Мемо-полях хранятся текстовые либо текстовые и числовые данные объемом не более 64000 символов (поля этого типа не индексируются)
Числовой	Поля этого типа содержат числовые значения, диапазон которых определяется параметром "Размер поля"
Дата/Время	Поля этого типа содержат значения даты и времени
Денежный	В полях этого типа могут храниться числа, содержащие 15 разрядов слева от десятичной точки и 4 разряда справа от нее
Счетчик	Специальный тип данных Access. Поле этого типа содержит число, которое автоматически увеличивается на единицу и некогда не повторяется, когда в таблицу добавляется новая запись
Логический	В таких полях сохраняются значения <i>Да</i> или <i>Нет</i> (поля этого типа не индексируются)
Поле объекта OLE	В поля данного типа помещаются объекты, обработанные OLE-сервером (такие поля не индексируются, и их объем не должен превышать 128 Мб)
Гиперссылка	В полях этого типа хранятся гиперссылки, которые представляют собой путь к файлу на жестком диске либо адрес в сетях Internet или интранет

Кроме вышперечисленных типов данных в списке колонки "Тип данных" присутствует элемент "Мастер подстановок", который позволяет представлять значения полей в виде простого или комбинированного списка или установить связи между таблицами.

Поле таблицы определяет множество однотипных данных из записей, объединенных по семантическому (смысловому) значению. *Например: поле "Имя" будет содержать множество данные в виде строк (строковый тип данных) и иметь значение имени какого-то объекта информация, о котором храниться в таблице.*

Для поля таблицы после назначения имени и определения его типа данных задаются ряд его свойств табл. 2.

Таблица 2. Названия свойств полей таблицы и их описание.

Свойство	Описание
Размер поля	Определяет максимальную длину текстового или числового поля (учтите, что при создании полей слишком большого размера неэкономно расходуется память, а полей маленького размера — искажается содержимое поля). Количество хранимой информации
Формат	Устанавливает формат отображения данных в форме и запросе
Число десятичных знаков	Определяет количество разрядов в дробной части десятичного числа


продолжение таблицы 2.



Маска ввода	Определяет маску данных при вводе
Название поля	Содержит надпись, которая выводится рядом с полем в форме или в отчете (данная надпись может и не совпадать с именем поля, а также может содержать поясняющие сведения)
Значение по умолчанию	Содержит значение, устанавливаемое по умолчанию в соответствующем поле таблицы при вводе новой записи
Условие на значение	Определяет множество значений, которые могут вводиться в поле
Сообщение об ошибке	Определяет сообщение, которое появляется на экране при вводе значения не соответствующего "Условию на значение"
Обязательное поле	Если установлено, то данное поле обязательно следует заполнить при вводе записи (Обеспечивает целостность данных)
Пустые строки	Установка, указывающая на то, что данное поле может иметь нулевую длину
Индексированное поле	Определяет простые индексы для ускорения поиска (в поле первичного ключа индекс устанавливается автоматически)

Величина полей устанавливается пользователем. В соответствии с ней ограничивается количество символов, вводимых в поле при заполнении таблицы. *При большом размере поля бесполезно расходуется память (она резервируется в полной мере независимо от того, заполняется поле полностью или частично).*

7.1. Редактирование таблиц с помощью "Конструктора таблиц"

Готовую спецификацию таблицы можно модифицировать, в частности, добавлять (удалять) поля или изменять их параметры. При этом все исправления в спецификацию таблицы следует внести до начала заполнения базы данных, поскольку попытка изменить параметры полей заполненной базы может повлечь за собой потерю или искажение данных.

Для редактирования таблицы необходимо переключиться в режим конструктора  (если таблица открыта в режиме просмотра) или открыть таблицу с помощью конструктора таблиц.

Для вставки нового поля следует установить курсор на строку спецификации, перед которой должна быть введена новая строка, и нажать соответствующую кнопку на панели инструментов   или вызвать команду "Добавить строки" или "Строку" из меню "Вставка".

Если из спецификации таблицы необходимо удалить поле (или несколько полей), то следует выделить соответствующую строку (или строки) и выбрать команду "Удалить строки" из меню "Правка". Строка таблицы выделяется с помощью столбца маркировки. Когда указатель мыши попадает в этот столбец,

он приобретает вид стрелки, указывающей вправо.


Примечание: При удалении поля удаляются все данные, находящиеся в нем. Однако перед удалением данных Access выдает предупреждающее сообщение и предоставляет пользователю возможность отказаться от удаления.

Если вы случайно удалили не ту строку, то отменить удаление можно одним из следующих способов:

1. Нажать кнопку Нет в окне с предупреждающим сообщением.
2. Выбрать команду "Отменить удаление" из меню "Правка". (Однако отменить операцию удаления таким образом можно только в том случае, если она была выполнена последней.)
3. Закрывать окно таблицы и нажать кнопку Нет в диалоговом окне запроса о необходимости сохранения изменений. (Однако в этом случае часть изменений также может не сохраниться.)

7.2. Установка поля первичного ключа

После определения всех полей таблицы следует указать, по меньшей мере, одно поле для использования в качестве поля первичного ключа, если таблица будет иметь связи с другими таблицами.

В большинстве случаев в таблицу включают одно поле первичного ключа. Первичный ключ может быть определен только в режиме конструктора таблиц. Чтобы объявить поле полем первичного ключа, следует установить в него курсор ввода и выполнить щелчок на кнопке с изображением ключа на панели инструментов  или выбрать команду "Первичный ключ" из меню "Правка". Слева от данного поля появится маленькое изображение ключа — признак поля первичного ключа.

Примечание. Проектируя таблицы, следует тщательно выбирать поле первичного ключа. Когда при установлении связи между таблицами используется поле первичного ключа, внести какие-либо изменения в таблицы практически невозможно. Если к моменту выхода из режима конструктора первичный ключ для создаваемой таблицы не объявлен, Access выдаст запрос о необходимости включения в таблицу поля первичного ключа. При положительном ответе Access создаст особое поле с именем ID, в которое для каждого блока данных будет вводиться уникальный номер.

Допускается задание первичного ключа для поля, уже содержащего данные, однако, если в этом поле имеются повторяющиеся значения или значения Null, будет выведено сообщение об ошибке. При получении такого сообщения возможны три варианта действий: выполнить запрос на поиск повторяющихся записей для поиска записей, содержащих повторяющиеся значения или значения Null, а затем изменить их; выбрать другое поле или добавить поле счетчика и определить его как ключевое. Для получения дополнительных сведений о по-

иске повторяющихся значений нажмите кнопку

Для составного ключа существенным может оказаться порядок образующих ключ полей. Сортировка записей осуществляется в соответствии с порядком ключевых полей в бланке в режиме конструктора таблицы. Если необходимо указать другой порядок сортировки без изменения порядка ключевых полей, то сначала определите ключ, как это описано выше, а затем нажатием кнопки Индексы на панели инструментов откройте окно "Индексы" и укажите другой порядок полей для индекса с именем "Первичный ключ".

7.3. Связывание таблиц


Access позволяет строить реляционные базы данных, отдельные таблицы которых могут быть связаны между собой. Связь между таблицами определяет тип отношения между их полями. Как правило, связывают ключевое поле, которое называют полем внешнего ключа, одной таблицы с соответствующим ему полем другой таблицы.

Примечание. *Связанные поля могут иметь разные имена, однако у них должны быть одинаковые типы данных и одинаковые значения свойств.*

При наличии связи между таблицами Access будет автоматически выбирать связанные данные из таблиц в запросах.

Для установки связей между таблицами удобно использовать "Мастер подстановок" в конструкторе таблиц или схему базы данных.

Примечание. *Мастер подстановок не позволяет установить тип связи, но с помощью мастера подстановок удобно подставлять данные из одной таблицы в другую.*

Чтобы установить связь между таблицами для начинающих рекомендуется с помощью мастера подстановок. Откройте в режиме конструктора таблицу, в которую будут подставляться данные. В поле значения, которого будут братья из другой таблицы, в типе данных поля выберите мастер подстановки. Появится окно создание подстановки. Нажмите кнопку "Далее". Выберите таблицу, из которой будут братья данные и нажмите кнопку "Далее". Выберите поле значения, которого являются постанавливаемыми данными, нажмите кнопку ">" (выбранное поле переместиться вправо) и кнопку "Далее" два раза. Нажмите кнопку готово. После этих действий между таблицами установиться связь. Измените связь с помощью схемы данных. Откройте схему данных с помощью кнопки  на панели инструментов или команды "Схема данных" меню "Сервис". Если схема базы данных не создавалась появиться окно добавления таблиц к схеме данных, с помощью которого следует добавить таблицы, после чего закрыть окно. Таблицы представлены на схеме данных в виде прямоугольников с обозначением полей таблицы, размеры и положение этих прямоуголь-

ников можно менять, а так же удалять таблицы из схемы данных (данные не исчезают). Для изменения связи наведите курсор мыши на линию соединяющую таблицы и нажмите правую кнопку мыши. Выберите пункт меню "Изменить связь".

Изменение связей [?] [X]

Таблица/запрос: Связанная таблица/запрос:

Наличие детей	Служащие
Код наличия детей	Код наличия детей

Обеспечение целостности данных
 каскадное обновление связанных полей
 каскадное удаление связанных записей

Тип отношения: один-ко-многим

ОК
Отмена
Объединение...
Новое..

Рис.3. Изменение свойств связи.

Поставьте птички на пунктах "обеспечивать целостность данных" (требуется обязательный ввод данных) и "каскадное обновление связанных полей" (для автоматического обновления связанных полей). Нажмите кнопку "Ок".

Примечание. Поля, через которые осуществляется связь, в главной и связанной таблицах могут иметь разные имена. Необходимым условием установления связи является совпадение типа данных и значений характеристик (в особенности размера).

Между двумя таблицами может быть задано только одно отношение, тип которого можно изменить.

8. Запросы

8.1. Основы языка SQL

Система управления реляционными базами данных System R разрабатывалась в исследовательской лаборатории фирмы IBM в 1975-1979 г.г. Эта работа оказала революционизирующее влияние на развитие теории и практики реляционных систем во всем мире. Именно System R практически доказала жизнеспособность реляционного подхода к управлению базами данных.

После успешного завершения работ по созданию этой системы и получения

экспериментальных результатов ее использования был разработан целый ряд коммерчески доступных реляционных систем, в том числе и на основе непосредственного развития System R (возможности одной из коммерчески доступных реляционных систем - DB2 - описываются в переведенной на русский язык книге К. Дейта "Руководство по реляционной СУБД DB2). Исключительно важен опыт, приобретенный при разработке этой системы. Практически во всех более поздних реляционных СУБД в той или иной степени используются методы, примененные в System R.

После завершения разработки System R фирма IBM активно продолжала работы по реляционным СУБД, причем в нескольких направлениях. Первое направление мы уже отмечали - разработка коммерческих реляционных СУБД. Второе направление - построение распределенной реляционной СУБД на основе идей System R. Экспериментальный вариант такой системы, был успешно разработан в IBM. Эта работа также существенно обогатила опыт исследователей и разработчиков распределенных СУБД. Наконец, третье направление - исследование и разработка реляционных систем, предназначенных для нетрадиционных приложений.

Организации СУБД System R посвящена обширная библиография. Хотя официально разработка этой системы началась в 1975 г., первые публикации, связанные с этой системой, появились еще в 1974 г. В частности, в одной из первых публикаций была предложена основа базового языка System R SQL (тогда этот язык назывался SEQUEL, и до сих пор многие называют его именно так; кстати, разработчики System R (а теперь и компания Oracle) рекомендуют произносить название SQL именно как SEQUEL). Поскольку публикации появлялись по ходу практической реализации системы, каждая из них отражает состояние дел (идейное и практическое) именно на том этапе работы, когда была написана соответствующая статья. Некоторые идеи и представления, естественно, изменялись по ходу работы. Сравнительно законченное представление о системе в целом дают только заключительные публикации. С другой стороны, многие интересные моменты совершенно не отражены в этих последних статьях.

8.2. Основные цели System R и их связь с архитектурой системы

Основными целями разработчиков System R являлись следующие:

- обеспечить ненавигационный интерфейс высокого уровня пользователя с системой, позволяющий достичь независимости данных и дать

возможность пользователям работать максимально эффективно;

- обеспечить многообразие допустимых способов использования СУБД, включая программируемые транзакции, диалоговые транзакции и генерацию отчетов;
- поддерживать динамически изменяемую среду баз данных, в которой отношения, индексы, представления, транзакции и другие объекты могут легко добавляться и уничтожаться без приостановки нормального функционирования системы;
- обеспечить возможность параллельной работы с одной базой данных многих пользователей с допущением параллельной модификации объектов базы данных при наличии необходимых средств защиты целостности базы данных;
- обеспечить средства восстановления согласованного состояния баз данных после разного рода сбоев аппаратуры или программного обеспечения;
- обеспечить гибкий механизм, позволяющий определять различные представления хранимых данных и ограничивать этими представлениями доступ пользователей к базе данных по выборке и модификации на основе механизма авторизации;
- обеспечить производительность системы при выполнении упомянутых функций, сопоставимую с производительностью существующих СУБД низкого уровня.

В основном поставленные цели при разработке System R были достигнуты. Основой System R является реляционный язык SQL. Иногда его называют языком запросов или языком манипулирования данными, но на самом деле его возможности гораздо шире. Средствами SQL (с соответствующей системной поддержкой) решаются многие из поставленных целей. Язык SQL включает средства динамической компиляции запросов, на основе чего возможно построение диалоговых систем обработки запросов. Допускается динамическая параметризация статически откомпилированных запросов, в результате чего возможно построение эффективных (не требующих динамической компиляции) диалоговых систем со стандартными наборами (параметризуемых) запросов.

Средствами SQL определяются все доступные пользователю объекты баз данных: таблицы, индексы, представления. Имеются средства уничтожения любого такого объекта. Соответствующие операторы языка могут выполняться

в любой момент, и возможность выполнения операции данным пользователем зависит от ранее предоставленных ему прав.

Что касается целостности баз данных, то в System R под целостным состоянием базы данных понимается состояние, удовлетворяющее набору сохраняемых при базе данных предикатов целостности. Эти предикаты, называемые в System R условиями целостности (assertions), задаются также средствами языка SQL. Любое предложение языка выполняется в пределах некоторой транзакции – неделимой в смысле состояния базы данных последовательности предложений языка. Неделимость означает, что все изменения, произведенные в пределах одной транзакции либо целиком отображаются в состоянии базы данных, либо полностью в нем отсутствуют. Последняя возможность возникает при откате транзакции, который может произойти по инициативе пользователя (при выполнении соответствующего оператора SQL) или по инициативе системы.

Одной из причин отката транзакции по инициативе системы является как раз нарушение целостности базы данных в результате действий данной транзакции (другие возможные условия отката транзакции по инициативе системы мы рассмотрим позже). Язык SQL содержит средство установки так называемых точек сохранения (savepoint). При иницируемом пользователем откате транзакции можно указать номер точки сохранения, выше которого откат не распространяется. Иницируемый системой откат транзакции производится до ближайшей точки сохранения, в которой условие, вызвавшее откат, уже отсутствует. В частности, откат иницируемый по причине нарушения условия целостности, производится до ближайшей точки сохранения, в которой условия целостности соблюдены.

Для реального выполнения отката транзакции необходимо запоминание некоторой информации о выполнении транзакции. В System R для этих и других целей используется специальный набор данных - журнал, в который помещаются записи обо всех меняющих состоянии базы данных операциях всех транзакций. При откате транзакции происходит процесс обратного выполнения транзакции (undo), в ходе которого в обратном порядке выполняются все изменения, запомненные в журнале.

В языке SQL имеется средство определения так называемых условных воздействий (triggers), позволяющих автоматически поддерживать целостность базы данных при модификациях ее объектов. Условное воздействие - это каталогизированная операция модификации, для которой задано условие ее автомати-

ческого выполнения. Особенно существенно наличие такого аппарата в связи с наличием рассматриваемых ниже представлений базы данных, которыми может быть ограничен доступ к базе данных для ряда пользователей. Возможна ситуация, когда такие пользователи просто не могут соблюдать целостность базы данных без автоматического выполнения условных воздействий, поскольку они просто "не видят" всей базы данных и, в частности, не могут представить всех ограничений ее целостности. Заметим, что, за исключением ранних публикаций по System R, реализация механизма условных воздействий нигде не описывалась, хотя в принципе подходы к реализации достаточно понятны. Этот механизм не реализован в коммерческих системах, возникших на базе System R. Видимо, это связано с возникающими дополнительными непредсказуемыми для пользователей накладными расходами при выполнении транзакций.

Язык SQL содержит средства определения представлений. Представление - это запомненный именованный запрос на выборку данных (из одной или нескольких таблиц). Поскольку SQL - это реляционный язык, то результатом выполнения любого запроса на выборку является таблица, и поэтому концептуально можно относиться к любому представлению как к таблице (при определении представления можно, в частности, присвоить имена полям этой таблицы). В языке допускается использование ранее определенных представлений практически везде, где допускается использование таблиц (с некоторыми ограничениями по поводу возможностей модификации через представления). Наличие возможности определять представления в совокупности с развитой системой авторизации позволяет ограничить доступ некоторых пользователей к базе данных выделенным набором представлений.

Авторизация доступа к базе данных основана также на средствах SQL. При создании любого объекта базы данных выполняющий эту операцию пользователь становится полновластным владельцем этого объекта, т.е. может выполнять по отношению к этому объекту любую функцию из предопределенного набора. Далее этот пользователь может выполнить оператор SQL, означающий передачу всех его прав на этот объект (или их подмножества) любому другому пользователю. В частности, этому пользователю может быть передано право на передачу всех переданных ему прав (или их части) третьему пользователю и т.д. Одним из прав пользователя по отношению к объекту является право на изъятие у других пользователей всех или некоторых прав, которые ранее им были переданы. Эта операция распространяется транзитивно на всех дальней-

ших наследников этих прав.

Наличие в языке средств определения представлений и авторизации в принципе позволяет обойтись при эксплуатации System R без традиционного администратора баз данных, поскольку практически все системные действия производятся на основе средств SQL. Тем не менее, если организационно администратор баз данных требуется, то его работа достаточно упрощается за счет унифицированного набора средств управления. Кроме того, в System R каталоги баз данных поддерживаются также в виде таблиц, и к ним применены все запросы языка SQL. Заметим, что в коммерческих СУБД появился ряд дополнительных утилит, не связанных с языком SQL (например, утилиты сбора статистики или массовой загрузки базы данных), и в этих системах, видимо, без администратора базы данных не обойтись.

По части обеспечения параллельной работы многих пользователей с одной базой данных, основной подход System R состоит в том, что пользователь не обязан знать о наличии других, конкурирующих с ним за доступ к базе данных, пользователей, т.е. система ответственна за обеспечение изолированности пользователей с гарантией отсутствия их взаимного влияния в пределах транзакций. Из этого следует, во-первых, что в интерфейсе пользователя с системой (т.е. в языке SQL) не должно быть средств регулирования взаимодействий с другими пользователями и, во-вторых, что система должна обеспечить автоматическую сериализацию набора транзакций, т.е. обеспечить режим выполнения этого набора транзакций, эквивалентный по конечному результату некоторому последовательному выполнению этих транзакций. Эта проблема решается в System R за счет автоматического выполнения синхронизационных захватов по отношению ко всем изменяемым объектам базы данных.

Одним из основных требований к СУБД вообще и к System R в частности является обеспечение надежности баз данных по отношению к различного рода сбоям. К таким сбоям могут относиться программные ошибки прикладного и системного уровня, сбой процессора, поломки внешних носителей и т.д. В частности, к одному из видов сбоев можно отнести упоминавшиеся выше нарушения целостности базы данных, и автоматический иницируемый системой откат транзакции - это системное средство восстановления базы данных после сбоев такого рода. Как мы отмечали, такое восстановление происходит путем обратного выполнения транзакции на основе информации о внесенных ею изменениях, запомненной в журнале. На информации журнала основано восста-

новление базы данных и после сбоев другого рода. Управление журнализацией и восстановлением в System R весьма интересно, применяемые методы в ряде случаев отличаются от методов, используемых в других СУБД.

Что касается естественных требований к эффективности системы, то здесь основные решения связаны со спецификой физической организации баз данных на внешней памяти, буферизацией используемых страниц базы данных в оперативной памяти и развитой техникой оптимизации запросов, сформулированных на SQL, производимой на стадии их компиляции.

Структурная организация System R вполне согласуется с поставленными при ее разработке целями и выбранными решениями. Основными структурными компонентами System R являются система управления реляционной памятью (Relational Storage System - RSS) и компилятор запросов языка SQL. RSS обеспечивает интерфейс довольно низкого, но достаточного для реализации SQL, уровня для доступа к хранимым в базе данным. Синхронизация транзакций, журнализация изменений и восстановление баз данных после сбоев также относятся к числу функций RSS. Компилятор запросов использует интерфейс RSS для доступа к разнообразной справочной информации (каталоги отношений, индексов, прав доступа, условий целостности, условных воздействий и т.д.) и производит рабочие программы, выполняемые в дальнейшем также с использованием интерфейса RSS. Таким образом, система естественно разделяется на два уровня - уровень управления памятью и синхронизацией, фактически, не зависящий от базового языка запросов системы, и языковый уровень (уровень SQL), на котором решается большинство проблем System R. Эта независимость скорее условная, чем абсолютная: язык SQL можно заменить на другой язык, но он должен обладать примерно такой же семантикой.

Далее рассмотрим SQL в MS Access.

8.3. Проектирование запросов

Запросы предназначены для просмотра, изменения и анализа данных. Они используются также в качестве источника записей при создании форм и отчетов т.п. Одним из наиболее распространенных запросов является запрос на выборку, который выполняет отбор данных из одной или нескольких таблиц в соответствии с заданными пользователем критериями. Запросы предназначены для отбора данных, удовлетворяющих заданным критериям (например, данные о клиентах, *проживающих в определенном регионе*, данные о торговых операциях за *последний квартал*). Результатом выполнения запроса является набор

записей, собранных в таблице, который называется Recordset (динамический, временный набор данных). В объекте Recordset допускается добавление, изменение и удаление записей. В нем отображаются также записи, добавляемые, удаляемые или изменяемые в исходных таблицах.

В Access 97 и 2000 запросы делятся на QBE-запросы (Query By Example — запрос по образцу), параметры которых устанавливаются в окне конструктора запросов, и SQL-запросы (Structured Query Language — структурированный язык запросов), при создании которых применяются операторы и функции языка SQL. В Access QBE-запрос соответствует SQL-запросу. Поэтому при запуске QBE-запроса выполняется соответствующий SQL-запрос.

8.4. QBE-запросы

Один из наиболее распространенных типов QBE-запроса — запрос на выборку. Этот запрос позволяет выбрать данные из таблицы или таблиц по определенному критерию QBE-запросе.

Чтобы создать запрос с окне базы данных откройте закладку "Запросы".

Запрос можно создавать в режиме конструктора и с помощью мастера. Выберите с помощью конструктора. Для Access 97 выберите создать новый запрос "Конструктор запроса".

В результате появятся пустое окно конструктора запроса и диалоговое окно "Добавление таблицы".

Диалоговое окно "Добавление таблицы" состоит из трех вкладок — "Таблицы", "Запросы" и "Таблицы и Запросы". На них осуществляется выбор таблиц и запросов, используемых при разработке запроса. Перейдите на вкладку "Таблицы", выделите необходимую таблицу и выполните щелчок на кнопке "Добавить". Закройте диалоговое окно, щелкнув на кнопке "Закреть". Имена таблиц появятся в окне проектирования запроса.

***Примечание.** Посредством команды "Добавление таблицы" из меню "Запросы" можно снова открыть диалоговое окно "Добавление таблицы" и пополнить список таблиц запроса.*

Окно конструктора запроса разделено на две части. В верхней половине находятся окна таблиц со списками полей. Имя каждой таблицы отображается в строке заголовка такого окна.

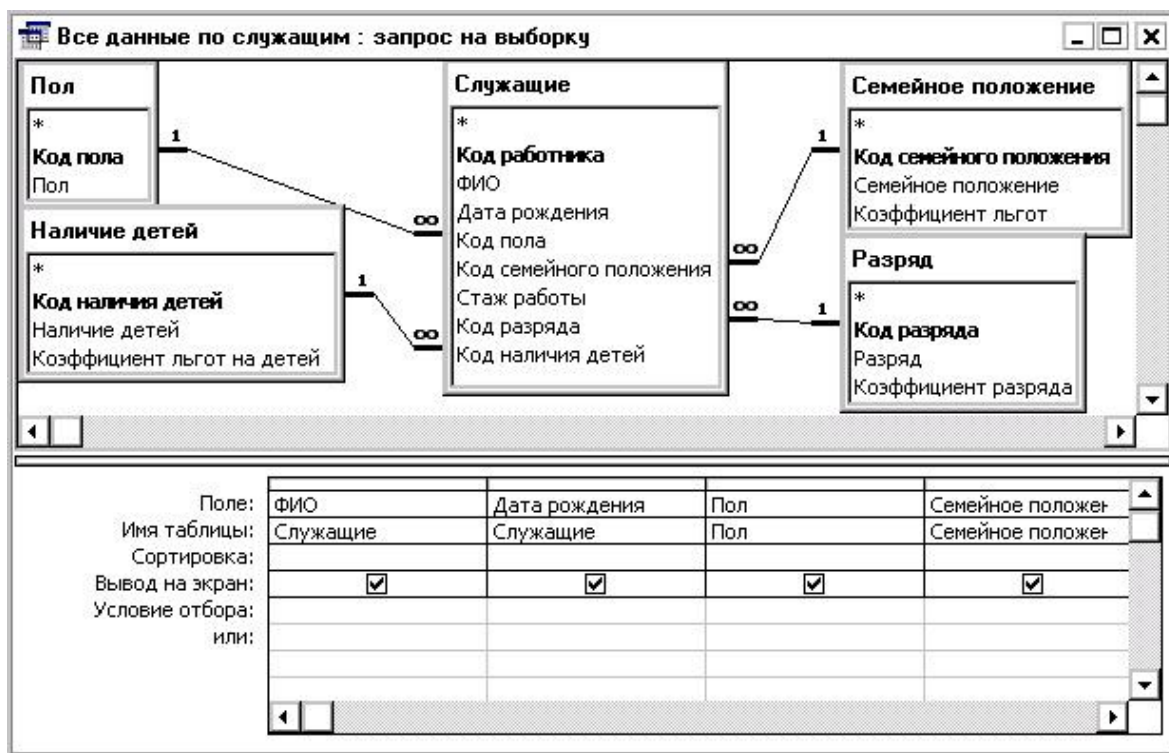


Рис.4. QBE-запрос в режиме конструктора

При составлении запроса на основе нескольких таблиц между ними необходимо установить связи (если их нет), задав отношение между полями. В противном случае результат обработки запроса может оказаться некорректным.

Разработка запроса выполняется в несколько этапов:

1. Добавление полей в запрос.
2. Установка критериев отбора записей.
3. Сортировка записей.

8.5. Добавление полей в запрос

В запрос не следует включать все поля выбранных таблиц, а только необходимые.

Добавить нужные поля в бланк запроса можно путем перетаскивания их имен из списка, находящегося в верхней части окна конструктора, в строку "Поле" бланка. Еще один способ – двойной щелчок на имени поля.

Первым полем запроса следует делать наиболее важную информацию (положение полей влияет на сортировку) далее по мере уменьшения значимости.

Ключевые поля, если выводятся, так же следует делать первыми.

В процессе разработки большинства запросов используется часть полей таблиц. Иногда требуется включить в запрос все поля таблицы. Это можно выпол-

нить несколькими способами:

1. Маркируйте все поля, выполнив двойной щелчок на строке заголовка таблицы в верхней части окна конструктора запроса, и перетащите их в первую строку бланка запроса. Access автоматически разместит каждое поле таблицы в отдельной колонке.

2. Маркируйте символ * в списке полей таблицы и перетащите его в бланк запроса. В результате все поля таблицы будут включены в запрос, но в бланке появится только запись *ИмяТаблицы. **.

Оба способа дают одинаковый результат. Однако второй имеет следующее преимущество: когда в исходной таблице добавляются или удаляются поля, автоматически изменяется и запрос.

8.6. Удаление полей и таблиц из запроса

Для удаления поля из бланка запроса следует маркировать колонку, в которой оно находится, и нажать клавишу [Del] или активизировать в меню "Правка" команду "Удалить столбцы".

Чтобы удалить из запроса таблицу, надлежит сначала маркировать ее в верхней части окна конструктора запроса, выполнив щелчок на имени любого из ее полей, а затем нажать клавишу [Del] или вызвать команду "Удалить" из меню "Правка". Поля удаленной из запроса таблицы, которые были помещены в бланк запроса, удаляются автоматически.

8.7. Установка критериев отбора записей

Посредством критериев отбора пользователь указывает Access, какие записи нужно выбирать в таблицах и отображать в таблице результатов выполнения запроса. Критерии отбора указываются для одного или нескольких полей.

Для установки критерия отбора выберите столбец с необходимым полем и установите в строке "Условие отбора" логическое условие с операторами, представленными в таблице 3.

Таблица 3. Операторы условий отбора

Оператор	Значение
<	меньше
>	больше
<=	меньше равно
>=	больше равно
=	равно
Not	не равно
And	и – объединяет два условия

При вводе данных сравнения следует учитывать тип поля (Нельзя сравнить разные типы данных). Числовые типы данных просто вводятся через пробел после условия. Строковые данные вводятся в кавычках. Данные времени заключаются с помощью символа решетки #.

8.8. Сортировка данных в запросе

Записи в генерируемом при выполнении запроса динамическом наборе данных Recordset могут быть рассортированы по алфавиту, по возрастанию или по убыванию. Можно также одновременно производить сортировку по содержанию нескольких полей (до десяти). Для выполнения сортировки следует перейти в столбец бланка запроса, содержащий имя поля, по которому необходимо произвести сортировку, и указать способ сортировки в строке "Сортировка". При выполнении щелчка на этой строке появляется поле списка с перечнем способов сортировки. Значение (отсутствует) позволяет отказаться от сортировки по содержимому данного поля.

***Примечание.** При закрытии окна конструктора запроса Access перемещает поля с заполненной ячейкой "Сортировка" и группирует их в левой части бланка запроса.*

8.9. Отмена отображения полей в запросе

В случае необходимости можно отменить отображение поля в результирующем наборе записей (даже того, по которому производится сортировка). Поле отображается в таблице результатов запроса, если в соответствующей колонке бланка запроса, в строке Вывод на экран включен контрольный индикатор. Для отмены отображения поля достаточно снять птичку со строки "Выводить на экран".

***Примечание.** Ненужные поля лучше удалять из запроса.*


Для полей запроса которые используются в вычислениях, но не нужны в отображении лучше отменить вывод на экран. Если их удалить будут неправильные вычисления.

8.10. Вычисления в запросе

Запрос можно использовать для выполнения расчетов и подведения итогов, обобщив данные из исходных таблиц. Для этих целей в Access предусмотрены статистические функции SQL (табл. 4). Статистическую функцию задают в строке "Групповая операция", которая появляется после нажатия кнопки с греческой литерой сигма, расположенной на панели инструментов. Посредством статистической функции можно обработать содержимое каждого поля запроса. Результат обработки появляется в результирующем наборе записей запроса.

Таблица 4. Статистические функции SQL

Функция	Действие
Sum	Суммирование значений определенного поля, находящихся в записях, которые отображены запросом
Avg	Вычисление среднего значения в тех записях определенного поля, которые отображены запросом
Min	Вычисление минимального значения в записях определенного поля, отображенных запросом
Max	Вычисление максимального значения в записях определенного поля, отображенных запросом
Count	Вычисление количества записей, отображенных запросом, в определенном поле
First	Определяется первое значение в указанном поле записей, отображенных запросом
Last	Определяется последнее значение в указанном поле записей, отображенных запросом
StDev	Вычисляется стандартное отклонение значений данного поля для всех записей, отображенных запросом
Var	Вычисляется вариация значений данного поля для всех записей, отображенных запросом

Кроме этого в запросе можно создавать вычисляемые поля. Для этого необходимо перейти в строку "Поле" и запустить "Построитель" (кнопка ). В "Построителе" необходимо набрать имя вычисляемого поля, поставить двоеточие ":" и вычисляемое выражение.

Исходными данными могут служить значения полей текущего запроса (окошко по середине под полем ввода). Для этого необходимо выделить поле и нажать кнопку "Вставить". Можно использовать большое количество встроенных функций Access. Это математические, статистические, финансовые, строковые, временные, управления и т.д.

Примечание. Чтобы в новом вычисляемом поле появилось предыдущее (введенное) вычисляемое поле, необходимо сохранить запрос.

Пример: Налог: [Зарплата]*1,2

Примечание. Если в запросе выполняются "Групповые операции", то все операции вычисляемые в запросе должны быть статистическими функциями. Поэтому если есть необходимость выполнения других операций, то их необходимо перенести в другой запрос.

9. Построитель выражений

Окно построителя выражений состоит из трех разделов.



Рис.5. Построитель выражений.

В верхней части окна построителя расположено поле выражения. Ниже находится раздел, предназначенный для создания элементов выражения и их последующей вставки в поле выражения. Допускается непосредственный ввод части выражения в поле выражения.

В средней части окна построителя находятся кнопки с часто используемыми операторами. При нажатии на одну из этих кнопок построитель вставит соответствующий оператор в текущую позицию поля выражения. Чтобы вывести полный список операторов, выберите папку **Операторы** в нижнем левом поле и нужный тип в среднем поле. В правом поле будут выведены все операторы выбранного типа.

В нижней части окна построителя находятся три поля.

В левом поле выводятся папки, содержащие таблицы, запросы, формы, объекты базы данных, встроенные и определенные пользователем функции, константы, операторы и общие выражения.

В среднем поле задаются определенные элементы или типы элементов для папки, заданной в левом поле. Например, если выбрать в левом поле **Встроенные функции**, то в среднем поле появится список всех типов функций Microsoft Access.

В правом поле выводится список значений (если они существуют) для элементов, заданных левым и средним полями. Например, если выбрать в левом поле **Встроенные функции** и тип функции в среднем, то в правом поле будет выведен список всех встроенных функций выбранного типа.


***Примечание.** Чтобы новые имена полей появились в строителе выражений, следует предварительно сохранить таблицу или запрос, содержащие такие поля.*

Если функция или объект не выводятся в нижней части строителя выражений, то это значит, что их использование недопустимо в позиции, из которой был вызван строитель. Например, нельзя ссылаться на другое поле или элемент управления в условии на значение для поля в режиме конструктора таблицы, таким образом папки **Таблицы**, **Запросы**, **Формы** и **Отчеты** не доступны при запуске строителя выражений из ячейки для свойства поля **Условие на значение (ValidationRule)** в режиме конструктора таблицы.

10. Создание запросов других типов

Мы рассмотрели запрос на выборку. В Access существуют и запросы других типов: запросы действия, параметрические запросы и перекрестные запросы.

10.1. Запрос на изменение

Запросы на изменение используют для изменения и переноса данных в таблицах, для обновления, добавления и удаления групп записей, а также для создания новой таблицы на основе одной или нескольких таблиц. Различают четыре типа запросов на изменение: запрос на добавление, запрос на удаление, запрос на обновление и запрос на создание таблицы. Выбор типа запроса производится с помощью кнопки  ▾.

Примечание: В окне базы данных перед именем запроса на изменение Access устанавливает восклицательный знак (!).

10.2. Запрос на добавление

С помощью запроса на добавление записи из одной таблицы (все или отобранные запросом) можно поместить в конец другой таблицы. Обе таблицы могут находиться как в одной, так и в разных базах данных.

Если необходимо добавить записи к таблице из другой базы данных, сначала следует присоединить таблицу-источник к базе, содержащей целевую таблицу, с помощью команды "Связать таблицы" из подменю "Получить внешние данные" меню "Файл". Для отбора добавляемых записей нужно составить запрос на выборку. Затем надлежит выполнить составленный запрос и оценить результат, переключившись в режим таблицы с помощью команды "Режим таблицы"

из меню "Вид". После этого необходимо вернуться в режим конструктора и активизировать команду "Запрос добавления" из меню "Запросы". В открывающемся при этом диалоговом окне пользователь должен задать в поле "Имя таблицы" имя таблицы, к которой он хочет присоединить данные из результирующего набора записей запроса. Закрытие диалогового окна посредством кнопки ОК приводит к тому, что Access добавляет в бланк запроса строку "Присоединить к". В эту строку автоматически (или вручную) вставляются имена тех полей целевой таблицы, которые совпадают с именами полей запроса. Далее следует выполнить запрос, нажав кнопку "Запуск" на панели инструментов. В специальном диалоговом окне Access укажет, сколько записей будет добавлено к целевой таблице, и потребует подтвердить выполнение этой операции.

10.3. Запрос на удаление

Запросы этого типа служат для удаления из таблицы групп записей, соответствующих некоторому критерию отбора. Поскольку записи, удаленные посредством запроса, нельзя восстановить, следует тщательно анализировать критерии отбора.

10.4. Запрос на обновление

Используя запрос на обновление, пользователь может изменить группу записей, отобранную на основе определенных критериев. При разработке запроса на обновление, прежде всего, создается и проверяется запрос на выборку. После этого в режиме конструктора активизируется команда "Запрос на обновление" из меню "Запросы", вследствие чего Access добавляет в бланк запроса строку "Обновление", которая предназначена для указания новых значений полей таблицы. В качестве таковых могут выступать и вычисляемые выражения. После нажатия кнопки "Запуск" Access укажет в специальном диалоговом окне, сколько записей изменится в таблице, и потребует подтвердить изменения.

10.5. Запрос на создание таблицы

На основе результирующего набора записей запроса можно построить новую таблицу с помощью запроса на создание таблицы. Такие запросы обычно применяют для архивирования старых записей или для сохранения резервных копий таблиц.

Сначала необходимо подготовить запрос на выборку и получить динамический набор данных Recordset, заодно проверив корректность запроса. Если результирующий набор записей соответствует вашим требованиям, вернитесь в

режим конструктора и в меню "Запросы" выберите команду "Запрос создания таблицы". Access откроет диалоговое окно "Создание таблицы", в которое следует ввести имя новой таблицы. Далее нужно выполнить запрос, нажав кнопку "Запуск" (кнопка с восклицательным знаком на панели инструментов). В специальном диалоговом окне Access укажет, сколько записей добавляется в новую таблицу, и потребует подтвердить выполнение этой операции.

10.6. Изменение запроса

Изменения в результирующем наборе записей запроса (например, перенос колонки) не отражаются в его спецификации. При повторном выполнении запроса перенесенная колонка появится в исходной позиции. Изменять параметры запроса надлежит в режиме конструктора.

В режиме конструктора запроса можно выполнить такие изменения:

- задать новые критерии;
- изменить способ сортировки;
- отменить или задать отображение поля;
- добавить, удалить или изменить вычисляемые выражения;
- добавить, удалить или перенести поля.

10.7. Сохранение запросов

Если пользователь хочет применять новый или измененный запрос в течение нескольких сеансов работы с Access, он должен сохранить его с помощью команды "Сохранить" из меню "Файл". При первом сохранении запроса открывается диалоговое окно "Сохранить как" для присвоения запросу имени, под которым он будет фигурировать в списке запросов в окне базы данных. Рекомендуется сохранять сложные запросы, содержащие много полей, критериев и функций. Обратите внимание на то, что сохраняется не результат выполнения запроса, а спецификация. Динамический набор данных Recordset ликвидируется при закрытии запроса, поэтому его следует сохранять отдельно также с помощью команды "Сохранить" из меню "Файл".

10.8. Редактирование запроса

Редактирование запроса осуществляется в режиме конструктора.

11. SQL-запросы

С помощью языка структурированных запросов SQL, реализованного в Access, можно составить любое число сложных запросов. Этот язык позволяет

также управлять обработкой запросов. SQL-запрос представляет собой последовательность инструкций, в которую могут входить выражения и статистические функции SQL.

Начинающим пользователям рекомендуется создавать запросы в окне конструктора, поскольку для описания запросов на языке SQL необходим опыт работы с базами данных. Примерами SQL-запросов являются запросы на объединение, запросы к серверу, управляющие и подчиненные запросы. Некоторые запросы не могут быть сформированы в окне QBE-запроса. Разработка запросов на объединение, запросов к серверу и управляющих запросов осуществляется только в окне запроса в режиме SQL. SQL-запросы могут служить не только для извлечения информации из базы данных, но и для получения сведений для формы и отчета, а также для заполнения полей списков в формах.

ПРИЛОЖЕНИЕ 2. МОЛОЧНО-ТОВАРНАЯ ФЕРМА

12. Математическая модель продуктивности коров.

Многие вопросы технического и технологического обеспечения животноводства сталкиваются с необходимостью формализации отклика животного на условия содержания и кормления. В случае молочного скотоводства функциональную зависимость удоя от расхода корма представим в следующем виде:

$$У_{д} = \begin{cases} 0, & \text{при } Дз \leq Дз_{\text{ПОД}} \\ K_1 \cdot (Дз - Дз_{\text{ПОД}}), & \text{при } Дз_{\text{ПОД}} \leq Дз \leq Дз_{\text{ПЕР}} \\ У_{д_{\text{ПОТ}}} - K_2 \cdot (Дз_{\text{ПОТ}} - Дз)^2, & \text{при } Дз_{\text{ПЕР}} \leq Дз \leq Дз_{\text{ПОТ}} \\ У_{д_{\text{ПОТ}}}, & \text{при } Дз \geq Дз_{\text{ПОТ}} \end{cases} \quad (1)$$

где: $У_{д}$ – годовой удой коровы, пересчитанный на жирность 4,0 %, л; $Дз$ – годовой расход корма, в кормовых единицах (к. ед.)²; $Дз_{\text{ПОД}}$ – поддерживающая доза корма, к. ед.; $Дз_{\text{ПЕР}}$ – доза корма, при которой наступает снижение эффективности молокоотдачи, к. ед.; $Дз_{\text{ПОТ}}$ – доза корма, при которой достигается текущий потенциал продуктивности коровы, к. ед.; K_1 – коэффициент на линейном участке зависимости (рис.1), по данным Л. Фредериксона равен 2.5 л/к. ед.; K_2 – коэффициент на параболическом участке зависимости (рис.1.).

Связь продуктивности (годового удоя) гипотетической коровы с расходом корма имеет три характерные точки (рис.1). Если доза кормления коровы не обеспечивает требуемой величины энергии на жизнеобеспечение ($Дз \leq Дз_{\text{ПОД}}$), то удой равен нулю. Затем следует линейный участок нарастания удоя, где прирост расхода корма на одну кормовую единицу обеспечивает прирост 2.5 литров молока.

Но в некоторой точке, разной для коров, отличающихся потенциальными возможностями, эффективность молокоотдачи начинает постепенно уменьшаться ($Дз \geq Дз_{\text{ПЕР}}$). Это уменьшение описывается параболической зависимостью. Затем наступает достижение предельной продуктивности коровы ($Дз = Дз_{\text{ПОТ}}$). Дальнейшее увеличение расхода корма не приводит к увеличению удоев.

² Кормовой единицей называется такое количества корма которое по питательности эквивалентно 1 кг овса. Если корм более питательный чем овес то доза в к. ед. будет больше единицы, если менее питательный – меньше единицы.

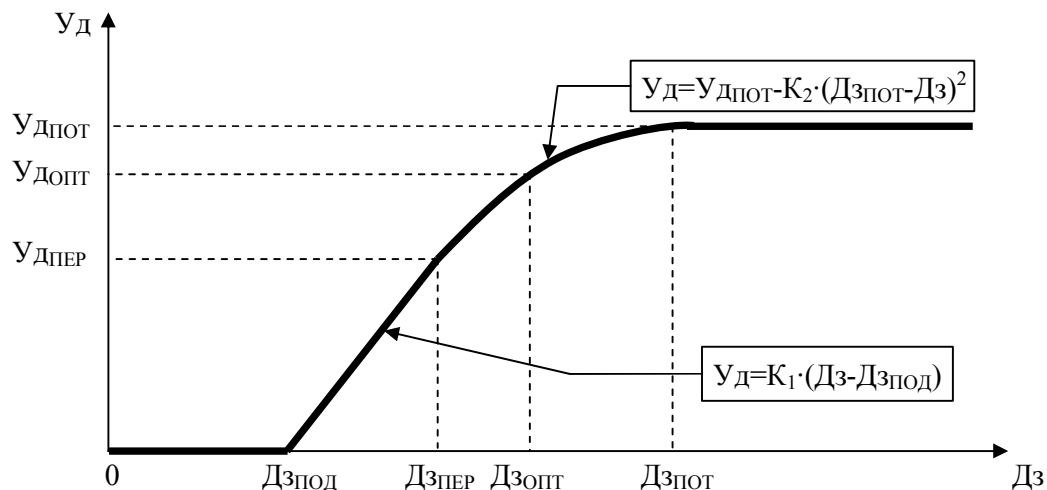


Рис.1. Зависимость годовой продуктивности молочной коровы $Уд$ от годового потребления корма $Дз$.

Модель (1) содержит еще одно предположение, что точка начала насыщения описывается уравнением

$$Уд = 1,875 \cdot (Дз_{ПОТ} - Дз_{ПОД}) \quad (2)$$

Если конкретная характеристика животного выше (рис.2), что соответствует более высокому потенциальному удою, то оно обладает уникальными данными, а если ниже, то, очевидно, наблюдается неэффективная трата корма.

Коэффициент $К_2$ определяется из условия непрерывности функции отклика (1)

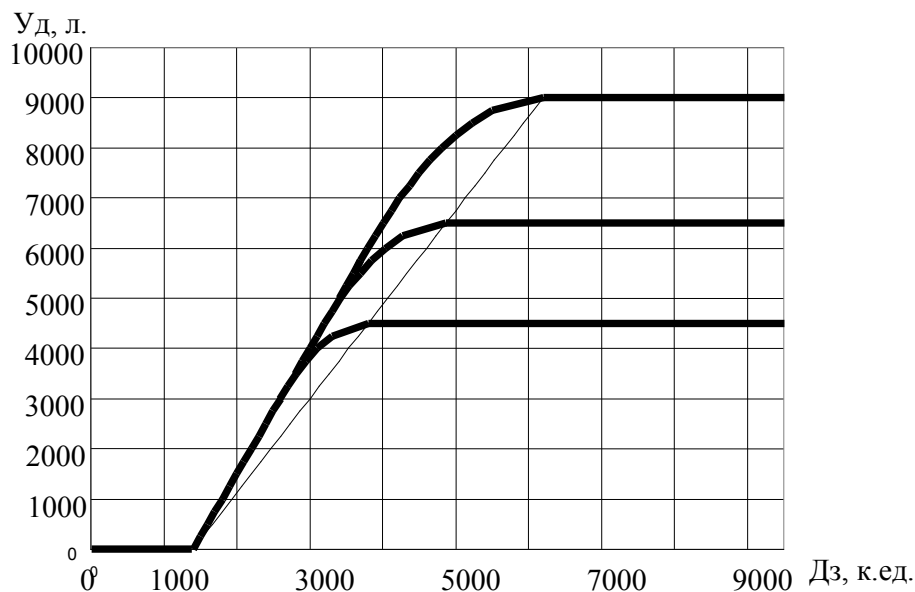


Рис.2. Количественная связь удоя и расхода корма по Л.Фредериксену

и ее первой производной в точке перегиба

$$\begin{cases} 2,5 \cdot (Дз_{ПЕР} - Дз_{ПОД}) = У_{ДПОТ} - К_2 \cdot (Дз_{ПОТ} - Дз_{ПЕР})^2 \\ 2,5 = 2 \cdot К_2 \cdot (Дз_{ПОТ} - Дз_{ПЕР}) \end{cases} \quad (3)$$

Подставляя (2) в (3) получим

$$К_2 = \frac{2,5}{Дз_{ПОТ} - Дз_{ПОД}} \quad (4)$$

Т.е., зная текущий потенциал животного ($У_{ДПОТ}$ или $Дз_{ПОТ}$) можно определить все параметры модели (1).

13. Оптимизация производства молока

Оптимизация производства молока по минимуму затрат корма на единицу продукции в рамках модели (1) определяется условием оптимальности

$$\frac{У_{Д}}{Дз} \rightarrow \max \quad (5)$$

Оптимальный удой по минимуму затрат корма на единицу продукции будет достигнут при дозе кормления соответствующей максимуму отношения (5).

Для определения оптимального удоя по этому критерию необходимо найти точку изменения знака роста отношения (5). Эта точка лежит на параболической зависимости удоя от дозы (рис.1.). Чтобы найти эту точку необходимо подставить (1) и (4) в (5), с полученного выражения найти первую производную и приравнять нулю

$$\frac{У_{Д}}{Дз} = \frac{У_{ДПОТ} - \frac{2,5 \cdot (Дз_{ПОД} - Дз)^2}{Дз_{ПОТ} - Дз_{ПОД}}}{Дз}$$

$$\frac{dУ_{Д}/Дз}{dДз} = 0$$

В вычисленное выражение подставляя (2) получается выражение **оптимальной дозы по максимуму удоя и минимуму затрат**

$$Дз_{ОПТ} = \sqrt{Дз_{ПОТ}^2 - 0,75 \cdot (Дз_{ПОТ} - Дз_{ПОД})^2} \quad (6)$$

Критерием оптимальность может быть максимум прибыли. Прибыль условно

можно подсчитать по формуле

$$\Pi = Дх - Зт = Уд \cdot Ц_{\text{М}} - 2,5 \cdot Дз \cdot Ц_{\text{К}} \quad (7)$$

где: Дх – доход; Зт – затраты; Уд – годовой удой коровы, пересчитанный на жирность 4,0 %, л; Ц_М – цена 1 л. молока, руб/л; Дз – годовое потребление корма, к. ед.; Ц_К – цена одной кормовой единицы руб/к. ед.; 2,5 – предположительный коэффициент из расчета, что затраты корма составляют 40% от общих затрат на содержание коровы.

Тогда условие оптимальности $\Pi \rightarrow \max$.

Доза при которой достигается оптимальный удой по критерию максимум прибыли вычисляется по формуле

$$Дз_{\text{ОПТ}} = Дз_{\text{ПОТ}} - 0,5 \cdot \frac{Ц_{\text{К}}}{Ц_{\text{М}}} \cdot (Дз_{\text{ПОТ}} - Дз_{\text{ПОД}}) \quad (8)$$

14. Способы кормления животных

Существует два способа кормления животных - групповое и индивидуальное дозирование корма.

При индивидуальном дозировании корма каждой корове дают дозу корма в зависимости от ее потенциальных возможностей (потенциального удоя) и удоя какой хотят получить.

При групповом кормлении коров разбивают на группы, дозу определяют по средней корове в группе и всем коровам в группе раздают одинаковую среднюю дозу.

Групповое кормление требует меньших капитальных вложений и эксплуатационных затрат.

ПРИЛОЖЕНИЕ 3. ОТКОРМ СВИНЕЙ

15. Математическая модель откорма свиней.

В создаваемой модели откорма свиней учитывается поддерживающая доза корма и зона насыщения привесов, нестационарность процессов.

В качестве зависимости привесов от дозы принимается параболическая модель откорма (рис.1.).

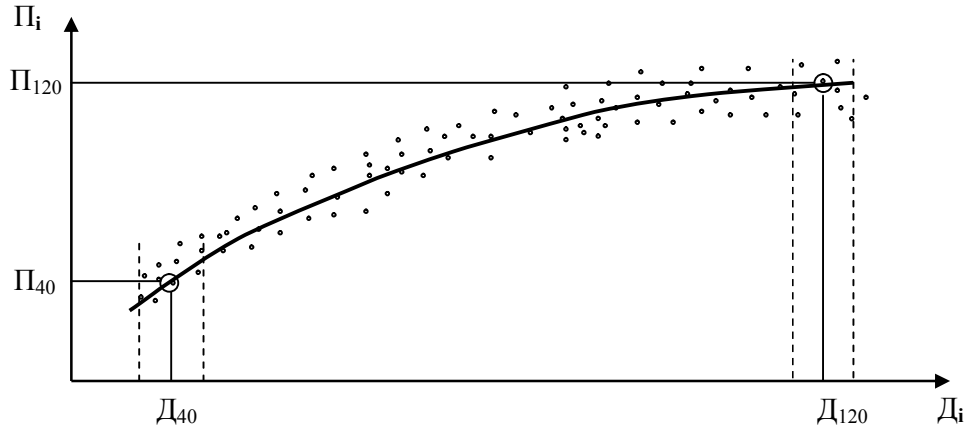


Рис.1. Зависимость привесов от дозы

Зависимость привесов от расхода корма можно представить в виде

$$P_i = \begin{cases} P_{nom,i} - a \cdot (D_{nom,i} - D_i)^k, & \text{если } D_i \leq D_{nom,i} \\ P_{nom,i}, & \text{если } D_i > D_{nom,i} \end{cases} \quad (1)$$

где P_i – привес в i сутки; $P_{pot,i}$ - потенциальный привес в i сутки; D_i – доза корма в i сутки; $D_{pot,i}$ - потенциальная доза корма в i сутки; a, k – коэффициенты отражающие зависимость привеса от дозы корма. Как видно из формулы (1) привес зависит от дозы в данные и предыдущие сутки. На практике зависимость привеса P_i от дозы D_i не четко выражена, так как на привес могут влиять и другие, не учитываемые факторы (рис.1.). Кроме того, точность выдаваемой дозы и точность измерения привесов всегда определяется с погрешностью.

Значения поддерживающей и потенциальной доз $D_{под,i}$, $D_{пот,i}$ зависят от массы животного в i сутки. Эти зависимости имеют параболический характер с насыщением, если животного кормить постоянной дозой $D_i = const$. Если доза D_i будет изменяться зависимости поддерживающей и потенциальной доз от текущей

дозы D_i – примут другой характер.

Для определения зависимости потенциальной дозы $D_{\text{пот},i}$ от текущей дозы D_i – животное следует постоянно перекармливать, $D_i > D_{\text{пот},i}$. Так как это сложно экспериментально проверить и вычислить, то в лабораторной работе будем считать, что эти зависимости определены.

$$D_{\text{под}} = \begin{cases} D_{\text{под,макс}} - 0,004 \cdot (120 - M_i)^{1,3}, & \text{если } 0 < M_i \leq 120 \\ D_{\text{под,макс}}, & \text{если } M_i > 120 \end{cases} \quad (2)$$

$$D_{\text{ном}} = \begin{cases} D_{\text{ном,макс}} - 0,004 \cdot (150 - M_i)^{1,5}, & \text{если } 0 < M_i \leq 150 \\ D_{\text{ном,макс}}, & \text{если } M_i > 150 \end{cases} \quad (3)$$

$$P_{\text{ном}} = P_{\text{ном,макс}} - 4 \cdot 10^{-8} \cdot (200 - M_i)^{3,2}, \text{ если } 0 < M_i \leq 400 \quad (4)$$

где M_i – текущая масса животного.

Для определения коэффициентов формулы (1) примем следующие допущения $D_{\text{пот}} = D_{120}$, $P_{\text{пот}} = P_{120}$ и $\kappa = 2$. Тогда задача сводится к определению коэффициента a .

$$a = \frac{P_{120} - P_{40}}{(D_{120} - D_{40})^2} \quad (5)$$

16. Критерий оптимальности производства свинины

Критерием оптимальности откорма свиней может быть максимум прибыли, максимум привесов, максимум привесов на единицу корма.

По критерию максимум привесов поросенка нужно кормить потенциальной дозой.

По критерию максимум привесов на единицу корма оптимальная доза должна определяться из условия

$$\frac{d\left(\frac{\text{Пр}}{\text{Дз}}\right)}{d\text{Дз}} \rightarrow \text{max}$$

и будет равна

$$\text{Дз}_{\text{ОПТ}} = \sqrt{D_{\text{ПОТ}}^2 - \frac{P_{\text{ПОТ}}}{a}} \quad (6)$$

17. Способы кормления

Кормление на свиноводческом комплексе частично автоматизировано. Автоматизирована раздача и приготовление корма. Доза корма определяется зоотехником и задается системе приготовления и раздаче оператором. Т.е. доза корма задается вручную на основании времени откорма и планируемых привесов. Периодически (один раз в несколько дней) взвешивается контрольная группа животных, для определения соответствия планируемых и реальных привесов.

При полной автоматизации кормления система управления приготовления и раздачи должна определять дозу корма сама. Значит, создатель системы управления должен заложит вычисление дозы корма. С точки зрения качества управления доза корма должна быть оптимальной, по одному из критериев.

Для того чтобы система управления определяла дозу корма, она должна автоматически взвешивать поросят. Чем чаще СУ будет взвешивать поросят, тем точнее будет определяться доза корма и управление будет стремиться к оптимальному.

18. Алгоритмы функционирования и управления

Алгоритм функционирования – алгоритм, определяющий развитие во времени процессов, связанных с функционированием некоторого объекта.

В работе алгоритмом функционирования это процесс кормления.

Алгоритм управления – алгоритм, формулирующий процесс управления некоторым объектом. Т.е. это последовательность действий приводящих к цели управления.

ПРИЛОЖЕНИЕ 4. ПРОИЗВОДСТВО ПТИЦЫ

19. Особенность производства птицы

Управления современным сельскохозяйственным производством является достаточно сложным, трудоемким и наукоемким процессом. Сельскохозяйственное производство состоит из определенных технических устройств и биологических объектов, оптимальное функционирование которых по разным критериям оптимальности позволит получить наибольшую выгоду, но разную прибыль. Основными критериями производства сельскохозяйственной продукции являются максимум производства или максимум прибыли. Чтобы добиться наилучших показателей необходимо: во-первых знать объект в виде технико-биолого-экономической математической модели; во-вторых использовать современные знания оптимизации для принятия решений по управлению производством и созданию сложных систем управления. Для анализа производственных моделей целесообразно на первом этапе исследований представить все производство как один объект, разделенный на несколько технологических процессов с построением их математических моделей. Сложность в построении математических моделей сельскохозяйственного производства связана с наличием в процессе производства биологического объекта, который является сложной системой (многомерной, существенно нелинейной, нестационарной, с распределенными параметрами). Поэтому для исследования таких объектов применяется современные методы проведения эксперимента (метод пассивного и активного эксперимента).

Подготовка к проведению эксперимента начинается с предварительного изу-

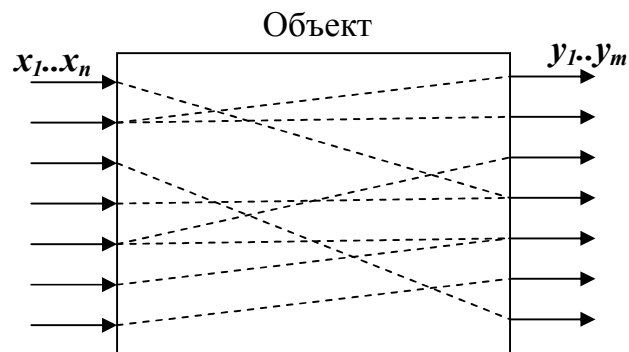


Рисунок 1. Структурная схема объекта

чения конструкции и технологических режимов работы объекта; при этом выявляются основные входные возмущающие и регулирующие воздействия, а так

же выходные регулируемые и контролируемые величины. Кроме того, определяются допустимые пределы отклонения выходных величин и уровень естественных шумов. На основании предварительных данных составляют структурную схему объекта, на которой изображают входные $x_1..x_n$ и выходные $y_1..y_m$ величины объекта и каналы связи между ними (рис.1.). Далее одним из методов

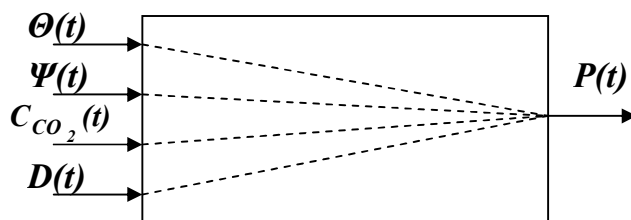


Рисунок 2. Структурная схема процесса производства птицы где $\Theta(t)$ – текущая температура; $\Psi(t)$ – текущая влажность; $C_{CO_2}(t)$ – текущая концентрация CO_2 ; $D(t)$ – текущая доза корма; $P(t)$ – текущий прирост птицы.

проведения эксперимента определяю функции связей выходных параметров от входных.

В процессе производства птицы на продуктивность птицы может влиять несколько параметров:

- параметры микроклимата;
- свойства корма и его доза;
- случайные внешние возмущения.

В нашем случае исключим случайные воздействия, число параметров микроклимата сократим до трех (температура, влажность, концентрация CO_2), свойства корма будем считать постоянными. Структурная схема будет иметь представленный на рисунке 2.

Если учесть экономическую сторону процесса производства птицы, то выходном параметр модели станет экономический показатель - прибыль $Pr(t)$. Прибыль будет зависеть от параметров процесса производства – количества энергии для поддержания температуры, дозы корма и количества выращенной продукции. Так же прибыль зависит от экономических показателей - затраты на отопительно-вентиляционную систему (ОВС), затраты на корм, цены на продукцию, т.е. стоимостной стороны производственных показателей (рис.3.).

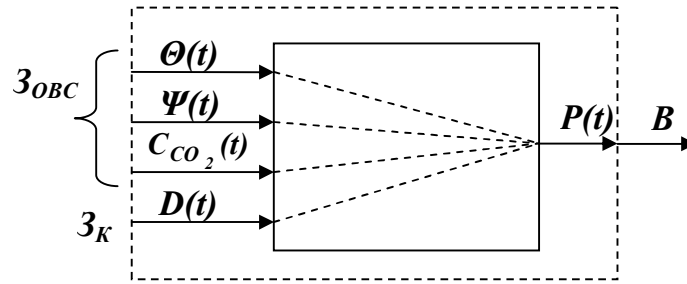


Рисунок 3. Структурная схема экономической модели процесса производства птицы

где $\Theta(t)$ – текущая температура; $\Psi(t)$ – текущая влажность; $C_{CO_2}(t)$ – текущая концентрация CO_2 ; $D(t)$ – текущая доза корма; $P(t)$ – текущий прирост птицы; Z_{OBC} – затраты на ОВС; Z_K – затраты на корм; B – выручка.

Предельное положение эмпирической линии регрессии, к которому она стремится при неограниченном увеличении числа наблюдений и одновременном уменьшении интервалов измерений, называется теоретической линией регрессии. Ее нахождение составляет задачу корреляционного анализа.

Уравнение линии регрессии находится в виде

$$y - y_{cp} = R_{xy} \sqrt{\frac{D_y}{D_x}} (x - x_{cp}) \quad (1)$$

где средние значения

$$y_{cp} = \frac{\sum_{i=1}^m y_i}{m}; \quad x_{cp} = \frac{\sum_{i=1}^m x_i}{m}; \quad (2)$$

коэффициент корреляции равен

$$R_{xy} = \frac{\sum_{i=1}^m (y_i - y_{cp})(x_i - x_{cp})}{(m-1) \sqrt{D_y \cdot D_x}} \quad (3)$$

дисперсии равны

$$D_y = \frac{\sum_{i=1}^m (y_i - y_{cp})^2}{m-1}; \quad D_x = \frac{\sum_{i=1}^m (x_i - x_{cp})^2}{m-1}; \quad (4)$$

20. Микроклимат птичника

Контроль микроклимата в помещениях для содержания птицы - один из самых важных технологических параметров в птицеводстве.

Наиболее серьезная проблема при выращивании цыплят - неадекватная среда и ее некачественный контроль. Цыплята страдают от высоких температур, очень низкой и высокой влажности, большой концентрации углекислого газа - результат недостаточной вентиляции. Многие производители, желая снизить топливные затраты, не соблюдают нормы вентилирования помещений. В этом случае особенно пагубная ситуация складывается в ночное время, когда концентрация вредных газов, температура и влажность намного превышают оптимальные параметры. То же самое происходит и в плохую погоду, и как результат - все это оказывает неблагоприятное воздействие на здоровье птицы в течение всего цикла выращивания.

Научный подход к контролю и управлению параметрами микроклимата создает реальную возможность производства качественной продукции при минимальных научно обоснованных затратах.

21. Параметры микроклимата

Поскольку бройлеры растут, то увеличивается и газообмен, и продукты жизнедеятельности. Куриный помет накапливается и со временем начинает влиять на качество воздуха в птичнике. Главными загрязнителями считаются: пыль, аммиак (NH_3), углекислый газ (CO_2), угарный газ (CO) и пары воды. Пыль и аммиак оказывают вредное воздействие на легкие птицы, снижают устойчивость к болезням, аппетит, а в результате снижаются и привесы.

21.1. Относительная влажность

Высокая влажность в помещении приводит к теплотерям птицы, и особенно вредна при повышении температуры в птичнике. Косвенное влияние влажности заключается в том, что она способствует развитию болезнетворных бактерий (однако достаточного количества исследований в этой области проведено не было). Правда, в конце периода выращивания, когда температура не превышает 24°C , влажность уже не оказывает сильного влияния на рост птицы.

21.2. Углекислый газ

Углекислый газ (CO_2) не имеет цвета, запаха и в полтора раза тяжелее воздуха. Довольно часто такой параметр, как содержание углекислого газа в воздухе помещения, просто-напросто игнорируется, но это ошибочное мнение. Опыт показал, что чаще всего вентиляция на обычном режиме работы удаляет излишки углекислого газа, но в больших количествах этот газ начинает вызывать удушье. Углекислота - побочный продукт обмена веществ наряду с теплоотдачей и влажностью. Некоторое количество CO_2 выделяется в результате разложения птичьего помета.

21.3. Аммиак

Аммиак (NH_3) бесцветен, легче воздуха, растворяется в воде и имеет острый запах. Этот газ - побочный продукт разложения помета, так что основные проблемы начинаются в конце периода выращивания, когда на подстилке накапливается довольно большое количество помета. Замеры содержания аммиака в воздухе птичника производятся одновременно с замерами углекислого газа.

21.4. Температура

Птица является теплокровным животным, поэтому выделяет тепло. Количество "животного" тепла зависит от массы птицы и от ее количества.

В холодный период года для поддержания оптимальной температуры в помещении птичника, воздух необходимо подогревать. А чтобы он был свежий то еще и вентилировать. С экономической точки зрения любые потери тепла несут убытки, но потери тепла связанные с вентилированием являются вынужденными, так как дают возможность выращивать птицу. Поэтому повышение эффективности отопительно-вентиляционной системы дает дополнительную прибыль.

В теплый период года, когда температура воздуха оптимальная температура воздуха внутри помещения становится равной или меньшей температуры наружного воздуха, то воздух не нагревается, а излишки тепла стараются удалить повышенным вентилированием.

Параметров микроклимата птичника может быть большое количество. Среди всего множества параметров с точки зрения управления следует выделить четыре группы параметров микроклимата:

1. не учитываемые в процессе производства;
2. не контролируемые параметры значение, которых находится в допус-

тимых пределах и обеспечивается протеканием технологического процесса;

3. контролируемые;
4. управляемые.

При информатизации и автоматизации рассматриваются последние две группы как наиболее влияющие на технические и экономические показатели производства. Наиболее влияющим параметром микроклимата на продуктивность и экономические показатели является температура в рабочей зоне. Поэтому этот параметр часто используется для управления. Такие параметры микроклимата как влажность воздуха, освещенность, концентрация CO₂ и вредных примесей в воздухе являются контролируемыми.

Рассмотрим структурную схему животноводческого помещения, как объект информатизации и управления параметрами микроклимата представлено на рис.4. Основное внимание уделим температуре воздуха в рабочей зоне.

Можно считать расчет и автоматизация параметров животноводческого помещения достаточно хорошо изученными. Так приблизительное уравнение теплового баланса в отапливаемый период

$$\Phi_{наг} + \Phi_{жс} + \Phi_{Т.О.} \pm \Phi_{Т.П.} + \Phi_{рек} - \Phi_{огр} - \Phi_{вен} - \Phi_{нар} = 0 \quad (5)$$

и уравнение теплового баланса в не отапливаемый период

$$\Phi_{жс} + \Phi_{Т.О.} \pm \Phi_{Т.П.} \pm \Phi_{огр} - \Phi_{венД} \pm \Phi_{нар} = 0 \quad (6)$$

Из тепловых потоков следует выделить тепловые потоки являющиеся случайными возмущающими воздействиями на температуру в рабочей зоне: тепловые потоки наружного воздуха, через ограждающие конструкции и вентиляции. Эти тепловые потоки связаны с наружной температурой, которая является случайным непредсказуемым фактором. Другая часть тепловых потоков – тепловые потоки от птицы, технологических процессов – изменяется во времени и зависит от параметров и количества животных. Хотя параметры птицы, в свою очередь, зависят от параметров микроклимата, эти тепловые потоки можно прогнозировать и вычислять во времени. Тепловые потоки, связанные с технологическим оборудованием наиболее известны и предсказуемы.

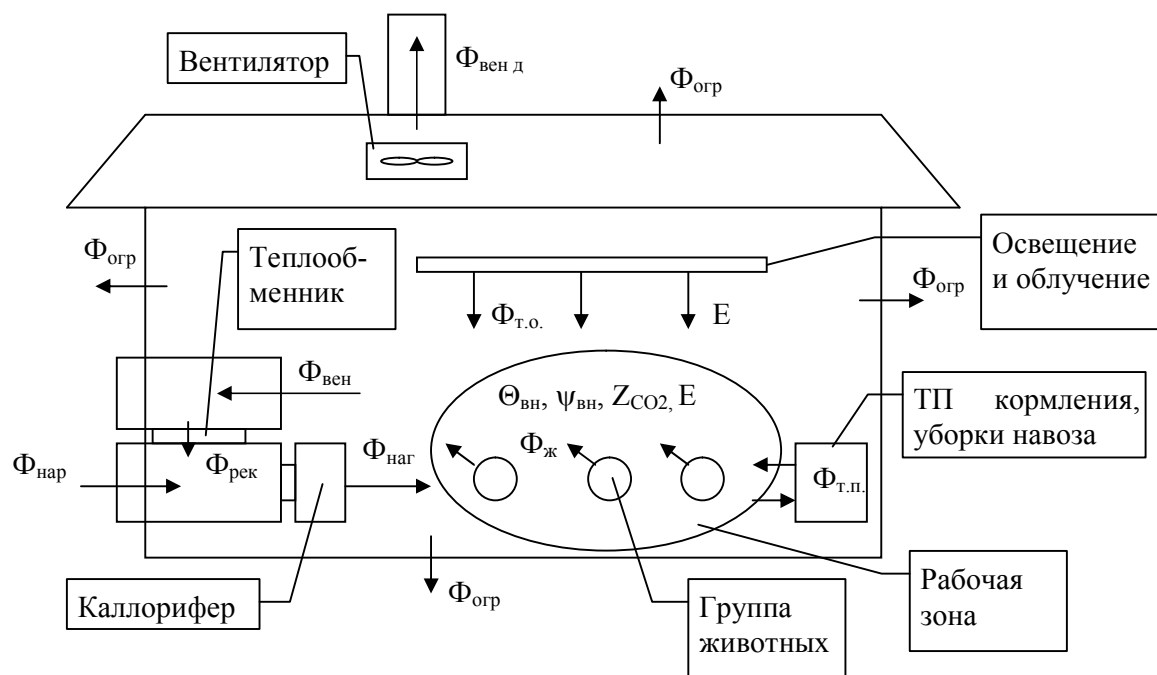


Рис.4. Структурная схема животноводческого помещения.

где $\Phi_{огр}$ – тепловой поток через ограждающие конструкции; $\Phi_{вен}$ – тепловой поток уносимый вентиляцией через теплообменник; $\Phi_{вен д}$ – тепловой поток уносимый вентиляцией в не отапливаемый период; $\Phi_{рек}$ – тепловой поток рекуперации; $\Phi_{нар}$ – тепловой поток наружного воздуха; $\Phi_{наг}$ – тепловой поток калорифера; $\Phi_{ж}$ – тепловой поток идущий от птицы; $\Phi_{т.о.}$ – тепловой поток от технологического оборудования; $\Phi_{т.о.}$ – тепловой поток от и на технологические процессы; E – освещенность; $\Theta_{вн}$ – температура в рабочей зоне; $\Psi_{вн}$ – влажность в рабочей зоне, Z_{CO_2} – содержание углекислого газа в рабочей зоне.

Влажность воздуха в рабочей зоне за день может изменяться на 60%. На ее сильно влияет способ кормления, условия содержания птицы, способ удаления помета. Неизменная часть влажности воздуха связана с другими технологическими процессами.

Освещенность задается зоотехническими нормами и не изменяется.

Облучение животных также проводится в строго заданных пределах.

Концентрация CO_2 и вредных примесей определяется из учета воздухообмена и изменяется в допустимых пределах.

Из анализа приведенных параметров микроклимата птичника помещения основную роль в энергоресурсосбережении играет уровень вентиляции, оптимальны уровень которого сильно изменяется во времени и трудно определить. На практике уровень вентиляции устанавливается с запасом выше оптимального.

22. Математическая модель продуктивности и производства птицы

Полученные экспериментальным путем математические модели позволяют изучить свойства объектов. Изучить зависимости многомерных объектов представляется сложным для понимания человека. Если исследуется зависимость одного входного параметра от одного выходного (одномерная модель), то графически эта зависимость представляется в виде графика на плоскости. Если модель имеет два входных и один выходной параметр (двухмерная модель), то зависимость строится в трех мерном пространстве. Зависимость более многомерной модели трудно отобразить для понимания человека, привыкшего думать в трех мерном пространстве. Поэтому для изучения зависимостей, моделей с мерностью три и более используют принцип "временного замораживания"

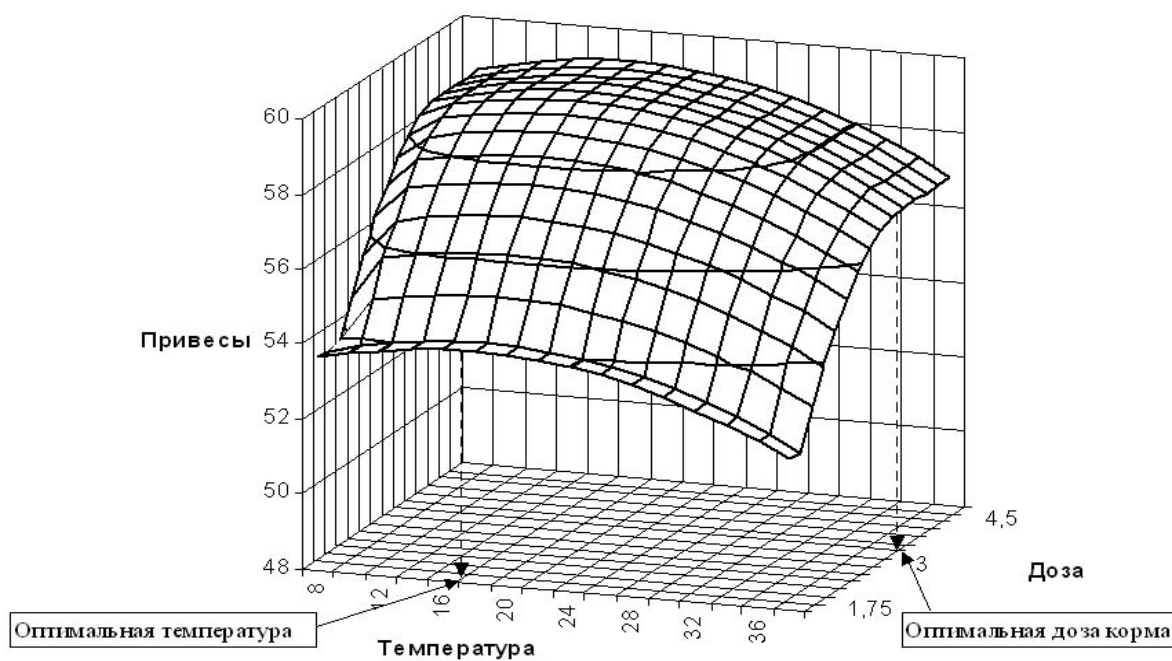


Рис.5. Зависимость привесов от температуры и дозы корма.

параметров – в определенный момент рассматриваю изменения двух параметров и строят зависимости в трехмерном пространстве, остальные параметры не изменяют.

По трехмерной модели определяется оптимум по двум параметрам. Например – оптимальная температура и доза корма. Критериями оптимальности может быть максимум привесов и максимум прибыли. По зависимости привесов

от дозы и температуры (рис.5.) определяется оптимальная температура и доза, которые соответствуют максимум привесов.

Математическая модель объекта позволяет не только изучить сам объект для построения систем автоматического управления, но и использовать математические модели управления непосредственно в управлении.