

Модели данных

Лекция № 4-5

Направление подготовки «Прикладная информатика»

230700.62

Курс «Теория экономических информационных систем»

Самостоятельная работа по пройденному материалу

1. Определение ИС.
2. Компоненты структуры системы управления экономическим объектом.
3. Основные функции управления.
4. Структура ИС.
5. Перечислить компоненты ИС в соответствии с ГОСТ 34.320-96.
6. Определение БД.
7. Перечислить типовые модели ЖЦ системы.

Модели данных

Представляют собой формальный аппарат для описания информационных потребностей пользователей.

СУБД ориентируются на конкретную модель данных, позволяя относительно быстро создавать работоспособный фрагмент ИС.

**Структура информационной базы,
отображающая в структурированном
виде информационную модель
предметной области, позволяет
сформировать:**

- логические записи**
- элементы логических записей**
- взаимосвязи между элементами**

Типы взаимосвязей

- **"один ко многим"**, когда одна запись взаимосвязана со многими другими записями;
- **"многие ко многим"**, когда одна и та же запись может входить в отношения со многими другими записями в различных вариантах.
- **"один к одному"**, когда одна запись может быть связана только с одной записью;

Основные модели данных

- иерархическая
- сетевая
- реляционная

Задача

Необходимо разработать логическую структуру БД для хранения данных о пяти поставщиках (S – SUPPLIES), которые могут поставлять детали шести наименований (P – PARTS) в следующих комбинациях:

поставщик S1 - все шесть наименований деталей;

поставщик S2 – детали P1, P2;

поставщик S3 – детали P2;

поставщик S4 – детали P2, P4, P5;

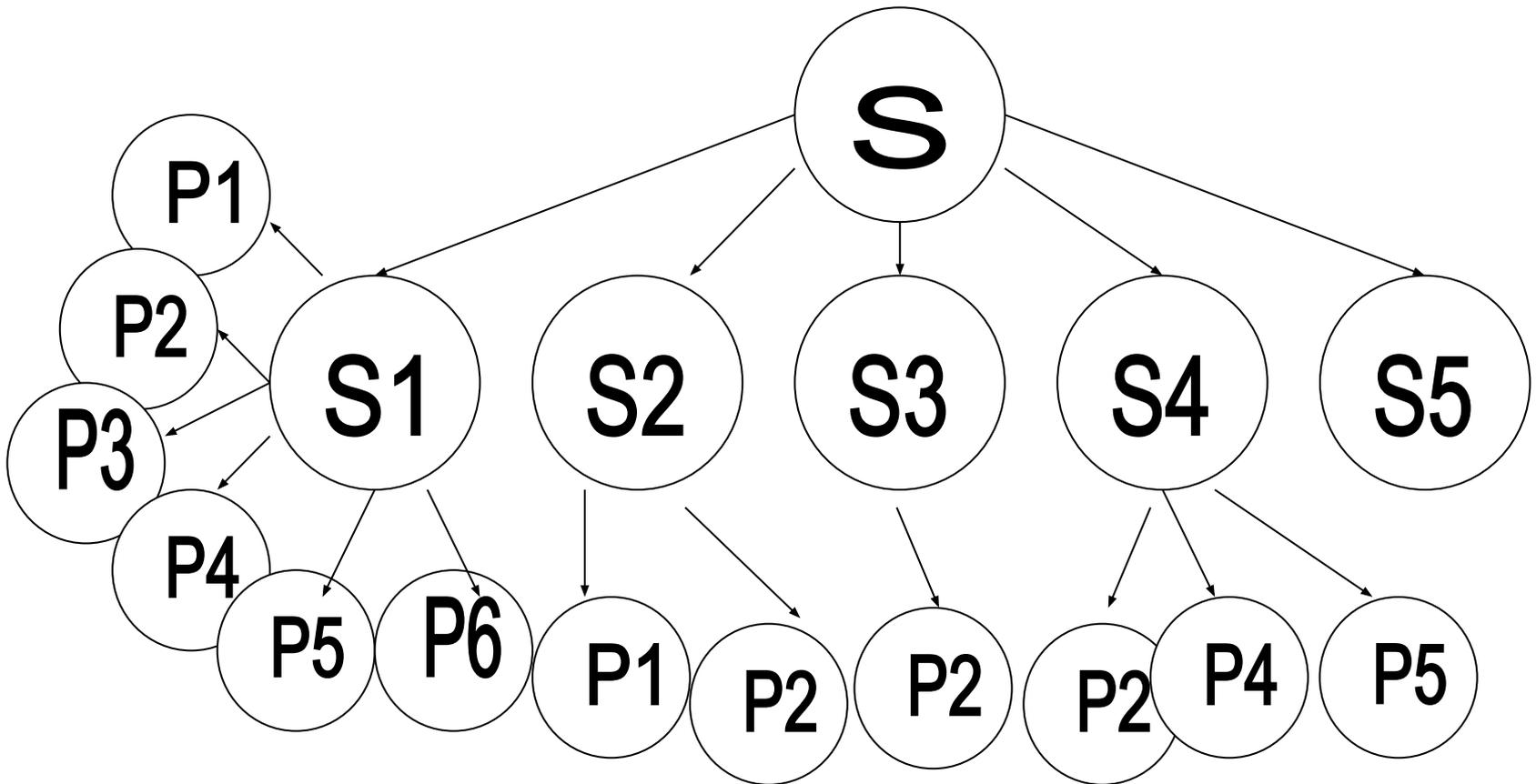
поставщик S5 - пока поставок не производил.

Иерархическая модель данных

Данные представлены в виде набора **древовидных структур**, а среди операций работы с иерархическими структурами есть операции перемещения по иерархическим путям вниз и вверх по деревьям.

Правило: каждый порожденный узел не может иметь больше одного порождающего узла; в структуре может быть только один непорожденный узел - корень.

Иерархическая модель данных



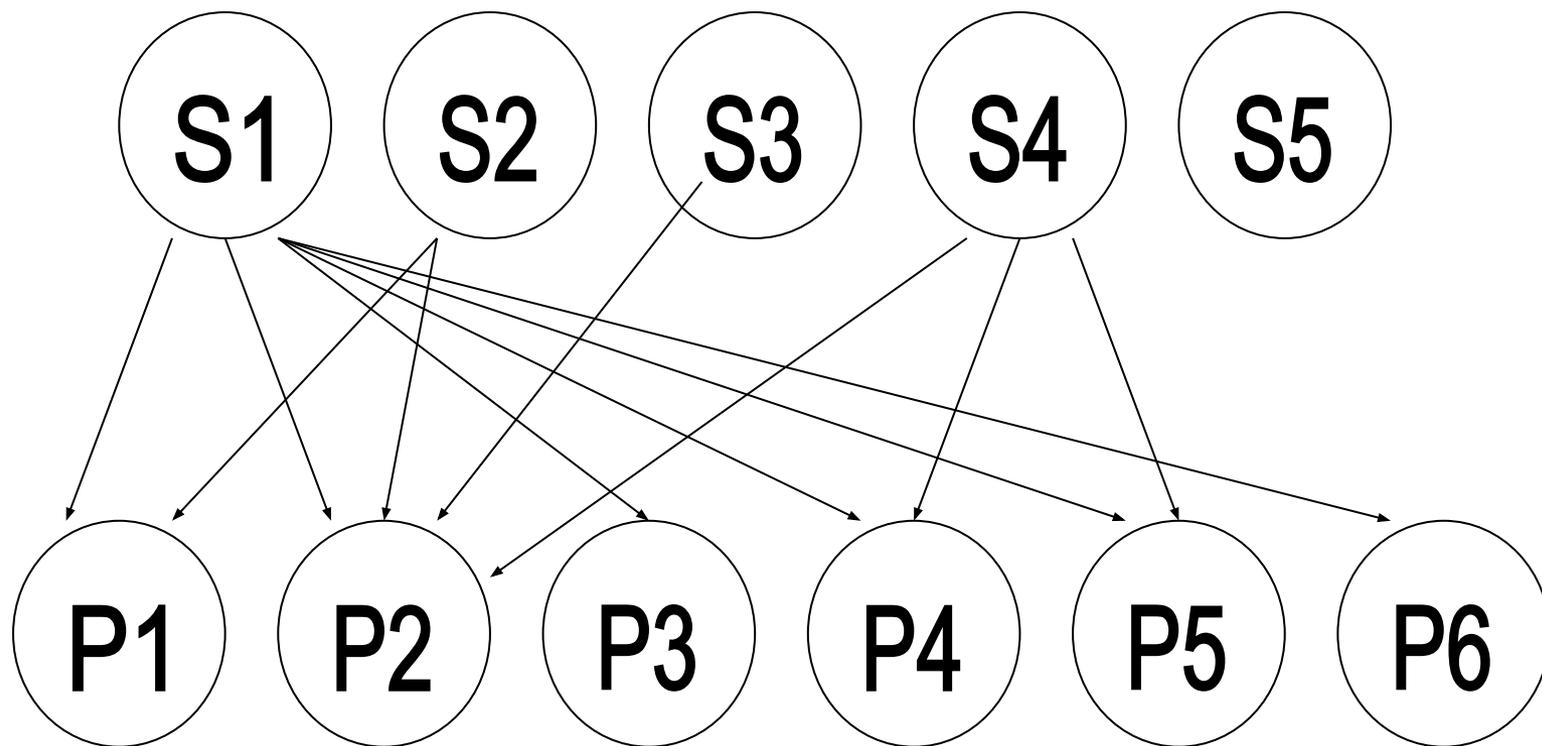
Сетевая модель данных

Данные представлены в виде **произвольного графа**. Для этой модели характерны все операции на множестве графов.

Сетевая модель для поставленной задачи представлена в виде диаграммы связей с указанием независимых ($S1\dots S5$) и зависимых ($P1\dots P6$) типов данных.

Правило: связь включает основную и зависимую записи.

Сетевая модель данных



Реляционная модель данных

*В реляционной модели данных взаимосвязи между элементами данных представляются в виде **двумерных таблиц**, называемых **отношениями**.*

Реляционная модель данных

SP (SHIPMENTS)	
S1	P1
S1	P2
S1	P3
S1	P4
S1	P5
S1	P6
S2	P1
S2	P2
S3	P2
S4	P2
S4	P4
S4	P5

P (PARTS)
P1
P2
P3
P4
P5
P6

S (SUPPLIES)
S1
S2
S3
S4
S5



Реляционная модель данных

Основные понятия

По К.Дж.Дейту «Введение в системы баз данных»

Реляционная модель – это способ рассмотрения данных или предписание для способа рассмотрения данных (посредством таблиц) и для способа работы с таким представлением (посредством операторов).

Реляционная модель – это набор взаимосвязанных отношений (таблиц)

Основные понятия

Три аспекта данных:

- **структура данных** – логическая организация данных в БД;
- **целостность** – безошибочность и точность информации, хранящейся в БД
- **обработка данных** – действия, совершаемые над данными в БД



Три части изучения:

- **объекты** (таблицы);
- **целостность** (обеспечивается внешними и первичными ключами)
- **операторы**

Детализация реляционной модели данных «Поставщики-детали»:

Эта база данных состоит из трех отношений: S, P, SP.

Отношение S представляет поставщиков. Каждый поставщик имеет номер, уникальный для этого поставщика, фамилию, значение рейтинга и местонахождение (город).

Отношение P описывает детали. Каждая деталь имеет уникальный номер, название, цвет, вес и город, где производится.

В отношении SP отражены поставки деталей. Оно служит для того, чтобы связать между собой два других отношения S и P.

Детализация реляционной модели данных «Поставщики-детали»:

S (поставщики): SUPPLIES – базовое отношение

S# (PK)	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Black	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Детализация реляционной модели данных «Поставщики-детали»:

P (детали): PARTS – базовое отношение

P# (PK)	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

Детализация реляционной модели данных «Поставщики-детали»:

SP (детали–поставщики): SHIPMENTS – поставки – связанное отношение

S# (FK)	P# (FK)	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

Реляционные объекты данных

- **Отношение** соответствует тому, что мы называли таблицей.
- **Кортеж** соответствует строке этой таблицы, **атрибут** – столбцу.
- Количество кортежей называется **кардинальным числом**, а количество атрибутов – **степенью**.
- **Первичный ключ** – это уникальный идентификатор для таблицы, то есть столбец или такая комбинация столбцов, что в любой момент времени не существует двух строк, содержащих одинаковое значение в этом столбце или комбинации столбцов.
- **Домен** – это общая совокупность значений, из которых берутся реальные значения для определенных атрибутов определенного отношения.

Пример реляционных объектов данных

Схема отношения S (S#, SNAME, STATUS, CITY)

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

Основные реляционные термины

Формальный реляционный термин	Неформальный эквивалент
Отношение	Таблица, сущность
Кортеж	Строка или запись, экземпляр сущности
Кардинальное число	Количество строк
Атрибут	Столбец или поле
Степень	Количество столбцов
Первичный ключ	Уникальный идентификатор
Домен	Общая совокупность допустимых значений атрибутов

Основные реляционные термины

- Переменная отношения – это обычная переменная, т. е. именованный объект, значение которого может изменяться.
- Значение этой переменной в любой момент времени является значением отношения.
- Отношение, определенное на множестве доменов, содержит две части: заголовок и тело (заголовок – строка заголовков столбцов; тело – множество строк данных).

S (S#, SNAME, STATUS, CITY) – заголовок отношения S

Свойства отношений

- *Нет одинаковых кортежей*
- *Кортежи не упорядочены* (сверху вниз)
- *Атрибуты не упорядочены* (слева направо)
- *Все значения атрибутов атомарные*

Виды отношений

- **Именованное отношение** – это переменная отношения, определенная в СУБД посредством операторов типа CREATE BASE RELATION.
- **Базовое отношение** – это именованное отношение, которое не определено через другие отношения. На практике к базовым отношениям относятся такие, которые проектировщик считает непосредственной частью базы данных.
- **Производное отношение** – это отношение, определенное через другие именованные отношения.
- **Выражаемое отношение** – это отношение, которое можно получить из набора именованных отношений посредством некоторого реляционного отношения.

Целостность реляционных данных

**Общие правила целостности накладывают
ограничения на потенциальные и внешние
КЛЮЧИ**

Потенциальный ключ

По К.Дж.Дейту «Введение в системы баз данных»

Потенциальный ключ отношения R (relation) – это некоторое подмножество множества атрибутов R , обладающее следующими свойствами:

- 1. Свойством уникальности.** Нет двух различных кортежей в отношении R с одинаковыми значениями потенциального ключа K (key).
- 2. Свойством неизбыточности.** Никакое из подмножеств потенциального ключа K не обладает свойством уникальности.

Потенциальный ключ

- Потенциальный ключ, состоящий из одного атрибута, называется простым, из нескольких атрибутов, составным.
- Если отношение имеет один потенциальный ключ, то его рассматривают как первичный (primary key - РК).
- Если базовое отношение имеет более одного потенциального ключа, то один из них рассматривается как первичный, остальные являются альтернативными
- (alternative key - АК)

Рассмотрим объект ЛИЧНОСТЬ

Атрибуты объекта ЛИЧНОСТЬ:

Номер паспорта

Серия паспорта

ИНН

страховое свидетельство

медицинский полис

Фамилия

Имя

Отчество

Дата рождения

Национальность

пол

возраст

Идентификация объекта ЛИЧНОСТЬ

- **Потенциальные ключи:** ИНН, паспортные данные, № медицинского полиса, страховое свидетельство
- Один из потенциальных ключей рассматривается как **первичный**, например ИНН
- Тогда **альтернативные ключи:** паспортные данные, № медицинского полиса, страховое свидетельство

Схема отношения: Личность (ИНН, паспортные данные, № медицинского полиса, страховое свидетельство, *Фамилия, Имя, Отчество, Дата рождения, Национальность, пол, возраст*), а может быть

Личность (ID, ИНН, паспортные данные, № медицинского полиса, страховое свидетельство, *Фамилия, Имя, Отчество, Дата рождения, Национальность, пол, возраст*)

Внешний ключ

Внешний ключ (*foreign key - FK*) – один или несколько атрибутов сущности, который одновременно является первичным ключом другой сущности.

Внешний ключ может быть как частью первичного ключа, так и не ключевым атрибутом.

Свойства внешних ключей

1. Внешние ключи, как и потенциальные, определены как множества атрибутов.
2. Каждое значение данного внешнего ключа должно являться значением соответствующего потенциального ключа. Обратное не требуется.
3. Данный внешний ключ будет составным тогда и только тогда, когда соответствующий потенциальный будет составным.
4. Каждый атрибут, входящий в данный внешний ключ, должен быть определен на том же домене, что и соответствующий атрибут соответствующего потенциального ключа.
5. Для внешнего ключа не требуется, чтобы он был компонентом первичного или какого-либо потенциального ключа в содержащем его отношении.

Правило ссылочной целостности

База данных не должна содержать несогласованных значений внешних ключей, т. е. каждому внешнему ключу должна соответствовать строка какого-либо объектного отношения.

Выполнение правила ссылочной целостности сводится к тому, чтобы:

- **CASCADES** - допустить любые операции (обновление или удаление) и провести ряд операций, компенсирующих некорректность
- **RESTRICTED** - запретить (ограничить) любые операции (обновление или удаление), приводящие к некорректному состоянию.

Примеры

1. Мы хотим удалить поставщика, для которого имеется хотя бы одна поставка. Другими словами, мы попытаемся удалить объект ссылки внешнего ключа. В общем случае существует по крайней мере две возможности:

CASCADES – каскадировать операцию удаления, удаляя также соответствующие поставки;

RESTRICTED - ограничить операцию удаления до того момента, когда не будет существовать соответствующих поставок.

Примеры

2. Мы хотим обновить номер поставщика, для которого существует по крайней мере одна соответствующая поставка. То есть, мы хотим обновить потенциальный ключ, на который ссылается внешний ключ. Как и в предыдущем примере существует хотя бы две возможности:

CASCADES – каскадировать операцию обновления, обновляя также внешний ключ в соответствующих поставках;

RESTRICTED – ограничить операцию обновления до тех пор, когда не будет существовать соответствующих поставок (в противном случае операция запрещается).

Примеры

3. Используя значения примерных данных поставщиков и деталей, определить, допустима ли каждая из следующих операций:

А) обновить деталь P6, установив атрибут CITY равным New York;

Б) обновить деталь P5, установив атрибут P# равным P4;

В) обновить поставщика S4, установив атрибут S# равным S8, если для соответствующего правила обновления установлена опция RESTRICTED;

Г) удалить поставщика S3, если для соответствующего правила удаления установлена опция CASCADES;

Д) обновить поставку S1-P1, установив атрибут S# равным S2;

Е) обновить поставку S4-P4, установив атрибут P# равным P6;

Ж) вставить поставку S5-P5.