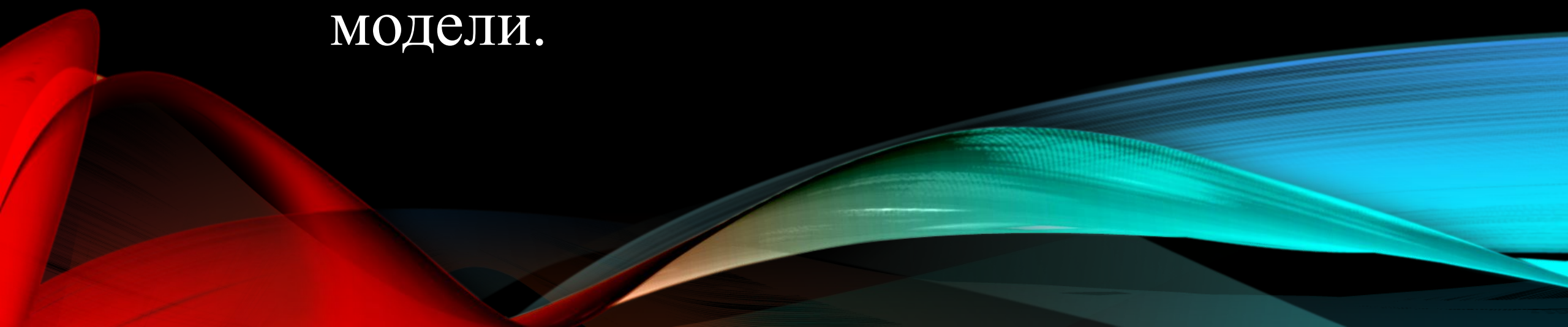


ЛЕКЦИЯ 5

МОДЕЛИ ДАННЫХ

1. Модели данных.
 2. Классификация моделей данных.
 3. Сетевая модель. Структуры данных.
 4. Управляющая часть модели.
 5. Иерархическая модель данных.
 6. Структурная и управляющая часть модели.
- 

МОДЕЛИ ДАННЫХ. КЛАССИФИКАЦИЯ МОДЕЛЕЙ ДАННЫХ

Моделью данных называется формализованное описание структуры единиц информации и операций над ними в информационной системе.

Модель данных — это некоторая абстракция, в которой отражаются самые важные аспекты функционирования выделенной предметной области, а второстепенные — игнорируются. Модель данных включает в себя набор понятий для описания данных, связей между ними и ограничений, накладываемых на данные.

В модели данных различают три главные составляющие:

- × структурную часть, определяющую правила порождения допустимых для данной СУБД видов структур данных;
- × управляющую часть, определяющую возможные операции над такими структурами;
- × классы ограничений целостности данных, которые могут быть реализованы средствами этой системы.

Каждая СУБД поддерживает ту или иную модель данных. По существу модель данных, поддерживаемая механизмами СУБД, полностью определяет множество конкретных баз данных, которые могут быть созданы средствами этой системы, а также способы модификации состояния БД с целью отображения тех изменений, которые происходят в предметной области.



КЛАССИФИКАЦИЯ МОДЕЛЕЙ ДАННЫХ

В настоящее время описано много разнообразных моделей, построение которых преследует разные цели. Из множества опубликованных моделей данных можно выделить три категории:

- × объектные модели данных;
- × модели данных на основе записей;
- × физические модели данных.

Модели данных

Идеологические модели

Диаграммы Бахмана

Модель сущность-связь (ER)

Ориентированные на формат документа

Даталогические модели

Документальные модели

Дескрипторные модели

Тезаурусные модели

Теоретико-графовые

Иерархическая

Сетевая

Фактографические модели

Объектно-ориентированные

Теоретико-множественные

Реляционная

Бинарных ассоциаций

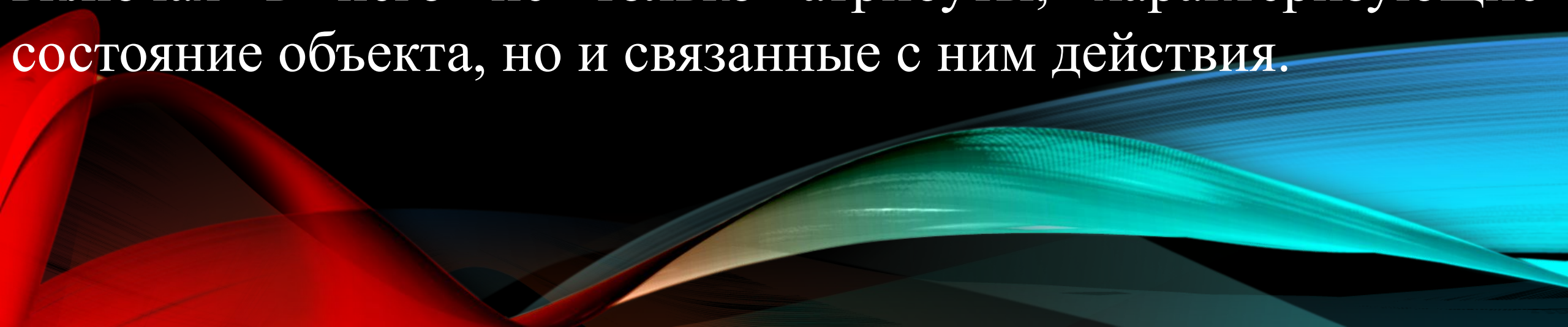
Физические модели

Основанные на файловых структурах

Основанные на странично-сегментной организации

Объектные модели данных

Среди объектных моделей следует выделить ER-модель, которая наиболее часто используется в методологии проектирования баз данных, а также объектно-ориентированную модель, последнее время широко используемую в технологиях баз данных. Объектно-ориентированная модель расширяет понятие объекта, включая в него не только атрибуты, характеризующие состояние объекта, но и связанные с ним действия.



Модели данных на основе записей

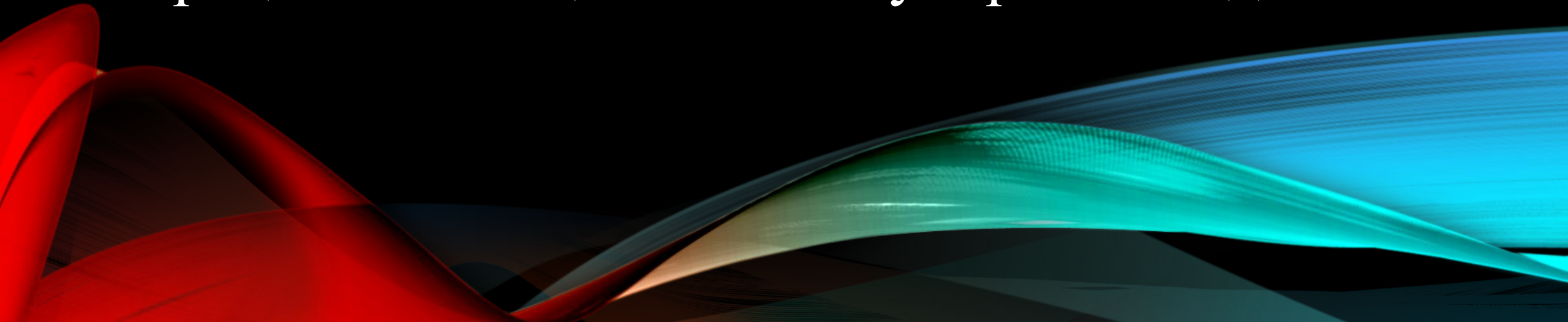
В модели данных на основе записей, база данных состоит из нескольких записей фиксированного формата, которые могут иметь разные типы.

В большинстве коммерческих СУБД используются ставшие классическими два типа такого рода моделей данных: теоретико-графовые (ТГ) и теоретико-множественные (ТМ) модели данных.

К теоретико-графовым моделям относятся две разновидности:

- × сетевые модели;
- × иерархические модели.

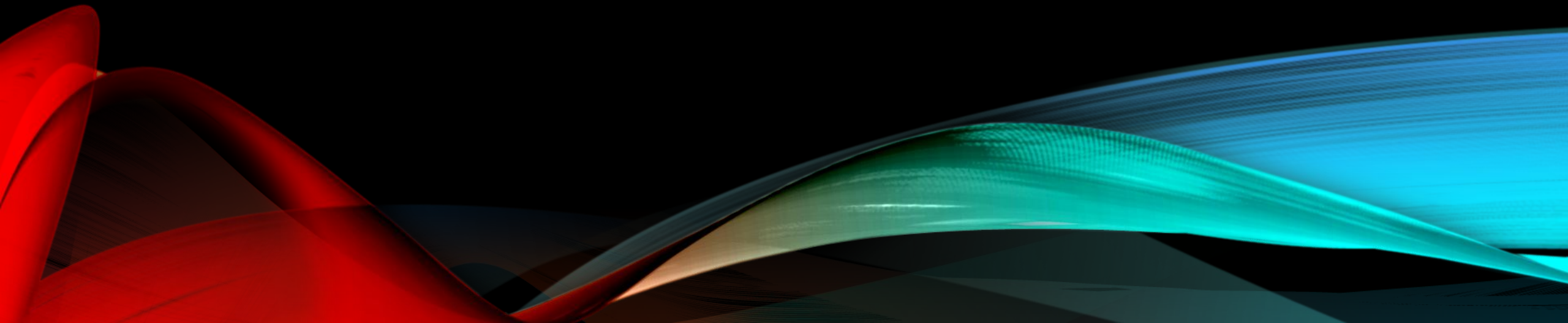
В таких моделях данных предусматриваются характерные для подобного рода структур операции навигации и манипулирования данными.



Аппарат навигации в ТГ-моделях служит для установки тех объектов данных, к которым будет применяться очередная операция манипулирования данными.

Теоретико-множественные модели используют математический аппарат, реляционную алгебру (знаковая обработка множеств), реляционное исчисление. К моделям данного типа относятся реляционные модели.

В соответствии с реляционной моделью данных БД представляется в виде совокупности таблиц, над которыми могут выполняться операции, формируемые в терминах реляционной алгебры и реляционного исчисления.



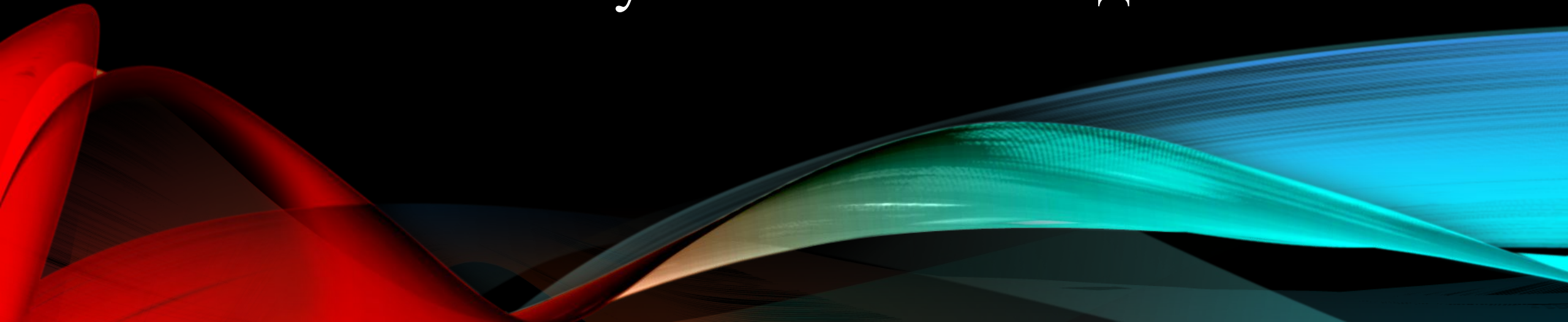
Физическая модель данных

Физические модели данных описывают то, как данные хранятся в компьютере, представляя информацию о структуре записей, их упорядоченности и существующих путях доступа. Наиболее распространены из них следующие: обобщающая модель и модель памяти кадров.

Сетевая модель

Сети — естественный способ представления отношений между объектами, всевозможных их взаимосвязей. Сетевая модель опирается на математическую структуру, которая называется направленным графом.

Направленный граф состоит из узлов, соединенных ребрами. В контексте моделей данных узлы представляют собой объекты в виде типов записей данных, а ребра — связи между объектами со степенью кардинальности "один к одному" или "один ко многим". Иерархическая модель данных является частным случаем сетевой модели.



Структуры данных сетевой модели

В сетевой модели используются несколько различных типовых структур данных, главными из которых являются типы записей и наборы. Для построения этих структур применяются такие конструктивные элементы, как элемент данных и агрегат

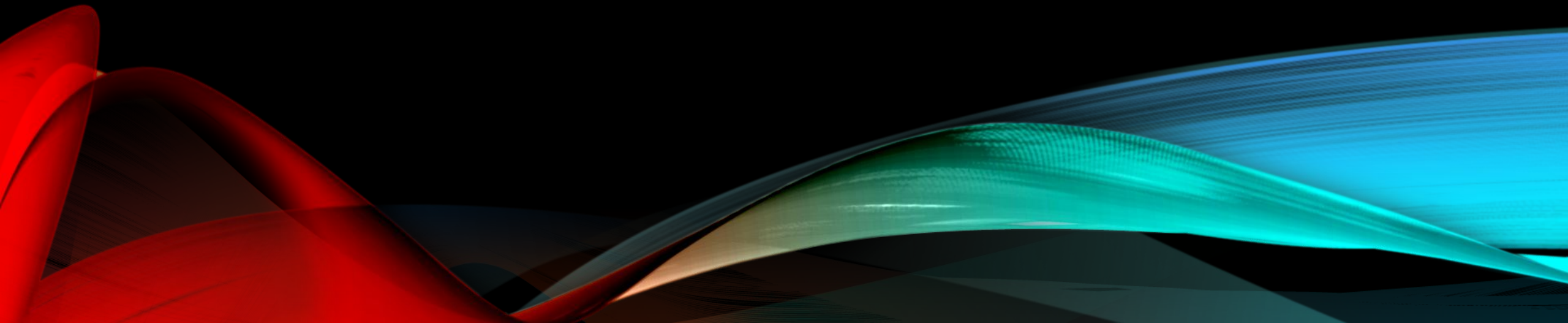


Элемент данных — это наименьшая поименованная информационная единица данных, доступная пользователю (аналог — поле в файловой системе). Элемент данных должен иметь свой тип (неструктурный, простой).

Агрегат данных соответствует следующему уровню обобщения — поименованная совокупность элементов данных внутри записи или другого агрегата

Дата		
День	Месяц	Год

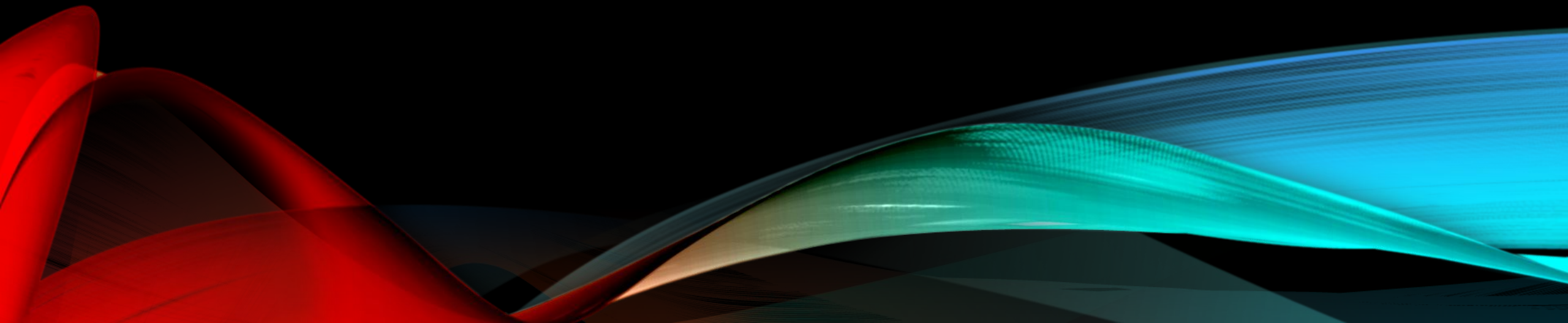
Запись — конечный уровень агрегации. Каждая запись представляет собой именованную структуру, содержащую один или более именованных элементов данных, каждый из которых обладает своим особым форматом.



Агрегат данных **Дата** входит в состав записи **Сотрудник**. Тип записей — это совокупность логически связанных экземпляров записей. Тип записей моделирует некоторый класс объектов реального мира.

Сотрудник					
Табельный номер	ФИО	Дата			Адрес
		День	Месяц	Год	

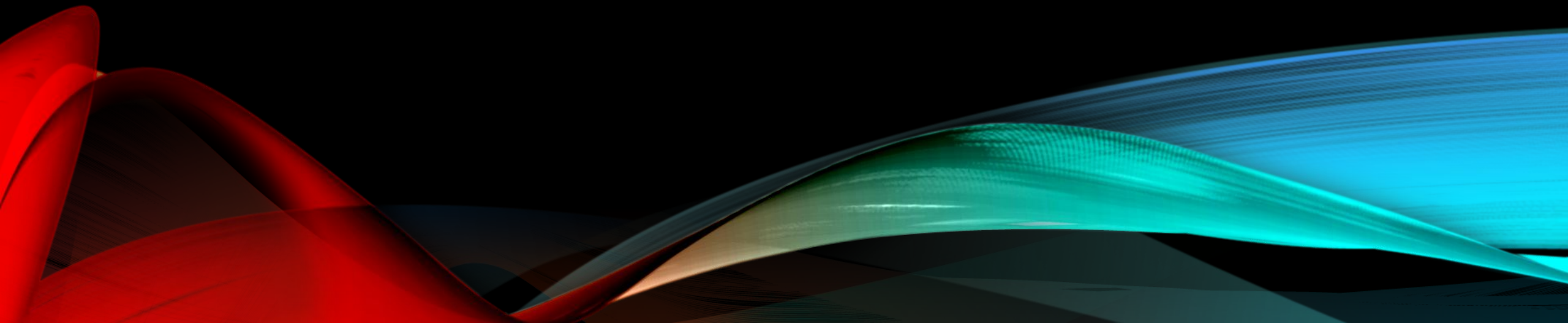
В качестве элемента данных могут быть использованы только простые типы, а в качестве агрегатов могут быть использованы сложные типы: вектор и повторяющаяся группа.

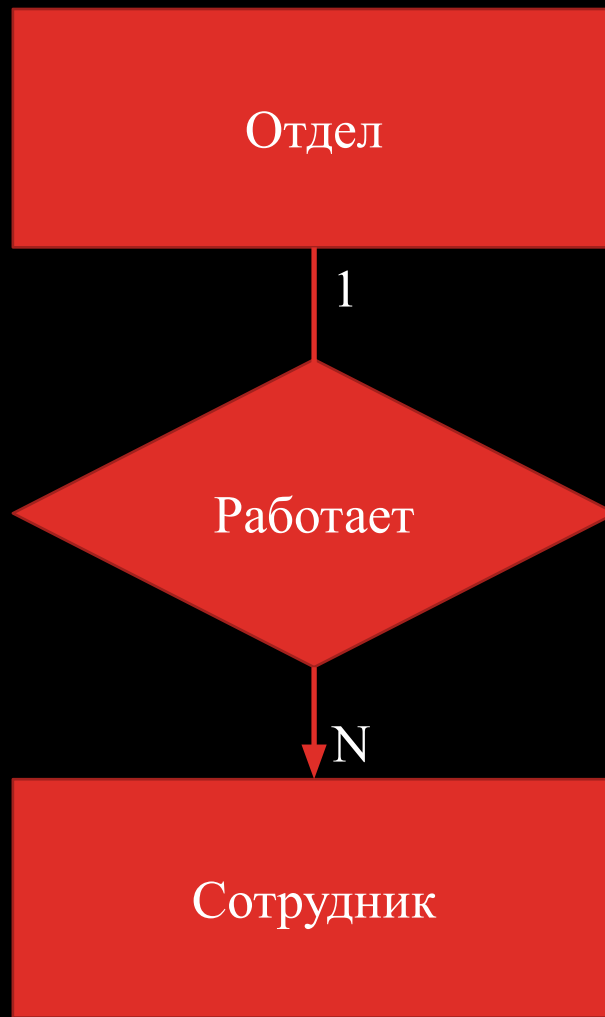


Агрегат типа вектор соответствует линейному набору элементов данных. Агрегат типа повторяющаяся группа соответствует совокупности векторов данных. Так, например, в заказе может быть указано несколько видов товаров с числом повторений 10.

Заказ						
Номер заказа	Дата заказа			Товар		
	День	Месяц	Год	Шифр товара	Количество товара	Наименование товара
				Повторяющаяся группа		

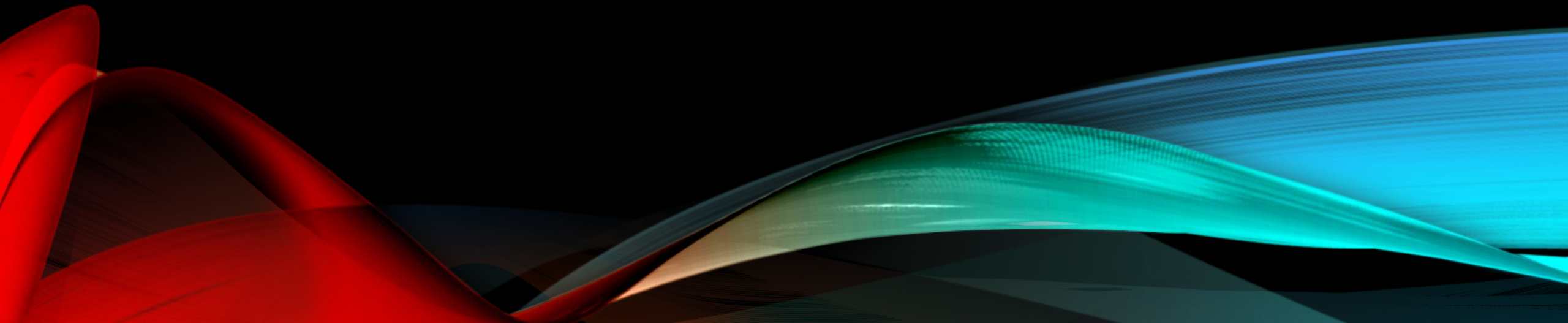
Набор — это поименованная двухуровневая иерархическая структура, которая содержит запись владельца и записи членов. Наборы выражают связи "один ко многим" или "один к одному" между двумя типами записей. Тип набора поддерживает работу с внутренними структурами типов записей.





Набор, определяет тип записи-владельца **Отдел** и тип записи-члена набора **Сотрудник**, а также тип связи между ними "один ко многим" — с именем **Работает**. Имя набора — это метка, присвоенная стрелке. Связь типа "один ко многим" допускает возможность того, что с данным экземпляром записи-владельца может быть связан ноль, один, или несколько экземпляров записи-члена.

Используя понятия сетевой модели данных, можно получить другое изображение такого набора, где представлены логические типы записей **Отдел** и **Сотрудник**, их структура и связь между типами записей **Работает**.



Отдел

Код	Название	Номер карты
-----	----------	-------------

Тип записи владельца

Работает

Тип записи члена

Табельный номер	ФИО	Должность	Образование
-----------------	-----	-----------	-------------


Сотрудник

Преобразование концептуальной модели в сетевую

Сетевая модель данных может быть без осложнений получена из концептуальной модели. Для этого надо предположить, что в последней используются только бинарные связи. Причем они должны принадлежать к типам "один к одному" или "один ко многим". При этом вместо сущностей концептуальной модели необходимо использовать типы записей сетевой модели, где имена сущностей становятся именами типов записей, атрибуты сущностей становятся полями записей, связь между сущностями превращается в связь между типами записей.

Бинарные связи, принадлежащие к типу "один ко многим", переносятся в сетевую модель следующим образом: тип записи со стороны "один" становится владельцем, а тип записи со стороны "много" становится типом записи-члена. Для связи типа "один к одному" выбор типа записи-владельца и типа записи-члена может быть осуществлен произвольно.





Для манипулирования данными в сетевой модели данных определен ряд типичных операций, которые можно подразделить на две группы: навигационные операции и операции модификации.

Навигационные операции осуществляют перемещение по БД путем прохождения по связям, определенным в схеме БД. В результате таких операций определяется запись, которая называется текущей.

К подобным операциям относятся:

- ✗ найти конкретную запись в наборе однотипных записей и сделать ее текущей;
- ✗ перейти от записи-владельца к записи-члену в некотором наборе;
- ✗ перейти к следующей записи в некоторой связи;
- ✗ перейти от записи-члена к владельцу по некоторой связи.

Операции модификации осуществляют как добавление новых экземпляров отдельных типов записей, так и экземпляров новых наборов, удаление экземпляров записей и наборов, модификацию отдельных компонентов самой записи. Для реализации этих операций в системе текущее состояние детализируется путем запоминания трех его составляющих: текущего набора, текущего типа записи, текущего экземпляра типа записи.

В такой ситуации возможны следующие операции:


- ✗ извлечь текущую запись в буфер прикладной программы для обработки;
- ✗ заменить в извлеченной записи значения указанных элементов данных на заданные новые их значения;
- ✗ запомнить запись из буфера в БД;
- ✗ создать новую запись;
- ✗ уничтожить запись;
- ✗ включить текущую запись в текущий экземпляр набора;
- ✗ исключить текущую запись из текущего экземпляра набора.


ИЕРАРХИЧЕСКАЯ МОДЕЛЬ ДАННЫХ

Первые системы управления базами данных использовали иерархическую модель данных, и во времени их появление предшествует появлению сетевой модели.

СТРУКТУРНАЯ ЧАСТЬ ИЕРАРХИЧЕСКОЙ МОДЕЛИ

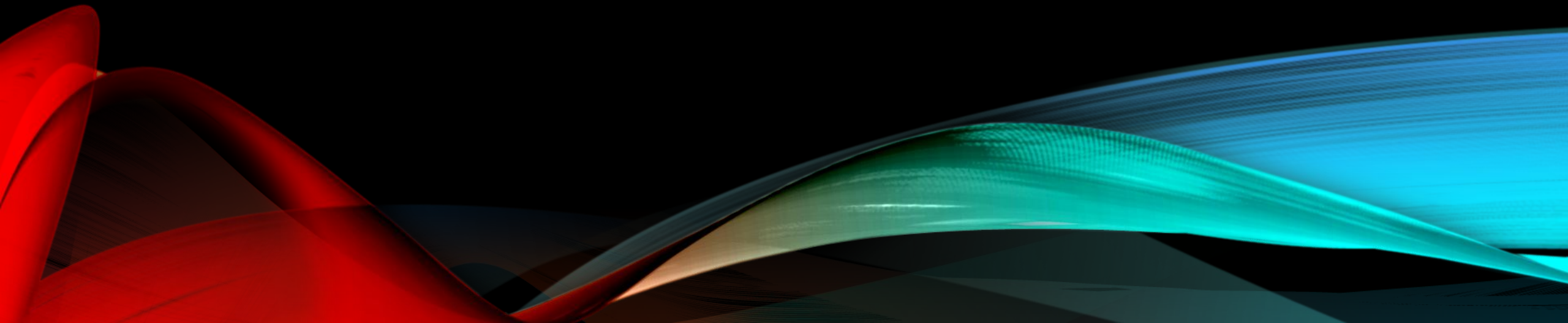
Основными информационными единицами в иерархической модели данных являются сегмент и поле. Поле данных определяется как наименьшая неделимая единица данных, доступная пользователю. Для сегмента определяются тип сегмента и экземпляр сегмента. Экземпляр сегмента образуется из конкретных значений полей данных. Тип сегмента — это поименованная совокупность входящих в него типов полей данных.



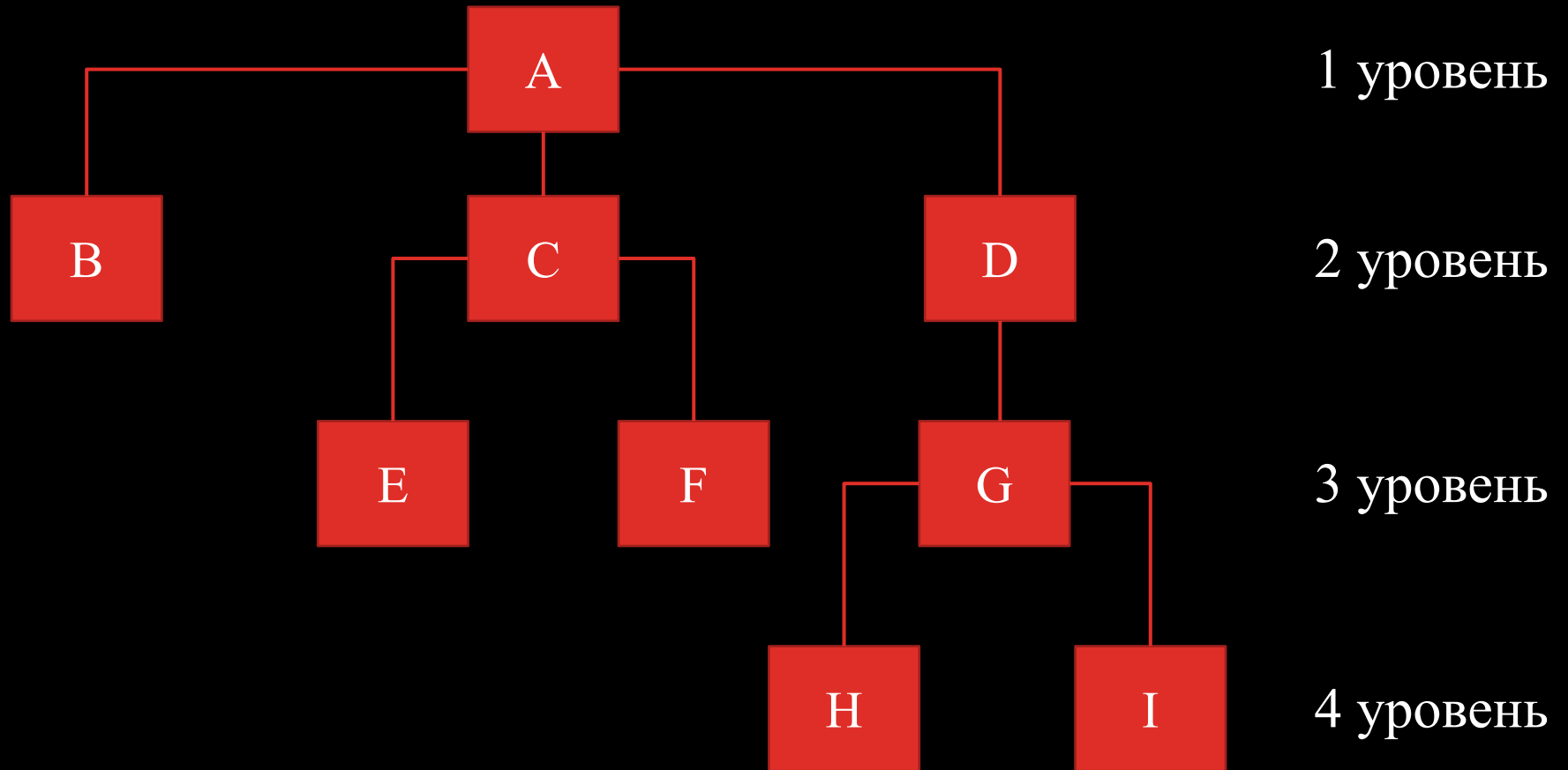


Как и сетевая, иерархическая модель данных базируется на графовой форме построения данных, и на концептуальном уровне она является просто частным случаем сетевой модели данных. В иерархической модели данных вершине графа соответствует тип сегмента или просто сегмент, а дугам — типы связей предок — потомок. В иерархических структурах сегмент — потомок должен иметь в точности одного предка.

Иерархическая модель представляет собой связный неориентированный граф древовидной структуры, объединяющий сегменты. Иерархическая БД состоит из упорядоченного набора деревьев




Иерархическая модель данных



Преобразование концептуальной модели в иерархическую модель данных

Преобразование концептуальной модели в иерархическую структуру данных во многом схоже с преобразованием ее в сетевую модель, но и имеет некоторые отличия в связи с тем, что иерархическая модель требует организации всех данных в виде дерева.

Преобразование связи типа "один ко многим" между предком и потомком осуществляется практически автоматически в том случае, если потомок имеет одного предка.



Каждый объект с его атрибутами, участвующий в такой связи, становится логическим сегментом. Между двумя логическими сегментами устанавливается связь типа "один ко многим". Сегмент со стороны "много" становится потомком, а сегмент со стороны "один" становится предком.

Ситуация значительно усложняется, если потомок в связи имеет не одного, а двух и более предков. Так как подобное положение является невозможным для иерархической модели, то отражаемая структура данных нуждается в преобразованиях, которые сводятся к замене одного дерева, например, двумя (если имеется два предка). В результате такого преобразования в базе данных появляется избыточность, так как единственно возможным выходом из этой ситуации — дублирование данных.

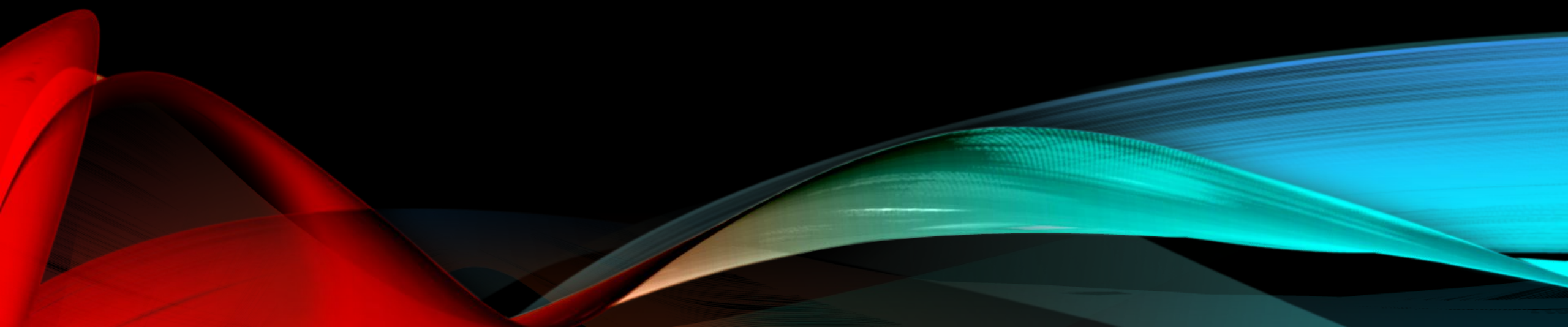
Управляющая часть иерархической модели

В рамках иерархической модели выделяют языковые средства описания данных (ЯОД) и средства манипулирования данными (ЯМД).

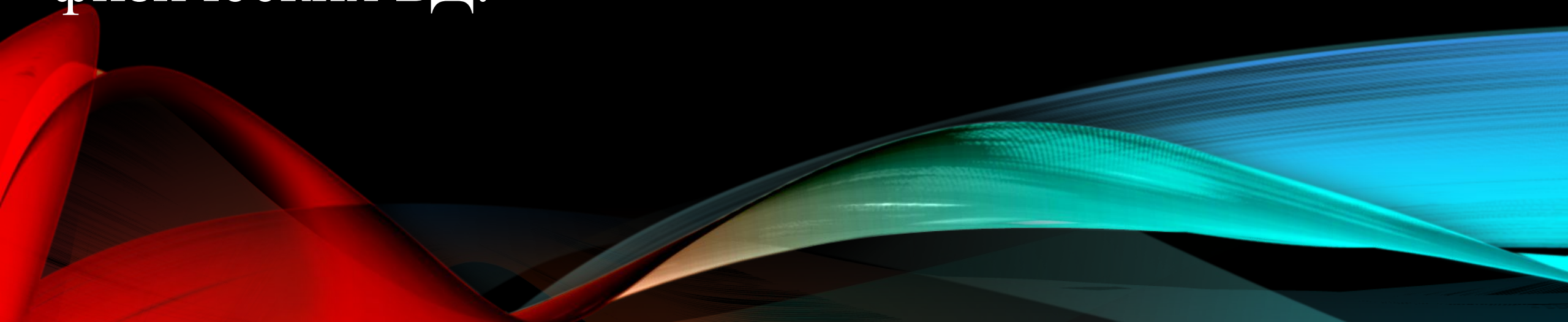
Каждая физическая база описывается набором операторов, обуславливающих как ее логическую структуру, так и структуру хранения БД. При этом способ доступа устанавливает способ организации взаимосвязи физических записей.

Определены следующие способы доступа:

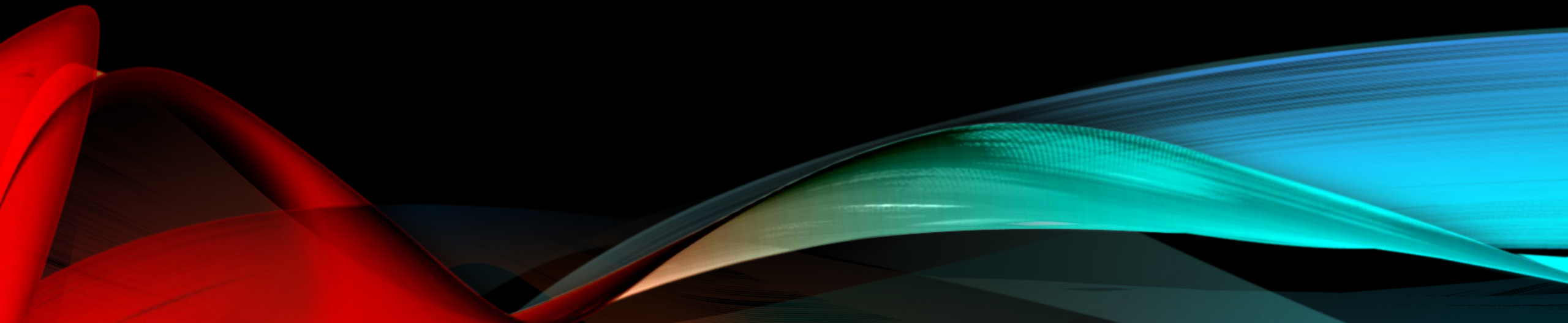
- × иерархически последовательный;
- × иерархически индексно-последовательный;
- × иерархически прямой;
- × иерархически индексно-прямой;
- × индексный.



Помимо задания имени БД и способа доступа описания должны содержать определения типов сегментов, составляющих БД, в соответствии с иерархией, начиная с корневого сегмента. Каждая физическая БД содержит только один корневой сегмент, но в системе может быть несколько физических БД.



Среди операторов манипулирования данными можно выделить операторов поиска данных, операторов поиска данных с возможностью модификации, операторов модификации данных. Набор операций манипулирования данными в иерархической БД невелик, но вполне достаточен.



Примеры типичных операторов поиска данных:

- × найти указанное дерево БД;
- × перейти от одного дерева к другому;
- × найти экземпляр сегмента, удовлетворяющий условию поиска;
- × перейти от одного сегмента к другому внутри дерева;
- × перейти от одного сегмента к другому в порядке обхода иерархии.

Примеры типичных операторов поиска данных с возможностью модификации:

- ✗ найти и удержать для дальнейшей модификации единственный экземпляр сегмента, удовлетворяющий условию поиска;
- ✗ найти и удержать для дальнейшей модификации следующий экземпляр сегмента с теми же условиями поиска;
- ✗ найти и удержать для дальнейшей модификации следующий экземпляр для того же родителя.

Примеры типичных операторов модификации иерархически организованных данных, которые выполняются после выполнения одного из операторов второй группы (поиска данных с возможностью модификации):

- ✗ вставить новый экземпляр сегмента в указанную позицию;
- ✗ обновить текущий экземпляр сегмента;
- ✗ удалить текущий экземпляр сегмента.



В иерархической модели автоматически поддерживается целостность ссылок между предками и потомками.

Основное правило: никакой потомок не может существовать без своего родителя.