# Что такое сетевые модели

Сетевая модель не ограничена иерархией или решеткой, а отношения между объектами могут быть один-к-одному, один-ко-многим или многие-ко-многим.

# История создания и развития

Хотя последующая работа продолжалась до начала 1980-х годов и кульминацией стала спецификация ISO, это не сильно повлияло на уже созданные СУБД имплементирующие данную модель.

# Отличие от иерархической модели

В то время как иерархическая модель базы данных структурирует данные в виде дерева записей, где каждая запись имеет одну родительскую запись и множество дочерних, сетевая модель позволяет каждой записи иметь несколько родительских и дочерних записей (многие-ко-многим), образуя обобщенную структуру графа.

Главный аргумент в пользу сетевой модели по сравнению с иерархической моделью заключался в том, что она позволяла более естественно моделировать отношения между сущностями.

# Устройство сетевых моделей

Сетевая модель имеет более простую структуру, нежели реляционная модель.

Элемент данных – это наименьшая информационная именованная единица данных, доступная пользователю, если проводит аналогию с реляционной базой данных, то элемент данных – один столбец таблицы реляционной БД.

Агрегат данных – именованная совокупность элементов или других агрегатов данных. Объяснить на примере: (Если разработчик рассматривает адрес как единое целое, то адрес – это элемент данных. Если же необходимо разделить адрес на части («страна» – «город» – «улица» – «номер дома» – «номер квартиры»), то адрес уже будет выступать как агрегат, состоящий из соответствующих элементов. При этом пользователь может запросить из базы данных как отдельно город или номер дома, так и адрес целиком, так как агрегат – это тоже именованный объект.

Запись в сетевой модели данных – это конечный уровень обобщения данных, что-то наподобие таблицы в реляционной базе данных. Каждая запись в сетевой базе данных должна содержать в себе, как минимум один именованный элемент данных, если элементов внутри записи более одного, то каждый элемент данных должен обладать уникальным форматом.

Набор в сетевой модели является иерархическим отношением между двумя типами записей. В сетевой модели один и тот же тип записи может участвовать в нескольких наборах. В частности, для любых двух типов записей может быть задано любое количество наборов, которые их связывают. Наличие подобных возможностей позволяет моделировать отношение объектов типа «многие-ко-многим», что выгодно отличает сетевую модель данных от иерархической.

# Метрики сетевых моделей

Сетевые модели могут быть описаны с помощью базовых метрик заявленных в теории графов, таких как: степень вершины, степень сети, длина пути, связность, коэффициент кластеризации и другие.

Разберем поподробнее:

1. Степень вершины — называется число ребер графа, которым принадлежит эта вершина (с которыми вершина соединена). Вершина называется нечетной, если ее степень — число нечетное. Вершина называется четной, если ее степень — число четное.
2. Степень сети — средняя степень вершин = (2 \* Кол-во рёбер) / Кол-во вершин.
3. Длина пути — минимальное количество рёбер, которое необходимо пройти между двумя вершинами. Для нахождения может использоваться алгоритм Дейкстры.
4. Связность — сеть является связанной, если из каждой вершины можно добраться до любой другой.
5. Коэффициент кластеризации — процент троек вершин, которые связаны друг с другом. В теории графов коэффициент кластеризации – это мера степени, с которой узлы в графе стремятся объединяться в кластеры. Существуют две версии этой меры: глобальная и локальная. Глобальная версия была разработана, чтобы дать общее представление о кластеризации в сети, тогда как локальная версия дает указание на встроенность отдельных узлов.

# Обобщение

Сетевая модель данных – это именованная совокупность экземпляров записей различного типа и экземпляров наборов, хранящих в себе типы связей между записями. Проще говоря, это все записи и все связи между записями.

Сетевые модели могут быть описаны с помощью различных метрик свойственных структурам, основанным на графах.

Сетевые модели внесли огромный вклад в развитие современных БД, легли в основу графо-ориентированных баз данных и повлияли на развитие реляционных СУБД, но на настоящий момент, по большей части, утратили свою актуальность.