

1 Иерархическая модель данных

Задача - хранение деревьев

1.1 Компоненты иерархической модели

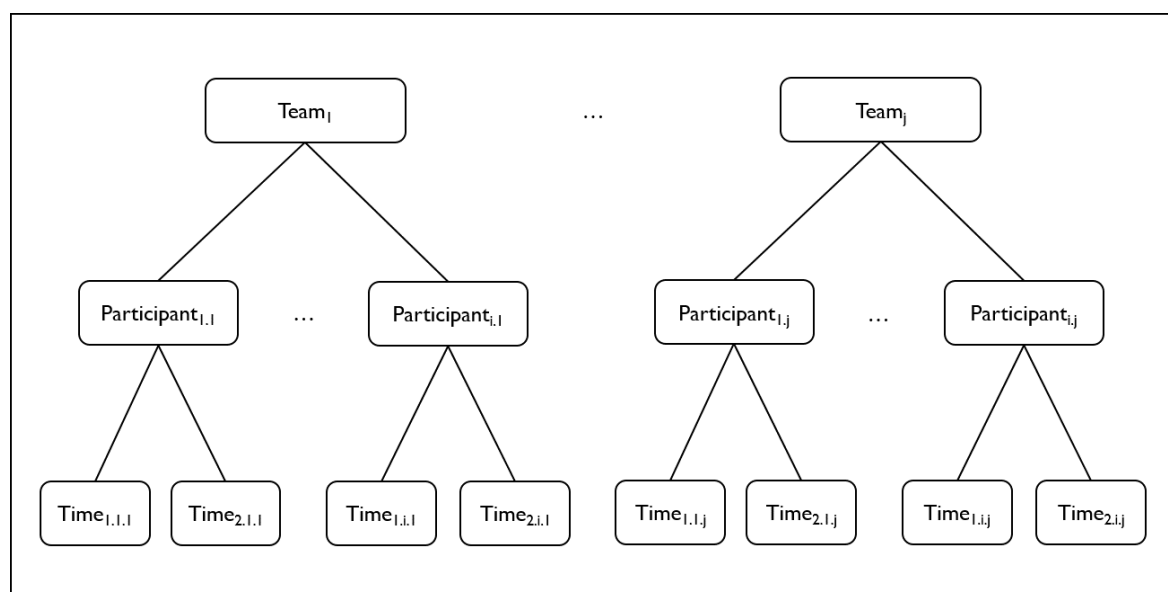
- *Поле данных* – неделимая, уникально адресуемая единица хранения данных.
- *Сегмент данных* – совокупность полей данных, имеющая уникальную идентификацию.

Как правило, поле данных – атрибут, сегмент данных – запись, экземпляр данных.

1.2 Проблемы

- Как оптимально хранить?
- Проблема скорости внесения изменений
- Дублирование данных
- Сложности с контролем целостности данных
- Любая реорганизация приводит к трудностям
- Невозможность связи - многие ко многим

Почему иерархическая модель? Это естественный, нативный способ представления данных.



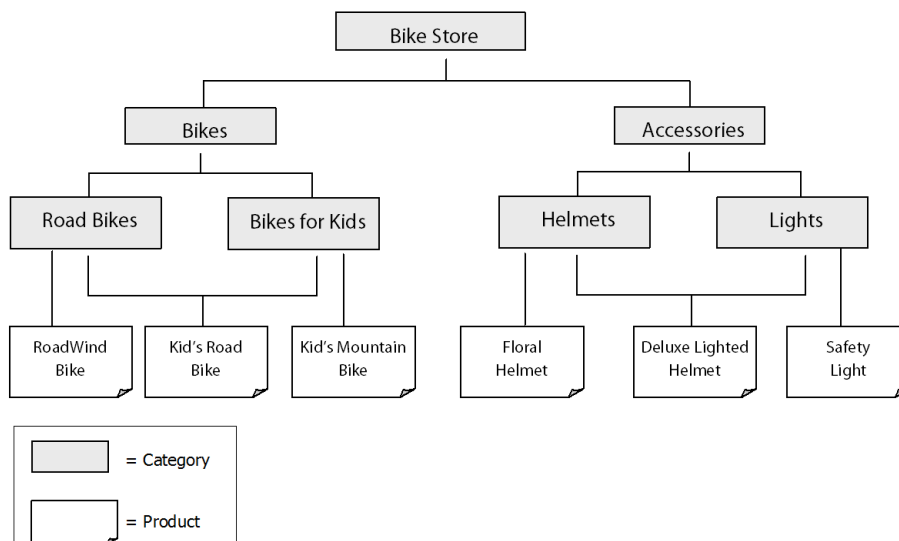
2 Сетевая модель данных

2.1 Компоненты сетевой модели

- *Поле данных*
- *Агрегат данных* – совокупность **множества** полей данных, имеющая уникальную идентификацию
- *Связь* – хранение при помощи ключа, хранение связи = пара ключей + вес

2.2 Новое

- Решает проблемы с дублированием и целостностью данных
- Начинаем разделять хранение связей от хранения самих связей
- Появление ключей
- Возможность связи многие ко многим



2.3 Проблемы

- Дорогое добавление данных, нужно проходить весь граф
- Ошибки при хранении пары ключей

3 Реляционная модель данных

3.1 Компоненты реляционной модели

- *Поле данных*
- *Отношение* – совокупность множества полей данных (множество кортежей)
- *Связь*

Реляционная модель – модель хранения отношений

- Один к одному: id
- Многие ко многим: новое отношение для хранения связи ключей

3.2 Новое

Реорганизуем хранение связей - переходим к реляционной модели

- Связи хранятся в отдельных отношениях - таблицах
- Возникновение предпосылок к избеганию дубликации данных
- При грамотной работе гарантируется целостность данных
- При грамотной работе гарантирует эффективное время для выполнения записи
- При грамотной работе гарантируется отсутствие дубликации данных

3.3 Проблемы

- Поле – неделимый элемент данных (не имеет внутренней структуры), поэтому возникает проблема с определением типа данных и их хранением

4 Постреляционная модель данных

Снимаем запрет на целостность поля данных - поле данных может само по себе являться агрегатом

- При нормализации отношений не гарантируется целостность данных, данные вынесены из понятия отношений
- Целостность данных - проблема разработчиков
- Проблема больших данных

Будем считать, что для нас скорость вычислений важнее памяти.

5 Многомерная модель

- Многомерное хранение неудобно с точки зрения модификации структуры
- OLAP кубы

6 Объектно-ориентированная модель данных

6.1 Особенности

- Экземпляр класса пытаемся сохранить в базе
- Улучшается производительность в ООП приложениях
- Удобно с точки зрения распределенности систем

6.2 Проблемы

- Замыкаемся на ОО подходе
- Целостность данных - обязанность кодовой базы