

Базы данных

Лекция 4

Проектирование баз данных.

Нормальные формы.

UID

Классификация по количеству атрибутов

- простые
- составные

Классификация по соответствию реальным объектам

- естественные
- искусственные

Общие требования

- Уникальность названий отдельных сущностей (отношений) в базе данных;
- Уникальность названий атрибутов у отдельных сущностей (отношений);
- Значения атрибута у экземпляров сущностей (кортежей) принадлежат одному домену;
- Целостность сущности: все экземпляры сущностей (кортежи) различны.

Первая нормальная форма

Сущность (отношение) находится в 1НФ, если:

- выполняются общие требования;
- каждый атрибут содержит атомарное значение.

Функциональные зависимости

- Атрибут С называется функционально зависимым от атрибутов А и В (обозначается $A, B \rightarrow C$), если значения атрибутов А и В однозначно определяют атрибут С.
- Выявление функциональной зависимости атрибутов является неформальным действием и определяется в результате тщательного анализа предметной области, возможных событий в ней и возможных значений атрибутов.

Вторая нормальная форма

Сущность (отношение) находится во 2НФ, если:

- она находится в 1НФ;
- каждый неключевой атрибут функционально полно зависит от первичного ключа.

Третья нормальная форма

Сущность (отношение) находится в 3НФ, если:

- она находится в 2НФ;
- каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Другие нормальные формы

- Кроме приведенных нормальных форм имеются и другие нормальные формы (нормальная форма Бойса-Кодда – 3НФ+, 4НФ, 5НФ и др.), однако их использование носит в большей степени теоретический характер.

Этапы проектирования сверху вниз

- создание логической модели (analysis);
- преобразование логической модели в физическую – схему таблиц (design);
- генерация скрипта и его выполнение (build).

Примеры CASE-систем

- CA ERwin Data Modeler;
- MySQL Workbench;
- Enterprise Architect;
- Oracle SQL Developer Data Modeler;
- Microsoft Visio.

Проектирование снизу вверх

- Создается одна таблица со всеми атрибутами;
- Осуществляется процесс нормализации.

Информационная система

факультета

Предположим, что в результате первичного рассмотрения принято решение, что система будет использовать следующие атрибуты.

StudentNumber – № студбилета	SubjectCode – код дисциплины, сданной студентом
LastName – фамилия студента	SubjectName – наименование дисциплины
FirstName – имя студента	SubjectHours – количество часов, выделенных на дисциплину
PatrName – отчество студента	TeacherNumber – табельный номер преподавателя по дисциплине
Group – группа, в которой учится студент	TeacherName – фамилия и инициалы преподавателя
BirthDate – дата рождения студента	Chair – кафедра, на которой работает преподаватель
City – город проживания студента	TeacherPosition – должность преподавателя
Address – адрес в городе проживания	Grade – оценка по дисциплине
Salary – размер стипендии	

Информационная система факультета

- На основе этой информации формируется *универсальная таблица* – таблица, содержащая столбцы, соответствующие наименованиям атрибутов.
- Зная значения в паре столбцов (StudentNumber, SubjectCode), можно однозначно определить значения всех остальных атрибутов, то есть эту пару можно выбрать в качестве первичного ключа. Учитывая остальные свойства, можно заключить, что универсальная таблица информационной системы факультета находится в первой нормальной форме.
- Сокращенно структура приведенной таблицы записывается в виде
Faculty(StudentNumber, LastName, FirstName, PatrName, Group, BirthDate, City, Address, Salary, SubjectCode, SubjectName, SubjectHours, TeacherNumber, TeacherName, Chair, TeacherPosition, Grade).

Недостатки универсальной таблицы

- База данных информационной системы, состоящая первоначально из одной (универсальной) таблицы, обладает серьезными дефектами, называемыми избыточностью данных и аномалиями обновления и проявляющимися при добавлении, удалении и изменении строк таблицы.
- Во-первых, при добавлении информации о сданном экзамене мы должны снова вводить полную информацию о студенте. С одной стороны, это противоречит принципу «однократный ввод – многократное использование», а с другой – является источником потенциальных ошибок.
- Во-вторых, база данных в форме универсальной таблицы подвержена *аномалиям обновления*.

Аномалии обновления

- Аномалии вставки:
 - при вводе информации об экзамене мы должны повторить информацию о студенте;
 - при вводе информации о студенте мы должны ввести информацию о каком-нибудь экзамене, поскольку код дисциплины является частью первичного ключа, который не может принимать значение null.
- Аномалии удаления:
 - при удалении ошибочной информации о единственном экзамене студента будет удалена и вся информация о студенте.
- Аномалии замены:
 - при замене какого-либо значения в столбце, не относящемся к сданным экзаменам, надо будет заменить все значения в этом столбце, относящиеся к тому же студенту; если это будет сделано не для всех таких строк, это приведет базу данных в противоречивое состояние.

Нормализация 2НФ

- Атрибуты LastName, FirstName, PatrName, Group, BirthDate, City, Address, Salary функционально зависят только от атрибута StudentNumber, в то время как значение атрибута SubjectCode не оказывает на них никакого влияния.
- Для преобразования таблицы в приведенном примере к 2НФ она должна быть разделена на две связанных таблицы: в первую из них (родительскую) входит атрибут StudentNumber и все зависящие от него неключевые атрибуты, а во вторую (дочернюю) – весь первичный ключ таблицы Faculty и оставшиеся неключевые атрибуты.
- Student(StudentNumber, LastName, FirstName, PatrName, Group, BirthDate, City, Address, Salary),
Exam(StudentNumber, SubjectCode, SubjectName, SubjectHours, TeacherNumber, TeacherName, Chair, TeacherPosition, Grade)

Нормализация 2НФ

- Таблица Student находится в 2НФ по той причине, что первичный ключ в ней состоит только из одного атрибута. В таблице Exam атрибуты SubjectName и SubjectHours зависят только от атрибута SubjectCode, и поэтому она также должна быть разделена на две:
Subject(SubjectCode, SubjectName, SubjectHours),
Exam2(StudentNumber, SubjectCode, TeacherNumber, TeacherName, Chair, TeacherPosition, Grade).
- Таблица Exam2 является при этом дочерней по отношению к обеим таблицам Student и Subject.
- Полученная база данных из 3 таблиц уже находится в 2НФ, поскольку для определения значений неключевых атрибутов таблицы Exam2 необходимо знать оба ключевых атрибута, поскольку у разных студентов одну и ту же дисциплину могут вести разные преподаватели, следовательно, частичные функциональные зависимости в третьей таблице отсутствуют.

Нормализация 3НФ

- Тем не менее, преобразование базы данных к 2НФ не устраняет всех аномалий обновления. Это связано с тем, что неключевые атрибуты **TeacherName**, **Chair**, **TeacherPosition** функционально зависят только от неключевого же атрибута **TeacherNumber**. Используется тот же прием выделения новой таблицы.
- В итоге база данных информационной системы факультета, приведенная к 3НФ, содержит четыре таблицы:
- **Subject**(SubjectCode, SubjectName, SubjectHours);
Session(StudentNumber, SubjectCode, TeacherNumber, Grade);
Student(StudentNumber, LastName, FirstName, PatrName, Group, BirthDate, City, Address, Salary);
Teacher(TeacherNumber, TeacherName, Chair).

Идентифицирующие связи

- Таблица Session является дочерней по отношению к трем другим таблицам, но при этом:
 - столбцы связи с таблицами Student и Subject (внешние ключи) входят в состав первичного ключа таблицы Session;
 - столбец связи с таблицей Teacher является неключевым.
- В этом случае принято говорить, что связи таблицы Session с таблицами Student и Subject являются *идентифицирующими*, а связь с таблицей Teacher – *неидентифицирующей*.