

Тема 4.1

«Реляционные базы данных: проектирование и применение»

Вступление

В современных веб-ориентированных приложениях базы данных активно используются для хранения и обработки информации.

В сравнении с использованием файловой системы использование БД даёт следующие **преимущества**:

- унифицированный способ доступа к данным;
- готовые надёжные механизмы считывания и записи сложноструктурированных данных;
- возможность обрабатывать большие объёмы данных, не загружая их в пространство памяти РНР;
- дополнительный уровень безопасности;
- широкие возможности по поиску и сортировке данных.

Определения и факты

Здесь мы рассмотрим только небольшой набор самых важных понятий. Более подробному рассмотрению реляционных БД посвящён курс «Базы данных» (и ему подобные). Итак.

Реляционная база данных – база данных, основанная на реляционной модели. Слово «реляционный» происходит от английского «relation» («отношение»). Для работы с реляционными БД применяют реляционные СУБД.

Основными понятиями реляционных БД являются: «**ОТНОШЕНИЕ**» и «**СВЯЗЬ**».

Отношение (таблица) – набор равноправных (неупорядоченных) строк, содержащих одинаковый набор полей (атрибутов) (которые также являются равноправными и неупорядоченными). Поля имеют имена. Строки могут идентифицироваться ключами – значениями одного или нескольких своих полей (атрибутов).

Связь – «взаимодействие» между сущностями (то, как сущности между собой связаны).

Определения и факты

Связи бывают следующих видов:

1. **Один ко многим** (например, у одного человека есть несколько телефонов).
2. **Многие ко многим** (например, один преподаватель ведёт курс у многих студентов, и каждый студент обучается у многих преподавателей).
3. **Один к одному** (например, у одного человека есть один паспорт или одно личное дело). Используется редко.

Данные, представленные в базе данных, должны быть нормализованы. Существует 5 (+ одна, выделяемая неявно) нормальных форм. Мы рассмотрим кратко первые три, т.к. для наших повседневных задач их вполне хватает.

Итак...

Нормальные формы

Отношение (таблица) находится в первой нормальной форме (**1НФ**), если все атрибуты отношения (поля таблицы) являются простыми, т.е. не имеют компонентов.

В большинстве случаев выполнить это требование достаточно просто – каждый простой атрибут должен иметь свою колонку в таблице.

Например: если мы делали отдельные колонки (поля, атрибуты) для e-mail адреса и телефона – мы привели таблицу к **1НФ**, а если мы записываем эти данные в одно поле – мы нарушаем **1НФ**.

Нормальные формы

Отношение находится во второй нормальной форме (**2НФ**), если оно находится в **1НФ**, и все неключевые атрибуты отношения функционально полно зависят от составного ключа отношения.

Иными словами, 2НФ требует, чтобы отношение не содержало частичных функциональных зависимостей.

Рассмотрим на «плохом примере» (**отношение НЕ находится во 2НФ**).

Пусть у нас есть отношение **Грузоперевозки**, содержащее поля «**гос_номер_автомобиля**», «**грузоподъёмность**» и «**номер_рейса**», строка таблицы идентифицируется по полю «**гос_номер_автомобиля**».

Мы видим, что поле «**грузоподъёмность**» не зависит от поля «**номер_рейса**», но зависит от поля «**гос_номер_автомобиля**».

Такая ситуация порождает кучу проблем.

Нормальные формы

Проблемы:

- невозможно занести в базу данных госномер и грузоподъёмность автомобиля, который не выполнил ещё ни одного рейса – можно только ввести для него фиктивный рейс;
- если удалить запись из таблицы после завершения рейса, то мы потеряем информацию об автомобиле (если он в данный момент не находится в другом рейсе);
- невозможно отразить факт переоборудования автомобиля и получения им новой грузоподъёмности, т.к. получится, что он в прошлом ездил недогруженным или перегруженным.

Таким образом, эту таблицу нужно **нормализовать**, разбив на подтаблицы, например:

Автомобиль (госномер, грузоподъёмность)

и

Рейс (на каком автомобиле, откуда, куда, дата отправки, дата прибытия, вес, процент_загруженности, описание_груза).

Подчёркнуты ключевые поля.

В таком случае все вышеперечисленные проблемы становятся не актуальными.

Нормальные формы

Рассмотрим результат, полученный в предыдущем преобразовании ко 2НФ, а именно – таблицу

Рейс (на каком автомобиле, откуда, куда, дата отправки, дата прибытия, вес, процент_загруженности, описание_груза).

Здесь поле «описание_груза» функционально зависит от номера автомобиля, даты отправки, даты прибытия, места отправки и места назначения, что, в общем случае неверно, т.к. фактически, описание груза должно зависеть только от самого груза.

Мы можем преобразовать эту таблицу в две новые:

Рейс (на каком автомобиле, номер рейса, откуда, куда, дата_отправки, дата_прибытия, процент_загруженности)

и

Груз (для_какого_рейса, вес, описание_груза).

Т.о. отношение находится в третьей нормальной форме (3НФ), если оно находится во 2НФ, и все неключевые атрибуты отношения зависят **только** от **первичного ключа** (уникальное значение поля или сочетания полей, которое не может повторяться в двух и более строках).

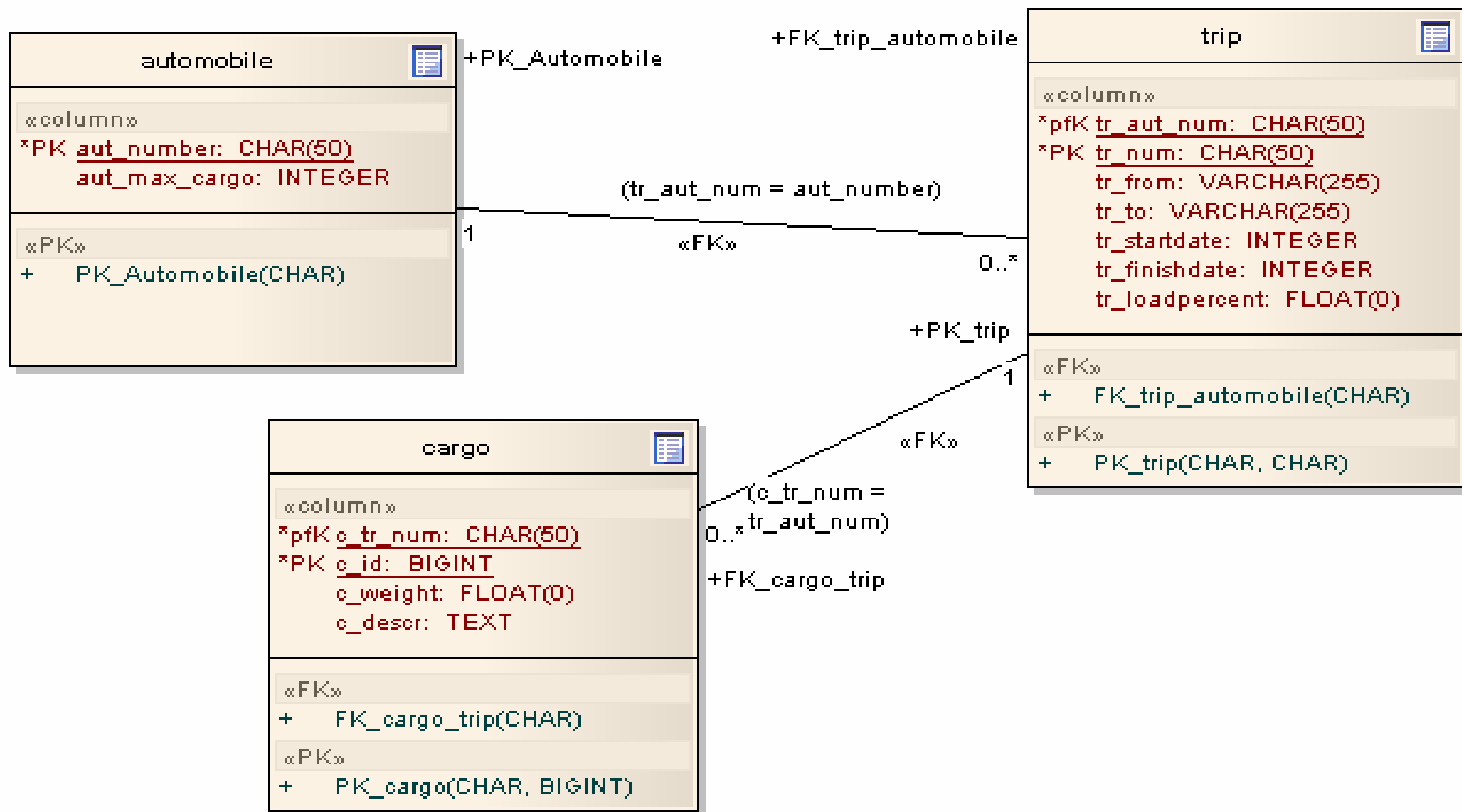
Проектирование баз данных

Представление нашей базы данных рассмотренным образом позволяет нам независимо и гибко управлять данными об автомобиле, назначать ему несколько рейсов (ставить их в очередь), а также назначать на каждый рейс произвольное количество грузов и свободно «перекидывать» их между рейсами, если возникнет такая необходимость.

При проектировании баз данных удобно использовать специальные **инструментальные средства**. Наиболее простыми из достаточно распространённых являются [Sparx Enterprise Architect](#) и [AllFusion ErWin Data Modeler](#). Они позволяют проектировать базы данных в визуальной форме, генерировать **SQL-код** для создания баз данных и выполнять множество иных операций.

Рассмотрим пример схемы нашей БД, выполненной в [Sparx Enterprise Architect](#).

Нормальные формы



Проектирование и применение БД

Т.о., **проектирование базы данных** включает:

1. Определение предметной области, выявление существующих в ней сущностей и связей между ними.
2. Представление сущностей и связей в виде таблиц базы данных и связей между ними.
3. Нормализация отношений.

Это – если **ОЧЕНЬ** кратко. Более подробно некоторые подзадачи мы рассмотрим далее на примерах.

Применение базы данных наиболее оправдано для хранения следующей информации:

- всей человекополезной немультимедийной информации (тексты новостей и сообщений, текстовое наполнение страниц и т.п.);
- некоторых настроек нашего программного обеспечения;
- структуры сайта;
- некоторой служебной информации (например, ip-адресов, действия с которых запрещены в настоящее время).