

# База Данных

## Лекция 4

# **Нормализация. Нормальные формы**

# Прошлая лекция

## Функциональные зависимости

- Полный (Full)
- Частичный (Partial)
- Транзитивный (Transitive)

## Аномалии обновления данных

- Модификации
- Вставки
- Удаления

# Сегодняшняя лекция

- Мы поговорим о нормализации и нормальных формах.
- Вы узнаете, какая нормальная форма соответствует “хорошему” дизайну
- Мы обсудим, как мы можем улучшить дизайн базы данных

# Нормализация

- Нормализация используется для проверки корректности логической модели данных. Логическая модель данных является источником информации для следующего этапа - проектирования физической базы данных.
- Нормализация – это процесс разложения сложных таблиц на простые таблицы с целью устранения нежелательных функциональных зависимостей и избыточности данных в сложных таблицах.

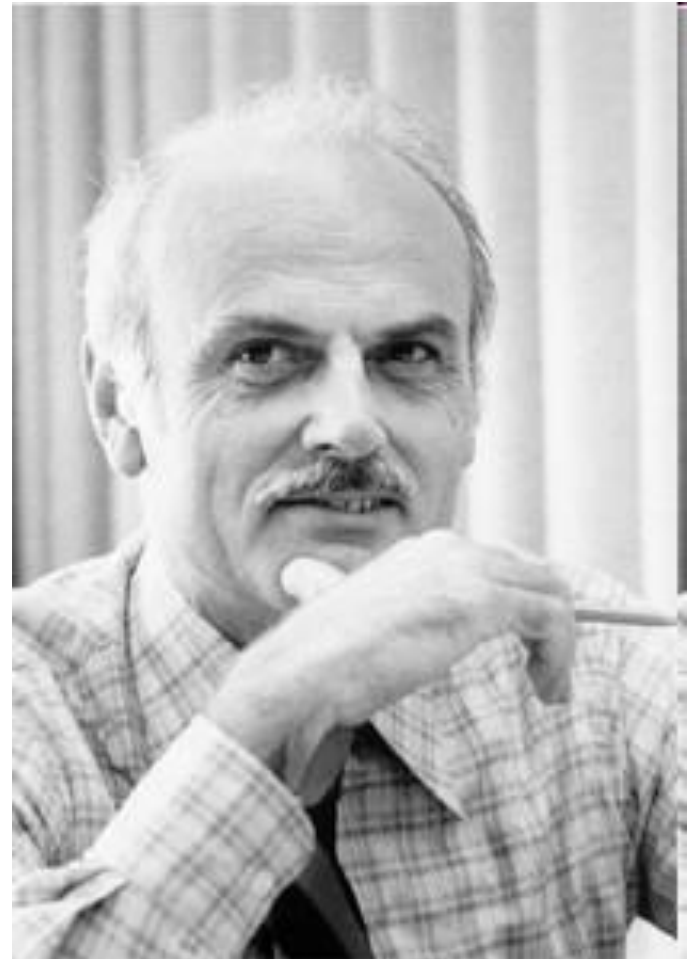
# Нормальные формы

- Нормальные формы (NF) в теории реляционных баз данных предоставляют критерии для определения степени защищенности таблицы от логических несоответствий и аномалий.
- Нормальная форма базы данных – это набор правил и критериев, которым должна отвечать база данных. Каждая следующая нормальная форма содержит более строгие правила и критерии, тем самым приводя базу данных к определённой нормальной форме, мы устраняем определённый набор аномалий

# История

Эдгар Ф. Кодд, изобретатель реляционной модели, ввел концепцию нормализации и то, что мы сейчас знаем как Первую нормальную форму (1NF), в 1970 году.

Кодд дал определение Второй нормальной форме (2NF) и Третьей нормальной форме (3NF) в 1971 году, а Кодд и Рэймонд Ф. Бойс определили нормальную форму Бойса-Кодда (BCNF) в 1974 году.





# “Хороший” дизайн

- Нормализация выполняется в виде последовательности шагов. Каждый шаг соответствует определенной нормальной форме, обладающей известными свойствами.
- По мере продолжения нормализации отношения становятся все более ограниченными (более строгими) по формату, а также менее уязвимыми для аномалий обновления.
- 3NF - это стандартная нормальная форма, в которой может находиться отношение, чтобы считаться “хорошим” дизайном.
- Таблицы в 3NF не содержат аномалий вставки, модификации (обновления) и удаления.



# Первая нормальная форма (1NF)

- Отношение находится в первой нормальной форме (1NF), когда соблюдаются следующие реляционные принципы:
  - в таблице не должно быть дублирующих строк;
  - в каждой ячейке таблицы хранится атомарное значение (одно не составное значение);
  - в столбце хранятся данные одного типа;
  - отсутствуют массивы и списки в любом виде.
- Отношение, которое не в 1NF, называется ненормализованным или UNF (0NF). Отношение, которое находится в UNF, будет состоять из многозначных атрибутов.

# Первая нормальная форма

- В таблице ниже хранится информация о преподавателях и кафедрах.
- Эта таблица находится в формате 1NF, поскольку каждый атрибут является атомарным.

Teacher_id	Last_name	Dep_id	Dep_name	Room
001	Teacher1	001	CET	409
002	Teacher2	001	CET	409
003	Teacher3	002	IS	803

# Первая нормальная форма

- Рассмотрим приведенное ниже отношение.
- Оно находится в UNF (не в 1NF).

<b>Teacher_id</b>	<b>Last_name</b>	<b>Degree</b>
001	Teacher1	{M.D., PhD}
002	Teacher2	{M.D.}
003	Teacher3	{M.D., PhD}

# Вторая нормальная форма (2NF)

- Отношение, находящееся во Второй нормальной форме (2NF), находится в 1NF и не имеет частичных зависимостей от первичного ключа (ПК).
- Вторая нормальная форма связана с
- аномалией модификации.

# Частичная зависимость

- Зависимость  $X \rightarrow Y$  является частичной зависимостью, если существует атрибут  $A$ , являющийся частью  $X$ , который может быть удален из  $X$ , и зависимость все еще сохраняется.
- Пример: Таблица преподавателей и курсов.
- РК - это  $\{teacher\_id, course\_id\}$

Teacher_id	Last_name	Course_id	Course_name	Credits
001	Teacher1	001	SDP1	3
001	Teacher1	002	SDP2	3
002	Teacher2	001	SDP1	3

# Частичная зависимость

- Также предположим, что это отношение имеет следующие зависимости:
- FD1: {teacher\_id, course\_id} -> {last\_name, course\_name, credits}
- FD2: {teacher\_id} -> {last\_name}
- FD3: {course\_id} -> {course\_name, кредиты}
  
- FD1 - частичная зависимость
- FD2 и FD3 - полные зависимости.

# Третья нормальная форма (3NF)

- Отношение находится в третьей нормальной форме (3NF), если оно находится в 2NF и не содержит атрибутов, которые имеют транзитивную зависимость.
- Третья нормальная форма определяется в терминах транзитивных зависимостей и связана с аномалиями вставки и удаления.

# Транзитивная зависимость

- Транзитивная зависимость - это условие, при котором  $X$ ,  $Y$  и  $Z$  являются атрибутами отношения, так что если  $X \rightarrow Y$  и  $Y \rightarrow Z$ , то  $Z$  транзитивно зависит от  $X$  через  $Y$ .
- Пример: Таблица Students and Groups.
- PK - идентификатор студента.

Student_id	Last_name	Group_id	Group_name
001	Student1	001	Group1
002	Student2	001	Group1
003	Student3	002	Group2



# Транзитивная зависимость

- Отношение имеет следующие функциональные зависимости:
  - {student\_id} -> {last\_name, group\_id, group\_name}
  - {group\_id} -> {group\_name}
- Это отношение содержит транзитивную зависимость, потому
- что {student\_id} -> {group\_id} -> {group\_name}

# Декомпозиция

- Декомпозиция - это процесс разбиения целого на части или элементы.
- Декомпозиция разбивает сложную таблицу на несколько таблиц в базе данных.
- Этот процесс должен происходить без потерь данных, поскольку данные в исходную таблицу должны быть точно восстановлены на основе разложенных таблиц.

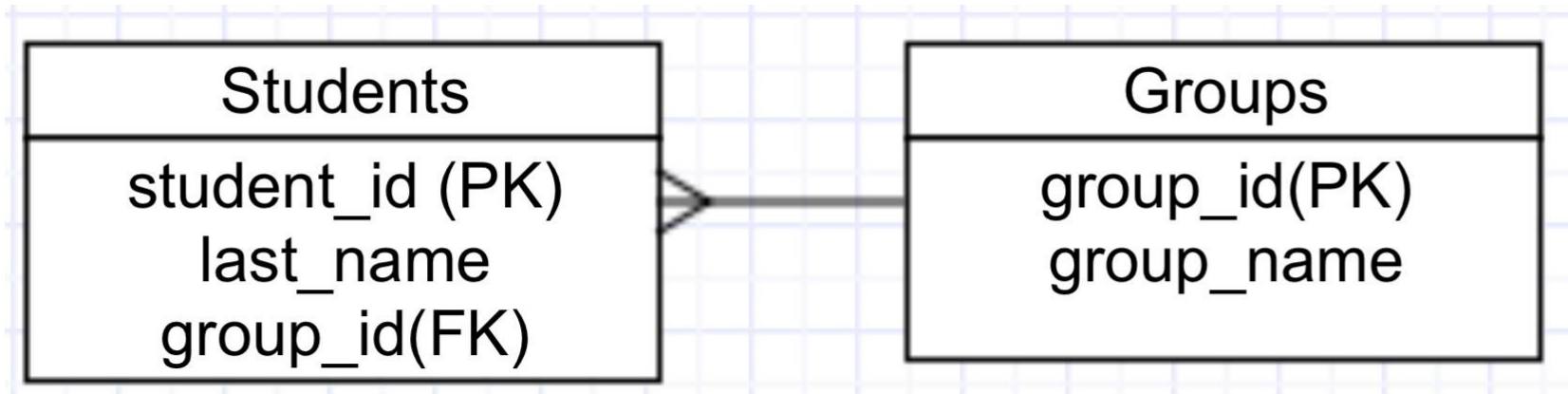
# Нормализация

- Давайте нормализуем две таблицы, которые мы рассмотрели.
- Во-первых, таблицу, содержащую данные о студентах и группах.

Student_id	Last_name	Group_id	Group_name
001	Student1	001	Group1
002	Student2	001	Group1
003	Student3	002	Group2

# Нормализация

- Мы сказали, что это отношение (таблица) содержит транзитивную зависимость и, таким образом, нарушает третью нормальную форму. Нам нужно удалить транзитивную зависимость из таблицы.
- Мы делаем это, разбивая (декомпозируя) отношение на две таблицы: Students и Groups.



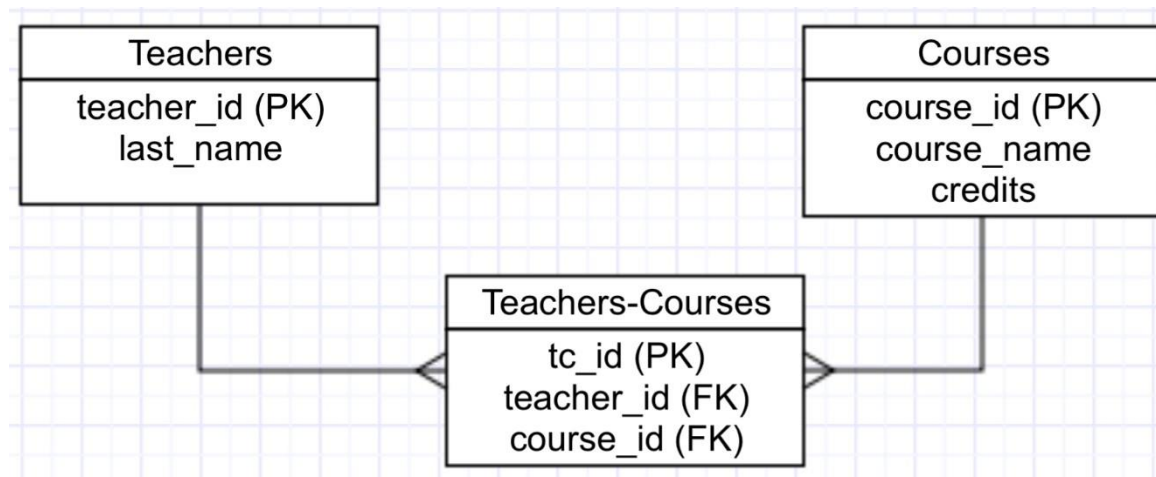
# Нормализация

- Теперь рассмотрим таблицу Преподавателей и Курсов.
- Составной первичный ключ (ПК) этого отношения был {teacher\_id, course\_id}.

Teacher_id	Last_name	Course_id	Course_name	Credits
001	Teacher1	001	SDP1	3
001	Teacher1	002	SDP2	3
002	Teacher2	001	SDP1	3

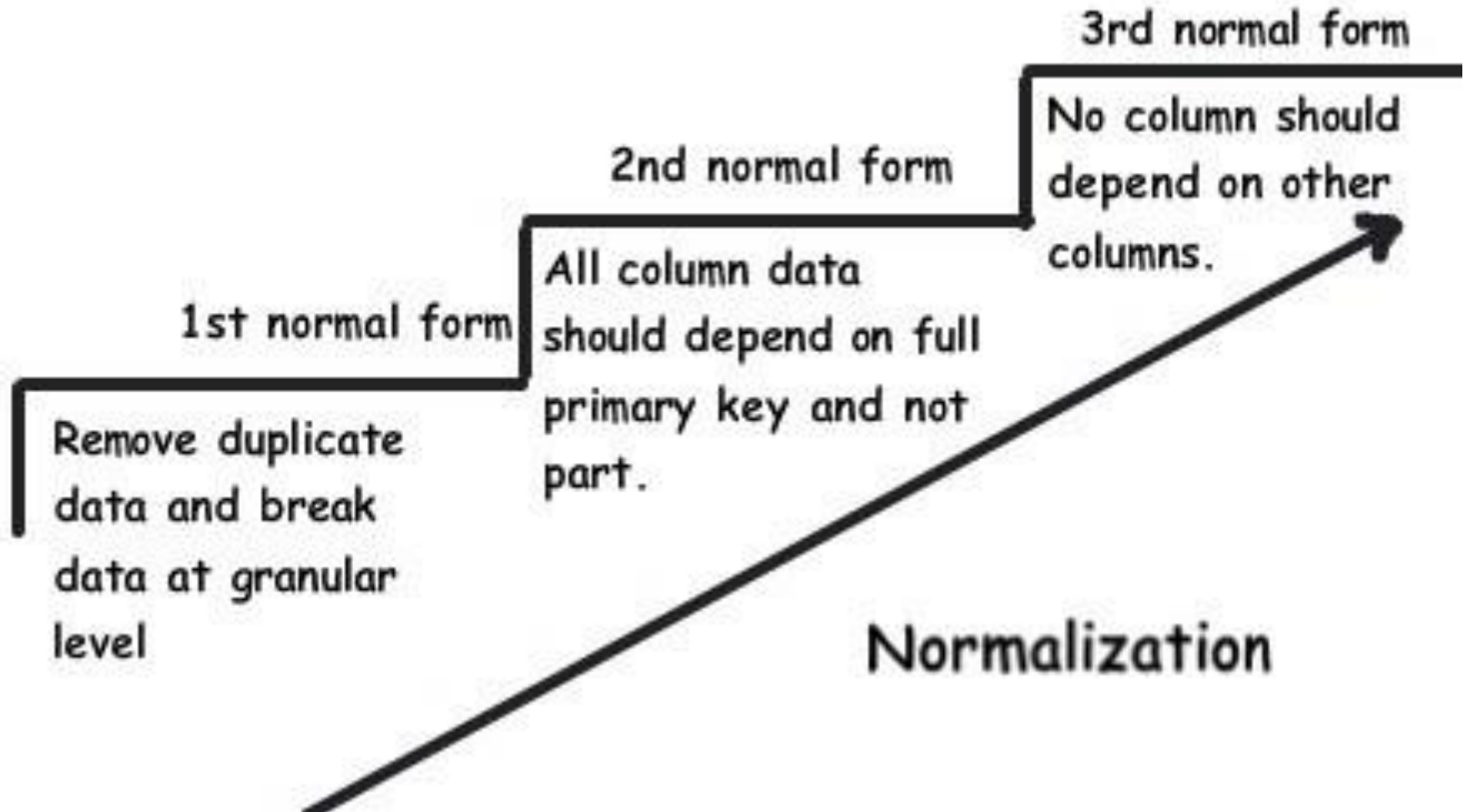
# Нормализация

- Также напомним, что функциональные зависимости (FD) для этого отношения были следующими
- FD1: {teacher\_id, course\_id} -> {last\_name, course\_name, credits}
- FD2: teacher\_id -> last\_name
- FD3: course\_id -> course\_name, credits
- Из-за частичных зависимостей этого отношения оно нарушает 2NF. Чтобы привести его к 2NF, мы создаем три отношения.



# Выводы

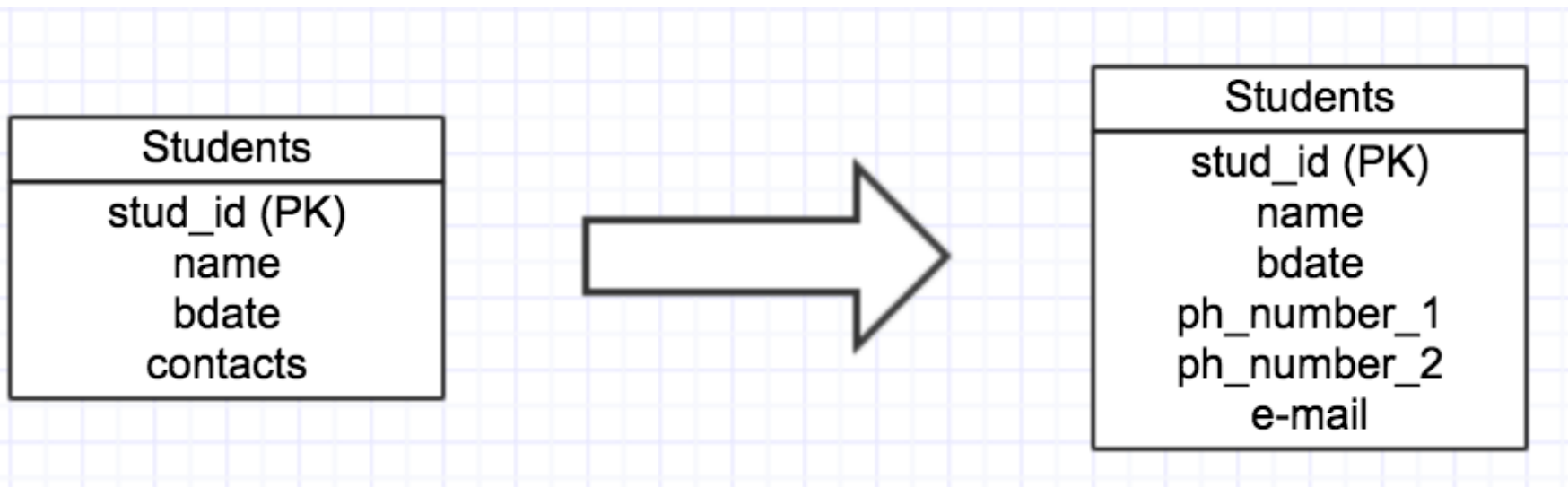
- Как правило, отношение (таблица) в 3NF считается хорошим дизайном базы данных – процесс нормализации обычно заканчивается, как только достигается значение 3NF.
- Как мы удаляем функциональные зависимости, которые нарушают заданную нормальную форму?
- Мы разбиваем отношение на множество более мелких отношений, сохраняя при этом взаимосвязь между данными.





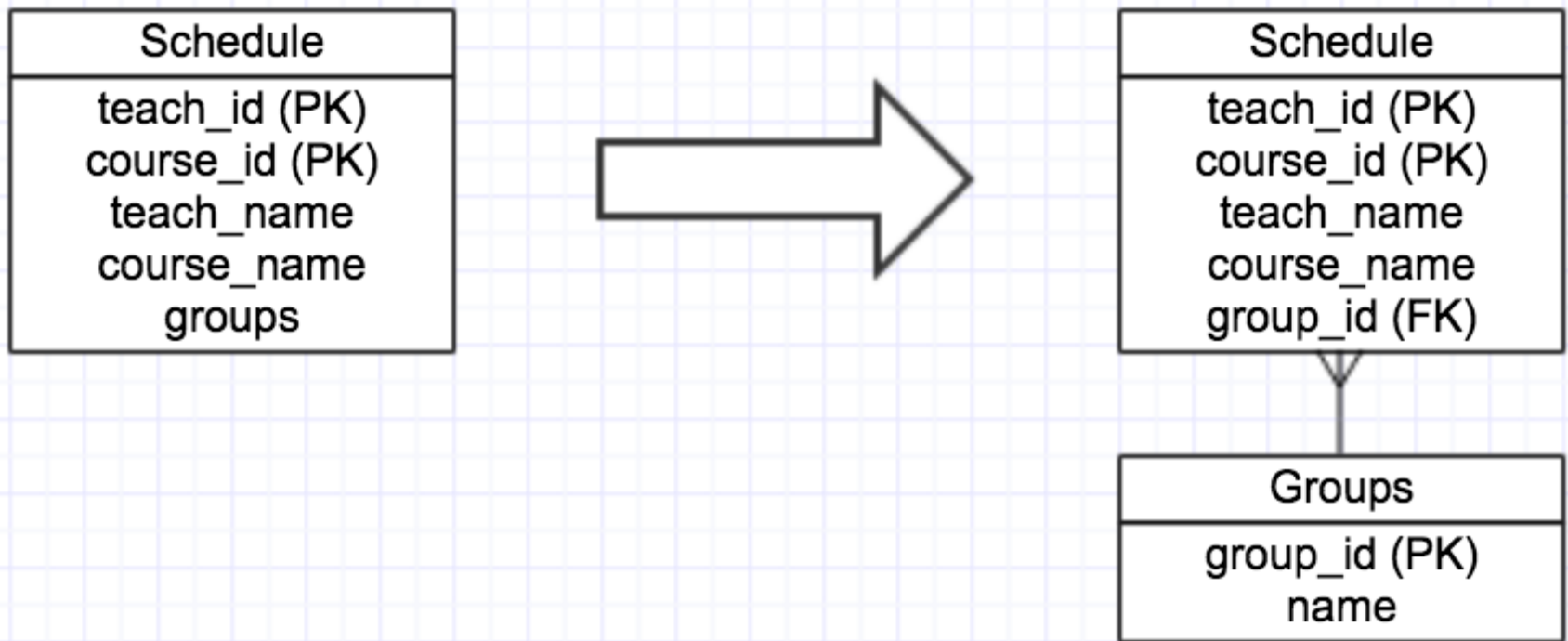
# Примеры

# CASE 1: от UNF до 1NF (версия 1)



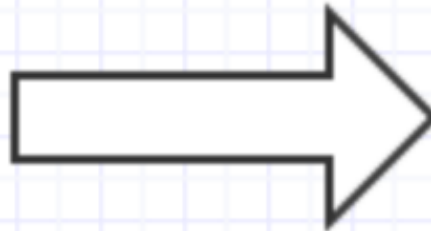
Контакты были разделены следующим образом: ph\_number\_1, ph\_number\_2, e-mail

# CASE 1: от UNF до 1NF (версия 2)



# Случай 2: от 1NF до 2NF

Schedule
teach_id (PK, FK)
course_id (PK, FK)
teach_name
course_name
group_id (FK)



Schedule
sch_id (PK)
teach_id (FK)
course_id (FK)
teach_name
course_name
group_id (FK)

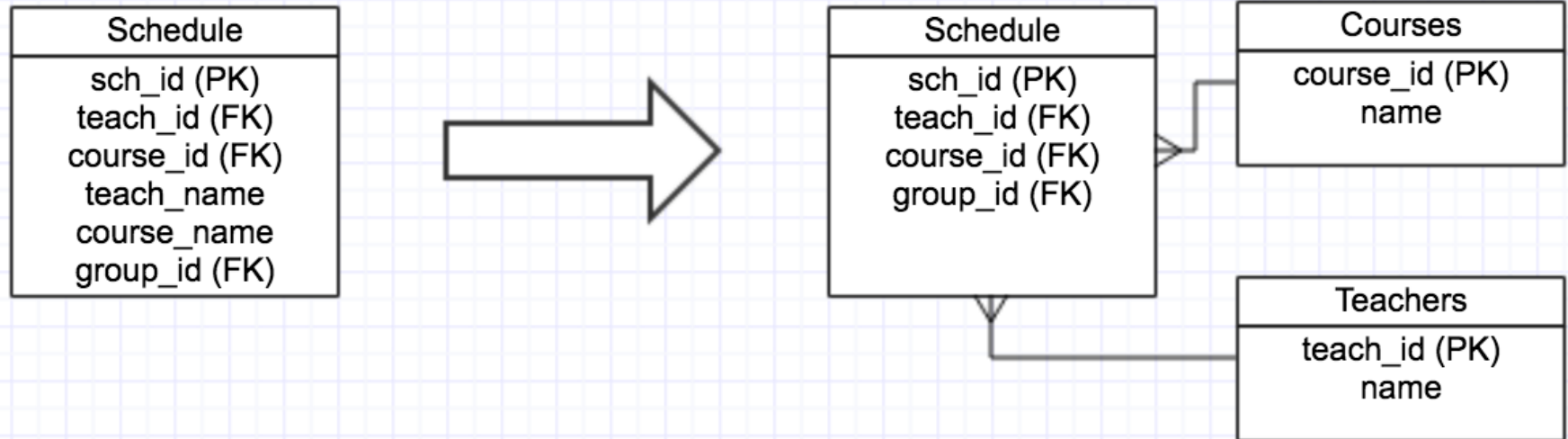
До:

FD1: {teach\_id, course\_id} -> {teach\_name, course\_name, group\_id}

FD2: {teach\_id} -> {teach\_name}

FD3: {course\_id} -> {course\_name}

# Случай 3: от 2NF до 3NF



До:

FD 1: {sch\_id} -> {teach\_id, course\_id, teach\_name, course\_name, group\_id}

FD2: {tech\_id} -> {tech\_name}

FD3: {course\_id} -> {course\_name}

# КНИГИ

- Коннолли, Томас М. Системы баз данных: практический подход к проектированию, внедрению и управлению / Томас М. Коннолли, Кэролин Э. Бегг.- Соединенные Штаты Америки: Pearson Education
- Гарсия-Молина, Х. Система баз данных: Полная книга / Эктор Гарсия-Молина.- Соединенные Штаты Америки: Пирсон Прентис Холл
- Шарма, Н. Основы баз данных: Книга для сообщества от сообщества / Нирадж Шарма, Ливиу Перниу.- Канада
- Э.Ф. Кодд, "Дальнейшая нормализация реляционной модели базы данных"