The background is a light blue grid. A solid blue horizontal line is positioned near the top, and another solid blue horizontal line is positioned near the bottom. A solid blue vertical line is on the left side, and another solid blue vertical line is on the right side. At the top-left corner, there is a blue L-shaped line with a small semi-circle at the corner. At the bottom-right corner, there is a blue L-shaped line with a small semi-circle at the corner.

Проектирование баз данных

Цели проектирования

- ◆ Определение всех необходимых данных
- ◆ Определение необходимого набора таблиц (отношений)
- ◆ Определение ограничений

Результат проектирования – схема базы данных

Как определить какие данные нужны?

- ◆ Моделируем предметную область
- ◆ Рассматриваем известные данные, связи между ними, процессы (бизнес-процессы)
- ◆ Разделяем на подзадачи
- ◆ Используются UML и case-средства

Избыточность первичных данных

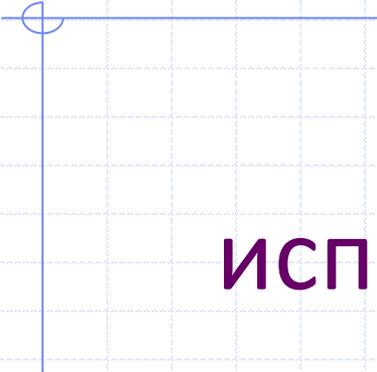
- ◆ Одни и те же данные в разных задачах/подсистемах
- ◆ Дублирование информации в рамках одной задачи
 - Одни и те же данные в разных отношениях
 - Одни и те же данные в одном отношении

Таблицы (отношения)

- ◆ Часто изначально проектируются на основе уже существующих таблиц
- ◆ В процессе проектирования могут подвергаться преобразованиям
- ◆ Противоречие
 - Разделение одного отношения на несколько более мелких отношений
 - Минимизация количества отношений, чтобы избежать множества join в запросах

Таблица

- ◆ Все начинается с таблицы (отношения)
- ◆ Целостность таблицы
 - на пересечении строки и столбца стоят атомарные значения
 - все строки различны (существует непустое подмножество атрибутов – возможный ключ, однозначно определяющих каждую строку)



Проблемы, связанные с использованием минимального количества отношений

Пример таблицы для базы данных «Тьютор»

Исходная таблица

Это еще не отношение !!!

студ#	ФИО	комн#	тел#	Курс	Предм	Балл
m3215	Иванов И.И.	120DH	2136123	PM16	MA84	87
					AG85	75
					PL83	60
i2345	Петров П.П.	238MP	2345671	IT211	MA84	70
					PL84	85
f1234	Сидоров С.С.	120DH	2136123	AS12	TP12	62

Пример отношения для базы данных «Тьютор»

студ#	ФИО	комн#	тел#	Курс	Предм	Балл
m3215	Иванов И.И.	120DH	2136123	PM16	MA84	87
m3215	Иванов И.И.	120DH	2136123	PM16	AG85	75
m3215	Иванов И.И.	120DH	2136123	PM16	PL83	60
i2345	Петров П.П.	238MP	2345671	IT211	MA84	70
i2345	Петров П.П.	238MP	2345671	IT211	PL84	85
f1234	Сидоров С.С.	120DH	2136123	AS12	TP12	62

Возможный ключ – {студ#, Курс, Предм}

Чердынцева М.И., ИММиКН ЮФУ

2023

9

Проблема вставки

студ#	ФИО	КОМН#	тел#	Курс	Предм	Балл
m3215	Иванов И.И.	120DH	2136123	PM16	MA84	87
m3215	Иванов И.И.	120DH	2136123	PM16	AG85	75
m3215	Иванов И.И.	120DH	2136123	PM16	PL83	60
i2345	Петров П.П.	238MP	2345671	IT211	MA84	70
i2345	Петров П.П.	238MP	2345671	IT211	PL84	85
f1234	Сидоров С.С.	120DH	2136123	AS12	TP12	62
m3456	Егоров Е.Е.	121MP	2123341	PM11	PL84	0

```
insert into TUTOR values('m3456','Егоров Е.Е.',  
                          '121MP','2123341','PM11','PL84',0)
```

```
select ФИО from TUTOR where Балл<60
```

Проблема обновления

студ#	ФИО	комн#	тел#	Курс	Предм	Балл
m3215	Иванов И.И.	120DH	2136123	PM16	MA84	87
m3215	Иванов И.И.	120DH	2136123	PM16	AG85	75
m3215	Иванов И.И.	120DH	2136123	PM16	PL83	60
i2345	Петров П.П.	238MP	2345671	IT211	MA84	70
i2345	Петров П.П.	238MP	2345671	IT211	PL84	85
f1234	Сидоров С.С.	120DH	2177777	AS12	TP12	62

update TUTOR set тел#='2177777' where ФИО='Сидоров С.С.'
select тел# from TUTOR where комн# = '120DH'

Проблема удаления

студ#	ФИО	комн#	тел#	Курс	Предм	87
m3215	Иванов И.И.	120DH	2136123	PM16	MA84	75
m3215	Иванов И.И.	120DH	2136123	PM16	AG85	60
m3215	Иванов И.И.	120DH	2136123	PM16	PL83	70
i2345	Петров П.П.	238MP	2345671	IT211	MA84	85
i2345	Петров П.П.	238MP	2345671	IT211	PL84	62
f1234	Сидоров С.С.	120DH	2177777	AS12	TP12	87

delete from TUTOR

where ФИО='Сидоров С.С.' and Предм='TP12'

select * from TUTOR where ФИО='Сидоров С.С.'

Чердынцева М.И., ИММиКН ЮФУ

2023

12

Как решить проблемы?

- ◆ Может быть помогут NULL
- ◆ Можно более жестко контролировать операции в приложении
- ◆ Выделить часть информации в отдельную таблицу
 - Какую информацию?
 - Достаточно ли будет еще одной таблицы или нужно больше?
 - Возможно ли будет восстановить исходную таблицу после этого?
- ◆ В основе решения – определение свойств отношения и разбиение на большее количество отношений путем декомпозиции без потерь

Нормализация путем декомпозиции

- ◆ Декомпозиция отношения на два или более отношений с целью разрешения проблем вставки, обновления и удаления данных
- ◆ Декомпозиция без потерь – операция join всегда позволит восстановить исходную таблицу
- ◆ Нормальная форма – способ классификации таблиц на основе содержащихся в них функциональных зависимостей (functional dependencies, FD)

Функциональные зависимости

- ◆ Функциональная зависимость устанавливает связь между двумя подмножествами атрибутов внутри одной переменной-отношения
- ◆ Виды функциональных зависимостей
 - которые могут быть определены всегда, такие ФЗ можно считать ограничениями целостности
 - которые могут определяться для конкретного момента времени (для конкретных кортежей), могут приводить к ошибкам в проектировании

Функциональные зависимости

$F: \{A\} \rightarrow \{B\}$



А и В непересекающиеся множества атрибутов

Если имеется ФЗ $F: \{X\} \rightarrow \{Y\}$,

то X – детерминант, Y – зависимая часть

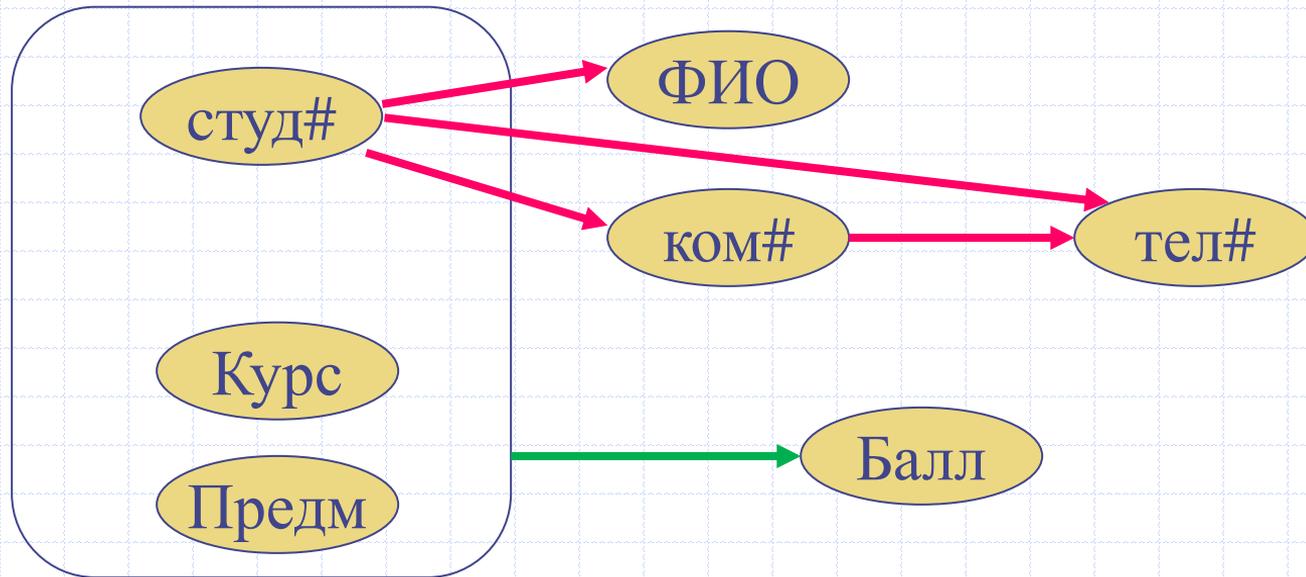
Возможный ключ

- ◆ Если ФЗ определяет в отношении зависимость всего кортежа от набора атрибутов (т.е. гарантирует основное свойство отношения-уникальность кортежей), тогда детерминант ФЗ является потенциальным или возможным ключом
- ◆ Отношение может иметь несколько потенциальных ключей
- ◆ Отношение всегда имеет один возможный ключ, совпадающий с полным набором атрибутов отношения

Функциональные зависимости

- ◆ Если X является потенциальным (возможным) ключом, то все не входящие в X атрибуты должны быть функционально зависимы от X
- ◆ Если переменная-отношение R удовлетворяет ФЗ $F: \{A\} \rightarrow \{B\}$ и A не является потенциальным ключом, то R будет характеризоваться некоторой избыточностью, приводящей к проблемам вставки/редактирования/удаления

Схема ФЗ для отношения «Тьютор»



Функциональные зависимости

- ◆ Количество определенных для переменной-отношения R функциональных зависимостей F может быть достаточно велико
- ◆ Контроль такого большого количества функциональных зависимостей может быть затруднен
- ◆ Следует минимизировать количество ФЗ, т.е. определим S

$$S \subset F$$

так, чтобы любая ФЗ из F могла быть получена на основе ФЗ из S

Правила вывода новых ФЗ

◆ Аксиомы Армстронга

R – кортежная переменная

$\{A\}, \{B\}, \{C\}, \{D\}$ – подмножества атрибутов в R

A1. Рефлексивность если $B \subset A$, то $A \rightarrow B$ (тривиальная ФЗ)

A2. Дополнение если $A \rightarrow B$, то $\{A \cup C\} \rightarrow \{B \cup C\}$

A3. Транзитивность если $A \rightarrow B$ и $B \rightarrow C$, то $A \rightarrow C$

Данный набор правил является полным и непротиворечивым

Правила вывода новых ФЗ

◆ Дополнительные правила

Д1. Самоопределение $A \rightarrow A$

Д2. Декомпозиция если $A \rightarrow \{B \cup C\}$, то $A \rightarrow B$ и $A \rightarrow C$

Д3. Объединение если $A \rightarrow B$ и $A \rightarrow C$, то $A \rightarrow \{B \cup C\}$

Д4. Композиция

если $A \rightarrow B$ и $C \rightarrow D$, то $\{A \cup C\} \rightarrow \{B \cup D\}$

◆ Общая теорема объединения

T0. Если $A \rightarrow B$ и $C \rightarrow D$, то $\{A \cup C \setminus B\} \rightarrow \{B \cup D\}$

Функциональные зависимости

◆ Замыкание множества ФЗ

- Если S – множество ФЗ переменной R , то все ФЗ, которые следуют из S , образуют замыкание S^+
- Строим такое S , замыкание которого S^+ совпадает с F

◆ Сокращение множества ФЗ

- Функциональная зависимость *тривиальна* тогда и только тогда, когда ее правая (зависимая) часть является подмножеством ее левой части (детерминанта)
- Будем рассматривать только нетривиальные ФЗ

Пример

$R(a,b,c,d,e,f)$

ФЗ:

$\{a\} \rightarrow \{b,c\}$

$\{b\} \rightarrow \{e\}$

$\{c,d\} \rightarrow \{e,f\}$

есть ли ФЗ $\{a,d\} \rightarrow \{f\}$?

$\{a\} \rightarrow \{b,c\}$

Д2

$\{a\} \rightarrow \{c\}$ $\{a\} \rightarrow \{b\}$

А2

$\{a,d\} \rightarrow \{c,d\}$

А3

$\{a,d\} \rightarrow \{e,f\}$

Д2

$\{a,d\} \rightarrow \{f\}$ $\{a,d\} \rightarrow \{e\}$

Что нужно проверить?

- ◆ Является ли детерминант потенциальным ключом

Пусть Z – интересующее нас множество атрибутов (проверяемый детерминант)

Обозначим $S^Z \subset S$, которое состоит из всех функциональных зависимостей с множеством атрибутов Z

Обозначим через Z^+ множество атрибутов, которые функционально зависимы от Z (замыкание множества атрибутов)

Замыкание множества атрибутов

- ◆ ФЗ $f:\{X\} \rightarrow \{Y\}$ будет принадлежать замыканию S^Z , если $Y \subset Z^+$
- ◆ Z является возможным (потенциальным) ключом R , если Z^+ содержит все атрибуты R

Замыкание множества атрибутов

Алгоритм

$$Z^+ = Z$$

повторять пока Z^+ изменяется

для каждой ФЗ $f:\{X\} \rightarrow \{Y\}$

если $\{X\} \in Z^+$, то добавить $\{Y\}$

$$Z^+ = Z^+ \cup \{Y\}$$

Пример

$R(a,b,c,d,e,f)$

$\{a\} \rightarrow \{b,c\}$

$\{e\} \rightarrow \{c,f\}$

$\{b\} \rightarrow \{e\}$

$\{c,d\} \rightarrow \{e,f\}$

Найти замыкание для множества атрибутов $\{a,b\}$

Пример

$$Z^+ = Z = \{a, b\}$$

$$\{a\} \rightarrow \{b, c\} \quad Z^+ = Z^+ \cup \{b, c\} = \{a, b, c\}$$

$$\{e\} \rightarrow \{c, f\}$$

$$\{b\} \rightarrow \{e\} \quad Z^+ = Z^+ \cup \{e\} = \{a, b, c, e\}$$

$$\{c, d\} \rightarrow \{e, f\}$$

Пример

$$Z^+ = \{a, b, c, e\}$$

$$\{a\} \rightarrow \{b, c\}$$

$$\{e\} \rightarrow \{c, f\}$$

$$\{b\} \rightarrow \{e\}$$

$$\{c, d\} \rightarrow \{e, f\}$$

$$Z^+ = Z^+ \cup \{c, f\} = \{a, b, c, e, f\}$$

Пример

$Z^+ = \{a, b, c, e, f\}$

$R(a, b, c, d, e, f)$

$\{a\} \rightarrow \{b, c\}$

$\{e\} \rightarrow \{c, f\}$

$\{b\} \rightarrow \{e\}$

$\{c, d\} \rightarrow \{e, f\}$

замыкание $Z^+ = \{a, b, c, e, f\}$

Пример

$$\mathbb{Z}^+ \neq \mathbb{R}$$

значит, $\{a, b\}$ не является возможным ключом

Показать, что возможным ключом является $\{a, d\}$

Неприводимое множество ФЗ

- ◆ Множество ФЗ называется неприводимым слева, если
 - Зависимые части – одноэлементные множества
 - Каждый детерминант является неприводимым (ни один атрибут не может быть исключен без изменения замыкания)
 - Ни одна ФЗ не может быть исключена, без изменения замыкания
- ◆ Для любого множества ФЗ существует по крайней мере одно эквивалентное множество, которое является неприводимым слева

Пример

$R(a,b,c,d)$

$\{a\} \rightarrow \{b,c\}$

$\{b\} \rightarrow \{c\}$

$\{a\} \rightarrow \{b\}$

$\{a,b\} \rightarrow \{c\}$

$\{a,c\} \rightarrow \{d\}$

Построить неприводимое множество ФЗ

Пример

Используем аксиомы

$$\{a\} \rightarrow \{b,c\}$$



$$\{a\} \rightarrow \{b\}$$

$$\{a\} \rightarrow \{c\}$$

$$\{b\} \rightarrow \{c\}$$

$$\{a\} \rightarrow \{b\}$$

$$\{a,b\} \rightarrow \{c\}$$

$$\{a,c\} \rightarrow \{d\}$$

Пример

Используем аксиомы

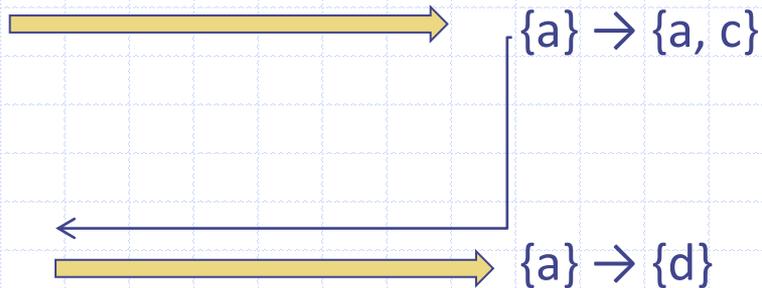
$$\{a\} \rightarrow \{b\}$$

$$\{a\} \rightarrow \{c\}$$

$$\{b\} \rightarrow \{c\}$$

$$\{a,b\} \rightarrow \{c\}$$

$$\{a,c\} \rightarrow \{d\}$$



Пример

Используем аксиомы

$$\{a\} \rightarrow \{b\}$$

$$\{a\} \rightarrow \{c\}$$

$$\{b\} \rightarrow \{c\}$$

$$\{a,b\} \rightarrow \{c\}$$

$$\{a\} \rightarrow \{d\}$$


$$\{a,b\} \rightarrow \{c\}$$

избыточная

Пример

Используем аксиомы

$$\{a\} \rightarrow \{b\}$$

$$\{a\} \rightarrow \{c\}$$

$$\{b\} \rightarrow \{c\}$$

$$\{a\} \rightarrow \{d\}$$



$$\{b\} \rightarrow \{c\}$$

избыточная (транзитивная)

Пример

Неприводимое множество зависимостей

$\{a\} \rightarrow \{b\}$

$\{a\} \rightarrow \{c\}$

$\{a\} \rightarrow \{d\}$

$R(a,b,c,d)$

Можно показать, что $\{a\}$ – потенциальный ключ

Нормальная форма Бойса-Кодда

- ◆ Переменная отношения находится в НФБК тогда и только тогда, когда каждая нетривиальная и неприводимая слева ФЗ имеет в качестве своего детерминанта возможный ключ
- ◆ Для большинства отношений НФБК гарантирует отсутствие аномалий (проблем) вставки, редактирования и удаления

Нормальная форма

Свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных.

Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

Поэтапное введение нормальных форм

1НФ

2НФ

3НФ

НФБК(усиленная 3НФ)

1НФ

- ◆ Отношение находится в первой нормальной форме (1НФ), если каждый его элемент имеет атомарное значение
- ◆ Для отношения в первой нормальной форме может быть определен набор атрибутов (возможный ключ), однозначно определяющий каждый кортеж отношения
- ◆ Среди возможных ключей выбирается один - первичный ключ

2НФ

- ◆ Отношение находится во второй нормальной форме, если оно находится в первой нормальной форме и не существует функциональных зависимостей неключевых атрибутов от части возможного ключа

3НФ

- ◆ Отношение находится в третьей нормальной форме, если оно находится во второй нормальной форме и не существует транзитивных функциональных зависимостей от ключевых атрибутов

НФБК (усиленная ЗНФ)

- ◆ Переменная отношения находится в НФБК тогда и только тогда, когда каждая нетривиальная и неприводимая слева ФЗ имеет в качестве своего детерминанта возможный ключ

Более высокие НФ

- ◆ 4НФ
- ◆ 5НФ
- ◆ ДКНФ (доменно-ключевая)
- ◆ 6НФ