

# Информационная система нормализации баз данных

Студент группы ИУ5-121

Кащавцев Максим

# Цель разработки

- Целью разработки является создание системы, позволяющей при проектировании базы данных, проанализировав список известных функциональных зависимостей предложить вариант «хорошей» схемы базы данных, а также предупредить о наличии условных взаимных функциональных зависимостях. Разрабатываемая система значительно упрощает процесс создания базы данных, учета функциональных зависимостей.

# Актуальность

- Актуальность работы обусловлена тем, что большинство программ для проектирования схем баз данных зачастую производят нормализацию до 3НФ, не учитывая взаимные функциональные зависимости. Может случиться ситуация, когда данные будут противоречивы, при этом сама БД «будет считать» что всё хорошо. Подобная ошибка может быть обнаружена слишком поздно. Далее будет приведен пример.

# Задачи, решенные в данной ИС

- 1. Разработка структуры данных, оптимизированной для хранения атрибутов и ФЗ
- 2. Реализация алгоритма поиска канонического покрытия
- 3. Реализация алгоритма поиска ключа отношения
- 4. Реализация алгоритма приведения в взаимно-независимой нормальной форме
- 5. Реализация правила объединения ФЗ с одинаковой левой частью
- 6. Реализация логирования работы алгоритмов.

# Назначение разработки

- Программный продукт предназначен для проектировщиков баз данных в различных организациях. Система позволяет нормализовать схему базы данных на основе введенных ФЗ, отыскать в них скрытые УВФЗ. Данная ИС также может являться вспомогательным средством проверки уже созданной схемы БД на скрытые УВФЗ.

# Описание Предметной области

- 1НФ - Все значения атрибутов должны быть атомарными (атомарное значение – не являющееся множеством значений или повторяющейся группой)
- 2 НФ - Никакие неключевые атрибуты не являются функционально зависимыми лишь от части ключа (т.е. отношение всегда во 2НФ, если детерминант из одного атрибута)
- 3 НФ - Удаляются транзитивные зависимости неключевого атрибута (С) от ключевого (А) через другой неключевой (В):  
 $A \rightarrow B, B \rightarrow C \Rightarrow A \rightarrow C$
- НФБК - запрещаются транзитивные зависимости среди ключевых атрибутов (либо: Любой детерминант является ключом)
- ВННФ можно рассматривать как синоним понятия ациклической БД. Если схема отношений не имеет ВФЗ

# Описание Предметной области

- ФЗ - Пусть  $R$  — это отношение, а  $X$  и  $Y$  — произвольные подмножества множества атрибутов отношения  $R$ . Тогда  $Y$  функционально зависимо от  $X$  ( $X \rightarrow Y$ ), тогда и только тогда, когда каждое значение множества  $X$  отношения  $R$  связано в точности с одним значением множества  $Y$  отношения  $R$ .
- ВФЗ - Основным свойством ВФЗ является взаимная однозначность значений для атрибутов левой и правой части, т.е. каждый кортеж должен иметь уникальные значения полей, входящих во ВФЗ.
- УВФЗ - Условной взаимной функциональной зависимостью атрибутов  $A$  и  $B$  называется пара функциональных зависимостей вида  $CB \rightarrow A$ ,  $CA \rightarrow B$ , где  $C$  — набор атрибутов (условие) такой, что  $C \cap A = \emptyset$ ,  $C \cap B = \emptyset$ . Обозначается  $C | A \leftrightarrow B$ .

# Описание Предметной области

- Пример. Предположим следующую ситуацию, есть база данных о выписанных нарушениях в ГАИ. Задано множество ФЗ:

{Свид.о регистрации}->{ФИО}{ГосНомер}

в свидетельстве указывается фио владельца и госномер автомобиля. Данная информация заводится в ГАИ.

{№Квитанции}-> {ГосНомер}{ФИО}{Модель}{Время}

на посту при нарушении в каждой квитанции указывается фио нарушителя, госномер, модель автомобиля и время. Данная информация заводится на посту ДПС.

{ГосНомер}{ФИО}{Время}->{№Квитанции}

в один момент времени один человек на одной машине может нарушить всего 1 раз. Общее ограничение.

Минимальное покрытие для этого множества ФЗ будет таким же, это означает, что построенная на их основе БД будет находиться в ЗНФ

# Описание Предметной области

#№_Квитанции	Госномер	ФИО	Модель	Время
1101	O111ap150	Иванов	BMV320	12.30
1102	O112ap150	Петров	Opel Astra	12.30

#Свид.о регистрации	Госномер	ФИО
123456	O111ap150	Иванов
123457	O112ap150	Петров

#Госномер	#ФИО	#Время	№_Квитанции
123456	O111ap150	Иванов	1101
123457	O112ap150	Петров	1103

Допустим в результате работы была допущена ошибка, отмеченная красным. В результате получаем противоречивость хранимой информации. Используем программу для поиска взаимных функциональных зависимостей.

# Описание Предметной области

- В результате работы алгоритма поиска УФЗ получили следующий список ФЗ:

{Свид.о регистрации}->{ФИО}{ГосНомер}

{№Квитанции}-> {ГосНомер}{ФИО}{Модель}{Время}

{ФИО}{Время}|{ГосНомер}<->{№Квитанции}

Из него видно, что необходимо установить ограничение на уникальность комбинаций

{ГосНомер}{№Квитанции} при фиксированном значении условия {ФИО}{Время}. Таким образом поиск ВФЗ заранее в множестве заданных ФЗ может в дальнейшем предупредить противоречивость данных.

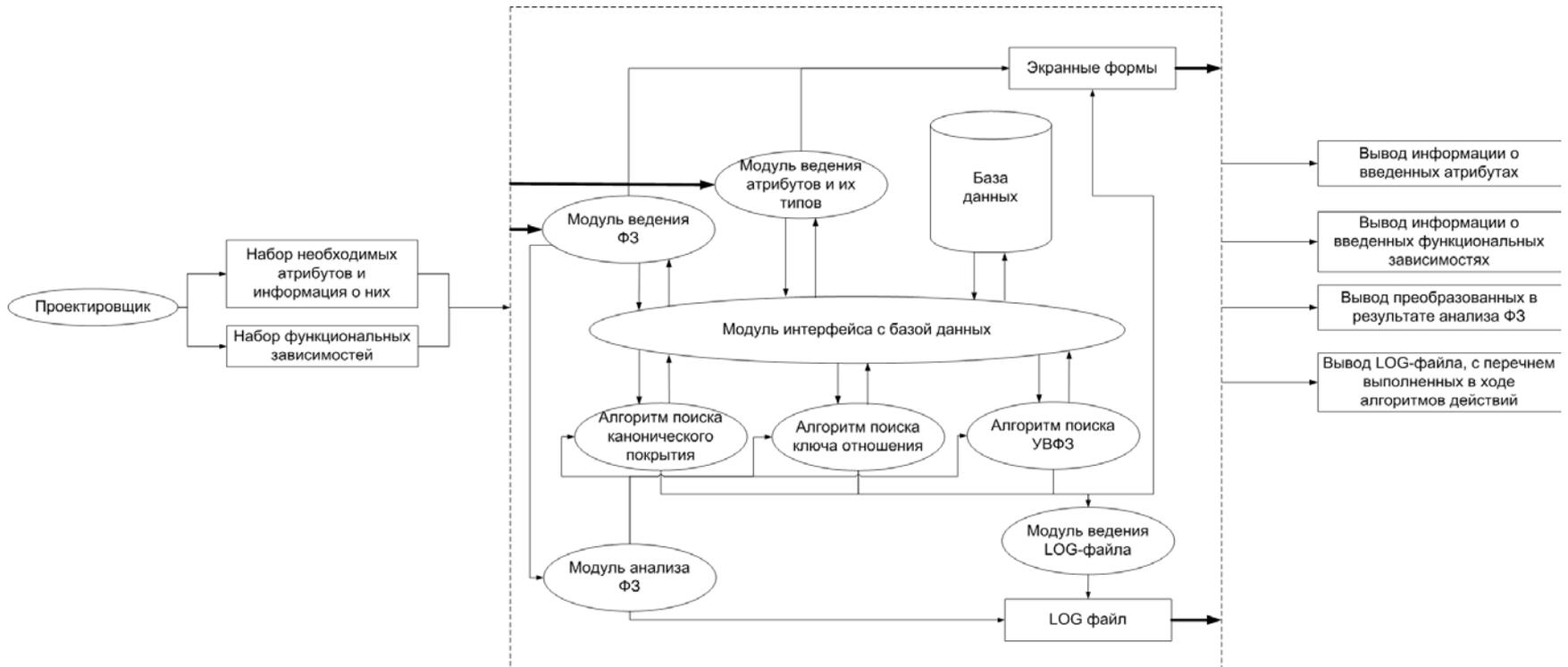
# Формализованное описание использованных алгоритмов

- Формализованное описание будет находиться на листах. В него войдут:
  - Описание понятий ФЗ, ВФЗ, УВФЗ на формализованном языке
  - Алгоритм нахождения канонического покрытия
  - Алгоритм поиска ключа отношения
  - Алгоритм поиска УВФЗ

# Структура Информационной системы

- Информационная система состоит из следующих модулей:
  - 1. Модуль ведения атрибутов и их типов
  - 2. Модуль ведения ФЗ
  - 3. Модуль анализа ФЗ
  - 4. Модуль ведения LOG-файла
  - 5. Модуль интерфейса с базой данных
- Схема взаимодействия модулей представлена на следующем слайде.

# Структура Информационной системы



# Технология работы с программой

- Технология работы с программой описана на листах. Работа подразумевает однопользовательский режим доступа, т.к. нет механизма блокировки данных. Данное допущение сделано т.к. работать с программой будет 1 человек, квалифицированный специалист, который «знает что елает».
- Можно выделить 3 этапа, через которые нужно пройти для обеспечения корректной работы системы:
  - 1. Ввести необходимые атрибуты с их типами
  - 2. Ввести множество заданных ФЗ-ей
  - 3. Перейти к анализу множества ФЗ-ей
  - 4. После анализа или во время можно открыть LOG-файл

В дальнейшем предполагается экспортировать результат анализа в виде DDL-сценария на языке SQL. Для этого необходимо добавить еще один модуль, который бы содержал справочник типов, редактор значений атрибутов, редактор имен таблиц, справочник конструкций для разных СУБД и т.д.

# Обзор аналогов и прототипов

- В качестве аналогов можно рассмотреть следующие системы:
- ERWin
  - ЗНФ достигается не с помощью алгоритмов, преобразующих начальную схему данных, а за счет указания связей:
    - При указании идентифицирующей связи ключ родительской сущности автоматически добавляется к ключевым атрибутам дочерней сущности (внешний ключ)
    - При указании неидентифицирующей связи ключ родительской сущности автоматически добавляется к неключевым атрибутам дочерней сущности
- Система нормализации БД (дипломная работа студента ИУ5)
  - Система используется алгоритм Ульмана для приведения схемы базы данных к ЗНФ. На вход подаются ФЗ в виде обозначений «А>В»

# ERWin

	ERWin	ИС нормализаци и БД
Работает на логическом уровне (можно задать логическую схему БД в виде ER-диаграмма)	+(+)	+(-)
Можно задать физическую схему БД	+	-
Использование 2ух типов связей: идентифицирующая и неидентифицирующая	+	-
Поиск УВФЗ	-	+
Приведение схемы БД к 3НФ	+	+
Категоризация – определить сущность как «супертип». Автоматически создаются нужные триггеры в DDL-сценарии	+	-
Ввод атрибутов на русском языке, а также ввод типов атрибутов	+	+
Создание DDL-сценария и представлений	+	-
Логирование алгоритмов преобразования и поиска	-	+

# Система нормализации БД (дипломная работа студента ИУ5)

	Система нормализации и БД	ИС нормализации и БД
Работает на логическом уровне	+	+
Приведение схемы БД к ЗНФ	+	+
Выполнение алгоритма приведения к ЗНФ по частям	+	-
Поиск УВФЗ	+	+
Ввод имен атрибутов на русском языке, а также ввод типов атрибутов	-	+
Логирование алгоритмов преобразования и поиска	-	+
Сохранение введенных данных об атрибутах и ФЗ	-	+

# Пример работы программы и экранные формы

- Запустить программу.

# Предметная область. Формализация

- В процессе разработки ИС было рассмотрено несколько алгоритмов нормализации:
  - Алгоритм Ульмана
  - Алгоритм Мейера

# Алгоритм Ульмана

- 1. *Получение неприводимого (нередуцируемого) справа покрытия ФЗ.* Для этого используется правило декомпозиции Армстронга. В результате каждая ФЗ имеет в правой части только один атрибут.
- $\forall (X \rightarrow A) (X \subseteq R, A \in R)$
- 2. *Получение избыточного покрытия.* ФЗ исключается из покрытия, если для нее соблюдается следующее условие:
- $(F - \{X \rightarrow A\})^+ = F^+ \Rightarrow F = F - \{X \rightarrow A\}$
- 3. *Получение неприводимого (нередуцируемого) слева покрытия ФЗ.* Для этого каждая ФЗ проверяется на возможность удаления атрибутов из левой части без изменения замыкания покрытия.
- $\forall (X \rightarrow A), \forall (B \subseteq X) A \in (X - B)^+ \Rightarrow (X - B) \rightarrow A$
- 4. *Декомпозиция отношения.* Для обеспечения свойства сохранения зависимостей отношение декомпозируется на отношения, содержащие охват (все атрибуты) каждой из ФЗ. Количество отношений равняется количеству ФЗ.
- Для обеспечения свойства соединения без потерь в схеме базы данных должно присутствовать отношение, содержащие ключ для всего отношения. Если оно отсутствует, то необходимо его добавить.
- Первые три пункта алгоритма требуются для нахождения минимального покрытия (в терминологии Ульмана), т.е. набор ФЗ должен быть избыточным, неприводимым (слева и справа) и содержать в правой части каждой ФЗ только один атрибут.

# Алгоритм Мейера

- Приведение к 3НФ производится посредством синтеза, а не декомпозиции.
- Алгоритм Синтеза:
- ВХОД: множество F-зависимостей
- ВЫХОД: полная схема базы данных для F
  1. Найти для F редуцированное минимальное кольцевое покрытие G
  2. Для каждой CF-зависимости  $(X_1^-, X_2^-, \dots X_k^-) \rightarrow Y$  из G построить схему отношения  $R = X_1 X_2 \dots X_k^- Y$  с выделенными ключами  $K = \{X_1 X_2 \dots X_k^-\}$
  3. Вернуться к множеству схем п.2