

МОДЕЛЬ БАЗЫ ДАННЫХ ДЛЯ МОДУЛЯ ИДЕНТИФИКАЦИИ ПОЛЬЗОВАТЕЛЕЙ В СИСТЕМЕ ТЕХНИЧЕСКОГО МЕНЕДЖМЕНТА СУДОВ

Использование информационных систем стало неотъемлемой составляющей деятельности судоходных компаний [1]. В базах информационных систем данные хранятся с определенной логической структурой – моделью представления данных (моделью данных), поддерживаемой СУБД. Одной из классических моделей является реляционная [2], в основе которой лежит понятие "отношение". Наглядная форма представления отношения – двумерная таблица, с помощью которой описывают данные и связи между ними. Поскольку в рамках одной таблицы не удастся описать сложные логические структуры данных из предметной области, применяют связывание таблиц. Достоинство реляционной модели данных – простота, наглядность и удобство реализации на ЭВМ.

При проектировании базы данных (БД) в общем случае задача формулируется следующим образом: выбрать подходящую логическую структуру для заданного массива данных, которые требуется поместить в базу.

Логическое проектирование заключается в определении числа и структуры таблиц, формировании запросов к БД, определении типов отчётных документов, разработке алгоритмов обработки информации, создании форм для ввода и редактирования данных в базе и решения других задач. Наиболее важной является проблема структуризации данных.

В большинстве случаев проектирование осуществляется независимо от приложения. Независимость от приложения желательна по той причине, что в момент проектирования БД обычно ещё неизвестны все возможные способы использования её данных. Но необходимо, чтобы созданный проект был стабильным, т.е. оставался работоспособным даже при возникновении в приложениях новых (т.е. неизвестных на момент создания исходного макета) требований к данным.

Эту задачу можно решить путём нормализации отношений, позволяющей исключить избыточность в отношениях и, как следствие, аномалии.

Основной задачей, решаемой в процессе проектирования БД, является задача нормализации её отношений. Метод нормальных форм

(НФ) является классическим методом проектирования реляционных баз данных.

Процесс проектирования БД с использованием метода НФ итерационный и заключается в последовательном переводе отношений из первой нормальной формы в нормальные формы более высокого порядка по определённым правилам. Каждая следующая нормальная форма ограничивает определённый тип функциональных зависимостей, устраняет соответствующие аномалии при выполнении операций над отношениями БД и сохраняет свойства предшествующих нормальных форм.

Отношение находится в первой нормальной форме, если все его атрибуты являются атомарными (имеют единственное значение).

Исходные отношения строятся таким образом, чтобы они были в 1НФ.

Процесс нормализации отношений методом НФ предполагает последовательное удаление из исходных отношений следующих межатрибутных зависимостей:

частичных зависимостей неключевых атрибутов от ключа (удовлетворение требований 2НФ);

транзитивных зависимостей неключевых атрибутов от ключа (удовлетворение требований 3НФ).

Частичная и транзитивная зависимости атрибутов от ключа приводят к явному и неявному избыточному дублированию данных в отношениях, следствием чего является проблема редактирования данных.

На практике построение 3НФ схем отношений в большинстве случаев является достаточным и приведением к ним процесс проектирования реляционной БД заканчивается [2].

Проектирование базы данных начинается с определения всех объектов, сведения о которых будут включены в базу, и определения их атрибутов.

Ниже предлагается фрагмент базы для регистрации пользователей и ограничения уровня доступа, который демонстрирует отношения, переведённые в 3НФ.

Отношения рассматриваемого фрагмента (рис. 1) базы данных имеют следующее назначение.

Переменная отношения *Access* задаёт уровень доступа к программе для каждой группы пользователей. У каждой группы пользователей есть: номер (*AccessId#*), который является уникальным; название группы (*Rights*); уровень доступа к определённым операциям в программе – добавление и редактирование компонента (*Comp*); удаление компонента (*CompDel*) и др.

```

VAR Access BASE RELATION
{ AccessId# NUMERIC,
Rights VARCHAR,
Comp TINYINT,
CompDel TINYINT,
...} ;
PRIMARY KEY { AccessId# };

VAR User BASE RELATION
{ UserId# NUMERIC,
Login VARCHAR,
Pass VARCHAR,
AccessId# NUMERIC,
...} ;
PRIMARY KEY { UserId # };
FOREIGN KEY { AccessId # } REFERENCES Access;

```

Рис. 1. Фрагмент базы данных (определение данных)

Переменная отношения *User* представляет пользователей, работающих с программой. Каждый пользователь имеет: уникальный номер (*UserId#*); учётное имя (*Login*), не обязательно уникальное (хотя оно может быть уникальным); пароль (*Pass*). Предполагается, что у пользователей с одинаковыми учётными именами должны быть различные пароли.

Отношение *User* в известном смысле служит для организации логической связи с переменной *Access*. Строка переменной отношения *User* связывает определенного пользователя с номером *UserId#* с соответствующим уровнем доступа, имеющим номер *AccessId#* в переменной отношения *Access*. Таким образом, каждый пользователь характеризуется уникальным номером, учётным именем, паролем и соответствующим уровнем доступа.

Пользователи и уровень доступа представляют собой основные сущности, о которых необходимо хранить информацию. В теории баз данных термин сущность обычно используется для обозначения любого различного объекта, который может быть представлен в базе данных (рис. 2). Уровень доступа к программе для каждой группы можно рассматривать как сущность, а пользователей – как связь между определенной группой прав и определенным пользователем. Однако связь можно считать особым видом сущности. Одно из преимуществ реляционных баз данных состоит именно в том, что все сущности, независимо от того, что на самом деле они могут являться связя-

ми, представляются одним универсальным способом, а именно - с помощью строк, объединенных в отношения.

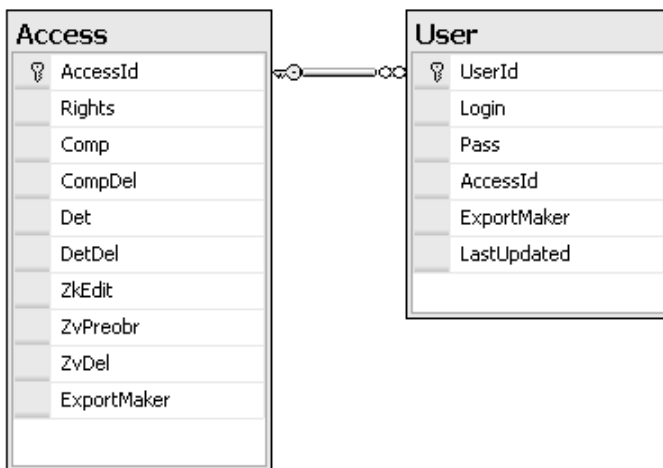


Рис. 2. Пример диаграммы для фрагмента базы данных

Модуль для регистрации пользователей и ограничения уровня доступа к функциям программы (рис. 3) системы технического менеджмента судов "Planner" [1] использует рассмотренные отношения и предусматривает различные права для различных групп пользователей.

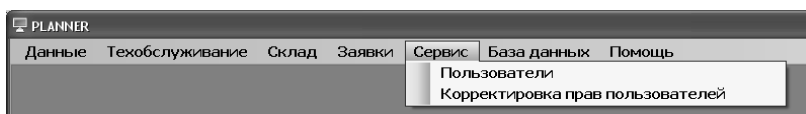


Рис. 3. Главное окно программы

На рис. 4 в окне "Корректировка прав пользователей", предназначенном для редактирования прав групп пользователей, в качестве примера заданы три группы пользователей, минимально достаточные для эксплуатации программы в судоходной компании:

Администратор;

Суперинтендант (в группу следует включить капитана, старшего механика и старшего помощника);

Судовой состав (в группу следует второго механика, электромеханика и других судовых специалистов, допущенных к работе с программой).

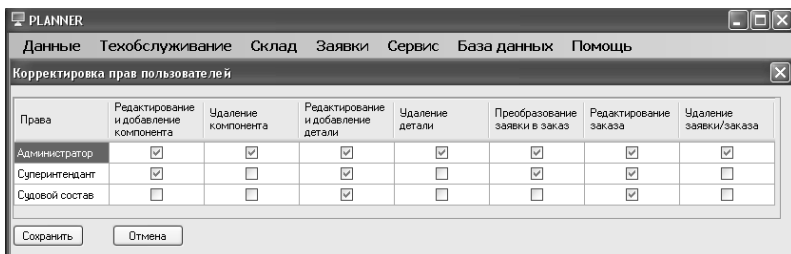


Рис. 4. Окно корректировки прав пользователей

Уровень доступа в программе задаётся для каждой группы пользователей. К группе может подключаться любое количество пользователей (в том числе и один). Пользователи одной группы имеют равный уровень доступа к функциям программы, но у каждого из них регистрируется своё учётное имя и свой пароль.

Кроме того, для запуска системы "Planner" каждый пользователь должен иметь свой пароль. Чтобы войти в программу необходимо ввести имя пользователя и пароль (рис. 5):

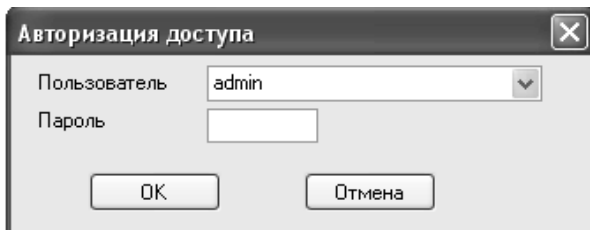


Рис. 5. Окно программы для авторизации доступа

Окно для регистрации пользователей служит для добавления/удаления пользователя с определенными правами и смены пароля (рис. 6).

Таким образом, в системе технического менеджмента судов реализована подсистема для регистрации пользователей и ограничения уровня доступа, позволяющая эффективно управлять правами пользователей.

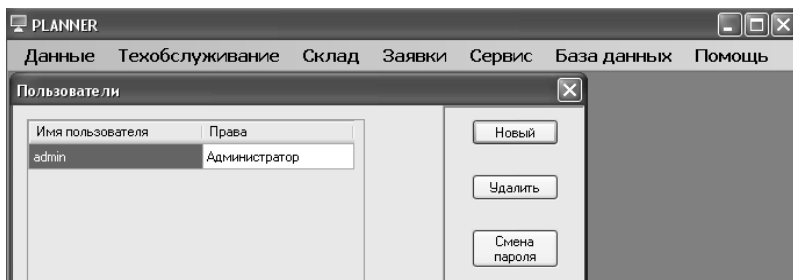


Рис.6. Окно регистрации пользователей

СПИСОК ЛИТЕРАТУРЫ

1. Горб С.И., Каменева А.В. Новая версия упрощенной информационной системы технического менеджмента // Автоматизация судовых технических средств: науч.-техн. сборник. – 2010. – Вып. 16. – Одесса: ОНМА. – С. 19 – 25.
2. Дейт К. Дж. Введение в системы баз данных. 8-е издание. – М.: Изд. дом "Вильямс", 2005. – 1328 с.