

гим» и «многие ко многим». Особого внимания требует связь «многие ко многим», такую связь необходимо нормализовать, поскольку она создает многозначную зависимость между сущностями. В таких случаях обычно создается третья промежуточная таблица, с которой легко работать. Можно заметить, что, используя промежуточные таблицы, общее количество таблиц в базе данных увеличивается и, соответственно, увеличивается объем занимаемой физической памяти.

Альтернативой ассоциативным таблицам являются битовые маски. Они применяются в случаях, когда значения одной из таблиц постоянны, например, виды разрешений для пользователей, жанры фильмов и другие. Числа в памяти компьютера хранятся в двоичной системе счисления в виде последовательности битов. Несложно провести аналогию между битами и значениями типа *bool*: значению бита 1 соответствует логическое *true*, а значению 0 – *false*. Число (последовательность битов) можно представить, как массив значений *bool*. При использовании битовых масок каждому биту ставится в соответствие определенное значение, массив этих значений образует число, которое хранится в соответствующем столбце. При работе с базой данных программист извлекает данное число из таблицы, переводит его в двоичную систему счисления, считывает биты и получает список значений.

Таким образом, битовые маски целесообразно использовать, когда имеются ограничения в объеме физической памяти и список значений, который записывается в битовую маску, фиксированный или же не подвергается частым изменениям, однако следует помнить, что хранимые в виде битовой маски числа требуют от программиста дополнительных действия по извлечению данных для дальнейшей работы.

**М. А. Рябиков, Л. К. Титова**  
(ГГТУ им. П. О. Сухого, Гомель)

## **ОБ ИСПОЛЬЗОВАНИИ НОВЫХ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ ПРИ СОЗДАНИИ WEB-ПРИЛОЖЕНИЙ**

Требование реализации проекта как Web-приложения приводит к необходимости анализа применения целого стека технологий, которые могут быть использованы для реализации программного комплекса.

*React.js* – это компонентная библиотека *JavaScript*, которая используется для создания интерактивного и динамического пользова-

тельского интерфейса для *web*-сайтов и мобильных приложений, особенно для разработки одностраничных приложений, в которых данные отображаются в режиме реального времени [1]. В настоящее время это одна из самых популярных интерфейсных библиотек *Javascript* на рынке. *React* представляется идеальным инструментом для создания масштабируемых *web*-приложений, особенно в тех ситуациях, когда приложение представляет *SPA*.

*Node.js* представляет среду выполнения кода на *JavaScript*, которая построена на основе движка *JavaScript Chrome V8*, который позволяет транслировать вызовы на языке *JavaScript* в машинный код. *Node.js* прежде всего предназначен для создания серверных приложений на языке *JavaScript*.

*Angular.js* представляет собой *opensource JavaScript*-фреймворк, использующий шаблон *MVC*. Для описания интерфейса используется декларативное программирование, а бизнес-аналитика отделена от кода интерфейса, что позволяет улучшить тестируемость и расширяемость приложений. Кроме того, *Angular.js* поддерживает такие функциональности, как *AJAX*, управление структурой *DOM*, анимация, шаблоны, маршрутизация и так далее.

Каждая из технологий имеет свои возможности и ограничения в индивидуальном порядке, что предоставляет разработчику широкий выбор инструментов разработки. Используя перечисленные технологии, повышается рейтинг веб-приложений и открываются новые возможности для привлечения новых клиентов.

### Литература

1 Бэнкс, А.А. *React и Redux. Функциональная веб-разработка* / А. А. Бэнкс, Е. К. Порселло. – М.; СПб. : Бином-Пресс ; Питер принт, 2019. – 329 с.

**Е. В. Синицин, Е. В. Комракова**  
(ГГТУ им. П. О. Сухого, Гомель)

### РОЛЬ МАТЕМАТИКИ В БАЛАНСЕ ИГР

Очень часто разработчики игр случайным образом балансируют игровые сущности. В большинстве случаев такой подход ведёт к неприятным проблемам на этапе контролирования и прогнозирования