# Select database

Contents:

\* [Status](#status)

\* [Summary](#summary)

\* [Context](#context)

\* [Options](#options)

\* [Decision](#decision)

\* [Consequences](#consequences)

##Status

Accepted.

##Summary

Для нашего приложения мы решили использовать sql базу данных mysql.

Так как мы поняли, что в нашем приложении предполагается большое количество связей между таблицами,

а на начальном этапе планирования мы точно не знаем какой формы у нас будут таблицы, поэтому структура БД каждый раз будет модифицироваться и видоизменяться.

##Context

##Options

Хорошо, потому что стандартность – использование языка SQL в программах стандартизировано международными организациями;

Хорошо, потому что есть независимость от конкретных СУБД – все распространенные СУБД используют SQL, т.к. реляционную базу данных можно перенести с одной СУБД на другую с минимальными доработками;

Хорошо, потому что табличная структура *реляционной БД* хорошо понятна, а потому язык SQL прост для изучения;

Хорошо, потому что есть возможность создания *интерактивных запросов* – SQL обеспечивает пользователям немедленный доступ к данным, при этом в интерактивном режиме можно получить результат *запроса* за очень короткое время без написания сложной программы;

Хорошо, потому что есть возможность программного доступа к БД – язык SQL легко использовать в приложениях, которым необходимо обращаться к *базам данных*. Одни и те же операторы SQL употребляются как для интерактивного, так и программного доступа, поэтому части программ, содержащие обращение к БД, можно вначале проверить в интерактивном режиме, а затем встраивать в программу;

Хорошо, потому что есть возможность обеспечения различного представления данных – с помощью SQL можно представить такую структуру данных, что тот или иной пользователь будет видеть различные их представления. Кроме того, данные из разных частей БД могут быть скомбинированы и представлены в виде одной простой *таблицы*, а значит, представления пригодны для усиления защиты БД и ее настройки под конкретные требования отдельных пользователей;

Хорошо, потому что есть поддержка архитектуры *клиент-сервер* – SQL – одно из лучших средств для реализации приложений на платформе *клиент-сервер*. SQL служит связующим звеном между взаимодействующей с пользователем клиентской системой и серверной системой, управляющей БД, позволяя каждой из них сосредоточиться на выполнении своих функций.

Плохо, потому что сложен в понимании, поэтому является только инструментом программиста.

Плохо, потому что поддерживает работу только с РБД.

Плохо, потому что предназначен только для формирования запросов к БД.

##Decision

На начальном этапе планирования мы не знаем точную структуру БД, поэтому она постоянно будет меняться.

##Consequences

пока не столкнулись с какой-либо проблемой в использовании.

Все реальные модели легко спроектировать в виде таблиц в SQL.

Нет какой-либо избыточности и дублирования благодаря тому, что используем 3-ю нормальную форму.

Deciders: Исламов Яхья, Лежнина Анна, Якушев Павел, Москвичев Михаил

Date: 02.03.2020