**Анализ существующих решений**

Существует большое количество новостных порталов, написанных на языке программирования PHP. Они очень сильно отличаются друг от друга и существует много способов по их реализации. На примере мы рассмотрим такие порталы как [www.ukr.net](http://www.ukr.net) , [www.mail.ru](http://www.mail.ru) , [www.korrespondent.net](http://www.korrespondent.net) , [www.siteua.org](http://www.siteua.org).

Новостной портал [www.korrespondent.net](http://www.korrespondent.net) :

Положительные стороны:

1. Сайт представлен на двух языках – русский и украинский что расширяет его аудиторию.
2. Присутствует раздел видео
3. Присутствует разделение содержимого портала на определённые тематики.
4. На портале присутствует баннер погоды и курсы валют.
5. Присутствуют блоги
6. Малое время выполнения запросов.

Отрицательные стороны:

1. Нет своего почтового сервиса.
2. Много рекламы.
3. Нет фильтрации нецензурных выражений в комментариях.
4. Неудобные комментарии.

Новостной портал [www.ukr.net](http://www.ukr.net):

Положительные стороны:

1. Удобный интерфейс
2. Рекламы хотя достаточно много, но она расположена удобно для пользователей и не мешает работе.
3. Присутствует раздел видео.
4. Присутствует свой почтовый сервис.
5. Удобный поиск по новостям
6. Присутствуют разделы новостей по определённым критериям.
7. Присутствует погода

Отрицательные стороны:

1. Отсутствуют комментарии к новостям.
2. Отсутствие форума.

Новостной портал [www.mail.ru](http://www.mail.ru):

Положительные стороны:

1. Присутствие блогов.
2. Присутствие видео.
3. Присутствие сервиса ответы, где пользователи могу получить ответ на свой вопрос.
4. Присутствие сервиса погода.
5. Присутствие чата.

Отрицательные стороны:

1. Достаточно много рекламы.
2. Новости занимают очень мало места на сайте и неудобно расположены.
3. Довольно медленно грузится.

Новостной портал [www.siteua.org](http://www.siteua.org) :

Положительные стороны:

1. Присутствие рейтинга новостей.
2. Разделение новостей на разделы.
3. Присутствие сервиса погоды и курса валют.

Отрицательные стороны:

1. Очень много рекламы.
2. Неудобный интерфейс.
3. Неудобный поиск по новостям.
4. Неудобная форма входа на сайт.

В своей программе я использую языки PHP и MySQL как базу данных.

Положительные стороны моего портала:

1. Удобный интерфейс.
2. Отсутствие рекламы.
3. Присутствие форума.
4. Присутствие комментариев.
5. Удобный просмотр новостей.
6. Присутствие администраторской части, что облегчит жизнь заказчику. Он сможет без помощи программиста управлять сайтом.

Отрицательные стороны:

1. Не очень удобный поиск.
2. Не очень красивый дизайн.

**Выбор конкретной СУБД**

**Основная информация о MySQL:**

MySQL - это популярная система управления базами данных (СУБД), очень часто применяемая в сочетании с PHP.

База данных представляет собой структурированную совокупность данных. Эти данные могут быть любыми - от простого списка предстоящих покупок до перечня экспонатов картинной галереи или огромного количества информации в корпоративной сети. Для записи, выборки и обработки данных, хранящихся в компьютерной базе данных, необходима система управления базой данных, каковой и является ПО MySQL. Поскольку компьютеры замечательно справляются с обработкой больших объемов данных, управление базами данных играет центральную роль в вычислениях. Реализовано такое управление может быть по-разному - как в виде отдельных утилит, так и в виде кода, входящего в состав других приложений.

MySQL - это система управления реляционными базами данных. В реляционной базе данных данные хранятся не все скопом, а в отдельных таблицах, благодаря чему достигается выигрыш в скорости и гибкости. Таблицы связываются между собой при помощи отношений, благодаря чему обеспечивается возможность объединять при выполнении запроса данные из нескольких таблиц. SQL как часть системы MySQL можно охарактеризовать как язык структурированных запросов плюс наиболее распространенный стандартный язык, используемый для доступа к базам данных.

MySQL - это ПО с открытым кодом. Применять его и модифицировать может любой желающий. Такое ПО можно получать по Internet и использовать бесплатно. При этом каждый пользователь может изучить исходный код и изменить его в соответствии со своими потребностями. Использование программного обеспечения MySQL регламентируется лицензией GPL (GNU General Public License), http://www.gnu.org/licenses/, в которой указано, что можно и чего нельзя делать с этим программным обеспечением в различных ситуациях.

Почему веб-программисты отдают предпочтение СУБД MySQL? MySQL является очень быстрым, надежным и легким в использовании. Если вам требуются именно эти качества, попробуйте поработать с данным сервером. MySQL обладает также рядом удобных возможностей, разработанных в тесном контакте с пользователями. Первоначально сервер MySQL разрабатывался для управления большими базами данных с целью обеспечить более высокую скорость работы по сравнению с существующими на тот момент аналогами. И вот уже в течение нескольких лет данный сервер успешно используется в условиях промышленной эксплуатации с высокими требованиями. Несмотря на то что MySQL постоянно совершенствуется, он уже сегодня обеспечивает широкий спектр полезных функций. Благодаря своей доступности, скорости и безопасности MySQL очень хорошо подходит для доступа к базам данных по Internet.

Технические возможности СУБД MySQL MySQL является системой клиент-сервер, которая содержит многопоточный SQL-сервер, обеспечивающий поддержку различных вычислительных машин баз данных, а также несколько различных клиентских программ и библиотек, средства администрирования и широкий спектр программных интерфейсов (API). Доступно также большое количество программного обеспечения для MySQL, в большей части - бесплатного.

**Устройство MySQL**

MySQL состоит из двух частей: серверной и клиентской.

Сервер MySQL постоянно работает на компьютере. Клиентские программы (например, скрипты PHP) посылают серверу MySQL SQL-запросы через механизм сокетов (то есть при помощи сетевых средств), сервер их обрабатывает и запоминает результат. То есть скрипт (клиент) указывает, какую информацио он хочет получить от сервера баз данных. Затем сервер баз данных посылает ответ (результат) клиенту (скрипту).

Почему всегда передается не весь результат? Очень просто: дело в том, что размер результирующего набора данных может быть слишком большим, и на его передачу по сети уйдет чересчур много времени. Да и редко когда бывает нужно получать сразу весь вывод запроса (то есть все записи, удовлетворяющие выражению запроса). Механизм использования сокетов подразумевает технологию клиент-сервер, а это означает, что в системе должна быть запущена специальная программа — MySQL-сервер, которая принимает и обрабатывает запросы от программ. Так как вся работа происходит в действительности на одной машине, накладные расходы по работе с сетевыми средствами незначительны (установка и поддержание соединения с MySQL-сервером обходится довольно дешево).

Cтруктура MySQL трехуровневая: базы данных — таблицы — записи. Базы данных и таблицы MySQL физически представляются файлами с расширениями frm, MYD, MYI. Логически - таблица представляет собой совокупность записей. А записи - это совокупность полей разного типа. Имя базы данных MySQL уникально в пределах системы, а таблицы - в пределах базы данных, поля - в пределах таблицы. Один сервер MySQL может поддерживать сразу несколько баз данных, доступ к которым может разграничиваться логином и паролем. Зная эти логин и пароль, можно работать с конкретной базой данных. Например, можно создать или удалить в ней таблицу, добавить записи и т. д. Обычно имя-идентификатор и пароль назначаются хостинг провайдерами, которые и обеспечивают поддержку MySQL для своих пользователей.

**Сравнение возможностей MySQL и PostgreSQL:**

MySQL обладает следующими преимуществами перед PostgreSQL:

MySQL обычно намного превосходит PostgreSQL по скорости работы. Кроме того, в MySQL 4.0 реализован кэш запросов. Он позволяет во много раз увеличить скорость обработки запросов для сайтов, на которых преобладают неоднократно повторяющиеся запросы на чтение.

По количеству пользователей MySQL также намного превосходит PostgreSQL. Поэтому код тестируется значительно более придирчиво и опытным путем доказана большая его надежность, нежели у PostgreSQL. MySQL чаще, чем PostgreSQL, используется на производстве, в основном потому, что компания MySQL AB (ранее - TCX DataKonsult AB) предоставляет высококачественную коммерческую техническую поддержку MySQL с момента появления этой системы на рынке, а у PostgreSQL до самого последнего времени никакой поддержки не было.

MySQL работает в среде Windows лучше, чем PostgreSQL. MySQL Server запускается как настоящее (родное) Windows-приложение (в NT/2000/XP - сервис), в то время как PostgreSQL запускается в среде эмуляции, Cygwin.

MySQL оснащен большим количеством API для других языков и поддерживается большим количеством существующих программ, нежели PostgreSQL.

MySQL работает на высоконадежных промышленных системах 24/7 (включенных 24 часа в сутки 7 дней в неделю). В большинстве случаев никаких ``чисток'' в MySQL производить не требуется. PostgreSQL же пока что не может работать в таких системах, так как иногда приходится запускать VACUUM для освобождения занятого последствиями работы команд UPDATE и DELETE пространства и проводить статистический анализ, необходимый для достижения максимальной производительности PostgreSQL. Запускать VACUUM необходимо и после каждого добавления к таблице нескольких столбцов. На напряженно работающих системах VACUUM нужно запускать более часто, в худших случаях - по несколько раз в день. А ведь во время работы VACUUM (а ее работа может продолжаться часы, если база данных достаточно велика) база практически ``мертва''. Впрочем, в PostgreSQL версии 7.2 выполнение основных функций этой программы больше не приводит к блокировке базы, и пользователи могут продолжать нормально работать с ней. Новая команда VACUUM FULL берется за дело более серьезно: она, как и в старых версиях, блокирует таблицу и сжимает копию таблицы на диске.

В комплект поставки MySQL входят два тестовых пакета, mysql-test-run и , а также пакет для замеров производительности. Тестовая система постоянно обновляется, в нее добавляется код для тестирования всех новых возможностей и почти всех воспроизводимых ошибок, которые попали в поле нашего зрения. Перед выпуском каждой новой версии мы используем эти пакеты для тестирования MySQL на нескольких платформах. Наши тесты значительно превосходят по своим возможностям все существующие в PostgreSQL аналоги, и обеспечивают высокое качество кода MySQL.

Книг о MySQL вышло значительно больше, нежели о PostgreSQL. Книги о MySQL выпустили издательства O'Reilly, SAMS, Que и New Riders. Все возможности MySQL детально описаны в документации, так как это является обязательным условием включения новых возможностей в код.

MySQL поддерживает больше стандартных функций ODBC, чем PostgreSQL.

MySQL обладает значительно более мощной реализацией ALTER TABLE.

В MySQL предусмотрена возможность создания таблиц без транзакций, что необходимо приложениям, требующим максимально возможной скорости работы. Эти таблицы могут храниться в памяти, относиться к типу HEAP-таблиц или дисковых MyISAM. See section 7 Типы таблиц MySQL.

MySQL может работать с двумя поддерживающими транзакции обработчиками таблиц, а именно - InnoDB и BerkeleyDB. Так как все системы поддержки транзакций в разных условиях работают по-разному, это дает разработчику возможность найти наилучшее решение для условий, в которых будет работать его система. See section 7 Типы таблиц MySQL.

Команда слияния таблиц MERGE предоставляет в ваше распоряжение уникальную возможность создать представление нескольких идентичных таблиц и работать с ними как с одной. Это особенно удобно для работы с журналами, разбитыми, например, по месяцам.

Возможность сжатия доступных только для чтения таблиц, не отменяющая прямого доступа к их записям, повышает производительность системы, снижая количество операций считывания с диска. Это особенно полезно при архивировании. В MySQL реализован полнотекстовый поиск. See section 6.8 Полнотекстовый поиск в MySQL.

Имеется возможность работы с несколькими базами через одно соединение (разумеется, в зависимости от привилегий пользователя).

Система MySQL с самого начала разрабатывалась в расчете на многопоточность, а PostgreSQL использует процессы. Переключение контекстов и доступ к общим данным несколькими потоками осуществляется значительно быстрее, нежели отдельными процессами. Таким образом MySQL Server в многопользовательских приложениях получает неплохое преимущество в производительности, а кроме того, таким образом MySQL Server удается значительно эффективней пользоваться преимуществами, предоставляемыми симметричными мультипроцессорными системами (SMP).

В MySQL реализована значительно более мощная система привилегий, нежели в PostgreSQL. В то время как PostgreSQL обеспечивает лишь привилегии INSERT, SELECT и UPDATE/DELETE над базой или таблицей, MySQL предоставляет возможность определения полного набора разнообразных привилегий на уровне базы, таблицы и столбца. Кроме того, MySQL позволяет задавать привилегии для комбинаций хост/пользователь..

В MySQL используется протокол связи между клиентом и сервером со сжатием данных, что увеличивает производительность системы в условиях низкоскоростных каналов связи.

Насколько нам известно, только в реляционной системе баз данных MySQL Server используется концепция ``обработчика таблиц''. Благодаря этому создается возможность работы с различными низкоуровневыми типами таблиц из ядра MySQL, причем каждая таблица может быть оптимизирована для различных характеристик производительности.

Все типы таблиц в MySQL (кроме InnoDB) реализованы в виде файлов (по одной таблице в файле), что значительно облегчает создание резервных копий, перенос, удаление и даже создание символьных ссылок между базами данных и таблицами, даже если сервер находится в нерабочем состоянии.

Наличие утилиты для восстановления и оптимизации таблиц MyISAM (наиболее распространенного типа таблиц в MySQL). Ее использование требуется только в случае физического повреждения файла данных (например, в результате аппаратного сбоя). Позволяет восстановить большую часть данных.

Обновление (апгрейд) MySQL проходит совершенно ``безболезненно''. При модернизации MySQL нет нужды в копировании/восстановлении данных, что приходится делать при установке большинства обновлений PostgreSQL.

**Недостатки MySQL по сравнению с PostgreSQL:**

Поддержка транзакций в MySQL пока что не настолько хорошо проверена, как в системе PostgreSQL.

Блокировка таблиц, применяющаяся в нетранзакционных таблицах MyISAM, во многих случаях работает быстрее, нежели блокировки на уровне страниц, строк или контроль версий. Недостаток этого подхода в том, что если не учитывать механизм работы блокирования таблиц, один длительный запрос может надолго заблокировать таблицу. Обычно этого эффекта можно избежать, приняв соответствующие меры при разработке приложения. Если это не удастся, всегда можно изменить тип таблицы и сделать ее транзакционной.

При помощи UDF (user-defined functions, определяемые пользователем функции) возможности MySQL можно расширить и дополнить обычными SQL-функциями или их объединениями. Но это сделать не так просто, да и система не настолько гибка в этом отношении, как PostgreSQL. See section 9.2 Добавление новых функций в MySQL.

В MySQL сложнее организовывались обновления, затрагивающие несколько таблиц сразу. Впрочем, это было исправлено в MySQL 4.0.2 реализацией многотабличного UPDATE и в MySQL 4.1 - с помощью подзапросов. В MySQL 4.0 можно одновременно удалять данные из нескольких таблиц.

**Выбор языка программирования**

**Преимущества языка PHP:**

Главным фактором языка РНР является практичность. РНР должен предоставить программисту средства для быстрого и эффективного решения поставленных задач. Практический характер РНР обусловлен пятью важными характеристиками:

традиционностью;

простотой;

эффективностью;

безопасностью;

гибкостью.

Существует еще одна «характеристика», которая делает РНР особенно привлекательным: он распространяется бесплатно! Причем, с открытыми исходными кодами ( Open Source ).

**Традиционность**

Язык РНР будет казаться знакомым программистам, работающим в разных областях. Многие конструкции языка позаимствованы из Си, Perl.

Код РНР очень похож на тот, который встречается в типичных программах на С или Pascal. Это заметно снижает начальные усилия при изучении РНР. PHP — язык, сочетающий достоинства Perl и Си и специально нацеленный на работу в Интернете, язык с универсальным (правда, за некоторыми оговорками) и ясным синтаксисом.

И хотя PHP является довольно молодым языком, он обрел такую популярность среди web-программистов, что на данный момент является чуть ли не самым популярным языком для создания web-приложений (скриптов).

**Простота**

Сценарий РНР может состоять из 10 000 строк или из одной строки — все зависит от специфики вашей задачи. Механизм РНР просто начинает выполнять код после первой экранирующей последовательности (<?) и продолжает выполнение до того момента, когда он встретит парную экранирующую последовательность (?>). Если код имеет правильный синтаксис, он исполняется в точности так, как указал программист.

PHP — язык, который может быть встроен непосредственно в html -код страниц, которые, в свою очередь будут корректно обрабатываться PHP -интерпретатором.

**Эффективность**

Эффективность является исключительно важным фактором при программировании для многопользовательских сред, к числу которых относится и web .

Очень важное преимущество PHP заключается в его «движке». «Движок» PHP не является ни компилятором, ни интерпретатором. Он является транслирующим интерпретатором. Такое устройство «движка» PHP позволяет обрабатывать сценарии с достаточно высокой скоростью.

По некоторым оценкам, большинство PHP-сценариев (особенно не очень больших размеров) обрабатываются быстрее аналогичных им программ, написанных на Perl. Однако, чтобы не делали разработчики PHP, откомпилированные исполняемые файлы будут работать значительно быстрее – в десятки, а иногда и в сотни раз. Но производительность PHP вполне достаточна для создания вполне серьезных web-приложений. Подробно об устройстве и характеристиках «движка» PHP можно ознакомиться здесь.

**Безопасность**

РНР предоставляет в распоряжение разработчиков и администраторов гибкие и эффективные средства безопасности, которые условно делятся на две категории: средства системного уровня и средства уровня приложения.

1. Средства безопасности системного уровня

В РНР реализованы механизмы безопасности, находящиеся под управлением администраторов; при правильной настройке РНР это обеспечивает максимальную свободу действий и безопасность. РНР может работать в так называемом безопасном режиме (safe mode), который ограничивает возможности применения РНР пользователями по ряду важных показателей. Например, можно ограничить максимальное время выполнения и использование памяти (неконтролируемый расход памяти отрицательно влияет на быстродействие сервера). По аналогии с cgi-bin администратор также может устанавливать ограничения на каталоги, в которых пользователь может просматривать и исполнять сценарии РНР, а также использовать сценарии РНР для просмотра конфиденциальной информации на сервере (например, файла passwd).

2. Средства безопасности уровня приложения

В стандартный набор функций РНР входит ряд надежных механизмов шифрования. РНР также совместим с многими приложениями независимых фирм, что позволяет легко интегрировать его с защищенными технологиями электронной коммерции (e-commerce). Другое преимущество заключается в том, что исходный текст сценариев РНР нельзя просмотреть в браузере, поскольку сценарий компилируется до его отправки по запросу пользователя. Реализация РНР на стороне сервера предотвращает похищение нетривиальных сценариев пользователями, знаний которых хватает хотя бы для выполнения команды View Source.

**Гибкость**

Поскольку РНР является встраиваемым (embedded) языком, он отличается исключительной гибкостью по отношению к потребностям разработчика. Хотя РНР обычно рекомендуется использовать в сочетании с HTML, он с таким же успехом интегрируется и в JavaScript, XML и другие языки. Кроме того, хорошо структурированные приложения РНР легко расширяются по мере необходимости (впрочем, это относится ко всем основным языкам программирования).

Нет проблем и с зависимостью от браузеров, поскольку перед отправкой клиенту сценарии РНР полностью компилируются на стороне сервера. В сущности, сценарии РНР могут передаваться любым устройствам с браузерами, включая сотовые телефоны, электронные записные книжки, пейджеры и портативные компьютеры, не говоря уже о традиционных ПК. Программисты, занимающиеся вспомогательными утилитами, могут запускать РНР в режиме командной строки.

Поскольку РНР не содержит кода, ориентированного на конкретный web-сервер, пользователи не ограничиваются определенными серверами (возможно, незнакомыми для них). Apache, Microsoft IIS, Netscape Enterprise Server, Stronghold и Zeus — РНР работает на всех перечисленных серверах. Поскольку эти серверы работают на разных платформах, РНР в целом является платформенно-независимым языком и существует на таких платформах, как UNIX, Solaris, FreeBSD и Windows 95/98/NT/2000/XP/2003.

Наконец, средства РНР позволяют программисту работать с внешними компонентами, такими как Enterprise Java Beans или СОМ-объекты Win32. Благодаря этим новым возможностям РНР занимает достойное место среди современных технологий и обеспечивает масштабирование проектов до необходимых пределов.

**Бесплатное распространение**

Стратегия Open Source, и распространение исходных текстов программ в массах, оказало несомненно благотворное влияние на многие проекты, в первую очередь — Linux, хотя и успех проекта Apache сильно подкрепил позиции сторонников Open Source. Сказанное относится и к истории создания РНР, поскольку поддержка пользователей со всего мира оказалась очень важным фактором в развитии проекта РНР.

Принятие стратегии Open Source и бесплатное распространение исходных текстов РНР оказало неоценимую услугу пользователям. Вдобавок, отзывчивое сообщество пользователей РНР является своего рода «коллективной службой поддержки», и в популярных электронных конференциях можно найти ответы даже на самые сложные вопросы.