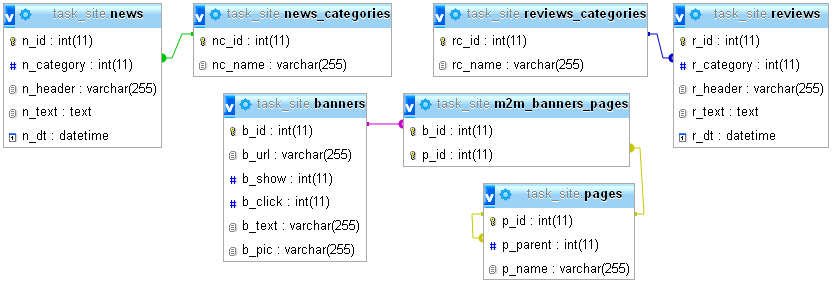
|  |
| --- |
| , RD Dep. |
| Практическое задание  CTECH.DB.06 Язык управления данными в MySQL |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| <1.0> | Первая версия | Святослав Куликов | <12.12.2012> |  |  |

**Задание № 1 («Сайт»).**

Схема базы данных:



Скрипт генерации базы данных (полезен для лучшего понимания структуры):

|  |  |  |
| --- | --- | --- |
| Таблица `banners` | Таблица `m2m\_banners\_pages` | Таблица `news` |
| CREATE TABLE IF NOT EXISTS `banners` (  `b\_id` int(11) NOT NULL  AUTO\_INCREMENT,  `b\_url` varchar(255) NOT NULL,  `b\_show` int(11) NOT NULL DEFAULT '0',  `b\_click` int(11) NOT NULL DEFAULT '0',  `b\_text` varchar(255) DEFAULT NULL,  `b\_pic` varchar(255) DEFAULT NULL,  PRIMARY KEY (`b\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=1 ; | CREATE TABLE IF NOT EXISTS `m2m\_banners\_pages` (  `b\_id` int(11) NOT NULL,  `p\_id` int(11) NOT NULL,  PRIMARY KEY (`b\_id`,`p\_id`),  KEY `p\_id` (`p\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8; | CREATE TABLE IF NOT EXISTS `news` (  `n\_id` int(11) NOT NULL  AUTO\_INCREMENT,  `n\_category` int(11) NOT NULL,  `n\_header` varchar(255) NOT NULL,  `n\_text` text NOT NULL,  `n\_dt` datetime NOT NULL,  PRIMARY KEY (`n\_id`),  KEY `n\_category` (`n\_category`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=1 ; |
| Таблица `news\_categories` | Таблица `pages` | Таблица `reviews` |
| CREATE TABLE IF NOT EXISTS `news\_categories` (  `nc\_id` int(11) NOT NULL  AUTO\_INCREMENT,  `nc\_name` varchar(255) NOT NULL,  PRIMARY KEY (`nc\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=1 ; | CREATE TABLE IF NOT EXISTS `pages` (  `p\_id` int(11) NOT NULL  AUTO\_INCREMENT,  `p\_parent` int(11) DEFAULT NULL,  `p\_name` varchar(255) NOT NULL,  PRIMARY KEY (`p\_id`),  KEY `p\_parent` (`p\_parent`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=1 ; | CREATE TABLE IF NOT EXISTS `reviews` (  `r\_id` int(11) NOT NULL  AUTO\_INCREMENT,  `r\_category` int(11) NOT NULL,  `r\_header` varchar(255) NOT NULL,  `r\_text` text NOT NULL,  `r\_dt` datetime NOT NULL,  PRIMARY KEY (`r\_id`),  KEY `r\_category` (`r\_category`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=1 ; |
| Таблица `reviews\_categories` | Связи | |
| CREATE TABLE IF NOT EXISTS `reviews\_categories` (  `rc\_id` int(11) NOT NULL  AUTO\_INCREMENT,  `rc\_name` varchar(255) NOT NULL,  PRIMARY KEY (`rc\_id`)  ) ENGINE=InnoDB DEFAULT CHARSET=utf8 AUTO\_INCREMENT=1 ; | ALTER TABLE `m2m\_banners\_pages`  ADD CONSTRAINT `m2m\_banners\_pages\_ibfk\_2` FOREIGN KEY (`p\_id`) REFERENCES `pages` (`p\_id`) ON DELETE CASCADE ON UPDATE CASCADE,  ADD CONSTRAINT `m2m\_banners\_pages\_ibfk\_1` FOREIGN KEY (`b\_id`) REFERENCES `banners` (`b\_id`) ON DELETE CASCADE ON UPDATE CASCADE;  ALTER TABLE `news`  ADD CONSTRAINT `news\_ibfk\_1` FOREIGN KEY (`n\_category`) REFERENCES `news\_categories` (`nc\_id`) ON DELETE CASCADE ON UPDATE CASCADE;  ALTER TABLE `pages`  ADD CONSTRAINT `pages\_ibfk\_1` FOREIGN KEY (`p\_parent`) REFERENCES `pages` (`p\_id`) ON DELETE CASCADE ON UPDATE CASCADE;  ALTER TABLE `reviews`  ADD CONSTRAINT `reviews\_ibfk\_1` FOREIGN KEY (`r\_category`) REFERENCES `reviews\_categories` (`rc\_id`) ON DELETE CASCADE ON UPDATE CASCADE; | |

Наполнение таблиц данными:

`pages`

|  |  |  |
| --- | --- | --- |
| **p\_id** | **p\_parent** | **p\_name** |
| 1 | *NULL* | Юридическим лицам |
| 2 | *NULL* | Физическим лицам |
| 3 | 1 | Образцы договоров |
| 4 | 1 | Банковские реквизиты |
| 5 | 2 | Схема проезда к офису |
| 6 | 2 | Почта и телефон |
| 7 | 3 | Договоры оптовых закупок |

`news\_categories`

|  |  |
| --- | --- |
| **nc\_id** | **nc\_name** |
| 1 | Финансы |
| 2 | Законодательство |
| 3 | Логистика |
| 4 | Строительство |

`reviews\_categories`

|  |  |
| --- | --- |
| **rc\_id** | **rc\_name** |
| 1 | Технологии |
| 2 | Товары и услуги |

`m2m\_banners\_pages`

|  |  |
| --- | --- |
| **b\_id** | **p\_id** |
| 1 | 1 |
| 2 | 1 |
| 7 | 1 |
| 1 | 2 |
| 4 | 3 |
| 1 | 4 |
| 2 | 4 |
| 1 | 5 |
| 2 | 5 |
| 3 | 5 |

`news`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **n\_id** | **n\_category** | **n\_header** | **n\_text** | **n\_dt** |
| 1 | 1 | Состояние валютного рынка | <to be added> | 2012-12-03 04:15:27 |
| 2 | 3 | Контрабанда железобетонных плит | <to be added> | 2011-09-14 06:19:08 |
| 3 | 3 | Почта России: вчера, сегодня и снова вчера |  | 2011-08-17 09:06:30 |
| 4 | 3 | Самолётом или поездом? | <to be added> | 2012-12-20 06:11:42 |
| 5 | 3 | Куда всё катится? | <to be added> | 2012-12-11 04:36:17 |

`reviews`

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **r\_id** | **r\_category** | **r\_header** | **r\_text** | **r\_dt** |
| 1 | 1 | Роботы на страже строек | <empty> | 2011-10-03 05:17:37 |
| 2 | 1 | Когда всё это кончится?! | Никогда! | 2012-12-12 06:31:13 |

`banners`

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **b\_id** | **b\_url** | **b\_show** | **b\_click** | **b\_text** | **b\_pic** |
| 1 | http://tut.by | 200 | 20 | TUT.BY | *NULL* |
| 2 | http://tut.by | 200 | 300 | *NULL* | tutby.png |
| 3 | http://onliner.by | 50 | 45 | ONLINER.BY | *NULL* |
| 4 | http://onliner.by | 50 | 1 | *NULL* | onlinerby.jpg |
| 5 | http://google.by | 10 | 10 | GOOGLE.BY | googleby.png |
| 6 | http://google.com | 1 | 1 | *NULL* | googlecom.png |
| 7 | http://habrahabr.ru | 999 | 997 | *NULL* | habrahabrru.png |
| 8 | http://habrahabr.ru | 50 | 49 | HABRAHABR.RU | *NULL* |
| 9 | http://gismeteo.by | 0 | 0 | Погода | *NULL* |
| 10 | http://gismeteo.ru | 0 | 0 | Погода | *NULL* |

Задание:

1. Написать запрос, считающий суммарное количество имеющихся на сайте новостей и обзоров.

В результате выполнения запроса должно получиться:

|  |
| --- |
| **SUM** |
| 7 |

======================

select sum(ids) as SUM from(select count(r\_id) as ids from reviews

union all select count(n\_id) as ids from news) as Sum2;

======================

1. Написать запрос, показывающий список категорий новостей и количество новостей в каждой категории.

В результате выполнения запроса должно получиться:

|  |  |
| --- | --- |
| **nc\_name** | **COUNT(`n\_id`)** |
| Финансы | 1 |
| Законодательство | 0 |
| Логистика | 4 |
| Строительство | 0 |

======================

select nc\_name, count1 from news\_categories left join (select n\_category, count(\*) as count1 from news group by n\_category) AS Temp on news\_categories.nc\_id=Temp.n\_category;

======================

1. Написать запрос, показывающий список категорий обзоров и количество обзоров в каждой категории.

В результате выполнения запроса должно получиться:

|  |  |
| --- | --- |
| **rc\_name** | **COUNT(`r\_id`)** |
| Технологии | 2 |
| Товары и услуги | 0 |

======================

select rc\_name, count2 from reviews\_categories left join (select r\_category, count(\*) as count2 from reviews group by r\_category) AS Temp2 on reviews\_categories.rc\_id=Temp2.r\_category;

======================

1. Написать запрос, показывающий список категорий новостей, категорий обзоров и дату самой свежей публикации в каждой категории.

В результате выполнения запроса должно получиться:

|  |  |
| --- | --- |
| **category\_name** | **last\_date** |
| Технологии | 2012-12-12 06:31:13 |
| Финансы | 2012-12-03 04:15:27 |
| Логистика | 2012-12-20 06:11:42 |

======================

select \* from (select news\_categories.nc\_name, news.n\_dt from news\_categories inner join news on nc\_id=n\_category and news.n\_dt=(select max(n2.n\_dt) from news as n2 where n2.n\_category=news\_categories.nc\_id) union all select reviews\_categories.rc\_name, reviews.r\_dt from reviews\_categories inner join reviews on rc\_id=r\_category and r\_dt=(select max(r2.r\_dt) from reviews as r2 where reviews\_categories.rc\_id=r2.r\_category))t2 ;

======================

1. Написать запрос, показывающий список страниц сайта верхнего уровня (у таких страниц нет родительской страницы) и список баннеров для каждой такой страницы.

В результате выполнения запроса должно получиться:

|  |  |  |
| --- | --- | --- |
| **p\_name** | **b\_id** | **b\_url** |
| Юридическим лицам | 1 | http://tut.by |
| Юридическим лицам | 2 | http://tut.by |
| Юридическим лицам | 7 | http://habrahabr.ru |
| Физическим лицам | 1 | http://tut.by |

======================

select p\_name, m2m\_banners\_pages.b\_id, b\_url from banners join m2m\_banners\_pages on banners.b\_id=m2m\_banners\_pages.b\_id join pages on m2m\_banners\_pages.p\_id=pages.p\_id where p\_parent IS NULL;

======================

1. Написать запрос, показывающий список страниц сайта, на которых есть баннеры.

В результате выполнения запроса должно получиться:

|  |
| --- |
| **p\_name** |
| Юридическим лицам |
| Физическим лицам |
| Образцы договоров |
| Банковские реквизиты |
| Схема проезда к офису |

======================

select p\_name from pages join m2m\_banners\_pages on m2m\_banners\_pages.p\_id=pages.p\_id group by p\_name

======================

1. Написать запрос, показывающий список страниц сайта, на которых нет баннеров.

В результате выполнения запроса должно получиться:

|  |
| --- |
| **p\_name** |
| Почта и телефон |
| Договоры оптовых закупок |

======================

select p\_name from pages left join m2m\_banners\_pages on m2m\_banners\_pages.p\_id=pages.p\_id where m2m\_banners\_pages.p\_id IS NULL group by p\_name ;

======================

1. Написать запрос, показывающий список баннеров, размещённых хотя бы на одной странице сайта.

В результате выполнения запроса должно получиться:

|  |  |
| --- | --- |
| **b\_id** | **b\_url** |
| 1 | http://tut.by |
| 2 | http://tut.by |
| 7 | http://habrahabr.ru |
| 4 | http://onliner.by |
| 3 | http://onliner.by |

======================

select m2m\_banners\_pages.b\_id, banners.b\_url from banners join m2m\_banners\_pages on banners.b\_id=m2m\_banners\_pages.b\_id group by m2m\_banners\_pages.b\_id

======================

1. Написать запрос, показывающий список баннеров, не размещённых ни на одной странице сайта.

В результате выполнения запроса должно получиться:

|  |  |
| --- | --- |
| **b\_id** | **b\_url** |
| 5 | http://google.by |
| 6 | http://google.com |
| 8 | http://habrahabr.ru |
| 9 | http://gismeteo.by |
| 10 | http://gismeteo.ru |

======================

select banners.b\_id, banners.b\_url from banners LEFT join m2m\_banners\_pages on banners.b\_id=m2m\_banners\_pages.b\_id where m2m\_banners\_pages.b\_id IS NULL;

======================

1. Написать запрос, показывающий баннеры, для которых отношение кликов к показам >= 80% (при условии, что баннер был показан хотя бы один раз).

В результате выполнения запроса должно получиться:

|  |  |  |  |
| --- | --- | --- | --- |
| b\_id | b\_url | rate |  |
| 2 | http://tut.by | 150.0000 | Да-да, в исходных данных ошибка. Потому тут – 150%. |
| 3 | http://onliner.by | 90.0000 |  |
| 5 | http://google.by | 100.0000 |  |
| 6 | http://google.com | 100.0000 |  |
| 7 | http://habrahabr.ru | 99.7998 |  |
| 8 | http://habrahabr.ru | 98.0000 |  |

======================

select banners.b\_id, banners.b\_url, banners.b\_click/banners.b\_show\*100 as rate from banners where b\_show>=1 having rate is not null and rate>80 ;

======================

1. Написать запрос, показывающий список страниц сайта, на которых показаны баннеры с текстом (в поле `b\_text` не NULL).

В результате выполнения запроса должно получиться:

|  |
| --- |
| **p\_name** |
| Юридическим лицам |
| Физическим лицам |
| Банковские реквизиты |
| Схема проезда к офису |

======================

select pages.p\_name from pages inner join m2m\_banners\_pages on m2m\_banners\_pages.p\_id=pages.p\_id inner join banners on banners.b\_id=m2m\_banners\_pages.b\_id where banners.b\_text IS NOT NULL group by pages.p\_name;

======================

1. Написать запрос, показывающий список страниц сайта, на которых показаны баннеры с картинкой (в поле `b\_pic` не NULL).

В результате выполнения запроса должно получиться:

|  |
| --- |
| **p\_name** |
| Юридическим лицам |
| Образцы договоров |
| Банковские реквизиты |
| Схема проезда к офису |

======================

select pages.p\_name from pages inner join m2m\_banners\_pages on m2m\_banners\_pages.p\_id=pages.p\_id inner join banners on banners.b\_id=m2m\_banners\_pages.b\_id where banners.b\_pic IS NOT NULL group by pages.p\_name;

======================

1. Написать запрос, показывающий список публикаций (новостей и обзоров) за 2011-й год.

В результате выполнения запроса должно получиться:

|  |  |
| --- | --- |
| **header** | **date** |
| Контрабанда железобетонных плит | 2011-09-14 06:19:08 |
| Почта России: вчера, сегодня и снова вчера | 2011-08-17 09:06:30 |
| Роботы на страже строек | 2011-10-03 05:17:37 |

======================

select \* from (select news.n\_header header, news.n\_dt from news where EXTRACT(YEAR FROM n\_dt)= 2011 union all select reviews.r\_header header, reviews.r\_dt from reviews where EXTRACT(YEAR FROM r\_dt)= 2011) t ;

======================

1. Написать запрос, показывающий список категорий публикаций (новостей и обзоров), в которых нет публикаций.

В результате выполнения запроса должно получиться:

|  |
| --- |
| **category** |
| Законодательство |
| Строительство |
| Товары и услуги |

======================

select \* from (select news\_categories.nc\_name from news\_categories left join news on nc\_id = n\_category where news.n\_category is null union all select reviews\_categories.rc\_name from reviews\_categories left join reviews on rc\_id = r\_category where reviews.r\_category is null) t2;

======================

1. Написать запрос, показывающий список новостей из категории «Логистика» за 2012-й год.

В результате выполнения запроса должно получиться:

|  |  |
| --- | --- |
| **n\_header** | **n\_dt** |
| Самолётом или поездом? | 2012-12-20 06:11:42 |
| Куда всё катится? | 2012-12-11 04:36:17 |

======================

select news.n\_header,news.n\_dt from news join news\_categories on news.n\_category=news\_categories.nc\_id and news\_categories.nc\_name='Логистика' where EXTRACT(YEAR FROM n\_dt)= 2012

======================

1. Написать запрос, показывающий список годов, за которые есть новости, а также количество новостей за каждый из годов.

В результате выполнения запроса должно получиться:

|  |  |
| --- | --- |
| **year** | **COUNT(\*)** |
| 2011 | 2 |
| 2012 | 3 |

======================

select EXTRACT(YEAR FROM n\_dt) as d, count(\*) from news group by d;

======================

1. Написать запрос, показывающий URL и id таких баннеров, где для одного и того же URL есть несколько баннеров.

В результате выполнения запроса должно получиться:

|  |  |
| --- | --- |
| **b\_url** | **b\_id** |
| http://tut.by | 1 |
| http://tut.by | 2 |
| http://onliner.by | 3 |
| http://onliner.by | 4 |
| http://habrahabr.ru | 7 |
| http://habrahabr.ru | 8 |

======================

select banners.b\_url,b\_id from banners where banners.b\_url IN(select b\_url from banners group by b\_url having count(\*) > 1);

======================

1. Написать запрос, показывающий список непосредственных подстраниц страницы «Юридическим лицам» со списком баннеров этих подстраниц.

В результате выполнения запроса должно получиться:

|  |  |  |
| --- | --- | --- |
| **p\_name** | **b\_id** | **b\_url** |
| Образцы договоров | 4 | http://onliner.by |
| Банковские реквизиты | 1 | http://tut.by |
| Банковские реквизиты | 2 | http://tut.by |

======================

select p.p\_name, b.b\_id, b.b\_url from pages as p2

inner join pages as p on p.p\_parent=p2.p\_id and p2.p\_name='Юридическим лицам'

inner join m2m\_banners\_pages as m on m.p\_id=p.p\_id

inner join banners as b on b.b\_id=m.b\_id;

======================

1. Написать запрос, показывающий список всех баннеров с картинками (поле `b\_pic` не NULL), отсортированный по убыванию отношения кликов по баннеру к показам баннера.

В результате выполнения запроса должно получиться:

|  |  |  |  |
| --- | --- | --- | --- |
| **b\_id** | **b\_url** | **rate** |  |
| 2 | http://tut.by | 1.5000 | Да-да, в исходных данных ошибка. Потому тут – 150%. |
| 5 | http://google.by | 1.0000 |  |
| 6 | http://google.com | 1.0000 |  |
| 7 | http://habrahabr.ru | 0.9980 |  |
| 4 | http://onliner.by | 0.0200 |  |

======================

select banners.b\_id, banners.b\_url, banners.b\_click/banners.b\_show as rate from banners where b\_pic is NOT NULL order by rate desc;

======================

1. Написать запрос, показывающий самую старую публикацию на сайте (не важно – новость это или обзор).

В результате выполнения запроса должно получиться:

|  |  |
| --- | --- |
| **header** | **date** |
| Почта России: вчера, сегодня и снова вчера | 2011-08-17 09:06:30 |

======================

select \* from (select news.n\_header header, news.n\_dt date from news union all select reviews.r\_header header, reviews.r\_dt date from reviews) t order by date asc limit 1;

======================

1. Написать запрос, показывающий список баннеров, URL которых встречается в таблице один раз.

В результате выполнения запроса должно получиться:

|  |  |
| --- | --- |
| **b\_url** | **b\_id** |
| http://google.by | 5 |
| http://google.com | 6 |
| http://gismeteo.by | 9 |
| http://gismeteo.ru | 10 |

======================

select banners.b\_url,b\_id from banners where banners.b\_url IN(select b\_url from banners group by b\_url having count(\*) = 1);

======================

1. Написать запрос, показывающий список страниц сайта в порядке убывания количества баннеров, расположенных на странице. Для случаев, когда на нескольких страницах расположено одинаковое количество баннеров, этот список страниц должен быть отсортирован по возрастанию имён страниц.

В результате выполнения запроса должно получиться:

|  |  |
| --- | --- |
| **p\_name** | **banners\_count** |
| Схема проезда к офису | 3 |
| Юридическим лицам | 3 |
| Банковские реквизиты | 2 |
| Образцы договоров | 1 |
| Физическим лицам | 1 |

======================

select p.p\_name, banners\_count from pages as p join (select m2m\_banners\_pages.p\_id, count(\*) as banners\_count from m2m\_banners\_pages group by m2m\_banners\_pages.p\_id ) as temp on temp.p\_id=p.p\_id order by banners\_count desc, p\_name asc;

======================

1. Написать запрос, показывающий самую «свежую» новость и самый «свежий» обзор.

В результате выполнения запроса должно получиться:

|  |  |
| --- | --- |
| **header** | **date** |
| Самолётом или поездом? | 2012-12-20 06:11:42 |
| Когда всё это кончится?! | 2012-12-12 06:31:13 |

======================

select news.n\_header as header, news.n\_dt as date from news where news.n\_dt=(select max(news.n\_dt) from news) union all select reviews.r\_header as header, reviews.r\_dt as date from reviews where reviews.r\_dt=(select max(reviews.r\_dt) from reviews);

======================

1. Написать запрос, показывающий баннеры, в тексте которых встречается часть URL, на который ссылается баннер.

В результате выполнения запроса должно получиться:

|  |  |  |
| --- | --- | --- |
| **b\_id** | **b\_url** | **b\_text** |
| 1 | http://tut.by | TUT.BY |
| 3 | http://onliner.by | ONLINER.BY |
| 5 | http://google.by | GOOGLE.BY |
| 8 | http://habrahabr.ru | HABRAHABR.RU |

======================

select banners.b\_id, banners.b\_url, banners.b\_text from banners where banners.b\_url like concat('%',b\_text,'%');

======================

1. Написать запрос, показывающий страницу, на которой размещён баннер с самым высоким отношением кликов к показам.

В результате выполнения запроса должно получиться:

|  |
| --- |
| **p\_name** |
| Юридическим лицам |

======================

select p.p\_name from pages as p inner join m2m\_banners\_pages as m on p.p\_id=m.p\_id inner join banners as b on m.b\_id=b.b\_id and (b.b\_click/b.b\_show) = (select max(banners.b\_click/banners.b\_show) from banners );

======================

1. Написать запрос, считающий среднее отношение кликов к показам по всем баннерам, которые были показаны хотя бы один раз.

В результате выполнения запроса должно получиться:

|  |
| --- |
| **AVG(`b\_click`/`b\_show`)** |
| 0.81224975 |

======================

select avg(banners.b\_click/banners.b\_show) from banners where banners.b\_show>=1;

======================

1. Написать запрос, считающий среднее отношение кликов к показам по баннерам, у которых нет графической части (поле `b\_pic` равно NULL).

В результате выполнения запроса должно получиться:

|  |
| --- |
| **AVG(`b\_click`/`b\_show`)** |
| 0.66000000 |

======================

select avg(banners.b\_click/banners.b\_show) from banners where banners.b\_pic is null;

======================

1. Написать запрос, показывающий количество баннеров, размещённых на страницах сайта верхнего уровня (у таких страниц нет родительских страниц).

В результате выполнения запроса должно получиться:

|  |
| --- |
| COUNT |
| 4 |

======================

select count(\*) from pages as p inner join m2m\_banners\_pages as m on p.p\_id=m.p\_id and p.p\_parent is null

======================

1. Написать запрос, показывающий баннер(ы), который(ые) показаны на самом большом количестве страниц.

В результате выполнения запроса должно получиться:

|  |  |  |
| --- | --- | --- |
| **b\_id** | **b\_url** | **COUNT** |
| 1 | http://tut.by | 4 |

======================

select banners.b\_id, b\_url, count(p\_id) count from banners

join m2m\_banners\_pages on (banners.b\_id=m2m\_banners\_pages.b\_id)

Group by b\_id, b\_url

having count=(

select Max(count) from(

select banners.b\_id, b\_url, count(p\_id) count from banners

join m2m\_banners\_pages on (banners.b\_id=m2m\_banners\_pages.b\_id)

Group by b\_id, b\_url order by count desc)

t );

======================

1. Написать запрос, показывающий страницу(ы), на которой(ых) показано больше всего баннеров.

В результате выполнения запроса должно получиться:

|  |  |
| --- | --- |
| **p\_name** | **COUNT** |
| Юридическим лицам | 3 |
| Схема проезда к офису | 3 |

======================

select \* from (select p.p\_name, banners\_count from pages as p join (select m2m\_banners\_pages.p\_id, count(\*) as banners\_count from m2m\_banners\_pages group by m2m\_banners\_pages.p\_id ) as temp on temp.p\_id=p.p\_id order by banners\_count desc, p\_name asc)t2

having banners\_count=(select Max(banners\_count) from (select p.p\_name, banners\_count from pages as p join (select m2m\_banners\_pages.p\_id, count(\*) as banners\_count from m2m\_banners\_pages group by m2m\_banners\_pages.p\_id ) as temp on temp.p\_id=p.p\_id order by banners\_count desc, p\_name asc)t3);

======================