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

SELECT SUM(two.num\_news\_reviews) AS 'sum'

FROM (

SELECT count(\*) AS num\_news\_reviews

FROM reviews

UNION

SELECT count(\*) AS num\_news\_reviews

FROM news

) two;

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

SELECT nc\_name, COUNT(n\_id)

FROM news\_categories

LEFT JOIN news ON n\_category = nc\_id

GROUP BY nc\_name;

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

SELECT rc\_name, COUNT(r\_id)

FROM reviews\_categories

LEFT JOIN reviews ON r\_category = rc\_id

GROUP BY rc\_name;

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

SELECT rc\_name, MAX(r\_dt)

FROM reviews\_categories

JOIN reviews ON r\_category = rc\_id

GROUP BY rc\_name

UNION

SELECT nc\_name, MAX(n\_dt)

FROM news\_categories

JOIN news ON nc\_id=n\_category

GROUP BY nc\_name;

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

SELECT p.p\_name, b.b\_id, b.b\_url

FROM pages p

JOIN m2m\_banners\_pages m2m ON p.p\_id = m2m.p\_id

JOIN banners b ON m2m.b\_id = b.b\_id

WHERE p.p\_parent IS NULL;

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

SELECT DISTINCT p.p\_name

FROM pages p

JOIN m2m\_banners\_pages m2m ON m2m.p\_id = p.p\_id

WHERE m2m.b\_id IS NOT NULL;

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

SELECT DISTINCT p.p\_name

FROM pages p

LEFT JOIN m2m\_banners\_pages m2m ON m2m.p\_id = p.p\_id

WHERE m2m.b\_id IS NULL;

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

SELECT DISTINCT b.b\_id, b.b\_url

FROM banners b

JOIN m2m\_banners\_pages m2m ON m2m.b\_id = b.b\_id

JOIN pages p ON m2m.p\_id = p.p\_id

WHERE p.p\_name IS NOT NULL;

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

SELECT DISTINCT b.b\_id, b.b\_url

FROM banners b

LEFT JOIN m2m\_banners\_pages m2m ON m2m.b\_id = b.b\_id

LEFT JOIN pages p ON m2m.p\_id = p.p\_id

WHERE p.p\_name IS NULL;

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

SELECT b\_id, b\_url, b\_click/b\_show\*100 AS rate

FROM banners

WHERE b\_click >= b\_show\*0.8 and b\_show <> 0;

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

SELECT DISTINCT p.p\_name

FROM pages p

JOIN m2m\_banners\_pages m2m ON m2m.p\_id = p.p\_id

JOIN banners b ON b.b\_id = m2m.b\_id

WHERE b.b\_text IS NOT NULL;

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

SELECT DISTINCT p.p\_name

FROM pages p

JOIN m2m\_banners\_pages m2m ON m2m.p\_id = p.p\_id

JOIN banners b ON b.b\_id = m2m.b\_id

WHERE b.b\_pic IS NOT NULL;

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

SELECT n\_header AS 'header', n\_dt AS 'date'

FROM news

WHERE n\_dt BETWEEN '2011-01-01' AND '2011-12-31'

UNION

SELECT r\_header,r\_dt

FROM reviews

WHERE r\_dt BETWEEN '2011-01-01' AND '2011-12-31';

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

SELECT rc\_name AS 'category'

FROM reviews\_categories

LEFT JOIN reviews ON rc\_id = r\_category

WHERE r\_category IS NULL

UNION

SELECT nc\_name

FROM news\_categories

LEFT JOIN news ON nc\_id = n\_category

WHERE n\_category IS NULL;

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

SELECT n\_header , n\_dt

FROM news

JOIN news\_categories ON n\_category = nc\_id

WHERE n\_dt BETWEEN '2012-01-01' AND '2012-12-31'

AND nc\_name = 'Логистика';

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

SELECT year(n\_dt) AS 'year', count(\*)

FROM news

GROUP BY year(n\_dt);

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

SELECT b.b\_url, b.b\_id

FROM banners b

JOIN banners s ON b.b\_url = s.b\_url

WHERE b.b\_id<s.b\_id

UNION

(SELECT s.b\_url,s.b\_id

FROM banners b

JOIN banners s ON b.b\_url = s.b\_url

WHERE b.b\_id<s.b\_id)

ORDER BY b\_id;

-- 18 Hаписать запрос, показывающий список непосредственных подстраниц страницы «Юридическим лицам» со списком баннеров этих подстраниц.

SELECT pp.p\_name, b.b\_id, b.b\_url

FROM pages p

JOIN pages pp ON p.p\_id =pp. p\_parent

JOIN m2m\_banners\_pages m2m ON pp.p\_id = m2m.p\_id

JOIN banners b ON m2m.b\_id = b.b\_id

WHERE p.p\_name = 'Юридическим лицам';

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

SELECT b\_id, b\_url , b\_click/b\_show AS 'rate'

FROM banners

WHERE b\_pic IS NOT NULL

ORDER by rate desc;

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

SELECT header, date

FROM (SELECT n\_header AS header, n\_dt AS date

FROM news

UNION

SELECT r\_header, r\_dt

FROM reviews) as h1

WHERE date = (SELECT LEAST(MIN(n\_dt), MIN(r\_dt)) FROM news, reviews);

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

SELECT b\_url, b\_id

FROM banners

WHERE b\_url IN (SELECT b\_url

FROM banners

GROUP BY b\_url

HAVING COUNT(\*) = 1);

-- 22 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.

SELECT p.p\_name , count(b.b\_id) as 'banners\_count'

FROM pages p

JOIN m2m\_banners\_pages m2m ON m2m.p\_id = p.p\_id

JOIN banners b ON b.b\_id = m2m.b\_id

group by p.p\_name

order by banners\_count DESC, p.p\_name ASC;

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

SELECT n\_header AS header, n\_dt AS date

FROM news

WHERE n\_dt=(SELECT MAX(n\_dt) FROM news)

UNION

SELECT r\_header, r\_dt

FROM reviews

WHERE r\_dt=(SELECT MAX(r\_dt) FROM reviews);

-- 24 Написать запрос, показывающий баннеры, в тексте которых встречается часть URL, на который ссылается баннер

SELECT b\_id,b\_url, b\_text

FROM banners

WHERE b\_text = substr(b\_url,8);

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

SELECT p.p\_name

FROM pages p

JOIN m2m\_banners\_pages m2m ON m2m.p\_id = p.p\_id

JOIN banners b ON b.b\_id = m2m.b\_id

ORDER by b.b\_click/b.b\_show DESC

LIMIT 1;

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

SELECT AVG(b\_click/b\_show)

FROM banners

WHERE b\_show <> 0;

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

SELECT AVG(b\_click/b\_show)

FROM banners

WHERE b\_show <> 0 AND b\_pic IS NULL;

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

SELECT COUNT(b.b\_id) AS COUNT

FROM pages p

JOIN m2m\_banners\_pages m2m ON m2m.p\_id = p.p\_id

JOIN banners b ON b.b\_id = m2m.b\_id

WHERE p.p\_parent IS NULL;

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

SELECT b.b\_id, b.b\_url, count( m2m.b\_id) AS `COUNT`

FROM m2m\_banners\_pages AS m2m

JOIN banners AS b ON b.b\_id = m2m.b\_id

GROUP BY m2m.b\_id

limit 1;

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

SELECT p\_name, count\_outer

FROM pages, (

SELECT p\_id, COUNT(\*) AS count\_outer

FROM m2m\_banners\_pages

GROUP BY p\_id

) t1

WHERE pages.p\_id = t1.p\_id

AND count\_outer = (

SELECT MAX(count\_inner)

FROM (

SELECT COUNT(\*) AS count\_inner

FROM m2m\_banners\_pages

GROUP BY p\_id

) t2

);