**Students Lab 20’ Pre Assignment Task**

**TASK #1 Database.**

Есть 5 таблиц :

STUDENT (id, first\_name, last\_name)

LECTURER (id, first\_name, last\_name)

TRAINING\_COURSE(id, name, teacher\_id)

STUDENT\_RESULT(id, student\_id, exam\_id, training\_course\_id, result, date)

EXAM\_RESULT(id, student\_id, exam\_id, teacher\_id, result)

Необходимо, по предоставленному описанию таблиц определить связи между таблицами, а также найти и исправить возможную избыточную/лишнюю или наоборот недостающую информацию в условии описания таблиц.

После этого составить следующие запросы:

1. Выбрать имена и фамилии студентов, успешно сдавших экзамен, упорядоченных по результату экзамена (отличники первые в результате)

2. Посчитать количество студентов, успешно сдавших экзамен на 4 и 5

3. Посчитать количество студентов, сдавших экзамен “автоматом” (нет записи в таблице exam\_result но есть положительный результат в таблице student\_result)

4. Посчитать средний балл студентов по предмету с наименованием “Системы управления базами данных”

5. Выбрать имена и фамилии студентов, не сдававших экзамен по предмету “Теория графов” (2 вида запроса)

6. Выбрать идентификатор преподавателей, читающих лекции по больше чем по 2 предметам

7. Выбрать идентификатор и фамилии студентов, пересдававших хотя бы 1 предмет (2 и более записи в exam\_result)

8. Вывести имена и фамилии 5 студентов с максимальными оценками

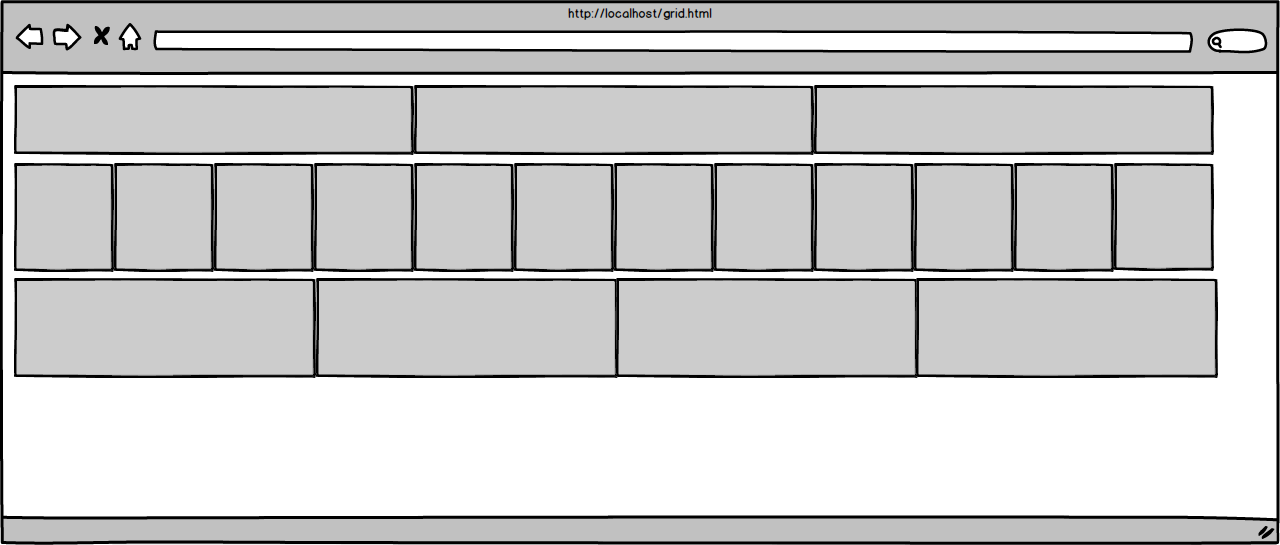
9. Вывести фамилию преподавателя, у которого наилучшие результаты по его предметам

10. Вывести успеваемость студентов по годам по предмету “Математическая статистика”

Как результат необходимо выслать графическое отображения таблиц и их связей (можно использовать любой графически тул, хоть Paint :)), а также 10 запросов, корректно работающих.

**TASK #2 CSS.**

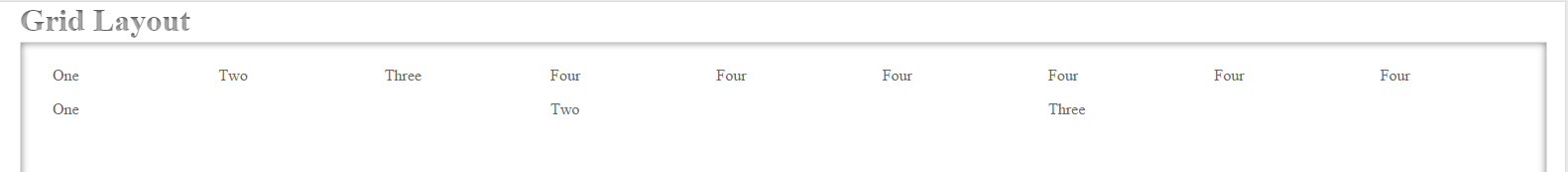
Создать разметку “grid layout” - виртуальную сетку, позволяющую удобно размещать элементы в ее ячейки. Примерный вариант grid layout:



Серые прямоугольники - это ячейки, в которых можно размещать необходимые данные (изображения, текст, таблицы, блоки и т.д.)

Важное условие: при изменении непосредственно ячейки - добавлении границы - она не должна перемещаться на следующую строку:

Ячейки без границ располагаются в одной строке:



добавление толстых границ не повлияло на расположение ячеек:



Grid Layout должен иметь следующую структуру:

Класс ‘jlab-row’ должен определять ряд, который занимал бы 100% ширины родительского компонента.

Класс ‘jlab-cell-$’, где $ - это число от 1 до 12, должен занимать $/12 ширины jlab-row. Например, jlab-cell-4 занимает третью часть ряда:



изменение размеров страницы не должно влиять на расположение ячеек - они не должны “падать” на следующую строку.

**TASK #3 Coding challenge.**

File Upload App

Необходимо написать приложение, которое будет предоставлять возможность загружать и отображать файлы определенных форматов (JPG, PNG, GIF).

1. Необходимо предусмотреть несколько типов валидаторов : на уникальность имени файла, на формат файла и на размер файла (не более 2 Мб)
2. Предусмотреть вывод сообщений если одно из условий выше не соблюдается
3. Познакомиться с Command Pattern и попробовать реализовать его (в нашем случае 2 типа команд - отобразить и загрузить)
4. Познакомиться с Factory Pattern попробовать его использовать для создания различных типов валидаторов
5. Файлы сохраняем на локальную машину (предусмотреть возможность использования приложения на разных операционных системах)

Технологии: Servlets, JSP, JS, Maven, Git, Heroku\*

Как результат необходимо развернуть приложение на Heroku (https://www.heroku.com/free) и предоставить ссылки на Github и на работающее приложение в клауде.