**Задание 1 (пункты а и б) представить в виде модуля, содержащего процедуры.**

**Задание 2 (пункты с и d) представить в виде триггеров.**

**Предусмотреть обработку пользовательских исключений в триггерах и в процедурах, в которых есть список входных параметров!**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  | | --- | --- | --- | |  | **Группа 3071** |  | | |  |  |  | | --- | --- | --- | | 1 | Асанова | Елизавета | | 2 | Белякова | Ольга | | 3 | Бычкова | Валентина | | 4 | Вольцов | Александр | | 5 | Глазков | Иван | | 6 | Голубев | Ярослав | | 7 | Матвеенко | Никита | | 8 | Панина | Оксана | | 9 | Сидоров | Игорь | | 10 | Теряев | Лев | | 11 | Чистяков | Владислав | |  |  | |  |  |  | |  |  |  | |  |

**Вариант 1**

1. Вывести на экран для заданного курса (входной параметр) 3-х лучших студентов, указав для каждого из балл, ФИО, группу, количество завершенных и количество действующих на текущую дату хобби. Данные отсортировать по убыванию балла.
2. Вывести на экран для каждого курса отдельно количество «отличников», «хорошистов» и «троечников».
3. Запретить перевод студента в группу, в которой средний балл ниже балла переводимого студента.
4. Предоставить пользователю возможность удалить только тех студентов из БД, чей балл < 4 и они не имеют ни одного действующего хобби. Все удаленные кортежи сохранить в Архиве с указанием даты и времени изменения и логина пользователя, внесшего эти изменения.

**Вариант 2**

1. Входной параметр (фамилия и имя куратора). Вывести на экран список студентов его групп, отсортировав данные по группе по возрастанию, внутри группы по среднему баллу по убыванию.
2. Входные параметры (номер курса, номер группы). Вывести список студентов всех групп заданного курса, имеющих средний балл не меньше среднего балла в указанной группе, указав для каждого из них количество действующих хобби и количество законченных хобби, отсортировать по группе по возрастанию, по баллу по убыванию.
3. Запретить изменять студенту балл в даты, не совпадающие со сроками сессий. Любые изменения в таблице студенты сохранять в «архиве».
4. Запретить изменять куратора во всех группах 3-го и 2-го курсов.

**Вариант 3**

1. Вывести на экран для заданного куратора (фамилия и имя куратора — входные параметры) список курируемых групп, указав для каждой из них количество отличников и троечников. Данные отсортировать по количеству отличников по убыванию.
2. Для каждого куратора вывести количество студентов отдельно для каждого из хобби, содержащихся в БД, учитывая только действующие хобби.
3. Ограничить количество студентов в группе (число n выбрать самостоятельно).
4. Предоставить пользователю возможность удалить куратора из БД, если курируемые им группы находятся на любом курсе, кроме первого. Все удаленные кортежи сохранить в Архиве с указанием даты и времени изменения и логина пользователя, внесшего эти изменения.

**Вариант 4**

1. Вывести на экран список текущих хобби заданного типа (входной параметр), которыми увлекаются не менее заданного количества студентов (входной параметр). Отсортировать данные по убыванию риска хобби.
2. Вывести полную информацию о самом популярном хобби в заданной группе (номер группы — входной параметр).
3. Запретить добавлять хобби тем студентам, у которых средний балл < среднего балла по группе.
4. Изменение данных и удаление данных из таблицы Студенты\_Хобби фиксировать в таблице «архив», указав действие, логин пользователя, внесшего изменения, дату изменения и количество измененных (удаленных) записей для каждой операции.

**Вариант 5**

1. Вывести на экран для заданного предмета (входной параметр) список преподавателей, указав для каждого из них количество групп, в которых он ведет занятия по данному предмету, и суммарное количество студентов в этих группах. Данные отсортировать по фамилии по возрастанию.
2. Для каждого курса вывести общее количество предметов, изучаемых на курсе, количество предметов текущего семестра и суммарное количество студентов на курсе. Отсортировать по курсу по убыванию.
3. Запретить изменять преподавателя по предмету в группе для всех групп 1 курса.
4. Предоставить пользователю возможность удалить предмет из БД, если его не изучают в текущем семестре. Все удаленные кортежи сохранить в Архиве с указанием даты и времени изменения и логина пользователя, внесшего эти изменения.

**Вариант 6**

1. Вывести на экран список студентов, которые увлекаются хобби с заданным риском (входной параметр) не менее заданного количества целых месяцев/целых кварталов/целых лет (входной параметр) от текущей даты.
2. Вывести для каждой группы список самых популярных хобби на текущую дату, учитывая только действующие хобби с риском в заданных пределах (границы значения риска — входные параметры) для студентов, чей балл не менее заданного значения (входной параметр).
3. Запретить изменять риск хобби на значение, которое более, чем на 10% отличается от имеющегося в обе стороны.
4. Запретить удалять законченные хобби из БД, если прошло менее 3-х месяцев от текущей даты до даты окончания хобби.

**Вариант 7**

1. Вывести на экран список студентов с заданного курса (номер курса — входной параметр), которые увлекаются более n месяцев (n — входной параметр) любым хобби.
2. Вывести для каждого курса количество отличников, имеющих только действующие хобби заданного типа (тип хобби — входной параметр).
3. Запретить изменять тип хобби всем хобби заданного типа.
4. Все удаленные кортежи из таблицы Студенты записывать в «Архив», указав действие, логин пользователя, внесшего изменения, дату и время изменения и значения всех атрибутов удаленных кортежей.

**Вариант 8**

1. Вывести на экран список завершенных хобби студентов заданной группы (входной параметр). Отсортировать данные по возрастанию риска.
2. Вывести полную информацию о самом молодом (самых молодых) студенте заданного курса (номер курса — входной параметр).
3. Запретить добавлять хобби всем студентам 1-го курса, у которых день рождения в текущем месяце.
4. Изменение данных и удаление данных из таблицы Студенты\_Хобби фиксировать в таблице «архив», указав действие, логин пользователя, внесшего изменения и дату изменения

**Вариант 9**

1. Вывести на экран для первого и последнего курсов n лучших студентов (n — входной параметр), указав для каждого из них балл, ФИО, группу, количество завершенных и количество действующих на текущую дату хобби. Данные отсортировать по убыванию балла в пределах курса.
2. Вывести на экран для каждого типа занятий для заданного курса (номер курса — входной параметр) отдельно количество занятий каждого типа (лекция, семинар, лабораторная работа и т. п.).
3. Запретить перевод студента в группу, в которой средний балл в группе меньше балла переводимого студента или в новой группе количество студентов >=n.
4. Предоставить пользователю возможность удалить только тех студентов из БД, у которых нет ни одного действующего хобби. Все удаленные кортежи сохранить в Архиве с указанием даты и времени изменения, логина пользователя, внесшего эти изменения.

**Вариант 10**

* 1. Входные параметры (N\_курса, Месяц рождения). Вывести на экран список студентов с заданного курса, имеющих день рождения в заданном месяце, которым исполнилось 20 полных лет на текущую дату. Отсортировать по группе по возрастанию, внутри группы по убыванию балла.
  2. Вывести на экран список студентов, имеющих не менее заданного количества действующих/законченных (входной параметр) хобби (название хобби — входной параметр), сгруппировав данные по номеру группы, внутри группы отсортировать по убыванию количества хобби.
  3. Запретить любые изменения во всех полях таблицы студенты за исключением номера группы и среднего бала. При попытке изменения других атрибутов выдавать сообщение о запрете изменения на экран. При переводе студента из одной группы в другую, фиксировать изменения в «архиве», указав дату и время изменения и логин пользователя, внесшего изменения.
  4. Запретить перевод студента в группу курсом выше вне заданного временного интервала.

**Вариант 11**

1. Вывести на экран для каждого предмета список групп, изучающих этот предмет в текущем семестре, указав форму отчетности. Если предмет не изучается ни одной из групп в текущем семестре, он все равно должен присутствовать в списке.
2. Для каждой из групп заданного курса (входной параметр) вывести количество зачетов, зачетов с оценкой и экзаменов отдельно в текущем/предыдущем (входной параметр) семестре.
3. Запретить любые изменения во всех полях таблицы преподаватели за исключением фамилии. При попытке изменения других атрибутов выдавать сообщение о запрете изменения на экран. При изменениях в таблице преподаватели\_группы\_предметы фиксировать эти изменения в «архиве», указав дату и время изменения и логин пользователя, внесшего изменения.
4. Запретить изменять предмет для групп, если семестр, в котором этот предмет изучается, является текущим семестром.