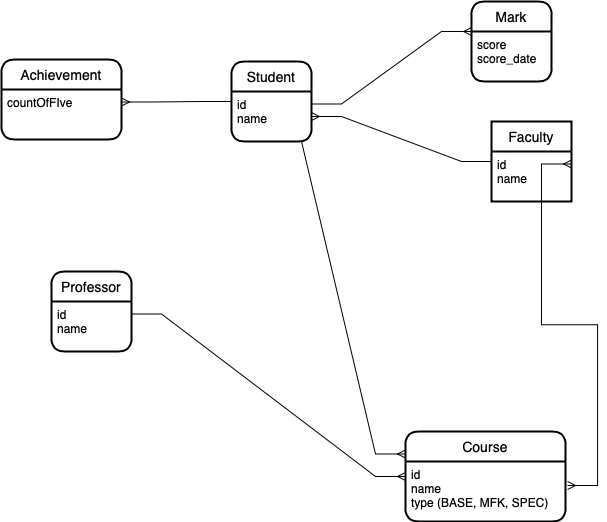
**ER-диаграмма и описание.**



1. **Описание сущностей**

* **Студент**:

Студент содержит атрибуты: id, имя, фамилия, дата рождения.

* **Достижения:**

Достижения содержат атрибуты: колонка для подсчёта отлов.

* **Профессор:**

Профессор содержит атрибуты: id преподавателя, имя, фамилия.

* **Курс:**

Курс содержит атрибуты: id курсы, название, тип.

* **Факультет:**

Факультет содержит атрибуты: id факультета, название.

* **Оценка:**

Оценка содержит атрибуты: сама оценка (2, 3, 4, 5).

1. **Общее описание**
2. Когда студент приходит в университет, то

* для добавления студента реализована хранимая процедура ***add\_student***, которая добавляет студента, если не существует студента с такими же данными (два раза одного студента не добавит).

1. Когда профессор приходит в университет, то

* будем заводить id профессора, его имя, фамилию;
* также будем ставить id профессора в конкретный курс (если, конечно, профессор ведёт этот курс, профессор может существовать и без курса).

1. Когда появляется новый факультет:

* заводим новый факультет с id и названием.

1. Когда появляется новый курс:

* курс бывает трёх типов (BASE, MFK, SPEC);
* устанавливается id, название, тип курса.

1. Когда студенту выставляется оценка:

* выставляется конкретному студенту по его id;
* сама оценка (2, 3, 4, 5);
* id курса, по которому выставляется оценка.

1. Достижения:

* табличка состоит их двух колонок: id студента и колонка countOfFives (количество пятёрок);
* с помощью хранимой процедуры ***counter\_of\_fives\_by\_student\_id*** для конкретного студента высчитывается общее количество отлов за все курсы.

1. Удаление преподавателя:

* для удаления преподавателя реализована хранимая процедура ***remove\_professor***, которая удаляет преподавателя только в том случае, если он не ведёт никаких курсов.

1. Запись студента на курс:

* есть две таблицы, таблица соответствия для предмета-курса-преподавателя и таблица доп.курсы. Запись студента на курс реализована с помощью хранимой процедуры ***enrolling\_student***. Сначала проверяется существует ли курс, на который мы хотим записать студента. Потом проверяем курс на тип. Если тип BASE, то запись невозможна, так как это базовый курс этого студента по факультету. Иначе записываем id студента и id курса в табличку доп.курсы. Также нельзя добавить два раза один и тот же курс одному студенту.

1. Описание триггеров:

* **Обновление достижений после добавления оценки.** Реализован триггер ***recalculate\_achievements***, который после добавления оценки с помощью описанной выше хранимой процедуры для подсчёта пятёрок, пересчитывает значение колокни ***count\_of\_five***.
* **Запись о пересдаче, если оценка получена после определенной даты.** Для этого создал табличку ***exam\_schedule***, в которой хранится расписание экзаменов по факультетам. Триггер же до выставления оценки проверяет дату экзамена. Если дата больше, чем дата этого экзамена, оценка за этот экзамен получает метку ***retaken*** в таблице ***marks*** (добавил новое поле). Иначе ***passed***.