### Информационная система вуза. Подсистема учета посещаемости

#### Предметная область

Студенты, организованные в группы, посещают занятия по дисциплинам в рамках учебного плана. Учебный процесс курируют деканаты факультетов, а преподаватели контролируют посещаемость занятий. Пропуски фиксируются преподавателями по каждому занятию с указанием причины (уважительная/неуважительная).

Система учета посещаемости предназначена для автоматизации процесса регистрации, хранения и анализа данных о посещаемости студентов. Она позволяет деканатам и преподавателям оперативно отслеживать нарушения дисциплины посещения, а студентам — контролировать свою посещаемость.

#### Виды запросов в информационной системе

1. Получить перечень и общее число студентов, имеющих пропуски за указанный период по заданной группе, курсу или факультету.
2. Получить данные о причинах пропусков занятий студентами за определённый период.
3. Получить список студентов, посещаемость которых ниже заданного уровня (например, 75%).
4. Сформировать отчет по посещаемости для конкретной группы, курса или факультета за семестр.
5. Получить данные о посещаемости конкретного студента за указанный период.
6. Сформировать список студентов, не посетивших занятия по конкретной дисциплине.

#### Постановка задачи

Разработать ИС учета посещаемости, которая позволяет выполнять следующие действия:

* Ввод, редактирование и хранение данных о посещаемости;
* Выполнение запросов, указанных выше;
* Обеспечение элементов защиты информации (пароль, логин);
* Создание резервной копии базы данных и восстановление данных.

#### Сущности

1. **Студенты**
   * ID\_студента (INT)
   * ФИО (VARCHAR(255))
   * Факультет (VARCHAR(255))
   * Курс (INT)
   * Группа (VARCHAR(20))
2. **Преподаватели**
   * ID\_преподавателя (INT)
   * ФИО (VARCHAR(255))
   * Должность (VARCHAR(80))
3. **Дисциплины**
   * ID\_дисциплины (INT)
   * Название дисциплины (VARCHAR(100))
   * ID\_Преподавателя (INT)
4. **Посещаемость**
   * ID\_студента (INT)
   * ID\_дисциплины (INT)
   * Дата занятия (DATE)
   * Статус посещаемости (BOOLEAN, где 1 — присутствовал, 0 — отсутствовал)
   * Причина пропуска (VARCHAR(255))

**Отношения**

* **Студент – Посещаемость**  
  Студент имеет записи о посещаемости занятий.  
  Связь: Студент может иметь записи о посещаемости по разным дисциплинам. Каждый студент может быть отмечен за посещение нескольких занятий по каждой дисциплине.
* **Студент – Дисциплина**  
  Студент изучает дисциплины согласно учебному плану.  
  Связь: Студент может изучать несколько дисциплин, в рамках которых будет отслеживаться посещаемость.
* **Преподаватель – Дисциплина**  
  Преподаватель ведет определенную дисциплину.  
  Связь: Преподаватель может вести несколько дисциплин, на которых будет отслеживаться посещаемость студентов.
* **Преподаватель – Посещаемость**  
  Преподаватель отслеживает посещаемость студентов на своих занятиях.  
  Связь: Преподаватель может отмечать посещаемость множества студентов на разных занятиях по своей дисциплине.
* **Посещаемость – Студент**  
  Запись о посещаемости принадлежит конкретному студенту.  
  Связь: Каждый студент имеет несколько записей о посещаемости по дисциплинам и занятиям, на которых он присутствовал или отсутствовал.

#### Процессы (события)

1. **Регистрация посещаемости**
   * Процесс добавления данных о посещаемости занятий.
   * Шаги:
     + Проверка наличия записи о студенте и дисциплине;
     + Запись данных о посещаемости (дата, статус, причина);
2. **Обновление данных**
   * Процесс корректировки ранее внесенных данных о посещаемости.
   * Шаги:
     + Проверка доступа пользователя;
     + Редактирование данных;
     + Логирование изменений.
3. **Генерация отчетов**
   * Процесс формирования отчетов о посещаемости.
   * Шаги:
     + Выбор параметров отчета (группа, курс, период);
     + Генерация отчета;
     + Сохранение или отправка отчета заинтересованным сторонам.

#### Пример использования

1. **Студенты**
   * Иванов И.И.
     + Факультет: ИТ
     + Группа: Группа 1
2. **Посещаемость**
   * Иванов И.И.
     + Математика
     + 01.06.2024
     + Пропуск (уважительная причина: медицинская справка).

#### Основные задачи разработки

* Разработать интерфейс для ввода и редактирования данных;
* Реализовать механизм генерации отчетов;
* Обеспечить защиту данных и доступ по ролям (администратор, преподаватель, студент);
* Настроить создание резервных копий базы данных.