

The diagram is a complex ER model for a 'Система управления учебными планами' (Study Plan Management System). It features numerous entities, each represented by a rectangle with a header containing the entity name and a list of attributes. Relationships are depicted by lines connecting entities, with crow's foot notation indicating cardinalities and roles.

Entities and their attributes:

- H_СВОЙСТВА_ОТДЕЛОВ:** ИД, НАИМЕНОВАНИЕ, ПРИМЕЧАНИЕ, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_ХАРАКТЕРИСТИКИ_ОТДЕЛОВ:** СВОТД, ИД, ОТД, ИД, ЗНАЧЕНИЕ, ПРИЗНАК, ПРИМЕЧАНИЕ, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_КВАЛИФИКАЦИИ:** ИД, НАИМЕНОВАНИЕ, КОГДА СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_НАПРАВЛЕНИЯ_СПЕЦИАЛ:** ИД, УРОВЕНЬ, ДАТА ГОС, КВАЛ, ИД, НС, ИД, ТС, ИД, НАПС, ИД, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ, ПРИМЕЧАНИЕ.
- H_ТИПЫ_СТАНДАРТОВ:** ИД, НАИМЕНОВАНИЕ, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_ОТДЕЛЫ:** ИД, КОРОТКОЕ_ИМЯ, ИМЯ В ИМИН ПАДЕЖЕ, ИМЯ В РОД ПАДЕЖЕ, ИМЯ В ВИН ПАДЕЖЕ, ИМЯ В ТВОР ПАДЕЖЕ, ИМЯ В ПРЕД ПАДЕЖЕ, ОТД, ИД, ДАТА СОЗДАНИЯ, ДАТА ЛИКВИДАЦИИ, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_УЧЕБНЫЕ_ГОДА:** ИД, НАЧАЛО, КОНЕЦ.
- H_НАПР_СПЕЦ:** ИД, КОД, НАПР, СПЕЦ, НАИМЕНОВАНИЕ, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_ПЛАНЫ:** ИД, ТП, ИД, учебный ГОД, ОТД, ИД, ЗАКРЕПЛЕН ЗА, НАПС, ИД, КУРС, ФО, ИД, СРОК ОБУЧЕНИЯ, ДАТА УТВЕРЖДЕНИЯ, ПЛАН, ПРИЕМА, ПРОЧИЙ, ПЛАН, ИД, ПЛАН, ИД, ОСНОВ, НА, ПЛАН, ПРИЕМА, БЮДЖЕТ, ПРОХОДНОЙ БАЛЛ, ПОЛУПРОХОДНОЙ БАЛЛ, СПЕЦИАЛЬНЫЙ, ПРОХОДНОЙ БАЛЛ, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ, ПРИМЕЧАНИЕ, НОМЕР.
- H_УЧЕНИКИ:** ИД, ЧЛВК, ИД, ПРИЗНАК, СОСТОЯНИЕ, НАЧАЛО, КОНЕЦ, ПЛАН, ИД, ГРУППА, П, ПРОК, ИД, ВИД, ОБУЧ, ИД, ПРИМЕЧАНИЕ, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ, КОНЕЦ ПО ПРИКАЗУ, ВМЕСТО В СВЯЗИ С, ТЕКСТ.
- H_ГРУППЫ_ПЛАНОВ:** ИД, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_ОБУЧЕНИЯ:** ИД, ЧЛВК, ИД, ВИД, ОБУЧ, ИД, НКЗ, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_ВИДЫ_ОБУЧЕНИЯ:** ИД, АББРЕВИАТУРА, НАИМЕНОВАНИЕ, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_ЛЮДИ:** ИД, ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, ПИН, ИИН, ДАТА РОЖДЕНИЯ, ПОЛ, МЕСТО РОЖДЕНИЯ, ИНОСТРАН, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ, ДАТА СМЕРТИ.
- H_ЛЮДИ_ПО_ПАДЕЖАМ:** ИД, ЧЛВК, ИД, ПАДЕЖ, ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_ЭЛЕМЕНТЫ_СТРОК:** ИД, ОТД, ИД, НОМЕР СЕМЕСТРА, СТП, ИД, НЕДЕЛЬ, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_СТРОКИ_ПЛАНОВ:** ИД, КОМ, ИД, ЦД, ИД, НОМЕР В ЦИКЛЕ, ДИС, ИД, НОМЕР ДИСЦИПЛИНЫ ПО ВЫБОРУ, ПЛАН, ИД, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_ДИСЦИПЛИНЫ:** ИД, КОРОТКОЕ_ИМЯ, НАИМЕНОВАНИЕ, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_КОМПОНЕНТЫ:** ИД, АББРЕВИАТУРА, НАИМЕНОВАНИЕ, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_СВОЙСТВА_ВР:** ИД, НАИМЕНОВАНИЕ, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_ХАРАКТЕРИСТИКИ_ВИДОВ_РАБОТ:** ИД, СВР, ИД, ВР, ИД, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_ВИДЫ_РАБОТ:** ИД, ПОРЯДОК, АББРЕВИАТУРА, НАИМЕНОВАНИЕ, ЕДИНИЦА ИЗМЕРЕНИЯ, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_ОЦЕНКИ:** ИД, КОД, ПРИМЕЧАНИЕ, СОРТ.
- H_ТИПЫ_ВЕДОМОСТЕЙ:** ИД, НАИМЕНОВАНИЕ, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_ВЕДОМОСТИ:** ИД, ЧЛВК, ИД, НОМЕР ДОКУМЕНТА, ОЦЕНКА, СРОК СДАЧИ, СЭС, ИД, ТВ, ИД, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ, БЕД, ИД, СОСТОЯНИЕ, ОТД, ИД, БУКВА.
- H_ИЗМ_ЛЮДИ:** ИД, ЧЛВК, ИД, ДАТА, ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, ПИН, ИИН, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_СЕССИЯ:** ИД, СЭС, ИД, ЧЛВК, ИД, ДАТА, ВРЕМЯ, АУДИТОРИЯ, ДАТА К, ВРЕМЯ К, АУДИТОРИЯ К, УЧГОД, ГРУППА, СЕМЕСТР, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.
- H_ЦИКЛЫ_ДИСЦИПЛИН:** ИД, АББРЕВИАТУРА, НАИМЕНОВАНИЕ, КТО СОЗДАЛ, КОГДА СОЗДАЛ, КТО ИЗМЕНИЛ, КОГДА ИЗМЕНИЛ.

Relationships:

- H_ОТДЕЛЫ** (1) to **H_ХАРАКТЕРИСТИКИ_ОТДЕЛОВ** (1) via **СВОТД**.
- H_ОТДЕЛЫ** (1) to **H_ПЛАНЫ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_УЧЕНИКИ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_ЛЮДИ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_ЛЮДИ_ПО_ПАДЕЖАМ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_ЭЛЕМЕНТЫ_СТРОК** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_СТРОКИ_ПЛАНОВ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_ДИСЦИПЛИНЫ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_КОМПОНЕНТЫ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_СВОЙСТВА_ВР** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_ХАРАКТЕРИСТИКИ_ВИДОВ_РАБОТ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_ВИДЫ_РАБОТ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_ОЦЕНКИ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_ТИПЫ_ВЕДОМОСТЕЙ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_ВЕДОМОСТИ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_ИЗМ_ЛЮДИ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_СЕССИЯ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_ЦИКЛЫ_ДИСЦИПЛИН** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_УЧЕБНЫЕ_ГОДА** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_НАПР_СПЕЦ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_ПЛАНЫ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_УЧЕНИКИ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_ЛЮДИ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_ЛЮДИ_ПО_ПАДЕЖАМ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_ЭЛЕМЕНТЫ_СТРОК** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_СТРОКИ_ПЛАНОВ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_ДИСЦИПЛИНЫ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_КОМПОНЕНТЫ** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to **H_СВОЙСТВА_ВР** (1) via **ОТД**.
- H_ОТДЕЛЫ** (1) to <

Условие:

Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:

таблицы: НТИПЫВЕДОМОСТЕЙ, НВЕДОМОСТИ.

Вывести атрибуты: НТИПЫВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, НВЕДОМОСТИ.ДАТА.

Фильтры (AND):

а) НТИПЫВЕДОМОСТЕЙ.ИД = 3.

б) НВЕДОМОСТИ.ЧЛВКИД = 153285.

Вид соединения: INNER JOIN.

```
SELECT Н_ТИПЫ_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н_ВЕДОМОСТИ.ДАТА
FROM Н_ТИПЫ_ВЕДОМОСТЕЙ
INNER JOIN Н_ВЕДОМОСТИ ON Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД = Н_ВЕДОМОСТИ.ВЕД_ИД
WHERE Н_ТИПЫ_ВЕДОМОСТЕЙ.ИД = 3
AND Н_ВЕДОМОСТИ.ЧЛВК_ИД = 153285;
```

Запрос 2

Условие:

Сделать запрос для получения атрибутов из указанных таблиц, применив фильтры по указанным условиям:

Таблицы: НЛЮДИ, НОБУЧЕНИЯ, НУЧЕНИКИ.

Вывести атрибуты: НЛЮДИ.ИМЯ, НОБУЧЕНИЯ.ЧЛВКИД, НУЧЕНИКИ.НАЧАЛО.

Фильтры: (AND)

а) НЛЮДИ.ФАМИЛИЯ = Афанасьев.

б) НОБУЧЕНИЯ.ЧЛВКИД > 163276.

с) Н_УЧЕНИКИ.ИД > 1.

Вид соединения: RIGHT JOIN.

```
SELECT Н_ЛЮДИ.ИМЯ, Н_ОБУЧЕНИЯ.ЧЛВК_ИД, Н_УЧЕНИКИ.НАЧАЛО
FROM Н_ЛЮДИ
RIGHT JOIN Н_ОБУЧЕНИЯ ON Н_ЛЮДИ.ИД = Н_ОБУЧЕНИЯ.ЧЛВК_ИД
RIGHT JOIN Н_УЧЕНИКИ ON Н_УЧЕНИКИ.ВИД_ОБУЧ_ИД = Н_ОБУЧЕНИЯ.ВИД_ОБУЧ_ИД
WHERE Н_ЛЮДИ.ФАМИЛИЯ = 'Иванов'
AND Н_ОБУЧЕНИЯ.ЧЛВК_ИД < 163276
AND Н_УЧЕНИКИ.ИД > 1;
```

Запрос 3

Условие: Есть ли студенты группы 3102, которым больше 25 лет?

```
SELECT EXISTS (
    SELECT Н_люди.ИМЯ
    FROM Н_УЧЕНИКИ
    INNER JOIN Н_люди ON Н_люди.ИД = Н_УЧЕНИКИ.ЧЛВК_ИД
    WHERE Н_УЧЕНИКИ.ГРУППА = '3102'
    AND EXTRACT (YEAR FROM AGE(Н_люди.ДАТА_РОЖДЕНИЯ)) > 25);
```

Запрос 4

Условие: Группы, где в 2011 году было ровно 5 студентов на ФКТИУ

```
SELECT ГРУППЫ_КТИУ_2011.ГРУППА, ГРУППЫ_КТИУ_2011.КОЛИЧЕСТВО FROM
(
    SELECT Н_УЧЕНИКИ.ГРУППА, count(Н_УЧЕНИКИ.ИД) AS КОЛИЧЕСТВО
    FROM Н_УЧЕНИКИ
    JOIN Н_ПЛАНЫ ON Н_УЧЕНИКИ.ПЛАН_ИД = Н_ПЛАНЫ.ИД
    JOIN Н_ОТДЕЛЫ ON Н_ОТДЕЛЫ.ИД = Н_ПЛАНЫ.ОТД_ИД
    WHERE Н_ПЛАНЫ.УЧЕБНЫЙ_ГОД = '2011'
    AND Н_ОТДЕЛЫ.КОРОТКОЕ_ИМЯ = 'КТИУ'
    GROUP BY Н_УЧЕНИКИ.ГРУППА
) AS ГРУППЫ_КТИУ_2011
WHERE ГРУППЫ_КТИУ_2011.КОЛИЧЕСТВО = 5;
```

Запрос 5

Условие: Выведите таблицу со средними оценками студентов группы 4100 (Номер, ФИО, Ср_оценка), у которых средняя оценка меньше минимальной оценк(е|и) в группе 1100.

GROUP BY в этом запросе выполняет две ключевые функции: группировку данных для агрегатных функций и обеспечение корректной структуры выборки.

```
SELECT Н_люди.ИД, ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, AVG(CAST(ОЦЕНКА AS NUMERIC)) AS
СРЕДНЯЯ_ОЦЕНКА
FROM Н_люди
JOIN Н_УЧЕНИКИ ON Н_УЧЕНИКИ.ЧЛВК_ИД = Н_люди.ИД AND Н_УЧЕНИКИ.ГРУППА =
```

```

'4100'
JOIN Н_ВЕДОМОСТИ ON Н_ВЕДОМОСТИ.ЧЛВК_ИД = Н_ЛЮДИ.ИД AND ОЦЕНКА NOT IN
('осв', 'неявка', 'зачет', 'незач')

GROUP BY Н_ЛЮДИ.ИД, Н_ЛЮДИ.ФАМИЛИЯ, Н_ЛЮДИ.ИМЯ, Н_ЛЮДИ.ОТЧЕСТВО

HAVING AVG(CAST (ОЦЕНКА AS NUMERIC)) < (SELECT MIN(CAST (ОЦЕНКА AS NUMERIC))
AS MARK
FROM Н_УЧЕНИКИ
JOIN Н_ВЕДОМОСТИ ON Н_УЧЕНИКИ.ГРУППА = '1101'
AND Н_УЧЕНИКИ.ЧЛВК_ИД = Н_ВЕДОМОСТИ.ЧЛВК_ИД
AND ОЦЕНКА NOT IN('осв', 'неявка', 'зачет', 'незач')
);

```

Функция `AVG(CAST(ОЦЕНКА AS NUMERIC))` требует **объединения строк в группы**, чтобы вычислить среднее значение оценок **для каждого студента отдельно**.

- **Без GROUP BY:**
 - `AVG` рассчитает общую среднюю оценку **всех студентов группы 4100**, что противоречит условию (нужна средняя **по каждому студенту**).
- **С GROUP BY:**
 - Строки группируются по уникальным комбинациям полей `ИД`, `ФАМИЛИЯ`, `ИМЯ`, `ОТЧЕСТВО` (идентифицирующим студента),
 - `AVG` вычисляется **внутри каждой группы**, то есть для каждого студента.

Условие `HAVING AVG(...) < ...` фильтрует **уже сгруппированные данные**.

- **GROUP BY** гарантирует, что:
 - В выборке будут только студенты группы 4100 (из-за `JOIN` с `Н_УЧЕНИКИ`),
 - `HAVING` получит **среднюю оценку каждого студента** для сравнения с минимальной оценкой группы 1100.
-

Запрос 6

Условие: Получить список студентов, отчисленных до первого сентября 2012 года с очной или заочной формы обучения. В результат включить:

номер группы;

номер, фамилию, имя и отчество студента;

номер пункта приказа;

Для реализации использовать соединение таблиц.

```
SELECT Н_УЧЕНИКИ.ГРУППА, Н_ЛЮДИ.ИД, Н_ЛЮДИ.ИМЯ, Н_ЛЮДИ.ФАМИЛИЯ,  
Н_ЛЮДИ.ОТЧЕСТВО, Н_УЧЕНИКИ.П_ПРКОК_ИД  
FROM Н_УЧЕНИКИ  
JOIN Н_ЛЮДИ ON Н_ЛЮДИ.ИД = Н_УЧЕНИКИ.ЧЛВК_ИД  
JOIN Н_ПЛАНЫ ON Н_ПЛАНЫ.ИД = Н_УЧЕНИКИ.ПЛАН_ИД  
JOIN Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ ON Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ.ИД =  
Н_ПЛАНЫ.НАПС_ИД  
JOIN Н_НАПР_СПЕЦ ON Н_НАПР_СПЕЦ.ИД = Н_НАПРАВЛЕНИЯ_СПЕЦИАЛ.НАПС_ИД  
WHERE Н_ПЛАНЫ.НАПС_ИД IN (  
Н_УЧЕНИКИ.ПРИЗНАК = 'отчисл'  
AND Н_УЧЕНИКИ.КОНЕЦ_ПО_ПРИКАЗУ < '2012-09-01';
```

Запрос 7

Условие: Люди, не являвшиеся студентами ИТМО (нет записей в Н_УЧЕНИКИ)

Важно: нельзя использовать DISTINCT

WITHOUT EXISTS

```
SELECT Н_ЛЮДИ.ИД,  
Н_ЛЮДИ.ФАМИЛИЯ,  
Н_ЛЮДИ.ИМЯ,  
Н_ЛЮДИ.ОТЧЕСТВО  
FROM Н_ЛЮДИ  
WHERE NOT EXISTS (  
SELECT *  
FROM Н_УЧЕНИКИ  
JOIN Н_ПЛАНЫ ON Н_УЧЕНИКИ.ПЛАН_ИД = Н_ПЛАНЫ.ИД  
JOIN Н_ОТДЕЛЫ ON Н_ПЛАНЫ.ОТД_ИД = Н_ОТДЕЛЫ.ИД  
WHERE Н_УЧЕНИКИ.ЧЛВК_ИД = Н_ЛЮДИ.ИД  
);
```

```
SELECT Л.ИД, Л.ФАМИЛИЯ, Л.ИМЯ, Л.ОТЧЕСТВО  
FROM Н_ЛЮДИ Л  
LEFT JOIN (  
SELECT У.ЧЛВК_ИД  
FROM Н_УЧЕНИКИ У  
JOIN Н_ПЛАНЫ П ON У.ПЛАН_ИД = П.ИД
```

```
JOIN Н_ОТДЕЛЫ О ON П.ОТД_ИД = О.ИД  
) Студенты ON Л.ИД = Студенты.ЧЛВК_ИД  
WHERE Студенты.ЧЛВК_ИД IS NULL;
```