--craete and drop table

DROP TABLE IF EXISTS Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ CASCADE;

DROP TABLE IF EXISTS Н\_НАПР\_СПЕЦ CASCADE;

DROP TABLE IF EXISTS Н\_ФОРМЫ\_ОБУЧЕНИЯ CASCADE;

DROP TABLE IF EXISTS Н\_СЕССИЯ CASCADE;

DROP TABLE IF EXISTS Н\_ПЛАНЫ CASCADE;

DROP TABLE IF EXISTS Н\_ОТДЕЛЫ CASCADE;

DROP TABLE IF EXISTS Н\_УЧЕНИКИ CASCADE;

DROP TABLE IF EXISTS Н\_ЛЮДИ CASCADE;

DROP TABLE IF EXISTS Н\_ВЕДОМОСТИ CASCADE;

DROP TABLE IF EXISTS Н\_ТИПЫ\_ВЕДОМОСТЕЙ CASCADE;

CREATE TABLE Н\_ТИПЫ\_ВЕДОМОСТЕЙ (

ИД SERIAL PRIMARY KEY,

НАИМЕНОВАНИЕ VARCHAR(100) NOT NULL

);

CREATE TABLE Н\_ВЕДОМОСТИ (

ИД SERIAL PRIMARY KEY,

ТВ\_ИД INTEGER,

ЧЛВК\_ИД INTEGER,

ОЦЕНКА NUMERIC,

ДАТА DATE

);

CREATE TABLE Н\_ЛЮДИ (

ИД SERIAL PRIMARY KEY,

ФАМИЛИЯ VARCHAR(100),

ИМЯ VARCHAR(100),

ОТЧЕСТВО VARCHAR(100),

ДАТА\_РОЖДЕНИЯ DATE

);

CREATE TABLE Н\_УЧЕНИКИ (

ИД SERIAL PRIMARY KEY,

ЧЛВК\_ИД INTEGER,

ПЛАН\_ИД INTEGER,

ГРУППА VARCHAR(20),

НАЧАЛО DATE,

КУРС INTEGER,

ПРИЗНАК VARCHAR(50),

П\_ПРКОК\_ИД INTEGER,

СОСТОЯНИЕ VARCHAR(50)

);

CREATE TABLE Н\_ПЛАНЫ (

ИД SERIAL PRIMARY KEY,

ФО\_ИД INTEGER,

ОТД\_ИД INTEGER,

НАПС\_ИД INTEGER,

КУРС INTEGER

);

CREATE TABLE Н\_ОТДЕЛЫ (

ИД SERIAL PRIMARY KEY,

КОРОТКОЕ\_ИМЯ VARCHAR(100)

);

CREATE TABLE Н\_СЕССИЯ (

ИД SERIAL PRIMARY KEY,

ЧЛВК\_ИД INTEGER,

ДАТА DATE,

ОЦЕНКА NUMERIC

);

CREATE TABLE Н\_ФОРМЫ\_ОБУЧЕНИЯ (

ИД SERIAL PRIMARY KEY,

НАИМЕНОВАНИЕ VARCHAR(100)

);

CREATE TABLE Н\_НАПР\_СПЕЦ (

ИД SERIAL PRIMARY KEY,

КОД\_НАПРСПЕЦ VARCHAR(50)

);

CREATE TABLE Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ (

ИД SERIAL PRIMARY KEY,

НС\_ИД INTEGER

);

--insert into برای جدوال پایه

INSERT INTO Н\_ТИПЫ\_ВЕДОМОСТЕЙ (НАИМЕНОВАНИЕ)

VALUES ('Акт'),

('Ведомость');

INSERT INTO Н\_ЛЮДИ (ФАМИЛИЯ, ИМЯ, ОТЧЕСТВО, ДАТА\_РОЖДЕНИЯ)

VALUES ('Иванов', 'Иван', 'Иванович', '1990-01-15'),

('Петров', 'Петр', 'Петрович', '1995-07-20'),

('Сидоров', 'Сидор', 'Сидорович', '1985-03-10');

INSERT INTO Н\_ОТДЕЛЫ (КОРОТКОЕ\_ИМЯ)

VALUES ('ФКТИУ');

INSERT INTO Н\_ФОРМЫ\_ОБУЧЕНИЯ (НАИМЕНОВАНИЕ)

VALUES ('Очная'),

('Заочная');

INSERT INTO Н\_НАПР\_СПЕЦ (КОД\_НАПРСПЕЦ)

VALUES ('230101');

INSERT INTO Н\_НАПРАВЛЕНИЯ\_СПЕЦИАЛ (НС\_ИД)

VALUES (1);

--tests برای جداول وابسته

INSERT INTO Н\_ПЛАНЫ (ФО\_ИД, ОТД\_ИД, НАПС\_ИД, КУРС)

VALUES (1, 1, 1, 1), -- پلان برای دانشجویان حضوری

(1, 1, 1, 1); -- پلان دوم برای تست (مثلا برای گروه 1101)

INSERT INTO Н\_ВЕДОМОСТИ (ТВ\_ИД, ЧЛВК\_ИД, ОЦЕНКА, ДАТА)

VALUES (1, 1, 4.5, '2009-05-10'),

(1, 2, 5.0, '2008-03-15');

INSERT INTO Н\_СЕССИЯ (ЧЛВК\_ИД, ДАТА, ОЦЕНКА)

VALUES (1, '2012-01-25', 5);

INSERT INTO Н\_СЕССИЯ (ЧЛВК\_ИД, ДАТА, ОЦЕНКА)

VALUES (1, '2012-01-25', 5);

-- دانشجوی گروه 4100، حضوری (для запроса 5 و 6)

INSERT INTO Н\_УЧЕНИКИ (ЧЛВК\_ИД, ПЛАН\_ИД, ГРУППА, НАЧАЛО, КУРС, ПРИЗНАК, П\_ПРКОК\_ИД, СОСТОЯНИЕ)

VALUES (1, 1, '4100', '2012-09-10', 1, 'зачисл', 100, 'активен');

-- دانشجوی گروه 1101 (для вычисления максимальной оценки в группе 1101)

INSERT INTO Н\_УЧЕНИКИ (ЧЛВК\_ИД, ПЛАН\_ИД, ГРУППА, НАЧАЛО, КУРС, ПРИЗНАК, П\_ПРКОК\_ИД, СОСТОЯНИЕ)

VALUES (2, 2, '1101', '2012-09-05', 1, 'зачисл', 101, 'активен');

-- دانشجوی نمونه برای تست Query3 (سن بالای 25 سال؛ برای مثال, Сидоров با ЧЛВК\_ИД = 3، تولد در 1985)

INSERT INTO Н\_УЧЕНИКИ (ЧЛВК\_ИД, ПЛАН\_ИД, ГРУППА, НАЧАЛО, КУРС, ПРИЗНАК, П\_ПРКОК\_ИД, СОСТОЯНИЕ)

VALUES (3, 1, '3102', '2010-09-01', 1, 'зачисл', 102, 'активен');

--test2 for every query

SELECT \* FROM Н\_ТИПЫ\_ВЕДОМОСТЕЙ

WHERE НАИМЕНОВАНИЕ < 'Ведомость';

SELECT \* FROM Н\_СЕССИЯ

WHERE ОЦЕНКА = 5 AND ДАТА > '2012-01-01';

SELECT \* FROM Н\_ЛЮДИ

WHERE DATE\_PART('year', AGE(ДАТА\_РОЖДЕНИЯ)) > 25;

SELECT \* FROM Н\_ОТДЕЛЫ

WHERE КОРОТКОЕ\_ИМЯ = 'ФКТИУ';

SELECT \* FROM Н\_УЧЕНИКИ

WHERE ГРУППА = '4100'

AND ПЛАН\_ИД IN (

SELECT ПЛАН\_ИД FROM Н\_ПЛАНЫ

WHERE ФО\_ИД = 1

);

SELECT MAX(ОЦЕНКА) FROM Н\_ВЕДОМОСТИ

WHERE ЧЛВК\_ИД IN (

SELECT ЧЛВК\_ИД FROM Н\_УЧЕНИКИ

WHERE ГРУППА = '1101'

);

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

-- Таблицы: Н\_ТИПЫ\_ВЕДОМОСТЕЙ, Н\_ВЕДОМОСТИ.

-- Вывести атрибуты: Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД.

-- Фильтры (AND):

-- a) Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ < Ведомость.

-- b) Н\_ВЕДОМОСТИ.ИД < 39921.

-- Вид соединения: LEFT JOIN.

SELECT Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ, Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

FROM Н\_ТИПЫ\_ВЕДОМОСТЕЙ

LEFT JOIN Н\_ВЕДОМОСТИ ON Н\_ТИПЫ\_ВЕДОМОСТЕЙ.ИД = Н\_ВЕДОМОСТИ.ТВ\_ИД

WHERE

Н\_ТИПЫ\_ВЕДОМОСТЕЙ.НАИМЕНОВАНИЕ < 'Ведомость'

AND Н\_ВЕДОМОСТИ.ИД < 39921;

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

-- Таблицы: Н\_ЛЮДИ, Н\_ВЕДОМОСТИ, Н\_СЕССИЯ.

-- Вывести атрибуты: Н\_ЛЮДИ.ИД, Н\_ВЕДОМОСТИ.ИД, Н\_СЕССИЯ.ЧЛВК\_ИД.

-- Фильтры (AND):

-- a) Н\_ЛЮДИ.ИД = 152862.

-- b) Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 153285.

-- c) Н\_СЕССИЯ.ДАТА = 2012-01-25.

-- Вид соединения: INNER JOIN.

SELECT Н\_ЛЮДИ.ИД, Н\_ВЕДОМОСТИ.ИД, Н\_СЕССИЯ.ЧЛВК\_ИД

FROM Н\_ЛЮДИ

JOIN Н\_ВЕДОМОСТИ ON Н\_ЛЮДИ.ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

JOIN Н\_СЕССИЯ ON Н\_ЛЮДИ.ИД = Н\_СЕССИЯ.ЧЛВК\_ИД

WHERE Н\_ЛЮДИ.ИД = 152862

AND Н\_ВЕДОМОСТИ.ЧЛВК\_ИД < 153285

AND Н\_СЕССИЯ.ДАТА = '2012-01-25';

-- 3. Составить запрос, который ответит на вопрос, есть ли среди студентов ФКТИУ те, кто старше 25 лет.

SELECT

CASE

WHEN COUNT (\*) > 0 THEN 'Да, среди студентов ФКТИУ есть те, кому больше 25 лет'

ELSE 'Нет, среди студентов ФКТИУ нет тех, кому больше 25 лет'

END AS результат

FROM Н\_ЛЮДИ

JOIN Н\_УЧЕНИКИ ON Н\_ЛЮДИ.ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД

JOIN Н\_ПЛАНЫ ON Н\_УЧЕНИКИ.ПЛАН\_ИД = Н\_ПЛАНЫ.ИД

JOIN Н\_ОТДЕЛЫ ON Н\_ПЛАНЫ.ОТД\_ИД = Н\_ОТДЕЛЫ.ИД

WHERE Н\_ОТДЕЛЫ.КОРОТКОЕ\_ИМЯ = 'ФКТИУ'

AND EXTRACT (YEAR FROM AGE(Н\_ЛЮДИ.ДАТА\_РОЖДЕНИЯ)) > 25;

-- 4. Найти группы, в которых в 2011 году было ровно 5 обучающихся студентов на ФКТИУ.

-- Для реализации использовать подзапрос.

SELECT ГРУППА

FROM (

SELECT Н\_УЧЕНИКИ.ГРУППА,

COUNT(Н\_УЧЕНИКИ.ЧЛВК\_ИД) AS КОЛИЧЕСТВО

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\_ОТДЕЛЫ.КОРОТКОЕ\_ИМЯ = 'ФКТИУ'

AND EXTRACT (YEAR FROM Н\_УЧЕНИКИ.НАЧАЛО) = 2011

GROUP BY

Н\_УЧЕНИКИ.ГРУППА

) AS ГРУППЫ\_СТУДЕНТОВ

WHERE "КОЛИЧЕСТВО" = 5;

-- 5. Выведите таблицу со средними оценками студентов группы 4100 (Номер, ФИО, Ср\_оценка),

-- у которых средняя оценка не меньше максимальной оценк(е|и) в группе 1101.

SELECT

Н\_УЧЕНИКИ.ГРУППА AS Номер,Н\_ЛЮДИ.ФАМИЛИЯ || ' ' || Н\_ЛЮДИ.ИМЯ || ' ' || COALESCE(Н\_ЛЮДИ.ОТЧЕСТВО, '') AS ФИО,

AVG (Н\_ВЕДОМОСТИ.ОЦЕНКА ::numeric) AS Ср\_оценка

FROM Н\_УЧЕНИКИ

JOIN Н\_ЛЮДИ ON Н\_УЧЕНИКИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД

JOIN Н\_ВЕДОМОСТИ ON Н\_УЧЕНИКИ.ЧЛВК\_ИД = Н\_ВЕДОМОСТИ.ЧЛВК\_ИД

WHERE Н\_УЧЕНИКИ.ГРУППА = '4100'

GROUP BY Н\_УЧЕНИКИ.ГРУППА, Н\_ЛЮДИ.ФАМИЛИЯ, Н\_ЛЮДИ.ИМЯ, Н\_ЛЮДИ.ОТЧЕСТВО

HAVING

AVG (Н\_ВЕДОМОСТИ.ОЦЕНКА ::numeric) >= (

SELECT MAX (Н\_ВЕДОМОСТИ.ОЦЕНКА::numeric)

FROM Н\_ВЕДОМОСТИ

JOIN Н\_УЧЕНИКИ ON Н\_ВЕДОМОСТИ.ЧЛВК\_ИД = Н\_УЧЕНИКИ.ЧЛВК\_ИД

WHERE Н\_УЧЕНИКИ.ГРУППА = '1101'

);

-- 6. Получить список студентов, зачисленных после первого сентября 2012 года на первый курс очной формы обучения.

-- В результат включить:

-- номер группы;

-- номер, фамилию, имя и отчество студента;

-- номер и состояние пункта приказа;

SELECT

Н\_УЧЕНИКИ.ГРУППА,

Н\_ЛЮДИ.ИД,

Н\_ЛЮДИ.ФАМИЛИЯ,

Н\_ЛЮДИ.ИМЯ,

Н\_ЛЮДИ.ОТЧЕСТВО,

Н\_УЧЕНИКИ.П\_ПРКОК\_ИД,

Н\_УЧЕНИКИ.СОСТОЯНИЕ

FROM Н\_УЧЕНИКИ

JOIN Н\_ЛЮДИ ON Н\_УЧЕНИКИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД

JOIN Н\_ПЛАНЫ ON Н\_УЧЕНИКИ.ПЛАН\_ИД = Н\_ПЛАНЫ.ИД

JOIN Н\_ФОРМЫ\_ОБУЧЕНИЯ ON Н\_ПЛАНЫ.ФО\_ИД = Н\_ФОРМЫ\_ОБУЧЕНИЯ.ИД

WHERE Н\_УЧЕНИКИ.ПРИЗНАК = 'зачисл'

AND Н\_УЧЕНИКИ.НАЧАЛО > '2012-09-01'

AND Н\_УЧЕНИКИ.КУРС = '1'

AND Н\_ФОРМЫ\_ОБУЧЕНИЯ.НАИМЕНОВАНИЕ = 'Очная';

-- 7. Вывести список людей, не являющихся или не являвшихся студентами СПбГУ ИТМО

-- (данные, о которых отсутствуют в таблице Н\_УЧЕНИКИ). В запросе нельзя использовать DISTINCT.

SELECT

Н\_ЛЮДИ.ИД,

Н\_ЛЮДИ.ФАМИЛИЯ,

Н\_ЛЮДИ.ИМЯ,

Н\_ЛЮДИ.ОТЧЕСТВО

FROM Н\_ЛЮДИ

WHERE NOT EXISTS (

SELECT 1

FROM Н\_УЧЕНИКИ

WHERE Н\_УЧЕНИКИ.ЧЛВК\_ИД = Н\_ЛЮДИ.ИД

);