Assignment 2 (Problem 1).

1.

1) $\pi_{Pname}\{Presidents\}$

2) $\pi_{Jname}\{\sigma_{(LawSchool=Yale\ OR\ LawSchool=Harvard)}(Judges)\}$

3) Yale_Pres <= $\pi_{Pname}\{[\pi_{Jname}\{\sigma_{(LawSchool=Yale)}(Judges)\} * Appoints]$
   Harv_Pres <= $\pi_{Pname}\{[\pi_{Jname}\{\sigma_{(LawSchool=Harvard)}(Judges)\} * Appoints]$
   Multi <= $\rho_{n1(jnameY)}$(Yale_Pres) ***(cross)*** $\rho_{n2(jnameH)}$(Harv_Pres)
   Result <= $\pi_{JnameY}\{\sigma_{(jnameY=jnameH)}(Multi)\}$

   Note: another way to do this is to use intersection.

4) num_judges <= $\{_{law\_school}\mathrm{Э}_{count(jname)}(judges)\}$

5) pairs <= $\rho_{n1(jname1,\ lawschool1)}(\pi_{Jname,LawSchool}\{Judges\})$ **(cross*)*** $\rho_{n2(jname2,\ lawschool2)}(\pi_{Jname,LawSchool}\{Judges\})$
   result <= $\pi_{Jname1,JName2}\{\sigma_{(lawschool1=lawschool2)}(pairs)\}$

   Note: Make sure you use pairs with conditions that judges names are different but law school is the same.

6) j_not_yale <= $\pi_{Jname}\{\sigma_{(LawSchool!=Yale)}(Judges)\}$
   party_not_yale <= π_{Party}\{(j_not_yale * Appoints) * Presidents\}
   all_party <= $\pi_{Party}\{Presidents\}$
   party_only_yale <= all_party – party_not_yale

   Note: Key words here is only Yale. Therefore, you will need to subtract “not yale” from “all”

7) numjudge <= $\rho_{n1(pname,\ count)}(_{pname}\mathrm{Э}_{count(jname)}(appoints))$
   $\pi_{pname}\{\sigma_{(count=2)}(numjudge)\}$

   Note: When you use group functions, you may need to rename the columns in order to use it again.

8) all_pres <= $\pi_{pname}\{presidents\}$
   app_pres <= $\pi_{pname}\{appoints\}$
   no_appoints = all_pres – app_pres

   Note: You cannot use group function here and test the count is zero. Because if the count is zero, it won’t even appear in the table.

9) judge_count <= $\rho_{n1(pname,\ count)}(_{pname}\mathrm{Э}_{count(jname)}(appoints))$
   $\pi_{pname}\{\sigma_{(count>2)}(judge\_count)\}$

10) judges_republican <= $\pi_{\text{jname, jdateofbirth, lawschool}}$ {[$\sigma_{(\text{party='republican'})}$(presidents) * appoints] * judges}
judges_not_oldest <= $\pi_{\text{judge1.jname}}$ {ρ_{judge1}(judges_republican) ***(join)***$_{\textbf{(judge1.jdateofbirth > judge2.jdateofbirth AND judge1.lawschool = judge2.lawschool)}}$ ρ_{judge2}(judges_republican)}
judges_oldest_per_lawschool = π_{jname}{judges_republican} – judges_not_oldest

Note: Another way to handle this is to use aggregate function when finding the oldest judge, i.e., the one with the smallest birth date.