Zac Fotelt
CPSC 321:01
September 23 2021

HW#2

① a) $\pi_{maker}(PC \bowtie_{hd=256GB} Product)$

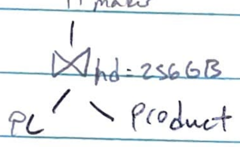
b) $\pi_{model, price}(\sigma_{maker=B}(Product \bowtie_{model} PC))$

c) $\pi_{maker}((Product \bowtie_{model} Laptop) \cap (Product \bowtie_{model} PC))$

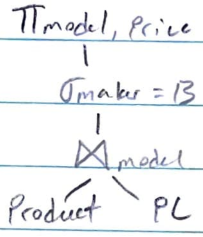
d) $\pi_{hd}(PC \bowtie_{hd=hd} PC)$

e) $\pi_{maker}(\sigma_{speed>4.2GHz}(Product \bowtie_{model} PC))$

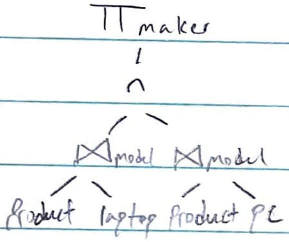
② ⓐ



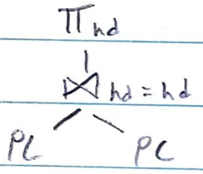
ⓑ



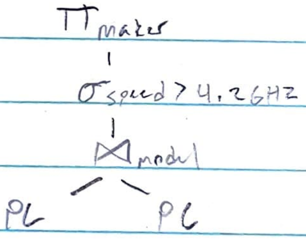
ⓒ



ⓓ



ⓔ



③ a) min = n+m, max = n+m

b) min = n, max = n+m

c) min = n, max = n+m

d) min = n+m, max = n+m

e) min = 0; max = $(n - len(\pi_A(R)) + m$

4

a) $\Pi_{\text{item-number, name}}(\sigma_{\text{part-list} > 10 \wedge \text{price} < 25}(\text{Sets}))$

b) $\Pi_{\text{item-number, name}}((\sigma_{\text{part-list.name} = \text{1x2 brick} \wedge \text{color} = \text{brightblue}}(\text{Sets}))$
$\bowtie (\sigma_{\text{part-list.name} = \text{2x4 plate} \wedge \text{color} = \text{bright red}}(\text{sets})))$

c) $\Pi_{\text{item\_number, name}}(\text{Sets} \bowtie_{\text{theme} = \text{Disney} \wedge \text{category} = \text{Building}}(\text{Sets}))$

d) $\Pi_{\text{item-number, name}}((\sigma_{\text{category} = \text{Building}}(\text{Sets})) \bowtie_{\text{eid}} (\sigma_{\text{category} = \text{Sports}}(\text{Sets})))$

e) $\Pi_{\text{eid, name, price}}(\text{Bricks} \bowtie_{\text{design\_id} \,!= \,\text{design\_id} \wedge \text{elem\_id} = \text{elem\_id} \wedge \text{price} < 0.20} \text{Bricks})$.

I needed to structure my Sets table more around the part list attribute to have access to those attributes

1) I. will be modifying an existing project I have been working on for some time, It will be an expense calculator written in Python that allows a user to track their expenses over a period of time. The current Database component only stores single transactions, tracks a user's current balance, & output a history of the user's transactions. I would like to add more features to flesh out this app. Giving users the ability to tag transactions for easy review would be beneficial. Also providing the users w/ visualizations & trend graphs would be extremely useful. Designing a simple & effective Gui to allow users to navigate the functions of the application is a important feature to add as well. Finally, I believe it would be of great benefit to the user to allow entry of multiple transactions at the same time.

2) For this version of the proposal, I decided on 2 new features to describe. First, implementing a GUI is paramount because it should be very clear and easy for a user to interact with the application properly. Additionally, it would be helpful for users to have a feature to enter transactions in bulk in case they are behind on entries.

③. User tasks in application

- The user will login using their credentials when prompted by the login screen. This will retrieve their associated data (OLTP)
- The user can enter a transaction amount, chose deposit or withdrawl, describe their transaction, and select a transaction tag. Once they click a submit button, the user's transaction will be processed & inserted into the database (OLTP)
- The user can click a history button to have their transactions compiled & displayed to them. This will likely open a new page to the users to allow them to filter out their results (OLAP)
- The user's homepage will display visualizations of their data in the form of trend graphs & other forms (OLAP)
- The user will have the capability in the history screen to delete or modify past entries. Ideally this will be accomplished with a button & a pop-up window (OLTP)