

CSC343 Worksheet 2 Solution

June 11, 2020

1. Exercise 2.4.1:

a) $\sigma_{speed \geq 3.0}(\text{Movies})$

Models 1005, 1006, 1013 have speed greater than 3.0

<i>model</i>	<i>speed</i>	<i>ram</i>	<i>hd</i>	<i>price</i>
1001	2.66	1024	250	2114
1002	2.10	512	250	995
1003	1.42	512	80	478
1004	2.80	1024	250	649
→ 1005	3.20	512	250	630
→ 1006	3.20	1024	320	1049
1007	2.20	1024	200	510
1008	2.20	2048	250	770
1009	2.00	1024	250	650
1010	2.80	2048	300	770
1011	1.86	2048	160	959
1012	2.80	1024	160	649
→ 1013	3.06	512	80	529

Notes:

- Select
 - Is indicated by σ
 - **Syntax:** $\sigma_{\text{QUERY}}(\text{SCHEMA_NAME})$
 - e.g. $\sigma_{length \geq 100 \text{ AND } studioName = 'Fox'}(\text{Movies})$

Relation - Movies

<i>title</i>	<i>year</i>	<i>length</i>	<i>inColor</i>	<i>studioName</i>	<i>producerC#</i>
Star Wars	1977	124	sciFi	Fox	12345
Galaxy Quest	1999	104	comedy	DreamWorks	67890

b) Notes:

- Project
 - **Syntax:** $\pi_{A_1, A_2, \dots, A_n}(\text{Rel})$
 - * A_1, \dots, A_n represents attributes
 - Picks certain columns
 - e.g.

What are the titles and years of movies made by Fox that are at least 100 minutes long?

$$\pi_{\text{title}, \text{year}}(\sigma_{\text{length} \geq 100 \text{ AND } \text{studioName} = \text{'Fox'}})(\text{Movies})$$

- Cross-Product / Cartesian Product
 - Combines two relations
 - **Syntax:** Relation 1 \times Relation 2
 - e.g. Names and GPAs of students with $HS > 1000$ who applied to CS and were rejected

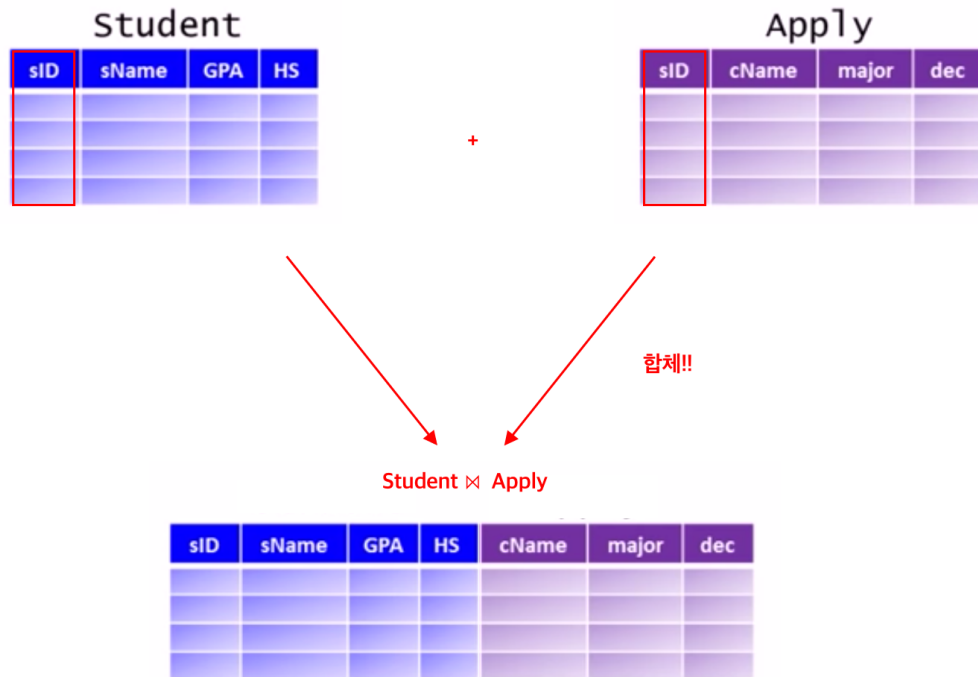
$$\pi_{sName, GPA}(\sigma_{\text{Student.sID} = \text{Apply.sID} \text{ AND } HS > 1000 \text{ AND } \text{major} = \text{'cs'} \text{ AND } \text{dec} = \text{'R'}})(\text{Student} \times \text{Apply})$$

College			Student				Apply			
cName	state	enr	sID	sName	GPA	HS	sID	cName	major	dec

- Natural Join
 - Enforce equality on all attributes with the same name
 - Eliminate one copy of duplicate attributes
 - Is symbolized by \bowtie
 - **Syntax:** Relation 1 \bowtie Relation 2
 - e.g.

Names and GPAs of students with $HS > 1000$ who applied to CS and were rejected.

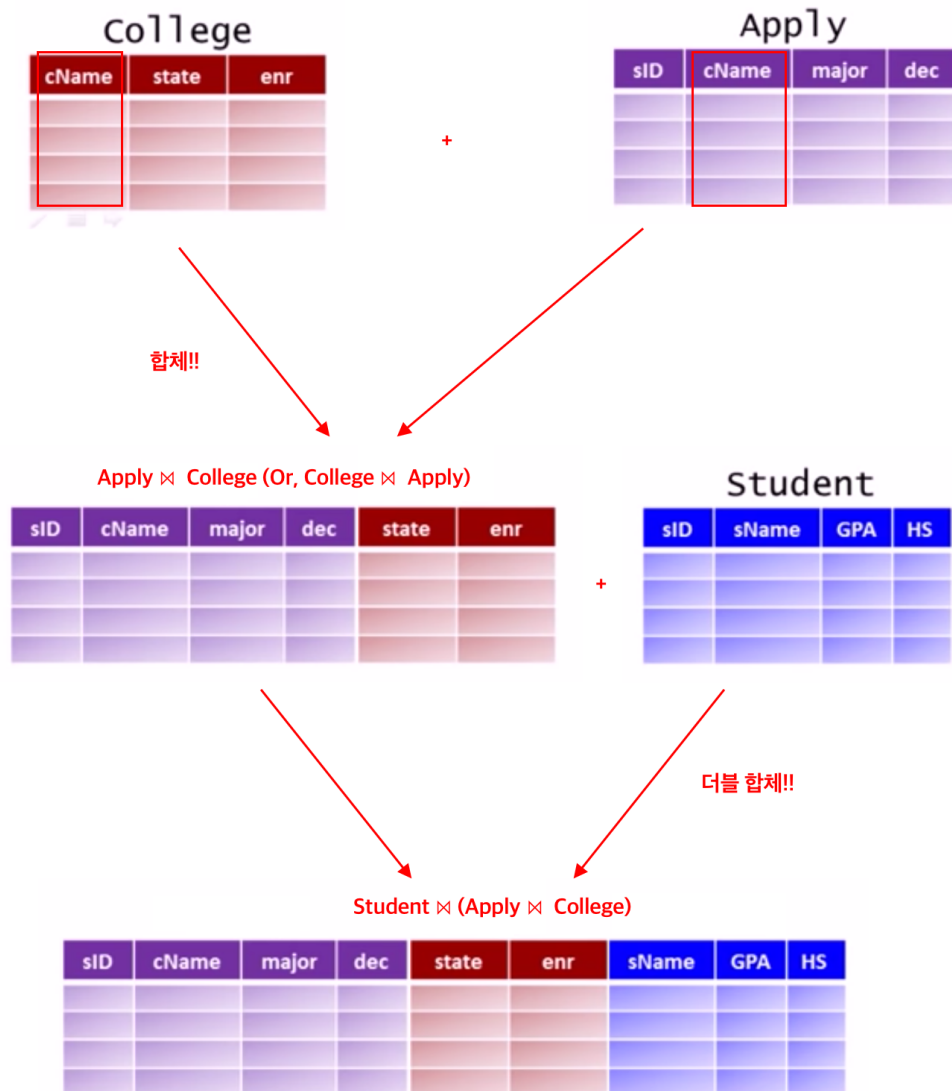
$$\pi_{sName, GPA}(\sigma_{HS > 1000 \text{ AND } \text{major} = \text{'cs'} \text{ AND } \text{dec} = \text{'R'}}(\text{Student} \bowtie \text{Apply}))$$



– e.g.2.

Names and GPAs of students with $HS > 1000$ who applied to CS at college with $enr > 20,000$ and were rejected

$$\pi_{sName, GPA} \left(\sigma_{HS > 1000 \text{ AND } enr > 20000 \text{ AND } major = 'cs' \text{ AND } dec = 'R'} (Student \bowtie (Apply \bowtie College)) \right)$$



- Union Operator
 - **Syntax** $R \cup S$
 - Is the set of elements that are in R or S or both.
 - An element appears only once in the union even if it is present in both R and S .
 - Is like **UNION** keyword in SQL
 - e.g.

List of college and student names

$$\pi_{cName}(\text{College}) \cup \pi_{sName}(\text{Student})$$

- Difference Operator

- **Syntax:** $R - S$
- Is also called the *difference* of R and S
- is the set of elements that are in R but not in S .
- Is like **EXCEPT** keyword in SQL
- e.g.

IDs and names of students who didn't apply anywhere

$$\pi_{sID}(\text{Student}) - \pi_{sID}(\text{Apply})$$

- Intersection Operator
 - **Syntax:** $R \cap S$
 - Is also called the *intersection* of R and S
 - Is the set of elements that are in both R and S
 - e.g.

Names that are both a college name and a student name

$$\pi_{cName}(\text{College}) - \pi_{sName}(\text{Student})$$