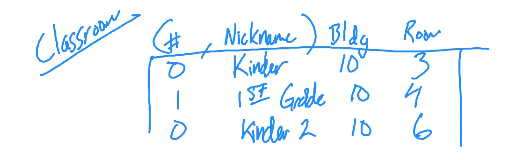
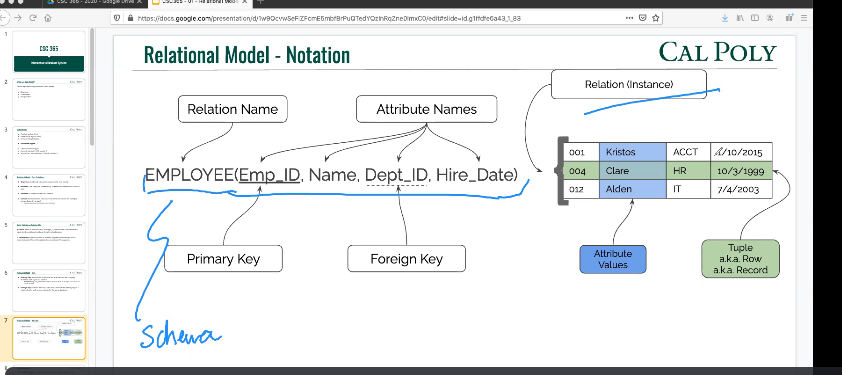
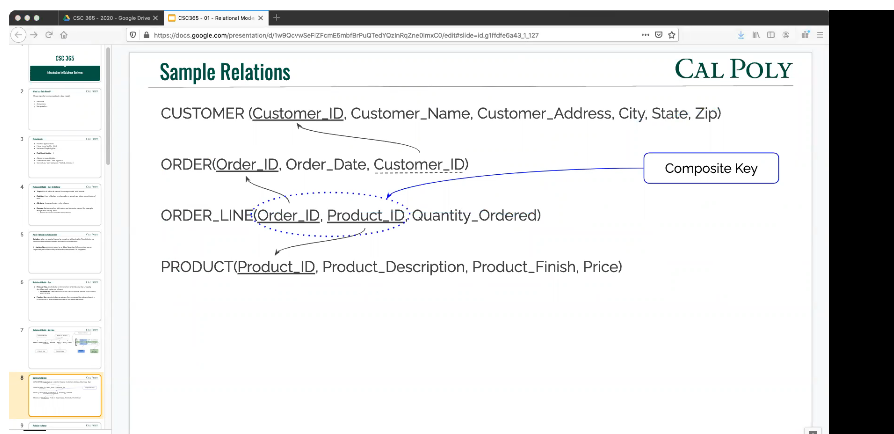
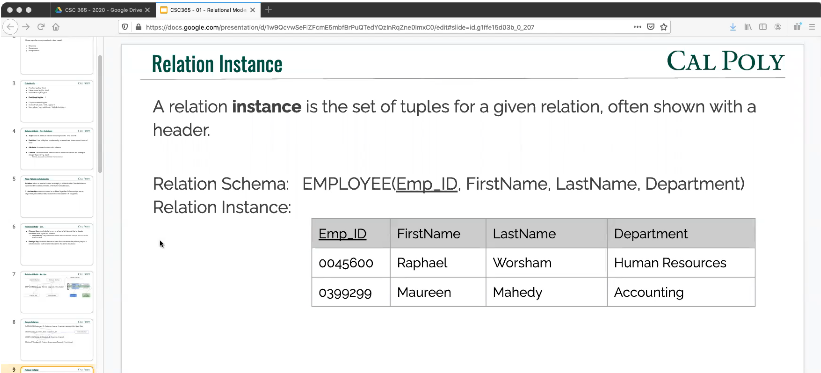
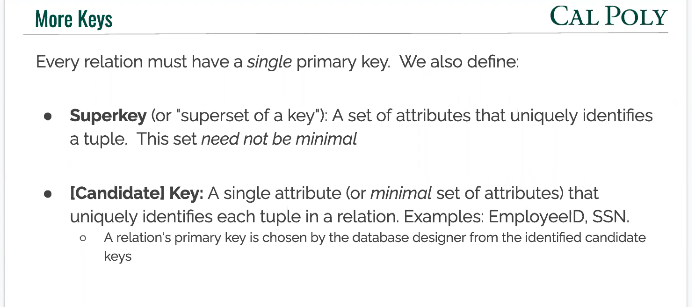
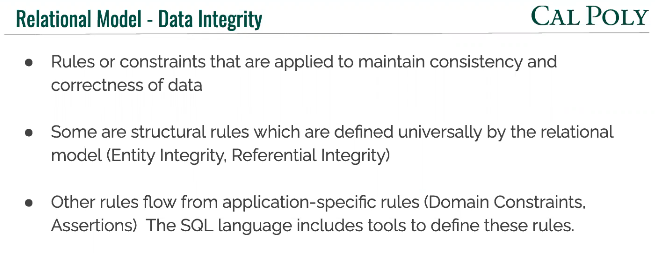
* Data model:
  + Structure
  + Constraints
  + Manipulation
* Our data model (1970’s): RELATIONAL – most popular.
  + Other models:
    - Hierarchical: harder to program since got to traverse complex tree.
    - Network/Graph: graphs duh.
    - Semi-structured / XML (1990’s).
    - Key-value / non relational / NoSQL (2000’s).
* Relational Model
  + Tuple: List of attribute values.
  + Relation: A set of tuples, or informally, a named two-dimensional table of data.
  + Attribute: A named column of a relation.
  + Domain: Data type of an attribute must be scalar-valued: float, integer, string, data.
* Relation refers to set of tuples (table of data).
* Relationship – connects table (or entities together).
* Primary Key – An attribute that uniquely identifies each tuple in a relation.
  + EX: Classroom
    - Number #0, building 10, room 3
    - The key could be the number.
    - Tuple would be a list of these keys.
    - 
* Composite Key – a key that consist of more than one attribute.
  + EX: suit and value in a deck of cards.
* Foreign Key: An attribute in a relation that serves as the primary key of a relation in the same database.
* 
* 
* Instance – a relation instance is the set of tuples for a given relation, often shown with a header.
* 
* Attributes names in a relation schema are a **set**, not a list
* Relations themselves are sets of rows (tuples) not lists.

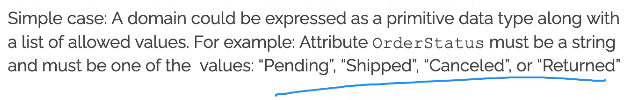
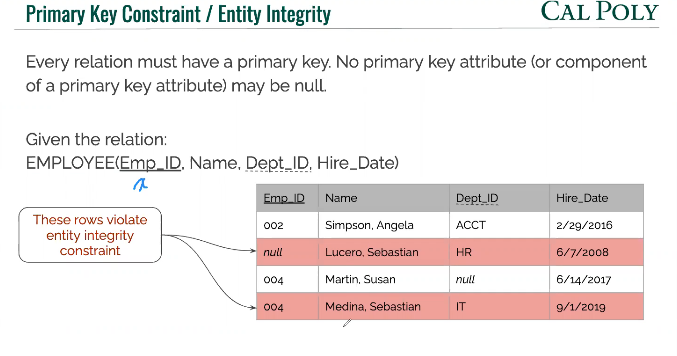


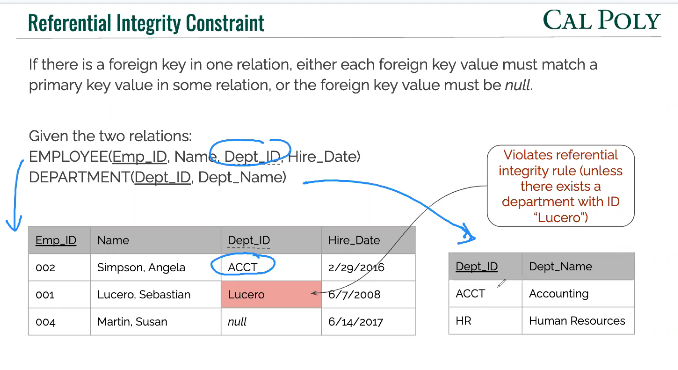


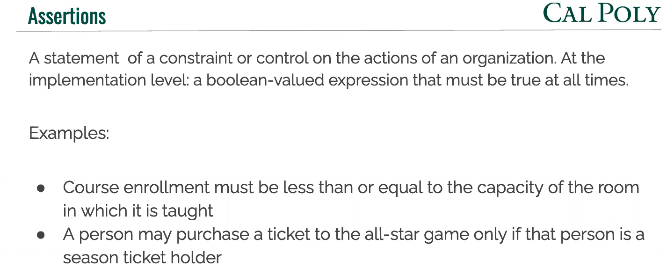
* Combination of all 5 could be a SUPERKEY.
* Movietitle, year, starname could be Candidate keys and primary key.

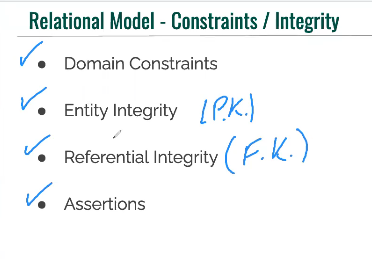
Contrstraints:

A classroom must have at least 1 teacher…

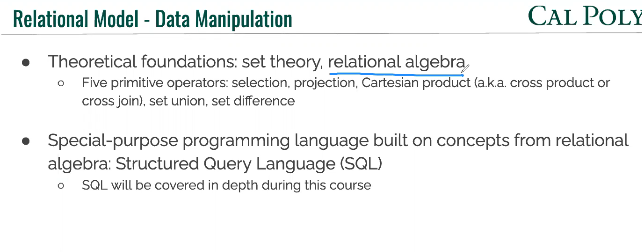
* Domain Constraints – All the values which a data element may contain (its structure).
* All values in a column must be taken from the same domain.
  + So class data table must contain class structure stuff.
* 
* Primary key cant be null or appear twice!
* 



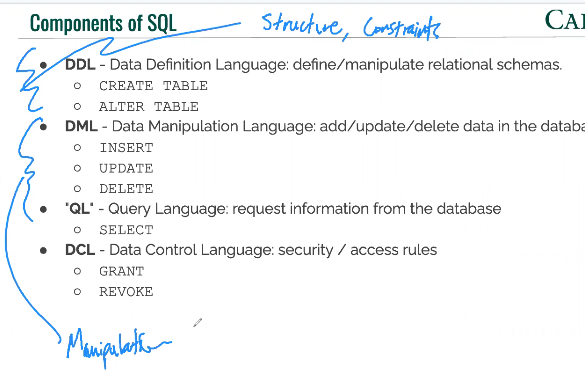


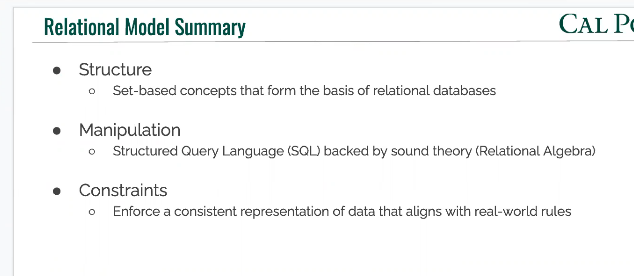


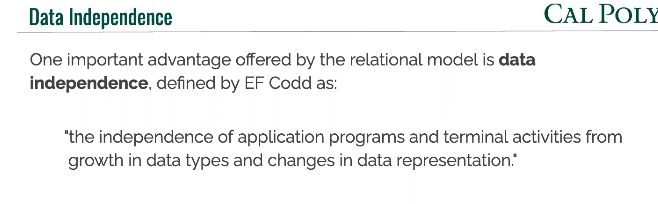
Manipulation = set manipulation!!!!



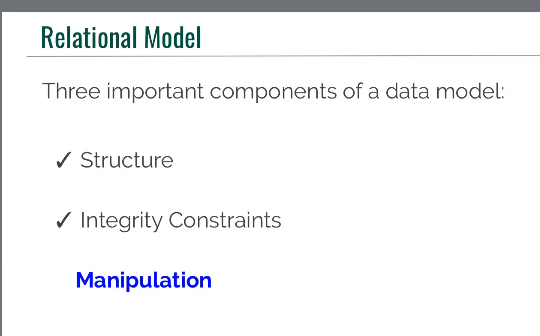
We using SQL.

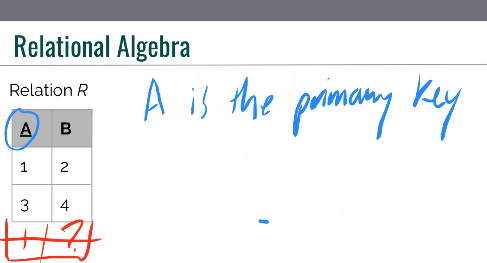


 THE RELATIONAL MODEL SUMMARY.

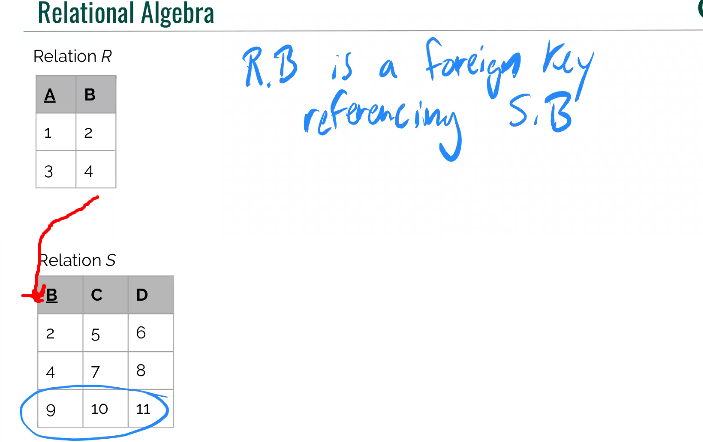


Lecture 2



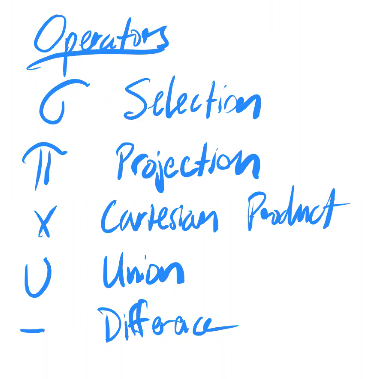


B in relation to R.

­­­

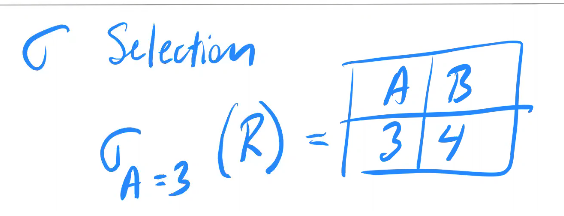
THE LANGUAGE:

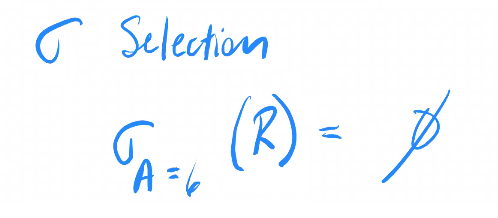
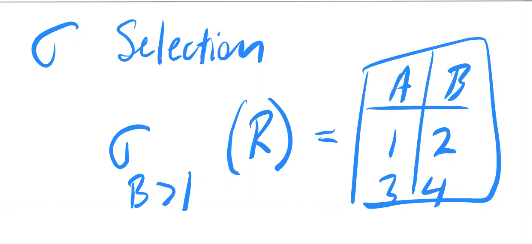
Operators.



SELECTION:

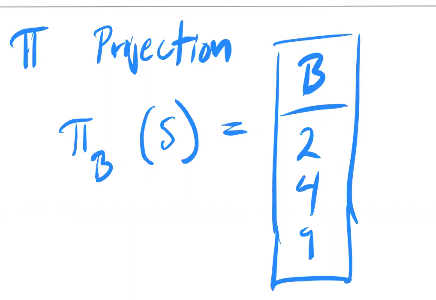
Takes single relation as its input and returns a subset of tuples.

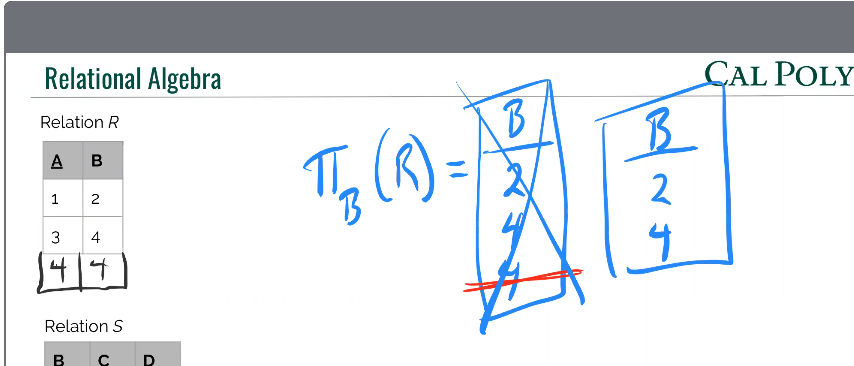


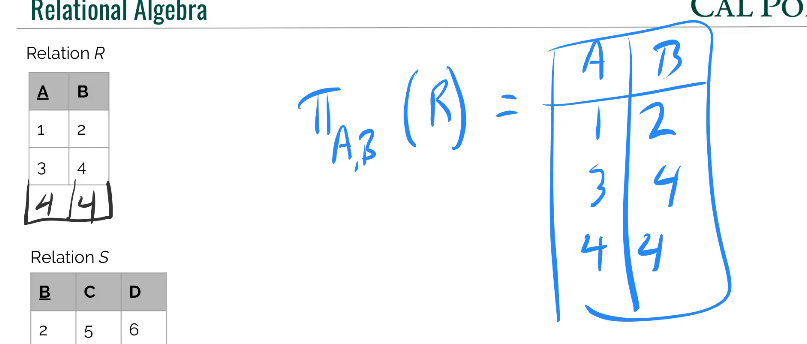


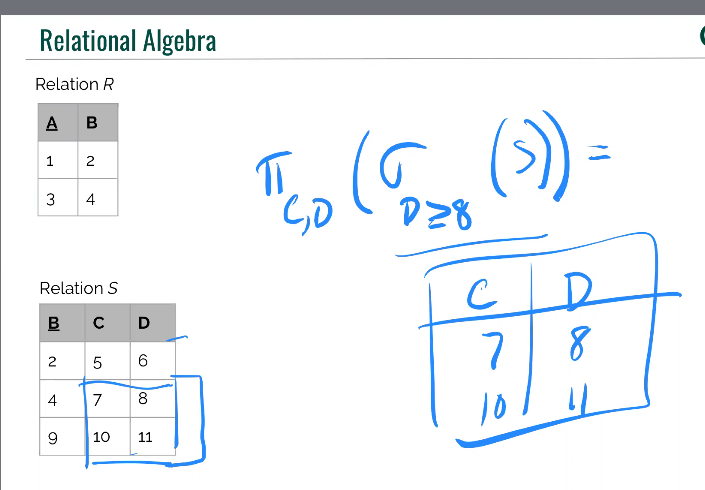
PROJECTION

Bring up certain columns / attributes. Takes list of attribute names.

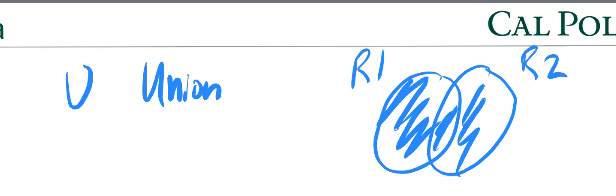
cardinality is 3 here.

ONLY Gets unique so no duplicates!

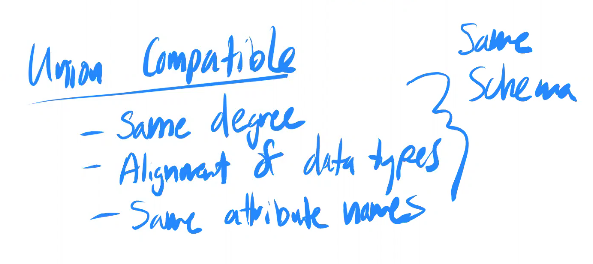


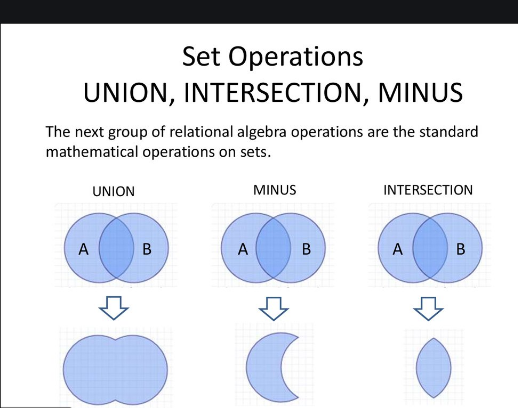
 Primary key here? Would be the composite key of c and d.

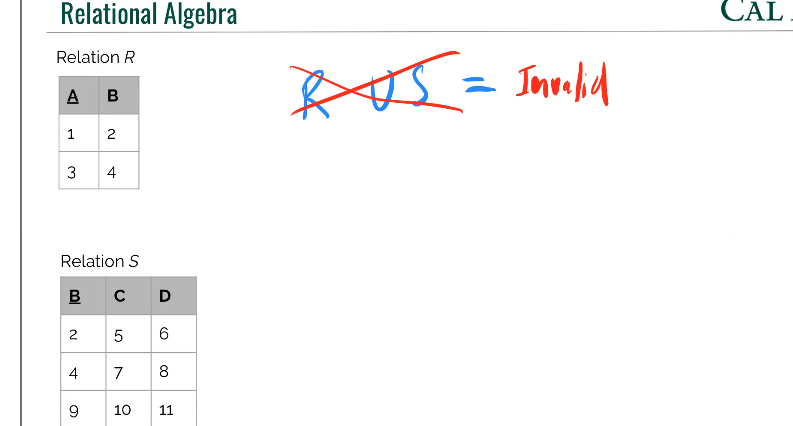
UNION



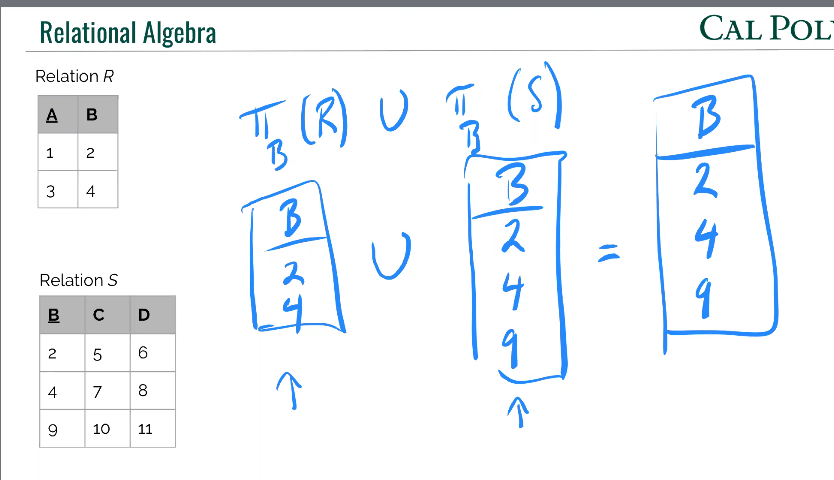
Union Compatible = same degree, alignment of data types, same attribute names.



side note

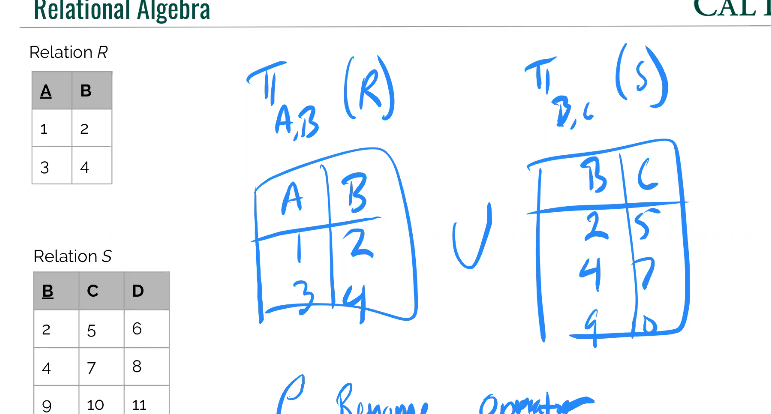


Solution? Use projection to it XD.

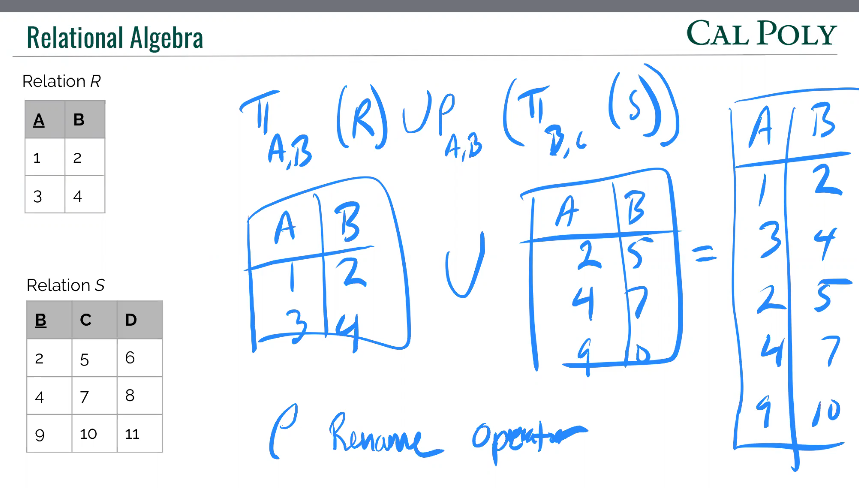


Use utility operator to change naming to create perfect schema matching.

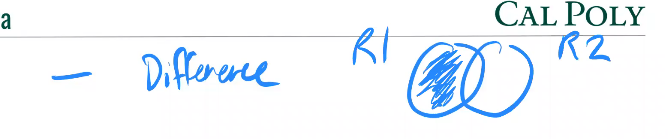
USE THE RENAME OPERATOR.

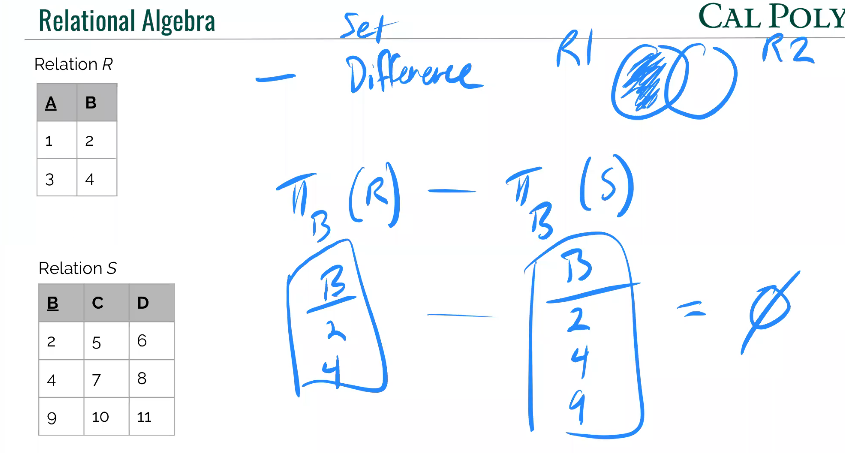


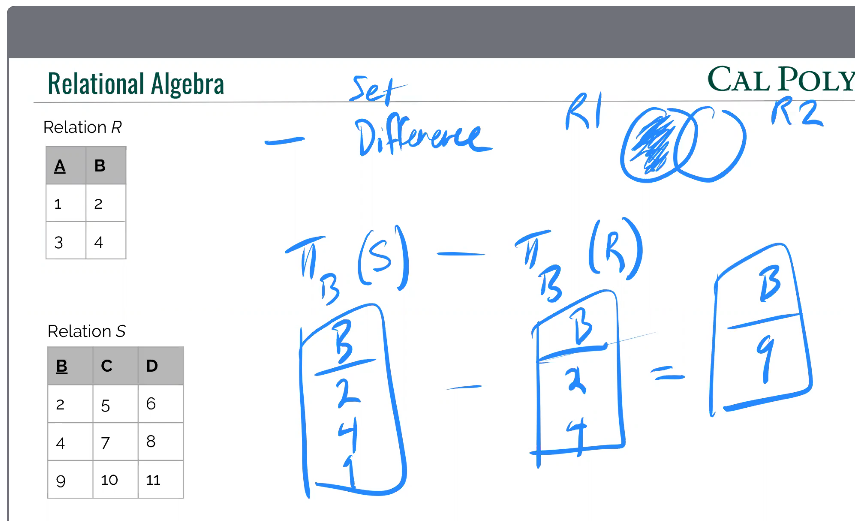


see that additional middle looking P thing. It lets you rename shit so the data becomes same to apply union on.

DIFFERENCE.

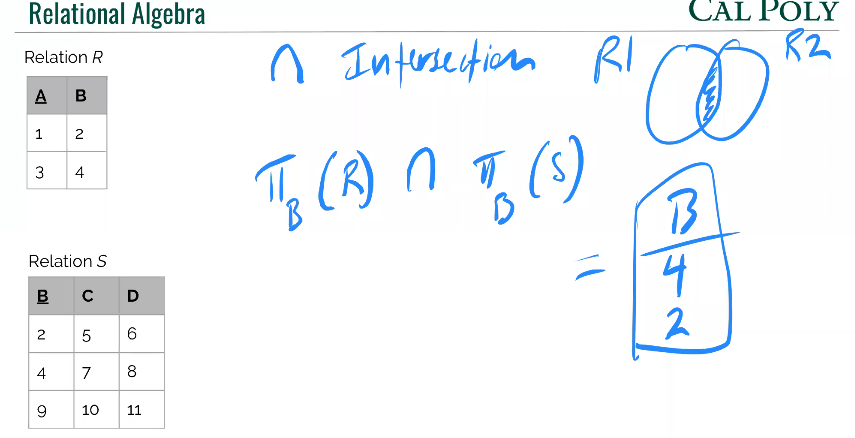




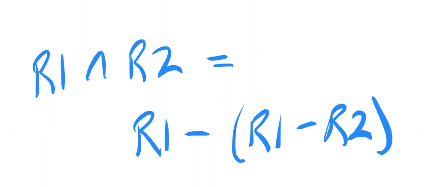


INTERSECTION





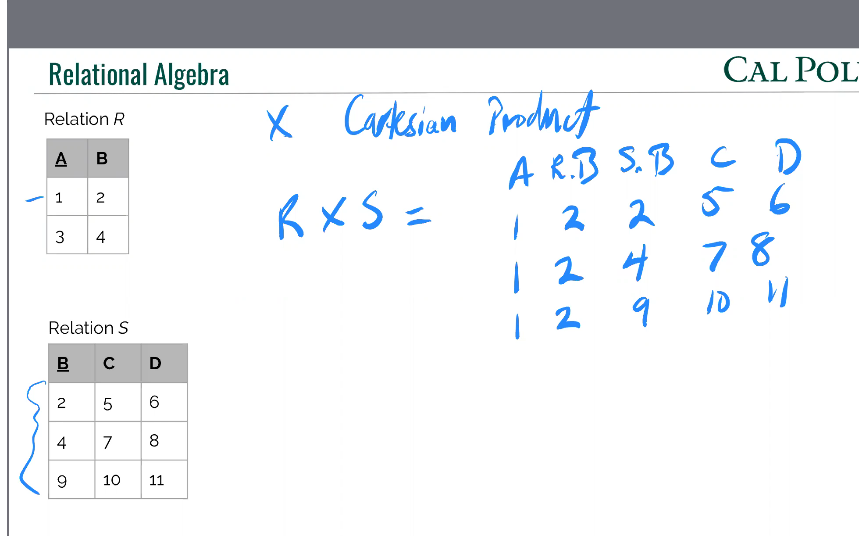
NOTE:

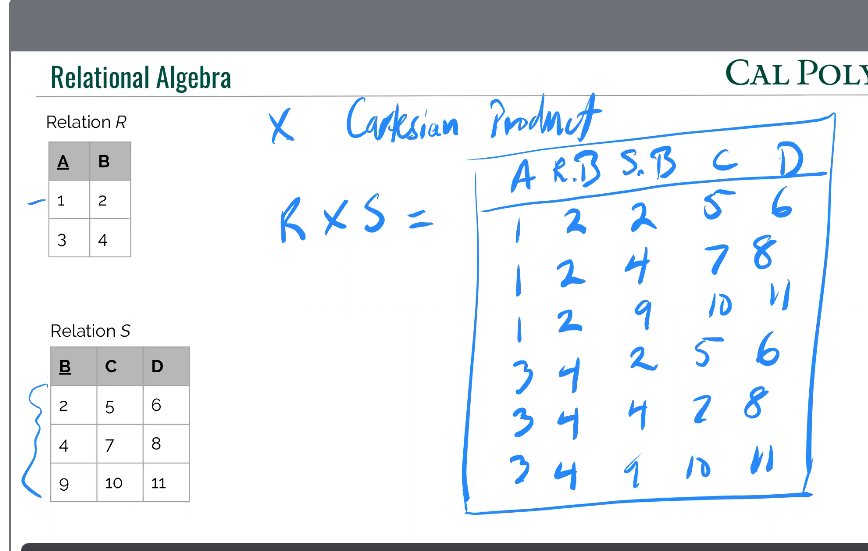


CARTESIAN PRODUCT

No compatibility requirement!

2 steps:





Degree 5, cardinality 6.