Oregon State University

COLLEGE OF ENGINEERING | School of Electrical Engineering and Computer Science

CS 340 Intro to DATABASES

Relational Algebra Operations
Part 3

Some material from Harrington, Jan L..
Relational Database Design and Implementation: Clearly Explained, 2016.

Some of this material comes from the course textbook:

Chapter 6 on Relational Algebra

which you can read for free from the OSU Library.

Harrington, Jan L.. *Relational Database Design and Implementation : Clearly Explained*, Elsevier Science & Technology, 2016. *ProQuest Ebook Central*, http://ebookcentral.proquest.com/lib/osu/detail.action?docID=4509772.

There is also a companion **Exploration RelaX RA Simulator activity** for students to complete in Canvas

Database systems are software programs that have to follow rules. The rules that define how a database will behave are a narrative model with mathematical symbols.

The relational algebra:
    Defines theoretical way of manipulating table contents using relational operators. Use of relational algebra operators on existing relations produces new relations.
    In short, these operators are designed to do the most common things that we need to do with relations in a database.

SELECT
UNION
PROJECT
DIFFERENCE
JOIN
PRODUCT
INTERSECT
DIVIDE
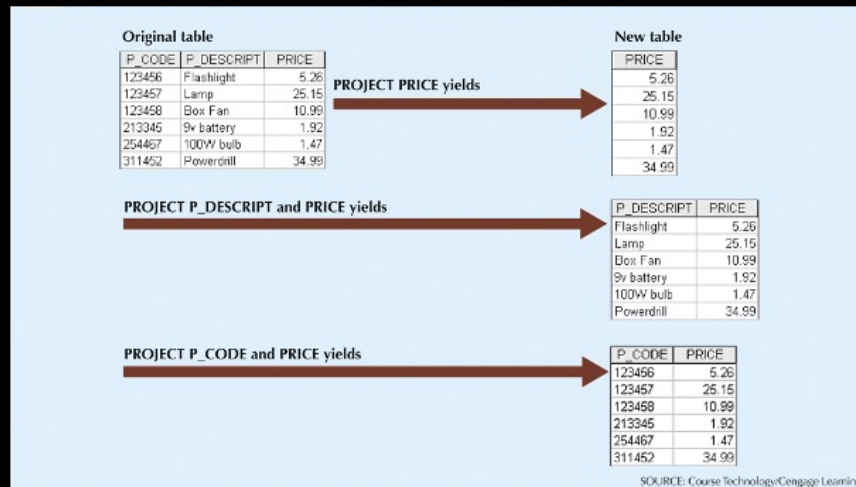
Project returns the attributes. Project does not limit the number of rows that are returned. We use the
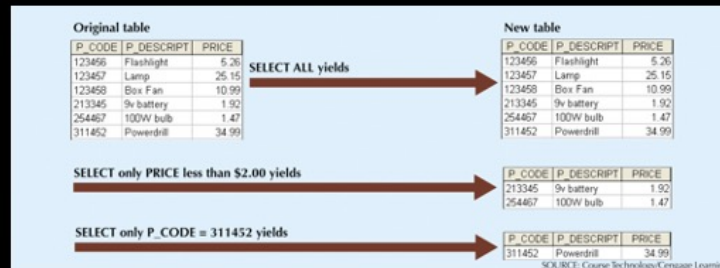
A Project is denoted by the letter pi

π p_code, price (parts)

A SELECT can be added to project to specify the columns and conditions desired.

π p_code, price (σ < 2000 (parts)

SELECT is used to obtain a subset of the tuples (attributes) of a relation that satisfy a *select condition*

Select give us a way to filter on a condition. We use the sigma operator for RA expressions

σ p_code<2000 (parts)

OR

SELECT all parts that are less than $2000

Union combines all rows from two tables and excludes duplicates. In order for two relations to be union-compatible or *unifiable*, both must have the same number of attributes (columns) and corresponding attributes (columns) must have the same set of allowable values for that attribute (e.g. integer or character).

Intersect yields only the rows that appear in both tables, and the tables also have to be unifiable.

The ∪ symbols is used for union and the ∩ symbol for intersection

Difference yields all rows in one table that are not found in the other table, or to put it another way, it subtracts one table from the other.

The Cartesian Product yields all possible pairs of rows from two tables. So if one table has 3 rows and the other table has 6 rows, the resulting set is 3 x 6 = 18 rows.

The – symbol is used for subtraction and the × for product

JOIN is performed when data are retrieved from more than one table at a time. The ⋈ symbol is used for join. There are three types of Join:

Natural join

Links tables by selecting rows with common values in common attributes (join columns). Natural join no need to specify condition. If there is no condition specifies then it returns the rows based on the common column

π movies.name, movies_genres.genre, movies.rank (movies ⋈ movies_genres)
Here is the corresponding SQL
```
select movies.name, movies_genres.genre, movies.rank
from movies natural join movies_genres
```

Inner join

Only returns matched records from the tables that are being joined. Inner Join joins two table on the basis of the column which is explicitly specified. In practice inner Join is used most times we need to write a Join query.

π movies.name, movies_genres.genre, movies.rank (movies ⋈ movies.id = movies_genres.movie_id movies_genres)
Here is the corresponding SQL
select movies.name, movies_genres.genre, movies.rank

from movies inner join movies_genres on movies.id = movies_genres.movie_id

Outer join

Matched pairs are retained, and any unmatched values in other table are left null

π movies.name, movies_genres.genre (movies $\bowtie$ movies.id = movies_genres.movie_id movies_genres)

Heres the corresponding SQL:
SELECT movies.name, movies_genres.genre FROM movies FULL OUTER JOIN movies_genres ON movies.id=movies_genres.movie_id;

Relational Algebra Summary

- The rules of RA help databases be compatible with one another.
  - RA is the basis for the SQL language.

- You will get to practice making RA expressions
  - But for the more complex queries we will use SQL

- Relax is a web RA calculator that you can use to practice RA expressions

8

To summarize, we spent time talking about relational algebra because it is the formal set of rules that define how a database should work. The fact that all relational databases follow these rules a tremendous benefit. It simplifies transferring data between different software vendor versions of DBs.

The rules of RA help databases be compatible with one another.
         RA is the formal definition for the SQL language.

We will use Relax to get some practice formulating simple RA expressions.
https://dbis-uibk.github.io/relax/

And more complex queries will be covered in SQL.