WEEK 4

INTRODUCTION TO QUERIES AND RELATIONAL LANGUAGES

STUDENT OBJECTIVES

- Upon completion of this video, you should be able to:
  - Given some tables, give examples of queries that you would ask of the tables.
  - Distinguish between procedural languages and non procedural languages
  - Determine, given operators for a data type, if the operators enforce closure.
  - List the 5 basic operators for relational algebra and determine if each of those operations works on one table (unary) or two tables (binary).

CONSIDER…

Quries .

- Fred needs to know how many people worked more than 40 hours on a project
- Sue needs to know who works in the Safety Department
- Homer wants to know how many supervising employees make over 2000 dollars

CONSIDER…

- Fred needs to know
  more than 40 hour
- Sue needs to know
  Department
- Homer wants to kn
  employees make o

Project : Table

ProjectName	ProjectNumber	ProjectLocation	DeptNumb
Accounting Upd	A1	Toronto	S7G
Acc3	A3	Springfield	G8H
Acct6	A6	Toronto	S7G
Inventory	I1	Toronto	G8H
Inventory2	I2	London	S7G
Payroll	P1	Springfield	G8H
Payroll2	P2	London	G8H
Payroll3	P3	London	G8H

Works_On : Table

SSN	ProjectNumb	Hours
1	A3	45
2	A3	56
3	A1	3
3	A6	45
3	I1	43
3	P1	9
4	A1	6
4	A3	5
4	A6	6
4	I1	43
4	I2	8
4	P1	67
4	P2	77
4	P3	67
6	I2	6
12	A3	56

Record: 1

Department : Table

DeptNumbe	DeptName	ManagerSSN	ManagerStartDate
G8H	Head Office	4	2/2/9
S7G	Safety Department	3	1/1/9
Y5J	Research Department	6	3/3/9

Employee : Table

SSN	LastName	MiddleInitia	FirstName	BDate	Address	Sex	Salary	SuperSSN	DeptNumb
1	Simpson	P	Bart	2/2/95	London	M	$1,000.00	2	G8H
2	Smithers	J	Waylan	1/1/60	Springfie	M	$2,000.00	4	S7G
3	Beuvieau	P	Patty	3/3/59	Toronto	F	$4,000.00	6	Y5J
4	Burns	P	Montgomer	7/7/20	Toronto	M	$5,000.00		S7G
6	Simpson	J	Lisa	6/6/90	London	F	$1,000.00	2	S7G
12	Simpson	J	Homer	8/8/61	Toronto	M	$2,000.00	2	G8H

Record: 1 of 6

RELATIONAL LANGUAGES

- Once we have our relational model, we need to manipulate the data within the model, we use a relational language
- Some relational languages are **Procedural**, they tell us how to get the data (e.g. **Relational Algebra**)
- Some relational languages are **Non-Procedural**, they tell us what data is needed (e.g. **Relational Calculus**)

implement in SQL

- Formally, Relational Algebra is equivalent to Relational Calculus, i.e. every expression in the algebra can be written also in the calculus and vise versa

- A language that can produce any relation that can be derived using relational calculus is relationally complete.

- Most relational query languages are relationally complete and more (i.e. they have additional power to do calculations, ordering, etc.)

- Relational algebra is a theoretical language with ***operations*** that perform on one or more relations and in turn produce relations based on the operations, thus both the input (**operands**) and output (results) are relations, i.e. a ***closed language***, i.e. integer - integer produces an integer and a relation - relation produces a relation

input => output

EXAMPLE OF CLOSURE WITH RELATIONS

- Pretend that the symbol  represents an operation for the operand tables (i.e. relations) (just like + represents an operation with the operands integers)
- Then would be a closed operation if and only if:

table

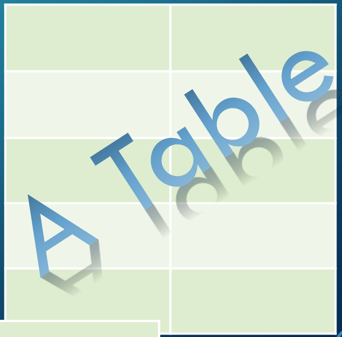
34	Pig	Red
445	Horse	Red
34	Cat	Blue

table

33	X	Mar	2018	London	22.2
44	Y	Jun	1964	Toronto	45.1
55	X	Feb	1982	Windsor	23.8
22	B	Jan	1977	Arva	0.1

=

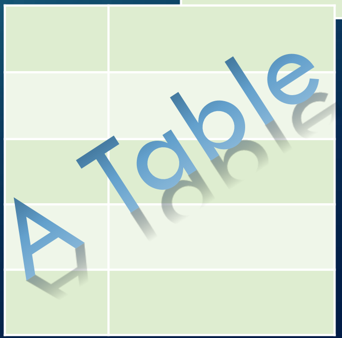
table.



- In this case is a binary operation (2 operands).
- Could be unary operation like this:

34	Pig	Red
445	Horse	Red
34	Cat	Blue

=



QUESTION: Do the operations -,+,/,* produce a closed language on integers? YES or NO? WHY?

Yes NO

NO! Because of / → Division e.g. 9 / 2

it produce a float, which is not int.

- Relational algebra is a set language, in which all tuples are manipulated in one statement, thus we don't use looping
- Because relations are produced, we can use the results in further operations, thus nesting our results, this is called **closure**; relations are closed under relational algebra

We can have the output from one expression as the input for the other expr.
This help in query refinement.

- **QUESTION: What are unary operations with integers that insure closure?**
- **ANSWER: Power, Absolute Value…eg. 77^2 or abs(77)**

- **QUESTION: What are unary operations with integers that do not allow for closure?**
- **ANSWER: Square Root…eg. $\sqrt{77}$**

- **QUESTION: What are some unary operations when working with bits?**
- **ANSWER: FLIP…eg FLIP (1011) = 0100**

BASIC OPERATIONS CAN BE USED TO BUILD OTHER MORE COMPLICATED OPERATIONS

- Eg. For integers, in order to create exponents, we could use the basic operation of multiplication:

3^4 is the same as 3 * 3 * 3 * 3

so * is a basic operation in arithmetic.

5 BASIC OPERATIONS IN RELATIONAL ALGEBRA

- **Selection:** Unary (works on 1 Relation (TABLE) only), returns only the tuples from a relation that satisfy the specified condition. (i.e. returns a row subset), written as:

$$\sigma_{\text{condition}} (R)$$

- **Projection:** Unary (works on 1 Relation only), returns only the requested attributes (with no duplicates) (returns a column subset), written as:

$$\pi_{\text{attribute1, attribute2}} (R)$$

- **Cartesian Product:** Binary (requires 2 Relations), returns a relation that is the concatenation of every tuple of relation R with every tuple of relation S, Symbol: S **X** R

- **Union:** Binary (requires 2 Relations), union of relations R and S with I and J tuples, respectively, is the concatenation of them into one relation with a maximum of I+J tuples, duplicate tuples being eliminated, R and S must be union compatible (i.e. R and S must have the same columns or attribute domains). Symbol: R **U** S

- **Set Difference:** (requires 2 Relations), **R-S** = a relation consisting of the tuples that are in relation R but not in S, R and S must be union compatible. S **-** R