BINF 8211/6211
Design and Implementation of Bioinformatics Databases
Lecture #9

Dr. D. Andrew Carr
Dept. Bioinformatics and Genomics UNCC
Spring 2016

Warm-up questions

- What function does this symbol represent and what does it do? Π
- For which of these operations does the order of the entities change the outcome: ∪ ∩ x \
- How many subsets will the Power Set of {1,2,3,4} contain?
- What does the property called closure tell us?
- How is the original entity Gene altered by the function Select?
- In performing a Select operation how do you specify the conditions that must be True?
- What are the permitted comparison operators?
- What are the permitted logical connectives?
- What 3 operators does the JOIN combine?
- What does UNION-compatible mean?

- Set theory is a field with axioms, applications and areas of study – relations are not *exactly* sets so some care has to be taken in generalizing.
- Everything can be assigned to groups – what is the correspondence between groups, what constitutes a subset.

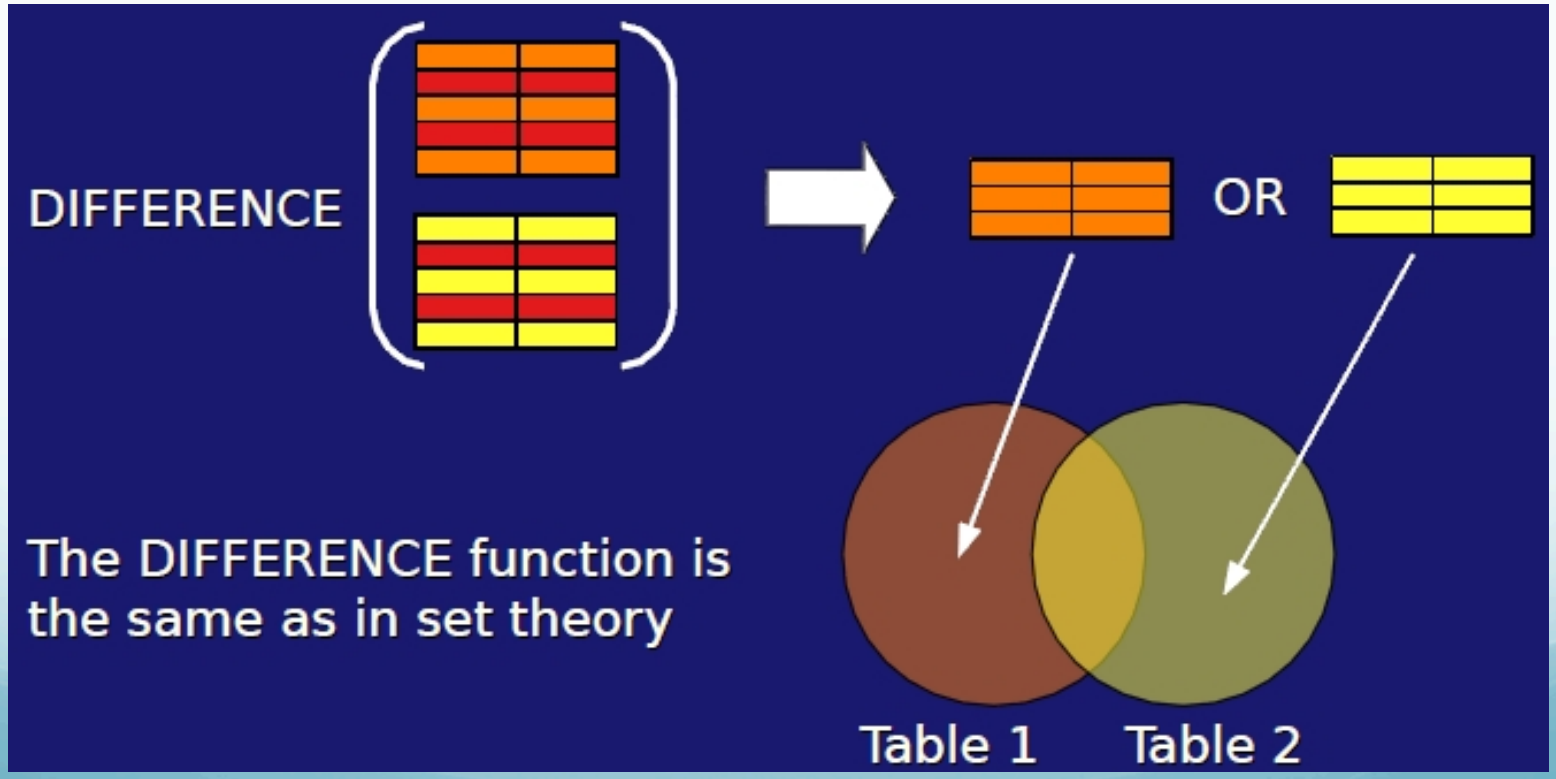
Symbol, Name	Use
σ Selection	Return rows of the input relation that are TRUE (satisfy the predicate)
Π Projection	Output columns from all rows of the input relation (remove duplicate tuples)
X Cartesian Product	Output all pairs of rows from TWO input relations
$\cup$ Union	Output the union of all tuples from two input relations.
- Difference	Allows you to find tuples in one relation but not another
ρ Rename	Given an expression E, ρ_x (E) returns the result of E under a name x.
|X| Join, with variations... e.g. ⟕ (Left Outer)	Return pairs of rows from TWO input relations where attribute values match (for attributes with the same name)

Topics

Clarification of Relational Algebra
Introduction to SQL

The DIFFERENCE function returns all rows from one table that are NOT in another.

- The columns in all tables tested must be union-compatible.
- O-Y has tuples in O but not Y
- Y-O has tuples in Y but not O (the \ 'NOT' operator)



Human

Protein ID	Gene_ID	Conc (ng/ml)	Organism	Condition
P02768	ALB	1.728E7	Human	Breast Cancer
P60709	ACTB	5.162E6	Human	Breast Cancer
B0I1T2	MYO1G	4.053E5	Human	HIV Disease
O94832	MYO1D	1.336E5	Human	Prostate Cancer
O43795	MYO1B	2.143E6	Human	Standard
P09486	SPARC	9.296E5	Human	Breast Cancer
P08134	RHOC	1.67E6	Human	Prostate Cancer
P13796	LCP1*	2.071E6	Human	Standard
Q9Y490	TLN1*	1.445E7	Human	Standard

Mouse

Protein ID	Gene_ID	Conc (ng/ml)	Organism	Condition
P07724	ALB	1.809E7	Mouse	Breast Cancer
P11531	DMD*	4.304E4	Mouse	Liver
P60710	ACTB	1.392E7	Mouse	Embryo 15 d
Q5SUA5	MYO1G	1.027E2	Mouse	Asthma model
Q5SYD0	MYO1D	3.398E5	Mouse	Influenza
P52480	PKM*	6.883E6	Mouse	Brain
P46735	MYO1B	9.499E4	Mouse	Standard
P70663	SPARC	9.296E5	Mouse	Breast Cancer
Q99PT1	RHOC	5.724E5	Mouse	Embryo 15 d

What proteins are expressed in Mouse and not expressed in Human?

$$\Pi_{ProteinID}\,(\sigma_{Gene\_ID}(\text{Mouse})) - \Pi_{ProteinID}\,(\sigma_{Gene\_ID}(\text{Human}))$$

ProteinID
P52480
P11531

PROT_Hs

Protein ID	GeneID	Conc (ng/ml)	Condition
P02768	ALB	1.728E7	Breast Cancer*
P60709	ACTB	5.162E6	Breast Cancer*
B0I1T2	MYO1G	4.053E5	HIV Disease
O94832	MYO1D	1.336E5	Prostate Cancer
O43795	MYO1B	2.143E6	Standard
P09486	SPARC	9.296E5	Breast Cancer*
P08134	RHOC	1.67E6	Prostate Cancer

PROT_Mm

Protein ID	GeneID	Conc (ng/ml)	Condition
P07724	ALB	1.809E7	Breast Cancer*
P60710	ACTB	1.392E7	Embryo 15 d
Q5SUA5	MYO1G	1.027E2	Asthma model
Q5SYD0	MYO1D	3.398E5	Influenza
P46735	MYO1B	9.499E4	Standard
P70663	SPARC	9.296E5	Breast Cancer*
Q99PT1	RHOC	5.724E5	Embryo 15 d

"What genes are expressed in both Human and Mouse in Breast Cancer?"

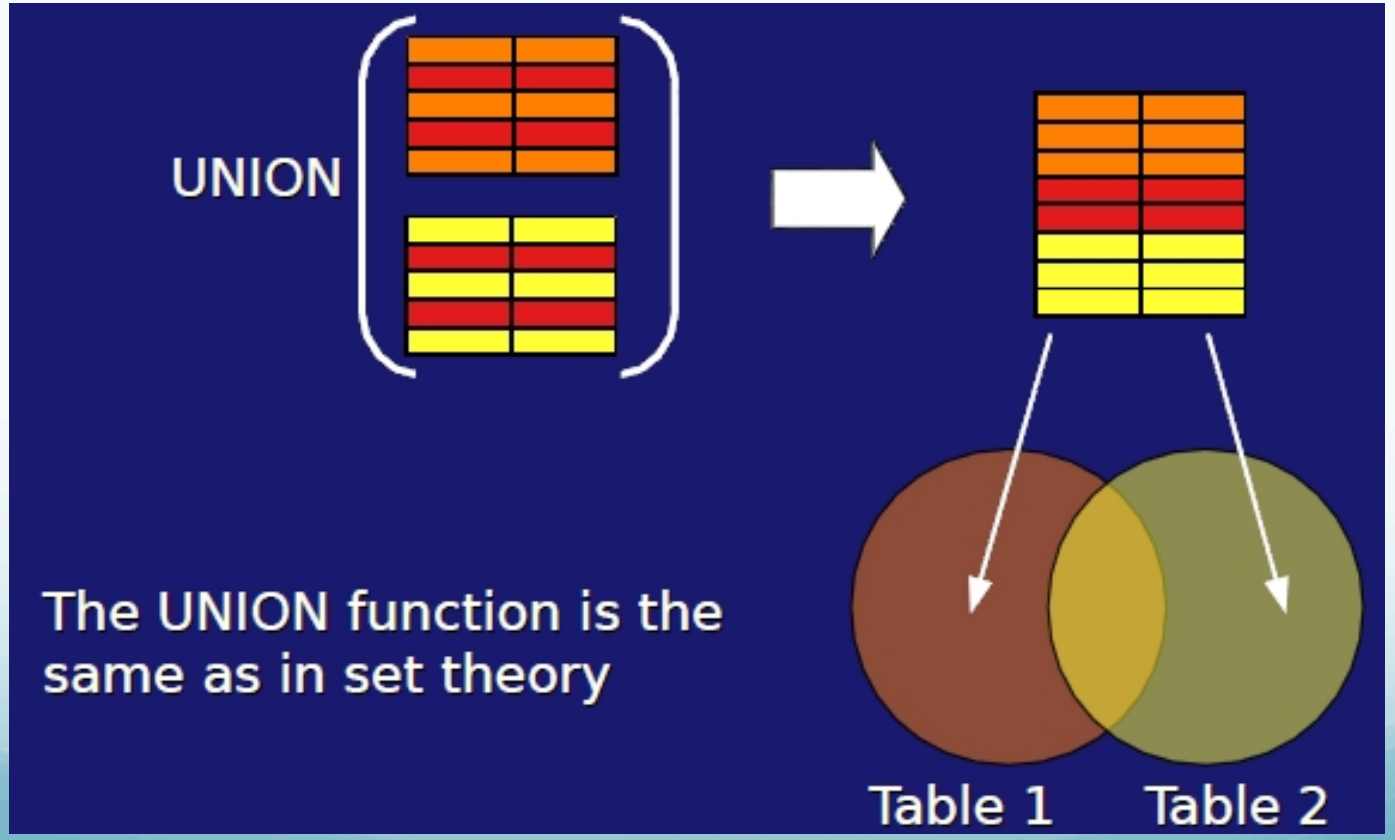
$$R \cap S = R - (R - S)$$

$$\Pi_{GeneID}(\sigma_{Condition=\text{“Breast Cancer”}}(\text{PROT\_Hs})) \cap \Pi_{GeneID}(\sigma_{Condition=\text{“Breast Cancer”}}(\text{PROT\_Mm}))$$

GeneID
ALB
SPARC

UNION function combines all of the rows from the specified tables BUT excludes duplicates.

- The tables must have the same NUMBER of columns (arity) and the columns must have the same domains (data formats and constraints) in order to be union compatible. Duplicates are eliminated.



PROTEIN_T1

Protein ID	Gene_ID	Conc (ng/ml)	Organism	Condition
P02768	ALB	1.728E7	Human	Liver Cancer_1
P60709	ACTB	5.162E6	Human	Liver Cancer_1
B0I1T2	MYO1G	4.053E5	Human	HIV Disease_1
O94832	MYO1D	1.336E5	Human	Prostate Cancer_1
P46735	Myo1b	9.499E4	Mouse	Standard_1
P09486	SPARC	9.296E5	Human	Liver Cancer_1
P07214	SPARC	5.379E4	Mouse	Standard_1

$\Pi_{ProteinID,\ Conc,\ Condition}(\sigma_{\ Organism\ =\ \text{“Human”}}(\text{Time1}))$

Protein ID	Conc (ng/ml)	Condition
P02768	1.728E7	Liver Cancer_1
P60709	5.162E6	Liver Cancer_1
B0I1T2	4.053E5	HIV Disease_1
O94832	1.336E5	Prostate Cancer_1
P09486	9.296E5	Liver Cancer_1

“Show proteinID, concentration and condition at Time 2 and Time 1, in humans”

PROTEIN_T2

Protein ID	Gene_ID	Conc (ng/ml)	Organism	Condition
P02768	ALB	2.032E7	Human	Liver Cancer_2
P60709	ACTB	4.0093E6	Human	Liver Cancer_2
B0I1T2	MYO1G	4.053E5	Human	HIV Disease_2
O94832	MYO1D	1.559E5	Human	Prostate Cancer_2
P46735	Myo1b	8.221E4	Mouse	Standard_2
P09486	SPARC	7.224E5	Human	Liver Cancer_2
P07214	SPARC	4.664E4	Mouse	Standard_2

$\Pi_{ProteinID,\ Conc,\ Condition}(\sigma_{\ Organism\ =\ \text{“Human”}}(\text{Time2}))$

Protein ID	Conc (ng/ml)	Condition
P02768	2.032E7	Liver Cancer_2
P60709	4.0093E6	Liver Cancer_2
B0I1T2	4.053E5	HIV Disease_2
O94832	1.559E5	Prostate Cancer_2
P09486	7.224E5	Liver Cancer_2

https://www.proteinspire.org/MOPED/mopedviews/proteinExpressionDatabase.jsf

Show ProteinID, Conc and Condition at Time 2 and Time 1, in humans

$\Pi_{ProteinID,\ Conc,\ Condition}(\sigma_{\ Organism\ =\ \text{“Human”}}(\text{Time1})) \cup \Pi_{ProteinID,\ Conc,\ Condition}(\sigma_{\ Organism\ =\ \text{“Human”}}(\text{Time2}))$

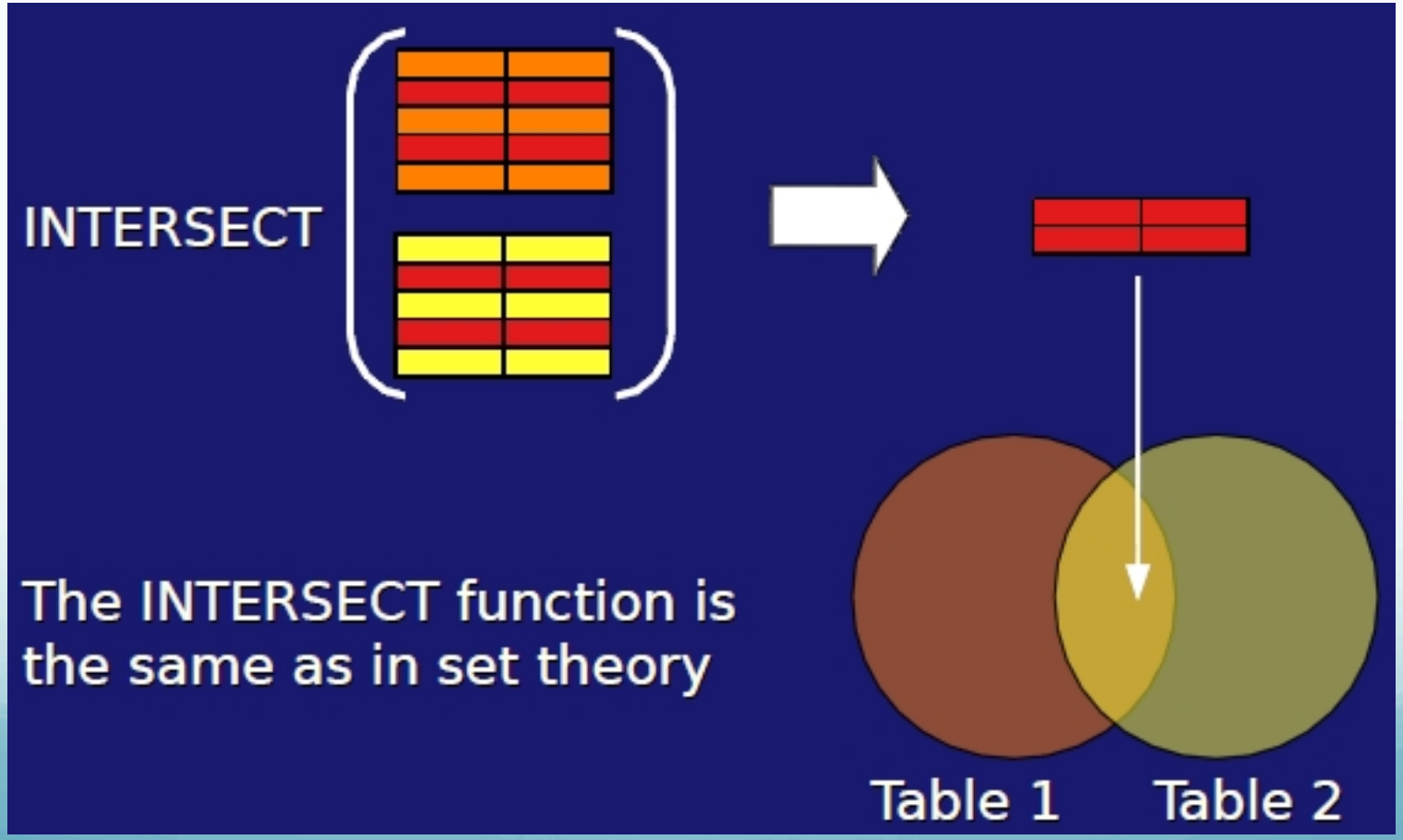
Protein ID	Conc (ng/ml)	Condition
P02768	1.728E7	Liver Cancer_1
P60709	5.162E6	Liver Cancer_1
B0I1T2	4.053E5	HIV Disease_1
O94832	1.336E5	Prostate Cancer_1
P09486	9.296E5	Liver Cancer_1

Protein ID	Conc (ng/ml)	Condition
P02768	2.032E7	Liver Cancer_2
P60709	4.0093E6	Liver Cancer_2
B0I1T2	4.053E5	HIV Disease_2
O94832	1.559E5	Prostate Cancer_2
P09486	7.224E5	Liver Cancer_2

What there is no differentiation the time points through the condition?

INTERSECT function returns those rows common to the specified tables, eliminating duplicates.

- The columns must be *union-compatible*.



The DIVISION function is binary, notation is ÷ returns all tuples from one table that match all tuples in the other .

- The columns in all tables tested must be union-compatible.
- $r \div s = \Pi_{R-S}(r) - \Pi_{R-S}((\Pi_{R-S}(r) \times s) - \Pi_{R-S,S}(r))$ Huh?

$t \in \Pi_{R-S}(r)$

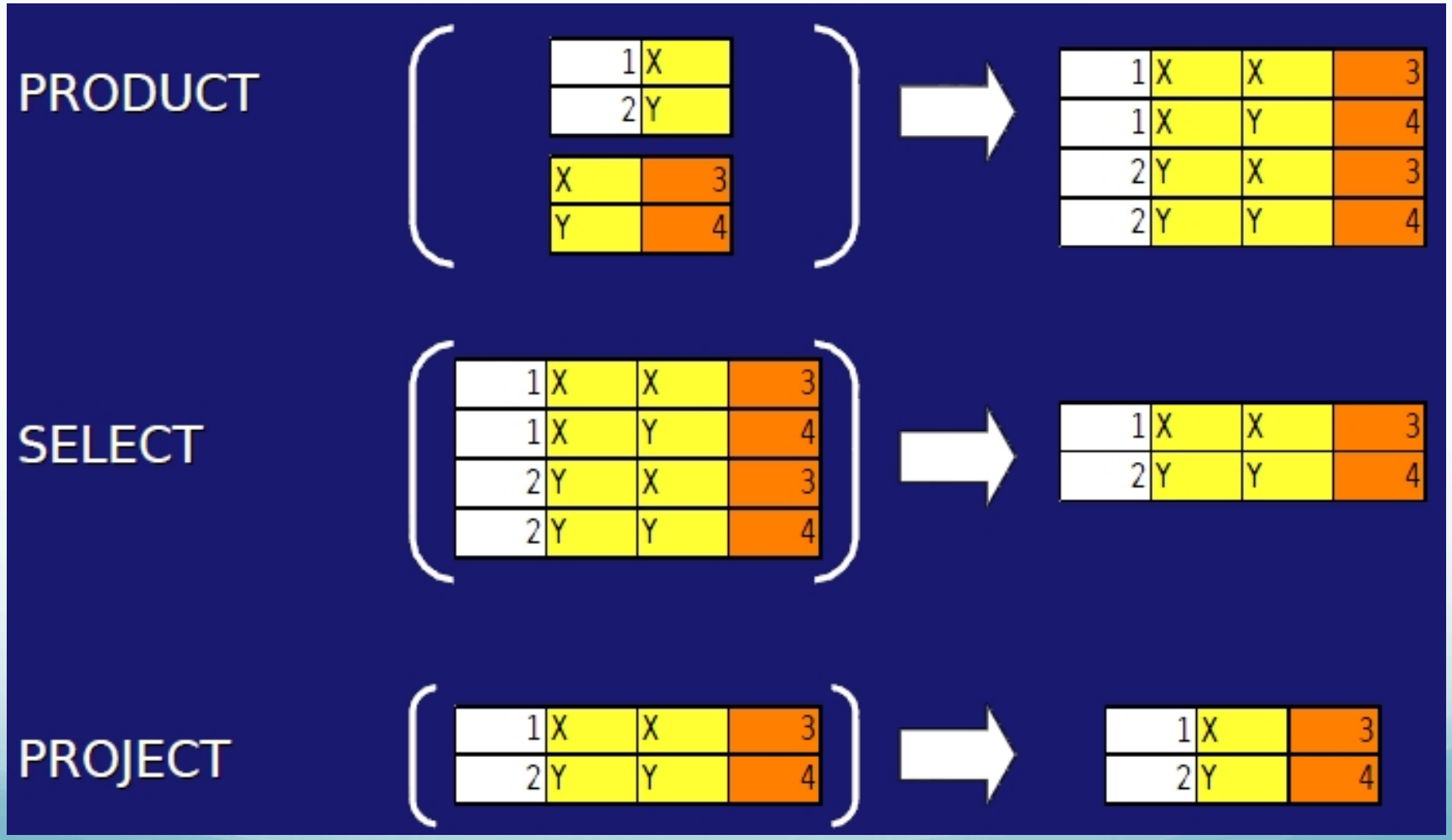
$\forall\ t_s \in s,\ \exists\ t_r \in r$ that satisfies both of:

(a) $t_r[S] = t_s[S]$

(b) $t_r[R-S] = t$

- This is used when your query is about 'all' of something: which genes occur in all of the cell locations?
- Which genes are expressed under all conditions that have been tested?
- The description after the all defines a set with s number of elements
  - you could reframe the question in terms of if/then
- If there is a cell location then this gene appears there or If there is a condition we have collected expression data for then the gene appears with a positive value.

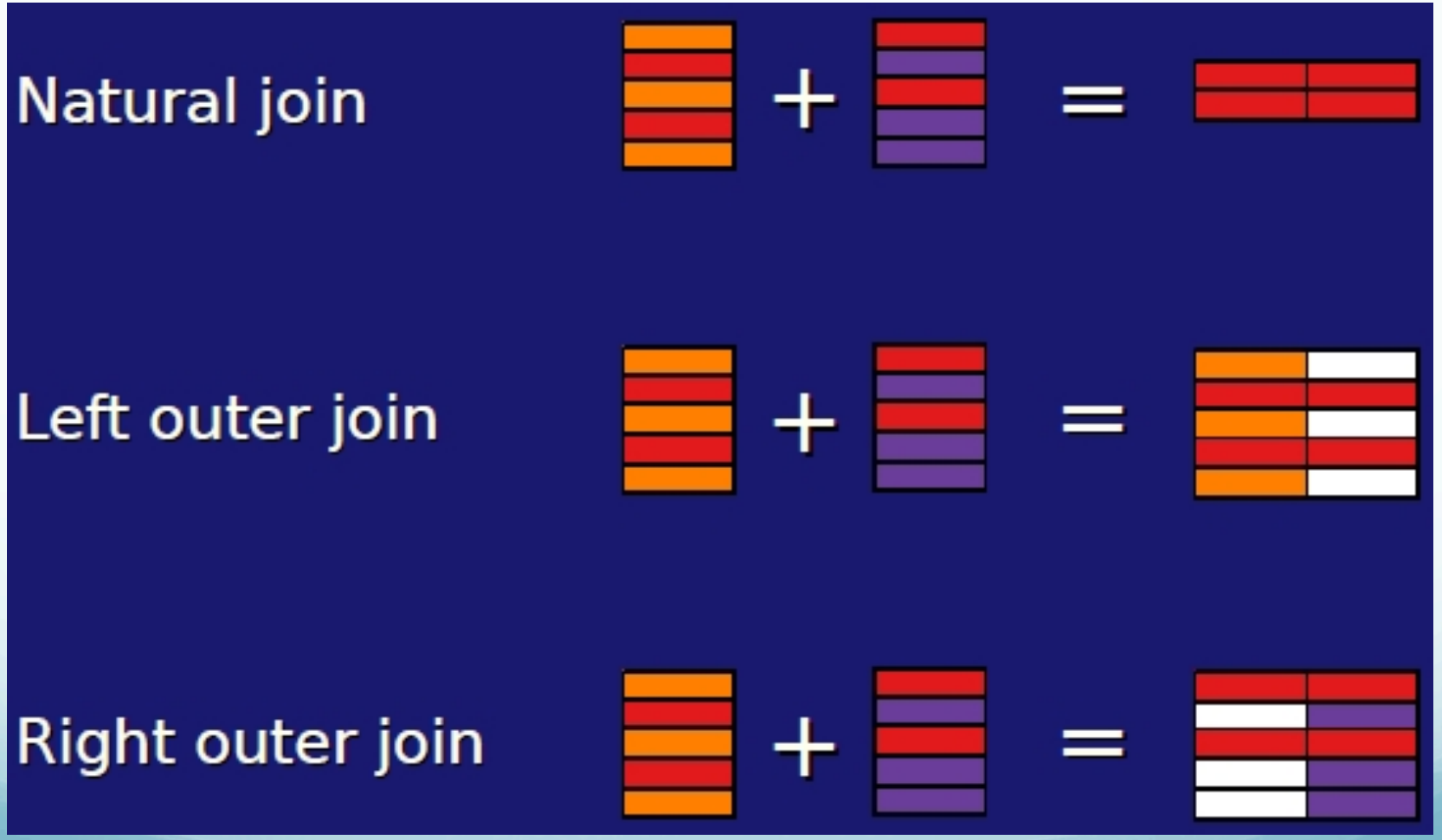
Natural Join produces the results from three consecutive steps.



Types of Joins

- A Natural Join is made on all attributes (headings) that have the same name in each relation.
- A THETA JOIN is more general than the Natural Join: instead of requiring an attribute value to match the condition can be any predicate on the attributes.
  - Duplicated columns are NOT removed
- An EQUIJOIN is when that predicate, θ, is the '=' comparison operator. $r \bowtie_\theta s = \sigma_\theta(r \times s)$

OUTER JOIN combines two tables but retains ***all*** of only one of them (direction matters).



What do you expect a FULL OUTER JOIN to include?

Operators and Functions

- Comparison: <,>,=,≠, ≤, ≥
- Arithmetic: +, -, *, divide, power and log
- Logical: AND, OR, NOT, IDENTITY
- Aggregate
  - SUM
  - MIMINUM
  - MAXIMUM
  - AVERAGE (also MEAN and MEDIAN)
  - COUNT

The Whole of the Thing: a Formal Definition of the Relational Algebra

- A basic expression, E, consists of either
  - A Relation (table in the db)
  - A Constant Relation, written by listing its tuples $\{A_1, A_2 \ldots A_n\}$
- A *general expression* is constructed from sub-expressions in the following list. If E_1 and E_2 are relational-algebra expressions, the following is the complete set of expressions:
  - $E_1 \cup E_2$
  - $E_1 - E_2$
  - $E_1 \times E_2$
  - $\sigma_p (E_1)$ where P is a predicate on attributes in E1
  - ? *s(E1)* where S is a list of some of E1's attributes
  - $\rho_x (E_1)$ where x is the new name for the result of E1
- Other operations *simplify queries* but do not add expressive power to the above.

SQL

- Structured Query Language,
  - a standard declarative language (with some procedural elements)
  - stating the relational operators in an RDBMS.
- The standardization
  - portable
  - widely adopted
  - eases interoperability
- Note that the devil is in the details – lining up data formats for example is a frequent task.
  - The language has been revised a number of times:
    - SQL86, 89, 92, 99, 03, 08 are all official but a given DBMS is not necessarily using the latest (SQLite → 99)
    - Most DBMS add non-standard declarations as well – very useful, not at all portable.

SQL

- SQL can serve as a data definition language (DDL)
  - This permits creation of a database, tables and objects with correctly defined logical types
- A declarative language (vs procedural)
  - The basic syntax is quite simple
- SQL can serve as a data manipulation language (DML)
  - This permits certain tasks to be correctly performed, including inserting, appending and deleting data.
  - New sets can be correctly formed from existing sets based on the logical structures and operators.

Does SQL make a database relational?

- NO -- The language is used to build and access the DBMS
  - The model underneath imposes the structure.
- The relational model imposes a particular logical structure on which reasoning can be applied

Can you call a database relational without using SQL?

- YES
- The access mode, implementation and operational details are handled by the software engine.
  - There are often *logically equivalent* models.
    - Because of performance tuning, some SQL DBMS may change the logical model
    - Although the relational model explicitly defines tuples as un-ordered, SQL does order the columns and rows of a table.

SQL and relational algebra

- Conforms to relational algebraic rules
  - It can deviate
- Common deviations:
  - duplicate rows can be allowed in an SQL table
  - un-named attribute in an SQL table
    - this means you cannot reference it in expressions.
  - Two or more columns of the same SQL table can have the same name
    - impossible to unambiguously reference one of them.

SQL and relational algebra Deviations continued

- The order of columns in an SQL table is defined and significant.
  - The Cartesian product and the UNION will then be non-commutative (order will change outcome).
- Updates to a View can be accepted but not appear as expected in your relational algebra expression (a CHECK OPTION function must be used)
- To have full relational algebra capabilities you must be allowed to have relations of degree (arity) zero
  - SQL requires every table to have at least one column.
- Acceptance of the NULL as a valid attribute value is really expressing a 3-value logic, not a 2-value logic.
- NULL is problematic because it is used to mean several things (sum of the empty set, average of the empty set, in a LEFT JOIN a null means there is no matching row in the right-hand operand, etc.) .

SQL Language components

- Clauses are the components of statements and queries
- Expressions can produce:
  - scalar values or new tables of rows and columns
- Predicates specify the conditions to evaluate
  - SQL permits True/False/Unknown.
- Queries retrieve data based on conditions
- Statements may effect either the schema or the data and may control transactions, connections, sessions or diagnostics
  - Statements use the semicolon as a statement terminator (standard syntax)

SQL Operators

Operators [edit]

Operator	Description	Example
`=`	Equal to	`Author = 'Alcott'`
`<>`	Not equal to (many DBMSs accept `!=` in addition to `<>`)	`Dept <> 'Sales'`
`>`	Greater than	`Hire_Date > '2012-01-31'`
`<`	Less than	`Bonus < 50000.00`
`>=`	Greater than or equal	`Dependents >= 2`
`<=`	Less than or equal	`Rate <= 0.05`
`BETWEEN`	Between an inclusive range	`Cost BETWEEN 100.00 AND 500.00`
`LIKE`	Match a character pattern	`First_Name LIKE 'Will%'`
`IN`	Equal to one of multiple possible values	`DeptCode IN (101, 103, 209)`
`IS` *or* `IS NOT`	Compare to null (missing data)	`Address IS NOT NULL`
`IS NOT DISTINCT FROM`	Is equal to value or both are nulls (missing data)	`Debt IS NOT DISTINCT FROM - Receivables`
`AS`	Used to change a field name when viewing results	`SELECT employee AS 'department1'`

SQL Conditional Statements

- SQL uses the case/when/then/else/end expression
  - formally called a searched case
  - equivalent to if/else in other programming languages.
- SQL tests WHEN conditions in the order they appear in the source.
- If an ELSE is not specified it defaults to ELSE NULL.

```
CASE WHEN n > 0
          THEN 'positive'
     WHEN n < 0
          THEN 'negative'
     ELSE 'zero'
END
```

```
CASE n WHEN 1
          THEN 'one'
       WHEN 2
          THEN 'two'
       ELSE 'I cannot count that high'
END
```

Data Definition Language (DDL)

- SQL is a data definition language
- SQL allows you to create tables with their attributes, and include referential and domain constraints.

General form

```
CREATE TABLE  <table name> (
<column name>       data type [constraint],
<column name>       data type [constraint],
.......
PRIMARY KEY (<column names>),
FOREIGN KEY (<column names>) REFERENCES <table name>,
CONSTRAINT     <constraint>
)
```

- Other commands let you alter, insert, drop, update, delete and truncate tables.

Example SQL Table Create

```
CREATE TABLE gene
(
gene_symbol         text      NOT NULL,
genome_start_loc    integer   NOT NULL,
genome_end_loc      integer   NOT NULL,
genome_strand       text      NOT NULL DEFAULT '+',
gene_seq            text,
operon_id           integer,
PRIMARY KEY(gene_symbol),
FOREIGN KEY(operon_id) REFERENCES operon
);
```