Set Operations and Databases

DATA 604

Leanne Wu

lewu@ucalgary.ca

Department of Computer Science

UNIVERSITY OF CALGARY

Motivation

- Queries (especially complex ones) are easier to write if you understand the formal underpinnings behind SQL

Quick rules:

- Everything is a set (tables should not have duplicates)
- Rows (tuples) are not ordered
- Results may be multi-sets (sets with duplicate copies)

Selects

Selects are often expressed formally as (for example):

$$\sigma_{\text{library='Nose Hill Library'}}(\text{library\_location})$$

Which would be read as

```
SELECT * from library_location
WHERE library = 'Nose Hill Library';
```

Selects with Boolean operators

- Conditions for selects may be joined using AND, OR, NOT
- For example,

$\sigma_{(\text{library}=\text{‘Nose Hill Library’ OR Square\_Feet} > 7500)}(\text{library\_location})$

```
SELECT * from library_location
WHERE library = ‘Nose Hill Library’ OR
     Square_Feet > 7500;
```

Projects

- Projects are expressed formally as (for example):

$$\pi_{Monday\_Open}(\text{library location})$$

Which would be read as

```
SELECT Monday_Open from library_location;
```

Mathematical operations on sets

- Used on relations (let's say R and S) which have the same attributes
- UNION (note: not the SQL version of union)
  - all tuples in R or in S or both. Duplicates are removed
- INTERSECTION
  - all tuples in R and S
- Set difference/MINUS
  - all tuples in R but not S

Cartesian (CROSS) Products

- Given two relations R and S, combines every element of R with every element of S

x	**y**
x1	y1
x2	y2

a	**b**
a1	b1
a2	b2

x	y	a	b
x1	y1	a1	b1
x1	y1	a2	b2
x2	y2	a1	b1
x2	y2	a2	b2