**Nicholas Donahue**

**CS 34800 – Homework 01**

# Part 1 - Relational Algebra and Relational Calculus

1. **Relational Algebra Expressions**
   1. NONE\_BORROWED ← ( ∏ Card\_no (BORROWER) - ∏ Card\_no (BOOK\_LOANS) )

ANSWER ← ∏ Name ( BORROWER ⨝ NONE\_BORROWED )

* 1. WL\_ID ← ∏ Branch\_id ( σ Branch\_name = “WLafayette” (LIBRARY\_BRANCH) )

ANSWER ← ∏ Title, Name, Address ( BOOK ⨝ BORROWER ⨝ WL\_ID )

* 1. BOOK\_COUNT ← Card\_no g count(Branch\_id) as count (∏ Branch\_id, Card\_no ( BOOK\_LOANS ))

MID\_STEP ← σ count >= 2 (BOOK\_COUNT ⨝ BORROWER)

ANSWER ← ∏ Name ( MID\_STEP )

* 1. KINGS\_BRANCHES ← σ Author\_name = “Stephen King” (BOOK\_COPIES ⨝ BOOK\_AUTHORS ⨝ LIBRARY\_BRANCH )

SUM\_BRANCHES ← Branch\_id g Sum(No\_of\_copies) as Sum ( KINGS\_BRANCHES )

COUNTS ← Author\_name g Count(Book\_id) as Count ( BOOK\_AUTHORS )

KING\_COUNT ← σ Author\_name = “Stephen King” ( COUNTS )

ALMOST ← σ Sum >= KING\_COUNT ( SUM\_BRANCHES )

ANSWER ← ∏ Branch\_name (ALMOST)

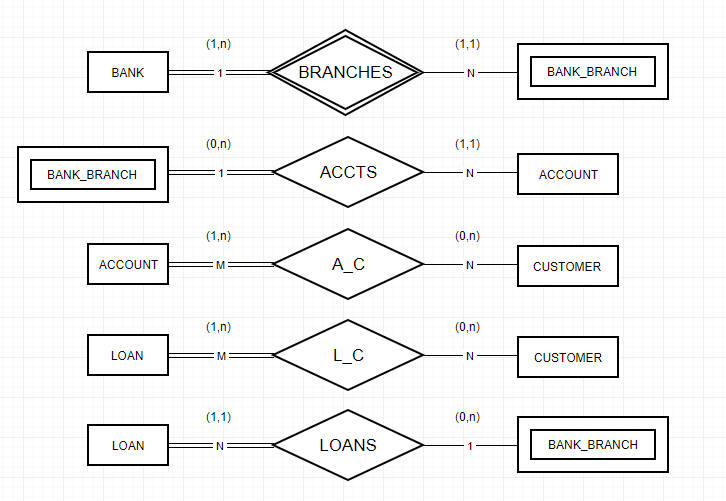
* 1. BRANCH\_FIVE ← σ Branch\_id = 5 ( BOOK\_LOANS ⨝ BORROWER ⨝ BOOK\_AUTHORS )

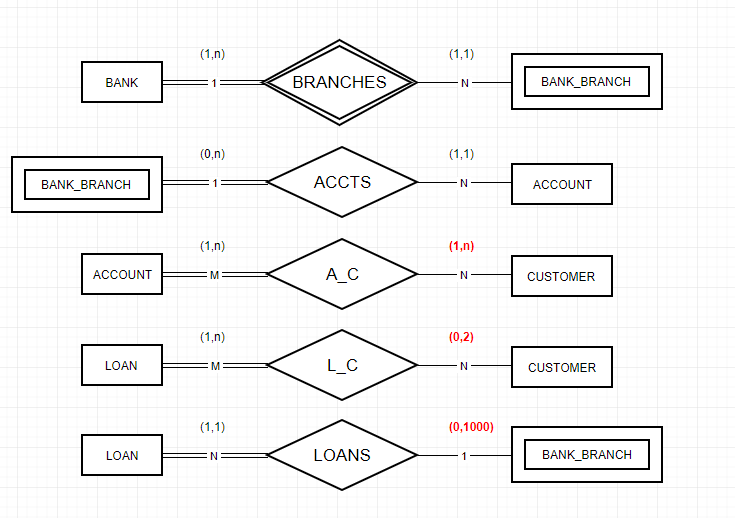
DOS\_OUT ← σ Author\_name = “Dostoevsky” ( BRANCH\_FIVE )

TOL\_OUT ← σ Author\_name = “Tolstoy” ( BRANCH\_FIVE )

ANSWER ← ∏ Name, Phone ( DOS\_OUT ⨝ TOL\_OUT )

# Part 2 - ER-Diagram Fundamentals

1. The strong (non-weak) entity types are: BANK, ACCOUNT, CUSTOMER, and LOAN
2. Yes, there is a weak entity type. The weak entity type is: BANK\_BRANCH. The partial key is: Branch\_No. The identifying relationship is: BRANCHES.
3. *Made with* [www.draw.io](http://www.draw.io)
4. *Made with* [www.draw.io](http://www.draw.io)*. Changes highlighted in red from part C.*



# Part 3 - Designing an E-R Diagram

**Assumptions:**  - There are no weak entity types as each entity can be defined by its own unique id (NOTE: e-mails are unique by design).

**-** An engineer is associated with **zero** **or more** expertise (indicated by single line)

- Each engineer can only be the contact of **one** project at a time.

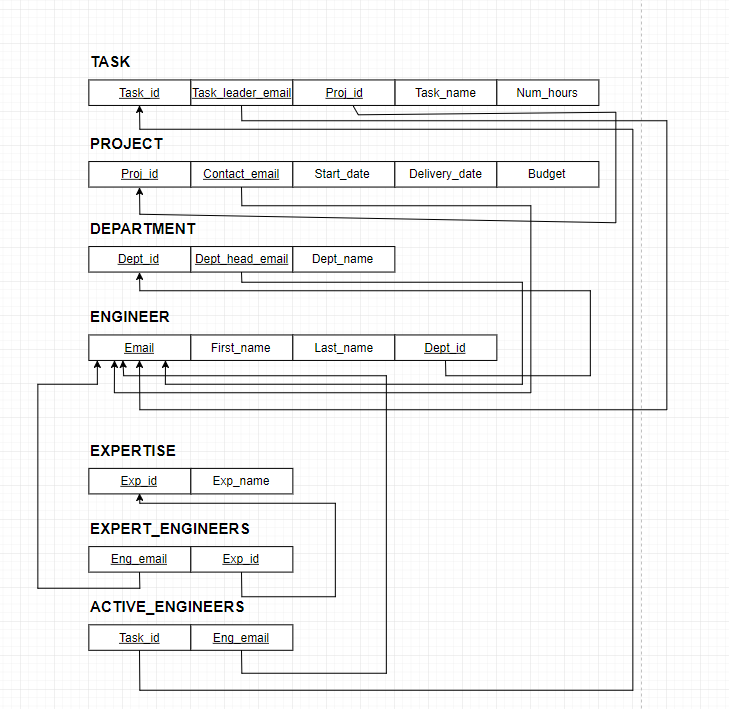
- A single engineer can be the leader of **multiple** tasks at a time.

- A single engineer can be assigned to **multiple** tasks at a time.

- Each engineer can only be the head of **one** department at a time.

( *Part 4 on next page… )*

# Part 4 - Converting an E-R Diagram to the relational model



For this relational model, I have underlined primary keys and foreign keys. The primary keys are the first entry in the model of each entity, and the foreign keys have arrows that point towards their origin (which entity they key is ‘foreign’ from).

I created two new tables (EXPERT\_ENGINEERS and ACTIVE\_ENGINEERS) as a design solution to properly handle 0 to N engineer’s expertise and 0 to N active engineers on task. By doing this, I can prevent null entries in the tables of ENGINEER/TASK **and** neatly organize their multi-valued attributes in their respective tables(a.k.a. expertise and active tasks). It is because of the nature of these multi-valued attributes that this must be done—since an engineer can have 0 to N area of expertise and/or active tasks that means there is a likelihood he will have multiple values for the same attribute.