# CSCI 4333 Section 1 Design of DB Systems

## 3/4/2024 (self - annotation)

Fall 2023 Mid-term:

A diagram of a computer

Description automatically generated with medium confidence

Rules for associations V, Y and Z:

**A1**. For a many to one association (e.g. Z in the diargram above) between C1, T: faculty (the class with the one multiplicity) and Cm, T: class, add a column R1\_Id (FacId) into the relation Rm, class (which implements Cm).

1. Assume that R1\_Id (FacId) is the primary key of the relation R1, faculty (which implements C1).
2. R1\_Id (facId: Z\_T\_Id, FK of T) is a foreign key of Rm: T (class) referencing R1: T(R1\_Id): T\_Id of T
3. The name R1\_Id may be renamed.
4. R1\_Id is not null in Rm iff (if and only if) 0 is not allowed (i.e., 1..1) for C1.
5. Any single valued attribute of the association is mapped to a column in Rm.
6. If you have composite or multi-valued attributes of the relationship, you should consider promoting the association to an association class or a regular class in your UML class diagram.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Rules for creating the table W:

**A2.** For a many to many association (including association classes) between classes CA and CB, create a*new* relation RAB(RA\_Id, RB\_Id).

1. (RA\_Id, RB\_Id) is a candidate key.
2. RA\_Id references RA(RA\_Id) as a foreign key.
3. RB\_Id references RB(RB\_Id) as a foreign key.

**Introduction to Relational Algebra and Relational Calculus**

by K. Yue

**1. Introduction**

* More theoretical query languages of the relational model:
  1. Relational Algebra (RA): a simple functional language: e.g. LISP
  2. Relational Calculus (RC): a simple declarative language
* Provide theoretical foundation for the relational model.
* Not used in practical DBMS.
* Some symbols in RA/RC: [RA\_RC\_Symbols.docx](https://dcm.uhcl.edu/yue/courses/joinDB/Spring2024/notes/rarc/RA_RC_Symbols.docx)

**2. Introduction to Relational algebra**

* Include a set of *basic* and *derived* set-theoretic operations.
* Procedural: specify a sequence of operations.
  + performance optimization is important.
* Operations can be unary or binary.
* The result is also a relation: closure property => chained operations.
* RA solutions are algorithmic.

***Example:***

See [toyu\_Ex.pdf](https://dcm.uhcl.edu/yue/courses/joinDB/Spring2024/notes/rarc/toyu_Ex.pdf) and [toyu\_RA\_sol\_even.pdf](https://dcm.uhcl.edu/yue/courses/joinDB/Spring2024/notes/rarc/toyu_RA_sol_even.pdf)

**3. Introduction to Relational Calculus**

* Non-procedural, *declarative*, and high level.
* Two kinds:
  1. Domain Relational Calculus (DRC)
  2. Tuple Relational Calculus (TRC)
* Results specified by the *set builder form*: {s | cond(s)}
* cond(s) is known as a *formula*.
* Constructs:
  1. Variables:
     1. TRC: tuples (bound to tuples): e.g,. s, t, student, class, etc.
     2. DRC: Attributes (bound to domain value): e.g., a, b, c, stuId, fname, etc.
     3. RC's variable is sometime known as 'dummy variable'.
  2. Constants: string, int, etc., E.g., 12, 'csci', 3.7.
  3. Comparison operators: <, <, =, etc.
  4. Boolean operators: and (conjunction, ∧ or just ,), or (disjunction ∨), not (¬), implies (⇒), etc.
  5. Membership functions: belongs to, ∈, not belongs to, ∉, etc.
  6. Quantifiers: there exists (existential, ∃), for all (universal ∀).

***Example:***

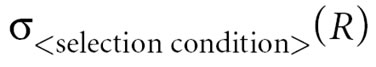
See [toyu\_Ex.pdf](https://dcm.uhcl.edu/yue/courses/joinDB/Spring2024/notes/rarc/toyu_Ex.pdf) and [toyu\_RC\_sol\_odd.pdf](https://dcm.uhcl.edu/yue/courses/joinDB/Spring2024/notes/rarc/toyu_RC_sol_odd.pdf)

**4. Relational Algebra**

**4.1 Basic Operations**

**4.1.1 Select**

1. Unary operation.
2. Select tuples (with the same schema) based on a Boolean condition.
3. Conditions may includes attributes in the relational schema.
4. The Boolean expression of the condition can be composite.
5. *'Horizontal' subset*.
6. Not to be confused with the Select statement in SQL.



σcond(R) = {t | t ∈ R and cond}, or simply

σcond(R) = {t | t ∈ R, cond}

**Example:** All information of students majoring in CSCI.

σmajor='CSCI'(Student)

+--------+-------+-------+-------+-------+---------+---------+  
| stuId  | fname | lname | major | minor | credits | advisor |  
+--------+-------+-------+-------+-------+---------+---------+  
| 100000 | Tony  | Hawk  | CSCI  | CINF  |      40 |    1011 |  
| 100001 | Mary  | Hawk  | CSCI  | CINF  |      35 |    1011 |  
| 100002 | David | Hawk  | CSCI  | ITEC  |      66 |    1011 |  
+--------+-------+-------+-------+-------+---------+---------+  
3 rows  
  
In SQL, this is just:

SELECT \*  
FROM Student  
WHERE major = 'CSCI';

**4.1.2 Project**

1. Unary operation
2. Select attributes from tuples.
3. Duplicate results removed (because a relation is a set).
4. 'Vertical' subset.

πc1, .., cm(R) = {s | ∃t ∈ R (t(ci) = s(ci), for 1 <= i <= m)},

or simply

πc1, .., cm(R) = {s | t ∈ R (t(ci) = s(ci), for 1 <= i <= m)}

project

**Example:** Names and majors of students  
  
πLName, FName, Major(Student):

+-----------+---------+-------+  
| FName     | LName   | Major |  
+-----------+---------+-------+  
| Tony      | Hawk    | CSCI  |  
| Mary      | Hawk    | CSCI  |  
| David     | Hawk    | CSCI  |  
| Catherine | Lim     | ITEC  |  
| Larry     | Johnson | ITEC  |  
| Linda     | Johnson | CINF  |  
| Lillian   | Johnson | CINF  |  
| Ben       | Zico    | NULL  |  
| Bill      | Ching   | ARTS  |  
| Linda     | King    | ARTS  |  
+-----------+---------+-------+  
10 rows  
  
**4.1.3.Cartesian Product**

1. Same as the usual definition of Cartesian Product of two sets.
   1. Remember that a relation is a set.
2. Merge all possible information from two relations.
3. Also called Cross Product or Cross Join.
4. Name ambiguity may be resolved by using full names.
5. The cardinality of a set S is |S|, the number of elements in the set.
6. |RxS|= |R| \* |S|
7. Not very useful in practice as the result can be large and constructing the result can be time consuming.



***Example:***

R(A,B,C) has three tuples. S(A,D) has four tuples.

The result of R x S always has 12 tuples with the schema (R.A, B, C, S.A, D).

**Example:** in toyu

student:  
+--------+-----------+---------+-------+-------+---------+---------+  
| stuId  | fname     | lname   | major | minor | credits | advisor |  
+--------+-----------+---------+-------+-------+---------+---------+  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 |  
+--------+-----------+---------+-------+-------+---------+---------+  
10 rows in set  
  
  
enroll:  
+--------+---------+-------+----------+  
| stuId  | classId | grade | n\_alerts |  
+--------+---------+-------+----------+  
| 100000 |   10000 | A     |        0 |  
| 100001 |   10000 | NULL  |     NULL |  
| 100002 |   10000 | B-    |        3 |  
| 100000 |   10001 | A     |        2 |  
| 100001 |   10001 | A-    |        0 |  
| 100000 |   10002 | B+    |        1 |  
| 100002 |   10002 | B+    |        2 |  
| 100000 |   10003 | C     |        0 |  
| 100002 |   10003 | D     |        4 |  
| 100004 |   10003 | A     |        0 |  
| 100005 |   10003 | NULL  |     NULL |  
| 100000 |   10004 | A-    |        1 |  
| 100004 |   10004 | B+    |     NULL |  
| 100005 |   10004 | A-    |        0 |  
| 100006 |   10004 | C+    |     NULL |  
| 100005 |   10005 | A-    |        0 |  
| 100006 |   10005 | A     |     NULL |  
| 100005 |   10006 | B+    |     NULL |  
| 100007 |   10007 | F     |        4 |  
| 100008 |   10007 | C-    |        0 |  
| 100007 |   10008 | A-    |        0 |  
| 100000 |   11001 | D     |        4 |  
+--------+---------+-------+----------+  
22 rows  
  
  
student x enroll:  
+--------+-----------+---------+-------+-------+---------+---------+--------+---------+-------+----------+  
| stuId  | fname     | lname   | major | minor | credits | advisor | stuId  | classId | grade | n\_alerts |  
+--------+-----------+---------+-------+-------+---------+---------+--------+---------+-------+----------+  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100000 |   10000 | A     |        0 |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100000 |   10000 | A     |        0 |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100000 |   10000 | A     |        0 |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100000 |   10000 | A     |        0 |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100000 |   10000 | A     |        0 |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100000 |   10000 | A     |        0 |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100000 |   10000 | A     |        0 |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100000 |   10000 | A     |        0 |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100000 |   10000 | A     |        0 |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100000 |   10000 | A     |        0 |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100001 |   10000 | NULL  |     NULL |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100001 |   10000 | NULL  |     NULL |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100001 |   10000 | NULL  |     NULL |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100001 |   10000 | NULL  |     NULL |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100001 |   10000 | NULL  |     NULL |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100001 |   10000 | NULL  |     NULL |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100001 |   10000 | NULL  |     NULL |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100001 |   10000 | NULL  |     NULL |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100001 |   10000 | NULL  |     NULL |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100001 |   10000 | NULL  |     NULL |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100002 |   10000 | B-    |        3 |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100002 |   10000 | B-    |        3 |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100002 |   10000 | B-    |        3 |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100002 |   10000 | B-    |        3 |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100002 |   10000 | B-    |        3 |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100002 |   10000 | B-    |        3 |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100002 |   10000 | B-    |        3 |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100002 |   10000 | B-    |        3 |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100002 |   10000 | B-    |        3 |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100002 |   10000 | B-    |        3 |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100000 |   10001 | A     |        2 |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100000 |   10001 | A     |        2 |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100000 |   10001 | A     |        2 |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100000 |   10001 | A     |        2 |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100000 |   10001 | A     |        2 |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100000 |   10001 | A     |        2 |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100000 |   10001 | A     |        2 |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100000 |   10001 | A     |        2 |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100000 |   10001 | A     |        2 |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100000 |   10001 | A     |        2 |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100001 |   10001 | A-    |        0 |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100001 |   10001 | A-    |        0 |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100001 |   10001 | A-    |        0 |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100001 |   10001 | A-    |        0 |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100001 |   10001 | A-    |        0 |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100001 |   10001 | A-    |        0 |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100001 |   10001 | A-    |        0 |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100001 |   10001 | A-    |        0 |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100001 |   10001 | A-    |        0 |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100001 |   10001 | A-    |        0 |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100000 |   10002 | B+    |        1 |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100000 |   10002 | B+    |        1 |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100000 |   10002 | B+    |        1 |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100000 |   10002 | B+    |        1 |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100000 |   10002 | B+    |        1 |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100000 |   10002 | B+    |        1 |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100000 |   10002 | B+    |        1 |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100000 |   10002 | B+    |        1 |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100000 |   10002 | B+    |        1 |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100000 |   10002 | B+    |        1 |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100002 |   10002 | B+    |        2 |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100002 |   10002 | B+    |        2 |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100002 |   10002 | B+    |        2 |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100002 |   10002 | B+    |        2 |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100002 |   10002 | B+    |        2 |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100002 |   10002 | B+    |        2 |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100002 |   10002 | B+    |        2 |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100002 |   10002 | B+    |        2 |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100002 |   10002 | B+    |        2 |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100002 |   10002 | B+    |        2 |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100000 |   10003 | C     |        0 |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100000 |   10003 | C     |        0 |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100000 |   10003 | C     |        0 |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100000 |   10003 | C     |        0 |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100000 |   10003 | C     |        0 |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100000 |   10003 | C     |        0 |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100000 |   10003 | C     |        0 |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100000 |   10003 | C     |        0 |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100000 |   10003 | C     |        0 |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100000 |   10003 | C     |        0 |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100002 |   10003 | D     |        4 |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100002 |   10003 | D     |        4 |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100002 |   10003 | D     |        4 |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100002 |   10003 | D     |        4 |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100002 |   10003 | D     |        4 |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100002 |   10003 | D     |        4 |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100002 |   10003 | D     |        4 |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100002 |   10003 | D     |        4 |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100002 |   10003 | D     |        4 |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100002 |   10003 | D     |        4 |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100004 |   10003 | A     |        0 |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100004 |   10003 | A     |        0 |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100004 |   10003 | A     |        0 |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100004 |   10003 | A     |        0 |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100004 |   10003 | A     |        0 |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100004 |   10003 | A     |        0 |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100004 |   10003 | A     |        0 |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100004 |   10003 | A     |        0 |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100004 |   10003 | A     |        0 |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100004 |   10003 | A     |        0 |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100005 |   10003 | NULL  |     NULL |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100005 |   10003 | NULL  |     NULL |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100005 |   10003 | NULL  |     NULL |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100005 |   10003 | NULL  |     NULL |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100005 |   10003 | NULL  |     NULL |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100005 |   10003 | NULL  |     NULL |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100005 |   10003 | NULL  |     NULL |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100005 |   10003 | NULL  |     NULL |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100005 |   10003 | NULL  |     NULL |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100005 |   10003 | NULL  |     NULL |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100000 |   10004 | A-    |        1 |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100000 |   10004 | A-    |        1 |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100000 |   10004 | A-    |        1 |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100000 |   10004 | A-    |        1 |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100000 |   10004 | A-    |        1 |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100000 |   10004 | A-    |        1 |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100000 |   10004 | A-    |        1 |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100000 |   10004 | A-    |        1 |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100000 |   10004 | A-    |        1 |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100000 |   10004 | A-    |        1 |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100004 |   10004 | B+    |     NULL |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100004 |   10004 | B+    |     NULL |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100004 |   10004 | B+    |     NULL |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100004 |   10004 | B+    |     NULL |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100004 |   10004 | B+    |     NULL |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100004 |   10004 | B+    |     NULL |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100004 |   10004 | B+    |     NULL |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100004 |   10004 | B+    |     NULL |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100004 |   10004 | B+    |     NULL |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100004 |   10004 | B+    |     NULL |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100005 |   10004 | A-    |        0 |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100005 |   10004 | A-    |        0 |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100005 |   10004 | A-    |        0 |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100005 |   10004 | A-    |        0 |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100005 |   10004 | A-    |        0 |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100005 |   10004 | A-    |        0 |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100005 |   10004 | A-    |        0 |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100005 |   10004 | A-    |        0 |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100005 |   10004 | A-    |        0 |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100005 |   10004 | A-    |        0 |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100006 |   10004 | C+    |     NULL |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100006 |   10004 | C+    |     NULL |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100006 |   10004 | C+    |     NULL |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100006 |   10004 | C+    |     NULL |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100006 |   10004 | C+    |     NULL |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100006 |   10004 | C+    |     NULL |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100006 |   10004 | C+    |     NULL |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100006 |   10004 | C+    |     NULL |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100006 |   10004 | C+    |     NULL |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100006 |   10004 | C+    |     NULL |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100005 |   10005 | A-    |        0 |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100005 |   10005 | A-    |        0 |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100005 |   10005 | A-    |        0 |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100005 |   10005 | A-    |        0 |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100005 |   10005 | A-    |        0 |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100005 |   10005 | A-    |        0 |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100005 |   10005 | A-    |        0 |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100005 |   10005 | A-    |        0 |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100005 |   10005 | A-    |        0 |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100005 |   10005 | A-    |        0 |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100006 |   10005 | A     |     NULL |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100006 |   10005 | A     |     NULL |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100006 |   10005 | A     |     NULL |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100006 |   10005 | A     |     NULL |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100006 |   10005 | A     |     NULL |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100006 |   10005 | A     |     NULL |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100006 |   10005 | A     |     NULL |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100006 |   10005 | A     |     NULL |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100006 |   10005 | A     |     NULL |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100006 |   10005 | A     |     NULL |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100005 |   10006 | B+    |     NULL |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100005 |   10006 | B+    |     NULL |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100005 |   10006 | B+    |     NULL |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100005 |   10006 | B+    |     NULL |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100005 |   10006 | B+    |     NULL |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100005 |   10006 | B+    |     NULL |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100005 |   10006 | B+    |     NULL |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100005 |   10006 | B+    |     NULL |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100005 |   10006 | B+    |     NULL |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100005 |   10006 | B+    |     NULL |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100007 |   10007 | F     |        4 |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100007 |   10007 | F     |        4 |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100007 |   10007 | F     |        4 |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100007 |   10007 | F     |        4 |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100007 |   10007 | F     |        4 |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100007 |   10007 | F     |        4 |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100007 |   10007 | F     |        4 |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100007 |   10007 | F     |        4 |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100007 |   10007 | F     |        4 |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100007 |   10007 | F     |        4 |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100008 |   10007 | C-    |        0 |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100008 |   10007 | C-    |        0 |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100008 |   10007 | C-    |        0 |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100008 |   10007 | C-    |        0 |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100008 |   10007 | C-    |        0 |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100008 |   10007 | C-    |        0 |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100008 |   10007 | C-    |        0 |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100008 |   10007 | C-    |        0 |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100008 |   10007 | C-    |        0 |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100008 |   10007 | C-    |        0 |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100007 |   10008 | A-    |        0 |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100007 |   10008 | A-    |        0 |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100007 |   10008 | A-    |        0 |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100007 |   10008 | A-    |        0 |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100007 |   10008 | A-    |        0 |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100007 |   10008 | A-    |        0 |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100007 |   10008 | A-    |        0 |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100007 |   10008 | A-    |        0 |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100007 |   10008 | A-    |        0 |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100007 |   10008 | A-    |        0 |  
| 100000 | Tony      | Hawk    | CSCI  | CINF  |      40 |    1011 | 100000 |   11001 | D     |        4 |  
| 100001 | Mary      | Hawk    | CSCI  | CINF  |      35 |    1011 | 100000 |   11001 | D     |        4 |  
| 100002 | David     | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100000 |   11001 | D     |        4 |  
| 100003 | Catherine | Lim     | ITEC  | CINF  |      20 |    1017 | 100000 |   11001 | D     |        4 |  
| 100004 | Larry     | Johnson | ITEC  | NULL  |      66 |    1017 | 100000 |   11001 | D     |        4 |  
| 100005 | Linda     | Johnson | CINF  | ENGL  |      13 |    1015 | 100000 |   11001 | D     |        4 |  
| 100006 | Lillian   | Johnson | CINF  | ITEC  |      18 |    1015 | 100000 |   11001 | D     |        4 |  
| 100007 | Ben       | Zico    | NULL  | NULL  |      16 |    NULL | 100000 |   11001 | D     |        4 |  
| 100008 | Bill      | Ching   | ARTS  | ENGL  |      90 |    1018 | 100000 |   11001 | D     |        4 |  
| 100009 | Linda     | King    | ARTS  | CSCI  |     125 |    1018 | 100000 |   11001 | D     |        4 |  
+--------+-----------+---------+-------+-------+---------+---------+--------+---------+-------+----------+  
220 row  
  
R \* S in SQL:

SELECT R.\*, S.\*  
FROM R, S; -- note that there is no join condition.

**4.1.4 Union**

1. The set union operator.
2. Condition for R U S: R and S must be union compatible. Their relation schema must have compatible schema with the same structures. Each corresponding attribute must have the same types (domains).

R U S = {t | t ∈ R V t ∈ S}

**Example:**

Suppose StaffID and FacultyID are union compatible.

 πStaffID(Staff) U πFacultyID(Faculty)

**Example:** All information of students majoring in CSCI or ARTS.

σ(major='CSCI') (Student) U σ(major='ARTS') (Student)

or

σ(major='CSCI') V (major='ARTS') (Student)

+--------+-------+-------+-------+-------+---------+---------+  
| stuId  | fname | lname | major | minor | credits | advisor |  
+--------+-------+-------+-------+-------+---------+---------+  
| 100008 | Bill  | Ching | ARTS  | ENGL  |      90 |    1018 |  
| 100009 | Linda | King  | ARTS  | CSCI  |     125 |    1018 |  
| 100000 | Tony  | Hawk  | CSCI  | CINF  |      40 |    1011 |  
| 100001 | Mary  | Hawk  | CSCI  | CINF  |      35 |    1011 |  
| 100002 | David | Hawk  | CSCI  | ITEC  |      66 |    1011 |  
+--------+-------+-------+-------+-------+---------+---------+  
5 rows

**4.1.4 Difference (Minus)**

1. The set difference operator.
2. R - S: R and S must be union compatible.

R - S = {t | t ∈ R and not (t ∈ S)}

or

R - S = {t | t ∈ R, t ∉ S}

**Example:** Information of all students majoring in CSCI but those not taken credits less than 40.

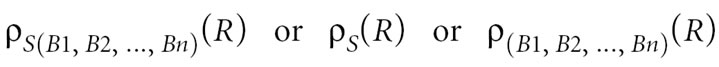
σmajor='CSCI'(Student) - σcredit <40 (Student)

+--------+-------+-------+-------+-------+---------+---------+  
| stuId  | fname | lname | major | minor | credits | advisor |  
+--------+-------+-------+-------+-------+---------+---------+  
| 100000 | Tony  | Hawk  | CSCI  | CINF  |      40 |    1011 |  
| 100002 | David | Hawk  | CSCI  | ITEC  |      66 |    1011 |  
+--------+-------+-------+-------+-------+---------+---------+  
2 rows  
  
Note that this is the same as:

σmajor='CSCI' and credit >=40(Student)

**4.1.6 Rename**

1. Rename the names of selected attributes in a relation.
2. Maybe used to rename attributes before a set operation.
3. Notation in Elmarsi (a popular db textbook):



* A better notation includes the original name and the new name.

Example:

ρ(FacultyId, department <- FacId, deptCode) (Faculty)  
  
+-----------+----------+----------+------------+---------------------+  
| facultyId | fname    | lname    | department | rank                |  
+-----------+----------+----------+------------+---------------------+  
|      1011 | Paul     | Smith    | CSCI       | Professor           |  
|      1012 | Mary     | Tran     | CSCI       | Associate Professor |  
|      1013 | David    | Love     | CSCI       | NULL                |  
|      1014 | Sharon   | Mannes   | CSCI       | Assistant Professor |  
|      1015 | Daniel   | Kim      | CINF       | Professor           |  
|      1016 | Andrew   | Byre     | CINF       | Associate Professor |  
|      1017 | Deborah  | Gump     | ITEC       | Professor           |  
|      1018 | Art      | Allister | ARTS       | Assistant Professor |  
|      1019 | Benjamin | Yu       | ITEC       | Lecturer            |  
|      1020 | Katrina  | Bajaj    | ENGL       | Lecturer            |  
|      1021 | Jorginlo | Neymar   | ACCT       | Assistant Professor |  
+-----------+----------+----------+------------+---------------------+  
11 rows

* The basic set of operations is complete. Other relational algebra operations can be derived from them.

**4.2. Common Derived Operations**

**4.2.1 Theta-join**

1. Allow the application of condition on Cartesian product.
2. There are still redundant data on common attributes.
3. Allow the query engine to throw away tuples not in the result immediately.
4. Conceptually, a Cartesian Product followed by a selection Θ.
5. Not usually used.

R1 |x|ΘR2 = σΘ(R1 x R2)

**Example:** All related information of students with 70 or more credits and a grade A or better in some courses.

Student |x|(credits >= 70 and grade = 'A') Enroll

+--------+-------+-------+-------+-------+---------+---------+--------+---------+-------+----------+  
| stuId  | fname | lname | major | minor | credits | advisor | stuId  | classId | grade | n\_alerts |  
+--------+-------+-------+-------+-------+---------+---------+--------+---------+-------+----------+  
| 100008 | Bill  | Ching | ARTS  | ENGL  |      90 |    1018 | 100000 |   10000 | A     |        0 |  
| 100008 | Bill  | Ching | ARTS  | ENGL  |      90 |    1018 | 100000 |   10001 | A     |        2 |  
| 100008 | Bill  | Ching | ARTS  | ENGL  |      90 |    1018 | 100004 |   10003 | A     |        0 |  
| 100008 | Bill  | Ching | ARTS  | ENGL  |      90 |    1018 | 100006 |   10005 | A     |     NULL |  
| 100009 | Linda | King  | ARTS  | CSCI  |     125 |    1018 | 100000 |   10000 | A     |        0 |  
| 100009 | Linda | King  | ARTS  | CSCI  |     125 |    1018 | 100000 |   10001 | A     |        2 |  
| 100009 | Linda | King  | ARTS  | CSCI  |     125 |    1018 | 100004 |   10003 | A     |        0 |  
| 100009 | Linda | King  | ARTS  | CSCI  |     125 |    1018 | 100006 |   10005 | A     |     NULL |  
+--------+-------+-------+-------+-------+---------+---------+--------+---------+-------+----------+  
8 rows in set (0.01 sec)  
  
  
 **4.2.2 Equi-join**

1. Theta-join where the condition involves only equality comparisons.
2. There are still redundant data on common attributes.
3. Common attributes are attributes that have the same names. The attributes may not have the same meaning.
4. Not usually used.

**Example:**

Student |x| (Student.StuId = Enrol.StuId) Enroll

+--------+---------+---------+-------+-------+---------+---------+--------+---------+-------+----------+  
| stuId  | fname   | lname   | major | minor | credits | advisor | stuId  | classId | grade | n\_alerts |  
+--------+---------+---------+-------+-------+---------+---------+--------+---------+-------+----------+  
| 100000 | Tony    | Hawk    | CSCI  | CINF  |      40 |    1011 | 100000 |   10000 | A     |        0 |  
| 100000 | Tony    | Hawk    | CSCI  | CINF  |      40 |    1011 | 100000 |   10001 | A     |        2 |  
| 100000 | Tony    | Hawk    | CSCI  | CINF  |      40 |    1011 | 100000 |   10002 | B+    |        1 |  
| 100000 | Tony    | Hawk    | CSCI  | CINF  |      40 |    1011 | 100000 |   10003 | C     |        0 |  
| 100000 | Tony    | Hawk    | CSCI  | CINF  |      40 |    1011 | 100000 |   10004 | A-    |        1 |  
| 100000 | Tony    | Hawk    | CSCI  | CINF  |      40 |    1011 | 100000 |   11001 | D     |        4 |  
| 100001 | Mary    | Hawk    | CSCI  | CINF  |      35 |    1011 | 100001 |   10000 | NULL  |     NULL |  
| 100001 | Mary    | Hawk    | CSCI  | CINF  |      35 |    1011 | 100001 |   10001 | A-    |        0 |  
| 100002 | David   | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100002 |   10000 | B-    |        3 |  
| 100002 | David   | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100002 |   10002 | B+    |        2 |  
| 100002 | David   | Hawk    | CSCI  | ITEC  |      66 |    1011 | 100002 |   10003 | D     |        4 |  
| 100004 | Larry   | Johnson | ITEC  | NULL  |      66 |    1017 | 100004 |   10003 | A     |        0 |  
| 100004 | Larry   | Johnson | ITEC  | NULL  |      66 |    1017 | 100004 |   10004 | B+    |     NULL |  
| 100005 | Linda   | Johnson | CINF  | ENGL  |      13 |    1015 | 100005 |   10003 | NULL  |     NULL |  
| 100005 | Linda   | Johnson | CINF  | ENGL  |      13 |    1015 | 100005 |   10004 | A-    |        0 |  
| 100005 | Linda   | Johnson | CINF  | ENGL  |      13 |    1015 | 100005 |   10005 | A-    |        0 |  
| 100005 | Linda   | Johnson | CINF  | ENGL  |      13 |    1015 | 100005 |   10006 | B+    |     NULL |  
| 100006 | Lillian | Johnson | CINF  | ITEC  |      18 |    1015 | 100006 |   10004 | C+    |     NULL |  
| 100006 | Lillian | Johnson | CINF  | ITEC  |      18 |    1015 | 100006 |   10005 | A     |     NULL |  
| 100007 | Ben     | Zico    | NULL  | NULL  |      16 |    NULL | 100007 |   10007 | F     |        4 |  
| 100007 | Ben     | Zico    | NULL  | NULL  |      16 |    NULL | 100007 |   10008 | A-    |        0 |  
| 100008 | Bill    | Ching   | ARTS  | ENGL  |      90 |    1018 | 100008 |   10007 | C-    |        0 |  
+--------+---------+---------+-------+-------+---------+---------+--------+---------+-------+----------+  
22 rows

It is important to note the difference between names and meanings. Consider

student(stuId, ... advisorFacId, ..., createTime) and

faculty(facId, ..., createTime)

1. The attributes createTime in student and faculty have the same name, but different meaning.
   1. Student(createTime) is the time the student row is inserted into the student table.
   2. Faculty(createTime) is the time the faculty row is inserted into the faculty table
2. student(advisorFacId) and faculty(facId) have different names but the same meaning. In fact, student(advisorFacId) is a foreign key that references faculty(facId).

**4.2.3 Natural Join**

1. Remove redundant common attributes from equi-joins
   1. Equi-join on all common attributes.
   2. Projection to remove redundant common attributes.
2. Used very frequently to combine two tables.
3. If two relations do not share any common attributes, their natural join is the same as their Cartesian Product.

Let C1, C2, ... Cm be the common attributes of R and S.

R |x| S = πA1, A2, .. Al(σR.C1=S.C1,.., R.Cm=S.Cm(RxS)

where A1, A2, ... Al is the list of components in RxS except S.C1, S.C2,.. S.Cm.

**Example:**

The schema of R(A,B) |x| S(A,C) is ABC. The schema of R(A,B) x S(A,C) is {R.A, B, S.A, C}.

**Example:**

Student |x| Enroll:

+--------+---------+---------+-------+-------+---------+---------+---------+-------+----------+  
| stuId  | fname   | lname   | major | minor | credits | advisor | classId | grade | n\_alerts |  
+--------+---------+---------+-------+-------+---------+---------+---------+-------+----------+  
| 100000 | Tony    | Hawk    | CSCI  | CINF  |      40 |    1011 |   10000 | A     |        0 |  
| 100000 | Tony    | Hawk    | CSCI  | CINF  |      40 |    1011 |   10001 | A     |        2 |  
| 100000 | Tony    | Hawk    | CSCI  | CINF  |      40 |    1011 |   10002 | B+    |        1 |  
| 100000 | Tony    | Hawk    | CSCI  | CINF  |      40 |    1011 |   10003 | C     |        0 |  
| 100000 | Tony    | Hawk    | CSCI  | CINF  |      40 |    1011 |   10004 | A-    |        1 |  
| 100000 | Tony    | Hawk    | CSCI  | CINF  |      40 |    1011 |   11001 | D     |        4 |  
| 100001 | Mary    | Hawk    | CSCI  | CINF  |      35 |    1011 |   10000 | NULL  |     NULL |  
| 100001 | Mary    | Hawk    | CSCI  | CINF  |      35 |    1011 |   10001 | A-    |        0 |  
| 100002 | David   | Hawk    | CSCI  | ITEC  |      66 |    1011 |   10000 | B-    |        3 |  
| 100002 | David   | Hawk    | CSCI  | ITEC  |      66 |    1011 |   10002 | B+    |        2 |  
| 100002 | David   | Hawk    | CSCI  | ITEC  |      66 |    1011 |   10003 | D     |        4 |  
| 100004 | Larry   | Johnson | ITEC  | NULL  |      66 |    1017 |   10003 | A     |        0 |  
| 100004 | Larry   | Johnson | ITEC  | NULL  |      66 |    1017 |   10004 | B+    |     NULL |  
| 100005 | Linda   | Johnson | CINF  | ENGL  |      13 |    1015 |   10003 | NULL  |     NULL |  
| 100005 | Linda   | Johnson | CINF  | ENGL  |      13 |    1015 |   10004 | A-    |        0 |  
| 100005 | Linda   | Johnson | CINF  | ENGL  |      13 |    1015 |   10005 | A-    |        0 |  
| 100005 | Linda   | Johnson | CINF  | ENGL  |      13 |    1015 |   10006 | B+    |     NULL |  
| 100006 | Lillian | Johnson | CINF  | ITEC  |      18 |    1015 |   10004 | C+    |     NULL |  
| 100006 | Lillian | Johnson | CINF  | ITEC  |      18 |    1015 |   10005 | A     |     NULL |  
| 100007 | Ben     | Zico    | NULL  | NULL  |      16 |    NULL |   10007 | F     |        4 |  
| 100007 | Ben     | Zico    | NULL  | NULL  |      16 |    NULL |   10008 | A-    |        0 |  
| 100008 | Bill    | Ching   | ARTS  | ENGL  |      90 |    1018 |   10007 | C-    |        0 |  
+--------+---------+---------+-------+-------+---------+---------+---------+-------+----------+  
22 rows  
  
**Exercise:**

Let the cardinality of R(A,B) be 5 and the cardinality of S(A,C) be 6. What is the range of the cardinality of R(A,B) |x| S(A,C)?

**4.2.4 Other Joins** (Additional Materials)

1. Some other joins are left join, right join, outer join, inner join and semi-join.
2. They can be defined through relational algebra expressions based on the basic operations.
3. Look them up when needs arise. For example: <https://en.wikipedia.org/wiki/Relational_algebra>

**4.2.5 Division** (Additional Materials)

1. R / S
2. Condition: the domain of S is a proper subset of R.
3. Let the schemes of R, S and T be dom(R), dom(S) and dom(T) = dom(R) - dom(S) respectively.
4. R / S = {t | t in dom(T) and for all s ∈ S, there exist r ∈ R such that r = st}.
5. In term of basic RA operations, R / S = πR-S(R) - πR-S((πR-S(R) x S) - R))

**Example:**

Find the student id of all students who enrolled in all courses offered by the faculty '1014':

Stuid and classNumber information (who is enrolled in which class):

π(stuId, classId) (Enroll): rows added to Class.

+--------+---------+  
| stuId  | classId |  
+--------+---------+  
| 100000 |   10000 |  
| 100000 |   10001 |  
| 100000 |   10002 |  
| 100000 |   10003 |  
| 100000 |   10004 |  
| 100000 |   11001 |  
| 100001 |   10000 |  
| 100001 |   10001 |  
| 100002 |   10000 |  
| 100002 |   10002 |  
| 100002 |   10003 |  
| 100004 |   10003 |  
| 100004 |   10004 |  
| 100005 |   10003 |  
| 100005 |   10004 |  
| 100005 |   10005 |  
| 100005 |   10006 |  
| 100006 |   10004 |  
| 100006 |   10005 |  
| 100007 |   10007 |  
| 100007 |   10008 |  
| 100008 |   10007 |  
+--------+---------+  
22 rows

Classes offered by faculty '1014':

π(classId) (σ(facId='1014) (Class)):

+---------+  
| classId |  
+---------+  
|   10003 |  
|   10004 |  
+---------+  
2 rows

Solution:

π(stuId, classId) (Enroll) / π(stuId, classId) (Enroll):

+--------+  
| stuId  |  
+--------+  
| 100000 |  
| 100004 |  
| 100005 |  
+--------+  
3 rows

**4.3 Query Optimization**

* Since RA is operational and thus algorithmic, there are multiple solutions with varying performance.
* Some heuristics for constructing effective RA solutions:
  1. Minimize the number of RA operations.
  2. Minimize the sizes of the intermediate results.
* In SQL:
  1. A SQL query execution plan breaks down a query into basic execution steps (based on RA).
  2. A SQL optimizer selects one from a list of execution plans.
  3. In MySQL, use the EXPLAIN statement to obtain information about the execution plan: <https://dev.mysql.com/doc/refman/8.1/en/explain-output.html>.

**4.4 Epilog**

Some shortcomings of Relational Algebra:

1. Cannot navigate tuples.
2. Cannot deal with recursion.
   1. e.g., for the relation Employee(SSN, Supervisor\_SSN, ...), find all supervisors (direct or indirect).
   2. May extend to logical databases, e.g. Datalog.
3. No group functions.
   1. e.g., Show the available total quantities of all parts.
4. Operations are too simple, resulting in long sequences.

**5. Relational Calculus  
  
5.1 Review**

* Non-procedural, *declarative*, and high level.
* Two kinds:
  1. Domain Relational Calculus (DRC)
  2. Tuple Relational Calculus (TRC)
* Results specified by the *set builder form*: {s | cond(s)}
* cond(s) is known as a *formula*.
* Constructs:
  1. Variables:
     1. TRC: tuples (bound to tuples): e.g,. s, t, student, class, etc.
     2. DRC: Attributes (bound to domain value): e.g., a, b, c, stuId, fname, etc.
     3. RC's variable is sometime known as 'dummy variable'.
  2. Constants: string, int, etc., E.g., 12, 'csci', 3.7.
  3. Comparison operators: <, <, =, etc.
  4. Boolean operators: and (conjunction, ∧ or just ,), or (disjunction ∨), not (¬), implies (⇒), etc.
  5. Membership functions: belongs to, ∈, not belongs to, ∉, etc.
  6. Quantifiers: there exists (existential, ∃), for all (universal ∀).

**5.2 More RC**

* An *atom* can be thought of as a simple Boolean expression:
  + e ∈ R, or
  + x op y where x and y are attributes or constants, and op is a comparison operation.
* A *formula* is either an atom or formula connected by Boolean operator or qualifiers.
* A formula that is not an atom can be thought of a compound Boolean expression.
* A variable is *bound* if it appears in qualifier expressions. Otherwise, it is a *free* variable.
* Free variables can only appear in the LHS of |.
* All RA expressions can be expressed in RC.
* RA and RC have the same expressive power.
* Any query language that can express all RA is known to be relational complete.
* Relational Calculus expressions need to be *safe*: results should be a*finite* set of tuples.
* Care should be taken especially for the negation operation. E.g. {s |¬ (s ∈ Student) } is unsafe.
* For a given implementation of relational calculus:
  + There may be restrictions in supported constructs.
  + There may be certain *canonical* requirements: e.g. conjunction (joined by the and operator) of disjunction (joined by the or operator).

**Example:**

{i | i ∈ I ∧ i % 2 =0}  
{i | i ∈ I, i % 2 =0} -- set builder form.

{t | ∃r ∈R, r.firstname = t.firstname, r.lastname = t.lastname}

* t is a *free* variable.
* It will have two attributes: t.firstname and t.lastname.

Alternatively, we can use the set builder form in the LHS before |:

{(r.firstname, r.lastname) | r ∈ R}

R(A,B,C,D) / S(C,D)

{(a,b) | (∀(c,d) ∈ S) (a,b,c,d) ∈ R)}

***Exercises:***

How do you use RC to implement RA operations?

**5.3 TRC**

* The variables in TRC are tuples.
* SQL is based on TRC.

**5.4. DRC**

* The variables in DRC are attributes (domain values).
* Query By Example (QBE) is based on DRC.

**Exercise:**

Work on some of the query questions listed in the [toyu Query Exercise](https://dcm.uhcl.edu/yue/courses/joinDB/Spring2024/notes/rarc/toyu_Ex.pdf) in DRC and TRC.

**Transforming UML Class Diagrams to Relational Models**

by K. Yue

**1. Transforming OO Model to the Relational Model**

* Once a *conceptual* OO data model is constructed, it needs to be mapped for implementation in the selected *logical* database model. See [IntroDataModeling.html](https://dcm.uhcl.edu/yue/courses/joinDB/Spring2024/notes/model/IntroDataModeling.html).
* If relational DB is used, the mapping will be from OO (classes, attributes, associations, etc.) to relational schema (relations, attributes, keys, etc.)
* Note that the relational model and the OO model are *very different*, even though diagrams representing the two models look similar (there are only finite number of common suitable shapes).
* Computer-Aided Software Engineering (CASE) tools or DB modeling tools may provide varying degrees of facilities for automatic generating relational schema and corresponding SQL statements.
* However, it is important to understand the mechanism a tool uses to generate the relational schema and make adjustment if needed.

**1.1 Model Transformation**

The problem is: Source Model --> Target Model.

* Source Model: UML
  1. Basic elements:
     1. class
     2. Attribute: can be multi-valued.
     3. Association
  2. Secondary elements:
     1. Object
     2. Multiplicity
     3. Data type
     4. Default value
     5. Constraint
     6. Stereotype for RDB extensions: e.g., candidate keys, primary keys, unique, derived, nullability, etc.
     7. ...
* Target Model: Relational Model
  1. Basic elements:
     1. Relation
     2. Attribute (column/field): should be single-valued (atomic).
  2. Secondary elements:
     1. Row (tuple)
     2. Data type
     3. Nullability
     4. Constraint
     5. Candidate key
     6. Primary key
     7. Foreign key
     8. Index
     9. ...
* There are only two basic elements in the targeted model to consider. Data in the relational model can be stoed only in two ways:
  1. relation:
     1. more flexible,
     2. can hold attributes to store a collection of logically related data
     3. more complex.
  2. attribute:
     1. should be single-valued (if good design, i.e., the first normal form, is to be assured.)
     2. less complex.
     3. Thus, if attributes are sufficient in model transformation, they are preferred.
* Different RDBMS provide different features.
  + 1. Thus, the targeted RDB model is not universal.
    2. It is necessary to define vendor-specific transformation rules.

**2. Transformation Rules**

* We present a set of mapping rules below. It is not meant to be complete or universal.
* As alternative examples of transformation rules:
  + A relative simple one: <http://web.fe.up.pt/~ssn/2010/lbaw/slides/lbaw-uml2rel.eng.pdf> a
  + A more elaborated one based on agile methodology: [http://www.agiledata.org/essays/mappingObjects.html](http://www.agiledata.org/essays/mappingObjects.html#Figure2IncludingShadowInformation).
* Do not mechanically follow these rules. Instead, understand the rationale behind the rules and adapt.
* All OO model details should be implemented in the targeted model in some ways.
  + database level: preferred.
  + middle layer level
  + application level

**2.1 Classes**

**C1**. A class C is mapped or transformed to a relation RC.

1. Relations may later be merged and/or reorganized in design refinement and performance tuning.
2. The relation may use the same name as the class.

**Rationale:**

1. A class is a logical unit for encapsulating related data and a relation has the same property.

**2.2 Attributes**

**Basic:**

**ATT1.** Map all single-valued attributes (with simple data types) of a class C as attributes of RC, the relation for the class C.

A diagram of a computer code

Description automatically generated with medium confidence

Transformed to:

A close-up of a computer screen

Description automatically generated

**ATT2**. For each multi-valued attribute A (Hobbies) of the class C (Member), create a *new* relation RCA (Hobby) containing the attribute A (Hobby) and the primary key, RCId (Mid), of the relation RC (Member) (which implements the class C).

1. (RCId, A): (MemberId, Hobby) is a composite candidate key.
2. RCId is a foreign key referencing RC(RCId)
3. A surrogate key, such as RCA\_Id, may be created to serve as the simple primary key.
4. The name of RCA should be meaningfully selected.

***Example:***

Multi-valued attributes: consider the class Member with the following attributes:

1. Member\_Id: <<PK>>
2. Member\_Screen]Name <<unique>>
3. Hobbies[0..\*]
4. Medals[0..\*]

A yellow card with black text

Description automatically generated

RC: Member(MemberId, ScreenName

Example object:

MemberId: M007  
 ScreenName: Bun  
 Hobbies: eating, sleeping, daydreaming  
 Medals: green, red, blue

**Member:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MemberId** | **ScreenName** | **~~Hobbies?~~** |  |  |
| M007 | Bun | ~~eating, sleeping, daydreaming~~  ~~(not atomic)~~ |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Member2: limited to 3; not very efficient

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MemberId** | **ScreenName** | **Hobby1** | **Hobby2** | **Hobby3** |
| M007 | Bun | eating | Sleeping | daydreaming |
|  |  |  |  |  |
|  |  |  |  |  |

**RCA: Hobby(MemberId, Hobby, HobbyId)**

|  |  |  |
| --- | --- | --- |
| **MemberId (FK)** | **Hobby** | **HobbyId** |
| M007 | Eating | 1 |
| M007 | Sleeping | 2 |
| M007 | Daydreaming | 3 |
| M117 | Eating | 4 |

**Medal**

|  |  |  |
| --- | --- | --- |
| **MemberId (FK)** | **Medal** | **MedalId** |
| M007 | Green | 1 |
| M007 | Red | 2 |
| M007 | blue | 3 |
| M117 |  | 4 |

Reasonable relation schema: three relations used

1. Member(MemberId, ScreenName):
   1. CK: [1] MemberId, [2] ScreenName
2. Hobby(HobbyId, MemberId, Hobby):
   1. CK: [1] HobbyId, [2] MemberId, Hobby
   2. FK: [1] MemberId references Member(MemberId)
   3. A surrogate key, HobbyId, is created as the primary key.
3. Medal(MedalId, MemberId, Medal):
   1. CK: [1] MedalId, [2] MemberId, Medal.
   2. FK: [1] MemberId references Member(MemberId)
   3. A surrogate key, MedalId, is created as the primary key.

**Secondary:**

**ATT3.** If the multiplicity of an attribute is specified, to handle the case of 0:

1. If 0 is allowed in the UML model (e.g,. 0..1, 0..\* in the UML class diagram), the attribute is nullable. Add the NULL specifier in the column definition in the RDBMS. (NULL is usually the default)
2. If 0 is not allowed, add the NOT NULL specifier in the column definition.

**ATT4.**The default value of an attribute may be directly implemented in SQL DDL.

**ATT5.** Data type mapping should be handled adequately, effectively and consistently.

1. In later modeling phases, one may use SQL data types of the targeted DBMS in the class diagram.
2. If available, consider using user-defined data types in the targeted DBMS.

***Example:***

UML for toyu.student:

A diagram of a student

Description automatically generated

Relational schema (in HW assignment format):

|  |  |
| --- | --- |
| **Relation** | Student(StuId, fname, lname, major, minor, ach, advisor) |
| Candidate Keys | [1] StuId |
| Foreign Keys | [1] major references department(deptCode), [2] minor references department(deptCode), [3] advisor references faculty(facId) |
| Nullable Attributes | major, minor, advisor, ach |
| Non-nullable Attributes | stuId, fname, lname |
| Notes |  |

Relational schema in SQL (with more implementation details):

CREATE TABLE IF NOT EXISTS Student    (  
    stuId        INT NOT NULL,  
    fname        VARCHAR(30) NOT NULL,  
    lname        VARCHAR(30) NOT NULL,  
    major        CHAR(4) NULL,  
    minor        CHAR(4) NULL,  
   -- ach: accumulated credit hours, including transferred credits.  
    ach          INTEGER(3) NULL DEFAULT 0,  
    advisor      INT NULL,  
    CONSTRAINT Student\_stuId\_pk PRIMARY KEY(stuId),  
   -- an artificial example of a CHECK constraint.  
    CONSTRAINT Student\_ach\_cc CHECK ((ach>=0) AND (ach < 250)),  
    CONSTRAINT Student\_major\_fk FOREIGN KEY (major)  
        REFERENCES Department(deptCode) ON DELETE CASCADE,  
    CONSTRAINT Student\_minor\_fk FOREIGN KEY (minor)  
        REFERENCES Department(deptCode) ON DELETE CASCADE,  
    CONSTRAINT Student\_advisor\_fk FOREIGN KEY (advisor)  
        REFERENCES Faculty(facId)  
);

***Example:***

Multi-valued attributes: consider the class Member with the following attributes:

1. Member\_Id: <<PK>>
2. Member\_Screen]Name <<unique>>
3. Hobbies[0..\*]
4. Medals[0..\*]

A yellow card with black text

Description automatically generated

Reasonable relation schema: three relations used

1. Member(MemberId, ScreenName):
   1. CK: [1] MemberId, [2] ScreenName
2. Hobby(HobbyId, MemberId, Hobby):
   1. CK: [1] HobbyId, [2] MemberId, Hobby
   2. FK: [1] MemberId references Member(MemberId)
   3. A surrogate key, HobbyId, is created as the primary key.
3. Medal(MedalId, MemberId, Medal):
   1. CK: [1] MedalId, [2] MemberId, Medal.
   2. FK: [1] MemberId references Member(MemberId)
   3. A surrogate key, MedalId, is created as the primary key.

All columns in the tables above are not nullable.

**ATT6.** A single-valued attribute of a composite data type (such as set, list, array) can be mapped in various ways.

1. If there is an comparable composite data type in the targeted DBMS, it can be implemented as an attribute of that data type in the relation.
   * The relation will no longer be in the first normal form.
   * Care should be taken in handling the difference in data type mapping.
2. Otherwise, regard the attribute as a multi-valued attribute and apply rule ATT2.

**ATT7.** For a derived attribute A: <<derived>> or \

1. Computed or derived values.
2. Not independent.
3. It can be implemented and stored as an attribute of the relation.
   * Mechanisms, such as triggers or stored procedures, should be used to ensure data consistency. That is the derived column should be consistent with the data that derives its value.
4. It may not be stored as a column directly in any relation.
   * Mechanisms, such as virtual columns, views or stored functions, may be used to provide standard access to the derived attributes.

***Example:***

A class Rectangle has three attributes:

* Length
* Width
* \Area or <<derived>>: derived.

A rectangular yellow rectangle with black text

Description automatically generated

What may the relational schema look like?

One solution:

Rectangle(RectangleId, Length, Width)

with a view Rect define as

select RectangleId, Length, Width, Length \* Width as Area  
from Rectangle;

Alternatively, using*virtual* column: a column that is computed in real-time by an expression and not stored.

CREATE or replace TABLE rectangle (  
  width DOUBLE,  
  height DOUBLE,  
  area DOUBLE AS (width \* height) virtual  
);

**2.3 Keys and Constraints**

**KC1.** If a relation R implements a class C or an association (class) A, and C or A has candidate keys K's, set **all** K's as candidate keys of R.

**KC2.**If a relation R implements a class C or an association (class) A, and C or A has no candidate key, create a surrogate primary key K for R.

1. This is needed as every relation must have at least one candidate key.

**KC3.** A candidate key can be implemented by using the 'unique' and non-null constraint together in SQL.

**KC4.** If many CK but no primary key are specified, select a candidate key as the primary key and set it accordingly in the relation.

***Example:***

for the class Department in toyu:

A yellow sign with black text

Description automatically generated

Relational schema (in HW assignment format):

|  |  |
| --- | --- |
| **3** | Department(deptCode, deptName, schoolCode, numStaff) |
| Candidate Keys | [1] deptCode, [2] deptName |
| Foreign Keys | [1] schoolCode references School(schoolCode) |
| Nullable Attributes | schoolCode, numStaff |
| Non-nullable Attributes | deptCode, deptName |
| Notes |  |

Relational schema in SQL (with more implementation details):

CREATE TABLE IF NOT EXISTS Department (  
    deptCode    CHAR(4) NOT NULL,  
    deptName    VARCHAR(30) NOT NULL*,*  
    schoolCode  CHAR(3) NULL,  
    numStaff  TINYINT NULL,  
    CONSTRAINT Department\_deptCode\_pk PRIMARY KEY (deptCode),  
   -- alternate keys: [1] deptName    
    CONSTRAINT Department\_name\_ck UNIQUE (deptName)*,*  
    CONSTRAINT Department\_schoolCode\_fk FOREIGN KEY (schoolCode)  
        REFERENCES School(schoolCode)  
);

**KC5.** For a stereotype:

1. Some may be directly implemented in SQL DDL, e.g., PK, CK, unique, etc.
2. Otherwise, it is necessary to consider where and how it is implemented.

**2.4 Associations**

**A1**. For a many to one association between C1: faculty (the class with the one multiplicity) and Cm: class, add a column R1\_Id (FacId) into the relation Rm, class (which implements Cm).

1. Assume that R1\_Id (FacId) is the primary key of the relation R1, faculty (which implements C1).
2. R1\_Id (facId) is a foreign key of Rm (class) referencing R1(R1\_Id): Faculty(facId).
3. The name R1\_Id may be renamed.
4. R1\_Id is not null in Rm iff (if and only if) 0 is not allowed (i.e., 1..1) for C1.
5. Any single valued attribute of the association is mapped to a column in Rm.
6. If you have composite or multi-valued attributes of the relationship, you should consider promoting the association to an association class or a regular class in your UML class diagram.

***Example:***

For:

A diagram of a diagram

Description automatically generated

We have the three numbered associations implemented by the three foreign keys below.

|  |  |
| --- | --- |
| **4** | Faculty(facId, fname, lname, *deptCode*, rank) |
| Candidate Keys | [1] facId |
| Foreign Keys | [1] deptCode references Department(deptCode) |
| Nullable Attributes | rank, *deptCode* |
| Non-nullable Attributes | facId, fname, lname |
| Notes |  |
| **6** | Class(classId, *courseId*, semester, year, *facId*, room) |
| Candidate Keys | [1] classId |
| Foreign Keys | [1] courseId references Course(courseId), [2] facId references Faculty(facId) |
| Nullable Attributes | room |
| Non-nullable Attributes | classId, *courseId*, semester, year, *facId* |
| Notes |  |

**A2.** For a many to many association (including association classes) between classes CA and CB, create a*new* relation RAB(RA\_Id, RB\_Id).

1. (RA\_Id, RB\_Id) is a candidate key.
2. RA\_Id references RA(RA\_Id) as a foreign key.
3. RB\_Id references RB(RB\_Id) as a foreign key.

A diagram of a group

Description automatically generated

1. An additional surrogate key, such as RAB\_Id, can be created.

***Example:***

For:

A diagram of a program

Description automatically generated

We have:

|  |  |
| --- | --- |
| **8** | Enroll(stuId, classId, grade, n\_alerts) |
| Candidate Keys | [1] stuId, classId |
| Foreign Keys | [1] stuId references Student(stuId), [2] classId references Class(classId), [3] grade references Grade(grade) |
| Nullable Attributes | grade, n\_alerts |
| Non-nullable Attributes | stuId, classId |

**A3.** For a one to one association between classes CA and CB, there are several options:

1. Treat CA as C1 and CB as Cm and apply A1.
2. Treat CA as Cm and CB as C1 and apply A1.
3. Merge the two relations RA and RB into one. (In this case, you may want to refactor the class diagram.)

**A4.** For any n-ary association (n>2), a new relation is needed.

1. You should consider using binary associations instead.
2. A ternary association can be modeled as a class with three binary associations with the participating classes in the ternary association.

***Example:***

Consider the ternary association between the classes Supplier, Part, and Warehouse with an association attribute quantity.

It can reasonably be replaced by a new class and three binary associations.

A diagram of a supply chain

Description automatically generated

Supply(SupplyId, SupplierId, PartId, WarehouseId, Quantity):

1. CK: [1] SupplyId, [2] SupplierId, PartId, WarehouseId
2. FK: [1] SupplierId references Supplier(SupplierId), [2] PartId references Part(PartId), [3] WarehouseId references Warehouse(WarehouseId).
3. All attributes in Supply is not nullable.

***Example:***

Toyu UML diagram:

A diagram of a computer program

Description automatically generated with medium confidence

toyu schema: [toyu\_schema.pdf](https://dcm.uhcl.edu/yue/courses/joinDB/Spring2024/notes/model/toyu_schema.pdf)

toyu SQL: [createtoyu.sql.txt](https://dcm.uhcl.edu/yue/courses/joinDB/Spring2024/notes/query/createtoyu.sql.txt)