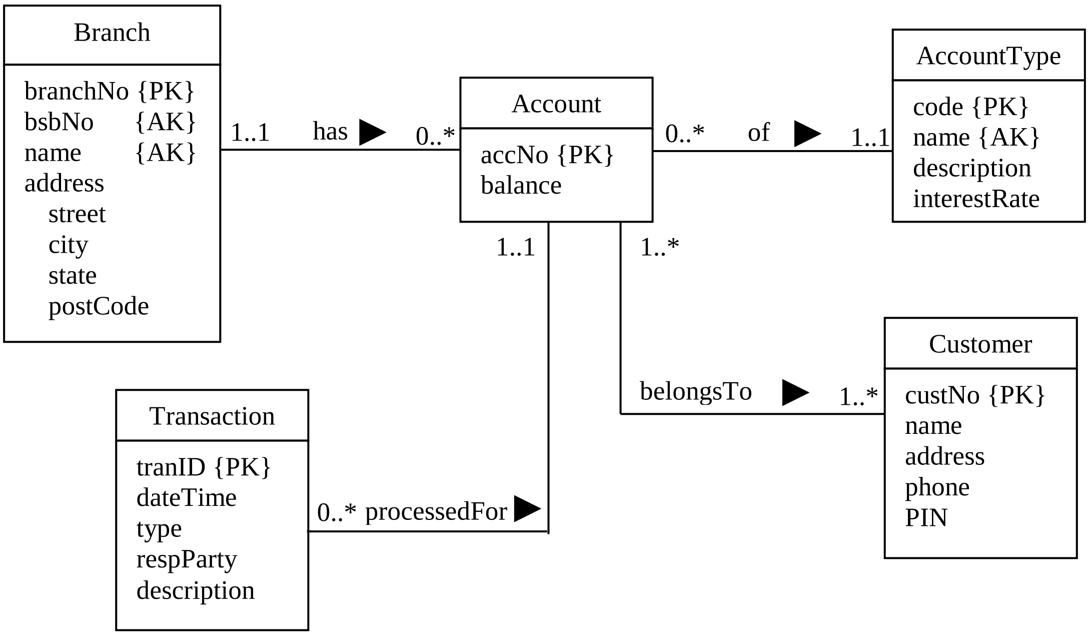
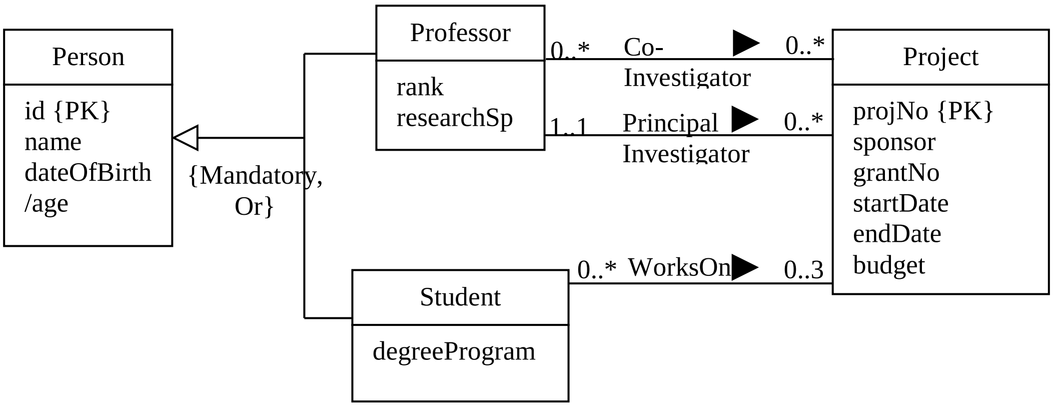
COMP1140 – S2, 2018
Prac/Tut Week 4
School of EEC, University of Newcastle
Callaghan, Australia

1. Map the following EER models to relational model

a. Bank Database



b. University Database



2. SQL Exercises

In the following SQL exercises, basic syntaxes for SQL operations are given. You are encouraged to refer to online documentation for examples of full syntaxes.

a. Log onto SQL Server and create the following relational database schema

Relational Schema:

Student(stdNo, login, lastName, givenNames, programCode**)**
Primary Key stdNo
Alternate Key login

Course(courseID, cName, credits**)**
Primary Key courseID
Alternate Key cName

Semester(semesterID, semester, year**)**
Primary Key semesterID
Alternate Key semester, year

Register(stdNo, courseID, semesterID, grade, mark**)**
Primary Key stdNo, courseID, semesterID
Foreign Key stdNo **references** Student(stdNo) **on update cascade on delete no action**
Foreign Key courseID **references** Course(courseID) **on update cascade on delete no action**
Foreign Key semesterID **references** Semester(semesterID) **on update cascade on delete no action**

Domains

stdNo: fixed length character string, length 5
login: fixed length character string, length 6
lastName: variable length character string, length 25
givenNames: variable length character string, length 50
programCode: fixed length character string, length 4
courseID: fixed length character string, length 8
cName: variable length character string, length 25
credits: integer, range 0-200
semesterID: integer, positive
semester: integer, 1-4
year: integer, 2000 – 9999
grade: fixed length character string, length 2
mark: decimal value, 0.00 – 100.00

Default values

credits: 20
marks: 0.0

Notes:

- **Domains:**
  Select appropriate data types for the domains. You can get a list of T-SQL data types available from Books Online. You may use CHECK constraints to specify ranges for domains.

- **UNIQUE and NOT NULL:**
  You can specify alternate keys using the UNIQUE constraint which ensures that duplicate values are not stored in the specified columns. However in SQL, NULLs are distinct and therefore there could be multiple NULL values that appear in a UNIQUE column. You need to specify both UNIQUE and NOT NULL constraints to avoid NULLs.

- **DEFAULT:**
  DEFAULT constraint allows a value to be assigned to a column when user does not specify it in an INSERT statement.
  E.g.
  ```
  age    integer        DEFAULT  16
  ```

- **Table Constraints:**
  In T-SQL, constraints on tables can be specified during table creation at
  1. At the column definition (only when the constraint is for the column only)

     E.g.
     ```
         dateOfBirth   datetime       NOT NULL
     ```

  2. At the end of table definition with or without a constraint name (required for constraints spanning multiple columns).

     E.g.
     ```
     CREATE TABLE Person
     (
        …,
     CHECK (age BETWEEN 0 AND 120),
     CONSTRAINT pkPerson PRIMARY KEY (lname, fname)
     )
     ```

- **Foreign Keys and Referential Integrity Constraints:**
  In T-SQL, foreign keys can be specified as follows:
  1. Along with the column definition:
     For example,
     ```
         stdNo CHAR(6) REFERENCES Student(StdNo)
     ```

  2. At the end of table definition with or without a constraint name:
     For example,
     (i.) Without constraint name:
     ```
     FOREIGN KEY(courseID) REFERENCES Course(courseID))
     ```

(ii.) With constraint name:

```
CONSTRAINT    fkRegisterStd    FOREIGN KEY(stdNo)
REFERENCES Student(stdNo),
```

Referential integrity ensures that the foreign key value references an **existing** candidate key value. The existing value may be modified or deleted on the referenced table by an UPDATE or DELETE statement. There are four types of actions that can be specified for the foreign key value:

- NO ACTION – This is the default and no action is taken. This disallows the UPDATE or DELETE statement from completing as the referential integrity constraint is violated.

- CASCADE - This will update/delete corresponding rows from the referencing table if that row is updated/deleted from the parent table.

- SET NULL - All the values that make up the foreign key are set to NULL when the corresponding row in the parent table is updated/deleted.

- SET DEFAULT - All the values that make up the foreign key are set to their default values when the corresponding row in the parent table is updated. For this constraint to execute, all foreign key columns must have default definitions.

  For example,

  ```
  stdNo CHAR(6) REFERENCES Student(StdNo) ON UPDATE
  CASCADE ON DELETE NO ACTION
  ```

Save the script as Prac3.sql as this schema will be used for the next prac/tute session.