# ISYS1055/1057 Assignment 2 - Due 25th May 2016

## Submission

This assignment is due 23:59 Wednesday 25th May 2016. It is marked out of 100 points and is worth 30% of the course assessment. You should submit one PDF document with all answers together. You can use Microsoft Word or another word processing application to work on your assignment. At the end, convert it into PDF format. Do not submit Word file. if that option is not available on your system there are free pdf converters online you can utilise. e.g. http://convertonlinefree.com/

Late submissions of assignments will be penalised as follows. For every (up to) 24 hours late, a penalty of 10% (i.e. 10% out of total marks, not 10% out of your marks) applies. For assignments more than 5*24 hours late, 100% penalty applies.

This is an individual assignment, to be submitted electronically using the `Blackboard` facility. All code or other material that is not original must be fully credited. That is, any material that is copied or derived from another source must be clearly identified as such and the original author must be identified. Sometimes students assist each other with an assignment, but end up working together too closely, so that the students' separate solutions have significant parts in common; unless the solutions were developed independently, they are regarded as plagiarised.

Plagiarism is a very serious offence. Any submissions determined to be a result of plagiarism will be given zero marks for that assignment. In the event that a hurdle requirement is not met, this will result in the failure of the course. You should familiarise yourself with the University Academic Integrity Policy:

http://www.rmit.edu.au/academicintegrity

## Part A: Relational Database Design (30 marks)

Consider the following relational schema and answer questions below.

```
Person (Id, TFN, Name, Phone)
PolicyOwner (PersonId, PolicyNumber, TFN, Start-Date)
InsurancePolicy (PolicyNumber, AgentNumber, PremiumAmount,
CoverageAmount)
Agent (AgentNumber, Name, Phone, OfficeNumber)
```

`Id` is the primary key for the `Person` relation and `TFN` (Tax File Number) is another candidate key for the `Person` relation. This relation has two candidate keys.

<`PersonId, PolicyNumber`> is the primary key for the `PolicyOwner` relation and <`TFN, PolicyNumber`> is another candidate key for the `PolicyOwner` relation. This relation also has two candidate keys. `Start-Date` is the date when this policy was first created. `PersonID` is a foreign key referencing `Person.Id`. `TFN` is a foreign key referencing `Person.TFN`. `PolicyNumber` is a foreign key referencing `InsurancePolicy.PolicyNumber`.

`PolicyNumber` is the primary key for the `InsurancePolicy` relation. `AgentNumber` is a foreign key referencing `Agent.AgentNumber`.

`AgentNumber` is the primary key for the `Agent` relation.

1. List all of the non-trivial FDs in the `InsurancePolicy` relation, including those implied by a key. Is the `InsurancePolicy` relation in BCNF? Explain briefly your answer. Note that it is possible for several, different insurance policies to have the same agent assigned (that is, to have the same agent number listed in the `AgentNumber` attribute). Does this kind of redundancy cause any update anomalies? Can we normalise this relation in order to remove this redundancy?
2. List all of the non-trivial FDs in the `Person` relation, including those implied by a key. Is the `Person` relation in BCNF? Explain your answer. If the `Person` relation is not in BCNF, decompose the `Person` relation into relations that are in BCNF.
3. List all of the non-trivial FDs in the `PolicyOwner` relation, including those implied by a key. Which normal form is this relation in? If it's not in BCNF, decompose the `PolicyOwner` relation into BCNF.

## Part B: SQL (30 Marks)

`LibraryDB` is a database system that keeps track of information concerning the books and their circulation in an imaginary library.

**Disclaimer**: The data that populates the database are artificially constructed and by no means correspond to actual real-world data.

The schema for the LibraryDB database is given below.

```
borrow(transactionID, personID*, borrowdate, duedate, returndate)
author(authorID, firstname, middlename, lastname)
book_copy(bookID, bookdescID*)
book(bookdescID, title, subtitle, edition, voltitle, volnumber,
language, place, year, isbn, dewey, subjectID*)
borrow_copy(transactionID*, bookID*)
person(personID, firstname, middlename, lastname, address, city,
postcode, phonenumber, emailaddress, studentno, idcardno)
publisher(publisherID, publisherfullname)
written_by(bookdescID*, authorID*, role)
published_by(bookdescID*, publisherID*, role)
subject(subjectID, subjecttype)
```

The primary keys are underlined. The foreign keys are denoted by asterisks (*).

### Description of the schema

- `person` -- keeps track of the people who borrow books from the library. The attributes contain personal and contact information.
- `author` -- keeps track of personal information about authors.
- `publisher` -- keeps track of the publisher information. To make simple, most of the attributes have been truncated in the sample database.
- `subject` -- this relation keeps information about the subjects on which the library collection have books (such as Mathematics, Database, etc)
- `book` -- contains information about the books that are available in the library. Every book can have one or more physical copies in the collection. Each book can have one or more authors and it is published by one or more publishers.
- `book_copy` -- keeps track of the physical copies of the books in the library collection.
- `borrow` -- keeps track of the check-ins and check-outs of the books. Every transaction is done by one person, however may involve with one or more book copies. If there is no return date, it means the book has been checked out but not returned.

- `written_by` -- associates books with authors. A book may be associated with several authors and an author may be associated with several books. There is also an attribute 'role' that specifies the role of the author for the book (author/ editor/ translator/ etc).
- `published_by` -- associates publishers with books. There is an attribute 'role' here too.
- `borrow_copy` -- associates physical copies of books with a transaction. Members are allowed to borrow several books in a single transaction.

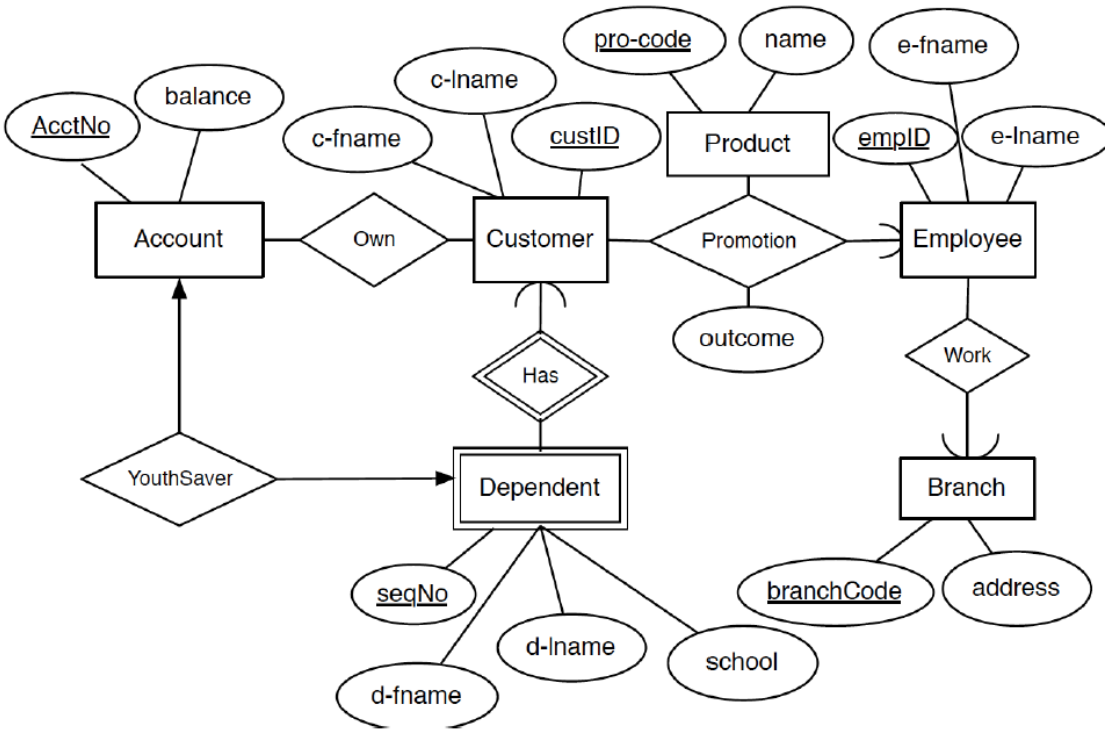
You can access a sample database instance of this library database system using Oracle SQL Developer (either on myDesktop or installed on your home computer). Please follow the instructions provided on Blackboard.

Write an SQL query for each task below. Each query is worth 6 marks.

1. Give the `bookdescID` of books by authors whose surname starts with “S” or by publishers whose name starts with “S”.
2. With which publisher(s) the author Alfred Aho published his book(s)? Display publishers' full names. You must not use any subquery.
3. Who are the authors that published books with the MC GRAW-HILL publisher? Display the firstname and lastname of these authors.
4. Display the first name and lastname of authors who wrote more than 3 books. Along with each name, display the number of books as well.
5. Display the title of the book which has the largest number of physical copies. If there are more than one book with the largest number of copies, show them all. Your query should show the number of copies along with the title.

## Part C: ER Model. (30 Marks)

Map the ER diagram below into a relational database schema. For each relation schema, underline the primary key and denote any foreign key with an asterisk (*).




## Part D. Research Questions (10 Marks).

1. The Library database schema in Part B assumes that all books borrowed in one transaction are to be returned together. However, this is an unreasonable assumption and quite contrary to the common practice across all forms of libraries. You can return books (borrowed together) separately. To accommodate this requirement, discuss what relation schemas need to be changed and give the new relation schemas.
2. Discuss if the Library database schema in Part B allows customers to extend their loans before the due date. If so, explain how loan extensions are likely handled and any limitations with the approach.