# Objectives:

INFO90002 Tutorial Week 4

This tutorial will cover:

1. Review of Normalisation concepts – 15 mins
2. Normalisation exercises – 35 mins
3. Optional extra exercises (homework) - approximately 50 minutes

**Key Concepts:**

***NOTE for students:*** *This is a brief summary of some of the concepts taught in the lecture on 'Normalisation'. The lectures contain detailed content related to these and many more concepts. These notes should be considered for quick revision,* ***not*** *a sole resource for the course material.*

## Anomalies

Consider the following instance of the relation Allocation (CourseNumber, Tutor, Room, Seats):

|  |  |  |  |
| --- | --- | --- | --- |
| **CourseNumber** | **Tutor** | **Room** | **Seats** |
| INFO20003 | Farah | Alice Hoy 109 | 30 |
| COMP10001 | Farah | EDS 6 | 25 |
| INFO30005 | Patrick | Sidney Myer G09 | 20 |
| COMP20005 | Alan | Sidney Myer G09 | 20 |

An **update** anomaly is a data inconsistency that results from data redundancy and partial update when one or more instances of duplicated data are updated but not all.

A **deletion** anomaly is an unintentional loss of certain attribute values due to the deletion of other data for other attributes.

An **insertion** anomaly is the inability to add certain attributes to a database due to absence of other attributes.

## Functional dependency

* Determinants
* Key and non-key attributes
* Partial functional dependency
* Transitive functional dependency

## Armstrong’s Axioms

* Reflexivity
* Augmentation
* Transitivity

# Normalisation and normal forms

Normalisation is a technique used to iteratively improve relations to remove undesired redundancy by decomposing relations and eliminating anomalies. The process is iterative and can be performed in stages generally referred to as Normal Forms. In First Normal Form (1NF), the relation is analysed and all repeating groups are identified to be decomposed into new relations. In Second Normal Form (2NF), all the partial dependencies are resolved/removed. The next stage is Third Normal Form (3NF) where all the transitive dependencies are removed.

归一化是一种通过分解关系和消除异常来迭代改进关系的技术，以消除不必要的冗余。这个过程是迭代的，可以按通常称为标准形式的阶段来执行。在第一范式(1NF)中，分析关系，并将所有重复组分解为新的关系。在第二范式(2NF)中，所有的部分依赖项都被解析/移除。下一个阶段是第三范式(3NF)，其中所有的传递依赖性都被移除。

# Exercises:

1. Consider the relation Diagnosis with the schema

Diagnosis (DoctorID,DocName, PatientID, DiagnosisClass) and the following functional dependencies:

DoctorID → DocName

DoctorID, PatientID → DiagnosisClass

Consider the following instance of Diagnosis:

|  |  |  |  |
| --- | --- | --- | --- |
| **DoctorID** | **DocName** | **PatientID** | **DiagnosisClass** |
| D001 | Alicia | P888 | Flu |
| D002 | John | P999 | Lactose intolerance |
| D003 | Jennifer | P000 | Flu |
| D002 | John | P111 | Fever |

1. Identify different anomalies that can arise from this schema using the above instance.
2. Consider a relation R (A, B, C, D) with the following FDs:

AB → C, AC → B, BC → A, B → D

The possible candidate keys of R are AB, AC, and BC, since each of those combinations is sufficient to uniquely identify each record. Let’s consider AB for instance. From AB → C we see that AB uniquely identifies C, and since B alone uniquely identifies D, AB together have covered CD, i.e. the entire set of attributes.

List all the functional dependencies that violate 3NF. If any, decompose R accordingly. After decomposition, check if the resulting relations are in 3NF, if not decompose further.

R的可能候选键是AB、AC和BC，因为这些组合中的每一个都足以唯一地标识每个记录。让我们以AB为例。从AB C中，我们可以看到AB唯一地标识了C，而因为B单独唯一地标识了D，所以AB一起覆盖了CD，即整个属性集。列出所有违反3NF的函数依赖项。如果有，则对R进行相应的分解。分解后，检查得到的关系是否在3NF中，如果没有进一步分解。

1. Consider the following relation StaffPropertyInspection:

StaffPropertyInspection (propertyNo, pAddress, iDate, iTime, comments, staffNo, sName)

The FDs stated below hold for this relation:

propertyNo, iDate → iTime, comments, staffNo, sName propertyNo → pAddress

staffNo → sName

From these FDs, it is safe to assume that propertyNo and iDate can serve as a primary key. Your task is to normalize this relation to 3NF. Remember in order to achieve 3NF, you first need to achieve 1NF and 2NF.

从这些fd中，可以安全地假设propertyNo和iDate可以作为主键。你的任务是将这种关系归一化到3NF中。记住，为了实现3NF，你首先需要实现1NF和2NF。

**END OF TUTORIAL**

# Optional Exercises

These are optional exercises for you to attempt if you there is time during the tute or as homework

1. The following Report table is used by a publishing house to keep track of the editing and design of books by a number of authors:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **report\_no** | **editor** | **dept\_no** | **dept\_name** | **dept\_addr** | **author\_id** | **auth\_name** | **auth\_addr** |
| 4216 | woolf | 15 | design | argus1 | 53 | mantel | cs-tor |
| 4216 | woolf | 15 | design | argus1 | 44 | bolton | mathrev |
| 4216 | woolf | 15 | design | argus1 | 71 | koenig | mathrev |
| 5789 | koenig | 27 | analysis | argus2 | 26 | fry | folkstone |
| 5789 | koenig | 27 | analysis | argus2 | 38 | umar | prise |
| 5789 | koenig | 27 | analysis | argus2 | 71 | koenig | mathrev |

By looking at the data, we see that functional dependencies in the Report table are the following:

report\_no → editor, dept\_no dept\_no → dept\_name, dept\_addr

author\_id → auth\_name, author\_addr

The candidate key for this relation is (report\_no, author\_id) since we need these two attributes to uniquely identify each record. Thus we have:

Report (report\_no, editor, dept\_no, dept\_name, dept\_addr, author\_id, auth\_name, auth\_addr)

### Is the Report table in 2NF? If not, put the table in 2NF.

1. Are there any insert, update or delete anomalies with these 2NF relations?

### Consider the following relation:

Class (courseNumber, roomNumber, instructorName, studentNumber, workshopNumber, grade, tutor)

The following functional dependencies hold for this relation:

workshopNumber → tutor

studentNumber, courseNumber → grade, workshopNumber courseNumber → roomNumber, instructorName

Normalize this relation into 3NF.