1814ict/2814ict/7003ict/1011ICT: Data Management/ Database Design/ Applied Computing

Topic 3.1: Normalisation

(Chapter 6)

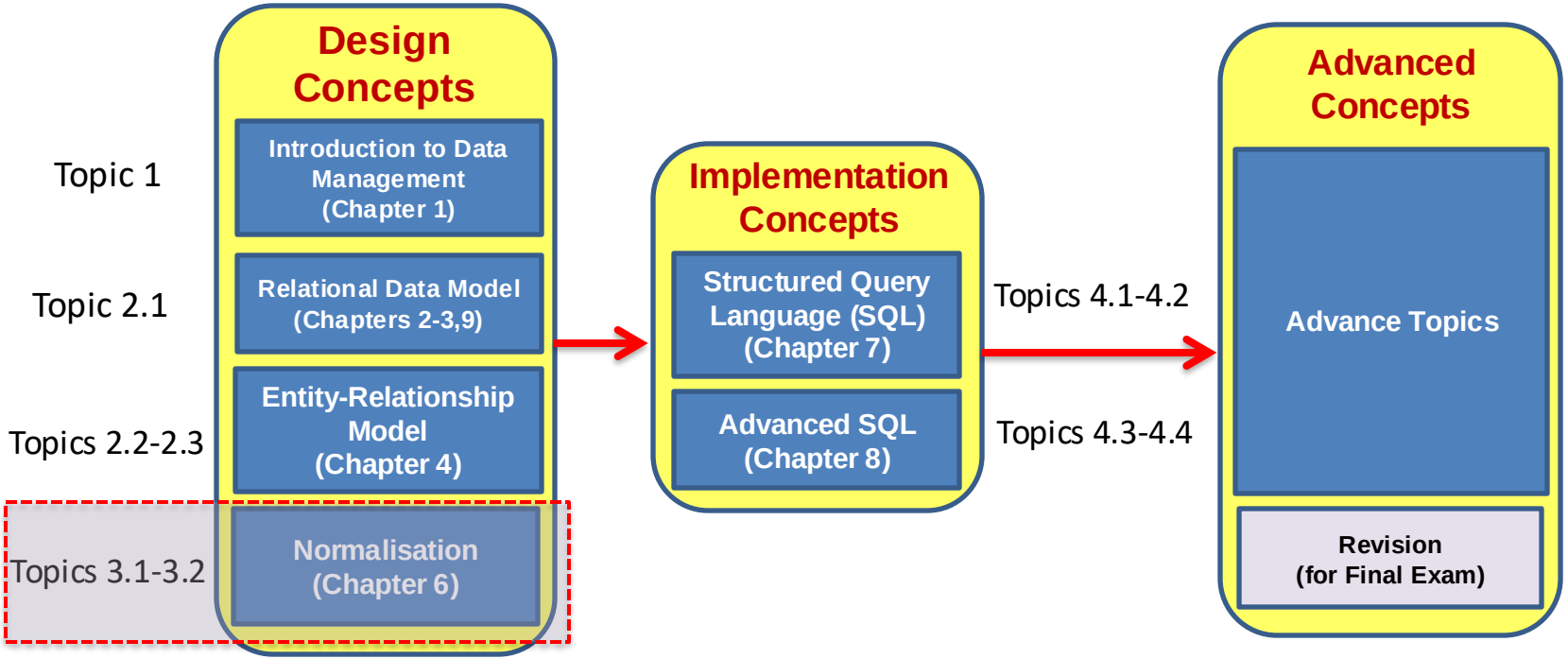
Convenor: AProf. Henry Nguyen - School of ICT

School of Information and Communication Technology

***Course developed by:** Dr Mohammad Awrangjeb; AProf John Wang and Dr Zhe Wang

Course bigger picture

- Chapter references are to textbook *Database Systems: Design, Implementation, & Management - By Carlos Coronel and Steven Morris*



Learning Outcomes

At the end of this lecture students will be able to know:

- Normalisation
- How to convert an un-normalised form (UNF) to 3rd normal form (3NF)

Content

- Importance of normalisation
- Pros & cons of normalisation

Outcome 1

- Revisit problems with the spreadsheet
- Functional dependency
- Steps to convert UNF to 3NF
- Examples

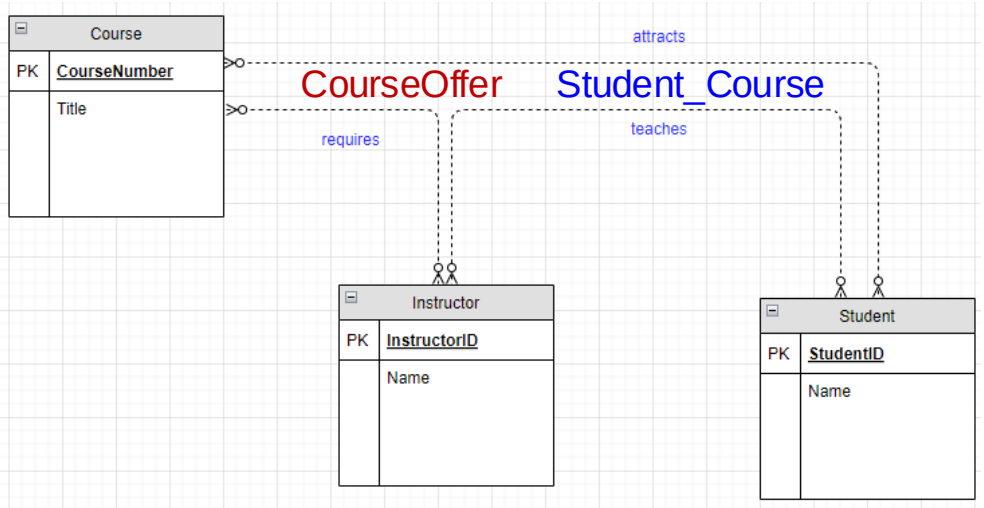
Outcome 2

Recap from Topic 2.3

Unary, Ternary or higher order relationship

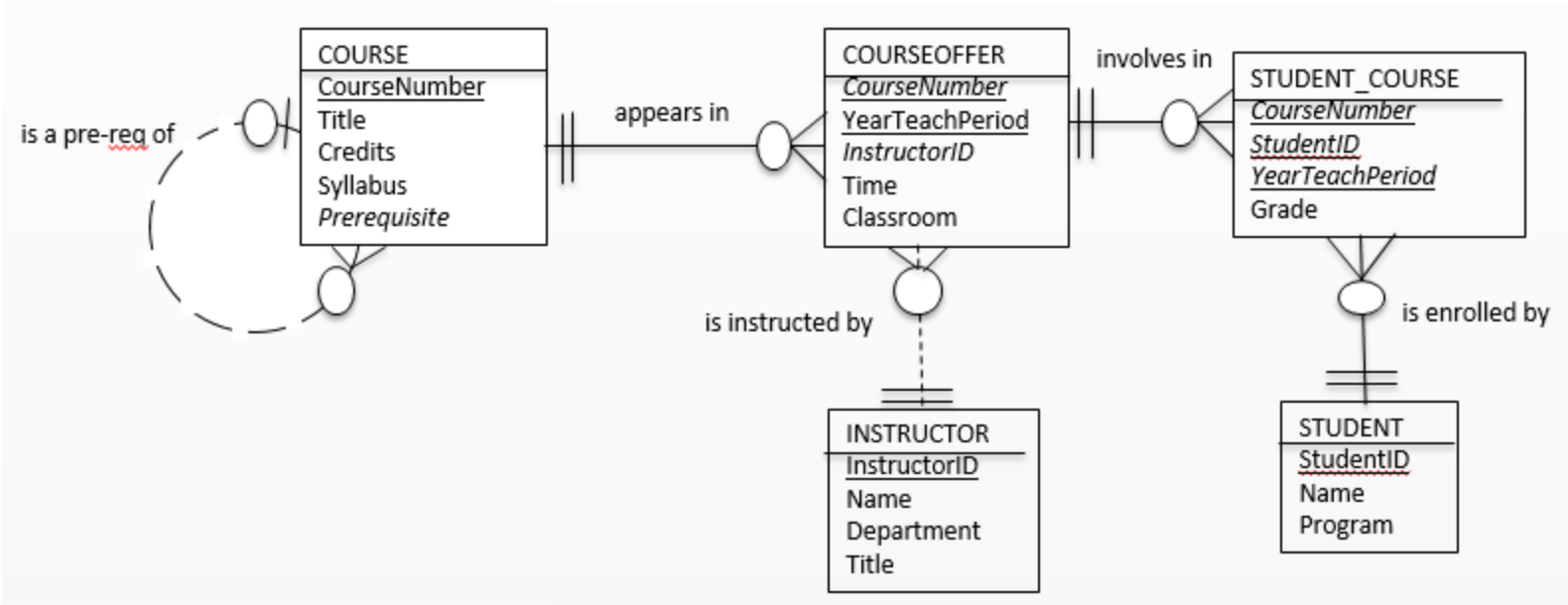
Example: Course – Instructor – Student

- **Unary** relationship happens when an entity has relationship with itself!
- **Ternary** or higher order relationship may happen because of M:N relationships among three or more entities.
- For example, consider the following business rules:
  - A course may be a pre-requisite of many other courses, but a course may have only one pre-requisite.
  - A course may attract many students and a student may enrol in many courses.
  - An instructor may teach many courses and a course may be taught by different instructors in different years and semesters.



- Now ask yourself
  - Where do you put year and semester of a course offer?
  - Where do you put student grade?

Ternary or higher order relationship



Find out:

- Strong and weak entities
- Associative entities
- Unary, binary and ternary relationships
- Strong and weak relationships

Revisit problems with Spreadsheet & Normalisation

Why Normalisation is required

- Find problems in the following table:

FIGURE 6.1 Tabular representation of the report format

Table name: RPT_FORMAT **Database name: Ch06_ConstructCo**

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CLASS	CHG_HOUR	HOURS
15	Evergreen	103	June E. Arbough	Elect. Engineer	84.50	23.8
		101	John G. News	Database Designer	105.00	19.4
		105	Alice K. Johnson *	Database Designer	105.00	35.7
		106	William Smithfield	Programmer	35.75	12.6
		102	David H. Senior	Systems Analyst	96.75	23.8
18	Amber Wave	114	Annelise Jones	Applications Designer	48.10	24.6
		118	James J. Frommer	General Support	18.36	45.3
		104	Anne K. Ramoras *	Systems Analyst	96.75	32.4
		112	Darlene M. Smithson	DSS Analyst	45.95	44.0
22	Rolling Tide	105	Alice K. Johnson	Database Designer	105.00	64.7
		104	Anne K. Ramoras	Systems Analyst	96.75	48.4
		113	Delbert K. Joenbrood *	Applications Designer	48.10	23.6
		111	Geoff B. Wabash	Clerical Support	26.87	22.0
		106	William Smithfield	Programmer	35.75	12.8
25	Starflight	107	Maria D. Alonzo	Programmer	35.75	24.6
		115	Travis B. Bawangi	Systems Analyst	96.75	45.8
		101	John G. News *	Database Designer	105.00	56.3
		114	Annelise Jones	Applications Designer	48.10	33.1
		108	Ralph B. Washington	Systems Analyst	96.75	23.6
		118	James J. Frommer	General Support	18.36	30.5
		112	Darlene M. Smithson	DSS Analyst	45.95	41.4

Why Normalisation is required

- **Problems:**
  - PROJ_NUM intended to be primary key, but it contains nulls!
  - JOB_CLASS invites entry errors e.g., Elec. Eng. vs Elect. Engineer
  - Redundant data
    - Charge per hour (e.g., $105/hour for Database designer)
    - Employee name (John G. News works in 2 projects, so repeated)
  - Redundancies cause anomalies
    - Insertion anomaly
    - Deletion anomaly
    - Update (or modification) anomaly

Normalisation

- NORMALISATION - SIMPLY 'COMMON SENSE'
- Converts a relation into relations of progressively smaller number of attributes and tuples until an optimum level of decomposition is reached - little or no data redundancy exists
- Normalisation is a Relational Database Implementation Model focused approach (it makes extensive use of FK's to connect relations)

Goals:

- Each table represents a single subject
- No data item will be unnecessarily stored in more than one table, i.e., No data redundancy
- All non-key attributes in a table are dependent on the primary key
- Each table is void of insertion, update, deletion anomalies
- Objective of normalisation is to ensure that all tables are in at least 3NF

Pros & Cons of Normalisation

Griffith UNIVERSITY

Advantages:

- Remove redundant data
- Prevent update/deletion/insertion anomalies
- Prevent data inconsistencies

Disadvantages:

- Retrieval of data may be penalised
- Need to retrieve data from a number of tables => reduce system speed
- May need to decide how far to normalise when performance is an issue
  - Example:
    - City and State together determine Postcode

Thank you