BIOM9450 (Biomedical Informatics) – Week 1

Introduction to biomedical informatics and database design

The Graduate School of Biomedical Engineering, UNSW Sydney

2021 - T3

Course overview

Lecture outline

Course overview

Diagnostic test

data science

Health data science

Biomedical informatics

- Sub-domain of biomedical informatics
- Importance of biomedical informatics

Australian health system challenges

Database

- Electronic medical record
- Database management system (DBMS)
- Data type
- File access
- Relational Database

Normalization

- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)

Week1 - Part 1

T3 Course Representatives Program

https://docs.google.com/forms/d/e/1FAIpQLSeKl1cypndvCCHNtYasQlYuAhTp-iINkq1a6Kt3hWWrAGlRbA/viewform

https://docs.google.com/presentation/d/13Z1qrvM5KwoP9FNy4O9lOgwuHmD4XbSdUy4_J2r01mA/edit#slide=id.gb6cf059226_0_3

BioMedical Informatics

Course cordinator & lecturer: Dr Hamid Rokny

Room 1002, Biological Science building (E26)

If you have any concern, just come and knock the door ☺

Microsoft TEAMS channel:
https://teams.microsoft.com/l/channel/19%3aIwXHSoF4Gmef9BOKwSf5eWL1CQNKWZ1sk8rlRRgrBLA1%40thread.tacv2/General?groupId=714f4b9c-65b4-4d59-80b0-c09248d6fb73&tenantId=3ff6cfa4-e715-48db-b8e1-0867b9f9fba3

Course overview

Lecture (most weeks)	**Tue 05:00PM - 06:30PM**	**Online**
Tutorial/Laboratory	Thu 03:00PM - 04:30PM	Online
Course Coordinator	Hamid Alinejad Rokny Room 1002, Level 1, E26 Building Phone 9385 1725; E-mail: h.alinejad@unsw.edu.au	
Lecturer **Guest Lecturers:**	Hamid Alinejad Rokny (HAR) Amin Beheshti (AB: Macquarie University and Director of AIP Research Centre), Masoud Ehsani (ME: Lead Cloud Architect at Finstro), Elizabeth Antoine (EA: Senior Data Scientist at Microsoft)	
Demonstrators:	Heba Khamis (HK), h.khamis@unsw.edu.au Daniel Al Mouiee (DAM), d.almouiee@unsw.edu.au	
Student representative	**Sheril Jose Sao** (SJS), s.josesao@student.unsw.edu.au	

Course overview

How can you contact us:

If Microsoft TEAMS sub_channel works then
Exit

Elseif Microsoft TEAMS direct message to HB or DAM or HAR works then
Exit

Elseif (email to h.khamis@unsw.edu.au (HK) or d.almouiee@unsw.edu.au (DAM) or h.alinejad@unsw.edu.au (HAR)) works then
Exit

Elseif F-2-F meeting (come to my office (Room 1002, Biological Science Building (E26)) (does not work in COVID-19 time)) works then
Exit

Else start the process again ☺

End if

Course overview

How we want to evaluate your learning?

Assessment	Contribution
Tutorials/Hand in Questions	40%
Exam	20%
Final project	40%
Bonus	Max 5%

How we want to evaluate your learning?

Assessment Tutorials/Hand in Questions	Contribution
database design	10%
structured query language	10%
Javascript and HTML	10%
dynamic web page development (PHP)	10%

- A major aspect of this course is practical computing skills.
- You will complete practicals and hand in assessments in the areas of **database design**, **structured query language**, **Javascript and HTML** and **dynamic web page development (PHP)**.
- Assignments should be submitted on time. **Marks may be deducted for late submission** without prior approval.

How we want to evaluate your learning?

Assessment - Exam	Contribution
written exam	10%
computer-based practical exam	10%

- An exam is scheduled at the end of semester. It comprises part written exam (10%) and part computer-based practical exam (10%).
- This assessment is a **direct test of the degree** to which the knowledge-based learning outcomes listed in the course outline have been achieved.
- If you have successfully completed the practical component of the course, then you should have no difficulty with the computer-based exam.
- Similarly, the written component of the exam should present no problems to people who have attended and participated in the lectures.

How we want to evaluate your learning?

Assessment – Major project	Contribution
Major project	30%
reports	10%

- A major assessment component of this course is a computer-based project that integrates information form the lectures and practicals to produce a workable system that encompasses one aspect of a medical information system.
- This assessment is a direct test of the degree to which the knowledge-based learning outcomes listed above can be practically applied.

- You will be divided into group of 3 students.
  - groups will be created based on a background assessment quiz.
- The project will be released in week 6, but we will give you a summary of what you have to do on week 3.
- Each group needs to prepare a written document.
  - Each member's contribution should be very clear in the project.

Lab/Tutor

You need to do your Lab practice in a group of 3. Basically, each group should work together during the tutorial.

To do this:
On Tuesday, 3:00pm – 4:30pm, You need to get online on TEAMS.
Do a group call between all group members (I would suggest to record the call to be used by all group members, later).
Start working on the tutorial.

The demonstrators will join you, regularly.
Or
You can ask them to join you if you have any questions.

Note: If you are not able to attend the tutorials, you can catch up with these tutorials in your own time.

Diagnostic test

Microsoft form:
https://forms.office.com/r/fYR3hsgfVa

Students will be assigned into groups based on their diagnostic assessment test result. For example, Jimmy (who got a high-level score in the diagnostic assessment test), Hannah (who got a mid-level score), and Karen (who got a low-level score) will be in a same group.

Diagnostic test

From where do you cometh?

Use this link: https://app.sli.do/event/jrfub4lm/live/polls

Or

use this Slido code: #396231

What is data science?

Data science is the application of computational and statistical methods to extract knowledge from structured and unstructured data in the real world.

What is data science not?

- Data science is not machine learning
- Data science is not statistics
- Data science is not big data

Data science is one of the best jobs

What is data science?

Data science = Statistics +
Data processing +
Machine learning +
Scientific question +
Pattern recognition +
Visualization +
Big/complex data + …

Image Source: http://www.kiwidatascience.com

What is data science not?
Data science is not Machine Learning (ML)
Machine learning contains computational and statistical problems, without any concerning about answering *scientific questions.*

The main focus of ML is method development, but data science is caring about the **entire data processing methodology**, which may or may not involves algorithm development.

... but sometimes the best way to solve a problem is just by visualizing the data.

What is data science not?
Data science is not statistics

Data scientists have usually engineering backgrounds, but statisticians have usually math background.

Statisticians use R or MATLAB, while data scientists use Python or MATLAB.

Statistics	Data Science
Estimating	Learning
Data point/observation	Example/instance
Classification	Supervised learning
Covariate/predictor/independent variable	Feature
Response/output/dependent variable	Labelling
Dummy variable/indicator coding	One-hot coding

What is data science not?
Data science is not big data!

Big data analysis involves data mining techniques on large number of databases.

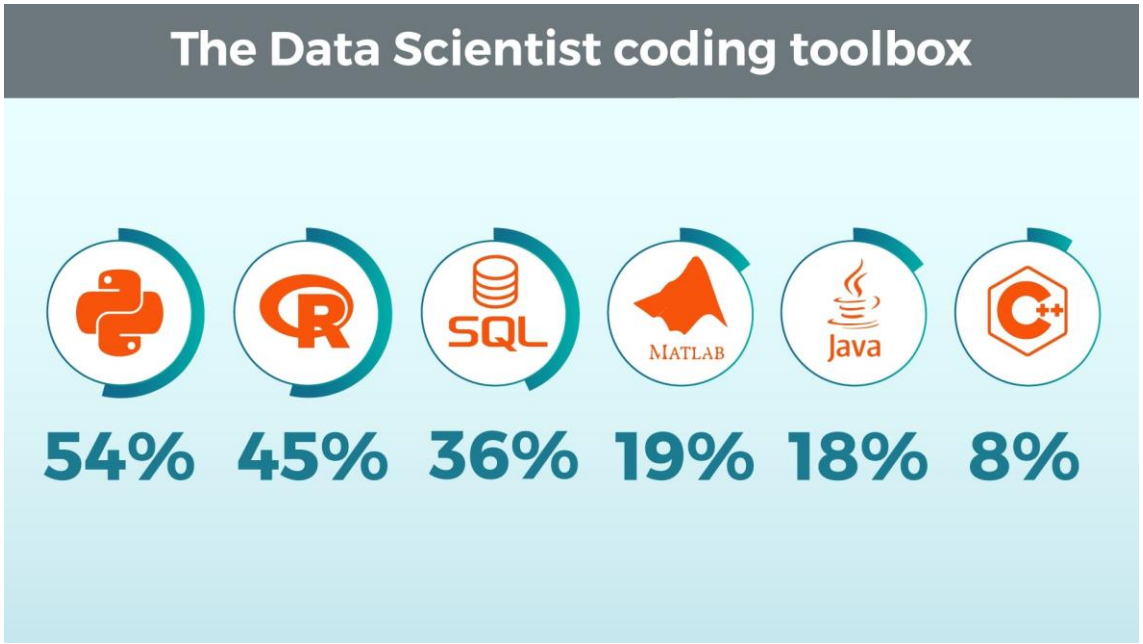
Data science uses ML or statistical algorithms to attract knowledge from databases, which may or may not be in a large scale as big data.

Don't create more work for yourself than you need to.

Image from: https://www.besanttechnologies.com/big-data-vs-data-science

How to become a data scientist?

Data scientists need to know how to “CODE”.



https://365datascience.com

Data science pipeline

Learning data science with Python - Libraries

Data science is one of the best jobs

Data science is the best job in America

And

One of the best jobs in Australia

https://www.pwc.com/us/en/library/data-science-and-analytics.html

In Australia, Data Scientists are a Hot Commodity

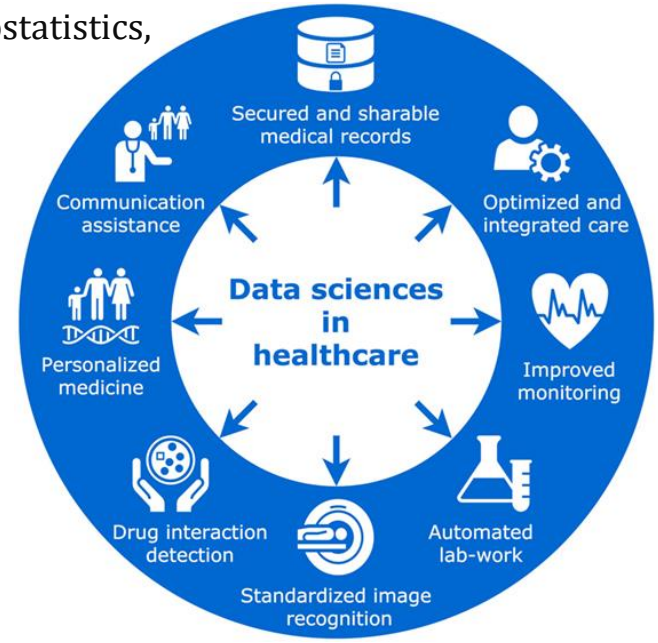
https://www.hiringlab.org/au/blog/2019/04/30/data-scientists-au/

Data Scientist has been called "the most promising career" by LinkedIn.

Health data science

Health Data Science (HDS) is an important emerging discipline, integrating mathematics, **statistics**, informatics/machine learning with biological and medical data to derive knowledge from health data and address important questions in health and biomedical sciences.

HDS is an INTER-DISCIPLINARY area, arising at the intersection of biostatistics, bioinformatics, computer science and health.



https://doi.org/10.3389/fpubh.2018.00099

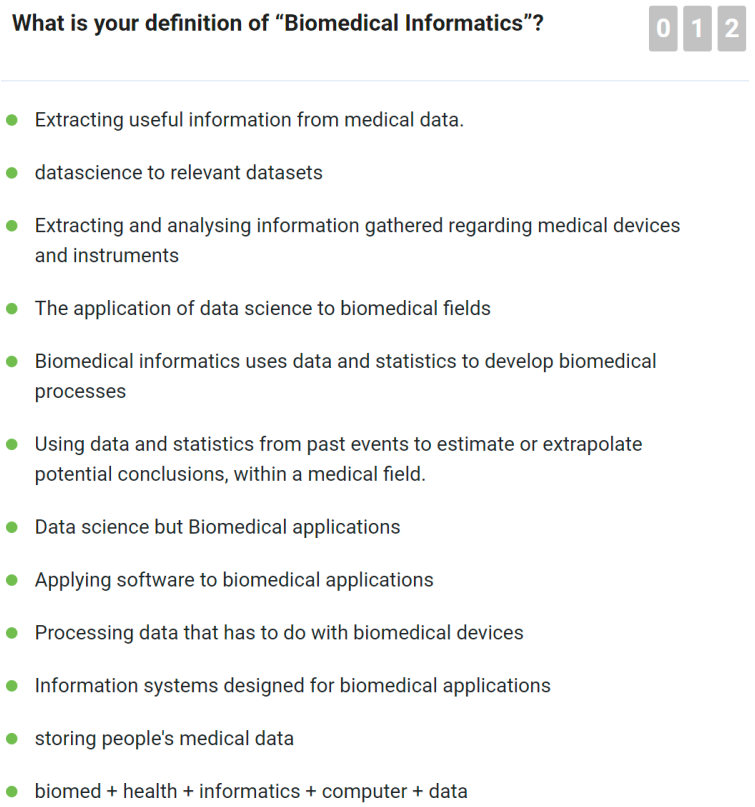
Biomedical informatics

Biomedical informatics is an Interdisciplinary field. What is your definition of “Biomedical Informatics”?

Use this link: https://app.sli.do/event/jrfub4lm/live/polls

Or

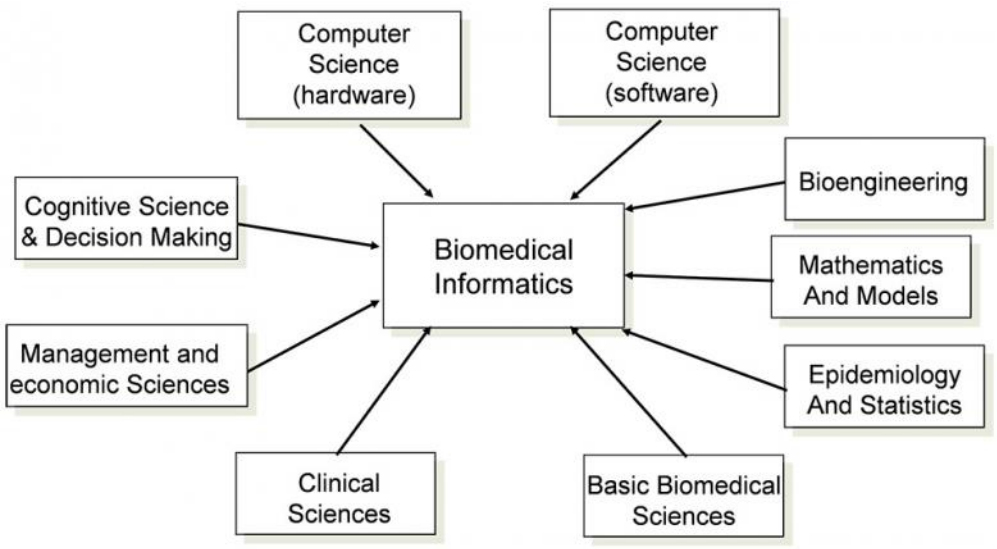
use this Slido code: #396231



Biomedical informatics

BioMedical Informatics (BMI) is a multi-disciplinary field that integrates knowledge of information/computer sciences and medical/biological sciences to optimize use and application of biomedical and health data across the spectrum, from molecules and cellular processes to individuals and populations.

Biomedical Informatics is now a key field to better understanding biology and medicine.



Origins of Biomedical informatics

–1960s and 1970s
–Laboratory information system (LIS)
–Hospital information system (HIS)
-Clinical decision support systems

–1970s and 1980s
-Information retrieval and delivery

–1980s and 1990s
–Internet-based information systems
–Clinical practice guidelines
-Personal health and precision medicine
–Apps and social networking
–Big data analytics (machine and deep learning)

Why Biomedical informatics

What is lacking in the paper-based approach?

Medication errors:

- Patient has allergic reactions
- Wrong medication given to patient
- Wrong dosage given to patient
- patient communication

The National Academies of Sciences, Engineering, and Medicine:

"Medication Errors Injure 1.5 Million People and Cost Billions of Dollars Annually; Report Offers Comprehensive Strategies for Reducing Drug-Related Mistakes"

Medication errors may be prevented by a simple scanning device

Why Biomedical informatics

What is lacking in the current paper-based approach?

Cost:

- Paper and storage
- Environmental Cost
- Storages are not scalable
- Paper-based medical records require physical space
- No backups and weak security.
- Paper-based managing systems need more personnel to manage and paper files

Read this paper:
https://www.ncbi.nlm.nih.gov/pmc/articles/PMC212784

Time:

- Time consuming
- Find record, perform tests

Why is it important to backup medical/health data?

The patient data is crucial for providing a better care and follow up treatments for patients.

Read this interesting guideline:
Do you have a recent backup of your important files?
https://www.digitalhealth.gov.au/sites/default/files/2020-11/Backups-Prepare_for_an_emergency.pdf

NHS lost 1.8 million patient records in a year

More than 5,000 confidential patient records are being lost by the NHS every day, according to new figures.

Electronic Medical Record

An electronic medical record (EMR) contains **data about a patient's personal information and health history**. For example, an EMR system can contain information about diagnoses, surgery, medicines, tests, allergies, immunizations, and treatment plans. An EMR system can be used by healthcare organization for documentation, monitoring and management.

Read this interesting documents:

https://www2.health.vic.gov.au/Api/downloadmedi4151B141DC98%7D

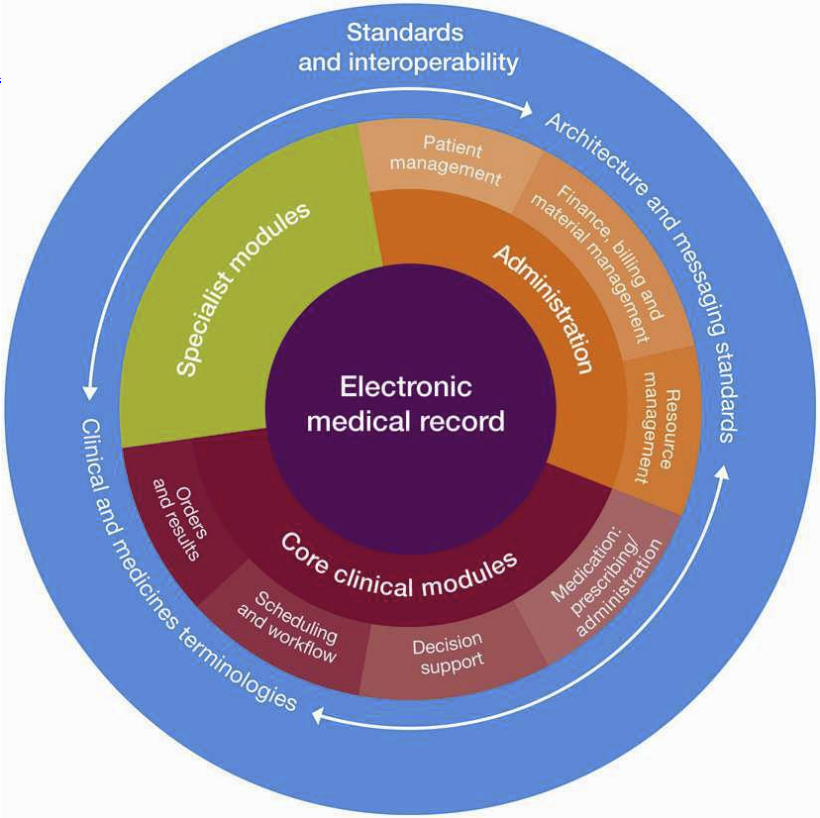


Figure credit: Australian department of health

The Traditional Electronic Medical Record (TEMR)

- Single user (i.e. little integration)
- Very poor communication between departments
- Information easily lost (single original)

But, the main problem with TEMR is **the lack of cooperation between different parts of the systems**, in which they are not able to communicate effectively with each other.

Sub-domains of Biomedical Informatics

- Clinical informatics
- Medical informatics
- Nursing informatics
- Public health informatics
- Bioinformatics
- Imaging informatics
- Pharmacy informatics
- Dental informatics
- Veterinary informatics
- Consumer health informatics
- eHealth
- Clinical research informatics
- Translational research informatics
- etc.

Sub-domain of biomedical informatics

Bioinformatics

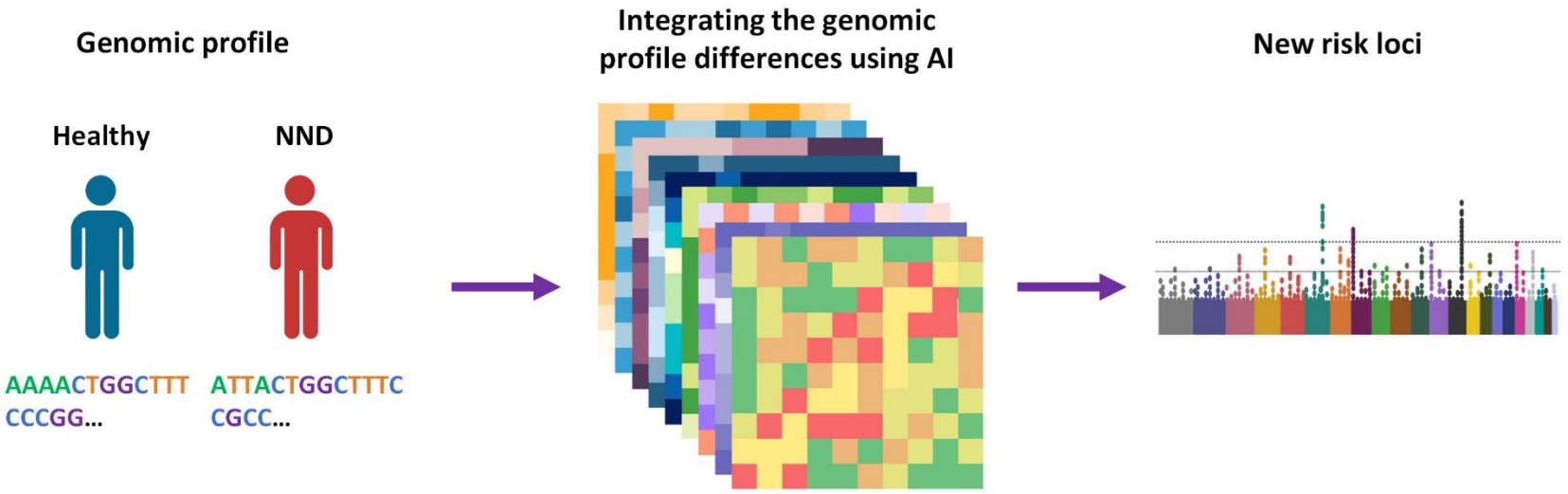
Bioinformatics is a subdiscipline of BMI, concerning the acquisition, storage, analysis, and dissemination of biological and medical data, starting at the molecular genetics and genomics level to medical imaging data.

Clinical informatics deals with the management, organization, and presentation of information related to the delivery of healthcare.

Bioinformatics focuses on the management and interpretation of information related to the underlying basic biological and medical data including gene and protein functions, evolutionary relationships, and predicting of biomarkers.

Bioinformatics

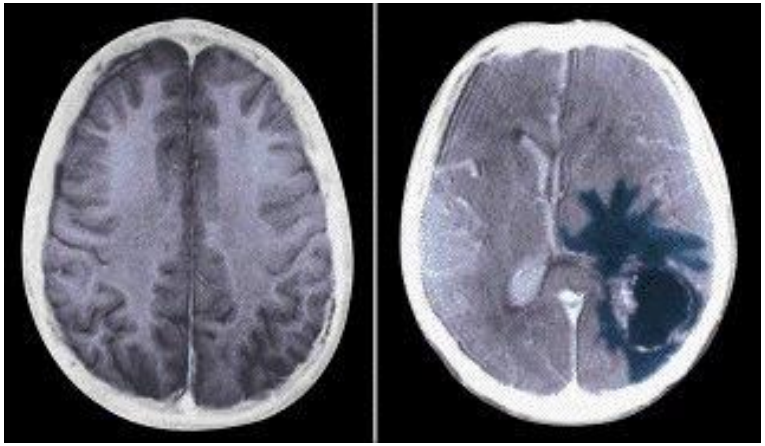
For example, using Artificial Intelligence in predicting disease-associated genes.



Biomedical imaging informatics

Biomedical Imaging Informatics (BII), considering to the clinical aspect is the **use of computer science and imaging-derived information techniques** to 1) improve patient outcomes through **better managing medical imaging** data within a healthcare/medical network; 2) **deliver efficient, accurate, and reliable medical imaging services** within a healthcare/medical network; 3) **diagnose imaging abnormalities** in medical and healthcare images (e.g., tumors).

BII has some overlap with medical physics, data mining, machine learning, computer science, and health data science, and genomics.



Healthy brain(left) vs brain tumor (in blue, right).

Biomedical imaging informatics

The History of Medical Imaging, PACS and Standards

	1970's	1980's	1990's	2000's	2000 - 2020
	US, CT, X-ray	US, CT, X-ray, MR	US, CT, X-ray, MR, PET, Nuclear Med	US, CT, X-ray, MR, PET, Nuclear Med	US, CT, X-ray, MR, PET, Nuclear Med - plus other new imaging modalities, Genomics and Proteomics
Standards	None	ARC-NEMA 1 and 2	DICOM → HL7 →		DICOM and HL7 merge into a new XML based standard Continua Alliance Various regulatory standards Privacy and Security e.g. Health Insurance Portability and Accountability Act (HIPAA) HIPAA →
Generations	Single machines	Multiple scanners in a department, each with one viewing station (no networking)	Unix based PACS Windows based PACS	Web-based PACS	Full web-based PACS (all data types), plus a central data repository

Picture from Archiving and Communications System (PACS)

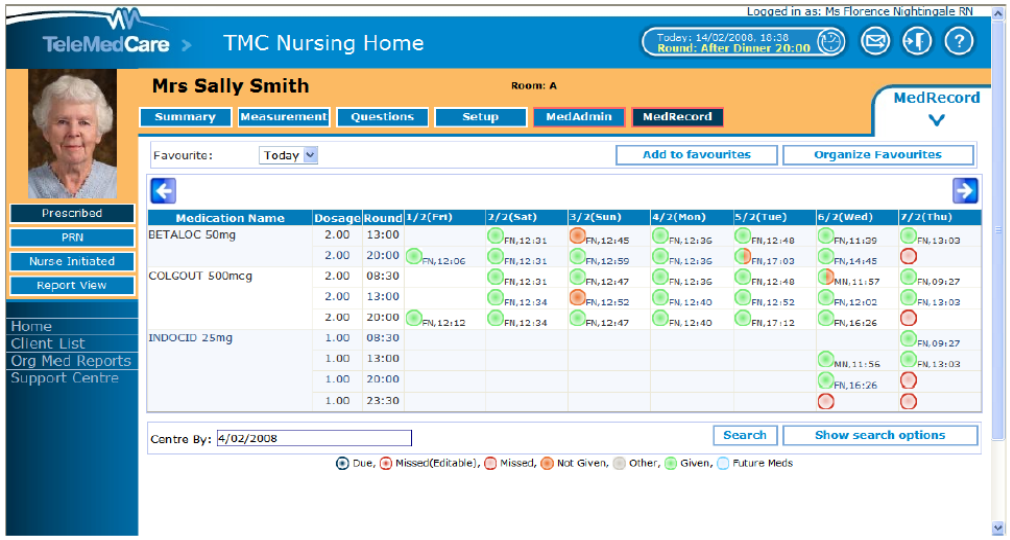
Clinical Informatics

Clinical Informatics (CI) is an **interdisciplinary filed of computer science, information technology**, database, and management science, concerning about how healthcare data is acquired, structured, stored, processed, analyzed, visualized, and communicated. The main task of a CI is to **discover knowledge form healthcare data** to be used in clinical decision-making systems.

As a CI scientist, you are not concerning about information technology (IT) or programming, but **your main focus is healthcare data** and **how can you manage and interpret** such data.

Clinical informatics can be used in different healthcare and medical settings such as hospital, clinic, physician's practice, and others.

For example, a Web-Based Medication and Diet Regime Management System



Sub-domain of biomedical informatics

Pharmacy Informatics is a sub-domain of medical informatics that integrates computer science and information technology into the **pharmaceutical practice** and medication-related data and knowledge to ensure optimal patient safety, compliance, and health outcomes.

A pharmacy informaticist is involved the delivery of **optimal medication selection** for patient care. Use and administration, acquisition, storage, and analysis of pharmaceutical related data.

Dental Informatics

Dental Informatics is the application of computer science and information technology to improve dental practice, research, and program administration.

(Eisner 1992)

Main applications of dental informatics field is dental education, dental record keeping, dental imaging, digital radiography, teledentistry, and research.

Veterinary Informatics

"Veterinary Informatics is the discipline concerned with the applications of information science, engineering, and computer technology to support veterinary teaching, research, and practice"

(Association for Veterinary Informatics)

eHealth is also written "e-health"
"e-Health is defined as the use of emerging interactive technologies such as internet, mobile device, digital technologies and telecommunications to **facilitate** health improvement and health care services".

For example
electronic health records (EHR)
mobile health (mHealth) apps
wearable devices
telehealth and telemedicine

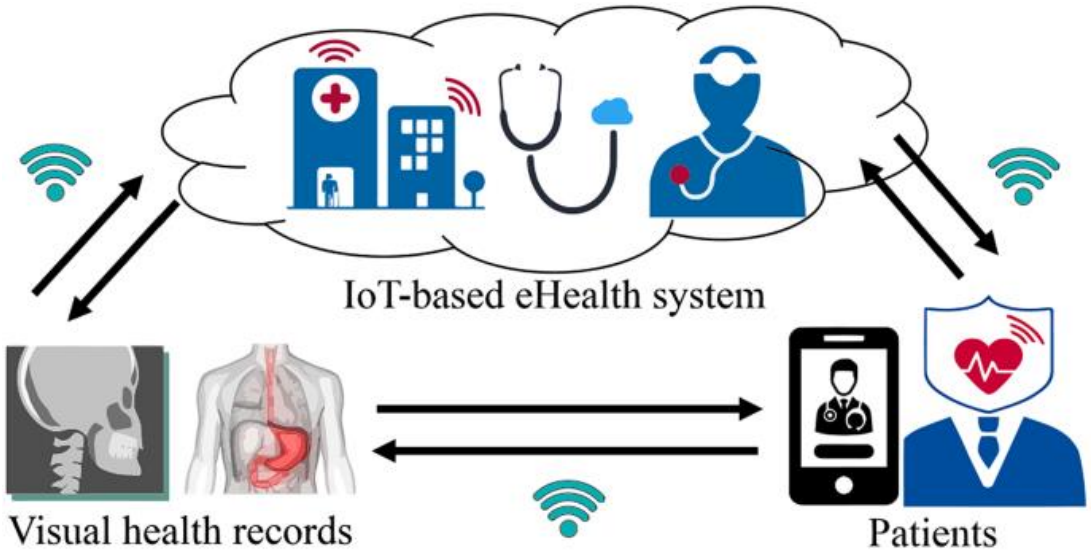


Figure credit: Yung-Yao Chen, Complex & Intelligent, Systems (2021)

mHealth

The use of mobile and wireless technologies to provide health-related services to improve health outcomes and patients care.

For example

1. 2. Maternal Care and Childcare:
2. Ovia Health: to help parents raise their families alongside careers, and pay less healthcare cost.

3. Diagnostic Tool: Sony's mSafety Mobile Health: An app with customizable smartwatch and streamlined cloud connectivity is a great example of mHealth diagnostic tool.

4. Primary Care from Home
6. Chronic Disease Management
7. Medical Data Management

Sub-domain of biomedical informatics

Telemedicine

Telemedicine refers the **use of electronic signals** such as internet, telephones, PCs, mobile, satellites, or videoconferencing **to transfer health data from one site to another** in order to improve access to health care. For example, A physician can use a telecommunications technique to deliver care to a patient who is in a different place.

Telesurgery: Performing surgery on a patient when the doctor is in different location.

Teleradiology: sending the radiological patient images to another location to be checked by a doctor.

Telecare: Using telecommunication technologies to provide remote assistance in therapy to patients.

Sub-domain of biomedical informatics

Public health informatics (PHI)
PHI is using information technology and computer science, data science to interpret and analyse **public health data** to advance strategies such as **surveillance, prevention, outbreak management**, electronic laboratory reporting, and predictive modeling.

PHI has been used to improve the safety and quality of patient care in the form of **patient portals, EMR, telehealth, healthcare applications, and data visualisation tools.**

Clinical decision support systems (CDSS)

CDSS are computer-based tools or a health information technology that **analyze health data within electronic health records** (EHRs) to provide clinicians, doctors, staffs, and patients knowledge and patient-specific information to promote healthcare outcomes.

- Task:
  - Diagnosis/interpretation
  - Therapy/management
- Scope:
  - Implementing a system for diagnosis of acute abdominal pain; infectious diseases Dx;
  - Implementing clinical guidelines at the point of care

Types of CDSS

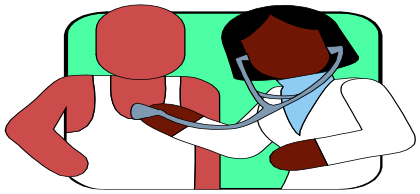
Control level:

Human-initiated consultation (e.g., MYCIN, QMR)
Data-driven reminder (e.g., MLMs)
Closed loop systems (e.g., ICU ventilator control)

Interaction style:

Prescriptive (e.g., ONCOCIN)
Critiquing (e.g., VT Attending)

Conventional data collection for clinical trial



Medical records

Data sheets

Clinical trial design

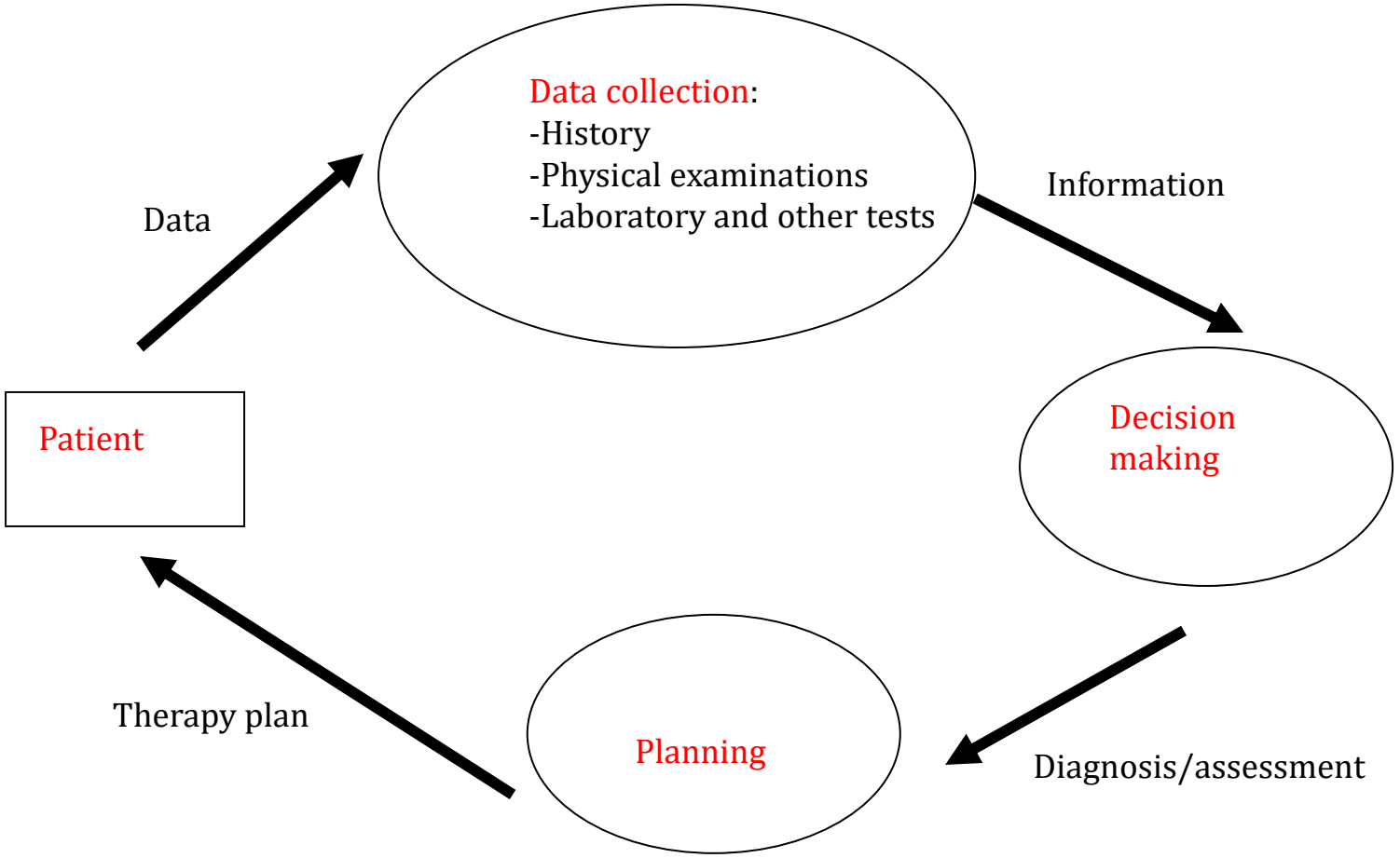
- Definition of data elements
- Definition of eligibility
- Process descriptions
- Stopping criteria
- Other details of the trial

Computer database

Analyses

Results

The Diagnostic-therapeutic Cycle

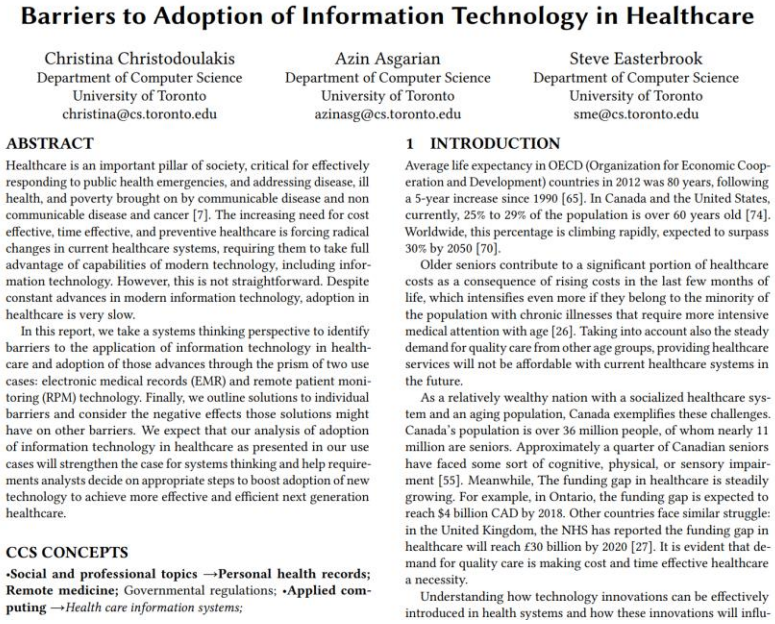


Importance of biomedical informatics

Past (and some present) reasons for lack of adoption of informatics in clinical care

- Involves complex organisms (unlike physical processes)
- Need sophisticated abstraction + detail
- Technology for gathering complex information was not available
- Reimbursement has not been linked to clinical information (many admin systems but few clinical)
- Inertia; fear; ignorance; cost
- Security; integrity; privacy; previous failures
- Lack of standards for messaging, coding classification, electronic medical records (EMR)
- Rapid turnover of technology
- data reliability and lack of data integration

Read this paper for more information about Barriers to Adoption of Information Technology in Healthcare: https://www.cs.toronto.edu/~christina/documents/ACM_CASCON2017.pdf



Barriers to Adoption of Information Technology in Healthcare

Christina Christodoulakis
Department of Computer Science
University of Toronto
christina@cs.toronto.edu

Azin Asgarian
Department of Computer Science
University of Toronto
azinasg@cs.toronto.edu

Steve Easterbrook
Department of Computer Science
University of Toronto
sme@cs.toronto.edu

ABSTRACT

Healthcare is an important pillar of society, critical for effectively responding to public health emergencies, and addressing disease, ill health, and poverty brought on by communicable disease and non communicable disease and cancer [7]. The increasing need for cost effective, time effective, and preventive healthcare is forcing radical changes in current healthcare systems, requiring them to take full advantage of capabilities of modern technology, including information technology. However, this is not straightforward. Despite constant advances in modern information technology, adoption in healthcare is very slow.

In this report, we take a systems thinking perspective to identify barriers to the application of information technology in healthcare and adoption of those advances through the prism of two use cases: electronic medical records (EMR) and remote patient monitoring (RPM) technology. Finally, we outline solutions to individual barriers and consider the negative effects those solutions might have on other barriers. We expect that our analysis of adoption of information technology in healthcare as presented in our use cases will strengthen the case for systems thinking and help requirements analysts decide on appropriate steps to boost adoption of new technology to achieve more effective and efficient next generation healthcare.

CCS CONCEPTS

•**Social and professional topics** →**Personal health records; Remote medicine;** Governmental regulations; •**Applied computing** →*Health care information systems;*

1 INTRODUCTION

Average life expectancy in OECD (Organization for Economic Cooperation and Development) countries in 2012 was 80 years, following a 5-year increase since 1990 [65]. In Canada and the United States, currently, 25% to 29% of the population is over 60 years old [74]. Worldwide, this percentage is climbing rapidly, expected to surpass 30% by 2050 [70].

Older seniors contribute to a significant portion of healthcare costs as a consequence of rising costs in the last few months of life, which intensifies even more if they belong to the minority of the population with chronic illnesses that require more intensive medical attention with age [26]. Taking into account also the steady demand for quality care from other age groups, providing healthcare services will not be affordable with current healthcare systems in the future.

As a relatively wealthy nation with a socialized healthcare system and an aging population, Canada exemplifies these challenges. Canada's population is over 36 million people, of whom nearly 11 million are seniors. Approximately a quarter of Canadian seniors have faced some sort of cognitive, physical, or sensory impairment [55]. Meanwhile, The funding gap in healthcare is steadily growing. For example, in Ontario, the funding gap is expected to reach $4 billion CAD by 2018. Other countries face similar struggle: in the United Kingdom, the NHS has reported the funding gap in healthcare will reach £30 billion by 2020 [27]. It is evident that demand for quality care is making cost and time effective healthcare a necessity.

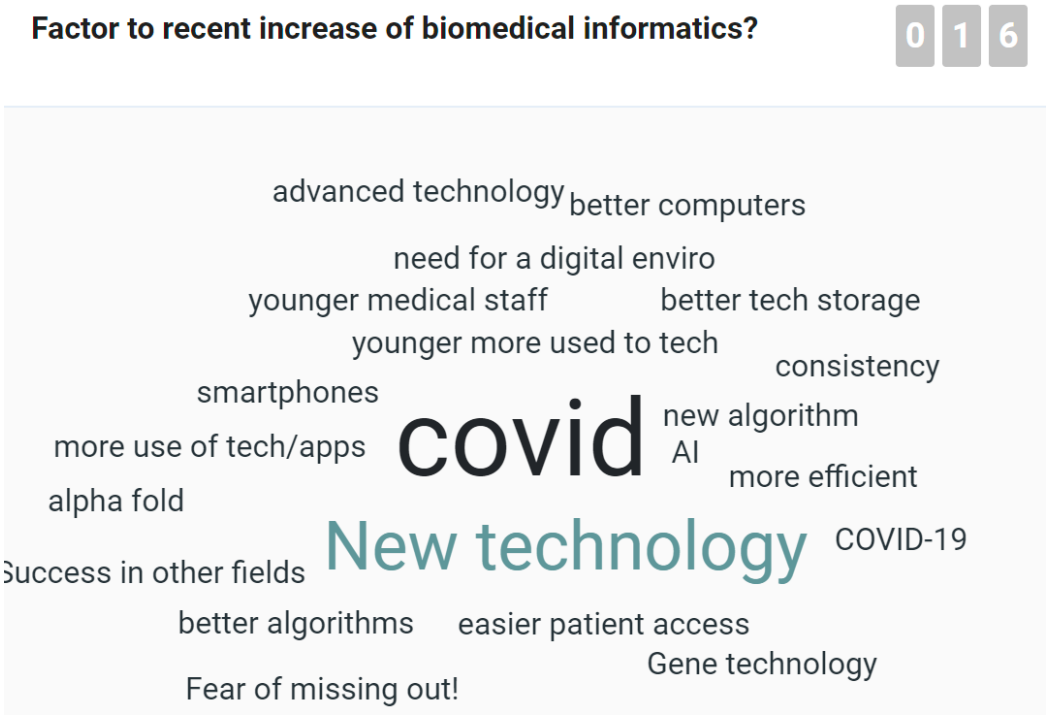
Understanding how technology innovations can be effectively introduced in health systems and how these innovations will influ-

Factor to recent increase of biomedical informatics

Use this link: https://app.sli.do/event/jrfub4lm/live/polls

Or

use this Slido code: #396231



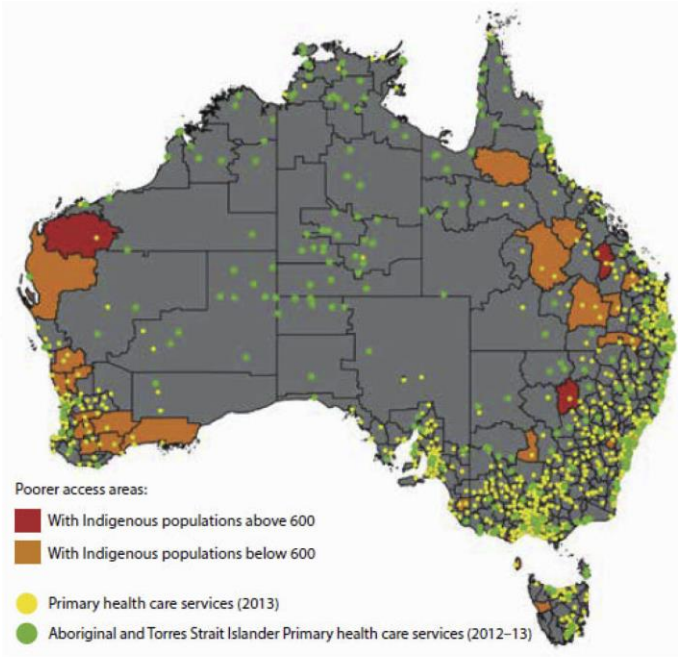
Factor to recent increase of biomedical informatics

- Increase in use of technology - more data generated
- Mobility of population - need to communicate
- Specialisation - need to communicate
- Managed care systems - need to communicate
- Rise in health costs - attempt to control care
- Improved hardware - faster and more memory
- Improved methods - acquisition, transfer, retrieval
- Reduced computer costs
- Pervasive use of computers, networks and smartphones
- Interoperability standards
- Increased awareness (Internet)

Australia's health system

UNSW SYDNEY | Australia's Global University

Australian Healthcare System



Australia has one of the best health system in the world. It provides quality, safe, and affordable health care everyone in the country.

Importance of biomedical informatics

Australian health system challenges include:

- an ageing population and increasing demand on health services
- increasing rates of chronic disease
- costs of medical research and innovations
- making the best use of emerging health technologies
- making better use of health data

Australian health system challenges and the impact of new technologies

Chronic diseases

Managing chronic conditions is another challenge. The rise in many chronic conditions also increases demand for flexible, person-centred treatment models.

To address this, governments are taking a national approach to coordinated care in their health plans and policies, including GP-led team-based care for patients with chronic and complex conditions. We continue to work on ways to improve sharing of health and medical information between providers.

Health and medical research

Advances in medical science are set to completely change health care. For example, genomic testing will help doctors diagnose health conditions and diseases earlier, as well as provide better prevention and treatment options for people. But these advances are very costly and come with some difficult ethical and legal issues that need to be worked through.

New technology

New technologies also have an impact on health and medical services — from digital health technologies to automated health and diagnostic services. These technologies help to improve the health system, but they can affect patients and the health workforce.

To meet some of these challenges, the Australian Government is investing in medical research and technological innovation through the Medical Research Future Fund. This will see more innovations developed, tested and made available for Australians in all areas of health care.

Better use of health data

Comprehensive data can help us to improve health policy, programs and services. That's why linking different health information across the health system is an important part of our work.

Text from Australian Government Department of Health

Importance of biomedical informatics

What is the Health Sector?

- $155 billion p.a. industry (9.4% of GDP) (UK: 9.1%, USA: 17.1%)
- ($59 billion hospitals, $55 billion primary health)
- 44,000 doctors & 180,000 nurses
- 753 public hospitals (costing $50 billion) & 593 private hospitals, 12,500 pharmacists
- On any day 342k people visit a GP, 742k medicines dispensed, 23k people admitted to hospital, and 17k emergency department presentations
- 3 Funding sectors
- Commonwealth (HIC / Medicare / GP Divisions)
- States (Public Hospitals)
- Non-government
- 50,000 other health workers
- More than 20% of population over the age of 65
- Australians enjoy one of the highest life expectancies in the world—79.5 years for males and 84.0 years for females
- Aboriginal and Torres Strait Islanders life expectancy is about 10 years shorter

The Health Care and Social Assistance industry is of the largest industry in Australia, in which in 2020 it provided over 1.6M new job opportunities in Australia. It is predicted to increase to more than 2M by 2024.

Week1 - Part 2

Let's have some break ☺

Database management systems, database design and normalisation

This section covers:

- Uses and advantages of an electronic medical record (EMR)
- Databases and database management systems (DBMS)
- Relational database topology
- Indexes, keys and referential integrity
- Database design
- Database normalisation

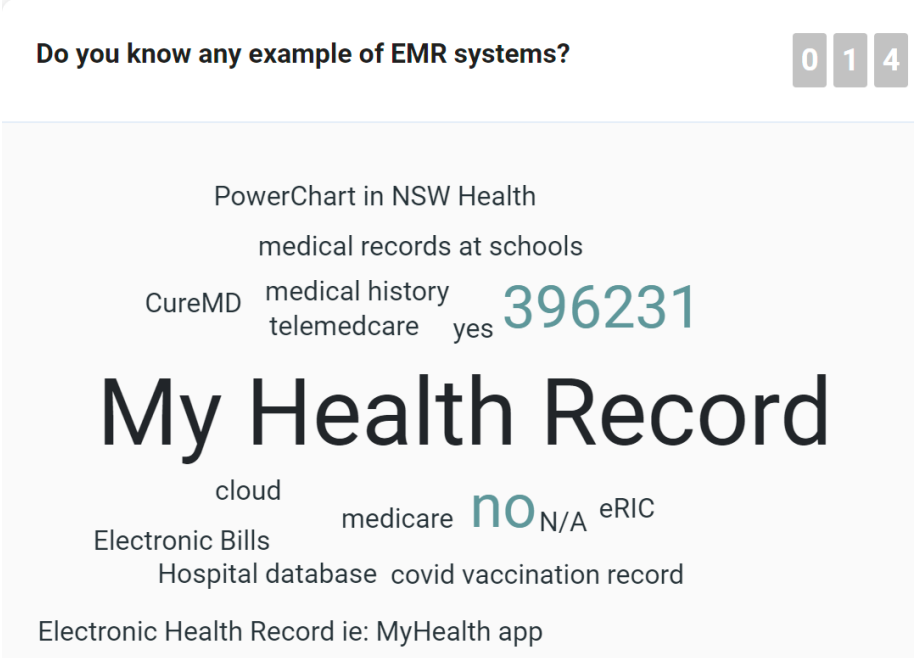
Electronic medical record

An electronic medical record (EMR) is a systematic collection of patients information and patient health record such as diagnoses, patient's history, medicines, tests, and treatment.

Use this link: https://app.sli.do/event/jrfub4lm/live/polls

Or

use this Slido code: #396231



Electronic medical record

An electronic medical record (EMR) is a systematic collection of patients information and patient health record such as diagnoses, patient's history, medicines, tests, and treatment.

Uses of the Health/Medical Record

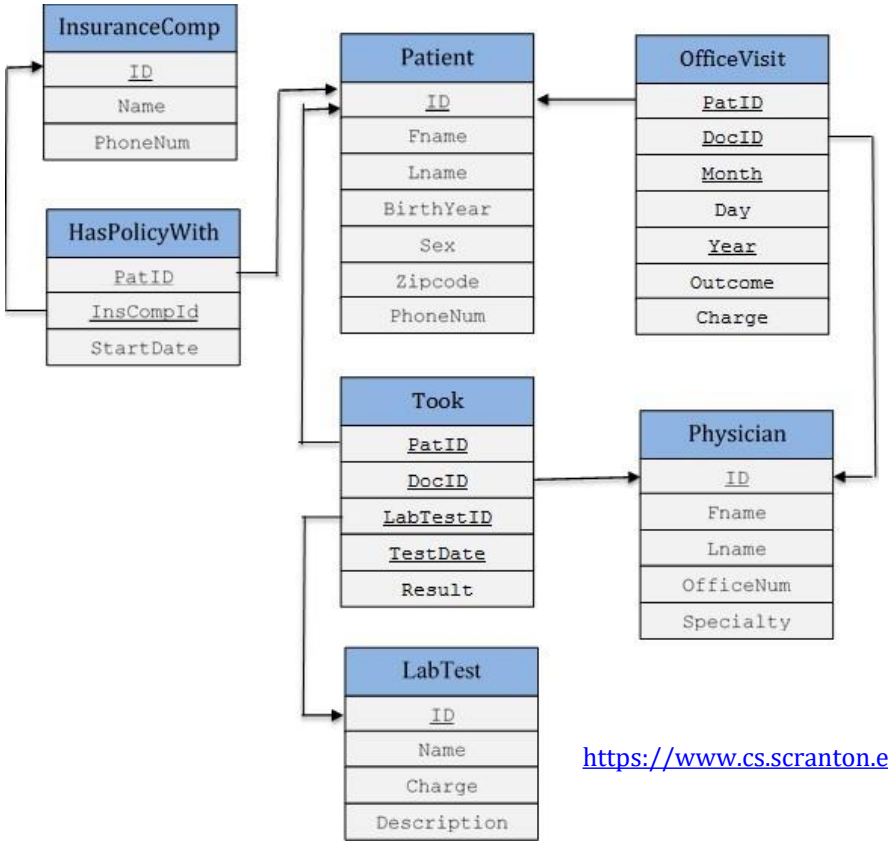
- Historical record for clinical care
- Communication (continuity of care)
- Preventive care (immunisation register, risks)
- Quality assurance (deviation in baby weight)
- Legal record (litigation, accrediting)
- Financial (coding for bills, prove charges valid)
- Prospective research (screening, tracking patients)
- Retrospective research - data mining (look across existing records)

Database

How can we keep EMRs?
Common requirement is for data storage and communications.

A database is needed to store EMRs.
EMRs are stored in a secure **computer databases** within a healthcare organization.

Databases can be used to store EMRs including **patient demographic data**, laboratory results, MRI, and **radiology ima**ges, patient diagnosis, treatment, as well as patient follow up information.



https://www.cs.scranton.edu/

Database

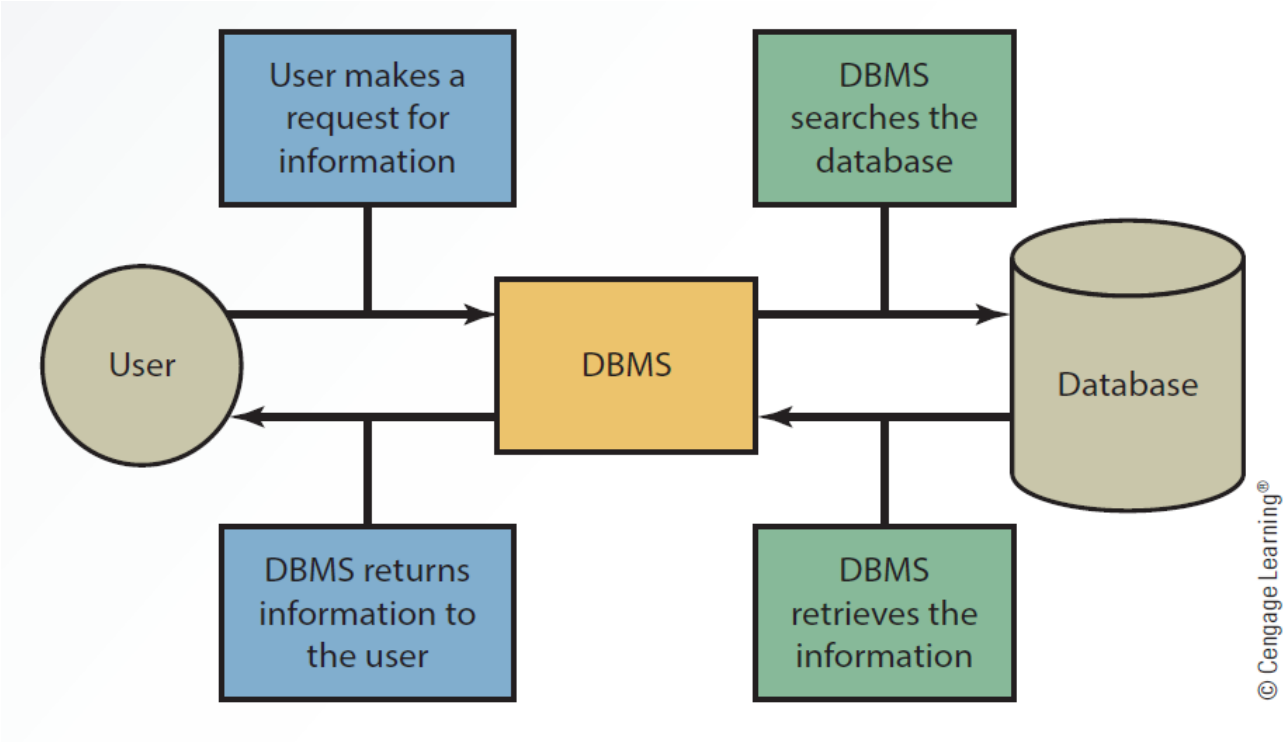
An organized collection of related data or structured information that is stored in a central location or in multiple locations (distributed database). Using database, the data, information and knowledge, can be accessed, managed and organized.

Data hierarchy: it refers to the systematic structuring and organization of data involving fields, records, and files. The three levels of the data hierarchy are attribute (field), record and file.

Database management system (DBMS)

- A software that is designed for creating, managing, storing, retrieving, maintaining, and accessing database files. DBMS also provides rules to validate and manipulate data in databases.
- DBMS makes using databases more efficient as well as cost-efficient.

Interaction between the User, DBMS and database



Database

Basic definitions

- **Database:**
  - A collection of related data.
- **Data:**
  - Known facts or information that have an implicit meaning and can be used to analyze, or plan something.
- **Mini-world in database:**
  - Some parts of the real-world data that is stored in a database. For instance, student grades and transcripts at a university.
- **Database Management System (DBMS):**
  - A software system to manage, facilitate, and manipulate computerized database.
- **Database System:**
  - The DBMS software system together with the data make a database system. the applications can also be included in a database system.

Types of data in a database

- Internal
  - Collected within an organization
  - Stored in the organization's internal databases
- External
  - Comes from a variety of resources
  - Stored in a data warehouse

Data Types

- Text
  - Fixed length (1 to 64 kB)
- Memo/Note
  - Variable length (unlimited size - cannot be indexed)
  - Often used to store XML
- Numeric

• Byte	1 byte	0 to 255
• Boolean	2 bytes	True or False
• Integer	2 bytes	-32,768 to 32,767
• Long	4 bytes	-2,147,483,648 to 2,147,483,647
• Floating	4 bytes	1.401298E-45 to 3.402823E38
• Double	8 bytes	4.9406E-324 to 1.7976E308
• Currency	8 bytes	

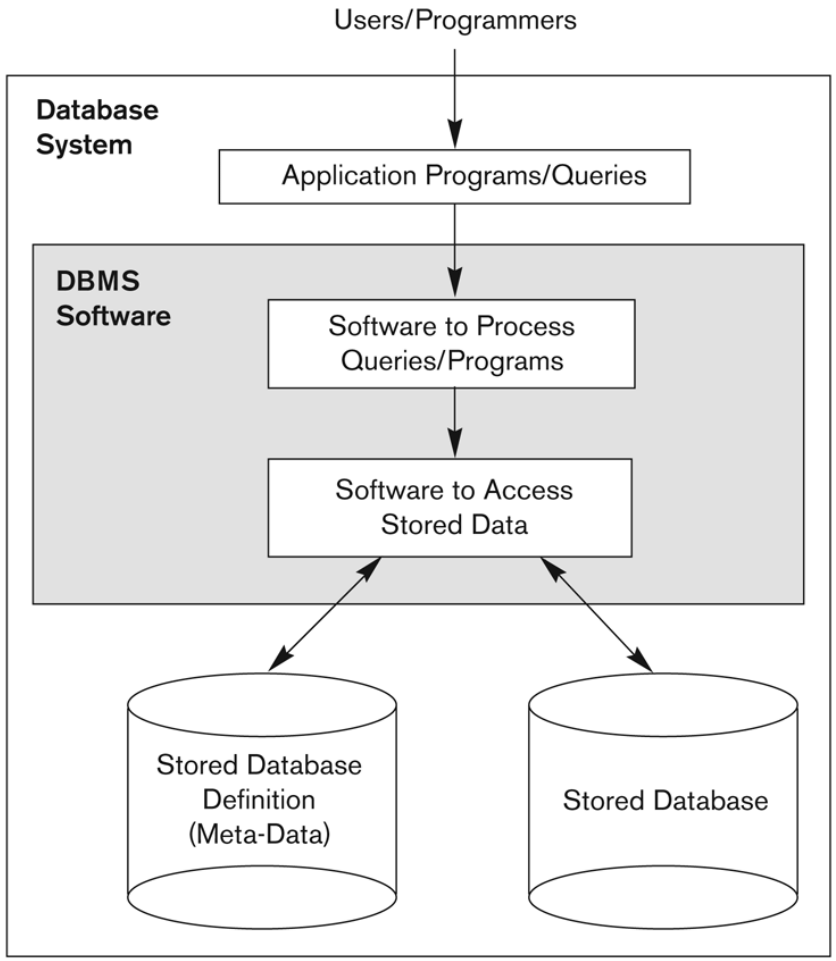
- Date/Time (international attributes)
- Logical
- Objects
  - Any type of data supported by the machine
  - Pictures, sound, video . . .

MS Access Data Types

Text	Use for text or combinations of text and numbers. 255 characters maximum	
Memo	Memo is used for larger amounts of text. Stores up to 65,536 characters. **Note:** You cannot sort a memo field. However, they are searchable	
Byte	Allows whole numbers from 0 to 255	1 byte
Integer	Allows whole numbers between -32,768 and 32,767	2 bytes
Long	Allows whole numbers between -2,147,483,648 and 2,147,483,647	4 bytes
Single	Single precision floating-point. Will handle most decimals	4 bytes
Double	Double precision floating-point. Will handle most decimals	8 bytes
Currency	Use for currency. Holds up to 15 digits of whole dollars, plus 4 decimal places. **Tip:** You can choose which country's currency to use	8 bytes
AutoNumber	AutoNumber fields automatically give each record its own number, usually starting at 1	4 bytes
Date/Time	Use for dates and times	8 bytes
Yes/No	A logical field can be displayed as Yes/No, True/False, or On/Off. In code, use the constants True and False (equivalent to -1 and 0). **Note:** Null values are not allowed in Yes/No fields	1 bit
Ole Object	Can store pictures, audio, video, or other BLOBs (Binary Large OBjects)	up to 1GB
Hyperlink	Contain links to other files, including web pages	
Lookup Wizard	Let you type a list of options, which can then be chosen from a drop-down list	4 bytes

Simplified database system environment

The five major part of a database are hardware, software, data, procedure system, and database access language.



Users/Programmers DATABASE Application Programs/Queries SYSTEM DBMS Software

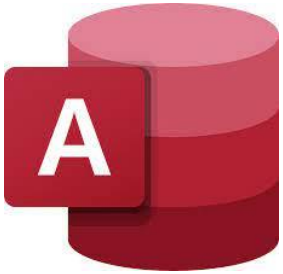
Typical DBMS Functionality

- *Create* a suitable database considering its data types, data structures, data relationship
- *Construct* the initial database and its contents on a secondary storage
- *Manipulating* and organizing data in the database:
  - Retrieval: querying in database, extracting data, generating reports
  - Modification: Insertions, deletions, modification and updates data/records
  - Accessing the data in the database via Web or internal applications
- *Processing* and *Sharing* by a set of concurrent users and application programs – but all data should still be valid
- Protection from unauthorized access
- Presentation and visualization of a set of data in a database
- Maintaining the database and associated programs over the lifetime of the database application

Examples of Commercial /Enterprise Systems

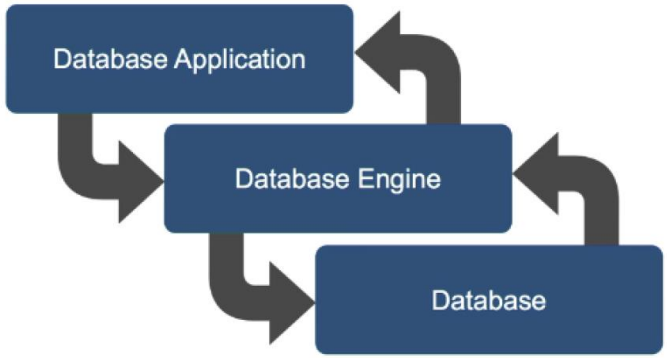
–Oracle
–MS SQL Server (Microsoft)
–PostgreSQL (Unix)
–Informix, DB2, SQL/DS (IBM)
–MongoDB

Desktop systems
–Access (Microsoft)
–Visual FoxPro (Microsoft), dBASE, Paradox…
–Open source (MySQL)



DBMS architecture

- Database application collects user input in the form of instructions.
- Database engine translates those instructions into code that's then used to retrieve information from the database itself.
- Once the information is retrieved, the process reverses.
- Database passes the information back into the engine.
- Translates that into a usable format (on-screen or in a printed report).



Example of a database (with a conceptual data model)

- **Mini-world for the example:**
  - Part of a UNIVERSITY environment.

- **Some mini-world *entities*:**
  - STUDENTs
  - COURSEs
  - SECTIONs (of COURSEs)
  - (academic) DEPARTMENTs
  - INSTRUCTORs

Example of a database (with a conceptual data model)

- **Some mini-world *relationships*:**
  - SECTIONs *are of specific* COURSEs
  - STUDENTs *take* SECTIONs
  - COURSEs *have prerequisite* COURSEs
  - INSTRUCTORs *teach* SECTIONs
  - COURSEs *are offered by* DEPARTMENTs
  - STUDENTs *major in* DEPARTMENTs

- Note: The above entities and relationships are typically expressed in a conceptual data model, for example, the ENTITY-RELATIONSHIP data model.

Example of a simple database

a database that store student and course information

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

File access

A file access is defined to control access to data in databases such as accessing to specified tables and columns.

The database administrator/organization is responsible to define access permissions by creating an access list for each author.

There are four method for accessing files:

Methods for accessing files / sequential access file structure

The simplest access method. In this method, records are organized and processed in sequential mode.

Organized based on a primary key.

- For example, Social security numbers or account numbers

This approach is usually used for **backup and archive files** as they rarely need updating.

Methods for accessing files / direct access

Direct access (also known as *relative access*) file systems is usually a specific access to database administrators (DBAs) and end-users do not have this access. End-users should never have direct access to DBs, otherwise, they can change records such as sales reports, making ad-hoc queries, etc.

Methods for accessing files / random access file structure

In random access, records can be accessed and modified in any order irrespective of the physical locations in the storage. This means the data is located exactly as it appears in memory.

It is **fast and very effective** when a small database needs to be processed daily or weekly.

Random files are not especially transportable.

Methods for accessing files / indexed sequential access method (ISAM)

In the ISAM systems, records *can be accessed sequentially or randomly* depending on the number of records that need to be accessed.

- Random access is used when the number of records is small
- Sequential access is used when a large number of records need to be processed.

Uses an index structure has two parts:

- Indexed value for fast accessing to records.
- Pointer to the disk location of the record matching the indexed value

Logical database design

Refers to the process of database implementation.

Physical view

- Involves how data is stored on and retrieved from storage media. This view mainly refers to the actual designing of the database.
  - Hard disks, magnetic tapes, or CDs

Logical view

- This process refers to how users can see the information and how the information can be managed and retrieved. This view mainly refers to collection of information about business needs.

- This process may include more than one logical view of data, depending on the user

Logical database design

Data model

- Determines how data is created, visualized, represented, organized, managed, and maintained.

- Contains
  - Data structure
  - Operations
  - Integrity rules

Hierarchical model

- Relationships between records form a treelike structure

Informal definitions in database

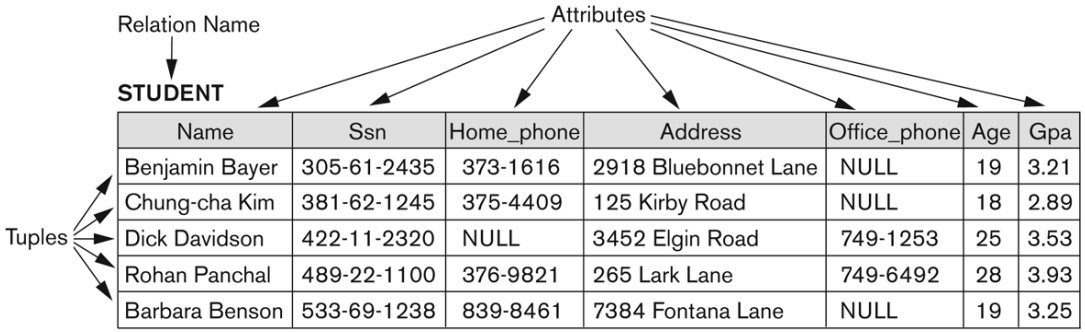
Informally, a **relation** looks like a **table** of values.

A relation typically contains a **set of rows (records)**. In the formal model, rows are also called **tuples**. Each row can have multiple columns (fields).

The data elements in each **row** represent certain facts that correspond to a real-world **entity** or **relationship**.

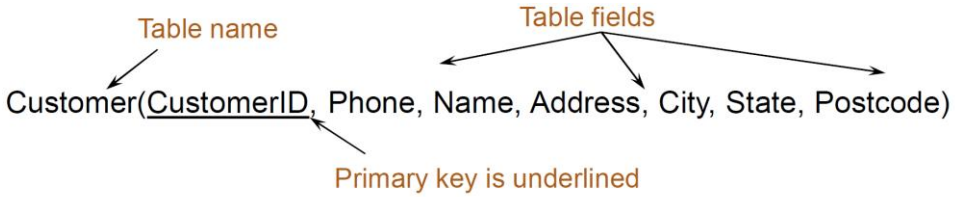
Each **column** has a column header that gives an indication of the meaning of the data items in that column. In the formal model, the column header is also called an **attribute name** (or just **attribute**)

Example of a relation



Name	Ssn	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25

Notation



CustomerID	Phone	LastName	FirstName	Address	City	State	Postcode
1	612-9966-7777	Johnson	Martha	125 Main Street	Kingsford	NSW	2032
2	0419-456789	Smith	Jack	873 Elm Street	Bowling Green	SA	5456
3	02-9777-7575	Washington	Elroy	95 Easy Street	Smithville	NSW	2346
4	07-333-9494	Adams	Samuel	746 Brown Drive	Alvaton	QLD	7212
5	07-474-4746	Rabitz	Victor	645 White Avenue	Bowling Green	QLD	7210

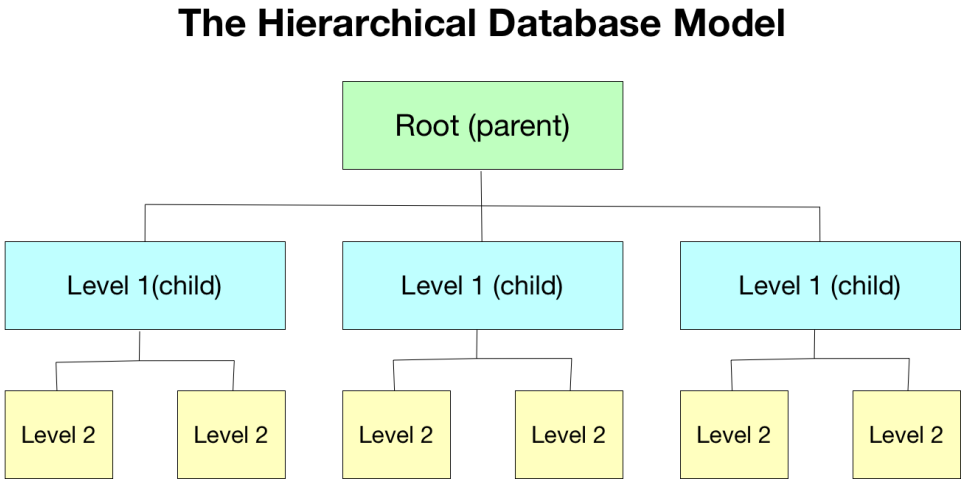
Definition summary

Informal Terms	Formal Terms
Table	Relation
Column Header	Attribute
All possible Column Values	Domain
Row	Tuple
Table Definition	Schema of a Relation
Populated Table	State of the Relation

Important types of database models

- Hierarchical database model
- Relational model
- Network model
- Object-oriented database model
- Entity-relationship model
- Document model
- Entity-attribute-value model
- Star schema

Hierarchical database model (HDM):
It is a data model in which data is represented in the form of records and is organised in the form of the tree-like structure. HDM has been using for widely in recording file system data, especially in banking and telecommunications.



Relational model

A Relational Database (RD), is organising relational data in the form of tables, columns and rows, as well as their relationship properties. RD uses a two-dimensional table of rows and columns of data, in which Rows are records and Columns are fields. Relational model was originally introduced in 1970 by an IBM researcher, Edgar Codd.

In RD models, each table has a **Primary key**, to be connected to other related tables using a **Foreign key** from those tables.
RD models are currently most popular model for developing database applications.

- These tables are linked (related) through the use of a common field(s).
- Allows complex and related data to be stored and analysed.
- The relational DBMS (RDBMS) manages concurrency.

Purchase table

Transaction ID	Customer ID	Product ID	Purchange date
1112	24221	8977	03-22-2010
1113	24222	8978	03-22-2010
1114	24223	8979	03-22-2010

Customer ID	Customer	Address
24221	Bob	123 East street
24222	Alice	223 Main street
24223	Martha	465 North street

Customer table

Product ID	Name	Price
8977	Banana	.79
8978	TV	400
8979	Watch	50

Product table

Concurrency and locking

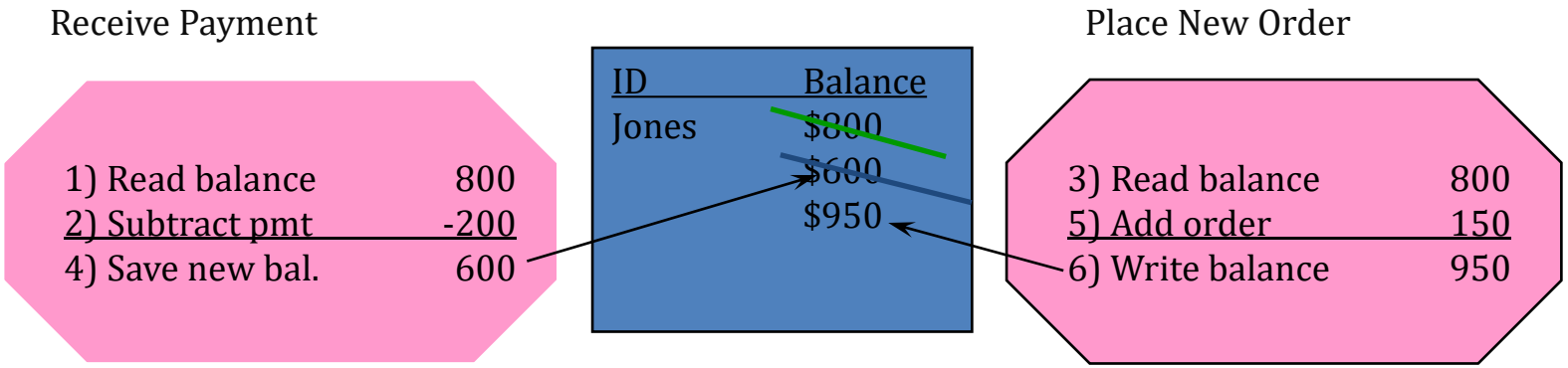
Concurrency - multiple users or processes changing the same data at the same time.

Approaches

- Delayed, batch updates
- Record locking
  - pessimistic - if any user gets hold of the data, then prevent all other users from accessing it in any way.
  - optimistic - several users can get the data, but after one has updated it all subsequent update attempts will fail.

Example of two processes

- Receive payment ($200)
- Place new order ($150)
- Initial balance $800
  - Result should be $800 -$200 + $150 = $750
  - Interference result is either $600 or $950



Relational Database Management System (RDBMS):
RDBMS (or just RDB) is a program/interface that allows users to manage, create, insert, modify, or remove records in a relational database. Most RDBMS use the structured query language (SQL) to make a query (i.e., an action on the database) and access the data in a database.

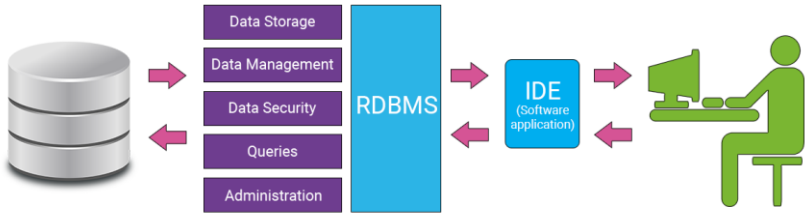
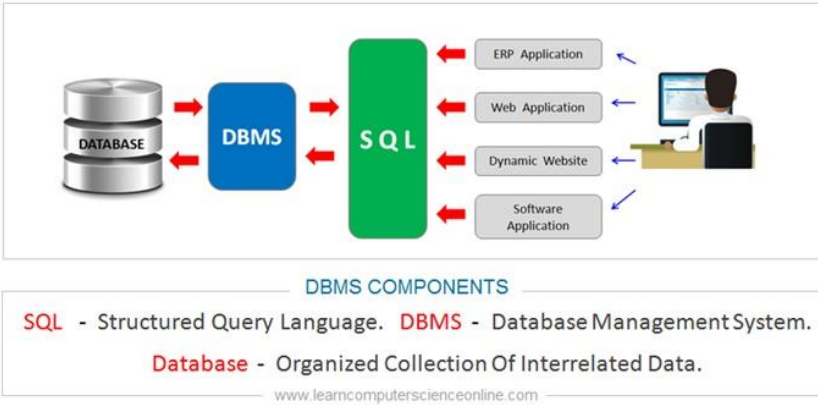


Figure credit: Radu, https://vertabelo.com/

SQL is a user-friendly programming language to access the data stored in a relational database. It is based on relational algebra. SQL has a quite simple syntax and very similar to the English language, which makes it very easy to communicate with database and the RDBMS to write, read, and interpret the data.



Structure of relational databases

- Uses a series of cells, grouped into vertical columns (attributes, values, fields) and horizontal rows (tuples, records)
- Cells store a single piece of information (they are atomic and should not be split into smaller units) – think of entering an address
- Columns store one (and only one) type of information
- Column names are unique (best not to contain spaces)
- Order in which columns appear does not matter
- Rows are unique.

Primary key

- A field (or combination of fields) that uniquely identifies a record (row) in a table. A primary key can not be duplicate, which means the same value should not exist more than once in the primary key column.

Foreign key

- In relational database, a field (or combination of fields) in a table that matches the primary key column of another table. A foreign key provides a link between data in two different tables.
- A foreign key can be used to cross-reference tables. This is the way in which tables are related.

Keys and Indexes

The relational model is based on the idea of controlled redundancy.
The controlling mechanism is the concept of keys.
Keyed fields are typically indexed (logically sorted on the key field or a combination of fields) to improve performance.
Indexes speed up finding data, but slow down inserting, updating or deleting data.

PatientandProcedures

PatProcID	MedRecID	FirstName	Surname	ProcedureDate	ProcedureType	ProcedureOperator	ProcedureData
1	101	Silas	Farquard	10-Jul-17	ECG	Nurse1	120 BPM
2	102	Ella	McFlame	09-Jun-17	SPIRO	Operator1	4 L
3	103	Thomas	Tankengine	08-Mar-17	BLOOD	Nurse2	1.1 Neutrophils
4	102	Ella	McFlame	23-May-17	ECG	Operator2	73 BPM

In example MedRecID would be a good primary key (but current database structure does not allow this so a new key PatProcID was introduced)

Database

In order to ensure **referential integrity** (consistency across tables) if a foreign key is not null then its value must match a value in the primary key in the table(s) to which it is related.

Primary keys must be unique for any record, foreign keys typically are not.

PatientandProcedures

PatProcID	MedRecID	FirstName	Surname	ProcedureDate	ProcedureType	ProcedureOperator	ProcedureData
1	101	Silas	Farquard	10-Jul-17	ECG	Nurse1	120 BPM
2	102	Ella	McFlame	09-Jun-17	SPIRO	Operator1	4 L
3	103	Thomas	Tankengine	08-Mar-17	BLOOD	Nurse2	1.1 Neutrophils
4	102	Ella	McFlame	23-May-17	ECG	Operator2	73 BPM

Read more here:
https://database.guide/what-is-referential-integrity/

Referential integrity protects related data that is stored in multiple tables.
For example, it would prevent a user in a Patients table from being deleted if the Patient's ID also appears in the Procedures table.

Relational database integrity rules

Entity Integrity

- All records must be unique
- The primary key must have a valid value

Referential Integrity

- Foreign key must have a matching entry in the primary key in the related table or must be null. e.g. a patient may not yet have a procedure, but if the patient does then the patient primary key (MedRecID) must exist in the Patient table.
- Prevents inconsistent data from being created in the database by ensuring that any data shared between tables remains consistent (e.g. prevent or cascade updates and deletions).

Database maintenance involves locking the tables, re-creating indexes and checking for orphaned records.

Procedures

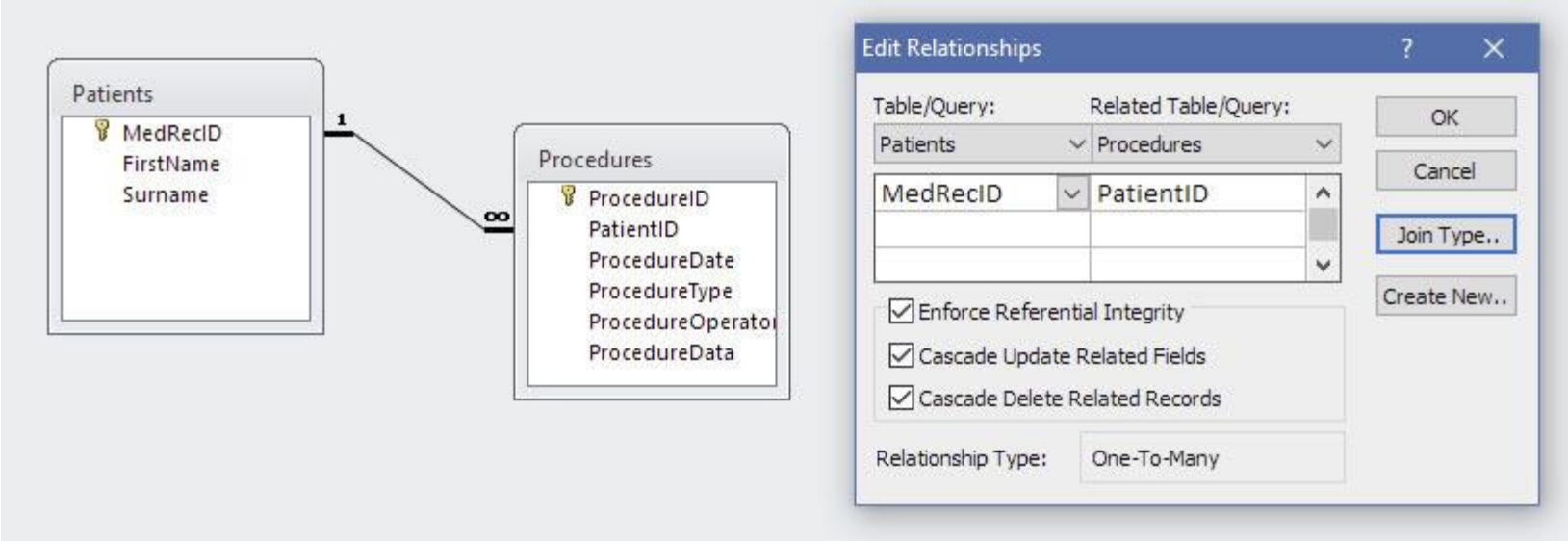
ProcedureID	PatientID	ProcedureDate	ProcedureType	ProcedureOperator	ProcedureData
1	101	10-Jul-17	ECG	Nurse1	120 BPM
2	102	09-Jun-17	SPIRO	Operator1	4 L
3	103	08-Mar-17	BLOOD	Nurse2	1.1 Neutrophils
4	102	23-May-17	ECG	Operator2	73 BPM

Patients

MedRecID	FirstName	Surname
101	Silas	Farquard
102	Ella	McFlame
103	Thomas	Tankengine

Tip: Use a relational model with multiple tables for a better design.

- Note that referential integrity can (should) also be enforced by the RDMBS.



Patients

MedRecID	FirstName	Surname
101	Silas	Farquard
102	Ella	McFlame
103	Thomas	Tankengine

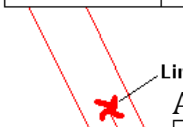
Procedures

ProcedureID	PatientID	ProcedureDate	ProcedureType	ProcedureOperator	ProcedureData
1	101	10-Jul-17	ECG	Nurse1	120 BPM
2	102	09-Jun-17	SPIRO	Operator1	4 L
3	103	08-Mar-17	BLOOD	Nurse2	1.1 Neutrophils
4	102	23-May-17	ECG	Operator2	73 BPM

An example of a database that has not enforced referential integrity

Artist

artist_id	artist_name
1	Bono
2	Cher
3	Nuno Bettencourt



Album

artist_id	album_id	album_name
3	1	Schizophonic
4	2	Eat the rich
3	3	Crave (single)

Database

Name the primary key in the Patients table and the primary key and foreign keys in the Patients and Procedures tables

Use this link: https://app.sli.do/event/jrfub4lm/live/polls

Or

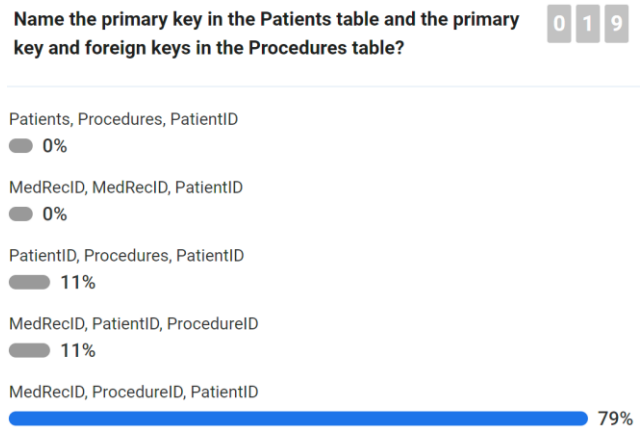
use this Slido code: #396231

Procedures

ProcedureID	PatientID	ProcedureDate	ProcedureType	ProcedureOperator	ProcedureData
1	101	10-Jul-17	ECG	Nurse1	120 BPM
2	102	09-Jun-17	SPIRO	Operator1	4 L
3	103	08-Mar-17	BLOOD	Nurse2	1.1 Neutrophils
4	102	23-May-17	ECG	Operator2	73 BPM

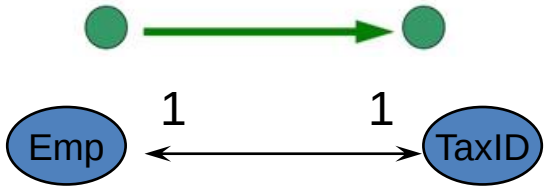
Patients

MedRecID	FirstName	Surname
101	Silas	Farquard
102	Ella	McFlame
103	Thomas	Tankengine



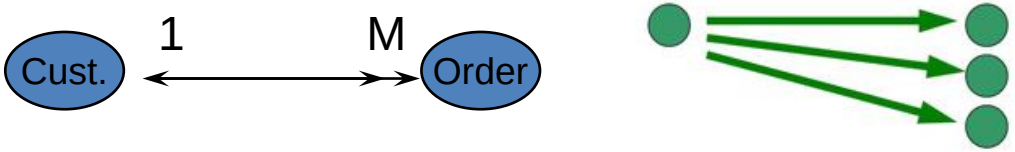
Type of relationships

One-to-one (1:1): one-to-one relationship is a link between the records in two tables in which **one record in table A can be linked to one related record in table B**, and a one record in table B can be linked to only one record in table A.



One-to-many (1:N) or Many-to-one (N:1): In a one-to-many relationship, **one record in a table A, can be associated with zero, one or more records in table B**, but a record in table B can only be associated with one matching records in table A.

•For example, a mother can have more than one child, but a child can have only one biological mother.

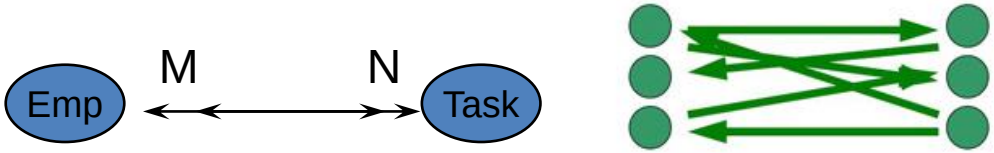


Many-to-many (M:N): many-to-many relationship happens when **multiple records in table A can be associated with multiple records in table B**.

•For example, an author can write more than one book and a book can be written by more than one author.

•Another example, customers can buy various products, and products can be purchased by different customers.

https://port.sas.ac.uk/



Normalization

Relationships represent business rules

- Sometimes common-sense
- Sometimes unique to an organization
- Most 1:1 relationships are created by business rules.

For example, a student should not have more than two instruments checked out at the same time.

Table and normalization

Normalization in database refers to a process **aiming to organize data in tables** and establish relationships between those tables. This process allows users to **filter out redundant data in a database**. This sometime involves re-structuring the tables.

Why normalisation?

There are four goals of the normalization process (some you have already seen)

- Eliminating redundant data (for example, storing the same data in more than one table)
- Ensuring data dependencies make sense (only storing related data in a table)
- Eliminating modification anomalies (only edit a piece of information in one place)
- Save space

A correctly normalized database should have the following characteristics:

- Scalar (atomic) values in each field.
- Absence of redundancy.
- Minimal use of null values.
- Minimal loss of information.

Read more here:
https://database.guide/what-is-normalization/

Table and normalization

To normalize tables, there are a series of rules that your databas's table structure must pass.

-Individual rules are called **normal forms** and they must be tested *in order* by passing through a series of checks

-We apply three normal forms (1NF, 2NF and 3NF) in this course.

Summary

- 1st NF: No repeating elements or groups of elements
- 2nd NF: No partial dependencies on a concatenated key
- 3rd NF: No dependencies on non-key attributes

Read more here:
https://database.guide/what-is-normalization/

Goal is to produce a list of *nicely behaved* tables. As summary:

- Objects are described by fields
- Each record contains data for one object
- Every table has a primary key
- The smallest set of columns that uniquely identifies any row
- Primary keys can span more than one column (concatenated keys)
- If needed we can create a primary key to insure uniqueness (e.g., PatientID, MedicationID, . . .

Database anomaly

Anomalies are problems that can happen in poorly planned databases where all the data is stored in one table (a flat-file database). For example, a duplicate data (row) is updated only in one place and not in the whole table. There are three types of anomalies: update, deletion and insertion anomalies.

Normalization removes processing anomalies.

There are three types of anomalies?

a) Insertion anomaly: It happens when the user can not insert a certain attribute (field) in the table without the adding value to other fields.

b) Update anomaly: It happens when updating/modifying a table will result in data redundancy and partial update.

c) Deletion Anomaly: It happens when deleting some data in a table results in the deletion of some other data.

Example / database anomaly

Prof_ID	Prof_Name	Dept.	Course Group
39404	Ashish	Marketing	???
39445	Sonam	Product	Sec B
43576	Anu Priya	Finance	Sec C
54325	Anu Priya	Finance	Sec C
99823	Anushka	HR	Sec D
14325	Anushka	HR	Sec E

An **insertion anomaly**. Until the new faculty member, Dr. Ashish, is assigned to teach at least one course, his details cannot be inserted.

An **update anomaly**. Updating Dr Anushka's department needs to be updated at two different place to maintain consistency.

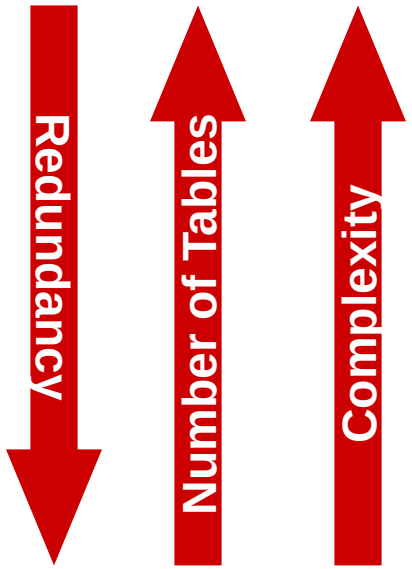
A **deletion anomaly**. All information about Dr. Sonam is lost when he temporarily stops to be assigned to any courses (e.g., if SEC B removes).

Levels of normalization

Levels of normalization is based on the amount of redundancy in the database.

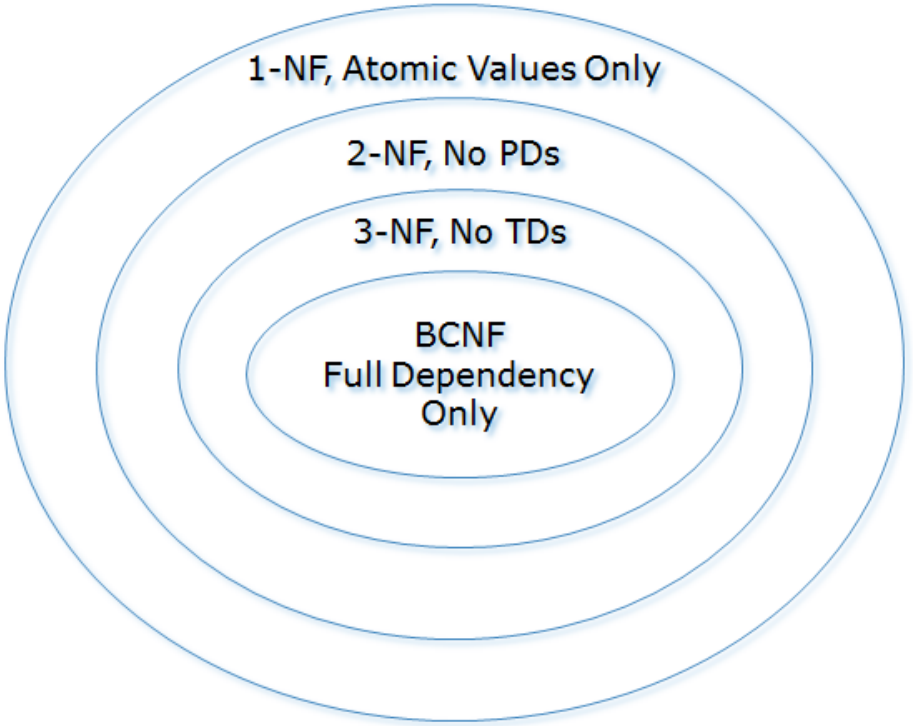
Various levels of normalization are:

- First Normal Form (1NF)
- Second Normal Form (2NF)
- Third Normal Form (3NF)
- Boyce-Codd Normal Form (BCNF)
- Fourth Normal Form (4NF)
- Fifth Normal Form (5NF)
- Domain Key Normal Form (DKNF)



Most databases should be 3NF or BCNF in order to avoid the database anomalies.

Levels of Normalization



*PD: Partial Dependencies
*TD: Transitive Dependencies

Each higher level is a subset of the lower level

First Normal Form (1NF)

A table is considered to be in 1NF if all the fields contain only scalar values (as opposed to list of values) and no repeating groups.

Example (Not 1NF)

ISBN	Title	AuName	AuPhone	PubName	PubPhone	Price
0-321-32132-1	Balloon	Sleepy, Snoopy, Grumpy	321-321-1111, 232-234-1234, 665-235-6532	Small House	714-000-0000	$34.00
0-55-123456-9	Main Street	Jones, Smith	123-333-3333, 654-223-3455	Small House	714-000-0000	$22.95
0-123-45678-0	Ulysses	Joyce	666-666-6666	Alpha Press	999-999-9999	$34.00
1-22-233700-0	Visual Basic	Roman	444-444-4444	Big House	123-456-7890	$25.00

Author and AuPhone columns are not scalar

Read more here:
https://www.w3schools.in/dbms/database-normalization
https://www.guru99.com/database-normalization.html
https://www.javatpoint.com/dbms-normalization

First Normal Form (1NF)

For a table to be in the **1NF**, it should follow the following three rules:

1. It must only have single(atomic) valued attributes/columns.
2. Values in each column should have a same data type.
3. All the columns/attributes in a table must have unique names.
4. Each Record needs to be unique and there are no repeating groups.

1NF - Decomposition

1. All fields should be atomic.
2. All items in the repeating group should be placed in a new table.
3. Provide a primary key for each new table that was produced in step 2.
4. Duplicate in the new tables generated from step 3, the primary key of the table from which the repeating group was extracted.

Example 1 - 1NF

TransID	RentDate	CustomerID	LastName	Phone	Address	VideoID	Copy#	Title	Rent
1	18/4/06	3	Washington	02-9777-7575	95 Easy Street	1	2	2001: A Space Odyssey	$1.50
1	18/4/06	3	Washington	02-9777-7575	95 Easy Street	6	3	Clockwork Grape	$1.50
2	30/4/06	7	Lasater	612-888-4474	67 S. Ray Drive	8	1	Spiderman II	$1.50
2	30/4/06	7	Lasater	612-888-4474	67 S. Ray Drive	2	1	Apocalypse Never	$2.00
2	30/4/06	7	Lasater	612-888-4474	67 S. Ray Drive	6	1	Clockwork Grape	$1.50
3	18/4/06	8	Jones	612-452-1162	867 Lakeside Drive	9	1	Luggage Of The Gods	$2.50
3	18/4/06	8	Jones	612-452-1162	867 Lakeside Drive	15	1	Sex, Lies and DVDs	$2.00
3	18/4/06	8	Jones	612-452-1162	867 Lakeside Drive	4	1	Boy And His Dog	$2.50
4	18/4/06	3	Washington	02-9777-7575	95 Easy Street	3	1	Blues Brothers	$2.00
4	18/4/06	3	Washington	02-9777-7575	95 Easy Street	8	1	Spiderman II	$1.50
4	18/4/06	3	Washington	02-9777-7575	95 Easy Street	13	1	Surf Nazis Must Die	$2.50
4	18/4/06	3	Washington	02-9777-7575	95 Easy Street	17	1	Witches of Eastwick	$2.00

Example 1 - 1NF

TransID	RentDate	CustomerID	LastName	Phone	Address	VideoID	Copy#	Title	Rent
1	18/4/06	3	Washington	02-9777-7575	95 Easy Street	1	2	2001: A Space Odyssey	$1.50
1	18/4/06	3	Washington	02-9777-7575	95 Easy Street	6	3	Clockwork Grape	$1.50
2	30/4/06	7	Lasater	612-888-4474	67 S. Ray Drive	8	1	Spiderman II	$1.50
2	30/4/06	7	Lasater	612-888-4474	67 S. Ray Drive	2	1	Apocalypse Never	$2.00
2	30/4/06	7	Lasater	612-888-4474	67 S. Ray Drive	6	1	Clockwork Grape	$1.50
3	18/4/06	8	Jones	612-452-1162	867 Lakeside Drive	9	1	Luggage Of The Gods	$2.50
3	18/4/06	8	Jones	612-452-1162	867 Lakeside Drive	15	1	Sex, Lies and DVDs	$2.00
3	18/4/06	8	Jones	612-452-1162	867 Lakeside Drive	4	1	Boy And His Dog	$2.50
4	18/4/06	3	Washington	02-9777-7575	95 Easy Street	3	1	Blues Brothers	$2.00
4	18/4/06	3	Washington	02-9777-7575	95 Easy Street	8	1	Spiderman II	$1.50
4	18/4/06	3	Washington	02-9777-7575	95 Easy Street	13	1	Surf Nazis Must Die	$2.50
4	18/4/06	3	Washington	02-9777-7575	95 Easy Street	17	1	Witches of Eastwick	$2.00

TransID	RentDate	CustID	Phone	LastName	FirstName	Address	City	State	Postcode
1	18/4/06	3	9777-7575	Washington	Elroy	95 Easy Street	Smith's Grove	NSW	2171
2	30/4/06	7	9888-4474	Lasater	Les	67 S. Ray Drive	Portland	SA	5148
3	18/4/06	8	9452-1162	Jones	Charlie	867 Lakeside Drive	Castalian Springs	SA	5031
4	18/4/06	3	9977-7575	Washington	Elroy	95 Easy Street	Smith's Grove	NSW	2171

TransID	VideoID	Copy#	Title	Rent
1	1	2	2001: A Space Odyssey	$1.50
1	6	3	Clockwork Grape	$1.50
2	8	1	Spiderman II	$1.50
2	2	1	Apocalypse Never	$2.00
2	6	1	Clockwork Grape	$1.50
3	9	1	Luggage Of The Gods	$2.50
3	15	1	Sex, Lies and DVDs	$2.00
3	4	1	Boy And His Dog	$2.50
4	3	1	Blues Brothers	$2.00
4	8	1	Spiderman II	$1.50
4	13	1	Surf Nazis Must Die	$2.50
4	17	1	Witches of Eastwick	$2.00

Example 2 - 1NF

ISBN	Title	AuName	AuPhone	PubName	PubPhone	Price
0-321-32132-1	Balloon	Sleepy, Snoopy, Grumpy	321-321-1111, 232-234-1234, 665-235-6532	Small House	714-000-0000	$34.00
0-55-123456-9	Main Street	Jones, Smith	123-333-3333, 654-223-3455	Small House	714-000-0000	$22.95
0-123-45678-0	Ulysses	Joyce	666-666-6666	Alpha Press	999-999-9999	$34.00
1-22-233700-0	Visual Basic	Roman	444-444-4444	Big House	123-456-7890	$25.00

ISBN	Title	PubName	PubPhone	Price
0-321-32132-1	Balloon	Small House	714-000-0000	$34.00
0-55-123456-9	Main Street	Small House	714-000-0000	$22.95
0-123-45678-0	Ulysses	Alpha Press	999-999-9999	$34.00
1-22-233700-0	Visual Basic	Big House	123-456-7890	$25.00

ISBN	**AuName**	AuPhone
0-321-32132-1	Sleepy	321-321-1111
0-321-32132-1	Snoopy	232-234-1234
0-321-32132-1	Grumpy	665-235-6532
0-55-123456-9	Jones	123-333-3333
0-55-123456-9	Smith	654-223-3455
0-123-45678-0	Joyce	666-666-6666
1-22-233700-0	Roman	444-444-4444

Example 3 - 1NF

Student Scheme: {<u>Student#</u>, student name, student family, Advisor, Adv-Room, Class1#, Class2#, Class3#}

Student#	Student name	Student family	Advisor	Adv-Room	Class1#	Class2#	Class3#
1022	Richard	Davenport	Jones	412	101-07	143-01	159-02
4123	Alistair	Forrest	Smith	216	101-07	143-01	179-04
3217	Diako	Ebrahimi	Mark,Helen	148,151	101-07	103-04	211-04

Example 3 - 1NF

Student Scheme: {Student#, student name, student family, Advisor, Adv-Room, Class1#, Class2#, Class3#}

Student#	Student name	Student family	Advisor	Adv-Room	Class1#	Class2#	Class3#
1022	Richard	Davenport	Jones	412	101-07	143-01	159-02
4123	Alistair	Forrest	Smith	216	101-07	143-01	179-04
3217	Diako	Ebrahimi	Mark,Helen	148,151	101-07	103-04	211-04

Student Scheme: {Student#, student name, student family, Advisor, Avd-Room, Class#}

Student#	Student name	Student family	Advisor	Adv-Room	Class#
1022	Richard	Davenport	Jones	412	101-07
1022	Richard	Davenport	Jones	412	143-01
1022	Richard	Davenport	Jones	412	159-02
4123	Alistair	Forrest	Smith	216	101-07
4123	Alistair	Forrest	Smith	216	143-01
4123	Alistair	Forrest	Smith	216	159-02
3217	Diako	Ebrahimi	Mark	148	101-07
3217	Diako	Ebrahimi	Mark	148	103-04
3217	Diako	Ebrahimi	Mark	148	211-04
3217	Diako	Ebrahimi	Helen	151	101-07
3217	Diako	Ebrahimi	Helen	151	103-04
3217	Diako	Ebrahimi	Helen	151	211-04

Still have problems
–Replication
–Hidden dependencies

Example 3 - 1NF

Student Scheme: {Student#, student name, student family, Advisor, Avd-Room, Class#}

Student#	Student name	Student family	Advisor	Adv-Room	Class#
1022	Richard	Davenport	Jones	412	101-07
1022	Richard	Davenport	Jones	412	143-01
1022	Richard	Davenport	Jones	412	159-02
4123	Alistair	Forrest	Smith	216	101-07
4123	Alistair	Forrest	Smith	216	143-01
4123	Alistair	Forrest	Smith	216	159-02
3217	Diako	Ebrahimi	Mark	148	101-07
3217	Diako	Ebrahimi	Mark	148	103-04
3217	Diako	Ebrahimi	Mark	148	211-04
3217	Diako	Ebrahimi	Helen	151	101-07
3217	Diako	Ebrahimi	Helen	151	103-04
3217	Diako	Ebrahimi	Helen	151	211-04

Student Scheme: {Student#, student name, student family, Advisor, Avd-Room}

Student#	Student name	Student family	Advisor	Adv-Room
1022	Richard	Davenport	Jones	412
4123	Alistair	Forrest	Smith	216
3217	Diako	Ebrahimi	Mark	148
3217	Diako	Ebrahimi	Helen	151

Class Scheme: {Student#, Class#}

Student#	Class#
1022	101-07
1022	143-01
1022	159-02
4123	101-07
4123	143-01
4123	159-02
3217	101-07
3217	103-04
3217	211-04
3217	101-07
3217	103-04
3217	211-04

Functional Dependencies

If one set of attributes in a table determines another set of attributes in that table, then the second set of attributes is said to be **functionally dependent on the first set** of attributes.

Example

ISBN	**Title**	**Price**
0-321-32132-1	Balloon	$34.00
0-55-123456-9	Main Street	$22.95
0-123-45678-0	Ulysses	$34.00
1-22-233700-0	Visual Basic	$25.00

Table Scheme: {ISBN, Title, Price}

Functional Dependencies: {ISBN} → {Title}

{ISBN} → {Price}

{Title, Price} → {ISBN}

Functional Dependencies

Example

AuID	AuName	AuPhone
1	Sleepy	321-321-1111
2	Snoopy	232-234-1234
3	Grumpy	665-235-6532
4	Jones	123-333-3333
5	Smith	654-223-3455
6	Joyce	666-666-6666
7	Roman	444-444-4444

Table Scheme: {AuID, AuName, AuPhone}

Functional Dependencies: {AuId} → {AuPhone}

{AuId} → {AuName}

{AuName, AuPhone} → {AuID}

Second Normal Form (2NF)

For a table to be in 2-NF, there are two requirements

- The database is in first normal form
- All **non-key** attributes in the table must be **functionally dependent on the entire primary key**

Note: 2-NF eliminates all the partial dependencies (PD)

Note: Remember that we are dealing with non-key attributes

Example 1 (Not 2NF)

Scheme → {Title, PubID, AuID, Price, AuAddress}

1. **Key → {Title, PubID, AuID}**
2. **{Title, PubID, AuID} → {Price}**
3. **{AuID} → {AuAddress}**
4. **AuAddress does not belong to a key**
5. **AuAddress functionally depends on AuID which is a subset of a key**

Read more here:
https://www.w3schools.in/dbms/database-normalization
https://www.guru99.com/database-normalization.html
https://www.javatpoint.com/dbms-normalization

Second Normal Form (2NF)

Example 2 (Not 2NF)

Scheme → {City, Street, HouseNumber, HouseColor, CityPopulation}

1. **key → {City, Street, HouseNumber}**
2. **{City, Street, HouseNumber} → {HouseColor}**
3. **{City} → {CityPopulation}**
4. **CityPopulation does not belong to any key.**
5. **CityPopulation is functionally dependent on the City which is a proper subset of the key**

Example 3 (Not 2NF)

Scheme → {studio, movie, budget, studio_city}

1. **Key → {studio, movie}**
2. **{studio, movie} → {budget}**
3. **{studio} → {studio_city}**
4. **studio_city is not a part of a key**
5. **studio_city functionally depends on studio which is a proper subset of the key**

2NF - Decomposition

1. If a data item is fully functionally dependent on only a part of the primary key, move that data item and that part of the primary key **to a new table**.
2. If other data items are functionally dependent on the same part of the key, place them in the new table also.
3. Make the partial primary key copied from the original table as the primary key for the new table. Place all items that appear in the repeating group in a new table.

Example 1 (Not 2NF)

Scheme → {Title, PubID, AuID, Price, AuAddress}

1. **Key → {Title, PubID, AuID}**
2. **{Title, PubID, AuID} → {Price}**
3. **{AuID} → {AuAddress}**
4. **AuAddress does not belong to a key**
5. **AuAddress functionally depends on AuID which is a subset of a key**

Convert to 2NF

Old Scheme → {<u>Title, PubID, AuID</u>, Price, AuAddress}

New Scheme → {<u>Title, PubID, AuID</u>, Price}

New Scheme → {<u>AuID</u>, AuAddress}

2NF - Decomposition

Example 2 (Not 2NF)
Scheme → {City, Street, HouseNumber, HouseColor, CityPopulation}

1. **key → {City, Street, HouseNumber}**
2. **{City, Street, HouseNumber} → {HouseColor}**
3. **{City} → {CityPopulation}**
4. **CityPopulation does not belong to any key.**
5. **CityPopulation is functionally dependent on the City which is a proper subset of the key**

Convert to 2NF

Old Scheme → {City, Street, HouseNumber, HouseColor, CityPopulation}

New Scheme → {City, Street, HouseNumber, HouseColor}

New Scheme → {City, CityPopulation}

2NF - Decomposition

Example 3 (Not 2NF)
Scheme → {studio, movie, budget, studio_city}

1. **Key → {studio, movie}**
2. **{studio, movie} → {budget}**
3. **{studio} → {studio_city}**
4. **studio_city is not a part of a key**
5. **studio_city functionally depends on studio which is a proper subset of the key**

Convert to 2NF

Old Scheme → {<u>studio</u>, <u>movie</u>, budget, studio_city}

New Scheme → {<u>studio, movie</u>, budget}

New Scheme → {<u>studio</u>, studio_city}

Example – 2NF

Student Scheme: {Student#, student name, student family, Advisor, Adv-Room}

Student#	Student name	Student family	Advisor	Adv-Room
1022	Richard	Davenport	Jones	412
4123	Alistair	Forrest	Smith	216
3217	Diako	Ebrahimi	Mark	148
3217	Diako	Ebrahimi	Helen	151

{Student#, Advisor} → {Adv-Room}
{Advisor} → {Adv-Room}
Adv-Room does not belong to any key.
Adv-Room is functionally dependent on the Advisor which is a proper subset of the key

Convert to 2NF

Advisor Scheme: {Student#, Advisor, Adv-Room}

Student#	Advisor	Adv-Room
1022	Jones	412
4123	Smith	216
3217	Mark	148
3217	Helen	151

Student Scheme: {Student#, Student name, Student family}

Student#	Student name	Student family
1022	Richard	Davenport
4123	Alistair	Forrest
3217	Diako	Ebrahimi

Example - 2NF

(<u>OrderNum</u>, OrderDate)
(<u>PartNum</u>, Description)
(<u>OrderNum</u>, <u>PartNum</u>, NumOrdered, QuotedPrice)

Orders

OrderNum	OrderDate	PartNum	Description	NumOrdered	QuotedPrice
21608	10/20/2010	AT94	Iron	11	$21.95
21610	10/20/2010	DR93	Gas Range	1	$495.00
21610	10/20/2010	DW11	Washer	1	$399.99
21613	10/21/2010	KL62	Dryer	4	$329.95
21614	10/21/2010	KT03	Dishwasher	2	$595.00
21617	10/23/2010	BV06	Home Gym	2	$794.95
21617	10/23/2010	CD52	Microwave Oven	4	$150.00
21619	10/23/2010	DR93	Gas Range	1	$495.00
21623	10/23/2010	KV29	Treadmill	2	$1290.00

Orders

OrderNum	OrderDate
21608	10/20/2010
21610	10/20/2010
21613	10/21/2010
21614	10/21/2010
21617	10/23/2010
21619	10/23/2010
21623	10/23/2010

Part

PartNum	Description
AT94	Iron
BV06	Home Gym
CD52	Microwave Oven
DL71	Cordless Drill
DR93	Gas Range
DW11	Washer
FD21	Stand Mixer
KL62	Dryer
KT03	Dishwasher
KV29	Treadmill

OrderLine

OrderNum	PartNum	NumOrdered	QuotedPrice
21608	AT94	11	$21.95
21610	DR93	1	$495.00
21610	DW11	1	$399.99
21613	KL62	4	$329.95
21614	KT03	2	$595.00
21617	BV06	2	$794.95
21617	CD52	4	$150.00
21619	DR93	1	$495.00
21623	KV29	2	$1290.00

Third Normal Form (3NF)

This form dictates that all **non-key** attributes of a table must be functionally dependent on a candidate key i.e. there can be no interdependencies among non-key attributes.

For a table to be in 3NF, there are two requirements

- The table should be second normal form
- No attribute is transitively dependent (TD) on the primary key

Note: 3-NF eliminates both partial and transitive dependencies and hence the redundancies caused by them would get eliminated.

A transitive dependency occurs **when one non-prime attribute is dependent on another non-prime attribute**.

Example (Not in 3NF)

Scheme → {Title, PubID, PageCount, Price }

1. **Primary key → {Title, PubID}**
2. **{Title, PubID} → {PageCount}**
3. **{PageCount} → {Price}**
4. **Both Price and PageCount depend on a key hence 2NF**
5. **Transitively {Title, PubID} → {Price} hence not in 3NF**

Read more here:
https://www.w3schools.in/dbms/database-normalization
https://www.guru99.com/database-normalization.html
https://www.javatpoint.com/dbms-normalization

Third Normal Form (3NF)

Example 2 (Not in 3NF)

Scheme → {Studio, StudioCity, CityTemp}

1. **Primary key → {Studio}**
2. **{Studio} → {StudioCity}**
3. **{StudioCity} → {CityTemp}**
4. **{Studio} → {CityTemp}**
5. **Both StudioCity and CityTemp depend on the entire key hence 2NF**
6. **CityTemp transitively depends on Studio hence violates 3NF**

Example 3 (Not in 3NF)

Scheme → {BuildingID, Contractor, Fee}

1. **Primary key → {BuildingID}**
2. **{BuildingID} → {Contractor}**
3. **{Contractor} → {Fee}**
4. **{BuildingID} → {Fee}**
5. **Both Contractor and Fee depend on the entire key hence 2NF**
6. **Fee transitively depends on the BuildingID**

BuildingID	Contractor	Fee
100	Randolph	1200
150	Ingersoll	1100
200	Randolph	1200
250	Pitkin	1100
300	Randolph	1200

3NF - Decomposition

1. Move all items involved in transitive dependencies to a new table.
2. Identify a primary key for the new table.
3. Place the primary key for the new table as a foreign key on the original table.

Example (Not in 3NF)

Scheme → {<u>Title, PubID</u>, PageCount, Price }

1. **Primary key → {Title, PubID}**
2. **{Title, PubID} → {PageCount}**
3. **{PageCount} → {Price}**
4. **Both Price and PageCount depend on a key hence 2NF**
5. **Transitively {Title, PubID} → {Price} hence not in 3NF**

Convert to 3NF

Old Scheme → {<u>Title</u>, <u>PubID</u>, PageCount, Price }

New Scheme → {<u>Title</u>, <u>PubID</u>, PageCount}

New Scheme → {<u>PageCount</u>, Price}

Example (Not in 3NF)

Scheme → {Studio, StudioCity, CityTemp}

1. **Primary key → {Studio}**
2. **{Studio} → {StudioCity}**
3. **{StudioCity} → {CityTemp}**
4. **{Studio} → {CityTemp}**
5. **Both StudioCity and CityTemp depend on the entire key hence 2NF**
6. **CityTemp transitively depends on Studio hence violates 3NF**

Convert to 3NF

Old Scheme → {Studio, StudioCity, CityTemp}

New Scheme → {Studio, StudioCity}

New Scheme → {StudioCity, CityTemp}

Example (Not in 3NF)

Scheme → {BuildingID, Contractor, Fee}

1. **Primary key → {BuildingID}**
2. **{BuildingID} → {Contractor}**
3. **{Contractor} → {Fee}**
4. **{BuildingID} → {Fee}**
5. **Both Contractor and Fee depend on the entire key hence 2NF**
6. **Fee transitively depends on the BuildingID**

BuildingID	Contractor	Fee
100	Randolph	1200
150	Ingersoll	1100
200	Randolph	1200
250	Pitkin	1100
300	Randolph	1200

Convert to 3NF

Old Scheme → {<u>BuildingID</u>, Contractor, Fee}

New Scheme → {<u>BuildingID</u>, Contractor}

New Scheme → {<u>Contractor</u>, Fee}

BuildingID	Contractor
100	Randolph
150	Ingersoll
200	Randolph
250	Pitkin
300	Randolph

Contractor	Fee
Randolph	1200
Ingersoll	1100
Pitkin	1100

Example – 3NF

Advisor Scheme: {<u>Student#</u>, Advisor, Adv-Room}

Student#	Advisor	Adv-Room
1022	Jones	412
4123	Smith	216
3217	Mark	148
3217	Helen	151

Scheme → {Student#, Advisor, Adv-Room}

1. **Primary key → {Student#}**
2. **{Student#} → {Adv-Room}**
3. **{Student#} → {Advisor}**
4. **{Advisor} → {Adv-Room}**
5. **Both Advisor and Adv-Room depend on the entire key hence 2NF**
6. **Adv-Room transitively depends on the Student#**

Convert to 3NF

Advisor Scheme: {<u>Student#</u>, Advisor}

Student#	Advisor
1022	Jones
4123	Smith
3217	Mark
3217	Helen

Faculty Scheme: {<u>Advisor</u>, Adv-Room, Dep#}

Advisor	Adv-Room	Dep#
Jones	412	35
Smith	216	22
Mark	148	20
Helen	151	20

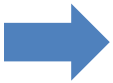
Example – 3NF

For the Customer table, you would remove LastName and FirstName because they depend on the determinant RepNum, which is not a Primary key.

A new table is formed, consisting of RepNum (as primary key), LastName, and FirstName:
Customer (CustomerNum, CustomerName, Balance, CreditLimit, RepNum)
Rep (RepNum, LastName, FirstName)

Customer

CustomerNum	CustomerName	Balance	CreditLimit	RepNum	LastName	FirstName
148	Al's Appliance and Sport	$6,550.00	$7,500.00	20	Kaiser	Valerie
282	Brookings Direct	$431.50	$10,000.00	35	Hull	Richard
356	Ferguson's	$5,785.00	$7,500.00	65	Perez	Juan
408	The Everything Shop	$5,285.25	$5,000.00	35	Hull	Richard
462	Bargains Galore	$3,412.00	$10,000.00	65	Perez	Juan
524	Kline's	$12,762.00	$15,000.00	20	Kaiser	Valerie
608	Johnson's Department Store	$2,106.00	$10,000.00	65	Perez	Juan
687	Lee's Sport and Appliance	$2,851.00	$5,000.00	35	Hull	Richard
725	Deerfield's Four Seasons	$248.00	$7,500.00	35	Hull	Richard
842	All Season	$8,221.00	$7,500.00	20	Kaiser	Valerie



Customer

CustomerNum	CustomerName	Balance	CreditLimit	RepNum
148	Al's Appliance and Sport	$6,550.00	$7,500.00	20
282	Brookings Direct	$431.50	$10,000.00	35
356	Ferguson's	$5,785.00	$7,500.00	65
408	The Everything Shop	$5,285.25	$5,000.00	35
462	Bargains Galore	$3,412.00	$10,000.00	65
524	Kline's	$12,762.00	$15,000.00	20
608	Johnson's Department Store	$2,106.00	$10,000.00	65
687	Lee's Sport and Appliance	$2,851.00	$5,000.00	35
725	Deerfield's Four Seasons	$248.00	$7,500.00	35
842	All Season	$8,221.00	$7,500.00	20

Rep

RepNum	LastName	FirstName
20	Kaiser	Valerie
35	Hull	Richard
65	Perez	Juan

Normalization

Which field from the previous example would be a candidate to be moved to another table to normalise this database?

Procedures

ProcedureID	PatientID	ProcedureDate	ProcedureType	ProcedureOperator	ProcedureData
1	101	10-Jul-17	ECG	Nurse1	120 BPM
2	102	09-Jun-17	SPIRO	Operator1	4 L
3	103	08-Mar-17	BLOOD	Nurse2	1.1 Neutrophils
4	102	23-May-17	ECG	Operator2	73 BPM

Patients

MedRecID	FirstName	Surname
101	Silas	Farquard
102	Ella	McFlame
103	Thomas	Tankengine

Summary

- 1NF - Eliminate duplicative columns from the same table, create separate tables for
- each group of related data and identify each row with a unique column (primary key)
- 2NF – Does each non-key column depend on the entire key? Remove subsets of data
- that apply to multiple rows of a table and place them in separate tables. Create
- relationships between these new tables and their predecessors through the use of
- foreign keys.
- 3NF - Does each column depend on the entire key and nothing but? Remove columns
- that are not fully dependent upon the primary key
- Verify that the tables can be reconnected (joined) to form the original tables
- ALWAYS enter sample data - look for replication and normaliZe.

Summary – Lecture Learning Outcomes

- Understand relational database design, concepts of keys and referential integrity
- Define 1st, 2nd and 3rd normal forms
- Be able to design a normalised database that has no duplicate information nor
- replication anomalies
- Be willing to put in the hours to have a “straight and level” flight path throughout this course
- Be able to recover a spinning plane hurtling towards the ground

Thanks