Data
දත්ත

Unprocessed raw material that cannot be meaningful when taken separately is known as data.
වෙන් වෙන් වශයෙන් ගත් විට අර්ථයක් දීමට නොහැකි නිසි ලෙස සකස් කර නොමැති අමු කරුණු දත්ත ලෙස හඳුන්වයි.

Structured data
ව්‍යුහගත දත්ත

Structured data is organized and follows a specific format.
ව්‍යුහගත දත්ත සංවිධානය කර ඇති අතර නිශ්චිත ආකෘතියක් අනුගමනය කරයි.

Ex. Databases, spreadsheets
දත්ත සමුදායන්, පැතුරුම්පත්

Unstructured data
ව්‍යුහගත නොකළ දත්ත

Unstructured data lacks a specific format or organization.
ව්‍යුහගත නොකළ දත්තවල නිශ්චිත ආකෘතියක් හෝ සංවිධානයක් නොමැත.

Ex. Text documents, images, videos
පෙළ ලේඛන, රූප, වීඩියෝ

Structured Data	Unstructured Data
Organized	Chaotic
Tabular	Varied
Database	Images
Easy	Complex
Fixed	Flexible
Queryable	Exploratory

Information
තොරතුරු

Data that has been processed, organized, or structured to provide meaning or context.
අර්ථයක් හෝ සන්දර්භයක් සැපයීම සඳහා සකසන ලද, සංවිධානය කරන ලද හෝ ව්‍යුහගත කර ඇති දත්ත වේ.

Note
There is no inherent distinction between data and information; rather, it depends on one's perspective. Something is considered data when it is viewed as raw facts or figures, while it becomes information when it is interpreted and given context.
දත්ත සහ තොරතුරු අතර ආවේණික වෙනසක් නොමැත; ඒ වෙනුවට, එය කෙනෙකුගේ දැක්ම මත රඳා පවතී. යමක් දෙස අමු කරුණු හෝ සංඛ්‍යා ලෙස බලන විට දත්ත ලෙස සලකනු ලැබිය හැකි අතර, එයට අර්ථකථනය කර සන්දර්භයක් ලබා දුන් විට එය තොරතුරු බවට පත් වේ.

Structured Information
ව්‍යුහගත තොරතුරු

This type of information is well-organized and follows a predefined format or schema.
මෙම වර්ගයේ තොරතුරු හොඳින් සංවිධානය වී ඇති අතර පූර්ව නිශ්චිත ආකෘතියක් හෝ යෝජනා ක්‍රමයක් අනුගමනය කරයි.

Ex. Databases, spreadsheets, standardized forms
දත්ත සමුදායන්, පැතුරුම්පත්, ප්‍රමිතිගත ආකෘති

Unstructured Information
ව්‍යුහගත නොකළ තොරතුරු

Unstructured information lacks a specific format or organization.
ව්‍යුහගත නොවන තොරතුරුවලට නිශ්චිත ආකෘතියක් හෝ සංවිධානයක් නොමැත.

Ex. Free-form text, emails, multimedia files, social media posts.
නිදහස් ආකෘති පෙළ, ඊමේල්, බහුමාධ්‍ය ගොනු, සමාජ මාධ්‍ය

Data	Information
Raw, unprocessed values or facts. අමු, සකස් නොකළ අගයන් හෝ කරුණු.	Processed, analyzed, and interpreted data. සකසන ලද, විශ්ලේෂණය කරන ලද සහ අර්ථකථනය කරන ලද දත්ත.
Individual observations or figures. තනි නිරීක්ෂණ හෝ සංඛ්‍යා.	Insights, trends, or conclusions derived from data. දත්ත වලින් ලබාගත් තීක්ෂ්ණ බුද්ධිය, ප්‍රවණතා හෝ නිගමන.
Numeric, alphanumeric, text, or multimedia. සංඛ්‍යාත්මක, අක්ෂරාංක, පෙළ, හෝ බහුමාධ්‍ය.	Summarized, contextualized, and structured. සාරාංශගත, සන්දර්භගත සහ ව්‍යුහගත.
Unstructured: Lacks specific format or organization. ව්‍යුහගත නොකළ: නිශ්චිත ආකෘතියක් හෝ සංවිධානයක් නොමැත.	Structured: Organized in a predefined format. ව්‍යුහගත: පූර්ව නිශ්චිත ආකෘතියකින් සංවිධානය කර ඇත.
Lack context or interpretation. සන්දර්භය හෝ අර්ථ නිරූපණය නොමැතිකම.	Provides context and meaning to data. දත්ත සඳහා සන්දර්භය සහ අර්ථය සපයයි.
Foundation for analysis and decision-making. විශ්ලේෂණය සහ තීරණ ගැනීම සඳහා පදනම.	Supports decision-making, understanding, or communication. තීරණ ගැනීම, අවබෝධය හෝ සන්නිවේදනය සඳහා සහාය වේ.
Requires processing to be valuable. වැදගත් ලෙස සැකසීමට අවශ්‍ය වේ.	Valuable for decision-making, problem-solving, or insights. තීරණ ගැනීම, ගැටළු විසඳීම හෝ තීක්ෂ්ණ බුද්ධිය සඳහා වැදගත් ය.
May require organization or transformation for usability. උපයෝගීතාවය සඳහා සංවිධානයක් හෝ පරිවර්තනයක් අවශ්‍ය විය හැකිය.	Often presented in accessible formats for ease of understanding. බොහෝ විට අවබෝධ කර ගැනීමේ පහසුව සඳහා ප්‍රවේශ විය හැකි ආකෘතිවලින් ඉදිරිපත් කෙරේ.
Sales figures, quantities, product codes. විකුණුම් සංඛ්‍යා, ප්‍රමාණ, නිෂ්පාදන කේත.	Total sales for a period, sales trends, best-selling products. කාල සීමාවක් සඳහා සම්පූර්ණ විකුණුම්, විකුණුම් ප්‍රවණතා, වැඩියෙන්ම අලෙවි වන නිෂ්පාදන.

Need for transformation data (unstructured) into information (structured)

දත්ත (ව්‍යුහගත නොවන) තොරතුරු (ව්‍යුහගත) බවට පරිවර්තනය කිරීමේ අවශ්‍යතාවය

Enhanced Decision-Making

Structured data is easier to analyze, enabling organizations to make informed decisions quickly based on clear insights.

ව්‍යුහගත දත්ත විශ්ලේෂණය කිරීමට පහසු වන අතර, පැහැදිලි තීක්ෂ්ණ බුද්ධිය මත පදනම්ව ඉක්මනින් දැනුවත් තීරණ ගැනීමට ආයතනවලට හැකි වේ.

Data Integration

Combining unstructured data from various sources (like social media, emails, and documents) with structured data (like databases) provides a comprehensive view of the information landscape.

විවිධ මූලාශ්‍රවලින් (සමාජ මාධ්‍ය, ඊමේල් සහ ලේඛන වැනි) ලැබෙන ව්‍යුහගත නොකළ දත්ත, ව්‍යුහගත දත්ත (දත්ත සමුදායන් වැනි) සමග ඒකාබද්ධ කිරීම තොරතුරු දර්ශනය පිළිබඳ පුළුල් දැක්මක් සපයයි.

Efficiency
Structured data can be easily queried and processed, leading to more efficient data management and retrieval.
ව්‍යුහගත දත්ත පහසුවෙන් විමසීමට සහ සැකසීමට හැකි වන අතර, එය වඩාත් කාර්යක්ෂම දත්ත කළමනාකරණයට සහ නැවත ලබා ගැනීමට මග පාදයි.

Improved Analytics
Analytical tools work better with structured data, allowing for advanced analytics, predictive modeling, and business intelligence applications.
විශ්ලේෂණ මෙවලම් ව්‍යුහගත දත්ත සමඟ වඩා හොඳින් ක්‍රියා කරයි, උසස් විශ්ලේෂණ, අනාවැකි ආකෘතිකරණය සහ ව්‍යාපාරික බුද්ධි යෙදුම් සඳහා ඉඩ සලසයි.

Compliance and Reporting
Structured information is easier to standardize, making it simpler to meet regulatory requirements and generate reports.
ව්‍යුහගත තොරතුරු ප්‍රමිතිකරණය කිරීමට පහසු වන අතර, නියාමන අවශ්‍යතා සපුරාලීම සහ වාර්තා උත්පාදනය කිරීම පහසු කරයි.

Customer Insights
Transforming customer feedback from unstructured sources can help businesses better understand customer needs and preferences, leading to improved products and services.
ව්‍යුහගත නොවන මූලාශ්‍රවලින් ලැබෙන පාරිභෝගික ප්‍රතිචාර පරිවර්තනය කිරීම මඟින් ව්‍යාපාරවලට පාරිභෝගික අවශ්‍යතා සහ මනාපයන් වඩාත් හොඳින් අවබෝධ කර ගැනීමට උදවු වන අතර, වැඩිදියුණු කළ නිෂ්පාදන සහ සේවා සඳහා මඟ පෙන්වයි.

Cost Reduction
Efficient data management reduces storage costs and improves resource allocation, as structured data requires less processing power for analysis.
ව්‍යුහගත දත්ත විශ්ලේෂණය සඳහා අඩු සැකසුම් බලයක් අවශ්‍ය වන බැවින් කාර්යක්ෂම දත්ත කළමනාකරණය ගබඩා පිරිවැය අඩු කරන අතර සම්පත් වෙන් කිරීම වැඩි දියුණු කරයි.

Knowledge Extraction
Turning unstructured data into structured formats enables the extraction of valuable insights that can drive innovation and strategic planning.
ව්‍යුහගත නොවන දත්ත ව්‍යුහගත ආකෘති බවට පත් කිරීමෙන් නවෝත්පාදන සහ උපාය මාර්ගික සැලසුම් මෙහෙයවිය හැකි වටිනා තීක්ෂ්ණ බුද්ධියක් උකහා ගැනීමට හැකියාව ලැබේ.

Categorization of data into group (tables)
දත්ත කාණ්ඩවලට(වගු) වර්ගීකරණය

Defining Categories
Identifies relevant criteria or attributes (e.g., age, location, product type) to group the data.
දත්ත කාණ්ඩ කිරීමට අදාළ නිර්ණායක හෝ ගුණාංග (උදා. වයස, ස්ථානය, නිෂ්පාදන වර්ගය) හඳුනා ගනී.

Data Segmentation
Sorts the data based on the defined categories, creating distinct groups.
වෙනස් කාණ්ඩ නිර්මාණය කරමින් අර්ථ දක්වා ඇති කාණ්ඩ මත පදනම්ව දත්ත වර්ග කරයි.

Tabular Format
Presents the categorized data in tables, where each row represents a data entry and each column represents a category or attribute.
එක් එක් පේළිය දත්ත ඇතුළත් කිරීමක් නියෝජනය කරන අතර සෑම තීරුවක්ම ප්‍රවර්ගයක් හෝ ගුණාංගයක් නියෝජනය කරන වගු තුළ වර්ගීකරණය කළ දත්ත ඉදිරිපත් කරයි.

Analysis
Uses the tables for various analyses, such as identifying trends, making comparisons, or summarizing information.
ප්‍රවණතා හඳුනා ගැනීම, සැසඳීම් කිරීම හෝ තොරතුරු සාරාංශ කිරීම වැනි විවිධ විශ්ලේෂණ සඳහා වගු භාවිතා කරයි.

Databases
දත්ත සමුදායන්

A database is an organized collection of structured information, or data, typically stored electronically or manually.
දත්ත සමුදායක් යනු සාමාන්‍යයෙන් ඉලෙක්ට්‍රොනිකව හෝ හස්තීය ලෙස ගබඩා කර ඇති ව්‍යුහගත තොරතුරු හෝ දත්තවල සංවිධානාත්මක එකතුවකි.

They facilitate efficient storage, retrieval, and management.
ඒවා මගින් කාර්යක්ෂමව ගබඩා කිරීම, ලබා ගැනීම සහ කළමනාකරණය සඳහා පහසුකම් සපයයි.

Databases enforce rules and constraints that help maintain data accuracy and consistency.
දත්ත සමුදායන් දත්ත නිරවද්‍යතාවය සහ අනුකූලතාව පවත්වා ගැනීමට උපකාරී වන නීති සහ සම්බාධක බලාත්මක කරයි.

Databases enable easy generation of reports and analytics, helping organizations make informed decisions based on data trends.
දත්ත සමුදායන් පහසුවෙන් වාර්තා සහ විශ්ලේෂණ උත්පාදනය සක්‍රීය කරයි, දත්ත ප්‍රවණතා මත පදනම්ව දැනුවත් තීරණ ගැනීමට ආයතනවලට උදවු කරයි

Aspect	**With a Database (Manual or Electronic)**	**Without a Database**
Organization සංවිධානය	Systematic and structured ක්‍රමානුකූල සහ ව්‍යුහගත	Scattered and unstructured විසිරුණු සහ ව්‍යුහගත නොවන
Efficiency කාර්යක්ෂමතාව	Faster data management වේගවත් දත්ත කළමනාකරණය	Time-consuming searches කාලය ගතවන සෙවීම්
Data Integrity දත්ත අඛණ්ඩතාව	Consistent and accurate ස්ථාවර සහ නිවැරදි	Higher error rates ඉහළ දෝෂ අනුපාත
Scalability පරිමාණය	Adapts to growth වර්ධනයට අනුගත වේ	Difficult to manage increasing data වැඩිවන දත්ත කළමනාකරණය කිරීමට අපහසුය
Accessibility ප්‍රවේශ්‍යතාව	Quick access for users පරිශීලකයින් සඳහා ඉක්මන් ප්‍රවේශය	Cumbersome retrieval අපහසු නැවත ලබා ගැනීම
Security ආරක්ෂාව	Data protection and access controls දත්ත ආරක්ෂාව සහ ප්‍රවේශ පාලනය	Vulnerable to unauthorized access අනවසර පිවිසුමට ගොදුරු විය හැක
Backup උපස්ථ	Easier backup and recovery පහසු උපස්ථ සහ ප්‍රතිසාධනය	Higher risk of data loss දත්ත නැතිවීමේ වැඩි අවදානම
Cost පිරිවැය	Higher initial costs, but long-term savings ඉහළ ආරම්භක පිරිවැය, නමුත් දිගුකාලීන ඉතිරිකිරීම්	Lower initial costs, higher long-term expenses අඩු ආරම්භක පිරිවැය, වැඩි දිගු කාලීන වියදම්
Maintenance නඩත්තු කිරීම	Ongoing maintenance required අඛණ්ඩ නඩත්තු කිරීම අවශ්‍ය වේ	Minimal but risky disorganization අවම නමුත් අවදානම් අසංවිධානාත්මකයි
Adaptability අනුවර්තනය වීමේ හැකියාව	Easier to implement changes වෙනස්කම් ක්‍රියාත්මක කිරීම පහසුය	Hard to adapt practices පුරුදු වලට අනුගත වීමට අපහසුය

Types of databases
දත්ත සමුදා වර්ග

Manual Databases **හස්තමය දත්ත සමුදා**	**Automated Databases** **ස්වයංක්‍රීය දත්ත සමුදායන්**
Requires manual input, prone to human error. හස්තීය ආදාන අවශ්‍ය වේ, මිනිස් දෝෂ වලට ගොදුරු වේ.	Automated input reduces errors. ස්වයංක්‍රීය ආදානය දෝෂ අඩු කරයි.
Access is often limited to physical documents. ප්‍රවේශ බොහෝ විට භෞතික ලේඛනවලට සීමා වේ.	Quick access through software and queries. මෘදුකාංග සහ විමසුම් හරහා ඉක්මන් ප්‍රවේශය.
Time-consuming to search and update. සෙවීමට සහ යාවත්කාලීන කිරීමට කාලය ගත වේ.	Fast retrieval and updates. වේගවත් ලබා ගැනීම සහ යාවත්කාලීන කිරීම්.
Difficult to scale with increasing data volume. දත්ත පරිමාව වැඩි කිරීමත් සමඟ පරිමාණය කිරීමට අපහසුය.	Easily scalable to handle large datasets. විශාල දත්ත කට්ටල හැසිරවීමට පහසුවෙන් පරිමාණය කළ හැකිය.
Higher risk of inconsistencies. නොගැලපීම් වැඩි අවදානමක්.	Enforced rules maintain consistency. බලාත්මක කරන ලද නීති අනුකූලතාව පවත්වා ගනී.
Manual backups increase the risk of data loss. හස්ත උපස්ථ කිරීම දත්ත නැතිවීමේ අවදානම වැඩි කරයි.	Automated backups enhance data security. ස්වයංක්‍රීය උපස්ථ දත්ත ආරක්ෂාව වැඩි දියුණු කරයි.
Limited to single-user access. තනි පරිශීලක ප්‍රවේශයට සීමා වේ.	Supports multiple users concurrently. එකවර බහු පරිශීලකයින්ට සහය දක්වයි.
Reporting is manual and time-consuming. වාර්තා කිරීම හස්තීය සහ කාලය ගත වේ.	Automated reporting and analytics. ස්වයංක්‍රීය වාර්තාකරණය සහ විශ්ලේෂණ.
Lower initial cost, but higher long-term costs due to inefficiencies. අඩු ආරම්භක පිරිවැය, නමුත් අකාර්යක්ෂමතාව හේතුවෙන් දිගු කාලීන පිරිවැය වැඩි වේ.	Higher initial investment, but cost-effective long-term. ඉහළ ආරම්භක ආයෝජනය, නමුත් දිගු කාලීනව ලාභදායී වේ.
Minimal maintenance, but labor-intensive. අවම නඩත්තු, නමුත් ශ්‍රමදැඩිය.	Requires ongoing software updates and maintenance. අඛණ්ඩ මෘදුකාංග යාවත්කාලීන කිරීම් සහ නඩත්තු කිරීම අවශ්‍ය වේ.
Hard to implement changes. වෙනස්කම් ක්‍රියාත්මක කිරීමට අපහසුය.	Easier to adapt and modify data structures. දත්ත ව්‍යුහයන් අනුවර්තනය කිරීමට සහ වෙනස් කිරීමට පහසුය.

Database models
දත්ත සමුදා ආකෘති

1. Flat File Systems
පැතලි ගොනු පද්ධති

A flat file system is a simple method of storing data in a single file, typically as plain text or in a spreadsheet format.
පැතලි ගොනු පද්ධතියක් යනු සාමාන්‍යයෙන් සරල පෙළ ලෙස හෝ පැතුරුම්පත් ආකෘතියකින් තනි ගොනුවක දත්ත ගබඩා කිරීමේ සරල ක්‍රමයකි.

Structure

- Single Layer
  Data is stored in one single table without any hierarchical or relational structure.
  කිසිදු ධූරාවලි හෝ සම්බන්ධතා ව්‍යුහයකින් තොරව දත්ත එක් වගුවක ගබඩා කර ඇත.

- Record Format
  Each record is usually a line of text, and fields within records are often separated by delimiters (Ex. commas, tabs).
  සෑම වාර්තාවක්ම සාමාන්‍යයෙන් එක් පෙළක් වන අතර, වාර්තා තුළ ඇති ක්ෂේත්‍ර බොහෝ විට පරිසීමක මගින් වෙන් කරනු ලැබේ (උදා. කොමා, ටැබ්).

Characteristics
Easy to create and manage, often using basic text editors or spreadsheet applications.
බොහෝ විට මූලික පෙළ සංස්කාරක හෝ පැතුරුම්පත් යෙදුම් භාවිතා කරමින් නිර්මාණය කිරීමට සහ කළමනාකරණය කිරීමට පහසුය.

Lacks the capability to represent relationships between different data entities.
විවිධ දත්ත භූතාර්ත අතර සම්බන්ධතා නියෝජනය කිරීමේ හැකියාවක් නොමැත.

Usage
Suitable for small applications with minimal complexity, such as simple lists (e.g., contact lists).
සරල ලැයිස්තු (උදා: සම්බන්ධතා ලැයිස්තු) වැනි අවම සංකීර්ණතා සහිත කුඩා යෙදුම් සඳහා සුදුසු වේ.

Often used for data exchange between systems due to its simplicity and compatibility with various formats.
පද්ධති අතර දත්ත හුවමාරුව සඳහා බොහෝ විට එය භාවිතා කරනුයේ එහි සරල බව සහ විවිධ හැඩතල සමඟ අනුකූල වීම හේතුවෙනි.

Advantages
Simple structure makes it easy to read and interpret.
සරල ව්‍යුහය කියවීමට සහ අර්ථ නිරූපණය කිරීම පහසු කරයි.

Minimal software requirements; can be managed with basic tools.
අවම මෘදුකාංග අවශ්‍යතා; මූලික මෙවලම් සමඟ කළමනාකරණය කළ හැකිය.

Disadvantages
As data grows, it becomes cumbersome to manage.
දත්ත වර්ධනය වන විට, එය කළමනාකරණය කිරීමට අපහසු වේ.

No normalization leads to duplication of data.
ප්‍රමතකරණයක් නැති වීම දත්ත අනුපිටපත් කිරීමට මග පාදයි.

Searching and updating data can be slow and inefficient, especially with large files.
විශේෂයෙන් විශාල ගොනු සමඟ දත්ත සෙවීම සහ යාවත්කාලීන කිරීම මන්දගාමී සහ අකාර්යක්ෂම විය හැක.

Lacks constraints to ensure data validity, which can lead to inconsistencies.
දත්ත වලංගුභාවය සහතික කිරීම සඳහා සීමාවන් නොමැති අතර, එමගින් නොගැලපීම් ඇති විය හැක.

Examples of flat file systems
පැතලි ගොනු පද්ධති සඳහා උදාහරණ

CSV Files (Comma-Separated Values)

A text file where each line represents a record, and fields are separated by commas.
එක් එක් පේළිය මගින් record එකක් නියෝජනය කරන සහ ක්ෂේත්‍ර කොමාවකින් වෙන් කරනු ලබන පෙළ ගොනුවකි.

Ex.

```
Name, Age, Email
John Doe, 28, john@example.com
Jane Smith, 34, jane@example.com
```

TXT Files (Plain Text Files)

Simple text files that can store any form of text data.
ඕනෑම ආකාරයක පෙළ දත්ත ගබඩා කළ හැකි සරල පෙළ ගොනු.

Ex.

```
Task 1: Review documents
Task 2: Prepare presentation
Task 3: Attend meeting
```

Excel Files (XLS/XLSX)

While Excel supports more complex data structures, individual sheets can be considered flat files when they contain simple tables.
Excel වඩාත් සංකීර්ණ දත්ත ව්‍යුහයන් සඳහා සහය දක්වන අතර, සරල වගු අඩංගු වන විට තනි පත්‍ර පැතලි ගොනු ලෙස සැලකිය හැක.

JSON Files (JavaScript Object Notation)

A text format that is easy to read and write for humans and machines, often used to represent simple data structures.
මිනිසුන්ට සහ යන්ත්‍ර සඳහා කියවීමට සහ ලිවීමට පහසු පෙළ ආකෘතියකි, බොහෝ විට සරල දත්ත ව්‍යුහයන් නියෝජනය කිරීමට භාවිතා කරයි.

Ex.

```
[
  {
      "name": "John Doe",
      "age": 28,
      "email":"john@example.com"
   },
   {
      "name": "Jane Smith",
      "age": 34,
      "email": "jane@example.com"
    }
]
```

Log Files

Text files that record events or activities within software applications or systems.
මෘදුකාංග යෙදුම් හෝ පද්ධති තුළ සිදුවීම් හෝ ක්‍රියාකාරකම් වාර්තා කරන පෙළ ගොනු.

Ex.

```
2024-10-01 10:00:00 - User logged in
2024-10-01 10:05:00 - File uploaded
```

ictfromabc.com

2. Hierarchical Model
ධුරාවලි ආකෘතිය

The hierarchical database model is a data structure that organizes data in a tree-like format, where each record has a single parent node and can have multiple child nodes.
ධුරාවලි දත්ත සමුදා ආකෘතිය යනු ගසක වැනි ආකෘතියකින් දත්ත සංවිධානය කරන දත්ත ව්‍යුහයකි, එහිදී සෑම වාර්තාවකටම parent node එකක් සිටින අතර child nodes කිහිපයක් සිටිය හැක.

```
Company
├── Sales Department
│   ├── Sales Manager
│   ├── Sales Representative 1
│   └── Sales Representative 2
├── IT Department
│   ├── IT Manager
│   └── Support Technician
└── HR Department
    ├── HR Manager
    └── Recruiter
```

Tree Structure
Data is represented as a hierarchy with a single root node at the top and branches representing child node records.
දත්ත ඉහලින් තනි මූල නෝඩයක් සහ child node records නියෝජනය කරන අතු සහිත ධුරාවලියක් ලෙස නිරූපණය කෙරේ.

Parent-Child Relationships
Each child node record is linked to only one parent node, forming a strict hierarchy.
සෑම child node record එකකටම දැඩි ධුරාවලියක් සාදමින් එක් parent node එකකට පමණක් සම්බන්ධ කර ඇත.

Navigational Access
Data retrieval follows a path from the root to the desired record.
දත්ත ලබාගැනීම මූලයේ සිට අපේක්ෂිත වාර්තාව දක්වා මාර්ගයක් අනුගමනය කරයි.

Characteristics

Each parent node can have multiple child nodes, but each child node can have only one parent node.
සෑම parent node එකකටම child nodes කිහිපයක් තිබිය හැකි නමුත් සෑම child node එකකටම තිබිය හැක්කේ එක් parent node එකක් පමණි.

The model inherently enforces data integrity through its structured relationships.
ආකෘතිය සහජයෙන්ම එහි ව්‍යුහගත සම්බන්ධතා හරහා දත්ත අඛණ්ඩතාව බලාත්මක කරයි.

The hierarchy must be defined at the outset, making it challenging to modify later.
ධුරාවලිය ආරම්භයේදීම නිර්වචනය කළ යුතු අතර, එය පසුව වෙනස් කිරීමට අපහසු වෙයි.

Advantages
Easy to understand and visualize due to its clear structure.
එහි පැහැදිලි ව්‍යුහය නිසා තේරුම් ගැනීමට සහ දෘශ්‍යමාන කිරීමට පහසුය.

Relationships help maintain consistent and valid data.
සබඳතා ස්ථාවර සහ වලංගු දත්ත පවත්වා ගැනීමට උපකාරී වේ.

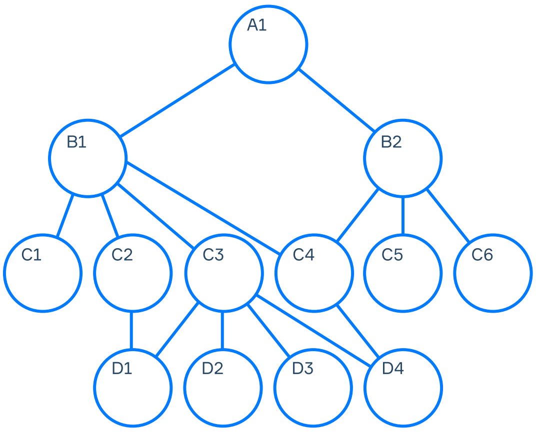
Disadvantages
Difficult to modify the structure once it’s established; adding new relationships can be complex.
ව්‍යුහය ස්ථාපිත කළ පසු එය වෙනස් කිරීමට අපහසුය; නව සබඳතා එකතු කිරීම සංකීර්ණ විය හැකිය.

Not well-suited for handling many-to-many relationships.
බහු බහු සබඳතා හැසිරවීමට සුදුසු නොවේ.

Managing a large hierarchy can become complicated.
විශාල ධුරාවලියක් කළමනාකරණය කිරීම සංකීර්ණ විය හැකිය.

3. Network Model
ජාල ආකෘතිය

The network database model is a data structure that allows multiple relationships between records, forming a graph-like structure.
ජාල දත්ත සමුදා ආකෘතිය යනු ප්‍රස්ථාර වැනි ව්‍යුහයක් සාදමින් වාර්තා අතර බහු සම්බන්ධතා ඇති කිරීමට ඉඩ සලසන දත්ත ව්‍යුහයකි.



Characteristics

Unlike the hierarchical model, which restricts relationships to one-to-many, the network model supports many-to-many relationships.
සම්බන්ධතා ඒක බහු වලට සීමා කරන ධූරාවලි ආකෘතිය මෙන් නොව, ජාල ආකෘතිය මගින් බහු බහු සබඳතා සඳහා සහය දක්වයි.

Relationships are maintained through pointers, ensuring that connections between records are valid.
වාර්තා අතර සම්බන්ධතා වලංගු බව සහතික කරමින් පොයින්ටර් හරහා සබඳතා පවත්වාගෙන යයි.

Requires a more sophisticated design and understanding of data relationships.
දත්ත සම්බන්ධතා පිළිබඳ වඩාත් සංකීර්ණ සැලසුමක් සහ අවබෝධයක් අවශ්‍ය වේ.

Advantages

Allows for a more dynamic representation of relationships between records.
වාර්තා අතර සම්බන්ධතා වඩාත් ගතික නිරූපණය සඳහා ඉඩ ලබා දේ.

Can access interconnected data quickly through direct links.
සෘජු සබැඳි හරහා අන්තර් සම්බන්ධිත දත්ත ඉක්මනින් ලබා ගත හැක.

Suitable for representing real-world scenarios with complex relationships.
සංකීර්ණ සම්බන්ධතා සහිත සැබෑ ලෝකයේ අවස්ථා නිරූපණය කිරීම සඳහා සුදුසු වේ.

Disadvantages

More challenging to design and manage compared to simpler models; requires a deep understanding of the relationships.
සරල මාදිලිවලට සාපේක්ෂව සැලසුම් කිරීම සහ කළමනාකරණය කිරීම වඩාත් අභියෝගාත්මක ය; සබඳතා පිළිබඳ ගැඹුරු අවබෝධයක් අවශ්‍ය වේ.

Data access is more complicated as it often requires navigating through multiple records and pointers.
බොහෝ විට බහු වාර්තා සහ දර්ශක හරහා සැරිසැරීමට අවශ්‍ය වන බැවින් දත්ත ප්‍රවේශය වඩාත් සංකීර්ණ වේ.

There is no universal standard for network databases, leading to variability in implementation.
ජාල දත්ත සමුදායන් සඳහා විශ්වීය සම්මතයක් නොමැත, එය ක්‍රියාත්මක කිරීමේ විචල්‍යතාවයට හේතු වේ.

4. Relational model
සම්බන්ධක ආකෘතිය

The relational database model is a way to organize and manage data using tables, which are known as relations.
සම්බන්ධක දත්ත සමුදා ආකෘතිය යනු සම්බන්ධතා ලෙස හැඳින්වෙන වගු භාවිතයෙන් දත්ත සංවිධානය කිරීමට සහ කළමනාකරණය කිරීමට පවතින මාර්ගයකි.

In this model, data is stored in rows and columns, where:
මෙම ආකෘතියේදී, දත්ත පේළි සහ තීරු වල ගබඩා කර ඇත, එහිදී:

Rows (also called tuples) represent individual records.
පේළි (() ලෙසද හැඳින්වේ) තනි වාර්තා නියෝජනය කරයි.

Columns (also called attributes) represent data fields for each record.
තීරු (උපලක්ෂණ ලෙසද හැඳින්වේ) එක් එක් වාර්තාව සඳහා දත්ත ක්ෂේත්‍ර නියෝජනය කරයි.

Each table represents a specific entity or type of data.
සෑම වගුවක්ම නිශ්චිත භූතාර්තයක් හෝ දත්ත වර්ගයක් නියෝජනය කරයි.

For example, a table for "Employees" might have columns like EmployeeID, Name, Department, and Salary. Each row would store the information for one employee.
උදාහරණයක් ලෙස, සේවකයින් සඳහා වන වගුවක සේවක හැඳුනුම්පත, නම, දෙපාර්තමේන්තුව, සහ වැටුප් වැනි තීරු තිබිය හැක. සෑම පේළියක්ම එක් සේවකයෙකු සඳහා තොරතුරු ගබඩා කරනු ඇත.

Employee

EID	Name	Dept	Salary
x	xxxxxx	xxxxxx	xxxxxx
x	xxxxxx	xxxxxx	xxxxxx
x	xxxxxx	xxxxxx	xxxxxx
x	xxxxxx	xxxxxx	xxxxxx

Key components of the relational model
සම්බන්ධතා ආකෘතියේ ප්‍රධාන සංරචක

Tables (Relations)
The fundamental building blocks in the relational model. Each table stores data about a particular subject.
සම්බන්ධතා ආකෘතියේ මූලිකකම කොටස් වේ. සෑම වගුවක්ම යම් එකක් පිළිබඳ දත්ත ගබඩා කරයි.

Each table must have a primary key (a unique identifier for each row) to ensure data integrity.
දත්ත අඛණ්ඩතාව සහතික කිරීම සඳහා සෑම වගුවකටම ප්‍රාථමික යතුරක් (එක් එක් පේළිය සඳහා අනන්‍ය හඳුනාගැනීමක්) තිබිය යුතුය.

Primary Keys
A column (or a combination of columns) that uniquely identifies each row in a table.
වගුවක එක් එක් පේළිය අනන්‍යව හඳුනා ගන්නා තීරුවක් (හෝ තීරු එකතුවක්) වේ.

For example, EmployeeID can be the primary key in the "Employees" table.
උදාහරණයක් ලෙස, EmployeeID "සේවකයින්" වගුවේ ප්‍රාථමික යතුර විය හැක.

Foreign Keys
A column (or a set of columns) in one table that is linked to the primary key of another table. Foreign keys establish relationships between tables.
එක් වගුවක තීරුවක් (හෝ තීරු කට්ටලයක්) තවත් වගුවක ප්‍රාථමික යතුරට සම්බන්ධ කර ඇත. ආගන්තුක යතුරු වගු අතර සම්බන්ධතා ඇති කරයි.

Relationships
One-to-One
One-to-Many
Many-to-Many

Will be discussed further in ER diagrams
ER රූප සටහන් වලදී තවදුරටත් සාකච්ඡා කරනු ඇත

Characteristics

Structured Data

Data is organized in a highly structured way, making it easy to search, query, and manipulate using SQL (Structured Query Language).

SQL (ව්‍යුහගත විමසුම් භාෂාව) භාවිතයෙන් සෙවීම, විමසීම් සහ හැසිරවීම පහසු කරමින් දත්ත ඉතා ව්‍යුහගත ආකාරයෙන් සංවිධානය කර ඇත.

Data Integrity

Constraints like primary keys, foreign keys, and unique constraints ensure that data is valid and consistent across the database.

ප්‍රාථමික යතුරු, ආගන්තුක යතුරු, සහ අනන්‍ය සම්බාධක වැනි සීමාවන් දත්ත සමුදාය පුරා දත්ත වලංගු සහ ස්ථාවර බව සහතික කරයි.

Normalization

This is the process of organizing the tables to minimize redundancy and dependency.

අතිරික්තය සහ යැපීම අවම කිරීම සඳහා වගු සංවිධානය කිරීමේ ක්‍රියාවලිය මෙයයි.

By splitting data into related tables, normalization reduces duplication and improves data integrity.

දත්ත අදාළ වගු වලට බෙදීමෙන්, ප්‍රමතකරණය මගින් අනුපිටපත් කිරීම අඩු කරන අතර දත්ත අඛණ්ඩතාව වැඩි දියුණු කරයි.

Benefits

Flexibility

New tables or relationships can be added easily without disrupting the existing structure.

පවතින ව්‍යුහයට බාධා නොකර නව වගු හෝ සම්බන්ධතා පහසුවෙන් එකතු කළ හැක.

Data Integrity

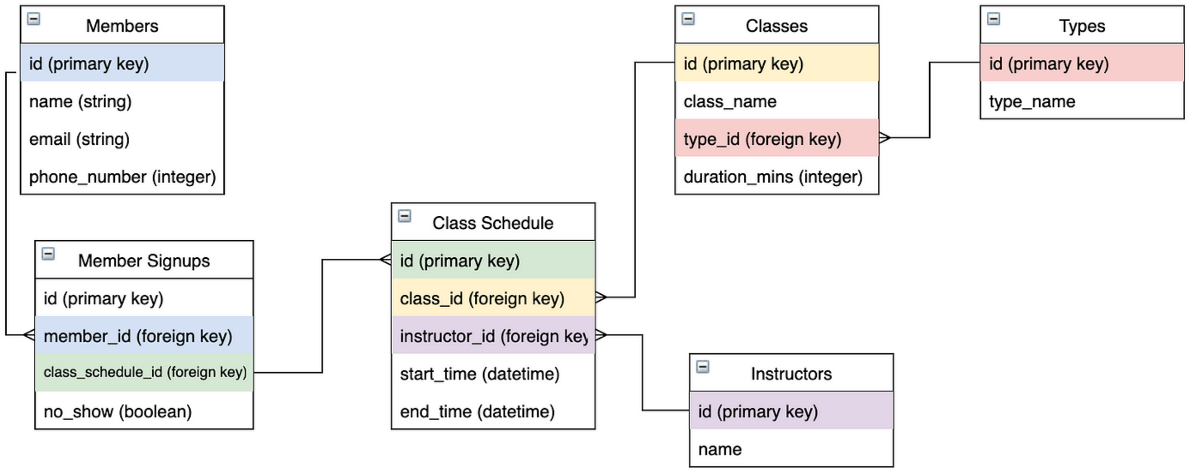
Relationships between tables ensure that data is consistent and valid.

වගු අතර සම්බන්ධතා දත්ත ස්ථාවර සහ වලංගු බව සහතික කරයි.

Efficient Data Retrieval

Relational databases are optimized for complex queries using SQL.

SQL භාවිතයෙන් සංකීර්ණ විමසුම් සඳහා සම්බන්ධතා දත්ත සමුදායන් ප්‍රශස්ත කර ඇත



5. Object-Relational Model
වස්තුව-සම්බන්ධතා ආකෘතිය

The Object-Relational Database (ORD) model is a hybrid approach that combines features of both relational databases and object-oriented databases.
වස්තු සම්බන්ධතා දත්ත සමුදා (ORD) ආකෘතිය යනු සම්බන්ධතා දත්ත සමුදායන් සහ වස්තු-නැඹුරු දත්ත සමුදායන් යන දෙකෙහිම ලක්ෂණ ඒකාබද්ධ කරන දෙමුහුන් ප්‍රවේශයකි.

It extends the traditional relational database model by incorporating elements from object-oriented programming, such as classes, objects, inheritance, and methods, into the relational framework.
එය පන්ති, වස්තු, උරුම සහ වැනි වස්තු-නැඹුරු ක්‍රමලේඛනවල මූලාංග සම්බන්ධතා රාමුවට ඇතුළත් කිරීමෙන් සම්ප්‍රදායික සම්බන්ධතා දත්ත සමුදා ආකෘතිය දිගු කරයි.

Classes

A class is a blueprint or template for creating objects. It defines the properties (also called attributes or fields) and behaviors (also called methods or functions) that the objects created from the class will have.
පන්තියක් යනු වස්තූන් නිර්මාණය කිරීම සඳහා වූ සැලැස්මක් හෝ අච්චුවකි. එමගින් පන්තියෙන් සාදන ලද වස්තූන් සතු ගුණාංග (උපලක්ෂණ හෝ ක්ෂේත්‍ර ලෙසද හැඳින්වේ) සහ හැසිරීම් (ක්‍රම හෝ ශ්‍රිත ලෙසද හැඳින්වේ) නිර්වචනය කරයි.

A class itself does not represent any real-world entity, but rather describes what those entities will look like and how they will behave.
පන්තියක් තනිව ගත් කළ කිසිදු තථ්‍ය-ලෝක වස්තුවක් නියෝජනය නොකරයි, ඒ වෙනුවට එම (ඩී) කෙබඳු වනු ඇත්ද සහ ඒවා හැසිරෙන්නේ කෙසේද යන්න විස්තර කරයි.

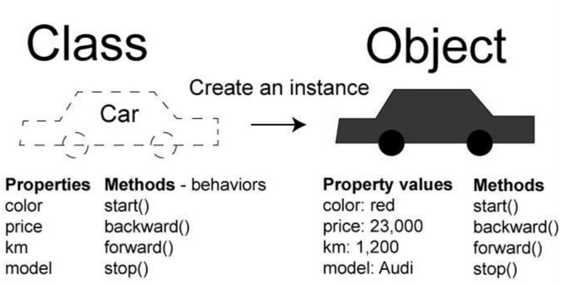
Think of a class as a recipe or blueprint for making objects, just like a blueprint for a house outlines the structure but is not the house itself.
නිවසක් සඳහා සැලැස්මක් ව්‍යුහය ගෙනහැර දක්වන නමුත් එය නිවසම නොවන පරිදි, වස්තු සෑදීමේ අච්චුවක් හෝ සැලැස්මක් ලෙස පන්තියක් සිතන්න.

Objects

An object is an instance of a class. When a class is defined, no memory is allocated until you create an object from it. Objects are actual entities that hold real data, and they can interact with one another.
වස්තුවක් යනු පන්තියක එක් අවස්ථාවකි. පන්තියක් නිර්වචනය කළ විට, ඔබ එයින් වස්තුවක් සාදන තුරු මතකයක් වෙන් නොකෙරේ. වස්තු යනු සත්‍ය දත්ත රඳවා තබා ගන්නා සත්‍ය භූතාර්ත වන අතර ඒවාට එකිනෙකා සමඟ අන්තර් ක්‍රියා කළ හැකිය.

If the class is the blueprint, the object is the actual house built from that blueprint.
පන්තිය සැලසුම නම්, වස්තුව එම සැලැස්මෙන් ගොඩනඟන ලද සැබෑ නිවසයි.

An object is something you can interact with, and each object has its own attributes (data) and can perform methods (actions).
වස්තුවක් යනු ඔබට අන්තර්ක්‍රියා කළ හැකි දෙයක් වන අතර සෑම වස්තුවකටම තමන්ගේම ගුණාංග (දත්ත) ඇති අතර ක්‍රම (ක්‍රියා) සිදු කළ හැකිය.



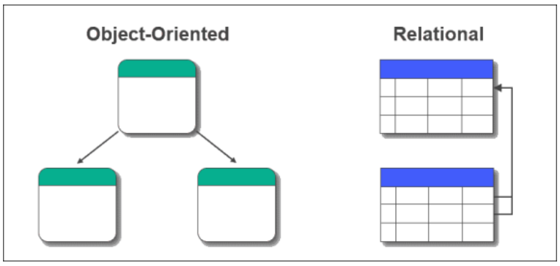
Similar to relational databases, data is still stored in tables where rows represent records, and columns represent fields.
සම්බන්ධක දත්ත සමුදායන් හා සමානව, දත්ත තවමත් ගබඩා කර ඇත්තේ පේළි මගින් වාර්තා නියෝජනය කරන සහ තීරු මගින් ක්ෂේත්‍ර නියෝජනය කරන වගු වලය.

However, in an object-relational database, rows can contain objects, which can encapsulate more complex data types and behaviors.
කෙසේ වෙතත්, වස්තු-සම්බන්ධතා දත්ත සමුදායක් තුළ, පේළිවල වස්තූන් අඩංගු විය හැකි අතර, එමඟින් වඩාත් සංකීර්ණ දත්ත වර්ග සහ හැසිරීම් සංග්‍රහ කළ හැක.

Inheritance

Just like in object-oriented programming, the object-relational model supports inheritance.
වස්තු-නැඹුරු ක්‍රමලේඛනයේදී මෙන්, වස්තු-සම්බන්ධතා ආකෘතිය inheritance සඳහා සහය දක්වයි.

This means that a new class (or table) can be derived from an existing class, inheriting its attributes and methods, while adding new ones or modifying existing ones.
මෙයින් අදහස් කරන්නේ නව පන්තියක් (හෝ වගුවක්) පවතින පන්තියක් මගින් එහි ගුණාංග සහ ක්‍රම උරුම කර ගනිමින්, නව ඒවා එකතු කරන අතරම හෝ පවතින ඒවා වෙනස් කරමින් ව්‍යුත්පන්න කළ හැකි බවය.



Benefits

Supports complex data types like multimedia and spatial data, making it ideal for applications that need more than simple tables.
බහුමාධ්‍ය සහ අවකාශීය දත්ත වැනි සංකීර්ණ දත්ත වර්ග සඳහා සහය දක්වයි, එය සරල වගුවලට වඩා අවශ්‍ය යෙදුම් සඳහා වඩාත් සුදුසු වේ.

Combines data and methods, allowing flexible and advanced operations directly within the database.
දත්ත සමුදාය තුළ සෘජුවම නම්‍යශීලී සහ උසස් මෙහෙයුම් වලට ඉඩ දෙමින් දත්ත සහ ක්‍රම ඒකාබද්ධ කරයි.

Works well with object-oriented programming, easing the transition between the application and database layers.
වස්තු-නැඹුරු වැඩසටහන් සමඟ හොඳින් ක්‍රියා කරයි, යෙදුම සහ දත්ත සමුදා ස්ථර අතර සංක්‍රමණය ලිහිල් කරයි.

Drawbacks

Adds complexity in design, management, and querying.
සැලසුම්, කළමනාකරණය සහ විමසීම් වල සංකීර්ණත්වය එක් කරයි.

Additional abstraction layers may reduce performance compared to traditional relational databases.
සාම්ප්‍රදායික සම්බන්ධතා දත්ත සමුදායන් හා සසඳන විට අතිරේක වියුක්ත ස්තර කාර්ය සාධනය අඩු කළ හැක.

Still relies on relational principles, limiting full use of object-oriented features.
වස්තු-නැඹුරු විශේෂාංගවල පූර්ණ භාවිතය සීමා කරමින් තවමත් සම්බන්ධතා මූලධර්ම මත රඳා පවතී.

Feature	**Flat File System**	**Hierarchical Model**	**Network Model**	**Relational Model**	**Object-Relational Model**
Data Structure දත්ත ව්‍යුහය	Simple files with no structure ව්‍යුහයක් නොමැති සරල ගොනු	Tree-like structure ගසක් වැනි ව්‍යුහය	Graph-like structure ප්‍රස්තාර වැනි ව්‍යුහය	Tables with rows and columns පේළි සහ තීරු සහිත වගු	Tables with object-oriented features වස්තු-නැඹුරු විශේෂාංග සහිත වගු
Complex Data Handling සංකීර්ණ දත්ත හැසිරවීම	Limited සීමා සහිතයි	Limited to hierarchical data ධූරාවලි දත්ත වලට සීමා වේ	Supports complex relationships සංකීර්ණ සබඳතා සඳහා සහාය වේ	Supports simple and moderately complex data සරල සහ මධ්‍යස්ථ සංකීර්ණ දත්ත සඳහා සහය දක්වයි	Supports complex data types සංකීර්ණ දත්ත වර්ග සඳහා සහය දක්වයි
Data Integrity දත්ත අඛණ්ඩතාව	Low අඩුයි	Moderate මධ්‍යස්ථ	Moderate මධ්‍යස්ථ	High ඉහළ	High ඉහළ
Flexibility නම්‍යශීලී බව	Low අඩුයි	Low අඩුයි	Moderate මධ්‍යස්ථ	High ඉහළ	High ඉහළ
Ease of Querying විමසීමේ පහසුව	Difficult, manual processing දුෂ්කර, අතින් සැකසීම	Difficult, using predefined paths දුෂ්කර, පූර්ව නිශ්චිත මාර්ග භාවිතා කිරීම	Complex, using network traversal සංකීර්ණ, ජාල සංක්‍රමණය භාවිතා කිරීම	Easy with SQL SQL සමඟ පහසුයි	Complex, using extended SQL or OOP methods සංකීර්ණ, විස්තීරණ SQL හෝ OOP ක්‍රම භාවිතයෙන්
Performance කාර්ය සාධනය	High (for simple data) ඉහළ (සරල දත්ත සඳහා)	High (for hierarchical queries) ඉහළ (ධූරාවලි විමසීම් සඳහා)	High for complex relationships සංකීර්ණ සබඳතා සඳහා ඉහළයි	Moderate to High මධ්‍යස්ථ සිට ඉහළ	Moderate (due to added complexity) මධ්‍යස්ථ (එකතු වූ සංකීර්ණත්වය හේතුවෙන්)
Use Case	Small, simple data storage කුඩා, සරල දත්ත ගබඩා කිරීම	Organizational charts, file systems ආයතනික ප්‍රස්ථාර, ගොනු පද්ධති	Telecommunications, CAD systems විදුලි සංදේශ, CAD පද්ධති	Business applications, general-purpose data ව්‍යාපාරික යෙදුම්, පොදු කාර්ය දත්ත	Complex applications, OOP environments සංකීර්ණ යෙදුම්, OOP පරිසරයන්

ictfromabc.com

Entity Relationship Diagrams
භූතාර්ත සම්බන්ධතා සටහන්

Peter Chen introduced the ER model in his paper titled "The Entity-Relationship Model, Toward a Unified View of Data".
පීටර් චෙන් ER ආකෘතිය හඳුන්වා දුන්නේ "The Entity-Relationship Model, Toward a Unified View of Data" යන ශීර්ෂය යටතේය.

Peter Chen

His goal was to create a way to represent database structure visually, making it easier to design databases and communicate between designers and stakeholders.
ඔහුගේ ඉලක්කය වූයේ දත්ත සමුදා ව්‍යුහය දෘශ්‍යමය වශයෙන් නිරූපණය කිරීමට ක්‍රමයක් නිර්මාණය කිරීම, දත්ත සමුදායන් සැලසුම් කිරීම සහ සංවර්ධකයන් සහ පාර්ශවකරුවන් අතර සන්නිවේදනය පහසු කිරීමයි.

The model uses entities (things or objects), relationships (associations between entities), and attributes (characteristics of entities or relationships).
මෙම ආකෘතිය භූතාර්ත, සම්බන්ධතා (භූතාර්ත අතර සම්බන්ධතාවය) සහ ගුණාංග (භූතාර්ත හෝ සම්බන්ධතාවල ලක්ෂණ) භාවිතා කරයි.

Chen's ER model was highly influential and led to the development of many other database models and notations.
Chen ගේ ER ආකෘතිය බෙහෙවින් බලගතු වූ අතර තවත් බොහෝ දත්ත සමුදා ආකෘති සහ අංකන වර්ධනය කිරීමට හේතු විය.

As databases became more complex, the ER model was extended to include more advanced concepts like generalization, specialization, inheritance, and aggregation.
දත්ත සමුදායන් වඩාත් සංකීර්ණ වූ විට, සාමාන්‍යකරණය, විශේෂීකරණය, උරුමය සහ එකතු කිරීම වැනි වඩාත් දියුණු සංකල්ප ඇතුළත් කිරීමට ER ආකෘතිය දීර්ඝ කරන ලදී.

This enhanced model, known as EER (Enhanced Entity-Relationship), allowed for the modeling of more complex real-world scenarios.
EER (Enhanced Entity-Relationship) ලෙස හැඳින්වෙන මෙම වැඩිදියුණු කළ ආකෘතිය, වඩාත් සංකීර්ණ සැබෑ-ලෝක අවස්ථා ආකෘතිකරණය සඳහා ඉඩ ලබා දී ඇත.

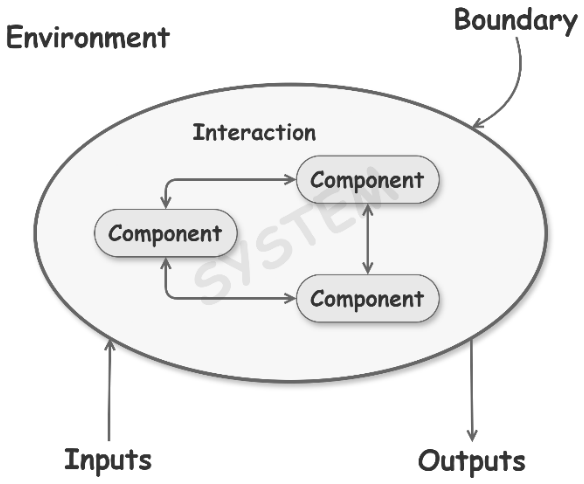
The Unified Modeling Language (UML) emerged in the 1990s as a comprehensive language for object-oriented design, incorporating concepts from ER diagrams.
ER සටහන් වල සංකල්ප ඇතුළත් කරමින් වස්තු-නැඹුරු නිර්මාණ සඳහා විස්තීර්ණ භාෂාවක් ලෙස 1990 දශකයේ දී Unified Modeling Language (UML) සංවර්ධනය කලේය.

UML includes class diagrams, which share many similarities with ER diagrams but are more focused on object-oriented principles.
UML හි පන්ති රූප සටහන් ඇතුළත් වන අතර, ඒවා ER රූප සටහන් සමඟ බොහෝ සමානකම් බෙදා ගන්නා නමුත් වස්තු-නැඹුරු මූලධර්ම කෙරෙහි වැඩි අවධානයක් යොමු කරයි.

ER diagrams started to be used not only for database design but also for business process modeling, helping organizations represent processes and data flow.
ER රූප සටහන් දත්ත සමුදා නිර්මාණය සඳහා පමණක් නොව ව්‍යාපාර ක්‍රියාවලි ආකෘති නිර්මාණය සඳහාද භාවිතා කිරීමට පටන් ගත් අතර, ක්‍රියාවලි සහ දත්ත ප්‍රවාහය නියෝජනය කිරීමට ආයතනවලට උපකාර කලේයා.

What is a System?
පද්ධතියක් යනු කුමක්ද?

A system is an organized set of interconnected components that work together to achieve a common goal or perform specific functions.
පද්ධතියක් යනු පොදු අරමුණක් සාක්ෂාත් කර ගැනීම සඳහා හෝ නිශ්චිත කාර්යයන් ඉටු කිරීම සඳහා එකට වැඩ කරන අන්තර් සම්බන්ධිත සංරචකවල සංවිධිත කට්ටලයකි



Characteristics of a System
පද්ධතියක ලක්ෂණ

1. A system consists of interrelated components or subsystems that function together. Each component performs a specific task, but they all contribute to the system's overall purpose.
පද්ධතියක් එකට ක්‍රියා කරන අන්තර් සම්බන්ධිත සංරචක හෝ උප පද්ධති වලින් සමන්විත වේ. සෑම සංරචකයක්ම නිශ්චිත කාර්යයක් ඉටු කරයි, නමුත් ඒවා සියල්ලම පද්ධතියේ සමස්ත අරමුණ සඳහා දායක වේ.

2. The components of a system are interconnected and depend on each other
පද්ධතියක සංරචක එකිනෙකට සම්බන්ධ වන අතර එකිනෙක මත රඳා පවතී

3. Systems take in inputs (such as resources, information, or energy), process them, and generate outputs (such as products, services, or results)
පද්ධති ආදාන ලබාගෙන (සම්පත්, තොරතුරු, හෝ බලශක්තිය වැනි), ඒවා ක්‍රියාවට නංවා, ප්‍රතිදානයන් ජනනය කරයි (නිෂ්පාදන, සේවා, හෝ ප්‍රතිඵල වැනි)

4. Every system has defined boundaries that distinguish it from its environment
සෑම පද්ධතියක්ම එහි පරිසරයෙන් වෙන්කර හඳුනාගත හැකි සීමාවන් නිර්වචනය කර ඇත

5. Systems are goal-oriented
පද්ධති ඉලක්ක-නැඹුරු වේ

6. Systems often have feedback loops that provide information about the performance of the system
පද්ධතිවල ක්‍රියාකාරීත්වය පිළිබඳ තොරතුරු සපයන ප්‍රතිපෝෂණ ලූප බොහෝ විට ඇත

7. Systems can vary in complexity, from simple to highly complex, depending on the number of components and the degree of interconnection between them
සංරචක සංඛ්‍යාව සහ ඒවා අතර අන්තර් සම්බන්ධතා මට්ටම මත පදනම්ව සරල සිට ඉතා සංකීර්ණ දක්වා පද්ධති සංකීර්ණත්වයෙන් වෙනස් විය හැක.

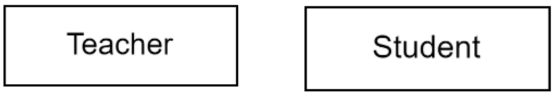
8. A system operates within an environment that can influence its behavior
පද්ධතියක් එහි හැසිරීමට බලපෑම් කළ හැකි පරිසරයක් තුළ ක්‍රියාත්මක වේ

9. A system should be viewed as a whole rather than focusing solely on individual components
පද්ධතියක් තනි තනි සංරචක කෙරෙහි පමණක් අවධානය යොමු නොකර සමස්තයක් ලෙස බැලිය යුතුය

What is an Entity?
භූතාර්ථයක් යනු කුමක්ද?

An entity is a distinct concept in a database about which data is stored. It can represent a real-world thing, person, place, event, or even an abstract idea that is relevant to the system being modeled.
භූතාර්ථයක් යනු දත්ත ගබඩා කර ඇති දත්ත සමුදාය තුළ ඇති වෙනස් සංකල්පයකි. එය සැබෑ ලෝකයේ දෙයක්, පුද්ගලයෙක්, ස්ථානයක්, සිදුවීමක් හෝ ආදර්ශයට ගන්නා පද්ධතියට අදාළ වියුක්ත අදහසක් පවා නියෝජනය කළ හැක.

An entity is represented by a rectangle in an ER diagram.
ER රූප සටහනක සෘජුකෝණාස්‍රයක් මඟින් භූතාර්ථයක් නිරූපණය කෙරේ.



NOTE
To identify an entity in a database, look for nouns in the system description, as nouns typically represent the objects or concepts being stored (Ex. Student, Course, Order).
දත්ත සමුදායක ඇති භූතාර්ථයක් හඳුනා ගැනීමට, නාම පද සාමාන්‍යයෙන් ගබඩා කර ඇති වස්තු හෝ සංකල්ප නියෝජනය කරන බැවින්, පද්ධති විස්තරයේ නාම පද සොයන්න (උදා., ශිෂ්‍ය, පාඨමාලාව, ඇණවුම).

Characteristics of an Entity
භූතාර්ථයක ලක්ෂණ

1. Entity can represent anything in the real world
භූතාර්ථයකට සැබෑ ලෝකයේ ඕනෑම දෙයක් නියෝජනය කළ හැක

Example

- A Person (Student, Teacher)
  පුද්ගලයෙක් (ශිෂ්‍යයා, ගුරුවරයා)

- A Place (Park, Restaurant, School, Hospital)
  ස්ථානයක් (උද්‍යානය, ආපන ශාලාව, පාසල, රෝහල)

- A Thing (Bicycle, Furniture, Appliance)
  දෙයක් (බයිසිකලය, ගෘහ භාණ්ඩ, උපකරණ)

- An Event (Order)
  සිදුවීමක් (ඇණවුම)

- A Concept (Project, Contract)
  සංකල්පයක් (උදා. ව්‍යාපෘතිය, කොන්ත්‍රාත්තුව)

2. Each entity should be distinguishable from others in the system.
සෑම භූතාර්ථයක්ම පද්ධතියේ අනෙක් අයගෙන් වෙන්කර හඳුනාගත හැකි විය යුතුය.

Example
In a university database, the "Student" entity must be distinguishable from other students.
විශ්වවිද්‍යාල දත්ත ගබඩාවක, "ශිෂ්‍ය" (Student) භූතාර්ථය අනෙකුත් සිසුන්ගෙන් වෙන්කර හඳුනාගත හැකි විය යුතුය.

The attribute Student ID can be used to uniquely identify each student. No two students will share the same Student ID, ensuring each student entity is distinct.
එක් එක් ශිෂ්‍යයා අනන්‍ය ලෙස හඳුනා ගැනීමට ශිෂ්‍ය හැඳුනුම්පත (Student ID) භාවිත කළ හැක. සිසුන් දෙදෙනෙකු එකම ශිෂ්‍ය හැඳුනුම්පත බෙදා නොගනී, එක් එක් ශිෂ්‍ය භූතාර්ථය වෙනස් බව සහතික කරයි.

3. Has an entity set: An entity set refers to the collection of similar entities.
භූතාර්ථ කුලකයක් ඇත: භූතාර්ථ කුලකයක් යනු සමාන භූතාර්ථවල එකතුවකි.
Ex.

- Entity set student consists of collection of students
  ශිෂ්‍යයා භූතාර්ථ කුලකය සිසුන්ගෙන් සමන්විත වේ
- Entity set Teacher consists of collection of Teachers
  ගුරුවරයා භූතාර්ථ කුලකය ගුරුවරුන්ගෙන් සමන්විත වේ

ictfromabc.com

4. An entity is described by its attributes, which represent its properties.
භූතාර්ථයක් එහි ගුණාංග නියෝජනය කරන එහි උපලක්ෂණ මගින් විස්තර කෙරේ.

Example

A Product entity in an inventory system may have attributes such as
ඉන්වෙන්ටරි පද්ධතියක නිෂ්පාදන (Product) භූතාර්ථයකට මෙවැනි ගුණාංග තිබිය හැක

- Product ID
- Name
- Price

5. Entities can be related to each other through relationships.
භූතාර්ථ සම්බන්ධතා හරහා එකිනෙකාට සම්බන්ධ විය හැක.

6. Some entities cannot exist without being related to other entities
සමහර භූතාර්ථ වෙනත් භූතාර්ථයකට සම්බන්ධ නොවී පැවතිය නොහැක

7. A system will not work properly without entities.
භූතාර්ථ නොමැතිව පද්ධතියක් නිසි ලෙස ක්‍රියා නොකරනු ඇත.

Example

In a banking system, The "Transaction" entity cannot exist without the "Account" entity.
බැංකු පද්ධතියක, “ගිණුම්“ භූතාර්ථය නොමැතිව “ගනුදෙනු“ භූතාර්ථය පැවතිය නොහැක.

A "Transaction" records actions like deposits, withdrawals, or transfers, but each transaction must be associated with a specific bank account.
“ගනුදෙනු“ තැන්පතු, මුදල් ආපසු ගැනීම් හෝ මාරු කිරීම් වැනි ක්‍රියා වාර්තා කරයි, නමුත් සෑම ගනුදෙනුවක්ම නිශ්චිත බැංකු ගිණුමක් සමඟ සම්බන්ධ විය යුතුය.

Entity identifier
භූතාර්ථ හඳුන්වනය

Entity identifiers are unique keys assigned to individual records within a database to ensure each entity can be distinctly identified and referenced.
භූතාර්ථ හඳුන්වනය යනු දත්ත සමුදායක් තුළ ඇති තනි වාර්තා සඳහා එක් එක් භූතාර්ථය පැහැදිලිව හඳුනා ගැනීමට සහ යොමු කිරීමට හැකි බව සහතික කිරීම සඳහා පවරා ඇති අද්විතීය යතුරු වේ.

Examples

National Identity Card Number (NIC)
ජාතික හැඳුනුම්පත් අංකය (NIC)
A unique identifier for individuals, such as 200512341234.
200512341234 වැනි පුද්ගලයන් සඳහා අනන්‍ය හඳුනාගැනීමක්.

ISBN
A unique identifier for books, like 978-3-16-148410-0.
978-3-16-148410-0 වැනි පොත් සඳහා අනන්‍ය හඳුනාගැනීමක්.

Vehicle Identification Number (VIN)
වාහන හඳුනාගැනීමේ අංකය (VIN)
A unique code for each vehicle, such as KX – 6644
KX - 6644 වැනි එක් එක් වාහනය සඳහා අනන්‍ය කේතයක්

Steps to Identify an Entity
භූතාර්ථයක් හඳුනා ගැනීමට පියවර

- **Understand the Purpose**
  අරමුණ තේරුම් ගන්න
  Know what the system is about (e.g., student management, e-commerce).
  පද්ධතිය කුමක් ගැනදැයි දැන ගන්න (උදා: ශිෂ්‍ය කළමනාකරණය, ඊ-වාණිජ්‍යය).

- **List Nouns**
  නාම පද ලැයිස්තුගත කරන්න
  Write down all the nouns from the requirements or description (e.g., Student, Course, Order).
  අවශ්‍යතා හෝ විස්තරයෙන් සියලුම නාම පද ලියන්න (උදා., ශිෂ්‍ය, පාඨමාලාව, ඇණවුම).

- **Check for Uniqueness**
  සුවිශේෂත්වය සඳහා පරීක්ෂා කරන්න
  Ensure each noun represents a distinct object or concept that needs to be stored.
  සෑම නාම පදයක්ම ගබඩා කළ යුතු වෙනස් වස්තුවක් හෝ සංකල්පයක් නියෝජනය කරන බවට සහතික වන්න.

- **Define a Key Attribute**
  ප්‍රධාන ගුණාංගයක් නිර්වචනය කරන්න
  Identify a unique identifier for each entity (Ex. Student ID, Order ID).
  එක් එක් භූතාර්ථය සඳහා භූතාර්ථ හඳුන්වනයක් හඳුනා ගන්න (උදා., ශිෂ්‍ය හැඳුනුම්පත, ඇණවුම් හැඳුනුම්පත).

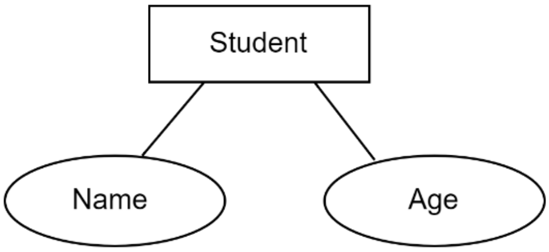
  Confirm the list of entities based on the above criteria.
  ඉහත නිර්ණායක මත පදනම්ව භූතාර්ථ ලැයිස්තුව තහවුරු කරන්න.

What are the attributes of an entity?
භූතාර්ථයක උපලක්ෂණ යනු මොනවාද?

Attributes of an entity refer to the characteristics or properties that define and describe an entity in a system.
භූතාර්ථයක උපලක්ෂණ යනු පද්ධතියක ඇති වස්තුවක් නිර්වචනය කරන සහ විස්තර කරන ලක්ෂණ හෝ ගුණාංග වේ.

Attributes provide specific details of an instance of an entity.
උපලක්ෂණ මගින් යම් භූතාර්ථයක නිදසුනක නිශ්චිත විස්තර සපයයි.

An attribute is usually represented by an oval in an ER diagram.
උපලක්ෂණයක් සාමාන්‍යයෙන් ER රූප සටහනක ඕවලාකාරයකින් නිරූපණය කෙරේ.



There are several types of attributes.
උපලක්ෂණ වර්ග කිහිපයක් ඇත.

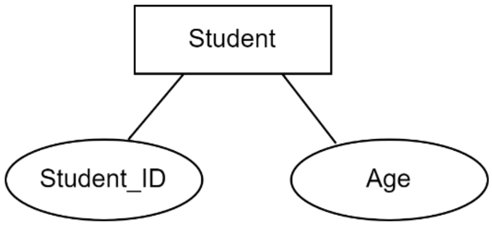
Simple attributes
සරල උපලක්ෂණ

Simple attributes refer to attributes that cannot be further divided into smaller components. They represent single, indivisible pieces of data.
සරල උපලක්ෂණ යනු කුඩා සංරචක වලට තවදුරටත් බෙදිය නොහැකි උපලක්ෂණ වේ. ඒවා තනි, බෙදිය නොහැකි දත්ත කොටස් නියෝජනය කරයි.

Simple attributes contain atomic values, meaning each one stores just a single piece of information.
සරල උපලක්ෂණවල පරමාණුක අගයන් අඩංගු වේ, එයින් අදහස් වන්නේ එක උපලක්ෂණය එකම තොරතුරක් පමණක් ගබඩා කරන බවයි.

They are represented by an oval in ER diagrams.
ඒවා ER රූප සටහන් වල ඕවලාකාරයකින් නිරූපණය කෙරේ.

Example
Here , Student_ID and Age are simple attributes of the entity Student
මෙහි , Student_ID සහ Age යනු Student යන භූතාර්ථයේ සරල උපලක්ෂණ වේ



Composite attributes
සංයුක්ත උපලක්ෂණ

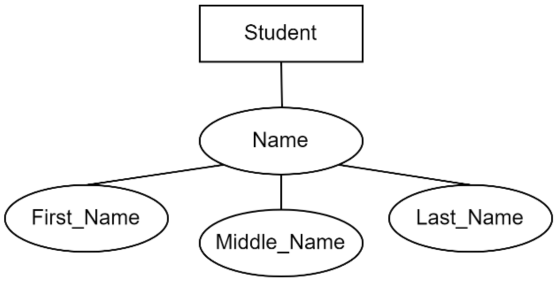
Composite attributes are attributes that can be divided into smaller sub-attributes, which represent more specific pieces of information.
සංයුක්ත උපලක්ෂණ යනු වඩාත් නිශ්චිත තොරතුරු කොටස් නියෝජනය කරන කුඩා උප-උපලක්ෂණ වලට බෙදිය හැකි උපලක්ෂණ වේ.

Each composite attribute is made up of two or more simple attributes.
සෑම සංයුක්ත උපලක්ෂණක්ම සරල උපලක්ෂණ දෙකකින් හෝ වැඩි ගණනකින් සෑදී ඇත.

The meaning of the composite attribute is derived from its constituent parts.
සංයුක්ත උපලක්ෂණයේ අර්ථය එහි සංඝටක කොටස් වලින් ලබා ගනී.

Composite attributes are represented by a larger oval labeled with the composite attribute's name, with smaller ovals for each sub-attribute connected to it, illustrating their relationship.
සංයුක්ත උපලක්ෂණ නිරූපනය කරනු ලබන්නේ සංයුක්ත උපලක්ෂණයේ නම සමඟ ලේබල් කර ඇති විශාල ඕවලාකාරයකින් වන අතර, ඒවාට සම්බන්ධ සෑම උප ගුණාංග සඳහාම කුඩා ඕවලාකාර, විශාල ඕවලාකාරයට සම්බන්ධ කර නිරූපණය කරනු ලැබේ

Example
In the Student entity, the Name composite attribute can be divided into sub attributes FirstName, MiddleName, and LastName.
ශිෂ්‍යයා භූතාර්ථය තුළ, නාම සංයුක්ත උපලක්ෂණය පළමු නම, මැද නම සහ අවසාන නම යන උප ගුණාංගවලට බෙදිය හැකිය.



Single valued attributes
තනි අගයක් සහිත උපලක්ෂණ

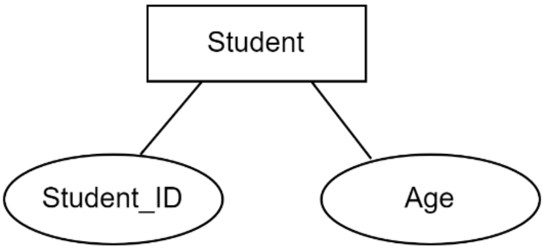
A single-valued attribute in a database refers to an attribute that holds only one value for a particular entity at any given time.
දත්ත සමුදායක තනි අගයක් සහිත උපලක්ෂණයක් යනු යම්කිසි අවස්ථාවකදී යම්කිසි භූතාර්ථයක් සඳහා එක් අගයක් පමණක් ඇති උපලක්ෂණයකි.

Single-valued attributes typically hold atomic values, meaning they represent one piece of information, which cannot be broken down further.
තනි අගය සහිත උපලක්ෂණ සාමාන්‍යයෙන් පරමාණුක අගයන් දරයි, එයින් අදහස් වන්නේ ඒවා තවදුරටත් බිඳ දැමිය නොහැකි එක් තොරතුරු කොටසක් නියෝජනය කරන බවයි.

Single-valued attributes are generally stored using simple data structures and there's no need for additional tables or arrays to manage multiple values
තනි වටිනාකම් සහිත උපලක්ෂණ සාමාන්‍යයෙන් සරල දත්ත ව්‍යුහයන් භාවිතයෙන් ගබඩා කර ඇති අතර බහු අගයන් කළමනාකරණය කිරීමට අමතර වගු හෝ අරා අවශ්‍ය නොවේ.

They are represented by an oval in ER diagrams.
ඒවා ER රූප සටහන් වල ඕවලාකාරයකින් නිරූපණය කෙරේ.

Example :
Here , Student_ID and Age are single valued attributes of the entity Student
මෙහි , Student_ID සහ Age යනු student යන භූතාර්ථයේ එක් අගයක් සහිත උපලක්ෂණය වේ



Multi valued attributes
අගයන් කිහිපයක් සහිත උපලක්ෂණ

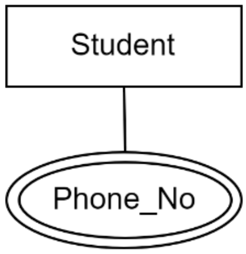
A multivalued attribute refers to an attribute in a database that can have more than one value for a particular entity at a given time.
අගයන් කිහිපයක් සහිත උපලක්ෂණ යන්නෙන් අදහස් කරන්නේ දත්ත සමුදායක් තුළ යම් නිශ්චිත භූතාර්ථයක් සඳහා යම් අවස්ථාවක දී එක් අගයකට වඩා තිබිය හැකි උපලක්ෂණයකි.

Multivalued attributes often require separate tables or complex data structures to store the multiple values.
අගයන් කිහිපයක් සහිත උපලක්ෂණ සඳහා බොහෝ විට බහු අගයන් ගබඩා කිරීම සඳහා වෙනම වගු හෝ සංකීර්ණ දත්ත ව්‍යුහයන් අවශ්‍ය වේ.

Multivalued attributes can complicate database design and are often handled by normalization.
අගයන් කිහිපයක් සහිත උපලක්ෂණ දත්ත සමුදාය නිර්මාණය සංකීර්ණ කළ හැකි අතර බොහෝ විට සාමාන්‍යකරණය මගින් හසුරුවනු ලැබේ.

They are represented by a double lined oval in ER diagrams.
ඒවා ER රූප සටහන් වල ද්විත්ව ඉරි සහිත ඕවලාකාරයකින් නිරූපණය කෙරේ.

Example
Phone_No attribute of student entity is a multivalued attribute since one student can have one or more phone number.
එක් සිසුවෙකුට දුරකථන අංක එකක් හෝ කිහිපයක් තිබිය හැකි බැවින්, Phone_no උපලක්ෂණ student භූතාර්ථයෙහි අගයන් කිහිපයක් සහිත උපලක්ෂණයකි .



ictfromabc.com

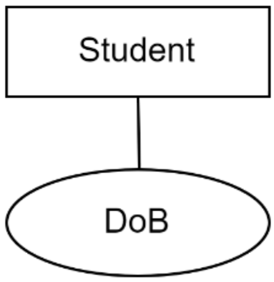
Stored attributes
ගබඩා කළ උපලක්ෂණ

Stored attributes in a database are attributes whose values are explicitly stored in the database, rather than being derived or calculated from other attributes.
දත්ත සමුදායක ගබඩා කළ උපලක්ෂණ යනු වෙනත් ගුණාංග වලින් ව්‍යුත්පන්න කිරීම හෝ ගණනය කිරීම වෙනුවට දත්ත ගබඩාවේ පැහැදිලිව ගබඩා කර ඇති ගුණාංග වේ.

These are directly entered and saved into the database and remain constant unless explicitly updated.
මේවා සමුදාය තුලට සෘජුවම ඇතුලත් කර දත්ත පවතින අතර පැහැදිලිව යාවත්කාලීන නොකළහොත් නියතව පවතී.

They are represented by an oval in ER diagrams.
ඒවා ER රූප සටහන් වල ඕවලාකාරයකින් නිරූපණය කෙරේ.

Example
DoB attribute in student entity is an example for a stored attribute.
Student භූතාර්ථයේ DoB උපලක්ෂණය ගබඩා කළ උපලක්ෂණයක් සඳහා උදාහරණයකි.



Derived attributes
ව්‍යුත්පන්න කළ උපලක්ෂණ

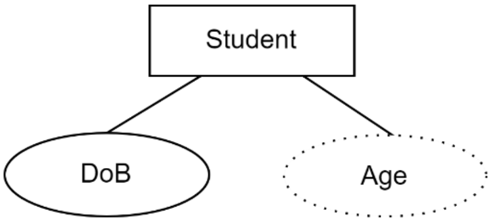
Derived attributes in a database are attributes whose values are not stored directly but are instead calculated or derived from other stored attributes.
දත්ත සමුදායක ව්‍යුත්පන්න කළ උපලක්ෂණ යනු අගයන් සෘජුව ගබඩා නොකරන නමුත් ඒ වෙනුවට ගණනය කර හෝ වෙනත් ගබඩා කර ඇති උපලක්ෂණ වලින් ව්‍යුත්පන්න කර ලබාගත් උපලක්ෂණ වේ.

Because they are computed at the time of retrieval, derived attributes reflect the most up-to-date information without requiring manual updates.
ඒවා ලබා ගැනීමේදී ගණනය කර ඇති නිසා, ව්‍යුත්පන්න උපලක්ෂණ අතින් යාවත්කාලීන කිරීම් අවශ්‍ය නොවී වඩාත්ම යාවත්කාලීන තොරතුරු පිළිබිඹු කරයි.

Derived attributes depend on the values of other stored attributes.
ව්‍යුත්පන්න ගුණාංග වෙනත් ගබඩා කර ඇති ගුණාංගවල අගයන් මත රඳා පවතී.

They are represented by a dashed oval in ER diagrams.
ඒවා ER රූප සටහන් වල කඩ ඉරි සහිත ඕවලාකාරයකින් නිරූපණය කෙරේ.

Example
Age attribute of student entity is an example for a derived attribute. Because age is derived from the stored attribute DoB.
Student භූතාර්ථයේ Age උපලක්ෂණය ව්‍යුත්පන්න උපලක්ෂණයක් සඳහා උදාහරණයකි. මක්නිසාද යත් Age ව්‍යුත්පන්න වී ඇත්තේ ගබඩා කර ඇති DoB යන උපලක්ෂණයෙනි.



Non-Key attributes
යතුරු නොවන උපලක්ෂණ

Non-key attributes in a database are attributes that do not participate in uniquely identifying an entity or record.
දත්ත සමුදායක ඇති යතුරු නොවන උපලක්ෂණ යනු භූතාර්ථයක් හෝ වාර්තාවක් අනන්‍ය ලෙස හඳුනා ගැනීමට සහභාගී නොවන උපලක්ෂණ වේ.

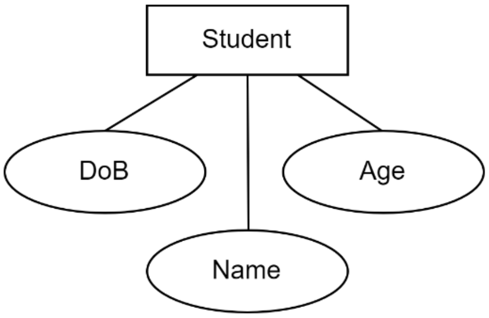
Multiple records can have the same value for a non-key attribute.
යතුරු නොවන උපලක්ෂණයක් සඳහා බහු වාර්තා වලට එකම අගයක් තිබිය හැක.

They add details that help define or describe the entity but do not influence its identification.
ඔවුන් භූතාර්ථය නිර්වචනය කිරීමට හෝ විස්තර කිරීමට උපකාර වන නමුත් එය හඳුනා ගැනීමට බලපෑම් නොකරන විස්තර එකතු කරයි.

In a well-designed database, non-key attributes should depend on the key attributes.
හොඳින් සැලසුම් කරන ලද දත්ත සමුදායක් තුළ, යතුරු නොවන උපලක්ෂණ යතුරු උපලක්ෂණ මත රඳා පවතී.

They are represented by an oval in ER diagrams.
ඒවා ER රූප සටහන් වල ඕවලාකාරයකින් නිරූපණය කෙරේ.

Example
DoB, Name, Address of student entity are some examples for non key attributes
Student භූතාර්ථයෙ DoB, Name, Address යතුරු නොවන උපලක්ෂණ සඳහා උදාහරණ වේ



Key attributes
යතුරු උපලක්ෂණ

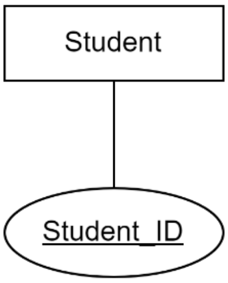
A key attribute in a database is an attribute or a set of attributes that uniquely identifies each entity or record in a table
දත්ත සමුදායක ඇති යතුරු උපලක්ෂණයක් යනු වගුවක ඇති එක් එක් භූතාර්ථයක් හෝ වාර්තාවක් අනන්‍යව හඳුනා ගැනීමට භාවිත කල හැකි උපලක්ෂණයකි.

Key attributes, especially primary keys, cannot have NULL values because they must uniquely identify records.
යතුරු උපලක්ෂණය, විශේෂයෙන්ම ප්‍රාථමික යතුරු, වාර්තා අනන්‍ය ලෙස හඳුනාගත යුතු බැවින්, ඒවාට හිස් (NULL) අගයන් තිබිය නොහැක.

Key attributes should not change over time. For example, it should remain constant to maintain a consistent reference to the entity.
යතුරු උපලක්ෂණ කාලයත් සමඟ වෙනස් නොවිය යුතුය. උදාහරණයක් ලෙස, භූතාර්ථයකට ස්ථාවර සඳහනක් පවත්වා ගැනීමට එය නියතව පැවතිය යුතුය.

They are represented by an oval with the name of attribute is underlined in ER diagrams.
ඒවා ER රූපසටහන්වල යටින් ඉරි ඇඳ ඇති උපලක්ෂණයේ නම සහිත ඕවලාකාරයකින් නිරූපණය කෙරේ.

Example :
Student_ID attribute in student entity is a key attribute.
Student භූතාර්ථයෙ Student_ID යතුරු උපලක්ෂණකි.



An attribute should be either
උපලක්ෂණයක් පහත ඒවාගෙන් එකක් වියයුතුය.

1. Simple or composite.
   සරල හෝ සංයුක්ත.

2. Single valued or multi valued.
   තනි අගයක් හෝ බහු අගයන් සහිත

3. Stored or derived.
   ගබඩා කල හෝ ව්‍යුත්පන්න කල.

4. Key or non-key
   යතුරු හෝ යතුරු නොවන

Ex:

- Simple single valued stored non-key attributes
  සරල තනි අගයක් සහිත ගබඩා කල යතුරු නොවන උපලක්ශණ
  Gender

- simple single valued stored key attribute
  සරල තනි අගයක් සහිත ගබඩා කල යතුරු උපලක්ශණ
  NIC

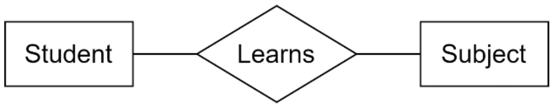
- composite single valued stored non-key attributes
  සංයුක්ත තනි අගයක් සහිත ගබඩා කල යතුරු නොවන උපලක්ශණ
  Name

- simple multivalued valued stored non-key attributes
  සරල අගයන් කිහිපයක් සහිත ගබඩා කල යතුරු නොවන උපලක්ශණ
  PhoneNumber

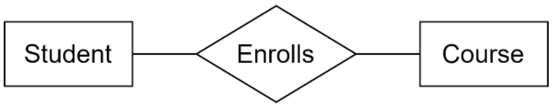
What is a relationship in ER diagrams?
ER රූප සටහන් වල සම්බන්ධතාවයක් යනු කුමක්ද?

A relationship in an Entity-Relationship (ER) diagram represents the interaction or association between two or more entities.
භූතාර්ථ-සම්බන්ධතා (ER) රූප සටහනක ඇති සම්බන්ධතාවයක් භූතාර්ථ දෙකක් හෝ වැඩි ගණනක් අතර අන්තර්ක්‍රියා හෝ සම්බන්ධය නියෝජනය කරයි.

Relationships are usually represented by a diamond shape and connected to the related entities using straight lines.
සබඳතා සාමාන්‍යයෙන් දියමන්ති හැඩයකින් නිරූපණය වන අතර සරල රේඛා භාවිතයෙන් අදාළ භූතාර්ථවලට සම්බන්ධ වේ.



NOTE:
It's easier to identify a relationship in a database or ER diagram by looking for verbs, as verbs describe the action or interaction between entities (e.g., a student enrolls in a course).
ක්‍රියා පද සෙවීමෙන් දත්ත සමුදායක හෝ ER රූප සටහනක සම්බන්ධතාවයක් හඳුනා ගැනීම පහසු වේ, ක්‍රියා පද මගින් භූතාර්ථ අතර ක්‍රියාව හෝ අන්තර් ක්‍රියාව විස්තර කරයි (උදා: ශිෂ්‍යයෙකු පාඨමාලාවකට ඇතුළත් වේ (enrolls))



Cardinality
ගණනීයතාව

Cardinality describes the quantitative relationships between entities.
ගණනීයතාව භූතාර්ථ අතර ප්‍රමාණාත්මක සම්බන්ධතා විස්තර කරයි.

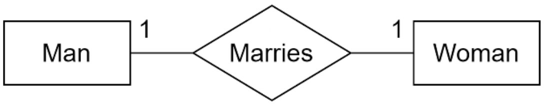
It defines how many instances of one entity can or must be associated with instances of another entity.
එය එක් භූතාර්ථයක අවස්ථා කීයක් තවත් භූතාර්ථයක අවස්ථා සමඟ සම්බන්ධ විය හැකිද හෝ තිබිය යුතුද යන්න නිර්වචනය කරයි.

Chen notation is a graphical method used to represent relationships between entities in a database system.
චෙන් අංකනය යනු දත්ත සමුදා පද්ධතියක භූතාර්ත අතර සම්බන්ධතා නියෝජනය කිරීමට භාවිතා කරන චිත්‍රක ක්‍රමයකි.

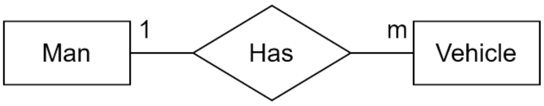
In Chen notation, cardinality is represented by placing numbers or symbols next to the lines that connect entities and relationships.
චෙන් අංකනයේදී, භූතාර්ත සහ සම්බන්ධතා සම්බන්ධ කරන රේඛා අසල සංඛ්‍යා හෝ සංකේත තැබීමෙන් ගණනීයතාව නිරූපණය කෙරේ.

Examples

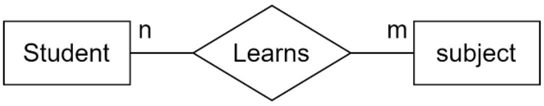
- One to one / ඒක ඒක (1:1)



- One to many / ඒක බහු (1:m)



- Many to many / බහු බහු (m:n)



Steps to Identify a Relationship
සම්බන්ධතාවයක් හඳුනා ගැනීමට පියවර

1. Understand the Interaction: Determine how the entities interact with each other based on the system's purpose (e.g., students enroll in courses, customers place orders).
අන්තර්ක්‍රියාව තේරුම් ගන්න: පද්ධතියේ අරමුණ මත පදනම්ව භූතාර්ථ එකිනෙකා සමඟ අන්තර් ක්‍රියා කරන ආකාරය තීරණය කරන්න (උදා: සිසුන් පාඨමාලා සඳහා ලියාපදිංචි වීම, පාරිභෝගිකයින් ඇණවුම් කිරීම).

2. Look for Verbs: Identify the verbs in the system description. (e.g., enrolls, orders, assigns).
ක්‍රියාපද සොයන්න: පද්ධති විස්තරයේ ක්‍රියාපද හඳුනා ගන්න (උදා: ලියාපදිංචි කිරීම, ඇණවුම් කිරීම, පැවරීම).

3. Define the Relationship: Use the verb to describe the relationship between entities (Ex. a "Student" enrolls in a "Course").
සම්බන්ධතාවය නිර්වචනය කරන්න: භූතාර්ථ අතර සම්බන්ධතාවය විස්තර කිරීමට ක්‍රියා පදය භාවිතා කරන්න (උදා: "ශිෂ්‍යයෙක්" "පාඨමාලාවකට" ඇතුළත් වේ (enrolls)).

4. Check Cardinality: Determine how many instances of one entity can be related to another (Ex. one-to-many, many-to-many).
ගණනීයතාව පරීක්ෂා කරන්න: එක් භූතාර්ථයක අවස්ථා කීයක් තවත් එකකට සම්බන්ධ විය හැකිද යන්න තීරණය කරන්න (උදා: one-to-many, many-to-many).

Types Of Relationships
සබඳතා වර්ග

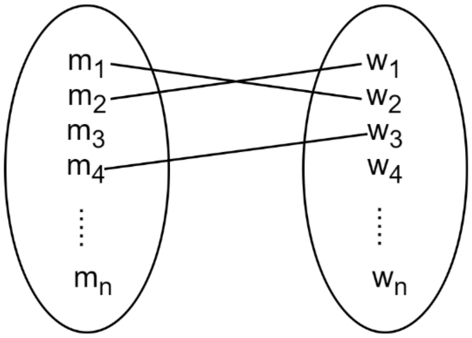
One to one Relationships (1:1)
ඒක ඒක සබඳතා

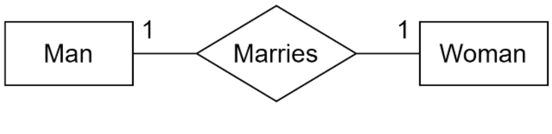
A one-to-one (1:1) relationship means that a single instance of one entity is associated with exactly one instance of another entity.
ඒක ඒක (1:1) සම්බන්ධතාවයක් යනු එක් භූතාර්ථයක තනි අවස්ථාවක් හරියටම වෙනත් භූතාර්ථයක එක් අවස්ථාවක් සමඟ සම්බන්ධ වී ඇති බවයි.

It is also possible for an instance of one entity to have no corresponding instance in the related entity.
එක් භූතාර්ථයක අවස්ථාවකට අදාළ භූතාර්ථයේ අනුරූප අවස්ථාවක් නොතිබීමටද හැකිය.

Example:
A man marries a woman.
පිරිමියෙක් කාන්තාවක් විවාහ කර ගනී.

In this case, one woman can have only one man, and one man can have only one woman. However, there can be men without women and women without men.
මෙම අවස්ථාවෙහිදී, එක් කාන්තාවකට සිටිය හැක්කේ එක් පිරිමියෙකු පමණක් වන අතර, එක් පුරුෂයෙකුට සිටිය හැක්කේ එක් කාන්තාවක් පමණි. කෙසේ වෙතත්, කාන්තාවන් නොමැතිව පිරිමින් සහ පිරිමින් නොමැතිව කාන්තාවන් සිටිය හැකිය





One to many relationships
ඒක බහු සබඳතා

A one-to-many relationship means that a single record in one entity can be associated with multiple records in another entity, but each record in the second entity is connected back to only one record in the first entity.
ඒක බහු සම්බන්ධතාවක් යන්නෙන් අදහස් වන්නේ එක් භූතාර්ථයක තනි වාර්තාවක් වෙනත් භූතාර්ථයක වාර්තා කිහිපයක් සමඟ සම්බන්ධ කළ හැකි නමුත් දෙවන භූතාර්ථයේ සෑම වාර්තාවක්ම පළමු භූතාර්ථයේ එක් වාර්තාවකට පමණක් සම්බන්ධ වී ඇති බවයි.

It is possible for an instance of one entity to exist without having any corresponding instances in the other entity.
එක් භූතාර්ථයක අවස්ථාවකට අනෙක් භූතාර්ථයේ අනුරූප අවස්ථා නොමැතිව පැවතිය හැකිය.කෙසේ

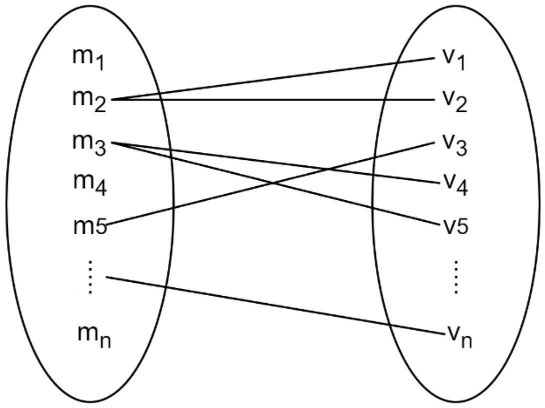
However, an instance of the other entity cannot exist without a corresponding instance in the first entity.
වෙතත්, පළමු භූතාර්ථයේ අනුරූප අවස්ථාවක් නොමැතිව අනෙක් භූතාර්ථයේ අවස්ථාවක් පැවතිය නොහැක.

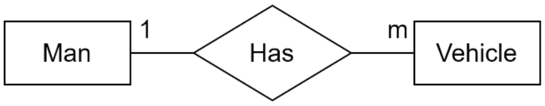
Example:

A Man has multiple vehicles
මිනිසෙකුට බොහෝ වාහන තිබේ.

Here, a man can have many vehicles, but a vehicle can have only one owner.
මෙහිදී මිනිසෙකුට බොහෝ වාහන තිබිය හැකි නමුත් වාහනයකට හිමිකරුවෙක් සිටිය හැක්කේ එක් අයෙකු පමණි.

A man can exist without having a vehicle, but a vehicle cannot exist without an owner.
වාහනයක් නොමැතිව මිනිසෙකුට පැවතිය හැකි නමුත් අයිතිකරුවෙකු නොමැතිව වාහනයකට පැවතිය නොහැක.





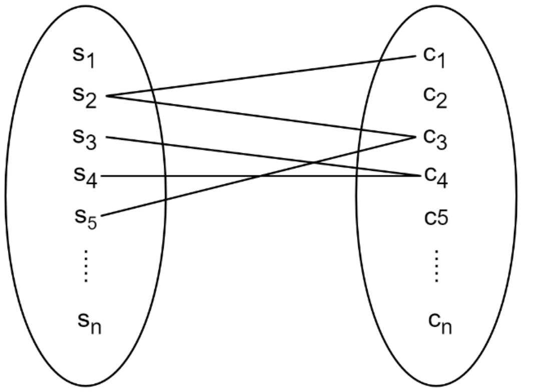
Many to many relationships
බහු බහු සබඳතා

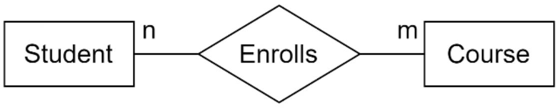
A many-to-many relationship occurs when multiple records in one entity are associated with multiple records in another entity.
එක් භූතාර්ථයක වාර්තා කිහිපයක් තවත් භූතාර්ථයක වාර්තා කිහිපයක් සමඟ සම්බන්ධ වූ විට බහු බහු සම්බන්ධතාවක් ඇතිවේ.

In a many-to-many relationship, both entities can exist independently without being in a relationship with each other.
බහු බහු සම්බන්ධතාවක් තුළ, භූතාර්ථ දෙකම එකිනෙකා සමඟ සබඳතාවයකින් තොරව ස්වාධීනව පැවතිය හැකිය.

Example:
A student enrolls in a course
ශිෂ්‍යයෙක් පාඨමාලාවකට ඇතුළත් වේ

Here, a student can enroll in multiple courses, and a course can have multiple students. A student can exist without enrolling in a course, and a course can exist without having any students.
මෙහිදී, ශිෂ්‍යයෙකුට පාඨමාලා කිහිපයකට ඇතුළත් විය හැකි අතර, පාඨමාලාවකට සිසුන් කිහිප දෙනෙකු සිටිය හැක. ශිෂ්‍යයෙකුට පාඨමාලාවකට බඳවා ගැනීමකින් තොරව පැවතිය හැකි අතර, සිසුන් නොමැතිව පාඨමාලාවක් පැවතිය හැකිය.





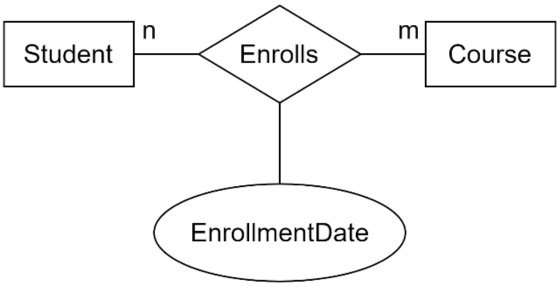
Descriptive Attributes
විස්තරාත්මක උපලක්ෂණ

Descriptive attributes linked to a relationship represent properties or characteristics that describe that particular relationship, rather than an entity itself.
සම්බන්ධතාවයකට සම්බන්ධ වූ විස්තරාත්මක උපලක්ෂණ නිරූපනය කරන්නේ යම් භූතාර්ථයකට වඩා එම විශේෂිත සම්බන්ධතාවය විස්තර කරන ගුණාංග හෝ ලක්ෂණ ය.

These attributes provide extra details about the relationship between entities but are not part of the entities themselves.
මෙම ගුණාංග භූතාර්ථ අතර සම්බන්ධතාවය පිළිබඳ අමතර විස්තර සපයන නමුත් ඒවා භූතාර්ථවලම කොටසක් නොවේ.

They help explain more about the interaction between the entities involved in that relationship.
එම සබඳතාවයට සම්බන්ධ භූතාර්ථ අතර අන්තර්ක්‍රියා ගැන වැඩි විස්තර පැහැදිලි කිරීමට උපකාර කරයි.

Example:
The Enrollment Date attribute in the Enrolls relationship between Student and Course specifies when the student enrolled.
ශිෂ්‍යයා(Student) සහ පාඨමාලා අතර ලියාපදිංචි වීමේ (Enrolls) සම්බන්ධතාවයේ ඇතුළත් වීමේ දින (Enrollment Date) ගුණාංගය මඟින් ශිෂ්‍යයා ඇතුළත් වූයේ කවදාදැයි සඳහන් කරයි.

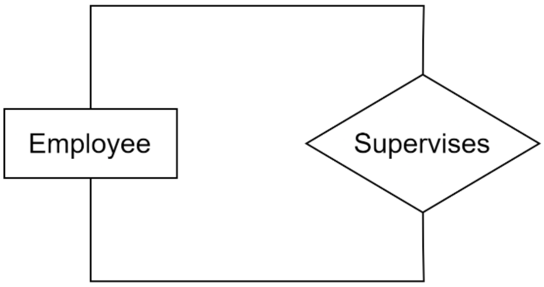


Degree of a relationship
සම්බන්ධතාවයක මාත්‍රය

Unary Relationship
ඒකීය සම්බන්ධතා

A unary relationship, also known as a recursive relationship, involves only one entity type participating in the relationship.
පුනරාවර්තන සම්බන්ධතාවයක් ලෙසද හැඳින්වෙන ඒකීය සම්බන්ධතාවයක් යනු එක් භූතාර්තයක අවස්ථා අතර හටගන්නා සම්බන්ධතාවයකි.

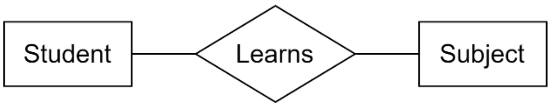
Example:
Employee entity where each employee can supervise other employees within the same organization.
එකම අයතනයේ සේවකයන් විසින් එම අයතනයේ අනෙක් සේවකයන් අධීක්ෂණය කිරීම



Binary Relationship
ද්විමය සම්බන්ධතා

This is the most common type of relationship. It connects instances of two different entities.
වඩාත්ම පොදු ආකාරයේ සම්බන්ධතා වර්ගය මෙයයි. එය භූතාර්ථ දෙකක අවස්ථා සම්බන්ධ කරයි.

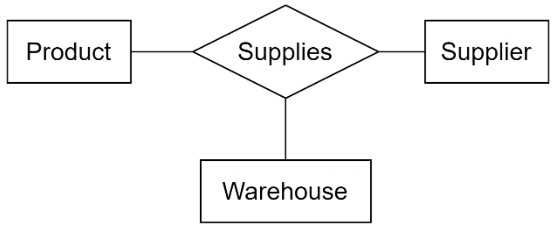
Example:
Student learns a subject.
ශිෂ්‍යයා විෂයයක් ඉගෙන ගනී



Ternary Relationship
ත්‍රිත්ව සම්බන්ධතා

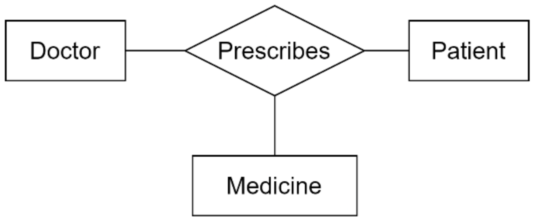
A ternary relationship involves three distinct entity types participating in the relationship.
ත්‍රිත්ව සම්බන්ධතාවයකට සම්බන්ධතාවයට සහභාගී වන වෙනස් භූතාර්ථ වර්ග තුනක් ඇතුළත් වේ.

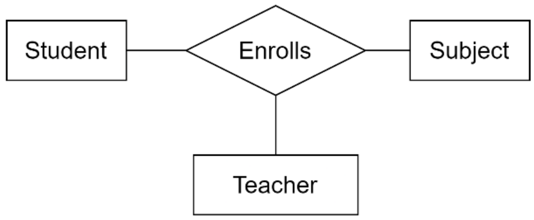
Example:
Supply chain management involves a Supplier supplying a Product to a Warehouse
සැපයුම් දාම කළමනාකරණයේදී සැපයුම්කරුවෙකු (Supplier) ගබඩාවකට (Warehouse) නිෂ්පාදනයක් (Product) සැපයීම



NOTE:
Relationship word should be carefully taken into consideration.
සම්බන්ධතාවය නිරූපණය කරන වචනය ප්‍රවේශමෙන් සැලකිල්ලට ගත යුතුය.

Example:

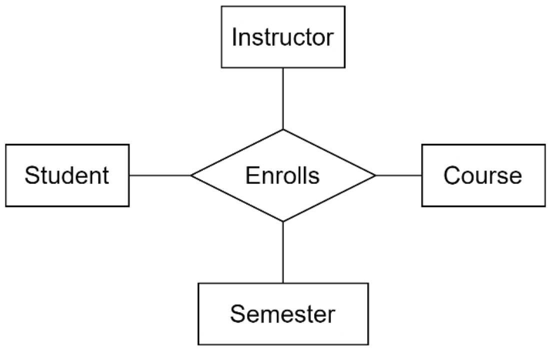




N-ary Relationship
N-කෝණික සම්බන්ධතා

An n-ary relationship in database design involves four or more entities that are interconnected, allowing for complex interactions among them
දත්ත සමුදා නිර්මාණයේ n-කෝණික සම්බන්ධතාවයක අන්තර් සම්බන්ධිත භූතාර්ථ හතරක් හෝ වැඩි ගණනක් ඇතුළත් වන අතර, ඒවා අතර සංකීර්ණ අන්තර්ක්‍රියා සඳහා ඉඩ සලසයි.

Example:
Students enroll in courses, which are taught by instructors and offered in specific semesters.
උපදේශකයින් (instructors) විසින් උගන්වනු ලබන සහ විශේෂිත අධ්‍යයන වාරවල (semesters) පිරිනමනු ලබන පාඨමාලා (courses) සඳහා සිසුන් (Students) ඇතුළත් වේ.



Ex:
Student-Course-Instructor-Semester-Classroom
Manufacturer-Distributor-Retailer-Product-Customer

Extended Entity Relationship Diagrams
විස්තෘත භූතාර්ත සම්බන්ධතා රූප සටහන් (EER)

Extended Entity-Relationship (EER) diagrams are an enhancement of the traditional Entity-Relationship (ER) diagrams.
විස්තෘත භූතාර්ත සම්බන්ධතා (EER) රූපසටහන් යනු සාම්ප්‍රදායික භූතාර්ත සම්බන්ධතා (ER) රූප සටහන් වල වැඩිදියුණු කිරීමකි.

Strong entities
ශක්තිමත් භූතාර්ත

It can exist independently of other entities. It has a primary key to uniquely identify each instance of the entity.
මේවාට වෙනත් භූතාර්ත වලින් ස්වාධීනව පැවතිය හැකිය. එයට භූතාර්ථයේ එක් එක් අවස්ථාව අනන්‍යව හඳුනා ගන්නා ප්‍රාථමික යතුරක් ඇත.

Can be uniquely identified by its attributes alone.
තමාගේ උපලක්ශණ වලින් පමණක් අනන්‍ය ලෙස හඳුනාගත හැකිය.
Strong entities represent standalone concepts.
ශක්තිමත් භූතාර්ත ස්වාධීන සංකල්ප නියෝජනය කරයි.

Strong entities are usually displayed with a normal rectangle in ER diagrams.
ශක්තිමත් භූතාර්ත සාමාන්‍යයෙන් ER සටහන් වල සාමාන්‍ය සෘජුකෝණාස්‍ර මගින් පෙන්වනු ලැබේ

Example.
Consider an entity "Customer" in a retail database system.
වෙළඳ දත්ත සමුදා පද්ධතියක "පාරිභෝගික" භූතාර්තය සලකා බලන්න.

Each customer can be uniquely identified by a CustomerID (Primary key), and the existence of a customer does not depend on any other entity.
සෑම පාරිභෝගිකයෙකුම පාරිභෝගික හදුන්වනයක් (ප්‍රාථමික යතුර) මගින් අනන්‍ය ලෙස හඳුනාගත හැකි අතර, පාරිභෝගිකයෙකුගේ පැවැත්ම වෙනත් කිසිදු භූතාර්තයක් මත රඳා නොපවතී.

Weak Entities
දුර්වල භූතාර්ත

They cannot exist independently without a relationship with a strong entity.
මේවාට ශක්තිමත් භූතාර්තයක් සමඟ සම්බන්ධතාවයකින් තොරව ස්වාධීනව පැවතිය නොහැක.

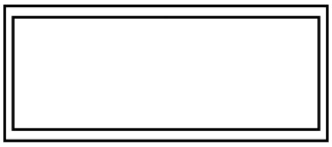
It does not have a primary key to uniquely identify each instance of the entity.
එයට භූතාර්ථයේ එක් එක් අවස්ථාව අනන්‍යව හඳුනා ගන්නා ප්‍රාථමික යතුරක් නැත.

Cannot be uniquely identified by its attributes alone.
තමාගේ උපලක්ශණ වලින් පමණක් අනන්‍ය ලෙස හඳුනාගත නොහැක.

They rely on a combination of some of its own attributes and the primary key attributes of the related strong entity.
එය රඳාපවතින්නේ එහි ඇතැම් උපලක්ශණ සහ එයට සම්බන්ධ ශක්තිමත් භූතාර්තයේ යතුරු උපලක්ශණ වල එකතුවක් මතය.

Weak entities typically represent dependent parts or components of a stronger entity.
දුර්වල භූතාර්ත සාමාන්‍යයෙන් ශක්තිමත් භූතාර්තයක යැපෙන කොටස් හෝ සංරචක නියෝජනය කරයි.

Weak entities are usually displayed with a double rectangle in ER diagrams.
දුර්වල භූතාර්ත සාමාන්‍යයෙන් ER සටහන් වල ද්විත්ව සෘජුකෝණාස්‍ර මගින් පෙන්වනු ලැබේ



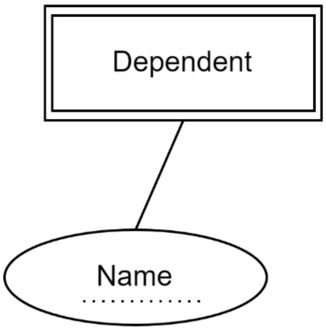
Partial Keys

A partial key in a weak entity is an attribute or a set of attributes that uniquely identifies an entity within the context of its related strong entity.
දුර්වල භූතාර්තයක ආංශික යතුරක් යනු එයට අදාල ශක්තිමත් භූතාර්ත සන්දර්භය තුළ එම දුර්වල භූතාර්තය අනන්‍ය ලෙස හඳුනා ගන්නා උපලක්ෂණයක් හෝ උපලක්ෂණ සමූහයකි.

Unlike a primary key in a strong entity, a partial key alone cannot uniquely identify a weak entity across the entire database because weak entities depend on strong entities for their identification.
ශක්තිමත් භූතාර්තයක ප්‍රාථමික යතුරක් මෙන් නොව, ආංශික යතුරකට පමණක් සම්පූර්ණ දත්ත සමුදාය හරහා දුර්වල භූතාර්තය අනන්‍ය ලෙස හඳුනාගත නොහැක, මන්ද දුර්වල භූතාර්ත ඒවාගේ හඳුනාගැනීම සඳහා ශක්තිමත් භූතාර්ත මත රඳා පවතින බැවිනි.

The partial key is combined with the primary key of the associated strong entity to create a composite key, which uniquely identifies the weak entity.
දුර්වල වස්තුව අනන්‍ය ලෙස හඳුනාගැනීම සදහා එහි ආංශික යතුර ආදාල ශක්තිමත් භූතාර්තයේ ප්‍රාථමික යතුර සමඟ සම්බන්ධ කර සංයුක්ත යතුරක් නිර්මාණය කරගනී.

A partial key is represented by a dotted line as the underline of the attribute.
ආංශික යතුරක් උපලක්ෂණයේ යටි ඉරි ලෙස තිත් රේඛාවකින් නිරූපණය කෙරේ.

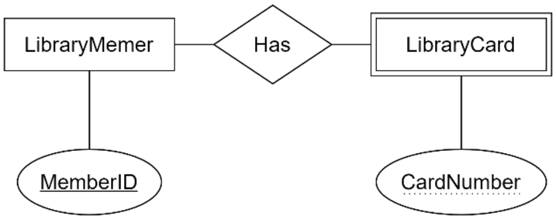


Example:
Consider a Library System:
පුස්තකාල පද්ධතියක් සලකා බලන්න:

Weak Entity: LibraryCard
Strong Entity: LibraryMember

A library card cannot exist without being associated with a library member.
පුස්තකාල සාමාජිකයෙකු සමඟ සම්බන්ධ නොවී පුස්තකාල කාඩ්පතක් පැවතිය නොහැක.

The card might be identified by a combination of the MemberID and CardNumber.
MemberID සහ CardNumber එකතුවකින් කාඩ්පත හඳුනාගත හැකිය.



Partial key – CardNumber
Primary key – MemberID

Combined key
MemberID + CardNumber

ictfromabc.com

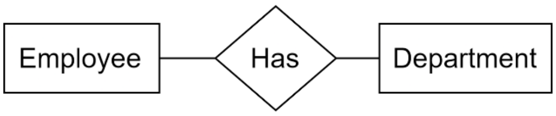
Strong Relationships

A strong relationship, also known as a regular relationship, connects strong entities or a combination of strong and weak entities.
ශක්තිමත් සම්බන්ධතාවයක් (සාමාන්‍ය සම්බන්ධතාවයක්) ශක්තිමත් භූතාර්ත දෙකක් හෝ ශක්තිමත් සහ දුර්වල භූතාර්ත දෙකක එකතුවක් සම්බන්ධ කරයි.

It does not include any ownership or existential dependency between the related entities. (Existence of one entity does not depend on the other)
එයට අදාළ භූතාර්ත අතර කිසිදු හිමිකාරිත්වයක් හෝ රඳාපැවැත්මක් නැත. (එක් භූතාර්තයක පැවැත්ම අනෙක මත රඳා පැවතීමක් හෝ එක් භූතාර්තයකට අනෙක් භූතාර්තය අයිති වීමක් නොපවතී.)

Represented using a single diamond in EERD.
EERD හි තනි දියමන්තියක් භාවිතයෙන් නිරූපණය කෙරේ.

Example:
The existence of a department doesn't depend on having employees, and vice versa.
දෙපාර්තමේන්තුවක පැවැත්ම රඳා පවතින්නේ සේවකයින් සිටීම මත නොවේ. සේවකයාගේ පැවැත්ම රඳා පවතින්නේ දෙපාර්තමේන්තුවක් තිබීම මත නොවේ.



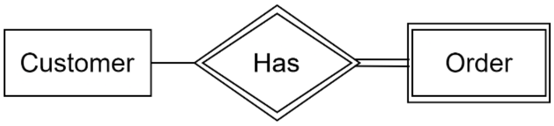
Weak Relationships

A weak relationship exists when a weak entity is connected to a strong entity, and there is ownership or existential dependence. (Existence of the weak entity depends on the strong entity)
දුර්වල භූතාර්තයක් ශක්තිමත් භූතාර්තයකට සම්බන්ධ වූ විට දුර්වල සම්බන්ධතාවයක් නිර්මාණය වන අතර එහිදී හිමිකාරිත්වයක් හෝ රඳාපැවතීමක් තිබේ. (දුර්වල භූතාර්තයේ පැවැත්ම ශක්තිමත් භූතාර්තය මත රඳා පැවතීමක් හෝ දුර්වල භූතාර්තය අනෙක් භූතාර්තයට අයිති වීමක් පවතී.)

Connection between two strong entities cannot ever be a weak relationship.
ශක්තිමත් භූතාර්ත දෙකක් අතර සම්බන්ධතාවය කිසිවිටෙක දුර්වල සම්බන්ධතාවයක් විය නොහැක.

Represented using a double diamond in EERD.
EERD හි ද්විත්ව දියමන්තියක් භාවිතයෙන් නිරූපණය කෙරේ.

Example:
An order cannot exist without being associated with a customer.
පාරිභෝගිකයෙකු සමඟ සම්බන්ධ නොවී ඇණවුමක් පැවතිය නොහැක



Partial Participation
අර්ධ වශයෙන් සහභාගීත්වය

Every instance of the entity is not required to participate in a relationship.
භූතාර්තයේ සෑම අවස්ථාවක්ම සම්බන්ධතාවයකට සහභාගී වීමට අවශ්‍ය නොවේ.

Partial participation is represented by a single line connecting the entity to the relationship.
ආංශික සහභාගීත්වය සම්බන්ධතාවයට භූතාර්තය සම්බන්ධ කරන තනි රේඛාවකින් නිරූපණය කෙරේ.

Most of the relationships are of partial participation unless told otherwise.
වෙනත් ආකාරයක් යැයි සඳහන් වුවහොත් මිස බොහෝ සබඳතා අර්ධ සහභාගීත්වයකින් යුක්ත වේ.

Total Participation
පූර්ණ වශයෙන් සහභාගීත්වය
(Also Known as Existence Dependency)

Every instance of an entity must participate in the relationship.
භූතාර්තයක සෑම අවස්ථාවක්ම සම්බන්ධතාවයට සහභාගී විය යුතුය.

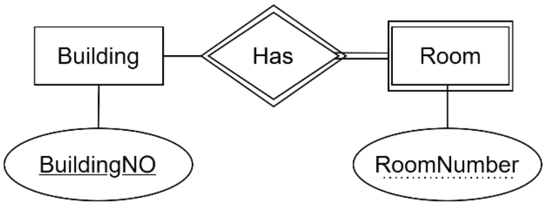
This concept can be applied to both strong and weak entities.
මෙම සංකල්පය ශක්තිමත් සහ දුර්වල භූතාර්ත දෙකම සඳහා යෙදිය හැකිය.

Represented as follows:
පහත ආකාරයට නියෝජනය කෙරේ.

Total Participation in Weak Entities
දුර්වල භූතාර්ත වල අර්ධ සහභාගීත්වය

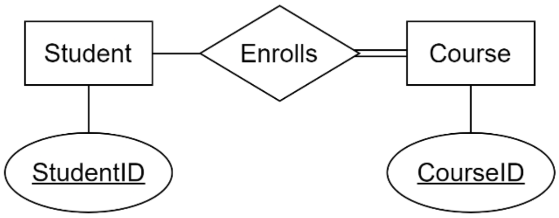
An entity "Room" might be a weak entity dependent on a "Building" entity.
Room යනු Building භූතාර්තය මත යැපෙන දුර්වල භූතාර්තයක් විය හැකිය.

Every room must be part of a building, so there's total participation of "Room" in the "Located In" relationship with "Building."
සෑම කාමරයක්ම ගොඩනැගිල්ලක කොටසක් විය යුතුය, එබැවින් Building සමඟ Located In සම්බන්ධතාවයෙහි Room පූර්ණ සහභාගීත්වය දක්වයි



Total Participation in Strong Entities
ශක්තිමත් භූතාර්ත වල අර්ධ සහභාගීත්වය

In a university system, if every student must enroll in at least one course, then the "Student" entity (a strong entity) would have total participation in the "Enrolled In" relationship with "Course."
විශ්ව විද්‍යාල පද්ධතියක් තුළ, සෑම ශිෂ්‍යයෙකුම අවම වශයෙන් එක් පාඨමාලාවකට ඇතුළත් විය යුතු නම්, Student භූතාර්තය (ශක්තිමත් භූතාර්තයක්) Course සමඟ Enrolled In සම්බන්ධතාවයේ සම්පූර්ණ සහභාගීත්වය දක්වයි.



ictfromabc.com

EER diagrams include additional modeling concepts to better represent complex real-world scenarios, such as inheritance, specialization, and generalization.
EER රූපසටහන් වලට තථ්‍ය-ලෝක අවස්ථා වඩාත් හොඳින් නිරූපණය කිරීමට උරුමය, විශේෂීකරණය සහ සාමාන්‍යකරණය වැනි සංකීර්ණ අමතර ආකෘති නිර්මාණ සංකල්ප ඇතුළත් වේ.

Generalization
සාමාන්‍යකරණය

Generalization is the process of extracting common properties from multiple entities to create a more general entity.
සාමාන්‍යකරණය යනු වඩාත් පොදු භූතාර්ථයක් නිර්මාණය කිරීම සඳහා භූතාර්ථ කිහිපයකින් පොදු උපලක්ෂණ උකහා ගැනීමේ ක්‍රියාවලියයි.

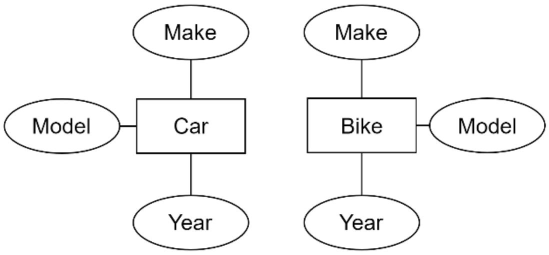
Useful when different entities share common attributes or behavior, and there's a need to eliminate redundancy.
විවිධ භූතාර්ථ වලට පොදු උපලක්ෂණ හෝ හැසිරීම් පවතින විට ප්‍රයෝජනවත් වන අතර අතිරික්ත දත්ත ඉවත් කිරීමට අවශ්‍ය වේ.

Represented as an inverted triangle connecting the generalized entity with the specific entities.
සාමාන්‍යකරණය කරන ලද භූතාර්ථය විශේෂිත භූතාර්ථ සමඟ සම්බන්ධ කල උඩු යටිකුරු ත්‍රිකෝණයක් ලෙස නිරූපණය කෙරේ.

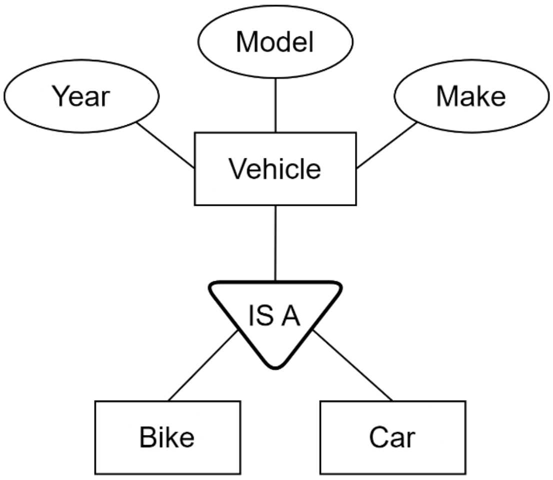
Generalization aims to encapsulate common attributes.
සාමාන්‍යකරණය පොදු උපලක්ෂණ encapsulate කිරීම සඳහා භාවිතා කරයි.

Example:

Suppose you have two entities, "car" and " bike". Both have attributes like "make," "model," and "year."
ඔබට car සහ bike ලෙස භූතාර්ථ දෙකක් ඇතැයි සිතන්න. ඒවා දෙකටම make, model සහ year වැනි උපලක්ෂණ ඇත.



You can generalize them into a more general entity called "Vehicle," which encapsulates the common attributes.
ඔබට ඒවා "වාහන" ලෙස හැඳින්වෙන වඩාත් සාමාන්‍ය භූතාර්ථයකට සාමාන්‍යකරණය කළ හැකිය. එය පොදු උපලක්ෂණ encapsulate කරයි.



Specialization
විශේෂීකරණය

Specialization is the opposite of generalization.
විශේෂීකරණය යනු සාමාන්‍යකරණයේ ප්‍රතිවිරුද්ධයයි.

It's the process of creating new entities from an existing entity based on some distinguishing characteristics.
එනම්, යම් සුවිශේෂී උපලක්ෂණ මත පදනම්ව පවතින භූතාර්ථයකින් නව භූතාර්ථ නිර්මාණය කිරීමේ ක්‍රියාවලියයි.

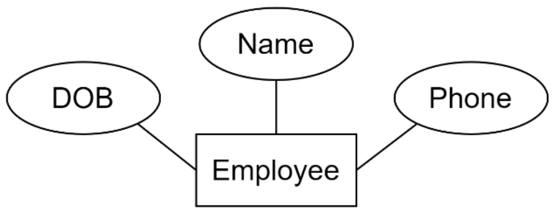
Ideal when an entity can be divided into more specific categories, each having unique attributes or relationships.
එක් එක් විශේෂ උපලක්ෂණ හෝ සම්බන්ධතා ඇති විට භූතාර්ථයක් වඩාත් කුඩා කාණ්ඩවලට බෙදීම සඳහා වඩාත් සුදුසුය.

Typically represented as a triangle connecting the higher-level entity with the specialized ones.
සාමාන්‍යයෙන් ඉහළ මට්ටමේ භූතාර්ථය විශේෂිත භූතාර්ථ සමඟ සම්බන්ධ කිරීම ත්‍රිකෝණයක් ලෙස දක්වයි.

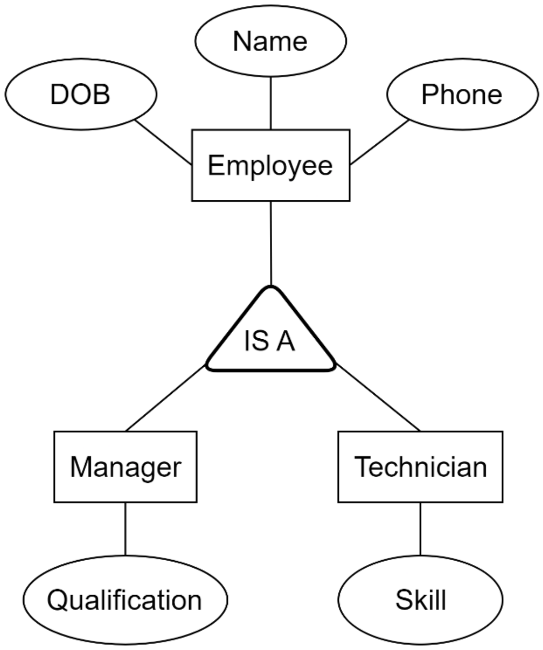
Specialization intends to highlight unique characteristics.
විශේෂීකරණය මගින් ආවේණික ලක්ෂණ ඉස්මතු කිරීමට අදහස් කරයි.

Example:

You have an entity "Employee," and you want to specialize it into "Manager" and "Technician," based on their roles.
ඔබට Employee භූතාර්ථයක් ඇති අතර, ඔබට එය ඔවුන්ගේ භූමිකාවන් මත පදනම්ව Manager සහ Technician ලෙස විශේෂීකරණය කිරීමට අවශ්‍ය වේ.



Here, the specialized entities may have attributes unique to them, such as "qualifications" for managers and "skills" for technicians.
මෙහිදී, විශේෂිත භූතාර්ථ අතර කළමනාකරුවන් සඳහා qualifications සහ කාර්මික ශිල්පීන් සඳහා skills වැනි ඒවාට අනන්‍ය වූ උපලක්ෂණ තිබිය හැක.



Inheritance

Sub-entities in a hierarchy can inherit attributes and relationships from their parent entities.
ධූරාවලියක උප භූතාර්ථ වලට ඒවාගේ parent භූතාර්ථවල ගුණාංග සහ සම්බන්ධතා උරුම විය හැක.

This is similar to the concept of inheritance in object-oriented programming.
මෙය වස්තු නැඹුරු ක්‍රමලේඛන වල inheritance යන සංකල්පයට සමානය.

Example:
A "Car" entity inherits attributes like "Make" and "Model" from its parent "Vehicle" entity.
"Car" යන භූතාර්ථයක් එහි parent භූතාර්ථය වන "Vehicle" මගින් "Make" සහ "Model" වැනි ගුණාංග උරුම කර ගනී.

Why use EER diagrams?
EER රූප සටහන් භාවිතා කරන්නේ ඇයි?

More Expressive Power
EER diagrams allow more complex data structures and relationships to be modeled compared to basic ER diagrams.
EER රූප සටහන් මූලික ER සටහන් වලට සාපේක්ෂව වඩාත් සංකීර්ණ දත්ත ව්‍යුහයන් සහ සම්බන්ධතා ආකෘතිගත කිරීමට ඉඩ සලසයි.

Hierarchical Representation
The specialization and generalization features make it easier to model hierarchical data.
විශේෂීකරණය සහ සාමාන්‍යකරණ විශේෂාංග ධූරාවලි දත්ත ආකෘතිකරණය පහසු කරයි.

Better Abstraction
The additional constructs help in abstracting real-world scenarios more effectively, making it suitable for designing complex databases.
සංකීර්ණ දත්ත සමුදායන් සැලසුම් කිරීම සඳහා එය යෝග්‍ය කරමින් සැබෑ ලෝකයේ අවස්ථා වඩාත් ඵලදායී ලෙස සාරාංශ කිරීමට අමතර සංවර්ධන උපකාරී වේ.

The terms "subclass" and "superclass" relate to the concepts of generalization and specialization.
උප පංතිය සහ සුපිරි පංතිය යන පද සාමාන්‍යකරණය සහ විශේෂීකරණය පිළිබඳ සංකල්පවලට සම්බන්ධ වේ.

Subclass
උප පංතිය

A subclass is a specialized entity that inherits attributes and relationships from a more general entity.
උප පංතියක් යනු වඩාත් පොදු භූතාර්ථයකින් ගුණාංග සහ සම්බන්ධතා උරුම කර ගන්නා විශේෂිත භූතාර්ථයකි.

It represents the derived class.
එය ව්‍යුත්පන්න පන්තිය නියෝජනය කරයි.

Superclass
සුපිරි පංතිය
A superclass is a more general entity that encapsulates the common attributes and relationships shared by its subclasses. It represents the base class in a hierarchy.
සුපිරි පන්තියක් යනු එහි උපපංතීන් විසින් බෙදාගන්නා පොදු උපලක්ශණ සහ සම්බන්ධතා කැටි කරගත් වඩාත් සාමාන්‍ය භූතාර්ථයකි. එය ධූරාවලියක මූලික පන්තිය නියෝජනය කරයි.

Summary

Feature	EER	ER
Entities	✓	✓
Relationships	✓	✓
Attributes	✓	✓
Primary Keys	✓	✓
Cardinality Constraints	✓	✓
Weak Entities	✓	✓
Specialization	✓	✕
Generalization	✓	✕
Inheritance	✓	✕
Subclasses	✓	✕
Superclasses	✓	✕

Mapping Entity Relationship Diagrams into Relational Database Tables
භූතාර්ථ සම්බන්ධතා රූප සටහන් සම්බන්ධිත දත්ත සමුදා වගු බවට හැරවීම

Relational schema
සම්බන්ධතා පටිපාටිය

Is a blueprint that represents the structure of a relational database derived from the ER model.
ER ආකෘතියෙන් ලබාගත් සම්බන්ධතා දත්ත සමුදායක ව්‍යුහය නියෝජනය කරන සැලැස්මකි.

It describes how the entities, attributes, and relationships from an ER diagram are transformed into tables.
එය ER රූප සටහනකින් ඇති භූතාර්ථ, උපලක්ෂණ සහ සම්බන්ධතා වගු බවට පරිවර්තනය වන ආකාරය විස්තර කරයි.

Ex.
`Student (StudentID, Name, Age)`

Relational instances
සම්බන්ධක අවස්ථා

A relational instance represents the current state of the data in a table, containing all the records (tuples) in that table at a given point in time.
සම්බන්ධක අවස්ථාවක් මඟින් කාලයේ ලබා දී ඇති යම් අවස්ථාවක වගුවේ ඇති සියලුම පේළි අඩංගු කරමින් එහි ඇති දත්තවල වත්මන් තත්ත්වය නියෝජනය කරයි.

Each row in a relational instance is a tuple, representing a unique record, and each column represents an attribute of the table.
සම්බන්ධක අවස්ථාවක සෑම පේළියක්ම ටපල් එකක් වන අතර, එය අනන්‍ය රෙකෝඩයක් නියෝජනය කරයි, සහ සෑම තීරුවක්ම වගුවේ උපලක්ෂණයක් නියෝජනය කරයි.

In the process of designing a database, Entity-Relationship Diagrams (ERDs) are converted into relational database tables as follows.
දත්ත සමුදායක් සැලසුම් කිරීමේ ක්‍රියාවලියේදී, භූතාර්ථ-සම්බන්ධතා රූප සටහන් (ERDs) සම්බන්ධිත දත්ත සමුදා වගු බවට පහත පරිදි පරිවර්තනය කෙරේ.

- System is converted into the Database.
  පද්ධතිය දත්ත සමුදාය බවට පරිවර්තනය වේ.

- Entities are converted into Tables.
  භූතාර්ථ වගු බවට පරිවර්තනය වේ.

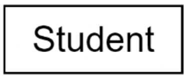
- Attributes are converted into Columns.
  උපලක්ෂණ තීරු බවට පරිවර්තනය වේ.

- Relationships are converted into keys.
  සම්බන්ධතා යතුරු බවට පරිවර්තනය වේ.

Apart from this, when modeling using SQL, Data types and Constraints are applied to each column of the table.
මීට අමතරව SQL මගින් නිර්මාණය කිරීමේදී වගුවේ සෑම තීරුවකටම දත්ත වර්ග සහ සම්බාධක යොදනු ලැබේ.

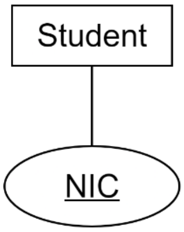
Mapping entities into tables
භූතාර්ථ වගු බවට හැරවීම

Each entity in an ERD is represented as a database table in a database.
ERD එකක ඇති සෑම භූතාර්ථයක්ම දත්ත සමුදායක දත්ත සමුදා වගුවක් ලෙස නිරූපණය කෙරේ.



Student

Mapping key attributes
යතුරු උපලක්ෂණ සිතියම්ගත කිරීම



Key attribute is e also represented as a simple column in a table but underlined.
යතුරු උපලක්ෂණද වගුවක සරල තීරුවක් ලෙස නිරූපණය වන නමුත් ඒවාට යටින් ඉරක් අඳිනු ලැබේ.

Key attribute, when represented in a database table is known as the primary key of the table.
යතුරු උපලක්ෂණය, දත්ත සමුදා වගුවක නිරූපණය වන විට වගුවේ ප්‍රාථමික යතුර ලෙස හැඳින්වේ.

student_table

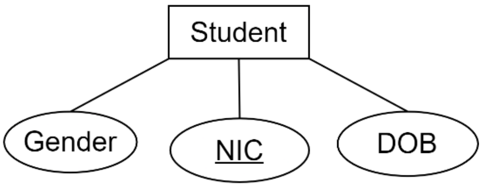
NIC	...	...

NOTE
Every table in a database must have a primary key or a key attribute to uniquely identify each record.
දත්ත සමුදායක ඇති සෑම වගුවකටම එක් එක් වාර්තාව අනන්‍ය ලෙස හඳුනා ගැනීමට ප්‍රාථමික යතුරක් තිබිය යුතුය.

The primary key ensures that no two rows (records) in the table have the same value in that key column, making each record distinct.
ප්‍රාථමික යතුර වගුවේ ඇති පේළි දෙකක් (වාර්තා) එම යතුරු තීරුවේ එකම අගයක් නොමැති බව සහතික කරයි, එමඟින් සෑම වාර්තාවක්ම වෙනස් වේ.

Mapping attributes into columns
උපලක්ෂණ වගුවේ තීරු බවට සිතියම්ගත කිරීම

Mapping common attributes
පොදු උපලක්ෂණ සිතියම්ගත කිරීම



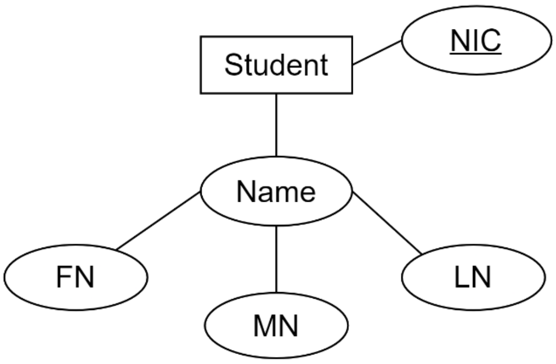
1. Simple attributes
සරල උපලක්ෂණ
2. Single valued attributes
තනි අගයක් සහිත උපලක්ෂණ
3. Stored attributes
ගබඩා කළ උපලක්ෂණ
4. Non key attributes
යතුරු නොවන උපලක්ෂණ
5. Key attributes
යතුරු උපලක්ෂණ

Each of these attributes are represented as a simple column in a table.
මෙම සෑම උපලක්ෂණයක්ම වගුවක සරල තීරුවක් ලෙස නිරූපණය කෙරේ.

Student_table

NIC	DOB	gender

Mapping composite attributes
සංයුක්ත උපලක්ෂණ සිතියම්ගත කිරීම



When mapping to a database table, each component of the composite attribute becomes a separate column.
දත්ත සමුදා වගුවකට සිතියම්ගත කිරීමේදී, සංයුක්ත උපලක්ෂණයේ එක් එක් සංරචක වෙනම තීරුවක් බවට පත්වේ.

The composite attribute itself does not become a column in the table—only the individual parts are represented.
සංයුක්ත උපලක්ෂණය වගුවේ තීරුවක් බවට පත් නොවේ (-) එක් එක් කොටස් පමණක් නියෝජනය වේ.

student_table

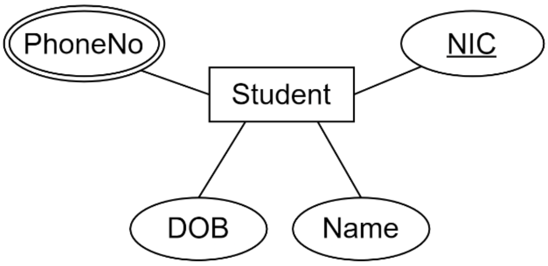
<u>NIC</u>	FN	MN	LN

Mapping derived attributes
ව්‍යුත්පන්න කල උපලක්ෂණ සිතියම්ගත කිරීම

Derived attributes are usually not stored: Instead of storing derived values in the table, they are computed when needed.
ව්‍යුත්පන්න උපලක්ෂණ සාමාන්‍යයෙන් ගබඩා නොවේ: වගුවේ ව්‍යුත්පන්න අගයන් ගබඩා කිරීම වෙනුවට, අවශ්‍ය විට ඒවා ගණනය කෙරේ.

Example:
Age can be derived from the DateOfBirth attribute, so Age is not stored directly in the database.
උපන්දිනය (DateOfBirth) යන ගුණාංගයෙන් වයස (Age) ව්‍යුත්පන්න කළ හැක, එබැවින් වයස සෘජුවම දත්ත ගබඩාවේ ගබඩා නොවේ.

Mapping multivalued attributes
අගයන් කිහිපයක් සහිත උපලක්ෂණ සිතියම්ගත කිරීම



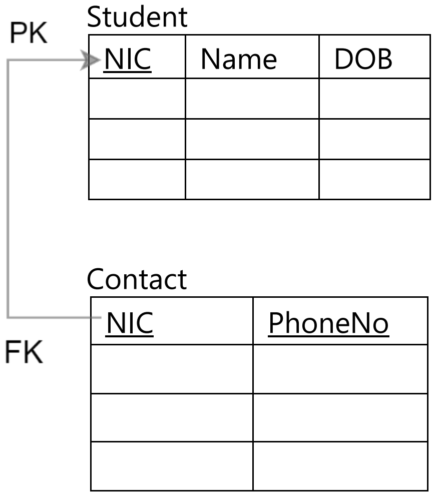
Multivalued attributes require a new table. Each value of the multivalued attribute is stored as a separate row in this table.
අගයන් කිහිපයක් සහිත උපලක්ෂණ සඳහා නව වගුවක් අවශ්‍ය වේ. අගයන් කිහිපයක් සහිත උපලක්ෂණයේ සෑම අගයක්ම මෙම වගුවේ වෙනම පේළියක් ලෙස ගබඩා කර ඇත.

The new table will contain a foreign key linking back to the original entity and a column for the multivalued attribute.
නව වගුවේ මුල් වගුව වෙත ආපසු සම්බන්ධ කරන ආගන්තුක යතුරක් සහ අගයන් කිහිපයක් සහිත උපලක්ෂණය සඳහා තීරුවක් අඩංගු වේ.

The new table’s primary key is usually a combination of the foreign key (NIC) and the multivalued attribute (PhoneNo)
නව වගුවේ ප්‍රාථමික යතුර සාමාන්‍යයෙන් ආගන්තුක යතුරක (NIC) සහ අගයන් කිහිපයක් සහිත උපලක්ෂණයේ (PhoneNo) එකතුවකි.

Never store multivalued attributes as comma-separated lists in a single column, as this violates database normalization rules.
අගයන් කිහිපයක් සහිත උපලක්ෂණ තනි තීරුවක කොමාවෙන් වෙන් කළ ලැයිස්තු දත්ත සමුදාය සාමාන්‍යකරණය කිරීමේ නීති උල්ලංඝනය කරන බැවින් එසේ ගබඩා කළ නොහැක .

We can't use multiple columns in the same table for multivalued attributes because it violates normalization principles, leads to data redundancy, complicates queries, and does not scale efficiently.
සාමාන්‍යකරණ මූලධර්ම උල්ලංඝණය කරන නිසාත්, දත්ත අතිරික්තයට මඟ පෙන්වන නිසාත්, විමසුම් සංකීර්ණ කරන නිසාත්, සහ කාර්යක්ෂමව පරිමාණය නොකෙරෙන නිසාත්, අගයන් කිහිපයක් සහිත උපලක්ෂණ සඳහා එකම වගුවේ තීරු කිහිපයක් භාවිත කළ නොහැක.

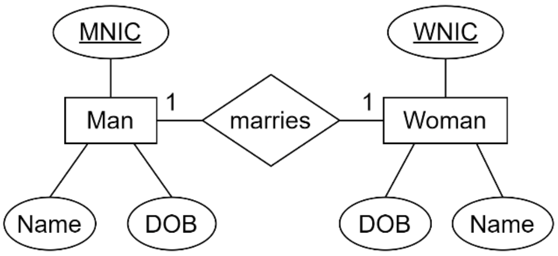


Mapping relationships into tables
සම්බන්ධතා වගු බවට හැරවීම

Mapping binary relationships
ද්විමය සම්බන්ධතා වගු බවට හැරවීම

When mapping binary relationships (relationships between two entities) from an ER diagram to a database, the approach depends on the cardinality of the relationship (one-to-one, one-to-many, or many-to-many).
ER රූප සටහනක සිට දත්ත සමුදායක් දක්වා ද්විමය සම්බන්ධතා සිතියම්ගත කිරීමේදී, එය රඳා පවතින්නේ සම්බන්ධතාවය (ඒක ඒක, ඒක බහු හෝ බහු බහු) මත ය.

1. One to one relationship
එකට එක සම්බන්ධතා වගු බවට හැරවීම



Create two separate tables for each entity.
එක් එක් භූතාර්තය සඳහා වෙන වෙනම වගු දෙකක් සාදන්න.

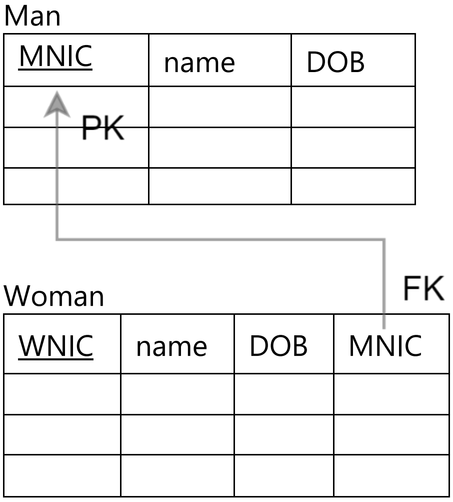
Represent attributes of each using separate columns.
වෙන වෙනම තීරු භාවිතා කරමින් එක් එක් උපලක්ශණ නිරූපණය කරන්න.

Method 1:

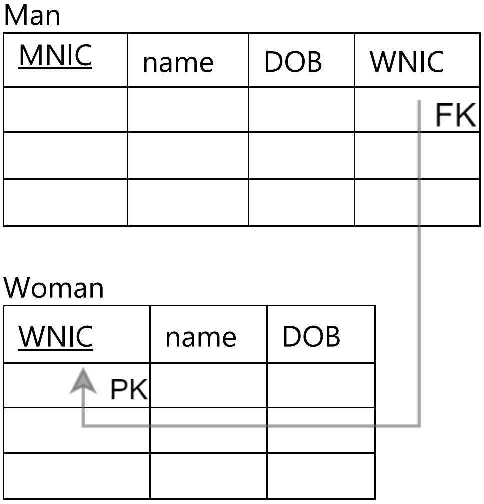
Primary key of any table can be interchanged as the foreign key of the other table to make the connection.
ඕනෑම වගුවක ප්‍රාථමික යතුර සම්බන්ධතාවය ඇති කිරීම සඳහා අනෙක් වගුවේ ආගන්තුක යතුර ලෙස හුවමාරු කර ගත හැක.

Man(MNIC,name,DOB)

Woman(WNIC,name,DOB,MNIC)



Man(MNIC,name,DOB,WNIC)
Woman(WNIC,name,DOB)



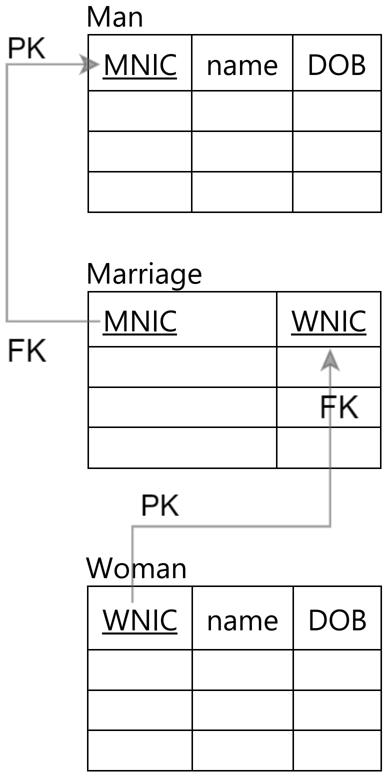
Method 2:

A separate relationship table can be created by using both primary keys of the table as a foreign key.
වගු දෙකේම ප්‍රාථමික යතුරු දෙකම ආගන්තුක යතුරු ලෙස භාවිතා කරමින් වෙනම සම්බන්ධතා වගුවක් සෑදිය හැක.

Man(MNIC,name,DOB)

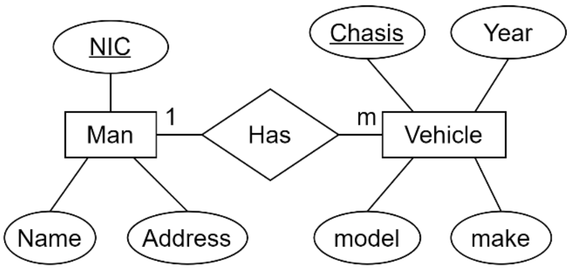
Marriage(MNIC,WNIC)

Woman(WNIC,name,DOB)



ictfromabc.com

2. One to many relationship
ඒක බහු සම්බන්ධතා වගු බවට හැරවීම



Create two separate tables for each entity.
එක් එක් භූතාර්ථය සඳහා වෙන වෙනම වගු දෙකක් සාදන්න.

Represent attributes of each using separate columns.
වෙන වෙනම තීරු භාවිතා කරමින් එක් එක් උපලක්ශණ නිරූපණය කරන්න.

Method 1:

Primary key of the "one" side table should be transferred to the "many" sides table. The other way is not possible to avoid horizontal data duplication.
“ඒක” වගුවේ ප්‍රාථමික යතුර “බහු” වගුව වෙත ආගන්තුක යතුරක් සේ මාරු කළ යුතුය. අනෙක් අතට හුවමාරුව සිදුකිරීමෙන් තිරස් අතට දත්ත අනුපිටපත්වීම වැළැක්විය නොහැක.

Man(<u>NIC</u>,name,address)
Vehicle(<u>chasis</u>,make,model,year,NIC)

Man

<u>NIC</u>	name	address
PK		

Vehicle

FK

<u>Chasis</u>	make	model	year	NIC

Method 2:

A separate relationship table can be created by using both primary keys of the two tables as foreign keys.
වගු දෙකේ ප්‍රාථමික යතුරු දෙකම ආගන්තුක යතුරු ලෙස භාවිත කිරීමෙන් වෙනම සම්බන්ධතා වගුවක් සෑදිය හැක.

Man(<u>NIC</u>,name,address)

Purchase(NIC,chasis)

vehicle(<u>chasis</u>,make,model,year)

PK

Man

<u>NIC</u>	name	address

Purchase

FK

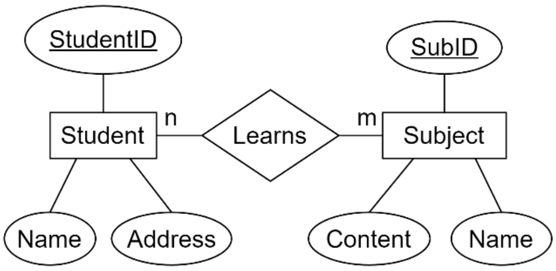
<u>NIC</u>	<u>Chasis</u>
	FK

PK

Vehicle

<u>Chasis</u>	make	model	year

3. Many to many relationship
බහු බහු සම්බන්ධතා වගු බවට හැරවීම

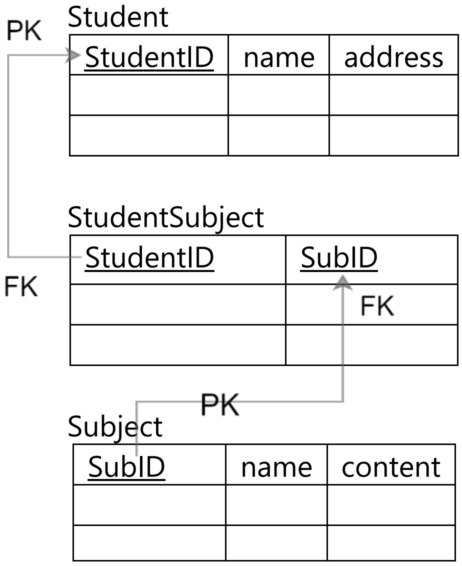


Create two separate tables for each entity.
එක් එක් භූතාර්තය සඳහා වෙන වෙනම වගු දෙකක් සාදන්න.
Represent attributes of each using separate columns.
වෙන වෙනම තීරු භාවිතා කරමින් එක් එක් උපලක්ශණ නිරූපණය කරන්න.

A separate relationship table can be created by using both primary keys of the two tables as foreign keys.
වගු දෙකේ ප්‍රාථමික යතුරු දෙකම ආගන්තුක යතුරු ලෙස භාවිත කිරීමෙන් වෙනම සම්බන්ධතා වගුවක් සෑදිය හැක.

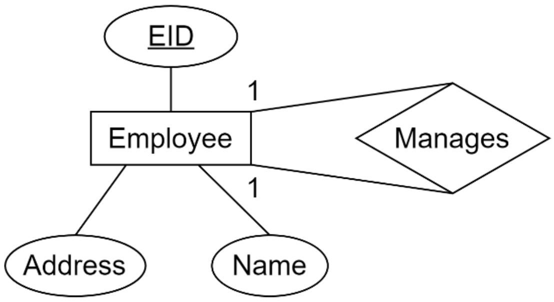
This avoids data duplication.
මෙමගින් දත්ත අනුපිටපත් වීම වළක්වයි.

```
Student(StudentID,name,address)
StudentSubject(StudentID,SubID)
subject(SubID,name,content)
```



Mapping unary relationships
ඒකීය සම්බන්ධතා වගු බවට හැරවීම

1. One to one relationship
එකට එක සම්බන්ධතා වගු බවට හැරවීම



Create the entity table with all the instances of the entity and their attributes. Add a self-referencing foreign key into the same table to represent the relationship.
භූථාර්තයේ සියලුම අවස්ථා සහ ඒවායේ ගුණාංග සමඟ භූථාර්ත වගුව සාදන්න. සම්බන්ධතාවය නියෝජනය කිරීම සඳහා ස්වයං-යොමු සහිත ආගන්තුක යතුරක් එම වගුවටම එක් කරන්න.

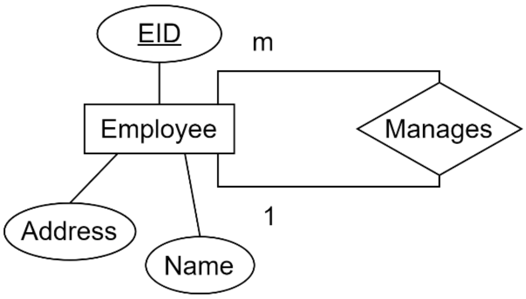
The foreign key column can have NULL values for records that do not have a corresponding related record.
ආගන්තුක යතුරු තීරුවට අදාළ වාර්තාවක් නොමැති වාර්තා සඳහා (NULL) අගයන් තිබිය හැක.

In an Employee table, with a one-to-one unary relationship, each employee can have a mentor who is also an employee, represented by adding a MentorID foreign key that references the EmployeeID of their mentor within the same table.
සේවක වගුවක් තුළ, එක් එක් සේවකයාට එකම වගුව තුළ තම උපදේශකයාගේ සේවක හැඳුනුම්පත (EID) යොමු කරන (MentorID) විදේශීය යතුරක් එක් කිරීමෙන් නියෝජනය වන සේවකයෙකු වන උපදේශකයෙකු සිටිය හැක.

2. One to many relationship
ඒක බහු සම්බන්ධතා වගු බවට හැරවීම

Create the entity table with all the instances of the entity and their attributes.
භූථාර්තයේ සියලුම අවස්ථා සහ ඒවායේ ගුණාංග සමඟ භූථාර්ත වගුව සාදන්න.

Add a self-referencing foreign key which allows multiple instances into the same table to represent the relationship.
සම්බන්ධතාවය නිරූපනය කිරීමට එකම වගුවෙහි අවස්ථා කිහිපයකට ඉඩ දෙන ස්වයං-යොමු සහිත ආගන්තුක යතුරක් එක් කරන්න.



For both 1:1 and 1:m

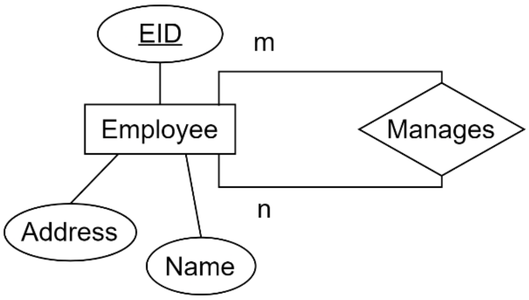
PK → FK

EID	Name	Address	MentorID
1	Akon	St. 1	Null
2	Bkon	St. 2	1
3	Ckon	St. 3	4
4	Dkon	St. 4	Null
5	Ekon	St. 5	1

3. Many to many relationship
බහු බහු සම්බන්ධතා වගු බවට හැරවීම

Create the entity table with all the instances of the entity and their attributes.
භූථාර්තයේ සියලුම අවස්ථා සහ ඒවායේ ගුණාංග සමඟ භූථාර්ත වගුව සාදන්න.

Create a separate junction table apart from the entity table containing two columns with the primary key of the entity. It also forms a composite primary key for the junction table.
සම්බන්ධතා නිරූපණය සඳහා භූතාර්තයේ ප්‍රාථමික යතුර ආගන්තුක යතුර ලෙස තීරු දෙකකට යොදාගත් වෙනම වගුවක් සාදන්න.
එය සම්බන්ධතා වගුව සඳහා සංයුක්ත ප්‍රාථමික යතුරක් ද සාදයි.



Employee

EID	Name	Address
1	Akon	St. 1
2	Bkon	St. 2
3	Ckon	St. 3
4	Dkon	St. 4
5	Ekon	St. 5

ManagingTable

EID	ManagerID
1	3
2	3
3	4
4	5
5	1

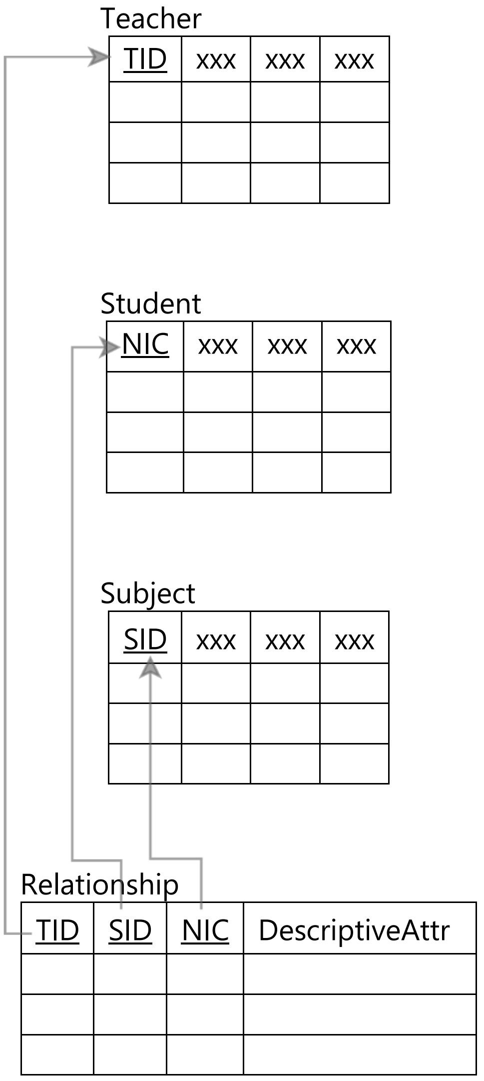
Mapping ternary relationships
තෘතීය සම්බන්ධතා වගු බවට හැරවීම

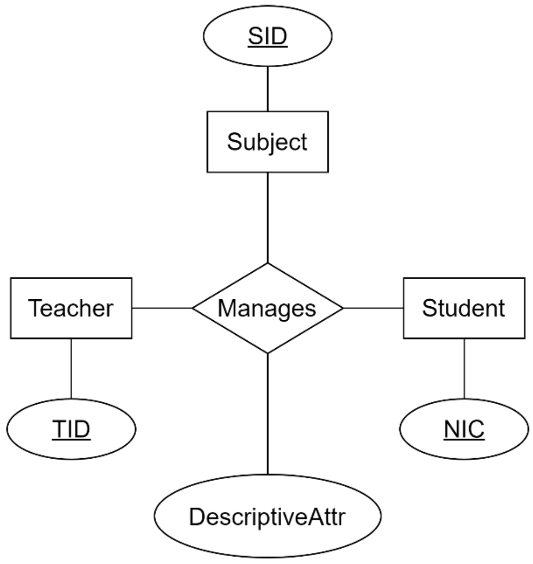
Create a new table. This table represents the ternary relationship.
නව වගුවක් සාදන්න. මෙම වගුව නියෝජනය කරන්නේ සම්බන්ධතාවයයි.

Include the primary keys from each of the three related entity tables as foreign keys in this new table.
භූතාර්ත වලට අදාළ වගු තුනෙන් ප්‍රාථමික යතුරු ගෙන මෙම නව වගුවේ ආගන්තුක යතුරු ලෙස ඇතුළත් කරන්න.

If the ternary relationship has any descriptive attributes, also add them as columns in this table.
ත්‍රිත්ව සම්බන්ධතාවයට යම් විස්තරාත්මක උපලක්ශණ තිබේ නම්, ඒවාද මෙම වගුවේ තීරු ලෙස එක් කරන්න.

Often, the composite of the three foreign keys will be used as the primary key of this table.
බොහෝ විට, මෙම වගුවේ ප්‍රාථමික යතුර ලෙස ආගන්තුක යතුරු තුනේ සංයුක්තය භාවිතා කරනු ලැබේ.



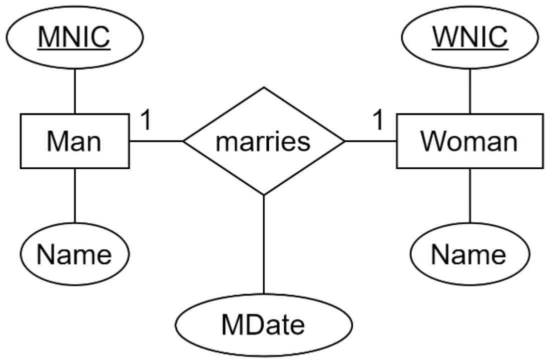


ictfromabc.com

Mapping relationship attributes (descriptive attributes) into tables
සම්බන්ධතා උපලක්ෂණ (විස්තරාත්මක උපලක්ෂණ) වගු බවට

Binary One-to-One Relationships
ද්විමය එකට එක සම්බන්ධතා

Here, descriptive attributes can be added in either tables, or even in a separate table linked via foreign keys.
මෙහිදී, විස්තරාත්මක උපලක්ෂණ එක් එක් භූථාර්ත වගු වලට හෝ ආගන්තුක යතුරු හරහා සම්බන්ධ කර ඇති සම්බන්ධතා වගුවට යෙදිය හැක.



Method 1:

Man

MNIC	name

Woman

WNIC	name	MNIC	MDate

Method 2:

Man

MNIC	name	WNIC	MDate

Woman

WNIC	name

Method 3:

Man

MNIC	name

Marriage

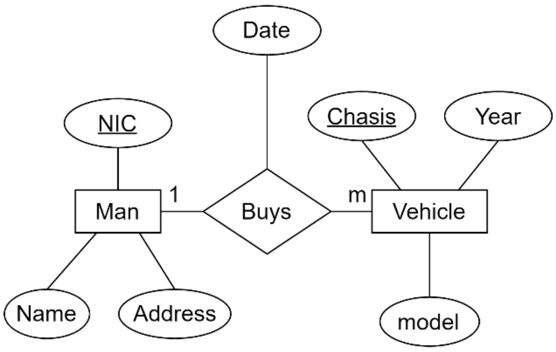
MNIC	WNIC	MDate

Woman

WNIC	name

Binary One-to-Many Relationships
ද්විමය ඒක බහු සම්බන්ධතා

Here, descriptive attributes should be added to the 'many' sides table or even in the separate table linked via foreign keys..
මෙහිදී, 'බහු' වගුවට හෝ ආගන්තුක යතුරු හරහා සම්බන්ධ කර ඇති සම්බන්ධතා වගුවට විස්තරාත්මක උපලක්ෂණය එකතු කළ යුතුය.



Method 1:

Man

NIC	name	address

Vehicle

Chasis	model	year	NIC	Date

Method 2:

Man

NIC	name	address

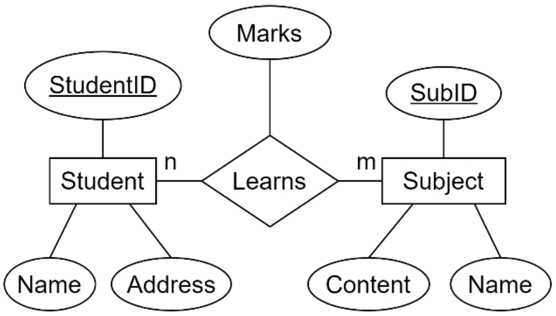
Purchase

NIC	Chasis	Date

Vehicle

Chasis	model	year

Binary Many-to-Many Relationships
ද්විමය බහු බහු සම්බන්ධතා



Here, descriptive attributes can be added to the relationship table to provide additional information about the relationship.
මෙහිදී, සම්බන්ධතාවය පිළිබඳ අමතර තොරතුරු සැපයීම සඳහා සම්බන්ධතා වගුවට විස්තරාත්මක උපලක්ෂණ යෙදීම සිදු කළ හැක.

Student

StudentID	name	address

StudentSubject

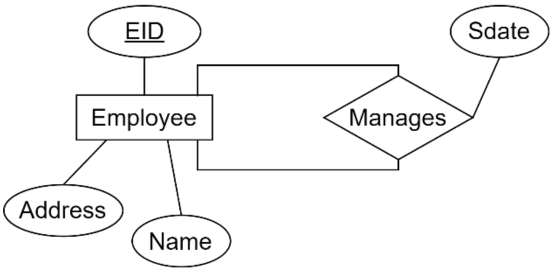
StudentID	SubID	Marks

Subject

SubID	name	content

Unary Relationships
ඒකීය සම්බන්ධතා

In a unary relationship, one to one and one to many, a descriptive attribute can be added directly to the table as a separate column.
ඒක ඒක සහ ඒක බහු ඒකීය සම්බන්ධතාවයකදී, විස්තරාත්මක උපලක්ෂණයක් වෙනම තීරුවක් ලෙස වගුවට කෙලින්ම එකතු කළ හැක.



<u>EID</u>	Name	MentorID	Sdate
1	Akon	Null	Null
2	Bkon	1	04/23
3	Ckon	4	05/30
4	Dkon	Null	Null
5	Ekon	1	06/09

In many to many, they should be added to the separate junction table containing two columns with the primary key of the entity.
බහු බහු සම්බන්ධතාවයකදී ඒවා භූතාර්තයේ ප්‍රාථමික යතුර සහිත තීරු දෙකක් අඩංගුවී සෑදෙන සම්බන්ධතා වගුවට එක් කළ යුතුය.

<u>EID</u>	Name
1	Akon
2	Bkon
3	Ckon
4	Dkon
5	Ekon

<u>EID</u>	MentorID	Sdate
2	1	04/23
3	4	05/30
5	1	06/09

Ternary Relationships
තෘතීය සම්බන්ධතා

Here, the descriptive attributes are included in the separate linking table containing foreign keys to each participating entity table.
මෙහිදී, සහභාගී වන එක් එක් භූථාර්ත වගුව සඳහා ආගන්තුක යතුරු අඩංගු කර සෑදෙන සම්බන්ධතා වගුවෙහි විස්තරාත්මක උපලක්ෂණ ඇතුළත් කරයි.

Example:
In a relationship between 'Suppliers', 'Products', and 'Stores', the relationship table include attributes like 'DeliveryDate'.
'සැපයුම්කරුවන්', 'නිෂ්පාදන' සහ 'වෙළඳසැල්' අතර සම්බන්ධතාවයකදී, සම්බන්ධතා වගුවේ DeliveryDate වැනි උපලක්ෂණ ඇතුළත් කරයි.

Different Types of keys
විවිධ වර්ගයේ යතුරු

Candidate Key
අපේක්ෂක යතුර

A column or a set of columns that can uniquely identify a record.
වාර්තාවක් අනන්‍යව හඳුනාගත හැකි තීරුවක් හෝ තීරු කට්ටලයක්.

A table can have multiple candidate keys, but only one is selected as the primary key.
වගුවකට අපේක්ෂක යතුරු කිහිපයක් තිබිය හැකි නමුත් ප්‍රාථමික යතුර ලෙස තෝරාගනු ලබන්නේ එකක් පමණි.

Example:
In an Employees table, both EmployeeID and NationalID could be candidate keys since each can uniquely identify an employee.
සේවක වගුවක(Employees table), සේවක හැඳුනුම්පත (EmployeeID) සහ ජාතික හැඳුනුම්පත (NationalID) යන දෙකම එක් එක් සේවකයෙකු අනන්‍ය ලෙස හඳුනා ගැනීමට භාවිත කල හැකි බැවින් අපේක්ෂක යතුරු වේ .

Primary Key
ප්‍රාථමික යතුර

A column or a set of columns that uniquely identifies each record in a table.
වගුවක ඇති සෑම වාර්තාවක්ම අනන්‍යව හඳුනා ගන්නා තීරුවක් හෝ තීරු කට්ටලයක්.

It cannot be NULL and must have unique values.
එය හිස් (NULL) විය නොහැකි අතර අනන්‍ය අගයන් තිබිය යුතුය.

Example:
StudentID in a Students table is a primary key because it uniquely identifies each student.
ශිෂ්‍ය වගුවක (Students table) ශිෂ්‍ය හැඳුනුම්පත (StudentID) මූලික යතුරක් වන්නේ එය එක් එක් ශිෂ්‍යයා අනන්‍ය ලෙස හඳුනා ගන්නා බැවිනි.

Alternate Key
විකල්ප යතුර

An alternate key is a candidate key that is not chosen as the primary key. It's an alternative unique identifier for a record.
විකල්ප යතුරක් යනු ප්‍රාථමික යතුර ලෙස තෝරා නොගත් අපේක්ෂක යතුරක් වේ. එය වාර්තාවක් සඳහා විකල්ප අනන්‍ය හඳුනාගැනීමකි.

Example
If EmployeeID is chosen as the primary key in an Employees table, NationalID would be an alternate key since it could also uniquely identify an employee but wasn’t selected as the primary key.
සේවක වගුවක (Employees table) ප්‍රාථමික යතුර ලෙස EmployeeID තෝරාගෙන තිබේ නම්, ජාතික හැඳුනුම්පත (NationalID) සේවකයෙකු අනන්‍ය ලෙස හඳුනාගත හැකි නමුත් ප්‍රාථමික යතුර ලෙස තෝරා නොගත් බැවින් එය විකල්ප යතුරක් වනු ඇත.

Foreign Key
ආගන්තුක යතුර

Foreign key is a copy of the primary key field of a certain table which is used inside another table for the reference of what relationship that the records of the second table is having with the records of the original table. This is not underlined.
ආගන්තුක යතුර යනු කිසියම් වගුවක ප්‍රාථමික යතුරු ක්ෂේත්‍රයේ පිටපතක් වන අතර එය දෙවන වගුවක වාර්තා මුල් වගුවේ වාර්තා සමඟ ඇති සම්බන්ධය නිරූපණය කිරීම සඳහා වගුවක් තුළ භාවිතා වේ. මෙය යටින් ඉරි ඇඳ නැත.

Foreign key types

- Unary one to one relationship
- Unary one to many relationships
- Unary many to many relationships
- Binary one to one relationship
- Binary one to many relationships
- Binary many to many relationships
- Ternary relationships
- Multivalued attributes

Composite Key
සංයුක්ත යතුර

A combination of two or more columns that together uniquely identify a record.
වාර්තාවක් අනන්‍යව හඳුනා ගන්නා තීරු දෙකක හෝ වැඩි ගණනක එකතුවකි.

Example:
In a CourseEnrollments table, the combination of StudentID and CourseID can form a composite key to uniquely identify each enrollment.
CourseEnrollments වගුවක, StudentID සහ CourseID සංයෝගය එක් එක් බඳවා ගැනීම් අනන්‍යව හඳුනා ගැනීමට සංයුක්ත යතුරක් ලෙස භාවිත කළ හැක.

Unique Key
අද්විතීය යතුර

A column that must have unique values but can accept NULL.
අනන්‍ය අගයන් තිබිය යුතු නමුත් හිස් (NULL) අගයන්ද තිබිය හැකි තීරුවක් වේ.

No duplicate entries in that column.
එම තීරුවේ අනුපිටපත් ඇතුළත් කිරීම් නොමැති විය යුතුය

Example
In a Users table, Email could be a unique key to ensure no two users have the same email address, though some rows might leave the email as NULL.
පරිශීලක වගුවක (Users table), සමහර පේළි හිස් ලෙස (NULL) විද්‍යුත් තැපෑල තැබිය හැකි වුවද, පරිශීලකයින් දෙදෙනෙකුට එකම විද්‍යුත් තැපැල් ලිපිනයක් නොමැති නිසා විද්‍යුත් තැපෑල අද්විතීය යතුරක් වේ.

Super Key
සුපිරි යතුර

A super key is a set of one or more columns that can uniquely identify a record. A super key can contain more columns than necessary to uniquely identify records.
සුපිරි යතුරක් යනු වාර්තාවක් අනන්‍යව හඳුනාගත හැකි තීරු එකක් හෝ කිහිපයක කට්ටලයක් වේ. සුපිරි යතුරක වාර්තා අනන්‍ය ලෙස හඳුනා ගැනීමට අවශ්‍ය ප්‍රමාණයට වඩා තීරු අඩංගු විය හැක.

Domain
වසම

A domain refers to the set of possible values that an attribute (or column) can take in a database table.
වසමක් යනු දත්ත සමුදා වගුවක උපලක්ෂණයක් (හෝ තීරුවක්) සඳහා පැවතිය හැකි අගයන් සමූහයකි.

The domain specifies the data type of the attribute (Ex. INTEGER, VARCHAR, DATE, etc.).
වසම ගුණාංගයේ දත්ත වර්ගය නියම කරයි (උදා. INTEGER, VARCHAR, DATE ආදිය).

It may also restrict the attribute to a specific range or a predefined set of values (Ex. gender can only be 'Male', 'Female', or 'Other').
එය විශේෂිත පරාසයකට හෝ පූර්ව නිශ්චිත අගයන් සමූහයකට (උදා., ස්ත්‍රී පුරුෂ භාවය 'පිරිමි', 'ගැහැණු', හෝ 'වෙනත්' පමණක් විය හැක.)

The domain ensures that all values in a column follow the same rules, helping to maintain data integrity.
තීරුවක ඇති සියලුම අගයන් එකම නීති රීති අනුගමනය කරන බව වසම මගින් සහතික කරයි, එය දත්ත අඛණ්ඩතාව පවත්වා ගැනීමට උපකාරී වේ.

Normalization
ප්‍රමතකරණය

Database normalization is the process of organizing a database into tables and columns in such a way that reduces redundancy and ensures data integrity.
දත්ත සමුදාය ප්‍රමතකරණය යනු අතිරික්ත අඩු කරන සහ දත්ත අඛණ්ඩතාව සහතික කරන ආකාරයට දත්ත සමුදායක් වගු සහ තීරු වලට සංවිධානය කිරීමේ ක්‍රියාවලියයි.

The goal is to eliminate unnecessary duplication of data, reduce potential anomalies, and improve efficiency in terms of storage and retrieval.
අරමුණ වන්නේ දත්තවල අනවශ්‍ය අනුපිටපත් ඉවත් කිරීම, පැවතිය හැකි විෂමතා අවම කිරීම සහ ගබඩා කිරීම සහ නැවත ලබා ගැනීම සම්බන්ධයෙන් කාර්යක්ෂමතාව වැඩි දියුණු කිරීමයි.

Why is normalization essential
ප්‍රමතකරනය අවශ්‍ය වන්නේ ඇයි?

Reduce Data Redundancy
Normalization ensures that data is not duplicated across multiple tables. By dividing data into related tables, we avoid storing the same information in multiple places. This not only saves storage space but also reduces the risk of inconsistency in data.
ප්‍රමතකරණය මඟින් බහු වගු හරහා දත්ත අනුපිටපත් නොවන බව සහතික කරයි. අදාළ වගු වලට දත්ත බෙදීමෙන්, එකම තොරතුරු ස්ථාන කිහිපයක ගබඩා කිරීමෙන් වලකියි. මෙය ගබඩා ඉඩ ඉතිරි කරනවා පමණක් නොව දත්තවල අනනුකූලතාවයේ අවදානමද අඩු කරයි.

Improve Data Integrity
Data integrity refers to the accuracy and consistency of data. When a database is normalized, the risk of anomalies (errors) during insert, update, or delete operations is minimized.
දත්ත අඛණ්ඩතාව යනු දත්තවල නිරවද්‍යතාවය සහ අනුකූලතාවයයි. දත්ත සමුදායක් ප්‍රමතකරණය කළ විට, ඇතුළු කිරීමේ, යාවත්කාලීන කිරීමේ හෝ මකා දැමීමේ මෙහෙයුම් වලදී විෂමතා (දෝෂ) ඇතිවීමේ අවදානම අවම වේ.

Simplify Maintenance
Normalized databases are easier to maintain and modify. As the database grows or changes, adding new data or modifying existing structures is simpler when data is properly normalized, without affecting other unrelated data.
ප්‍රමතකරණය කළ දත්ත සමුදායන් නඩත්තු කිරීමට සහ වෙනස් කිරීමට පහසු වේ. දත්ත සමුදාය වර්ධනය වන විට හෝ වෙනස් වන විට, නව දත්ත එකතු කිරීම හෝ පවතින ව්‍යුහයන් වෙනස් වෙනත් සම්බන්ධයක් නැති දත්තවලට බලපෑම් නොකර සිදු කිරීම දත්ත නිසි ලෙස ප්‍රමතකරණය කළ විට පහසු වේ.

Enhanced Query Efficiency
By breaking down data into smaller tables with well-defined relationships, normalization makes querying data more structured and logical. This can lead to faster and more efficient data retrieval when properly indexed.
හොඳින් අර්ථ දක්වා ඇති සම්බන්ධතා සහිත කුඩා වගු වලට දත්ත බිඳ දැමීමෙන්, ප්‍රමතකරණය මගින් දත්ත විමසීම වඩාත් ව්‍යුහගත සහ තාර්කික කරයි. මෙය නිවැරදිව සුචිගත කළ විට වේගවත් හා වඩා කාර්යක්ෂම දත්ත ලබා ගැනීමට හේතු විය හැක.

Easier to Enforce Data Constraints
Normalization allows you to enforce data integrity constraints more effectively. For example, enforcing unique values for certain fields (like email addresses) becomes easier when data is stored in properly normalized tables.
ප්‍රමතකරණය මඟින් දත්ත අඛණ්ඩතා සීමාවන් වඩාත් ඵලදායී ලෙස බලාත්මක කිරීමට ඔබට ඉඩ සලසයි. උදාහරණයක් ලෙස, දත්ත නිසි ලෙස ප්‍රමතකරනය කර වගු තුළ ගබඩා කර ඇති විට ඇතැම් ක්ෂේත්‍ර සඳහා අනන්‍ය අගයන් බලාත්මක කිරීම (ඊමේල් ලිපින වැනි) පහසු වේ.

Data Duplication
දත්ත අනුපිටපත් වීම

Data duplication refers to the existence of identical copies of data within a database. It can lead to inefficiencies, increased storage costs, and potential inconsistencies in data management.
දත්ත අනුපිටපත් වීම යනු දත්ත සමුදායක් තුළ දත්තවල සමාන පිටපත් පැවතීමයි. එය අකාර්යක්ෂමතාව, ගබඩා කිරීමේ පිරිවැය වැඩි කිරීම සහ දත්ත කළමනාකරණයේ ඇති විය හැකි නොගැලපීම් වලට හේතු විය හැක.

1. Row Level Duplication
පේළි මට්ටමේ අනුපිටපත් වීම

Row level duplication occurs when multiple identical rows exist in a table. This means that all attributes of a particular row are repeated for different entries in the table.
සමාන පේළි කිහිපයක් වගුවක පවතින විට පේළි මට්ටමේ අනුපිටපත් වීම සිදුවේ. මෙයින් අදහස් කරන්නේ වගුවේ විවිධ ඇතුළත් කිරීම් සඳහා යම් පේළියක සියලුම ගුණාංග පුනරාවර්තනය වන බවයි.

Ex.

EID	Name	Department	Salary
001	Akon	HR	50000
001	Akon	HR	50000
002	Bkon	IT	60000
003	Ckon	Finance	70000
001	Akon	HR	50000

In this example, the second and the last row are duplicated; they contain exactly the same values as the first row.
තමම උදාහරණයේ□ දෙවන පේළිය සහ අවසන් පේළිය අනුපිටපත් වී ඇත; ඒවායේ පළමු පේළියට සමාන අගයන් අඩංගු වේ

2. Column Level Duplication
තීරු මට්ටමේ අනුපිටපත් වීම

Column level duplication occurs when the same attribute value is repeated across different records in a table.
තීරු මට්ටමේ අනුපිටපත් වීම සිදු වන්නේ එකම උපලක්ෂණ අගය වගුවක විවිධ පේළි හරහා පුනරාවර්තනය වන විටයි

This is not a complete row duplication but rather a situation where a particular column (or attribute) has redundant data.
තමය සම්පූර්ණ පේළි අනුපිටපතක් නොව යම් තීරුවක (හෝ උපලක්ෂණයක්) අතිරික්ත දත්ත ඇති තත්ත්වයකි

Ex.

SID	Sname	CID	Cname	FacID
1	Akon	C1	S.Eng	Eng
2	Bkon	C2	CS	Comp
3	Ckon	C1	S.Eng	Eng
4	Dkon	C1	S.Eng	Eng
5	Ekon	C3	AIDS	Comp

In this example, CID, Cname, and FacID columns have repeated values in different rows.
මෙම උදාහරණයේදී, CID, Cname, සහ FacID යන තීරුවල විවිධ පේළිවල නැවත නැවත එකම අගයන් ඇත.

Redundancies and Anomalies
පුනර්කරණය සහ විෂමතා

Redundancies in databases occur when the same piece of data is stored in multiple places.
එකම දත්ත කොටසක් ස්ථාන කිහිපයක ගබඩා කර ඇති විට දත්ත සමුදායේ අතිරික්තයන් ඇතිවේ.

This leads to inefficient storage and increases the chances of errors, as changes need to be made in several locations.
මෙය අකාර්යක්ෂම ගබඩා කිරීමට හේතු වන අතර ස්ථාන කිහිපයක වෙනස්කම් සිදු කළ යුතු බැවින් දෝෂ ඇතිවීමේ සම්භාවිතාව වැඩි කරයි.

Normalization aims to eliminate redundancies to ensure that data is stored only once.
දත්ත එක් වරක් පමණක් ගබඩා කර ඇති බව සහතික කිරීම සඳහා අතිරික්තයන් ඉවත් කිරීම ප්‍රමතකරණය මගින් අරමුණු කරයි.

Anomalies are problems that arise due to data redundancy in a poorly structured database.
විෂමතා යනු දුර්වල ව්‍යුහගත දත්ත ගබඩාවක දත්ත අතිරික්තය හේතුවෙන් පැන නගින ගැටළු වේ.

There are three main types of anomalies that occur during data operations: Insert, Update, and Delete anomalies.
දත්ත මෙහෙයුම් වලදී සිදුවන ප්‍රධාන විෂමතා වර්ග තුනක් ඇත: ඇතුල් කිරීමේ, යාවත්කාලීන කිරීමේ සහ මකා දැමීමේ විශමතා.

1. Insert Anomalies

An insert anomaly happens when you cannot insert data into a table without including additional unrelated data.
අමතර අසම්බන්ධ දත්ත ඇතුළත් නොකර වගුවකට දත්ත ඇතුළු කළ නොහැකි විට ඇතුළු කිරීමේ විෂමතාවයක් සිදු වේ.

Ex.

SID	Sname	CID	Cname	FacID
1	Akon	C1	S.Eng	Eng
2	Bkon	C2	CS	Comp
3	Ckon	C1	S.Eng	Eng
4	Dkon	C1	S.Eng	Eng
5	Ekon	C3	AIDS	Comp

1. New student registration

SID	Sname	CID	Cname	FacID
6	Fkon			

If the student hasn't chosen a particular course when registered, other columns remain null.
ලියාපදිංචි වීමේදී ශිෂ්‍යයා නිශ්චිත පාඨමාලාවක් තෝරාගෙන නොමැති නම්, අනෙකුත් තීරු ශූන්‍යව පවතී.

2. New course introduction

SID	Sname	CID	Cname	FacID
		C4	Design	

Neither students nor a faculty might be there when introducing a new course
නව පාඨමාලාවක් හඳුන්වා දීමේදී සිසුන් හෝ පීඨයක් ඒ සඳහා නොපැවතිය හැකිය

3. New faculty

SID	Sname	CID	Cname	FacID
				Med

Neither students nor courses might be there in the newly introduced faculty.
අලුතින් හඳුන්වා දුන් පීඨයේ සිසුන් හෝ පාඨමාලා නොතිබිය හැකිය.

Update Anomalies

An update anomaly occurs when the same piece of data exists in multiple places and all instances are not updated consistently. This happens due to redundancy in the database.
එකම දත්ත කොටස ස්ථාන කිහිපයක පවතින විට සහ සියලු අවස්ථා අඛණ්ඩව යාවත්කාලීන නොවන විට යාවත්කාලීන විෂමතාවයක් ඇතිවේ. දත්ත සමුදායේ අතිරික්තය හේතුවෙන් මෙය සිදු වේ.

Ex.

SID	Sname	CID	Cname	FacID
1	Akon	C1	S.Eng	Eng
2	Bkon	C2	CS	Comp
3	Ckon	C1	S.Eng	Eng
4	Dkon	C1	S.Eng	Eng
5	Ekon	C3	AIDS	Comp

If the FacID of S.Eng is changed from Eng to Comp, three records will have to updated. All three records might not be updated, and even if they do get updated, it will consume more computing power.
S.Eng හි FacID එක Eng සිට Comp දක්වා වෙනස් කළහොත්, පේළි තුනක් යාවත්කාලීන කිරීමට සිදුවේ. වාර්තා තුනම යාවත්කාලීන නොවිය හැකි අතර, ඒවා යාවත්කාලීන වුවද, එමගින් වැඩි පරිගණක බලයක් පරිභෝජනය කරයි.

Delete Anomalies

A delete anomaly occurs when deleting one piece of data inadvertently causes the loss of additional, related data that should not be removed.
මකාදැමීමේ විෂමතාවයක් ඇති වන්නේ එක් දත්ත කොටසක් මකා දැමීමෙන් නොසැලකිලිමත් ලෙස ඉවත් නොකළ යුතු අතිරේක, ඊට සම්බන්ධිත දත්ත නැතිවීම හේතුවෙනි.

Ex.

SID	Sname	CID	Cname	FacID
1	Akon	C1	S.Eng	Eng
2	Bkon	C2	CS	Comp
3	Ckon	C1	S.Eng	Eng
4	Dkon	C1	S.Eng	Eng
5	Ekon	C3	AIDS	Comp

There is only one student who follows the course AIDS.
පාඨමාලාව හදාරන්නේ එක් සිසුවෙක් පමණි.

If that student decides to change the course or leaves the current course, details of the course AIDS will also be changed or updated.
එම ශිෂ්‍යයා පාඨමාලාව වෙනස් කිරීමට තීරණය කළහොත් හෝ දැනට පවතින පාඨමාලාව හැර ගියහොත් AIDS පාඨමාලවෙහි තොරතුරු ද වෙනස් කිරීම හෝ මකා දැමීම සිදුවේ.

How normalization avoids these anomalies
ප්‍රමතකරණය මගින් මෙම විෂමතා මඟ හරින ආකාරය

1. Avoiding Insert Anomalies
ඇතුළත් කිරීමේ විෂමතා වළක්වා ගැනීම

Normalization ensures that data is stored in separate, relevant tables.
ප්‍රමතකරණය මඟින් දත්ත වෙනම, අදාළ වගු වල ගබඩා කර ඇති බව සහතික

This way, new data can be inserted independently without needing irrelevant data.
මේ ආකාරයෙන්, නව දත්ත අදාළ නොවන දත්ත අවශ්‍ය නොවී ස්වාධීනව ඇතුළත් කළ හැකිය.

2. Avoiding Update Anomalies
යාවත්කාලීන විෂමතා වළක්වා ගැනීම

By eliminating redundancy, normalization stores each piece of data in only one place. As a result, when data is updated, you only need to modify it in one table.
අතිරික්තය ඉවත් කිරීමෙන්, ප්‍රමතකරණය මගින් එක් එක් දත්ත කොටස ගබඩා කරන්නේ එක් ස්ථානයක පමණි. එහි ප්‍රතිඵලයක් වශයෙන්, දත්ත යාවත්කාලීන කරන විට, එය එක් වගුවක පමණක් වෙනස් කළ යුතුය.

This ensures consistency and prevents different versions of the same data from existing.
මෙය අනුකූලතාව සහතික කරන අතර එකම දත්තවල විවිධ අනුවාද පවතින ඒවා වළක්වයි.

3. Avoiding Delete Anomalies
මකා දැමීමේ විෂමතා වළක්වා ගැනීම

Normalization separates unrelated data into distinct tables. This prevents the accidental deletion of important related data.
ප්‍රමතකරණය සම්බන්ධිත නොවන දත්ත වෙනස් වගු වලට වෙන් කරයි. මෙය වැදගත් සම්බන්ධ දත්ත අහම්බෙන් මකා දැමීම වළක්වයි.

Functional Dependencies (FD)
කාර්යබද්ධ පරායත්තතා

Functional Dependency (FD) is a fundamental concept in relational database theory that describes a constraint between two attributes (or sets of attributes) in a relation (table).
කාර්යබද්ධ පරායත්තතාවක් යනු සම්බන්ධතාවයක (වගුව) ගුණාංග දෙකක් (හෝ ගුණාංග කට්ටලයක්) අතර සම්බාධකයක් විස්තර කරන සම්බන්ධතා දත්ත සමුදා න්‍යායේ මූලික සංකල්පයකි.

It establishes a relationship where the value of one attribute (or a group of attributes) uniquely determines the value of another attribute.
එය එක් ගුණාංගයක (හෝ ගුණාංග සමූහයක) අගය වෙනත් ගුණාංගයක අගට අනන්‍ය ලෙස තීරණය කරන සම්බන්ධතාවයක් ස්ථාපිත කරයි.

Definition
If A and B are attributes in a relation, we say that B is functionally dependent on A (denoted as A → B) if for each value of A, there is exactly one corresponding value of B. In other words, knowing the value of A allows you to uniquely identify the value of B.
A සහ B යනු සම්බන්ධතාවයක ගුණාංග නම්, A හි එක් එක් අගය සඳහා B හි හරියටම අනුරූප එක් අගයක් තිබේ නම්, B, A මත කාර්යබද්ධව රඳාපවතින බව (A → B ලෙස දක්වනු ලැබේ) කියනු ලැබේ. එසේත් නොමැති නම් A අගය දැන සිටීම B හි අගය අනන්‍ය ලෙස හඳුනා ගැනීමට ඉඩ සලසයි.

Example

```
Student(SID, SName, CID)
```

SID → SName
This means that knowing the Student ID allows us to uniquely determine the Student name.
මෙයින් අදහස් කරන්නේ ශිෂ්‍ය හදුන්වනය දැන ගැනීමෙන් අපට ශිෂ්‍ය නාමය අනන්‍ය ලෙස තීරණය කළ හැකි බවයි.

There are three types of dependencies.
පරායත්තතා වර්ග තුනක් ඇත.

1. Fully Functional Dependencies
පූර්ණ කාර්යබද්ධ පරායත්තතා

2. Partial Dependencies
ආංශික පරායත්තතා

3. Transitive Dependencies
සංක්‍රාන්තික පරායත්තතා

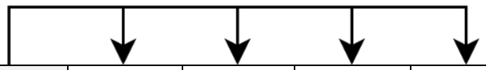
1. Fully Functional Dependencies
පූර්ණ කාර්යබද්ධ පරායත්තතා

Fully Functional Dependency (FFD) occurs in a database table when all the non-prime attributes are fully dependent on the primary key (Whether it's a single field or composite key).
පූර්ණ කාර්යබද්ධ පරායත්තතා දත්ත සමුදා වගුවක ඇති වන්නේ සියලුම ප්‍රාථමික නොවන උපලක්ෂණ ප්‍රාථමික යතුර (එය තනි ක්ෂේත්‍රයක් හෝ සංයුක්ත යතුරක් වුවද) මත පූර්ණ ලෙස රඳා පවතින විටදීය.

```
Table(A, B, C, D, E)
```

A → {B, C, D, E}
This means that knowing the value of A allows us to determine that values of all B, C, D, and E.
මෙයින් අදහස් කරන්නේ A හි අගය දැන ගැනීමෙන් B, C, D, E යන සියලුම අගයන් තීරණය කිරීමට අපට ඉඩ සලසන බවයි.

Ex.



NIC	Name	Adrs.	Phone	Gender
1	Akon	St.1	071	M
2	Bkon	St.2	072	F
3	Ckon	St.3	074	F
4	Dkon	St.4	077	M
5	Ekon	St.5	078	F

All the non-prime attributes name, address, phone and gender are fully dependent on the primary key NIC.
සියලුම ප්‍රාථමික නොවන උපලක්ෂණ වන name, address, phone සහ gender භාවය සම්පූර්ණයෙන්ම ප්‍රාථමික යතුර වන NIC මත රඳා පවතී.

This means that knowing a particular NIC allows us to know all other details of that person.
මෙයින් අදහස් කරන්නේ යම් NIC අගයක් දැන ගැනීමෙන් එම පුද්ගලයාගේ අනෙකුත් සියලුම තොරතුරු දැන ගැනීමට අපට ඉඩ සලසන බවයි.

2. Partial Dependencies
ආංශික පරායත්තතා

Partial dependency occurs when a non-prime attribute is dependent on only a part of a composite primary key, rather than the entire composite key.
ප්‍රාථමික නොවන උපලක්ෂණයක් සම්පූර්ණ සංයුක්ත ප්‍රාථමික යතුර වෙනුවට සංයුක්ත ප්‍රාථමික යතුරක කොටසක් මත පමණක් රඳා පවතින විට ආංශික පරායත්තතාව ඇති වේ.

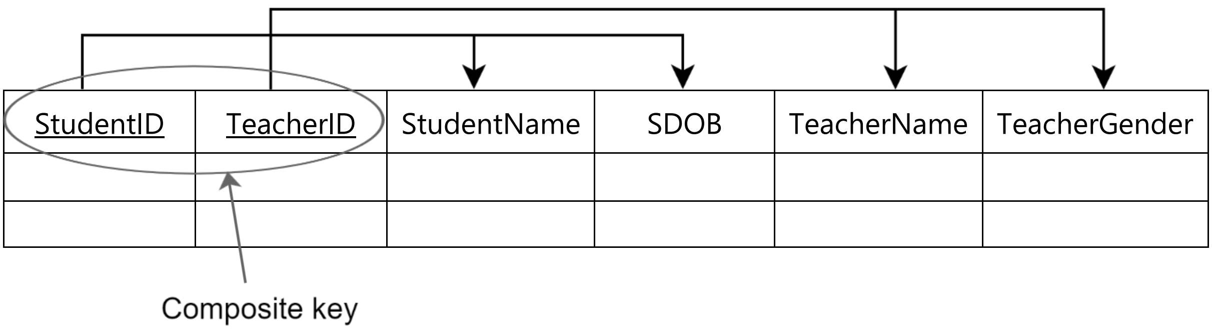
Note:
This occurs only in tables where the primary key consists of more than one attribute (a composite key.
ප්‍රාථමික යතුර සඳහා එක් උපලක්ෂණයකට වඩා (සංයුක්ත යතුරක්) සමන්විත වන වගු වලදී පමණක් මෙය සිදු වේ.

```
Table(A, B, C, D, E, F)
```

A → {C, D}
B → {E, F}
C and D can uniquely be Identified by using only A which is a part of the primary key.
ප්‍රාථමික යතුරේ කොටසක් වන A පමණක් භාවිතා කිරීමෙන් C සහ D අනන්‍යව හඳුනාගත හැක.

E and F can uniquely be Identified by only B which is the other part of the primary key.
ප්‍රාථමික යතුරේ අනෙත් කොටස වන B පමණක් භාවිතා කිරීමෙන් E සහ F අනන්‍යව හඳුනාගත හැක.

Ex.



`ST(StudentID, TeacherID, StudentName, SDOB, TeacherName, TeacherGender)`

Fully Functional Dependency

{StudentID, TeacherID} → {StudentName, SDOB, TeacherName, TeacherGender}

Partial Dependencies

{StudentID} → {StudentName, SDOB}

{TeacherID} → {TeacherName, TeacherGender}

Here, StudentName and SDOB are dependent on a part of the primary key which is StudentID.
මෙහිදී, StudentName සහ SDOB ප්‍රාථමික යතුරේ කොටසක් වන StudentID මත රඳා පවතී.

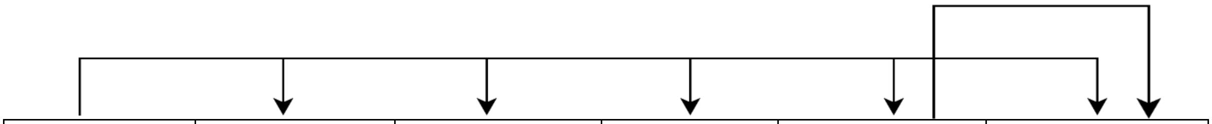
TeacherName and TeacherGender are dependent on the other part the primary key which is TeacherID.
TeacherName සහ TeacherGender ප්‍රාථමික යතුරෙහි අනෙක් කොටස වන TeacherID මත රඳා පවතී.

Thus, non-prime attributes being dependent on parts of the primary key is known as partial dependency.
මෙලෙස ප්‍රාථමික නොවන උපලක්ෂණ ප්‍රාථමික යතුරේ කොටස් මත රඳා පැවතීම ආංශික පරායත්තතාව ලෙස හැඳින්වේ.

Note:
Here, when the prime attributes (Composite key) change, the non-prime attributes will also change accordingly. Therefore, it is a fully functional dependency.
මෙහිදී ප්‍රථමික උපලක්ෂණ (සංයුක්ත යතුර) වෙනස් වන විට, ප්‍රාථමික නොවන උපලක්ෂණ ද ඒ අනුව වෙනස් වේ. එබැවින්, එය පූර්ණ කාර්යබද්ධ පරායත්තතාවකි.

But all the prime attributes don’t have to change in order for the non-prime attributes to change. In other words, when a part of prime attributes changes, some of the non-attributes will also change. Therefore, it is a partial dependency.
නමුත් ප්‍රාථමික නොවන උපලක්ෂණ වෙනස් වීම සඳහා සියලුම ප්‍රාථමික උපලක්ෂණ වෙනස් විය යුතු නැත. වෙනත් වචන වලින් කිවහොත්, ප්‍රාථමික උපලක්ෂණවල කොටසක් වෙනස් වූ විට, සමහර ප්‍රාථමික නොවන ගුණාංග ද වෙනස් වේ. එබැවින්, එය ආංශික පරායත්තතාවකි.

3. Transitive Dependencies
සංක්‍රාන්තික පරායත්තතා



VehicleCode	VehicleName	VehicleYear	Color	CountryCode	Country

Transitive Dependency occurs when a non-prime attribute depends on another non-prime attribute.
ප්‍රාථමික නොවන උපලක්ෂණයක් තවත් ප්‍රාථමික නොවන උපලක්ෂණයක් මත රඳා පවතින විට සංක්‍රාන්ති පරායත්තතාවක් ඇතිවේ.

In simpler terms, it's when one non-key attribute depends on another non-key attribute rather than directly depending on the primary key.
එනම් සරලව, ප්‍රාථමික යතුර මත සෘජුවම රඳා පවතිනවා වෙනුවට එක් යතුරු-නොවන උපලක්ෂණයක් තවත් යතුරු නොවන උපලක්ෂණයක් මත රඳා පවතින විට මෙය සිදුවේ.

```
Table(A, B, C)
```

FFD
{A} → {B,C}

Transitive Dependency
{B} → {C}

In this example, VehicleCode is the primary key and all the other non-key attributes are fully functionally dependent on it.
මෙම උදාහරණයේ දී, VehicleCode ප්‍රාථමික යතුර වන අතර අනෙකුත් සියලුම යතුරු නොවන උපලක්ෂණ පූර්ණ කාර්යබද්ධ ලෙස එය මත රඳා පවතී.

But if know a particular Country code, we can uniquely identify the corresponding country.
නමුත් යම් Country code එකක් දන්නේ නම්, අනුරූප රට අනන්‍ය ලෙස හඳුනාගත හැකිය.

Therefore, in the above table, Country is dependent on CountryCode.
එබැවින්, ඉහත වගුවේ, Country එක, CountryCode මත රඳා පවතී.

This dependency is known as transitive dependency.
මෙම පරායත්තතාවය සංක්‍රාන්ති පරායත්තතාවය ලෙස හැඳින්වේ

Summary

Prime attributes

Prime attributes refer to the attributes that are part of the primary key.
යතුරු උපලක්ෂණ යනු ප්‍රාථමික යතුරේ කොටස් වන උපලක්ෂණයි.

Non-prime attributes

Non-prime attributes are those that are not part of the primary key.
යතුරු නොවන උපලක්ෂණ යනු ප්‍රාථමික යතුරේ කොටසක් නොවන උපලක්ෂණ වේ.

Fully Functional Dependency

A non-prime attribute is fully functionally dependent on the primary key if it depends on all prime attributes.
යතුරු නොවන උපලක්ෂණයක් සියලුම යතුරු උපලක්ෂණ මත රඳා පවතී නම් එය ප්‍රාථමික යතුර මත පූර්ණ කාර්යබද්ධ පරායත්ත වේ.

Partial Dependency

A partial dependency occurs when a non-prime attribute is dependent on a part of prime attributes.
යතුරු නොවන උපලක්ෂණයක් යතුරු උපලක්ෂණ වල කොටසක් මත රඳා පවතින විට ආංශික පරායත්තතාවයක් ඇතිවේ.

Transitive Dependency

A transitive dependency happens when a non-prime attribute depends on another non-prime attribute.
යතුරු නොවන උපලක්ෂණයක් වෙනත් යතුරු නොවන උපලක්ෂණයක් මත රඳා පවතින විට සංක්‍රාන්ති පරායත්තතාවයක් ඇති වේ.

Levels of Normalization
ප්‍රමතකරණයේ මට්ටම්

1. Zero Normal Form
   ශූන්‍ය ප්‍රමත අවස්ථාව
2. First Normal Form
   පළමු ප්‍රමත අවස්ථාව
3. Second Normal Form
   දෙවන ප්‍රමත අවස්ථාව
4. Third Normal Form
   තෙවන ප්‍රමත අවස්ථාව

Zero Normal Form (0NF)
ශූන්‍ය ප්‍රමත අවස්ථාව

0NF refers to a stage where no normalization rules have been applied.
0NF යනු ප්‍රමතකරණ නීති කිසිවක් යොදවා නොමැති අදියරකි.

Data is unorganized and may be stored in a single large table, possibly with repeating groups or nested data.
දත්ත අසංවිධානාත්මක වන අතර තනි විශාල වගුවක, සමහරවිට පුනරාවර්තන කාණ්ඩ හෝ කැදලි දත්ත සමඟ ගබඩා වී ඇත.

The same data may be repeated multiple times, leading to inefficiency and storage issues.
එකම දත්ත කිහිප වතාවක් පුනරාවර්තනය විය හැකි අතර, අකාර්යක්ෂමතාවය සහ ගබඩා කිරීමේ ගැටළු වලට මග පාදයි.

Insert, update, and delete anomalies are common, making data maintenance difficult.
ඇතුළු කිරීමේ, යාවත්කාලීන කිරීමේ සහ මකා දැමීමේ විශමතා පොදු වන අතර, දත්ත නඩත්තු කිරීම අපහසු වේ.

Example of 0NF

SID	Name	SubName
101	Akon	Maths
102	Bkon	Maths, ICT
103	Ckon	Physics, BIO
104	Dkon	ICT, Accounts

First Normal Form (1NF)
පළමු ප්‍රමත අවස්ථාව

Rules for 1NF
1NF සඳහා වන නීති

- **Atomic values**
  Each column must contain indivisible values, meaning no lists or sets are allowed within a single cell.
  සෑම තීරුවකම බෙදිය නොහැකි අගයන් අඩංගු විය යුතුය, එනම් තනි කෝෂයක් තුළ ලැයිස්තු හෝ කුලකවලට ඉඩ නොදේ.

- **Unique rows**
  Every record in a table must be unique; there should be no duplicate rows.
  වගුවක ඇති සෑම පේළියක්ම අනන්‍ය විය යුතුය; අනුපිටපත් පේළි නොතිබිය යුතුය.

- **Single-valued columns**
  Each column must contain a single value per row, so no arrays, lists, or multiple entries within a single field.
  සෑම තීරුවකම පේළියකට තනි අගයක් අඩංගු විය යුතුය, එබැවින් තනි ක්ෂේත්‍රයක් තුළ අරා, ලැයිස්තු හෝ බහු ඇතුළත් කිරීම් නොමැත.

- **Consistent column types**
  Each column must have a consistent data type across all rows.
  සෑම තීරුවකම සියලුම පේළි හරහා ස්ථාවර දත්ත වර්ගයක් තිබිය යුතුය.

Converting the 0NF table into 1NF.
0NF වගුව 1NF බවට පරිවර්තනය කිරීම.

SID	Name	SubName
101	Akon	Maths
102	Bkon	Maths, ICT
103	Ckon	Physics, BIO
104	Dkon	ICT, Accounts

Solution 1

SID	Name	SubName
101	Akon	Maths
102	Bkon	Maths
102	Bkon	ICT
103	Ckon	Physics
103	Ckon	BIO
104	Dkon	ICT
104	Dkon	Accounts

Primary key = SID + SubName

This approach coverts the table to 1NF. But there are too many repetitions of the same value.
මෙම ප්‍රවේශය 1NF වෙත වගුව පරිවර්තනය කරයි. නමුත් එකම අගයේ පුනරාවර්තන විශාල ප්‍රමාණයක් තිබේ.

Solution 2

SID	Name	Sub1	Sub2	.....
101	Akon	Maths	Null	Null
102	Bkon	Maths	ICT	Null
103	Ckon	Physics	BIO	Null
104	Dkon	ICT	Accounts	Null

There are too many Null values in the table if this approach is followed.
මෙම ප්‍රවේශය අනුගමනය කළහොත් වගුවේ බොහෝ Null අගයන් ඇතිවේ.

Solution 3

Student

PK

SID	Name
101	Akon
102	Bkon
103	Ckon
104	Dkon

Subject

PK

SubID	SubName
1	Maths
2	Physics
3	BIO
4	Accounts
5	ICT

StudentSubject

FK FK

SID	SubID
101	1
102	1
102	5
103	2
103	3
104	4
104	5

Second Normal Form
දෙවන ප්‍රමත අවස්ථාව

Rules for 2NF
2NF සඳහා වන නීති

The table must already be in 1NF.
වගුව දැනටමත් 1NF හි තිබිය යුතුය.

No partial dependency
Non-key attributes must depend on the entire composite primary key, not just a subset of it.
යතුරු-නොවනෙ උපලක්ෂණ ප්‍රාථමික යතුරේ උප කුලකයක් මත නොව සම්පූර්ණ සංයුක්ත ප්‍රාථමික යතුර මත රඳා පැවතිය යුතුය.

(This rule only applies to tables with composite keys.)
(මෙම රීතිය අදාළ වන්නේ සංයුක්ත යතුරු සහිත වගු සඳහා පමණි.)

Note:
If the table is already in 1NF and has only one attribute for the primary key, that table is in 2NF.
වගුව දැනටමත් 1NF හි තිබේ නම් සහ ප්‍රාථමික යතුර සඳහා එක් උපලක්ෂණයක් පමණක් තිබේ නම්, එම වගුව 2NF හි ඇත.

SID	CID	SName	CName	Grade
101	1	Akon	S.Eng	A
102	2	Bkon	CS	A
102	3	Bkon	AIDS	C
103	4	Dkon	Archt.	B
104	4	Ekon	Archt	A

This table is in 1NF. But it is not in 2NF because it contains partial dependencies.
මෙම වගුව 1NF හි ඇත. නමුත් එය 2NF හි නොමැත, මන්ද එහි ආංශික පරායත්තතා අඩංගු වේ.

Converting the 1NF table into 2NF.
1NF වගුව 2NF බවට පරිවර්තනය කිරීම.

SID	CID	SName	CName	Grade
101	1	Akon	S.Eng	A
102	2	Bkon	CS	A
102	3	Bkon	AIDS	C
103	4	Dkon	Archt.	B
104	4	Ekon	Archt	A

Solution

PK

Student

SID	SName
101	Akon
102	Bkon
103	Dkon
104	Ekon

PK

Course

CID	SName
1	S.Eng
2	CS
3	AIDS
4	Archt

FK

StudentCourse

SID	CID	Grade
101	1	A
102	2	A
102	3	C
103	4	B
104	4	A

FK

Third Normal Form (3NF)
තෙවන ප්‍රමත අවස්ථාව

Rules for 3NF
2NF සඳහා වන නීති

The table must already be in 2NF.
වගුව දැනටමත 2NF හි තිබිය යුතුය.

No transitive dependency
If a non-key attribute depends on another non-key attribute, it must be removed by creating a new table.
යතුරු නොවන උපලක්ෂණයක් වෙනත් යතුරු නොවන උපලක්ෂණයක් මත රඳා පවතී නම්, එය නව වගුවක් නිර්මාණය කිරීමෙන් ඉවත් කළ යුතුය.

SID	SName	FacID	FacName
101	Akon	1	Engineering
102	Bkon	2	Medical
103	Ckon	3	Computing
104	Dkon	3	Computing

This table has one attribute for the primary key and is in 1NF. Therefore, this table is in 2NF.
මෙම වගුවේ ප්‍රාථමික යතුර සඳහා එක් උපලක්ෂණයක් ඇති අතර එය 1NF හි ඇත. එබැවින්, මෙම වගුව 2NF හි ඇත.

But the faculty name can be uniquely identified by the faculty ID.
නමුත් එක මගින් එක අනන්‍ය ලෙස හඳුනාගත හැකිය.

This is a transitive dependency and therefore, this table is not in 3NF.
මෙය සංක්‍රාන්ති පරායත්තතාවයක් වන අතර එබැවින්, මෙම වගුව 3NF හි නොමැත

Converting the 1NF table into 2NF.
1NF වගුව 2NF බවට පරිවර්තනය කිරීම.

Solution 1 (The best approach)

StudentFaculty

<u>SID</u>	SName	FacID
101	Akon	1
102	Bkon	2
103	Ckon	3
104	Dkon	3

FK

PK

Faculty

<u>FacID</u>	FacName
1	Engineering
2	Medical
3	Computing

Solution 2 (Not the best approach)

PK

Student

<u>SID</u>	SName
101	Akon
102	Bkon
103	Ckon

PK

Faculty

<u>FacID</u>	FacName
1	Engineering
2	Medical
3	Law
4	Computing

FK

StudentFaculty

FK

<u>SID</u>	<u>FacID</u>
101	1
102	2
103	3
104	3

Summary

Anomaly	**0NF**	**1NF**	**2NF**	**3NF**
Multi-Valued dependency	✓	✗	✗	✗
Partial dependency	✓	✓	✗	✗
Transitive dependency	✓	✓	✓	✗