DATABASE SYSTEM

FUNDAMENTALS & DESIGN

Volume 1

WAN AZLINIE | SALMIZA | SAIFUL AZIZI

KEMENTERIAN PENDIDIKAN TINGGI

POLITEKNIK MALAYSIA
SULTAN MIZAN ZAINAL ABIDIN

DATABASE SYSTEM

FUNDAMENTALS & DESIGN

(VOLUME 1)

Wan Azlinie binti Wan Ahmad
Salmiza binti Said
Saiful Azizi bin Abdullah

DEPARTMENT OF ELECTRICAL ENGINEERING
POLITEKNIK SULTAN MIZAN ZAINAL ABIDIN
2023

DATABASE SYSTEM FUNDAMENTALS & DESIGN (VOLUME 1)
First Edition 2023

Writer:
Wan Azlinie binti Wan Ahmad
Salmiza binti Said
Saiful Azizi bin Abdullah

Editor:
Wan Azlinie binti Wan Ahmad

Published by:
Department of Electrical Engineering
Politeknik Sultan Mizan Zainal Abidin
KM. 08, Jalan Paka,
23000 Dungun, Terengganu
☎ 09-8400869

PREFACE

Database systems are widely used in various industries and applications, including business management, e-commerce, finance, healthcare, and more. It is a software application specifically designed to efficiently store, manage, and retrieve data. Essentially, a database system consists of two main components: the database itself, which is a collection of related data, and the database management system (DBMS), which is the software that controls and interacts with the database.

The core components of a database system are tables, relationships, and queries. Tables are the fundamental structure where data is arranged in rows and columns, representing entities and their corresponding attributes. Relationships establish links between tables using shared attributes, facilitating streamlined querying and efficient retrieval of data from interconnected tables. Users can extract precise information by searching, filtering, and sorting data based on specific criteria by utilizing query languages like SQL. These core components play a crucial role in designing an effective and functional database system.

This book is designed to facilitate students' understanding of the basic concepts in designing a Database System. It is organized based on the curriculum of Malaysian polytechnics, specifically the Database System course (DEC40073) for Diploma in Electronic Engineering (Computer) under the Department of Electrical Engineering at Politeknik Sultan Mizan Zainal Abidin, Dungun, Terengganu.

The authors would like to express their deepest appreciation to all those involved in the publication of this book. It is hoped that this book will be advantageous in aiding students to comprehend the fundamental concepts related to constructing a database system, aligning with the requirements outlined in the course syllabus.

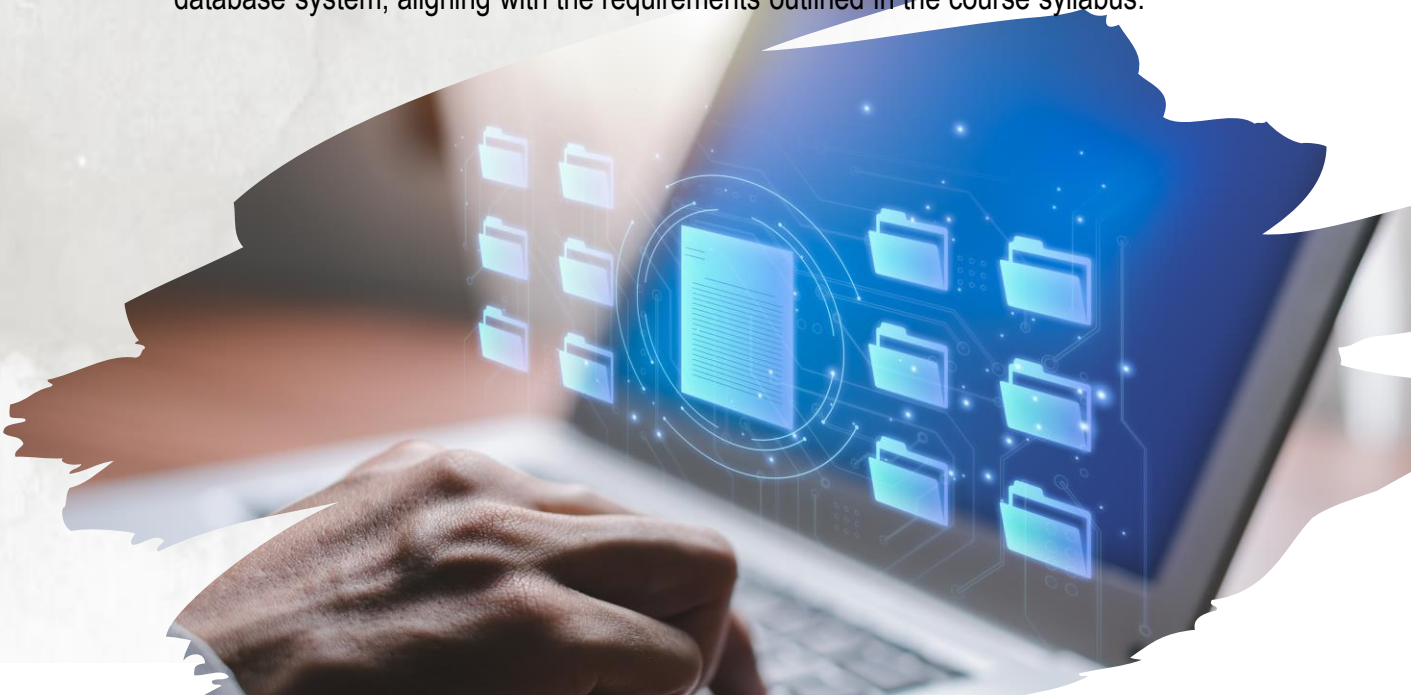


TABLE OF CONTENTS

1

DATABASE CONCEPT

Introduction

On a daily basis, we consistently produce and engage with data, encompassing personal information like names, addresses, and contact details, as well as health-related metrics such as daily step counts, heart rates, and sleep patterns. The significance of data and databases in our routine is profound, significantly influencing how we engage with information and technology.

A database is a structured collection of data designed for efficient retrieval and management. Utilizing a database ensures data accuracy, reliability, and ease of access.

Whether you're engaged in banking, scientific research, government record-keeping, or simply browsing popular websites like Amazon, YouTube, Netflix, or Wikipedia, databases play a crucial role in powering these systems.

Beyond large-scale applications, databases are also valuable for small businesses and groups, making information management more streamlined. Their ubiquity stems from the fact that databases simplify the process of accessing information through computers.

.

DATA

Known facts that can be recorded and **have an implicit meaning**.

Raw facts, that have not yet been processed to reveal their meaning to the end user.

Data represents facts of people, places, objects, entities, events or concepts.

It can be in variety of format including text, numbers, media, bytes, etc.

Name : SITI
Gender : Female
Reg. No : AK992015
Program : DTK
Quiz 1 : 10/10
Quiz 2 : 5/10
Department : JKE
Phone No : 011788288

VS

INFORMATION

Facts (data) that are **arranged in meaningful patterns**.

A processed and organized data presented in a meaningful context.

Student's Info

Name	Gender	RegNo	Prog	Dept	Phone
SITI	Female	AK992015	DTK	JKE	011788288

Assessment Sheet

Name	RegNo	Quiz1	Quiz2	Average
SITI	AK992015	10	5	7.5

what is DATABASE?

A **collection of related data**.

Designed to **meet the information needs** of an organization

Is **organized** so that it can easily be accessed, managed, and updated.

The main goal of the database is to **handle a large amount of information**.

DATABASE SYSTEM

A **database system** is a structured and organized collection of data that is managed and accessed electronically.

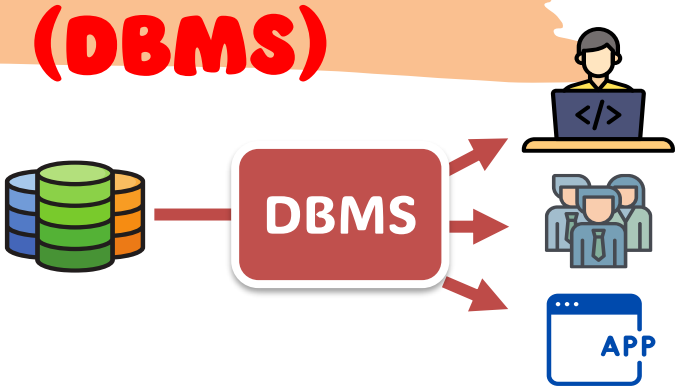
It consists of a database, which stores the data, and a database management system (DBMS), which facilitates the management and retrieval of that data.

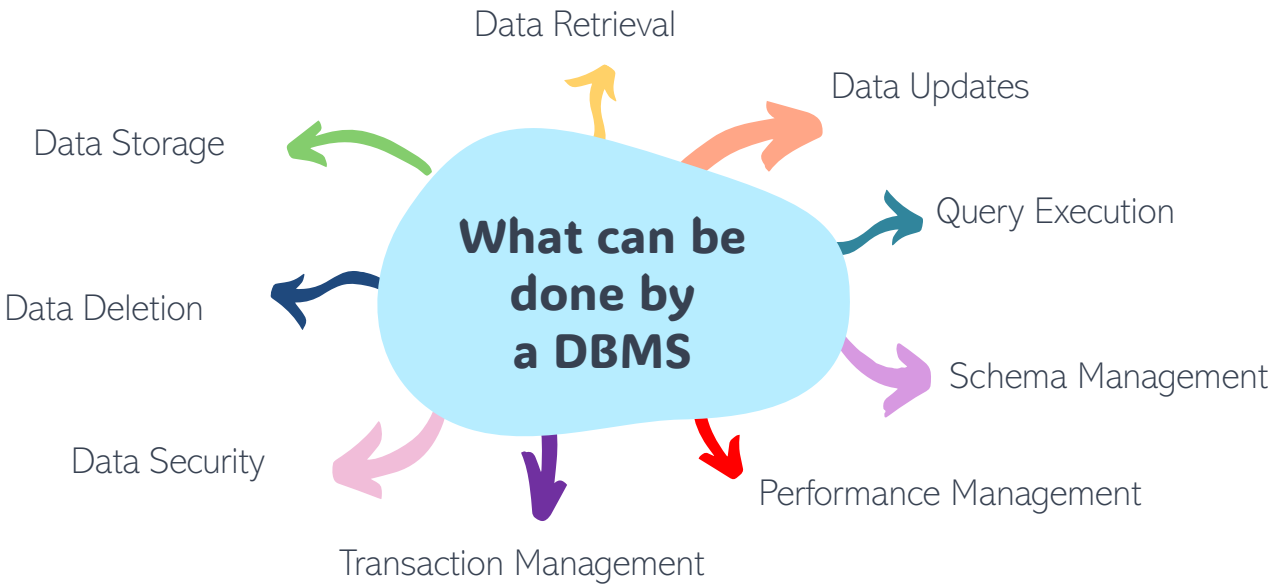
The DBMS ensures data integrity, security, and efficient querying.

Example:
eCommerce platforms, healthcare systems, social media platforms, online banking systems, email services, and online learning platforms.

DATABASE MANAGEMENT SYSTEM (DBMS)

A software package to facilitate the creation and maintenance of a computerized database.





Example of DBMS

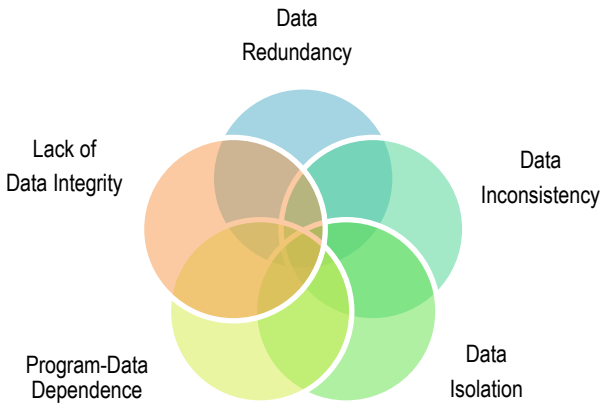
FILE BASED SYSTEM

A file-based system is a **method of managing and storing data on a computer that relies on files and directories.**

In this system, data is typically stored in **separate files**, and each file may contain records or data related to a specific application.

Each application has its own set of files, and there is often redundancy in data storage.

The **weakness** of using this methods:



File Based System Environment

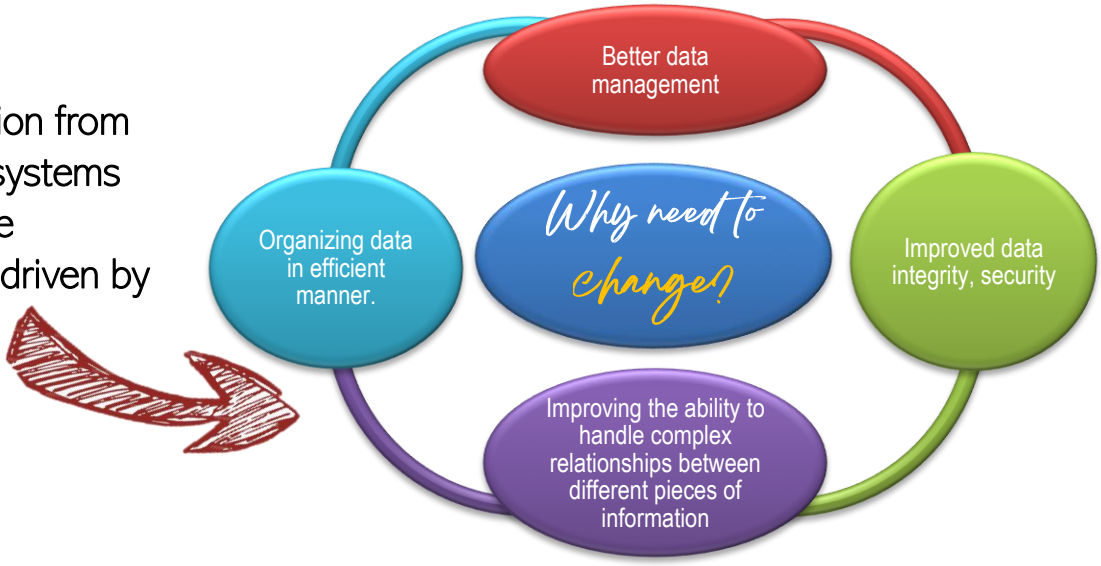
STUDENT AFFAIR DEPARTMENT

COURSE ENROLLMENT UNIT

EXAMINATION UNIT

- Student's enrolment
- Course
- Course Registration
- Exam Paper
- Student's Result

The transition from file-based systems to database systems is driven by the need:



HISTORY OF DATABASE

Everything had to be recorded on paper (manual record)

File-Based Systems

Database System

TYPES OF DATABASE SYSTEMS

01 **Personal Databases**

- Support single user
- PCs, laptops, personal digital assistants (PDA), mobile phones

02 **Workgroup Databases**

- Workgroups –typically less than 25 people collaborating on the same project or application
- Typically stored on a server and access is via a (LAN)

03 **Department Databases**

- Support functional units of an organization – sales, personnel, marketing, manufacturing, accounting
- Large numbers of users often in geographically dispersed locations

04 **Enterprise Databases**

- Covers an entire organization (or at least many different departments!)
- Support organization-wide operations, decision making

DATA MODEL SCHEMA

The **description of a database**.

Includes descriptions of the **database structure, data types**, and the **constraints** on the database.

A component of the schema or an object within the schema, e.g., STUDENT, COURSE.

The database schema **changes very infrequently**.

Schema is also called **intension**.

DATA MODEL INSTANCE

The **actual data stored in a database at a particular moment in time**. This includes the collection of all the data in the database.

Refers to the content of a database at a moment in time.

Also called database, record instance, table instance or entity instance

The database instance **changes every time the database is updated**.

Instance is also called **extension**.

Example of Data Model Scheme

STUDENT

#	Name	Type	Collation	Attributes	Null	Default
1	REG_NO	varchar(12)	latin1_swedish_ci		No	
2	STUD_NAME	varchar(200)	latin1_swedish_ci		Yes	*NULL*
3	PROGRAM	char(15)	latin1_swedish_ci		Yes	*NULL*
4	DEPT_CODE	varchar(5)	latin1_swedish_ci		Yes	*NULL*
5	AGE	int(11)			Yes	*NULL*

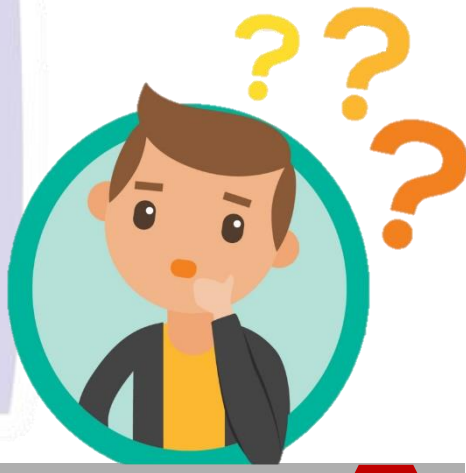
Example of Data Model Instance

STUDENT

REG_NO	STUD_NAME	PROGRAM	DEPT_CODE	AGE
DDTF1001	MAY	DDT	JTMK	19
DEMF1002	RAJU	DEM	JKM	21
DKAF1008	DAUD	DKA	JKA	20
DTKF1001	ALI	DTK	JKE	20
DTKF1010	MEENA	DTK	JKE	19

DIFFERENCES BETWEEN DATABASE APPROACH & TRADITIONAL FILE PROCESSING

Traditional File Processing	Database Processing
Information is entered many times in many formats due to the output, which is required.	Information is only entered once, and it can be executed many times in several formats.
Takes time to manipulate data.	Gathers, stores and manipulates data quickly
Takes time to execute complex calculations.	Easy to manage complex calculations.
A lot of data redundancies	Reduced data redundancy.
Data can only be used by one user at one time (single user).	Data can be used by many users at one time (multiuser).
The file may contain data that is contrary to the actual data if the last user is not updating.	There is software that used to control the concurrency in the process of data updates by users, in DBMS.
Data cannot be controlled by the verified user because there is no security performed.	In database system, DBMS can control the data from being used by unverified user.
Waste of space (cupboard, rack) if there is repeated data.	No data repetition, so less disk space needed.



2

DATABASE LANGUAGE

Introduction

There are several database languages used for interacting with databases. SQL (Structured Query Language) encompasses various commands and statements for interacting with relational databases.

The two main categories of SQL operations are:

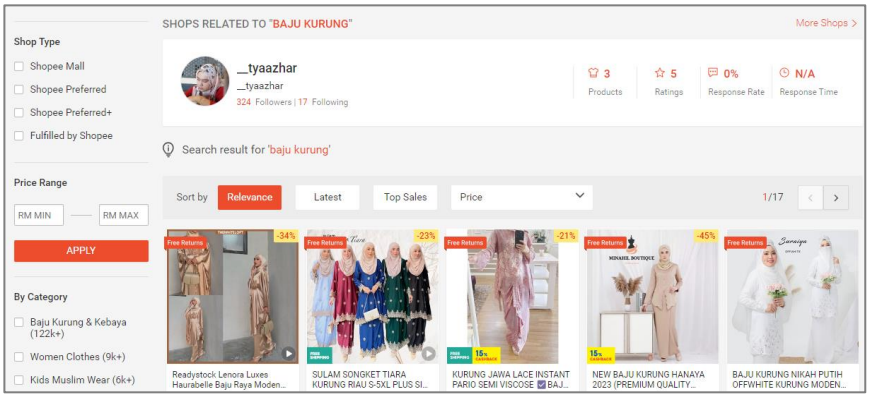
- Data Definition Language (DDL) — 01
- Data Manipulation Language (DML) — 02

CONCEPT OF SQL

The user specifies a certain condition **(query).**

The program will go through all the records in the database file and select those records that satisfy the condition **(searching).**

The result of the query will then be stored in form of a table.



Example of query on online shopping application.

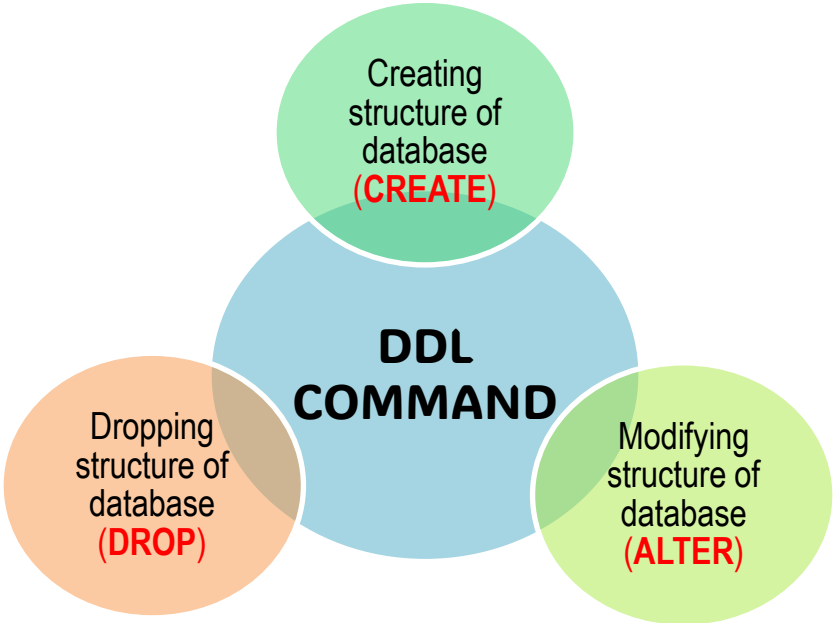
DATA DEFINITION LANGUAGE (DDL)

- Description language.
- A special language used to **specify a database conceptual scheme** using a set of definitions.
- Define or modify the schema not manipulate data. For example, the STUDENT table.

STUDENT

#	Name	Type	Collation	Attributes	Null	Default
1	REG_NO	varchar(12)	latin1_swedish_ci		No	
2	STUD_NAME	varchar(200)	latin1_swedish_ci		Yes	*NULL*
3	PROGRAM	char(15)	latin1_swedish_ci		Yes	*NULL*
4	DEPT_CODE	varchar(5)	latin1_swedish_ci		Yes	*NULL*
5	AGE	int(11)			Yes	*NULL*

- Supports the definition or declaration of database objects.
- Associated integrity and security constraints.
- Allows the Database Administrator (DBA) or user to **describe and name** the entities, attributes and relationships required for the application, together with any associated integrity and security constructions.



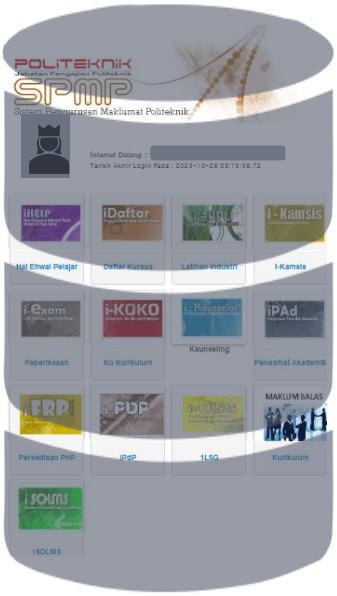


TABLE CREATION

Syntax:

```
CREATE TABLE relation-name (
attribute-name1 domain1 not null unique,
attribute-name2 domain2,
…
attribute-namen domain),
PRIMARY KEY (attribute-name1))
```

1. Write the SQL command to create table **STUDENT.** Assume REG_NO as a Primary Key.

```
CREATE TABLE STUDENT
( REG_NO varchar(10) not null unique,
  STUD_NAME varchar(200),
  PROGRAM varchar(20),
  DEPARTMENT varchar(4),
  AGE integer,
  PRIMARY KEY (REG_NO) )
```

Output

STUDENT

REG_NO	STUD_NAME	PROGRAM	DEPARTMENT	AGE

TABLE DELETION

Syntax:

```
DROP TABLE relation_name;
```

1. Write the SQL command to delete STUDENT table from the database.

```
DROP TABLE STUDENT;
```

Output

STUDENT

REG_NO	STUD_NAME	PROGRAM	DEPARTMENT	AGE

REFERENCE TO ANOTHER TABLE

Syntax:

```
CREATE TABLE relation-name (
attribute-name1 domain1 not null
  unique,
attribute-name2 domain2 REFERENCES
  reference-table(reference-attribute)
  ,
…
attribute-namen domain),
PRIMARY KEY (attribute-name1))
```

1. Write the SQL command to create table **STUDENT**. Assume **DEPARTMENT** in Student's Table as a Foreign Key.

```
CREATE TABLE STUDENT
( REG_NO varchar(10) not null unique,
  STUD_NAME varchar(200),
  PROGRAM varchar(20),
  DEPARTMENT varchar(4) REFERENCES DEPARTMENT(DEPT_NAME),
  AGE integer,
  PRIMARY KEY (REG_NO) )
```

STUDENT

REG_NO	STUD_NAME	PROGRAM	DEPARTMENT	AGE
DTKF1001	ALI	DTK	JKE	20
DTKF1010	MEENA	DTK	JKE	19
DDTF1001	MAY	DDT	JTMK	19
DEMF1002	RAJU	DEM	JKM	21
DKAF1008	DAUD	DKA	JKA	20

Attention:
DEPT_CODE must be the primary key of **DEPARTMENT table**

DEPARTMENT

DEPT_CODE	DEPT_NAME
JKE	JAB. KEJ. ELEKTRIK
JKM	JAB. KEJ. MEKANIKAL
JKA	JAB. KEJ. AWAM
JTMK	JAB. TEKNOLOGI MAKLUMAT & KOMUNIKASI

TABLE ALTERATION

1. **ALTER TABLE - ADD**
   To add new attribute (column) to the existing table.
2. **ALTER TABLE – DROP**
   To delete attribute from the table.
3. **ALTER TABLE - ALTER/MODIFY**
   To modify the attribute (e.g.: name, domain, field size) in the existing table

ALTER TABLE - ADD

Syntax:

```
ALTER TABLE relation_name
ADD Attribute Domain
```

1. Write the SQL command **to add a gender** to the STUDENT.

References

STUDENT

REG_NO	STUD_NAME	PROGRAM	DEPARTMENT	AGE

```
ALTER TABLE STUDENT
ADD GENDER varchar (2)
```

Output

STUDENT

REG_NO	STUD_NAME	PROGRAM	DEPARTMENT	AGE	GENDER

ALTER TABLE - DROP

Syntax:

```
ALTER TABLE relation_name
DROP Attribute
```

1. Write the SQL command to **delete a gender** from the STUDENT table.

```
ALTER TABLE STUDENT
DROP GENDER
```

Output

STUDENT

REG_NO	STUD_NAME	PROGRAM	DEPARTMENT	AGE	GENDER

ALTER TABLE - CHANGE

Syntax:

Rename the attribute's name

```
ALTER TABLE relation_name
CHANGE OldAttribute NewAttribute Domain
```

1. Write the SQL command to alter the Department column into DeptCode in the STUDENT table.

References

STUDENT

#	Name	Type
1	REG_NO	varchar(12)
2	STUD_NAME	varchar(200)
3	PROGRAM	varchar(3)
4	DEPARTMENT	varchar(5)

Output

Name	Type
REG_NO	varchar(12)
STUD_NAME	varchar(200)
PROGRAM	char(15)
DEPT_CODE	varchar(5)

```
ALTER TABLE STUDENT
CHANGE DEPARTMENT DEPTCODE varchar(5)
```

ALTER TABLE - MODIFY

Syntax:

Change the attribute's data type (domain)

```
ALTER TABLE relation_name
MODIFY COLUMN Attribute NewDomain
```

1. Write the SQL command to Change the data type of the column Program in the STUDENT table from varchar to char.

References

STUDENT

#	Name	Type
1	REG_NO	varchar(12)
2	STUD_NAME	varchar(200)
3	PROGRAM	varchar(3)
4	DEPARTMENT	varchar(5)

Output

Name	Type
REG_NO	varchar(12)
STUD_NAME	varchar(200)
PROGRAM	char(5)
DEPT_CODE	varchar(5)

```
ALTER TABLE STUDENT
MODIFY COLUMN PROGRAM char(5)
```

DATA MANIPULATION LANGUAGE (DML)

- A group of computer languages that includes commands enabling users to **interact with and manipulate data within a database**.

STUDENT

REG_NO	STUD_NAME	PROGRAM	DEPARTMENT	AGE
DTKF1001	ALI	DTK	JKE	20
DTKF1010	MEENA	DTK	JKE	19
DDTF1001	MAY	DDT	JTMK	19
DEMF1002	RAJU	DEM	JKM	21
DKAF1008	DAUD	DKA	JKA	20

- These manipulations encompass tasks such as **inserting new data** into database tables, **retrieving existing data**, **deleting records** from existing tables, and **modifying the content** of existing data.

- Most of SQL statements is categorized as DML (Data Manipulation Language), encompassing SQL commands specifically designed for modifying data within a database.

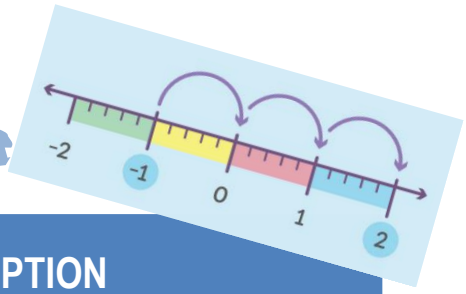
DML COMMAND

1. **INSERT**
   - to **add one or more records** to a database table
2. **DELETE**
   - to **removes one or more records** from a table according to specified conditions
3. **UPDATE**
   - to **modifies data** of one or more records
4. **SELECT**
   - to **retrieve rows** from a table
5. **Other Queries**
   - Simple Selections, Advanced queries (Aggregations, etc.), Nested Queries, Views

BASIC STRUCTURE OF AN SQL QUERY

GENERAL STRUCTURE	SELECT, ALL / DISTINCT, *, AS, FROM, WHERE
COMPARISON	IN, BETWEEN, LIKE "% _"
GROUPING	GROUP BY, HAVING, COUNT( ), SUM( ), AVG( ), MAX( ), MIN( )
DISPLAY ORDER	ORDER BY, ASC / DESC
LOGICAL OPERATORS	AND, OR, NOT
OUTPUT	INTO TABLE / CURSOR TO FILE [ADDITIVE], TO PRINTER, TO SCREEN
UNION	UNION
GENERAL STRUCTURE	SELECT, ALL / DISTINCT, *, AS, FROM, WHERE
COMPARISON	IN, BETWEEN, LIKE "% _"

SET COMPARISON IN SQL



COMPARISON COMMAND	DESCRIPTION
LIKE	column value is similar to specified character(s).
IN	column value is equal to any one of a specified set of values.
BETWEEN...AND	column value is between two values, including the end values specified in the range.
IS NULL	column value does not exist.
GROUP BY	in conjunction with the aggregate functions to group the result-set by one or more columns.

References

All DML operation are based on the following reference tables.

STUDENT

REG_NO	STUD_NAME	PROGRAM	DEPARTMENT	AGE
DTKF1001	ALI	DTK	JKE	20
DTKF1010	MEENA	DTK	JKE	19
DDTF1001	MAY	DDT	JTMK	19
DEMF1002	RAJU	DEM	JKM	21
DKAF1008	DAUD	DKA	JKA	20

COURSE

COURSE_CODE	COURSE_NAME	CREDIT
DBM20033	MATH	3
DEC40073	DATABASE SYSTEM	3
DEC50103	OS	3
DEE40082	PROJECT 1	2
DUE30022	ENGLISH COMM	2

REGISTER

REG_NO	COURSE_CODE
DTKF1001	DEC40073
DTKF1001	DEE40082
DTKF1010	DEE40082
DTKF1001	DUE30022
DKAF1008	DUE30022
DEMF1002	DBM20033

DATA INSERTION

Syntax:

```
INSERT INTO relation_name [(Att [,Att]*)]
VALUES (value [,value]*)
```

1. Write the SQL command to **insert new record** of into a COURSE table. The course info is:
   COURSE CODE : DEC30032
   COURSE NAME : COMPUTER ARCHITECTURE
   CREDIT : 2

```
INSERT INTO COURSE
VALUES ('DEC30032', 'COMPUTER ARCHITECTURE', 2)
```

COURSE_CODE	COURSE_NAME	CREDIT
DBM20033	MATH	3
DEC30032	COMPUTER ARCHITECTURE	2
DEC40073	DATABASE SYSTEM	3
DEC50103	OS	3
DEE40082	PROJECT 1	2
DUE30022	ENGLISH COMM	2

Output

RECORD DELETION

Syntax:

```
DELETE FROM relation_name
[WHERE condition]
```

1. Write the SQL command to **delete the Computer Architecture record** from COURSE table.

```
DELETE FROM COURSE
WHERE COURSE_NAME = 'COMPUTER ARCHITECTURE'
```

References

COURSE_CODE	COURSE_NAME	CREDIT
DBM20033	MATH	3
DEC30032	COMPUTER ARCHITECTURE	2
DEC40073	DATABASE SYSTEM	3
DEC50103	OS	3
DEE40082	PROJECT 1	2
DUE30022	ENGLISH COMM	2

Output

COURSE_CODE	COURSE_NAME	CREDIT
DBM20033	MATH	3
DEC40073	DATABASE SYSTEM	3
DEC50103	OS	3
DEE40082	PROJECT 1	2
DUE30022	ENGLISH COMM	2

DATA UPDATING

Syntax:

```
UPDATE relation_name
SET attribute = expression
[WHERE condition]
```

1. Write the SQL command to **change the course name** of OS into Operating System.

```
UPDATE COURSE
SET COURSE_NAME = 'OPERATING SYSTEM'
WHERE COURSE_CODE = 'DEC50103'
```

Output

COURSE_CODE	COURSE_NAME	CREDIT
DBM20033	MATH	3
DEC30032	COMPUTER ARCHITECTURE	2
DEC40073	DATABASE SYSTEM	3
DEC50103	OPERATING SYSTEM	3
DEE40082	PROJECT 1	2
DUE30022	ENGLISH COMM	2

RECORD RETRIEVING USING SELECT

Syntax:

```
SELECT target-list
FROM relation-list
[WHERE qualification]
```

ALL ATTRIBUTES (*)

1. Write the SQL command to **display the record of all students**.

```
SELECT *
FROM STUDENT
```

Output

REG_NO	STUD_NAME	PROGRAM	DEPARTMENT	AGE
DDTF1001	MAY	DDT	JTMK	19
DEMF1002	RAJU	DEM	JKM	21
DKAF1008	DAUD	DKA	JKA	20
DTKF1001	ALI	DTK	JKE	20
DTKF1010	MEENA	DTK	JKE	19

CERTAIN ATTRIBUTES

1. Write the SQL command to **display registration number and name of all students.**.

```
SELECT REG_NO, STUD_NAME
FROM STUDENT
```

Output

REG_NO	STUD_NAME
DDTF1001	MAY
DEMF1002	RAJU
DKAF1008	DAUD
DTKF1001	ALI
DTKF1010	MEENA

BASED ON CONDITION

1. Write the SQL command to display the name for all students whose program is DKA or age less than 20.

```
SELECT STUD_NAME
FROM STUDENT
WHERE PROGRAM = 'DKA'OR AGE <20
```

Output

STUD_NAME
MAY
DAUD
MEENA

DISTINCT

1. Write the SQL command to **return the distinct value of department** in STUDENT table.

```
SELECT DISTINCT DEPARTMENT
FROM STUDENT
```

Output

DEPARTMENT
JTMK
JKM
JKA
JKE

ARITHMETIC IN SQL

1. Write an SQL command to calculate the **age of a student in the next 5 years**.

```
SELECT *, AGE+5 AS NEW_AGE
FROM STUDENT
```

Output

REG_NO	STUD_NAME	PROGRAM	DEPARTMENT	AGE	NEW_AGE
DDTF1001	MAY	DDT	JTMK	19	24
DEMF1002	RAJU	DEM	JKM	21	26
DKAF1008	DAUD	DKA	JKA	20	25
DTKF1001	ALI	DTK	JKE	20	25
DTKF1010	MEENA	DTK	JKE	19	24

COLUMN RENAMING

1. Write the SQL command to calculate total age for all students and display it as TOTAL_AGE.

```
SELECT SUM( AGE ) AS TOTAL_AGE
FROM STUDENT
```

Output

TOTAL_AGE
99

COUNT

1. Write the SQL command to **calculate total courses** from the COURSE table.

```
SELECT COUNT(COURSE_CODE) AS TOTAL_COURSE
FROM COURSE
```

Output

TOTAL_COURSE
5

GROUP BY

1. Write an SQL command to **count the number of student records for each department** in STUDENT table.

```
SELECT DEPARTMENT, COUNT( * ) AS TOTAL_RECORD
FROM STUDENT
GROUP BY DEPARTMENT
```

Output

DEPARTMENT	TOTAL_RECORD
JKA	1
JKE	2
JKM	1
JTMK	1

HAVING

1. Write an SQL command to **count the number of student for each department** in STUDENT table. Display the department with more or equal 2 students.

```
SELECT DEPARTMENT, COUNT(*) AS TOTAL_RECORD
FROM STUDENT
GROUP BY DEPARTMENT
HAVING COUNT(*)>= 2
```

Output

DEPARTMENT	TOTAL_RECORD
JKE	2

IN

1. Write the SQL command to retrieve the student records where the **age is 19 and 21** .

1ST METHOD

```
SELECT *
FROM STUDENT
WHERE AGE IN (19, 21)
```

2ND METHOD

```
SELECT *
FROM STUDENT
WHERE AGE = 19 OR AGE = 21
```

Output

REG_NO	STUD_NAME	PROGRAM	DEPARTMENT	AGE
DDTF1001	MAY	DDT	JTMK	19
DEMF1002	RAJU	DEM	JKM	21
DTKF1010	MEENA	DTK	JKE	19

BETWEEN...AND

1. Write the SQL command to retrieve the student records **within the age range of 19 to 21**.

1ST METHOD

```
SELECT *
FROM STUDENT
WHERE AGE BETWEEN 19 AND 21
```

2ND METHOD

```
SELECT *
FROM STUDENT
WHERE AGE >= 19 AND AGE <= 21
```

Output

REG_NO	STUD_NAME	PROGRAM	DEPARTMENT	AGE
DDTF1001	MAY	DDT	JTMK	19
DEMF1002	RAJU	DEM	JKM	21
DKAF1008	DAUD	DKA	JKA	20
DTKF1001	ALI	DTK	JKE	20
DTKF1010	MEENA	DTK	JKE	19

LIKE

1. Write the SQL command to retrieve the student records where the **names begin with 'M'** .

```
SELECT *
FROM STUDENT
WHERE STUD_NAME LIKE "M%"
```

Output

REG_NO	STUD_NAME	PROGRAM	DEPARTMENT	AGE
DDTF1001	MAY	DDT	JTMK	19
DTKF1010	MEENA	DTK	JKE	19

MIN, MAX & AVG

1. Write an SQL command to display minimum, maximum and average value of student's age.

```
SELECT MIN(AGE), MAX(AGE), AVG(AGE)
FROM STUDENT
```

Output

MIN(AGE)	MAX(AGE)	AVG(AGE)
19	21	19.8000

3

DATABASE SYSTEM ENVIRONMENT

Introduction

DBMS, or Database Management System, consists of various components that collaborate to establish and oversee a comprehensive system for creating and managing databases. Data, referring to individual items stored in the database, requires secure and efficient management to ensure that only authenticated individuals can access, modify, or store it seamlessly.

COMPONENT OF DBMS ENVIRONMENT

Hardware

Software

People

Procedures

Data

HARDWARE

- Hardware in a database environment refer to the **computers and associated peripherals utilized for database management**.

- Comprising a collection of tangible electronic devices such as computers, I/O devices, and storage devices, this component serves as the bridge between computers and the actual system in the real world.

- It is responsible for the storage and maintenance of data within the database.

- DBMS cannot be implemented and use without hardware. Database also created, accessed and/or managed using hardware.

https://www.youtube.com/watch?app=desktop&v=YWoesdg6YNw

SOFTWARE

- Software is **a set of programs used to manage and control the database.**
- It is including the database software, operating system and network software that used to share the data with other users, and the applications used to access the data.
- The operating system software manages all hardware components and enabling the execution of various software on servers and computers.
- The DBMS furnishes a user-friendly interface for storing, retrieving, updating, and deleting data, while application programs are responsible for accessing and manipulating the data.

https://images.app.goo.gl/i2HChs7L5ryCh9Nw8

https://images.app.goo.gl/bGr8B6qFAmPsKLAz7

DATA

- Data consists of unprocessed facts and that need to be organized.
- This structured information has the potential to be transformed into something meaningful.
- As databases expand and gain complexity, it will be increasing the difficulty to keep track of what all the entities and attributes mean.

- Many organizations utilize a data dictionary to store and disseminate the characteristics, definitions, and relationships of data.
- Additionally, organizations often use an information catalogue to document the business details and rules related to their databases.

PROCEDURE

- Procedures refer to the guidelines directing the design, configuration, and administration of the DBMS. Standards and procedures play a crucial role in ensure the consistency and effectiveness of the database environment.
- Furthermore, it become difficult to maintain the database without standards and procedures.

Manage the whole database operations.

Database administrators

Defines the details of the database design, including tables, indexes, views, constraints, etc.

Database designers

Design and develop business applications that capture and store business data in databases.

Analysts and programmers

Who indirectly connect to databases through business applications

End users

https://images.app.goo.gl/8UaRKHDTPyC59Wze8

PEOPLE

- People is referring to **human who used or deal with the database system**
- They able to control, manage and administrate the entire database by performing various types of query and operations on the databases in the DBMS.
- Database users are categorized into various groups, typically based on their roles or occupations.
- Some of these user groups include:.
  a) Database administrators (DBAs)
  b) Database designers
  c) Analysts and programmers
  d) End users

CHOOSE THE CORRECT ANSWER…

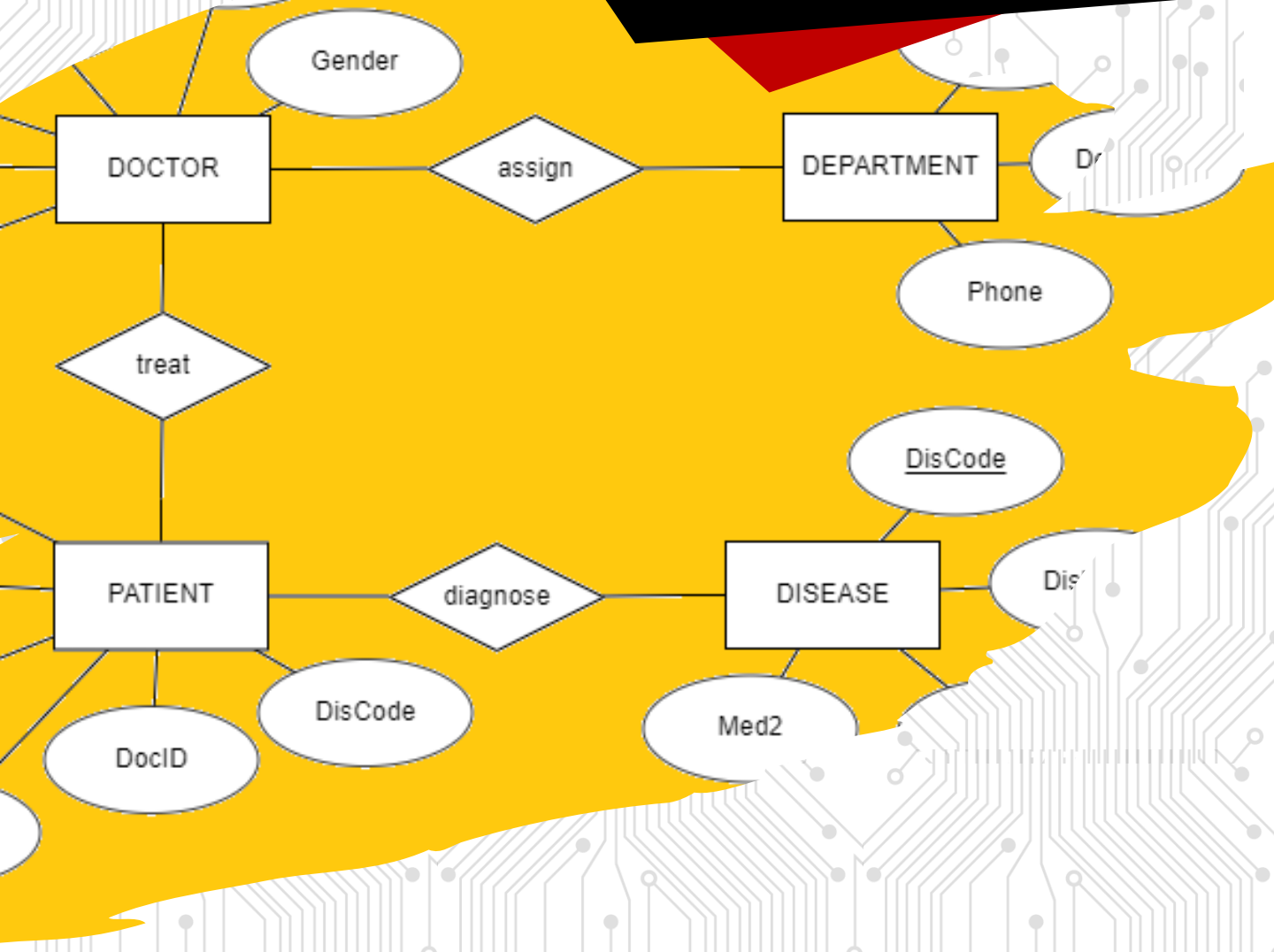
1) **The database environment has all the following components except**
   A). Database administrator
   B). Database
   C). Users
   D). Separate files

2) **The database system is composed of four major parts:**
   A). Software, You, Me, DBA, Client
   B). Hardware, Hard drive, Monitor, Data, User
   C). Hardware, Software, People and Data
   D). DBMS, Hardware, User, Programmer, Engineer

3) **Which of the following is related to a component of DBMS known as personnel?**
   A). Database administrator
   B). Application programmer
   C). End users
   D). All

4

ENTITY RELATIONSHIP MODEL (ERM)



Introduction

Data models explain how a database is organized in logical structure. It define how data is connected and the ways they are managed and stored in the system. Three main components outline by a data model are entities, relationship, and requirements.

The data model encapsulates the description of the database. It offers a transparent view of data and aiding in designing a real database. It guides us through the journey from data design to proper data implementation.

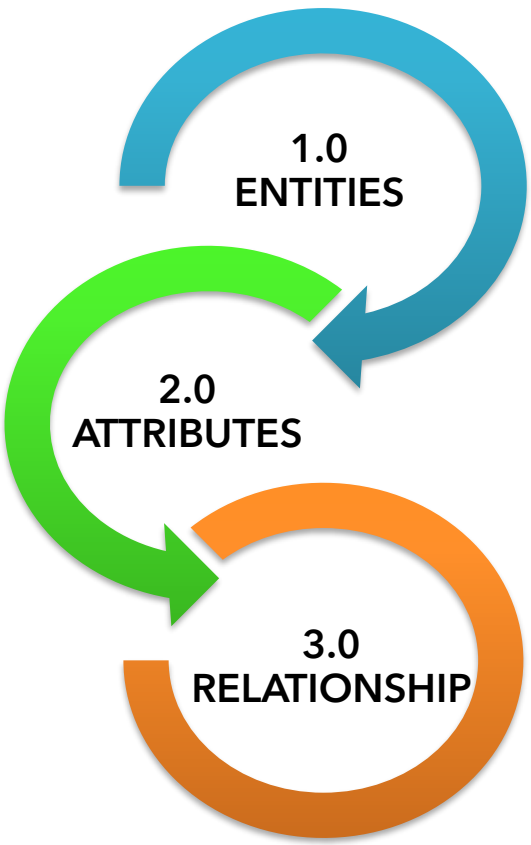
A set of symbols, text or a standard format is used to represent it so that all members of the organization can understand how the data is organized. It provides a set of conceptual tools that are widely used to illustrate the description of data.

There are 3 types of data model in database system:

a) Record Based Data Model
b) Object Based Data Model
c) Physical Based Data Model

ENTITY RELATIONSHIP MODEL (ERM)

- Object Based Data Models are used to describe data and its relationship. It uses concepts such as entities, attributes and relationships.
- It has flexible data structuring capabilities.
- Data integrity constraints can be explicitly specified using objects-based data models.
- **Entity Relationship Model** is one of Object Based Data Models. It is a conceptual model that represents components in system graphically.



ELEMENTS OF ERM

ENTITY

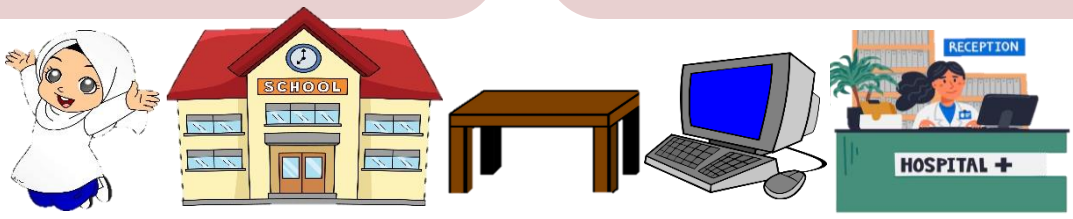
DEFINITION

An object type that is uniquely identifiable.

Entity is a person, place, object, event or concept, which you want to store the information in a database.

CHARACTERISTICS

- Contain information.
- Possibility of more than one occurrence.
- Each occurrence must be uniquely identifiable
- Will become a TABLE in database



TYPES OF ENTITY

TYPES	SYMBOL/ NOTATION	EXPLAINATION
Strong Entities	Entity	• The Strong entity is independent of any other entity in a schema. • Often called parent entities. • They will also have a primary key
Weak Entities	Weak Entity	• Weak entity depends on the strong entity for its existence • They don't have primary keys and have no meaning in the diagram without their parent entity.
Associative (Composite) Entities	Associative Entity	• Associative entities relate the instances of several entity types • Used to break up M:N relationships into 1:M relationships.

ATTRIBUTE

DEFINITION

- A **property of an entity** or a relationship type

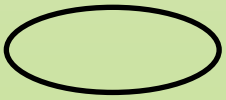
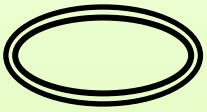
EXAMPLES

- A student entity may have **name**, **class**, and **age** as attributes.

VALUE SET (DOMAIN)

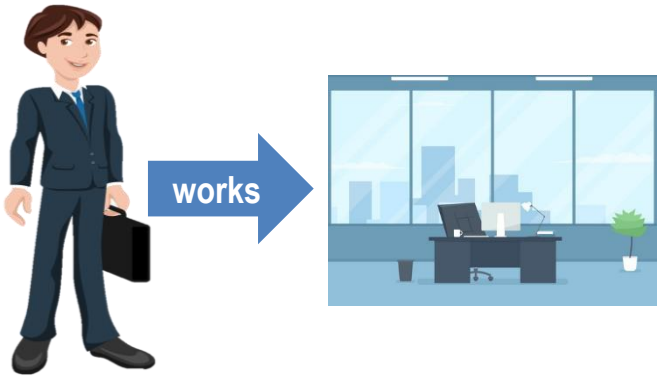
- Is the **set of allowable values** for **one or more attributes.**
- **Data type** specified for each domain
- For example, a student's name cannot be a numeric value. It has to be alphabetic. A student's age cannot be negative.

TYPES OF ENTITY

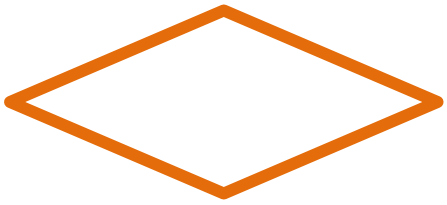
TYPES	SYMBOL/ NOTATION	EXPLAINATION
Single Valued		• An attribute that holds a single value for a single entity • Example; The majority of people have only one **name**.
Multi Valued		• An attribute that holds multiple values for a single entity. • A staff may have 2 tel_no, which are home tel_no and hp tel_no.
Derived		• An attribute that represents a value that is derivable from the value of a related attribute, not necessarily in the same entity. • Age can be known from Date_of_birth
Composite		• An attribute that have more than one attributes. • An address comprises of city, postcode and state

RELATIONSHIP

- An **association between two or more entities**.
- Example;
  An employee works at a department, a student enrolls in a course
  (**works** and **enroll** are called relationships)
- May have attributes associated with them.
- Have degree of association with their participating entities.

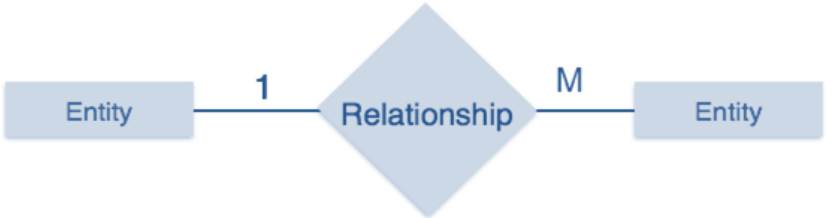


NOTATION

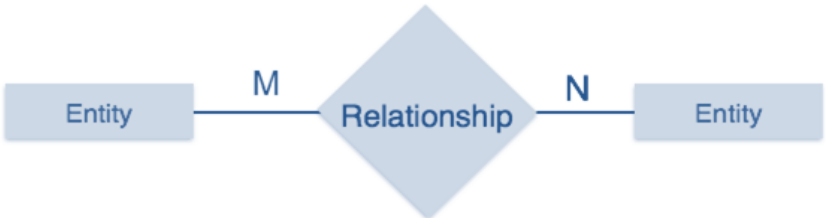


TYPES OF RELATIONSHIP (CARDINALITY)

1:1 (one to one)



1:M (one to many)

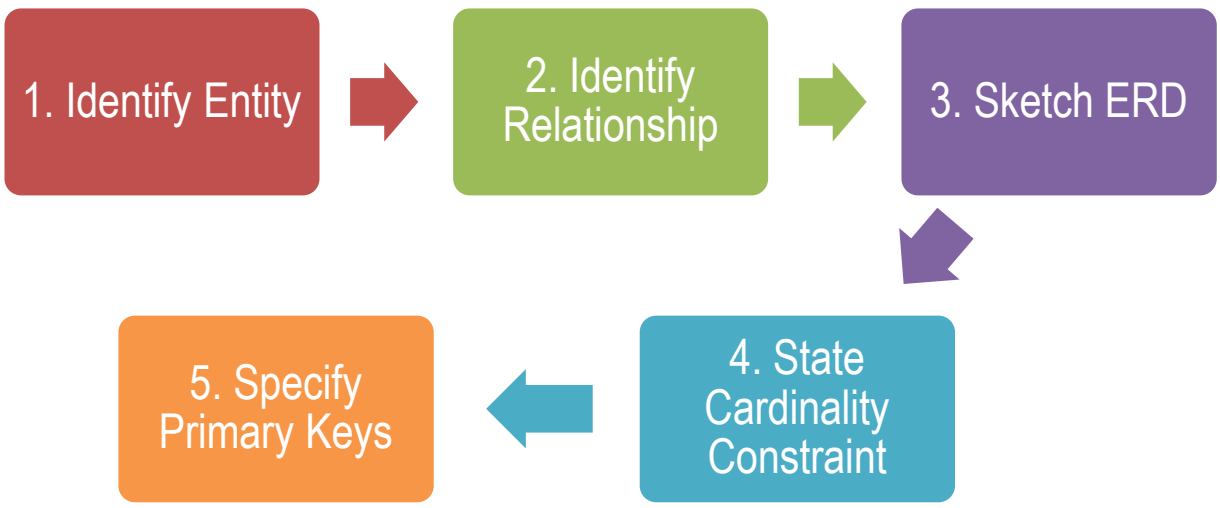


M:N (many to many)

KEY ATTRIBUTE

SUPER KEY	• An attribute, or set of attributes, that uniquely identifies a tuple within a relation.
CANDIDATE KEY	• The attribute or a set of attributes in a relation that qualifies for uniqueness of each tuple/row. • Any of the identified candidate keys can be used as the table's primary key.
PRIMARY KEY	• The Candidate key attribute/column that is most suited to maintain uniqueness in a table at the tuple/row level
ALTERNATE KEY	• The other Candidate key attribute/columns that you didn't choose as Primary key.
FOREIGN KEY	• An attribute in one table whose values must match the primary key in another table or whose must be null

DEVELOPING ER DIAGRAM



Example

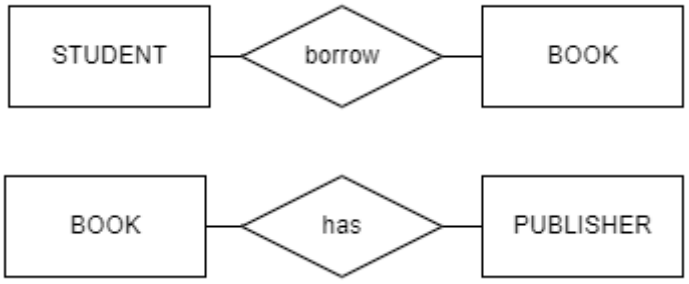
Based on the case study below:

A student can borrow one or more books. A book has a supplier and a supplier can supply many books.The information of students are matric no, student name, class, semester and phone no.The important data of book are serial no, book title and year of published. Supplier have id supplier, address and phone no.

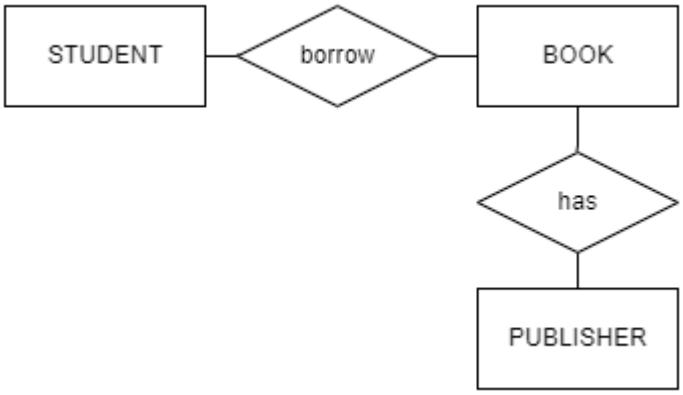
1 Identify Entity

1. STUDENT
2. BOOK
3. SUPPLIER

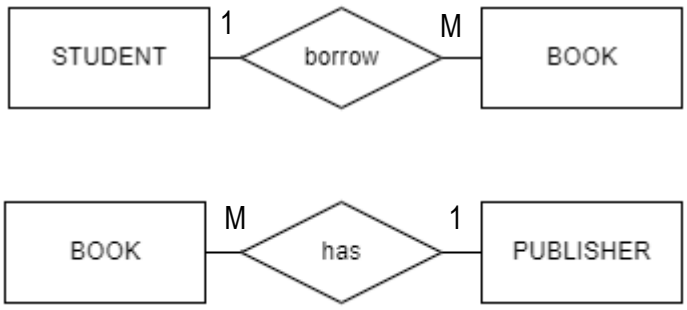
2 Identify Relationship



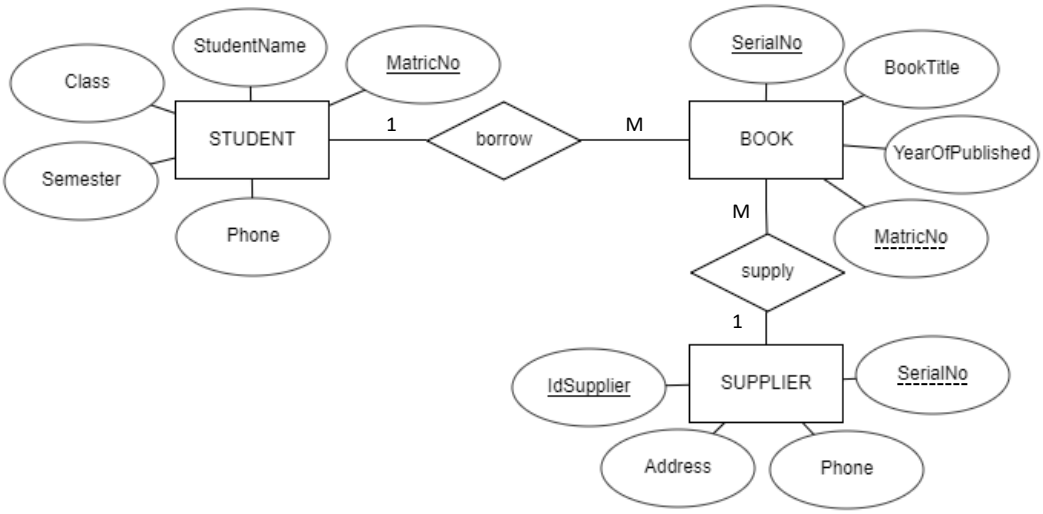
3 Sketch ERD



4 State Cardinality Constraint



5 Specify Primary Keys



REFERENCES

Connolly T.M and Begg C.E (2009). Database System – A Practical Approach to Design, Implementation and Management 5 th Edition. Pearson Addison Wesley.

Coronel C, Morris S. and Rob P. (2012). Database System: Design, Implementation and. Management 10th .Edition Cengage Learning.

Dassanayake, D. (2022, November 17). The 5 Components of DBMS. Retrieved from https://analytixminds.com/the-5-components-of-dbms/

Database Management Systems. (n.d.). Database Management Systems. Retrieved from https://nexwebsites.com/database/database-management-systems/

DEC40073 PW6.pdf. (2022). Retrieved from https://www.coursehero.com/file/153772952/DEC40073-PW6pdf/

Elmasri R. and Navathe S. (2007).Fundamentals of Database Systems. 5th Edition. Addison-Wesley.

Ganesan, P. (2021, June 23). What is XAMPP? and How to Install XAMPP on Local Computer? Retrieved from https://wpblogx.com/what-is-xampp/

Mangavkar, S. (2020, July 16). Components of DBMS Environment. Retrieved from https://medium.com/@mangavkarsonali/components-of-dbms-environment-f4f0651e6daf

Singh S.K (2006). Database System: Concepts, Design and Application. Dorling Kindersley.

Techopedia. (2020, September 21). Retrieved from Data Definition Language (DDL): https://www.techopedia.com/definition/1175/data-definition-language-ddl

Wan Azlinie binti Wan Ahmad and Nur Aisyah binti Mohamad Rafiuddin . (2022). INSTRUCTOR & STUDENT REFERENCE BOOK DEC40073 DATABASE SYSTEM. Dungun: Department Of Electrical Engineering, PSMZA.

Zahid, M. (2017, November 04). DBMS MCQ Questions With Answers Set 2. Retrieved from https://www.mcqsquestions.com/2017/11/dbms-mcq-questions-with-answers-set-2.html

Department of Electrical Engineering
Politeknik Sultan Mizan Zainal Abidin
KM 08, Jalan Paka,
23000 Dungun, Terengganu
☎ 09-8400869

e ISBN 978-967-0047-49-2