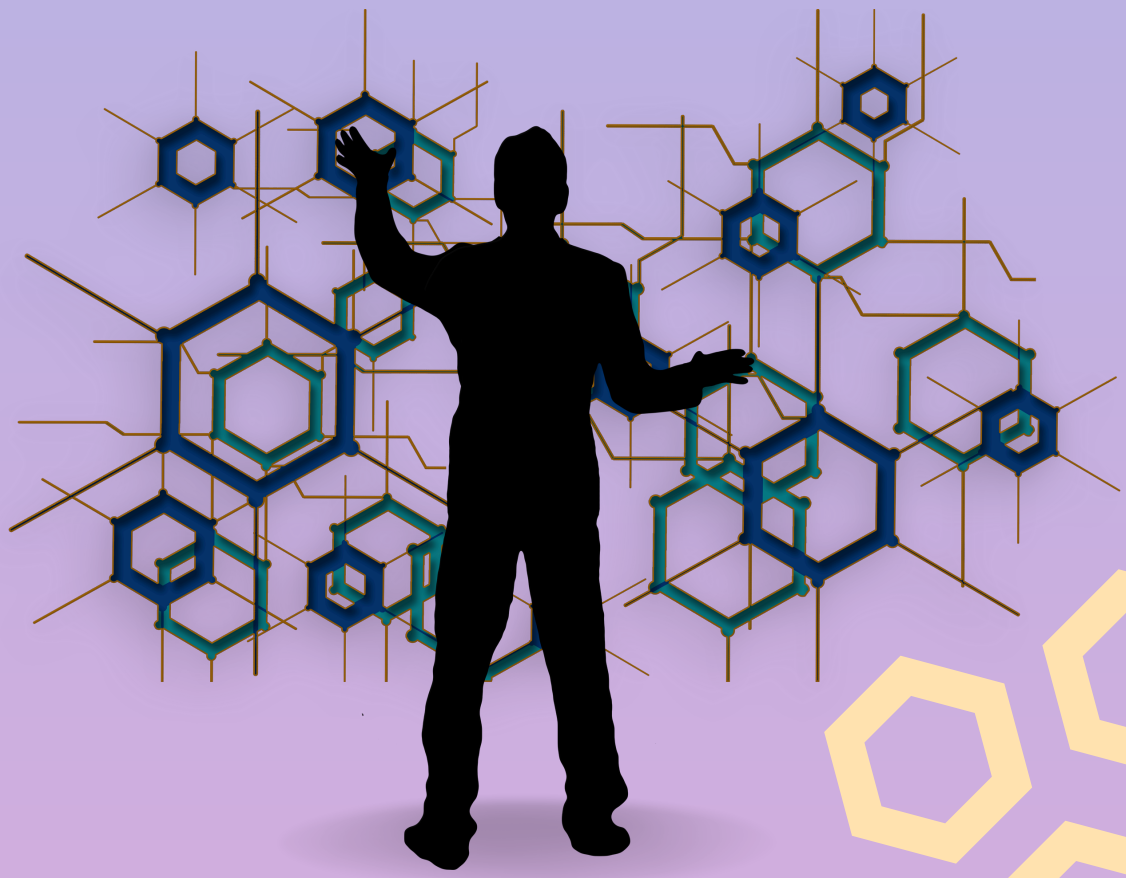
POLITEKNIK MALAYSIA

MUAT

# Smart

# DBMS

# WORKBOOK



NURUL FADHLUN AWANG
AZALINDA BINTI MAT SAAD

# Smart DBMS WORKBOOK

GUIDED FOR STUDENTS OF POLITEKNIK MALAYSIA


## AUTHORS

NURUL FADHLUN AWANG

AZALINDA MAT SAAD



## PUBLISHER

POLITEKNIK NILAI

KOMPLEKS PENDIDIKAN NILAI

BANDAR ENSTEK

71760 LABU

NEGERI SEMBILAN

First edition: 2025

Perpustakaan Negara Malaysia Cataloguing-in-Publication Data
Smart DBMS Workbook/ Nurul Fadhlun Awang
e ISBN 978-967-2742-52-4

Reviewer:
Norhafeeza Bambang Shaidi

Authors:
Nurul Fadhlun Awang
Azalinda Mat Saad

Cover Design and Interior Layout:
Nurul Fadhlun Awang

Interactive  Design
Azalinda Mat Saad

Publisher
Politeknik Nilai

# ACKNOWLEDGEMENT

In the name of Allah, the Most Gracious, the Most Merciful.

All praise is due to Allah for granting us the strength, clarity, and determination to complete this Samrt DBMS Workbook. His guidance has been our greatest source of motivation.

We would like to express our heartfelt thanks to our families for their endless support, patience, and encouragement throughout this journey. Their presence behind the scenes made this effort possible.

Special appreciation is extended to Norhafeeza binti Bambang Shaidi, Head of Program (Diploma in Business Information System) for her role as content reviewer. Her insights and feedback were invaluable in refining the structure and quality of this workbook.

We are also grateful to Politeknik Nilai for providing the academic foundation and curriculum support that shaped this resource. It is our sincere hope that this workbook will serve as a practical and meaningful tool for students and educators alike.
To all who contributed in any way, directly or indirectly, thank you. May this effort benefit many in the pursuit of knowledge.

Nurul Fadhlun Awang & Azalinda Mat Saad

# PREFACE

The Smart DBMS Workbook is developed as a practical guide to support students in understanding and applying core database concepts. It is aligned with the syllabus of the Diploma in Business Information Systems at polytechnic across Malaysia.

This workbook combines concise notes with structured exercises to strengthen both theoretical understanding and hands-on skills. Interactive elements such as QR codes linking to notes, tutorial videos, quizzes, and .sql practice files are included to enhance learning. Additionally, Augmented Reality (AR) features are integrated to provide a more engaging and visual learning experience.

This workbook is the result of a commitment to making learning more accessible, interactive, and student-centered. It is our hope that this resource not only supports classroom learning but also encourages students to explore databases beyond the textbook.
We welcome feedback for future improvements and extend our appreciation to everyone who contributed to the development of this workbook.

# ABSTRACT

The Smart DBMS Workbook is an interactive learning resource tailored for polytechnic-diploma-level students, particularly those in the field of Business Information Systems. This workbook integrates theoretical concepts with hands-on activities to strengthen students' understanding of database fundamentals.

Key topics include data modeling, entity-relationship diagrams, relational schema design, and SQL commands. To enhance engagement and accessibility, the workbook incorporates digital features such as QR codes linking to notes, video tutorials, interactive quizzes, and downloadable .sql files. It also introduces Augmented Reality (AR) elements to provide a more immersive and visual learning experience.

By combining traditional workbook elements with modern learning tools, this resource supports both classroom instruction and independent learning, aiming to promote a deeper and more practical understanding of database systems.

# TABLE OF CONTENTS

# CHAPTER 1

## Introduction to Database System

# CONCEPT OF DATABASE

1. Define term database.

.....

.....

.....

.....

.....

2. Identify **FOUR (4)** purposes of database.

.....

.....

.....

.....

.....

3. Explain how does database is differ from a traditional file system.

.....

.....

.....

.....

.....

.....

.....

.....

.....

4. State **THREE (3)** examples of database system in real life.

.....

.....

.....

HELPFUL TIPS

Remember the keywords of database!

- Organized
- Structured
- Systematic
- Accessible
- Reliable

NOTES

SCAN ME!

# DATA AND INFORMATION

1. Define the term data.

.....

.....

.....

.....

.....

2. Identify data required to:

- declare exam results

.....

.....

- issue biometric ID cards

.....

.....

- register hotel booking information

.....

.....

3. Discuss the role of data in decision making.

.....

.....

.....

.....

.....

4. Define information.

.....

.....

.....

.....

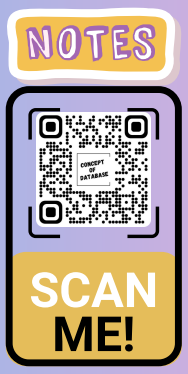
.....


Remember the keywords of data!

- Unorganized
- Raw
- Carry no meaning

Remember the keywords of information!

- Organized
- Processed
- Meaningful



# DATA AND INFORMATION

5 Identify **FOUR (4)** examples of information.

.....

.....

.....

.....

.....

6 Explain how unstructured data can be converted into meaningful information.

.....

.....

.....

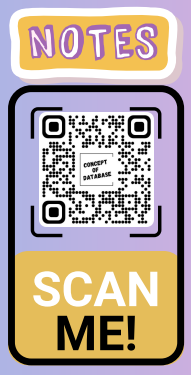
.....

.....

.....

.....

.....




7 State **THREE (3)** differences between data and information.

.....

.....

.....

.....

.....

.....

.....

.....

# DATA COLLECTION METHODS

1. Describe data collection.

.....

.....

.....

.....

.....

2. List **FIVE (5)** data collection methods.

.....

.....

.....

.....

.....

3. Differentiate between primary and secondary data collection methods.

.....

.....

.....

.....

.....

4. Determine how does observation as a data collection method help researchers.

.....

.....

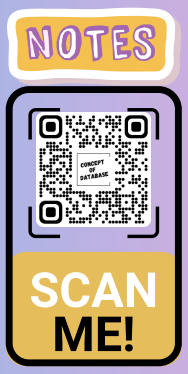
.....

.....

.....


Remember the keywords of data collection!

- Gather data
- Relevant resources
- Evaluate outcome



# DATABASE 3 SCHEMA ARCHITECTURE

1. Define database schema.

.....

.....

.....

.....

.....

2. Define database three schema architecture.

.....

.....

.....

.....

.....

3.

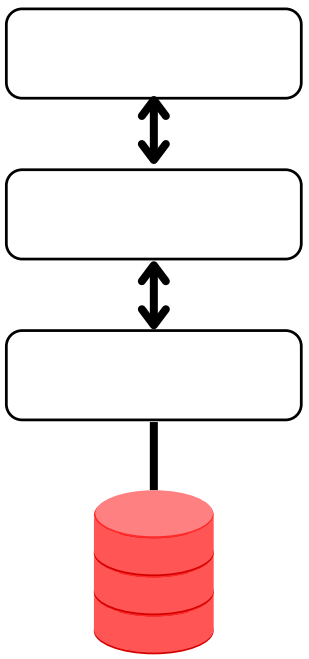



**Figure 1.1: Database 3 schema architecture**

Fill in the text box with an appropriate arrangement of three levels of database 3 schema architecture.

HELPFUL TIPS

Remember the keywords of database schema!

- Structure
- Define the organization of data

NOTES

SCAN ME!

# DATABASE 3 SCHEMA ARCHITECTURE

4 Classify the given statements into the three levels of database schema architecture.

| STATEMENTS | LEVEL OF DATABASE SCHEMA ARCHITECTURE |
|---|---|
| It is known as view level. | |
| It is known as logical level. | |
| It is known as physical level | |
| This level is the lowest layer, focusing on the physical storage of data. It describes how data is stored on hardware and accessed by the system. | |
| This level provides a unified, abstract view of the entire database, focusing on what data is stored and the relationships between data. | |
| This level is the topmost layer that interacts with end-users. It defines how individual users or user groups perceive the data. | |

# DATABASE MANAGEMENT SYSTEM (DBMS)

1. Define Database Management System (DBMS).

.....

.....

.....

.....

.....

2. List **THREE (3)** examples of DBMS software.

.....

.....

.....

3. The statements below refer to components of a database management system. Identify which component each statement belongs to.

| STATEMENTS | COMPONENTS OF DBMS |
|---|---|
| A basic language that allows users to write commands to perform desired operations on the data stored in the database. | |
| A collection of programs that instruct the computer on how to perform its functions | |
| Used to ensure that there is an orderly method for monitoring and auditing both the data that enters the database and the information generated by that data. | |


Remember the keywords of DBMS!

- Software
- Manipulate data
- Organized
- Reliable

# DATABASE MANAGEMENT SYSTEM (DBMS)

4 Explain **TWO (2)** advantages of database management system.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

5 Explain **TWO (2)** disadvantages of database management system.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

HELPFUL TIPS

Remember

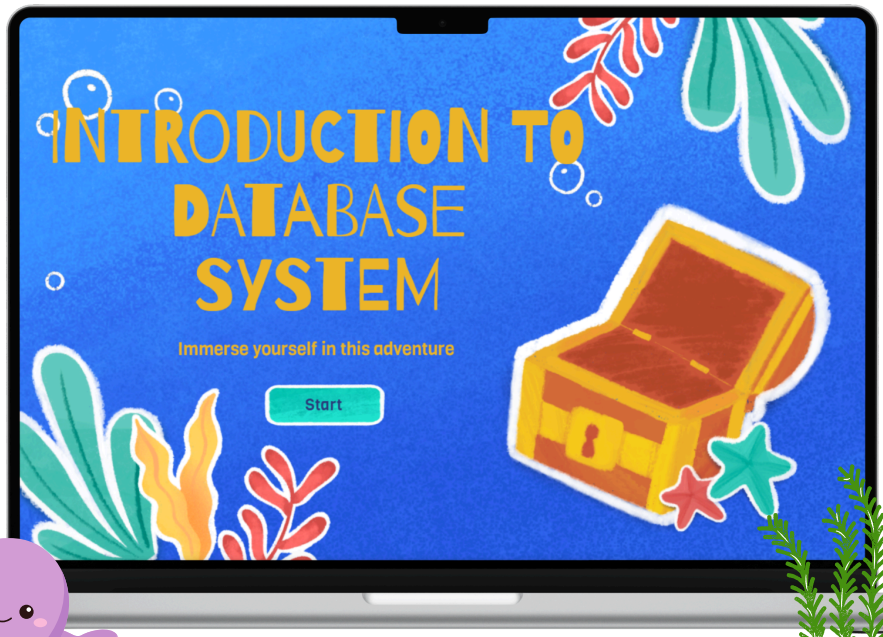
- DBMS is a software system for creating and managing databases.

NOTES

SCAN ME!

## INTERACTIVE QUIZ

# An Ocean Adventure into the Basics of Databases!




Scan me

# CHAPTER 2

## Data Modelling

# ENTITY RELATIONSHIP DIAGRAM (ERD)

1. Define Entity Relationship Diagram.

.....

.....

.....

.....

.....

2. List **THREE (3)** components of ERD.

.....

.....

.....

3. Draw the symbols for each ERD component and name them.

| Figures | Symbol | ERD Component Name |
|---|---|---|
| Rectangle | | |
| Ellipse | | |
| Diamond | | |
| Line | | |
| Double ellipse | | |
| Double rectangle | | |

4. A food delivery platform needs to keep track of restaurants, customers, and orders to improve its operations. Based on this scenario, propose **THREE (3)** entities that should be included in the database and provide **FOUR (4)** attributes for each entity.

| Entity | **Attributes** |
|---|---|
| | |
| | |
| | |


Remember the keywords of data Entity Relationship Diagram!

- Data modelling technique
- Give graphical representation

# ENTITY RELATIONSHIP DIAGRAM (ERD)

5. An e-commerce company wants to develop a database to manage its products, customers, and sales transactions. Based on this scenario, propose **THREE (3)** entities that should be included in the database and provide **FOUR (4)** attributes for each entity.

| Entity | Attributes |
|---|---|
| | |
| | |
| | |

6. A school needs a database system to manage information about students, teachers, and classes. Based on this scenario, propose **THREE (3)** entities that should be included in the database and provide **FOUR (4)** attributes for each entity.

| Entity | Attributes |
|---|---|
| | |
| | |
| | |

7. A university library wants to develop a system to manage book borrowing. The system should track details about the books, the members, and the borrowing transactions. Based on this scenario, propose **THREE (3)** entities that should be included in the database and provide **FOUR (4)** attributes for each entity.

| Entity | Attributes |
|---|---|
| | |
| | |
| | |

HELPFUL TIPS

Remember the term

- Entities - objects or concepts that represent important data.
- Attribute - characteristic of an entity.
- Relationship - association between two or more entities

NOTES

SCAN ME!

# ENTITY RELATIONSHIP DIAGRAM (ERD)

**8** A university database needs to store information about students (identified by student_id, with name, date_of_birth, and address), courses (identified by course_id, with title and credit_hours), and instructors (identified by instructor_id, with name, department, and phone). Assume the situations as below:

- Students enroll in courses; and
- Course is taught by instructor.

Create an ERD for this university system, including all entities and their attributes.

**ANSWER**



**9** A hotel database needs to track guests (identified by guest_id, with name, email, and phone), rooms (identified by room_id, with type, rate, and availability), and reservations (identified by reservation_id, with check_in_date and check_out_date). Assume the situations as below:

- Guests make reservations for specific rooms; and
- Rooms can have multiple reservations.

Design an ERD for the hotel booking system, specifying all entities and attributes.

**ANSWER**




Remember

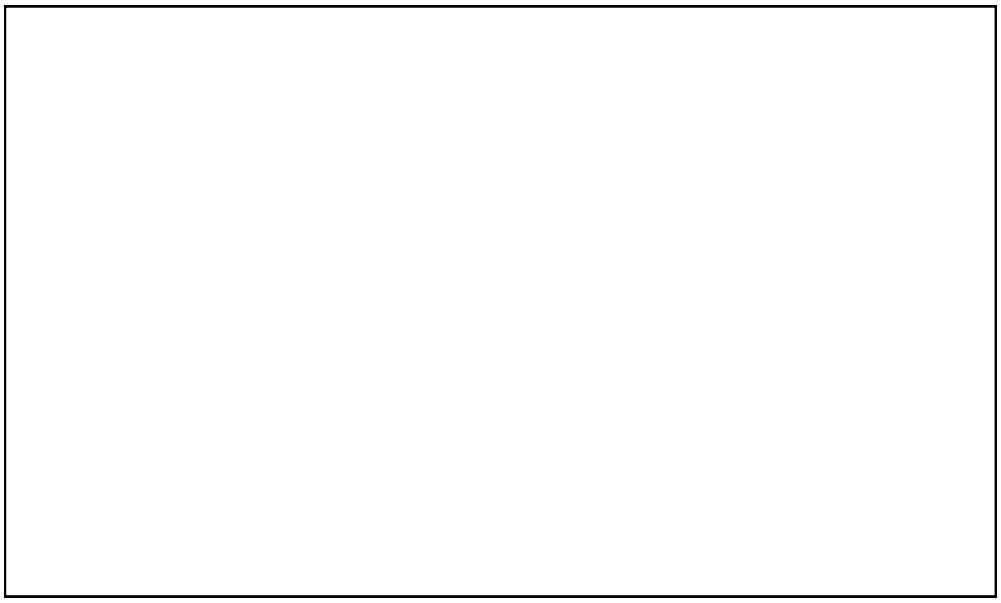
- An ERD has three main components: entities, relationships, and attributes connected by lines

# ENTITY RELATIONSHIP DIAGRAM (ERD)

10 An e-commerce platform database needs to manage customers (identified by customer_id, with name, email, and phone), products (identified by product_id, with name, price, and stock_quantity), orders (identified by order_id, with order_date and total_price), and payment (identified by payment_id, with type, status, and transaction_date). Assume the situations as below:

- Customers place orders;
- An order can include multiple products;
- Customer make payment; and

Create an ERD for this e-commerce platform, defining all entities and their attributes.




Remember

- use a singular nouns for “entities” name


# ENTITY RELATIONSHIP DIAGRAM (ERD)

11 A hospital database needs to manage information about patients (identified by patient_id, with name, age, and contact), doctors (identified by doctor_id, with name, specialization, and contact), wards (identified by ward_id, with name, capacity, and location), and medical records (identified by record_id, with diagnosis, treatment_plan, and date_of_record). Assume the situations as below:

- Patients are assigned to wards;
- Doctors treat patients;
- Patient has a medical record; and
- Wards are supervised by a doctor.

Design an ERD for this hospital patient management system, specifying all entities and attributes.

# DATABASE KEYS

1. Define database key.

.....

.....

.....

.....

.....

2. Explain **THREE (3)** uses of a database key.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. List **THREE (3)** types of key in a database.

.....

.....

.....

4. Explain one type of key in a database.

.....

.....

.....

.....

.....

.....

HELPFUL TIPS

Remember the keywords of database key!

- An attribute or group of attributes
- Uniquely identify rows

NOTES

SCAN ME!

# DATABASE KEYS

**5** Identify the primary key.

| Table | Attributes | Primary Key |
|---|---|---|
| Student | StudentID, Name, Age, Major | |
| Book | ISBN, Title, Author, Genre, PublishYear | |
| Order | OrderID, CustomerID, Date, TotalAmount | |
| Employee | EmployeeID, FirstName, LastName, Email | |
| Movie | Title, Director, Year, Length, MovieID, Producer | |

**ANSWER**

HELPFUL TIPS

Remember

- Primary keys serve as unique identifiers for each row in a database table.

NOTES

TYPES OF KEYS

SCAN ME!

# CARDINALITY

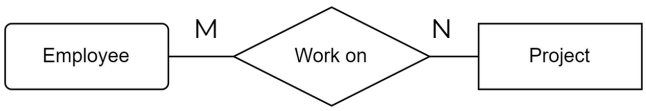
1. List **THREE (3)** types of entity relationship cardinality.

.....

.....

.....

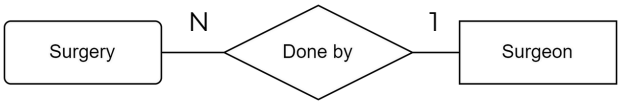
2. State the cardinality and existence of the relationship in each case.




.....

.....

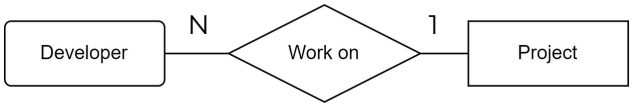
.....




.....

.....

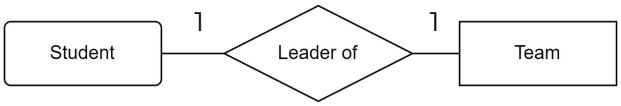
.....




.....

.....

.....



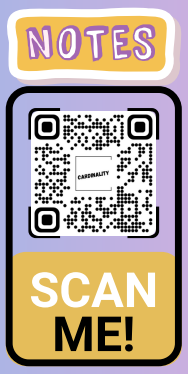

.....

.....

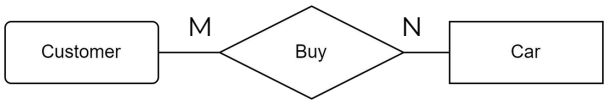
.....


Remember the keywords of cardinality!

- Relationship between tables
- Number of rows involved in a relationship



# CARDINALITY




.....

.....

.....

**3** Create an ERD with cardinality for each of the following situation.

| SITUATION | | SQL STATEMENT |
|---|---|---|
| a. | A students have to opt for a single course, but a course can have many students. | |
| b. | A customer can place many orders, but an order cannot be placed by many customers. | |
| c. | A student has only one identification card and an identification card is given to one student. | |
| d. | An employee can be assigned to many projects and a project can have many employees. | |
| e. | A professor teaches many classes. A class is taught by a single professor. | |
| f. | A student enrolls in many classes. A class has many students | |

HELPFUL TIPS

Remember the types of cardinality!

- One-to-One.
- One-to-Many.
- Many-to-One.
- Many-to-Many.

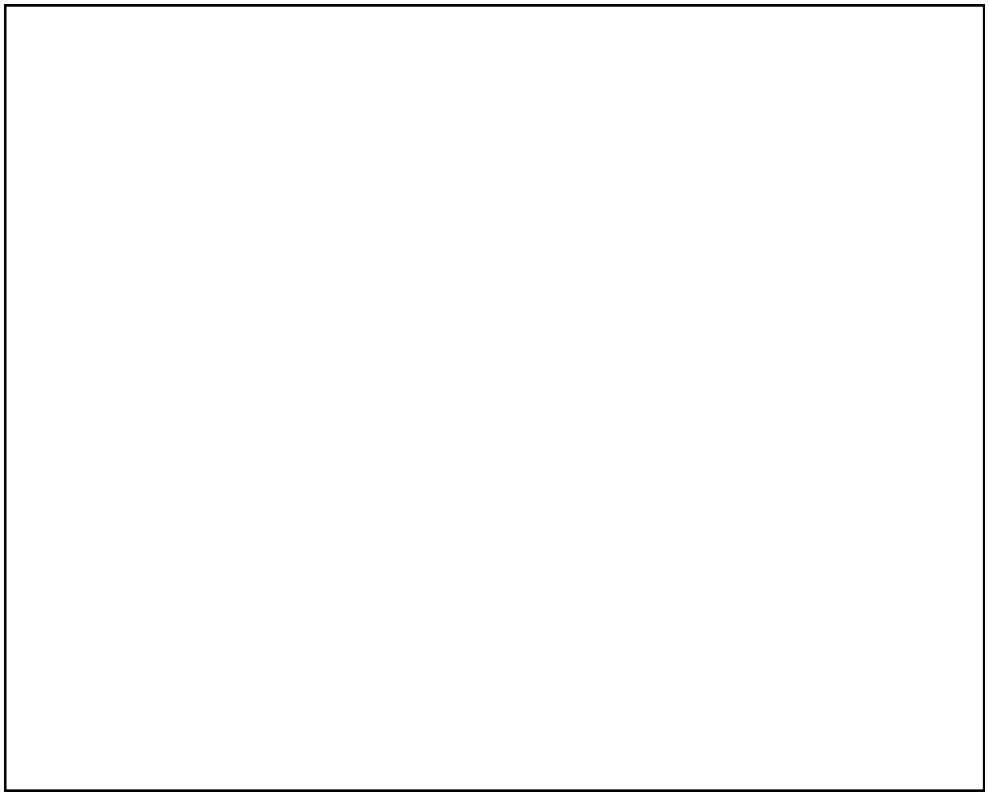
NOTES

SCAN ME!

# CARDINALITY

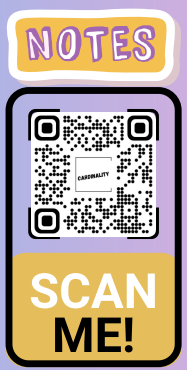
4 Draw an ERD of the data model described by the business rules below. Make sure the ERD has entities, attributes and cardinalities.

- A Hospital Management System consists of four entities that are Doctor, Patient, Ward, and Disease with identifying attributes for primary key that are Doc_Id, Patient_no, Ward_no and Disease_no.
- Doctor are described by Doc_name and Department, Patient by Patient_name and age, Ward by Ward_name and Disease by Disease_name and Price.
- One doctor will treat numerous patients and one patient can be treated by any doctors.
- Each patient can have multiple diseases and one disease can affect many patients.
- A patient will only be assigned into one ward and one ward can only be assigned to one patient.
- One doctor will handle a disease at a time and one disease can be handled by many doctors.




Simple Tips for Drawing ERDs

- **Identify Entities:** Use nouns to name real-world objects
- **Define Relationships:** Connect entities with lines to show how they relate
- **Add Attributes:** Include key details for each entity, like ID, Name, or Date.
- **Mark Primary Keys:** Use a unique identifier for each entity
- **Show Cardinality:**
  - one-to-one (1:1)
  - one-to-many (1:N),
  - many-to-many (M:N).
- **Keep It Clean:** Avoid crossing lines and keep the diagram simple for easy reading.



# CARDINALITY

5 Draw an ERD of the data model described by the business rules below. Make sure the ERD has entities, attributes and cardinalities.

- For each reality show, its name, genre, basic_info and participants.
- For each producer, the company name, company country. A show is produced by exactly one producer. And one producer produces exactly one show.
- For each television, its name, start year, head office. A television may broadcast multiple shows. Each show is broadcasted by exactly one television.
- For each user, his/her username, password, and age. A user may rate multiple shows, and a show may be rated by multiple users.

HELPFUL TIPS

Simple Tips for Drawing ERDs

- **Identify Entities:** Use nouns to name real-world objects
- **Define Relationships:** Connect entities with lines to show how they relate
- **Add Attributes:** Include key details for each entity, like ID, Name, or Date.
- **Mark Primary Keys:** Use a unique identifier for each entity
- **Show Cardinality:**
  - one-to-one (1:1)
  - one-to-many (1:N),
  - many-to-many (M:N).
- **Keep It Clean:** Avoid crossing lines and keep the diagram simple for easy reading.

NOTES

SCAN ME!

# CARDINALITY

6 Draw an ERD of the data model described by the business rules below. Make sure the ERD has entities, attributes and cardinalities.

- Each department can offer any number of courses and a course only can be offered by one department.
- Many instructors can work in a department. A department has many instructors.
- Each instructor can take any number of courses. A course can be taken by only one instructor.
- A student can enroll for any number of courses. Each course can have any number of students.
- Each student can be registered in a department and one department can have multiple students.

HELPFUL TIPS

Simple Tips for Drawing ERDs

- **Identify Entities:** Use nouns to name real-world objects
- **Define Relationships:** Connect entities with lines to show how they relate
- **Add Attributes:** Include key details for each entity, like ID, Name, or Date.
- **Mark Primary Keys:** Use a unique identifier for each entity
- **Show Cardinality:**
  - one-to-one (1:1)
  - one-to-many (1:N),
  - many-to-many (M:N).
- **Keep It Clean:** Avoid crossing lines and keep the diagram simple for easy reading.

# CARDINALITY

7 WonderPark wants to improve customer experience by offering targeted promotion for its members. As a database designer, you are assigned to design the new database system according to the following requirements:

- For each member we keep track of the unique MemID, as well as Name.
- For each membership we keep track of the unique M_ID, as well as MName and Price.
- For each pass category we keep track of the unique PassCatID and Data.
- For each one-day pass we keep track of the unique PassID and Date.
- For each merchandise item we keep track of the unique MercID, as well Price.
- Each member pays for exactly one membership type and each membership type can has many members.
- Each member can buy many one-day passes but each one-day pass can be bought by exactly one member

HELPFUL TIPS

Simple Tips for Drawing ERDs

- **Identify Entities:** Use nouns to name real-world objects
- **Define Relationships:** Connect entities with lines to show how they relate
- **Add Attributes:** Include key details for each entity, like ID, Name, or Date.
- **Mark Primary Keys:** Use a unique identifier for each entity
- **Show Cardinality:**
  - one-to-one (1:1)
  - one-to-many (1:N),
  - many-to-many (M:N).
- **Keep It Clean:** Avoid crossing lines and keep the diagram simple for easy reading.

# DEGREES OF RELATIONSHIP

1. List **THREE (3)** types of degree relationship.

.....

.....

.....

2. Explain **THREE (3)** types of degree relationship.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

HELPFUL TIPS

Remember the keywords of degree of relationship!

- Uni = One
- Bi = Two
- Ter = Three

NOTES

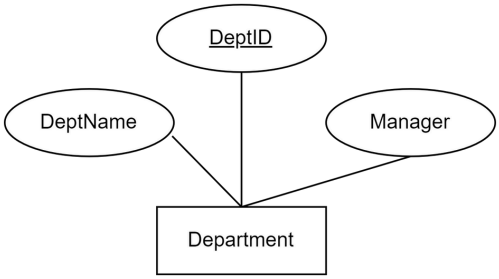
SCAN ME!

3. Draw a diagram for each of the types of degree relationship.

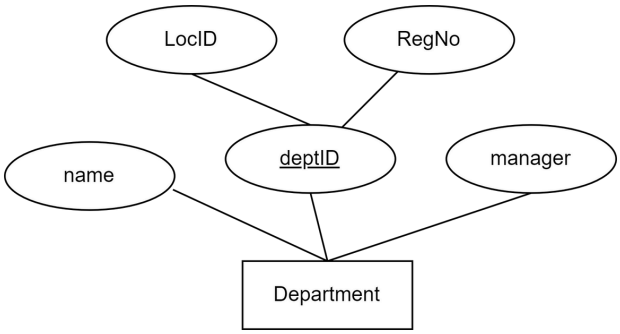
# RELATIONAL MODEL

## STRONG ENTITY

1




2




HELPFUL TIPS

Tips to Convert an ERD with a Strong Entity into a Relational Table:

- Require one table (if no multivalued attributes in the strong entity set)
- Remove derived attributes
- Take simple attributes of composite attributes

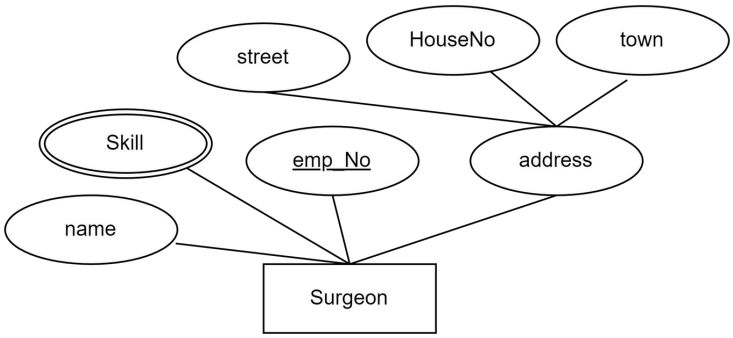
NOTES

SCAN ME!

# RELATIONAL MODEL

## STRONG ENTITIY

3




HELPFUL TIPS

Tips to Convert an ERD with a Strong Entity into a Relational Table:

- Require one table (if no multivalued attributes in the strong entity set)
- Remove derived attributes
- Take simple attributes of composite attributes

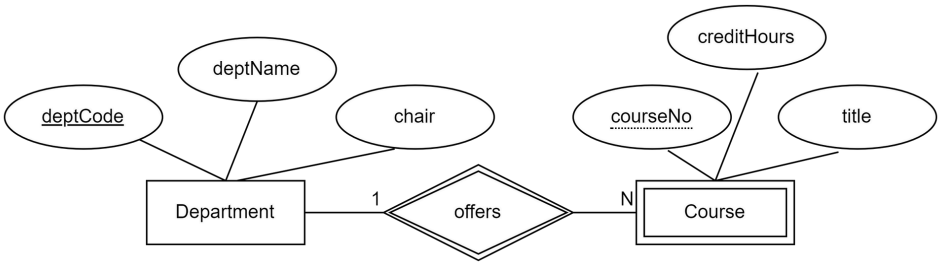
NOTES

SCAN ME!

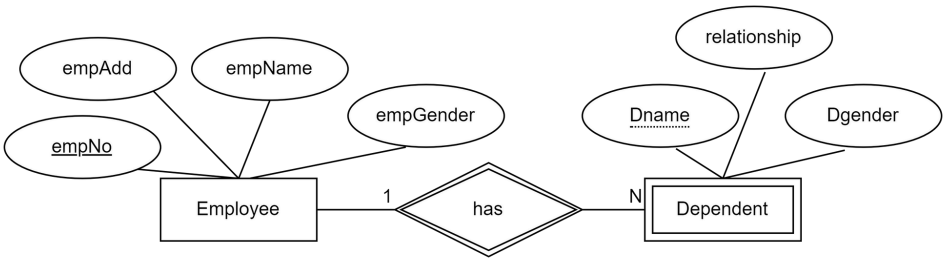
# RELATIONAL MODEL

## WEAK ENTITY

1




2




HELPFUL TIPS

Tips to Convert an ERD with a Weak Entity into a Relational Table:

- Require two tables
- Key attributes from both entities will be the composite key

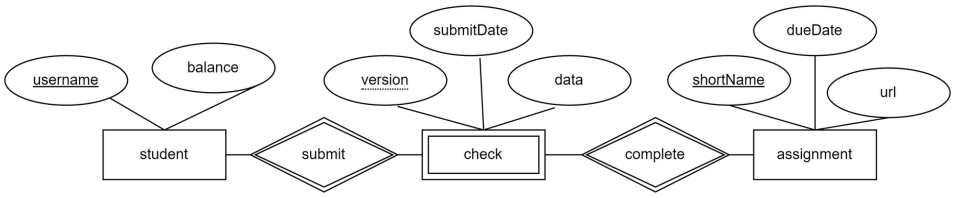
NOTES

SCAN ME!

# RELATIONAL MODEL

## WEAK ENTITY

3




HELPFUL TIPS

Tips to Convert an ERD with a Weak Entity into a Relational Table:

- Require two tables
- Key attributes from both entities will be the composite key

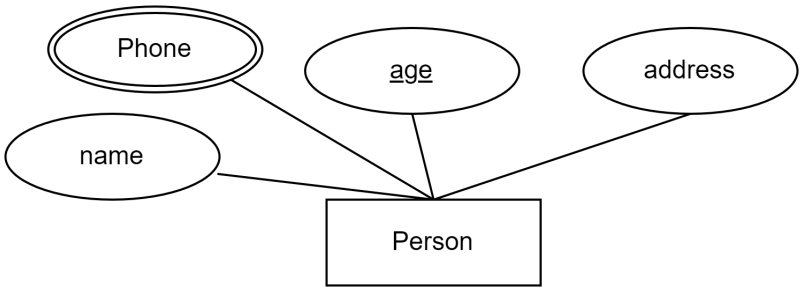
NOTES

SCAN ME!

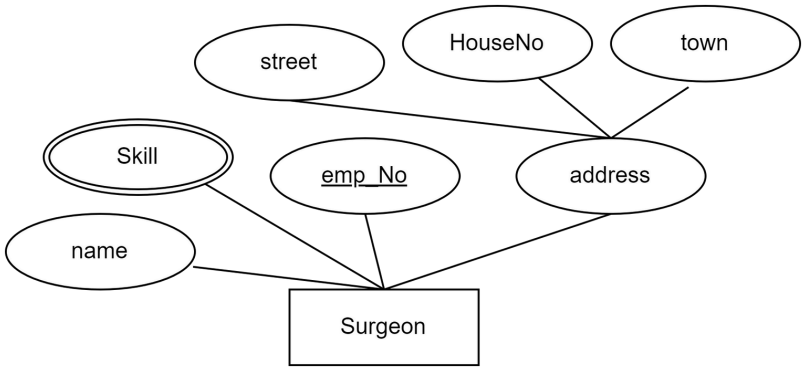
# RELATIONAL MODEL

## MULTIVALUED ATTRIBUTES

1




2




HELPFUL TIPS

Tips to Convert an ERD with a Multivalued Attributes into a Relational Table:

- Each multivalued attribute requires its own table
- Use the original entity's PK to connect the new table back to the entity

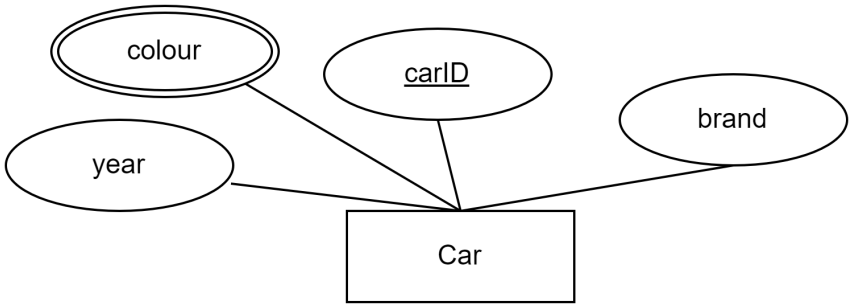
NOTES

SCAN ME!

# RELATIONAL MODEL

## MULTIVALUED ATTRIBUTES

3




HELPFUL TIPS

Tips to Convert an ERD with a Multivalued Attributes into a Relational Table:

- Each multivalued attribute requires its own table
- Use the original entity's PK to connect the new table back to the entity

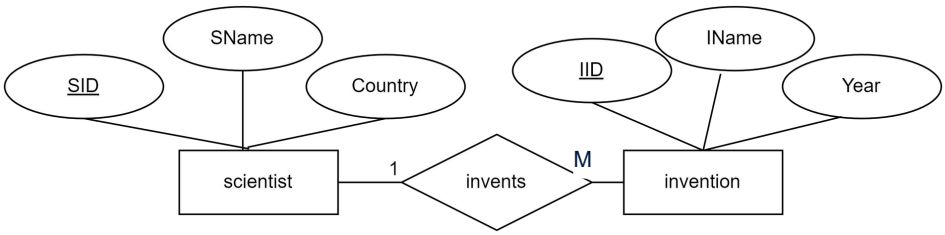
NOTES

SCAN ME!

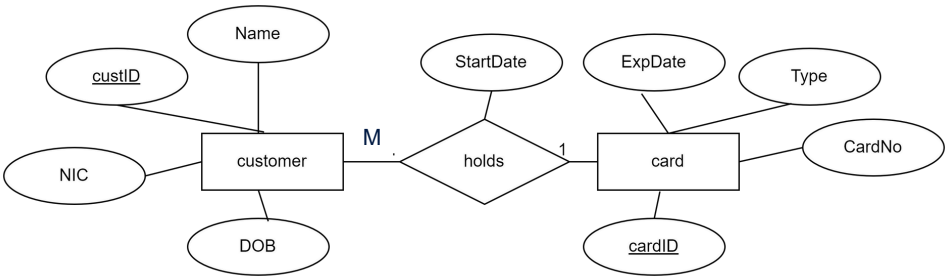
# RELATIONAL MODEL

## 1:N RELATIONSHIP

1




2




HELPFUL TIPS

Tips to Convert an ERD with a 1:N Relationship into a Relational Table:

- Primary Key (PK) - Unique identifier for the "one" side entity
- Foreign Key (FK) - Attribute added to the "many" side entity to reference the "one" side's primary key

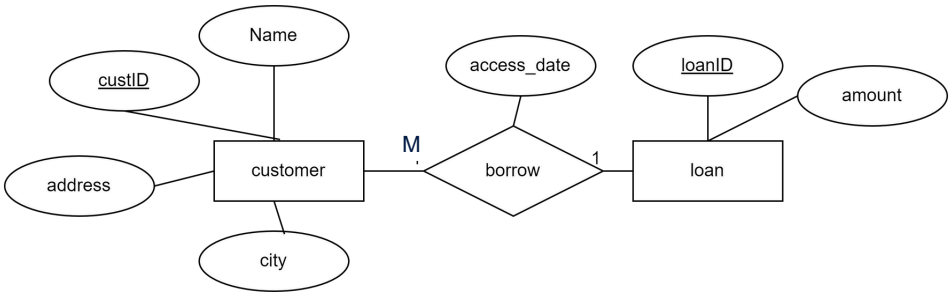
NOTES

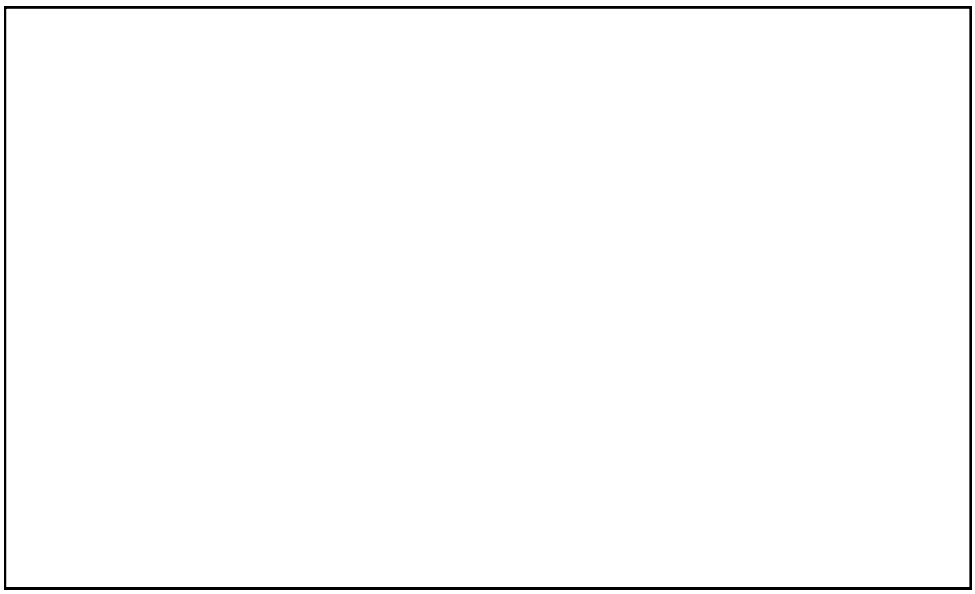
SCAN ME!

# RELATIONAL MODEL

## 1:N RELATIONSHIP

3



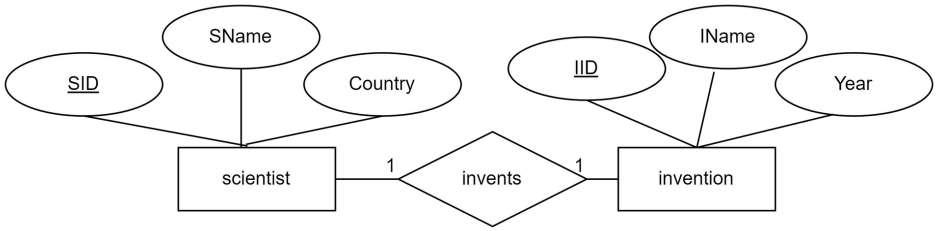

Tips to Convert an ERD with a 1:N Relationship into a Relational Table:

- Primary Key (PK) - Unique identifier for the "one" side entity
- Foreign Key (FK) - Attribute added to the "many" side entity to reference the "one" side's primary key

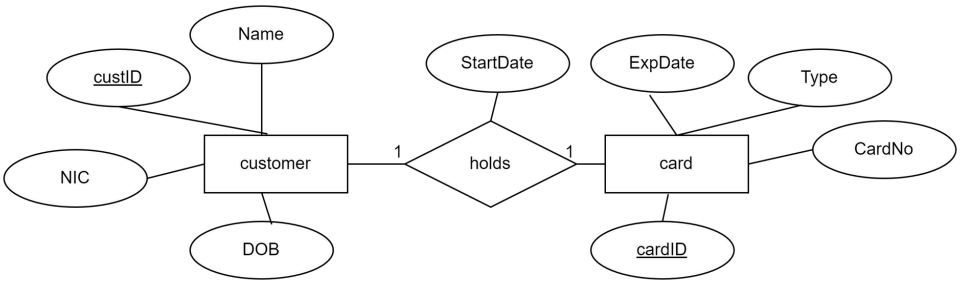
# RELATIONAL MODEL

## 1:1 RELATIONSHIP

1




2




HELPFUL TIPS

Tips to Convert an ERD with a 1:1 Relationship into a Relational Table:

- Three tables required
- Merge both entities into a single relation

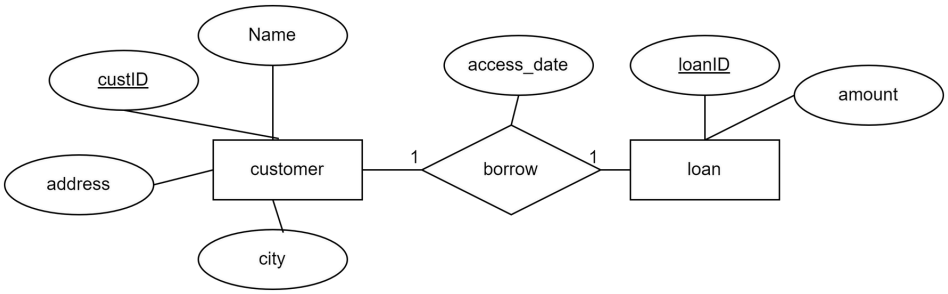
NOTES

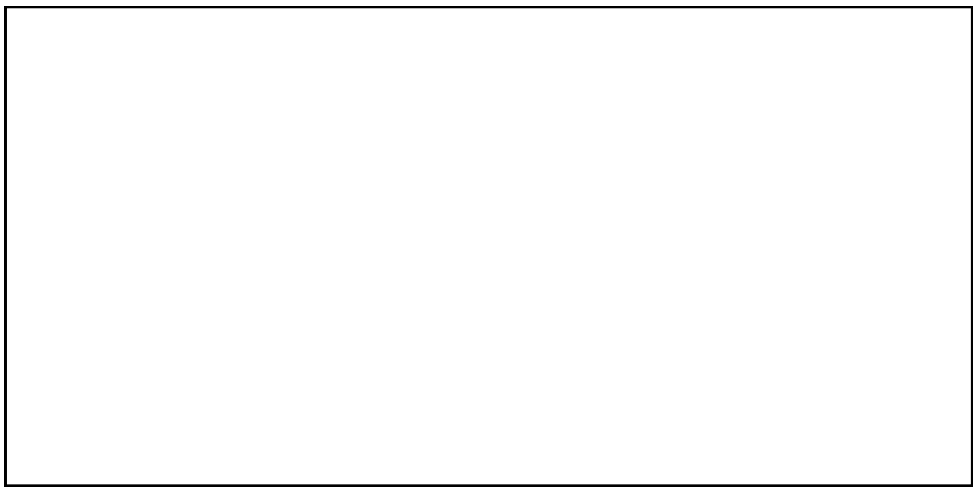
SCAN ME!

# RELATIONAL MODEL

## 1:1 RELATIONSHIP

3



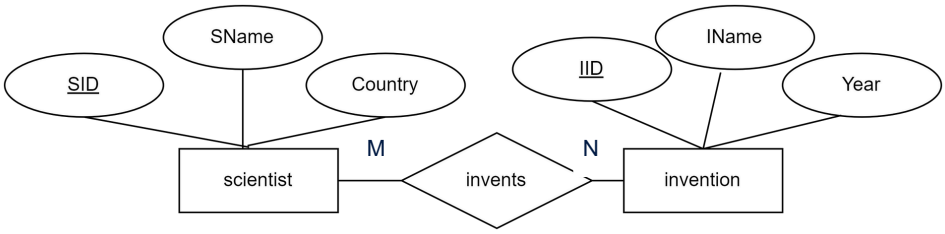

Tips to Convert an ERD with a 1:1 Relationship into a Relational Table:

- Three tables required
- Merge both entities into a single relation

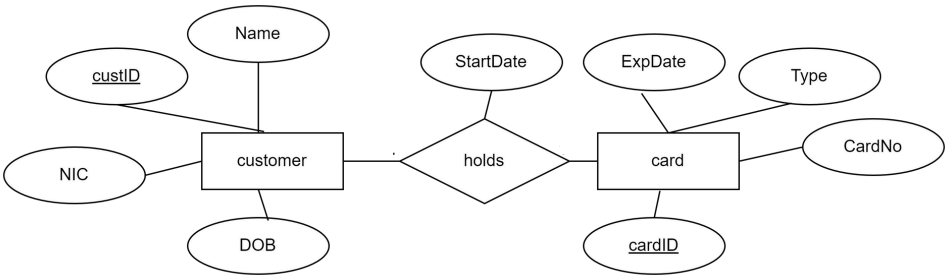
# RELATIONAL MODEL

## M:N RELATIONSHIP

1




2




HELPFUL TIPS

Tips to Convert an ERD with a M:N Relationship into a Relational Table:

- Join Table: Create a new relation to represent the relationship
- Primary Keys (PKs): Include the primary keys of both entities in the join table
- Composite Key: The combination of the primary keys serves as the primary key for the join table.
- Attributes: If the relationship has attributes, include them in the join table

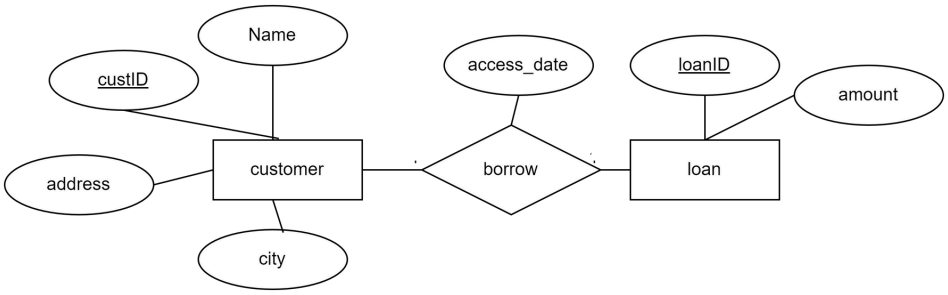
NOTES

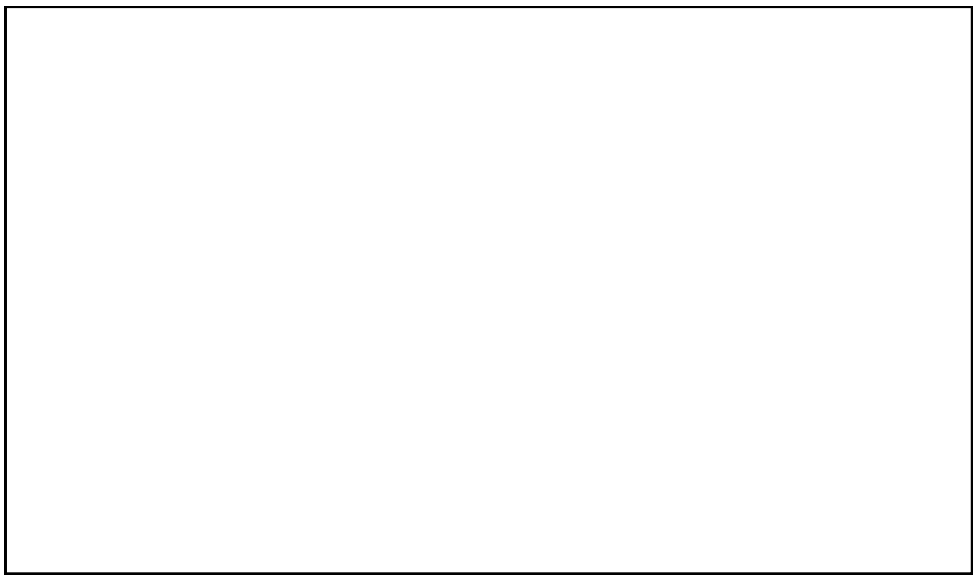
SCAN ME!

# RELATIONAL MODEL

## M:N RELATIONSHIP

3



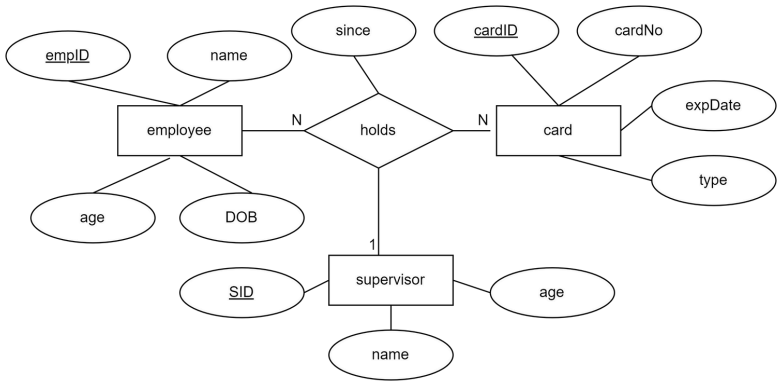

Tips to Convert an ERD with a M:N Relationship into a Relational Table:

- Join Table: Create a new relation to represent the relationship
- Primary Keys (PKs): Include the primary keys of both entities in the join table
- Composite Key: The combination of the primary keys serves as the primary key for the join table.
- Attributes: If the relationship has attributes, include them in the join table

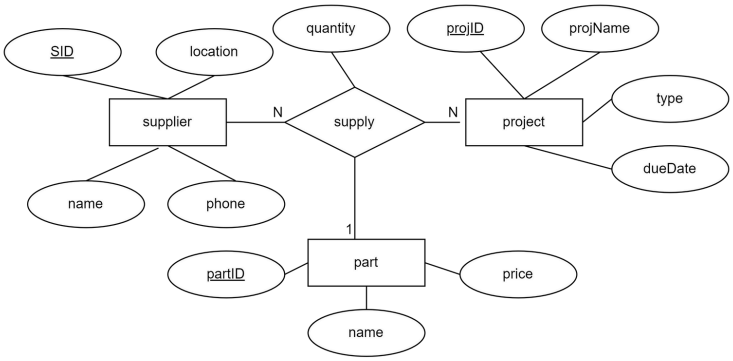
# RELATIONAL MODEL

## N-ary RELATIONSHIP

1




2




**HELPFUL TIPS**

Tips to Convert an ERD with a N-ary Relationship into a Relational Table:

- Create a new table to represent the N-ary relationship
- Include the primary keys of all participating entities as attributes in the new table
- Add any attributes of the relationship to the new table

NOTES

SCAN ME!

# DATABASE NORMALIZATION

1. Define database normalization.

.....

.....

.....

.....

.....

2. Discuss **THREE (3)** importances of database normalization.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

3. List **THREE (3)** types of normalization form in database.

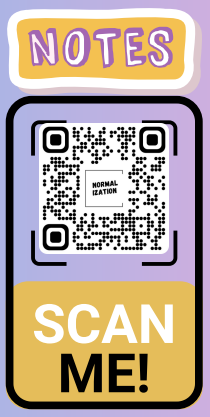
.....

.....

.....


Remember the keywords of normalization!

- Systematic approach to organizing data
- Minimize redundancy
- Divide into smaller table



# DATABASE NORMALIZATION

## 1NF

**1** Student Table

| StudentID | Name | Courses |
|---|---|---|
| 101 | Alice | Math, Science |
| 102 | Bob | English |

Explain why it does not satisfy 1NF.

.....

.....

Suggest how to normalize the table.

.....

.....

After normalization


Remember the keywords of 1NF!

- A single cell must not hold more than one value (atomic)
- Each record should be unique

# DATABASE NORMALIZATION

## 1NF

2

| empid | emp_name | emp_mobile | emp_skills |
|---|---|---|---|
| 1 | Reza | 0132685555 | Phyton, JavaScript |
| 2 | Harlina | 0165442334 | HTML, CSS, JavaScript |
| 3 | Suresh | 0186675223 | Java, Linux, C++ |

Explain why it does not satisfy 1NF.

.....

.....

Suggest how to normalize the table.

.....

.....

After normalization

HELPFUL TIPS

Remember the keywords of 1NF!

- A single cell must not hold more than one value (atomic)
- Each record should be unique

NOTES

SCAN ME!

# DATABASE NORMALIZATION

## 2NF

**1** StudentEnrolment Table

| StudentID | CourseID | CourseName | Instructor |
|---|---|---|---|
| 101 | CSE101 | Databases | Dr. Smith |
| 102 | CSE101 | Databases | Dr. Smith |
| 103 | CSE102 | Networks | Dr. Brown |

Explain why it does not satisfy 2NF.

.....

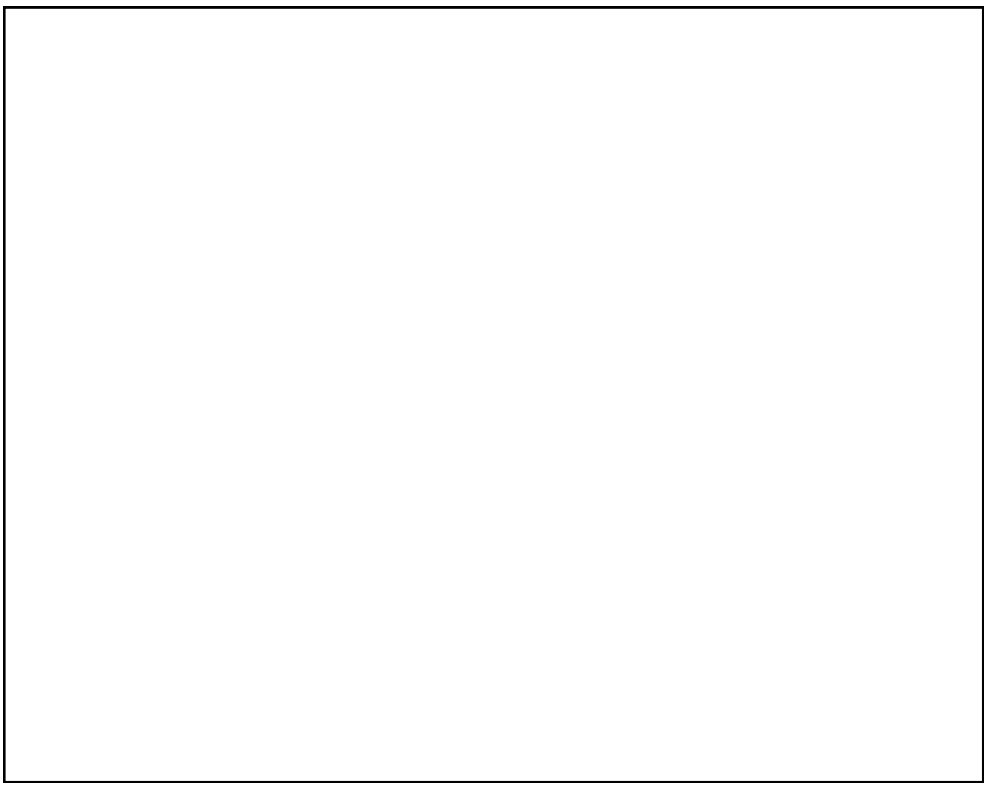
.....

Suggest how to normalize the table.

.....

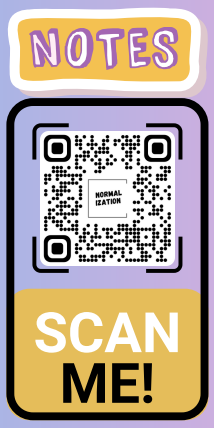
.....

After normalization




Remember the keywords of 2NF!

- Already in 1NF
- Has no partial dependency
- All non-key attributes are fully dependent on a primary key.



# DATABASE NORMALIZATION

## 2NF

**2**

Sales Table

| OrderID | ProductID | ProductName | CustomerID | CustomerName |
|---|---|---|---|---|
| 1 | 101 | Laptop | C001 | Alice |
| 2 | 102 | Keyboard | C002 | Bob |

Explain why it does not satisfy 2NF.

.....

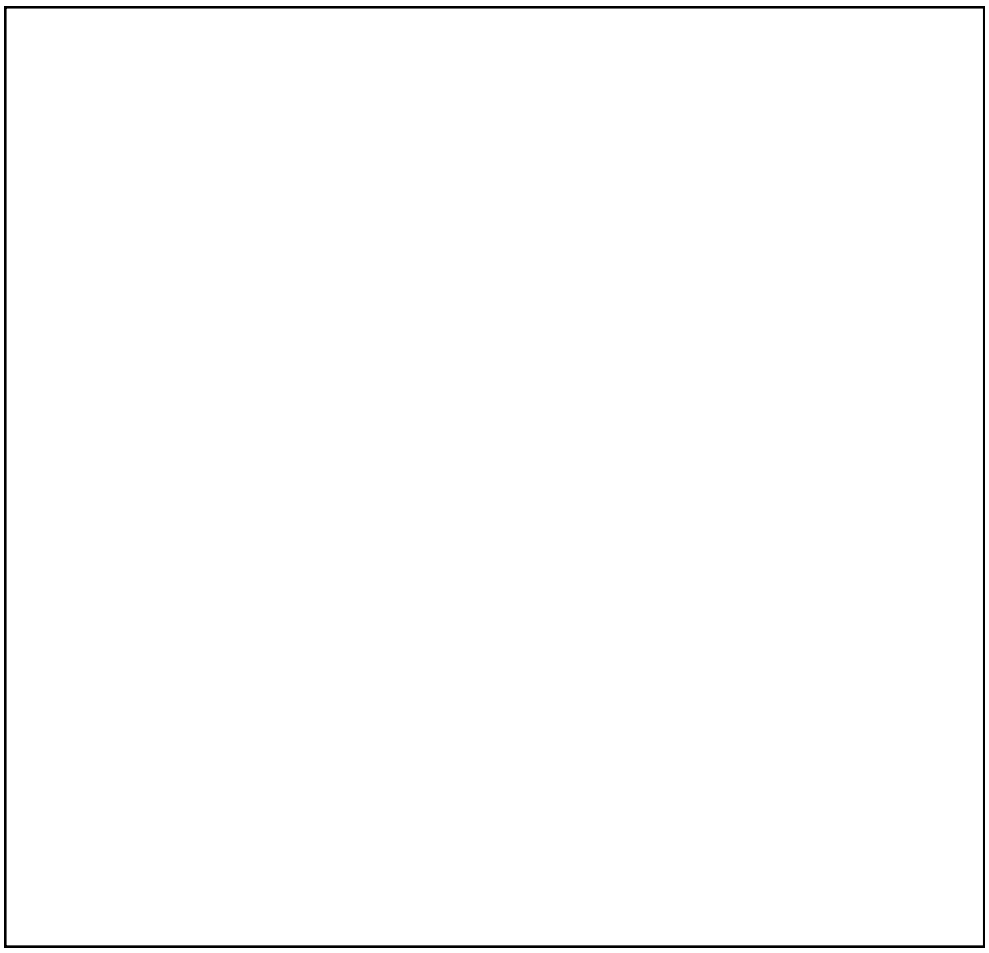
.....

Suggest how to normalize the table.

.....

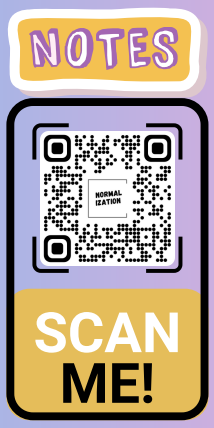
.....

After normalization




Remember the keywords of 2NF!

- Already in 1NF
- Has no partial dependency
- All non-key attributes are fully dependent on a primary key.



# DATABASE NORMALIZATION

## 2NF

3

| studname | studbirth | courseID | coursename | grade |
|---|---|---|---|---|
| ali | 1995 | IS102 | C++ | 70 |
| ali | 1995 | IS1205 | Web Design | 80 |
| alisa | 1995 | IS202 | Programming | 80 |
| alisa | 1995 | IS304 | DB | 79 |
| alisa | 1995 | IS102 | C++ | 85 |

Explain why it does not satisfy 2NF.

.....

.....

Suggest how to normalize the table.

.....

.....

After normalization

HELPFUL TIPS

Remember the keywords of 2NF!

- Already in 1NF
- Has no partial dependency
- All non-key attributes are fully dependent on a primary key.

NOTES

SCAN ME!

# DATABASE NORMALIZATION

## 3NF

1

Employee Table

| EmployeeID | DepartmentID | DepartmentName | ManagerName |
|---|---|---|---|
| 1 | 101 | HR | Alice |
| 2 | 102 | IT | Bob |

Explain why it does not satisfy 3NF.

.....

.....

Suggest how to normalize the table.

.....

.....

After normalization

HELPFUL TIPS

Remember the keywords of 2NF!

- Already in 2NF
- have no transitive partial dependency

NOTES

SCAN ME!

# DATABASE NORMALIZATION

## 3NF

2

| empID | empname | empposcode | empstate | empcity |
|---|---|---|---|---|
| 112 | Yahya | 47000 | Sg. Buloh | Selangor |
| 123 | Zaki | 66200 | Ipoh | Perak |
| 132 | Sathia | 45500 | Kuantan | Pahang |
| 122 | Cynthia | 47500 | Klang | Selangor |

Explain why it does not satisfy 3NF.

.....

.....

Suggest how to normalize the table.

.....

.....

After normalization

HELPFUL TIPS

Remember the keywords of 2NF!

- Already in 2NF
- have no transitive partial dependency

NOTES

SCAN ME!

# DATABASE NORMALIZATION

## 3NF

**3**

PatientVisit Table

| VisitID | PatientID | PatientName | DoctorID | DoctorName | Specialty |
|---|---|---|---|---|---|
| 1 | 101 | John | 201 | Dr. Smith | Cardiology |
| 2 | 102 | Mary | 202 | Dr. Brown | Neurology |

Explain why it does not satisfy 3NF.

.....

.....

Suggest how to normalize the table.

.....

.....

After normalization

HELPFUL TIPS

Remember the keywords of 2NF!

- Already in 2NF
- have no transitive partial dependency

NOTES

SCAN ME!

# DATABASE NORMALIZATION

## 3NF

**4** Shipment Table

| ShipmentID | ProductID | ProductName | SupplierID | SupplierName |
|---|---|---|---|---|
| S001 | P001 | Laptop | SUP01 | Supplier A |
| S002 | P002 | Phone | SUP02 | Supplier B |

Explain why it does not satisfy 3NF.

.....

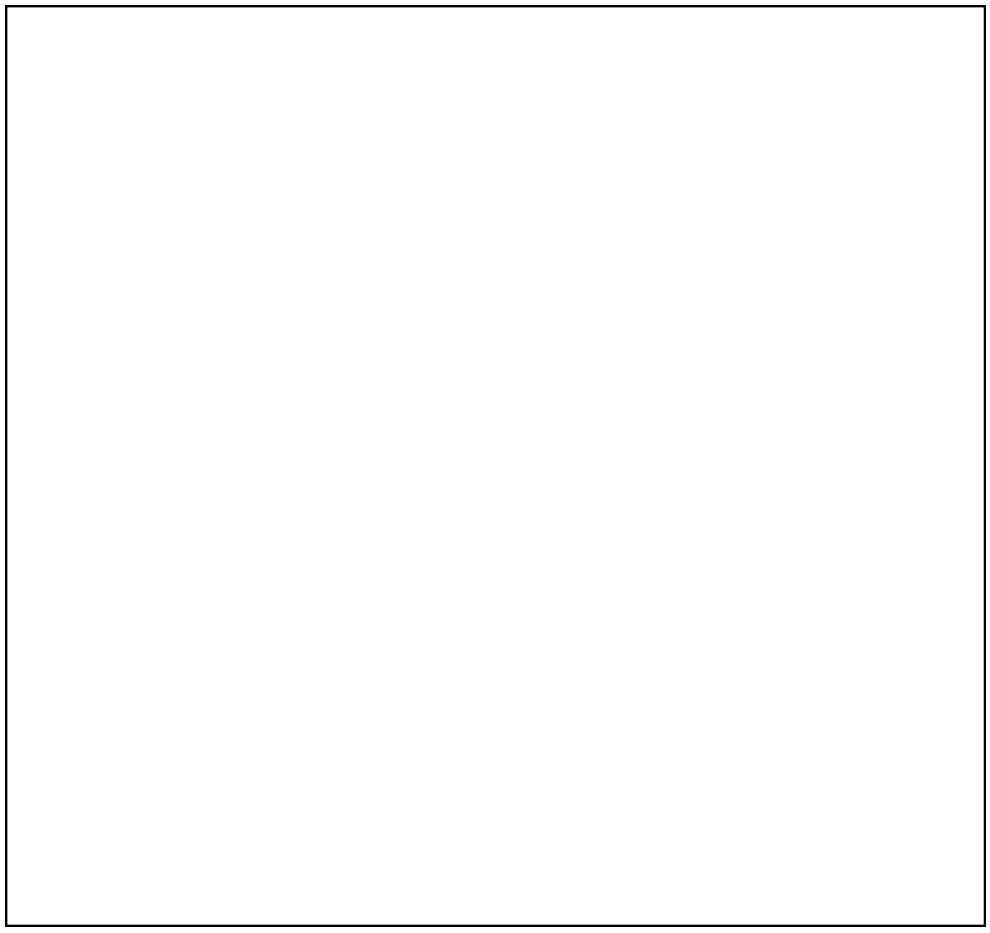
.....

Suggest how to normalize the table.

.....

.....

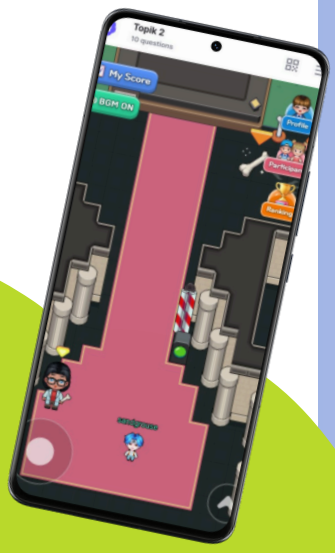
After normalization

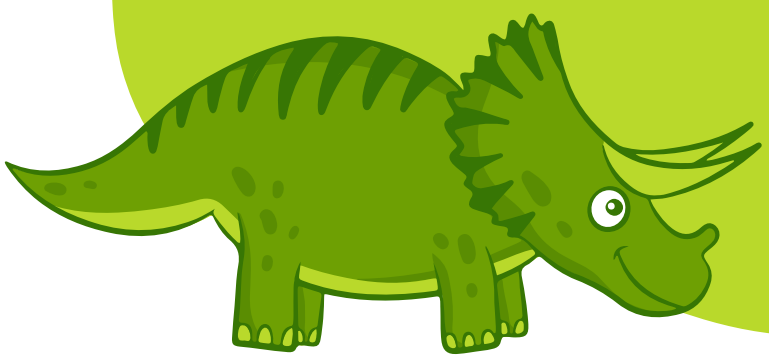



Remember the keywords of 2NF!

- Already in 2NF
- have no transitive partial dependency

# LET'S PLAY



# CHAPTER 3

## Structured Query Language (SQL)

# SQL CONCEPT

1. Explain Structured Query Language (SQL).

.....

.....

.....

.....

.....

2. List **FIVE (5)** functions of SQL.

.....

.....

.....

.....

.....

3. List **THREE (3)** types of data type.

.....

.....

.....

4. Recognize the suitable data type and size for the attributes given.

Matric Number : .....

Date of Birth : .....

Phone Number : .....

IC Number : .....

Marks : .....

Price : .....

Name : .....

Email : .....

HELPFUL TIPS

Remember the keywords of SQL!

- Programming language
- Managing and manipulating relational databases
- Used to communicate with a database

NOTES

SCAN ME!

# SQL CONCEPT

5 Define SQL constraints.

.....

.....

.....

.....

.....

6 List **FIVE (5)** SQL constraints.

.....

.....

.....

.....

.....

7 Explain PRIMARY KEY, FOREIGN KEY and NOT NULL.

.....

.....

.....

.....

.....

.....

HELPFUL TIPS

Remember the keywords of SQL constraints!

- Rules applied to columns in a table

NOTES

SCAN ME!

# DATA DEFINITION LANGUAGE (DDL)

1. Explain Data Definition Language (DDL) functions in SQL.

.....

.....

.....

2. State the usage for each of the command below.

CREATE :.....

DROP :.....

ALTER :.....

3. Discuss the implications of using the DROP statement in a database. What precautions should be taken before executing this command?

.....

.....

.....

.....

4. Identify the objects can be created using CREATE statement.

.....

.....

4. Discuss the role of ALTER command in helping modifying database structures. Provide examples of common modifications.

.....

.....

.....

.....


Remember the keywords of DDL!

- SQL command
- Define, create and manipulate the structure of databases

# DATA DEFINITION LANGUAGE (DDL)

## CREATE

1. Write an SQL statement to create a database named movie.

2. Write an SQL statement to create a table named countries including columns country_id, country_name and region_id.

3. Write an SQL statement to create a table employees including columns employee_id, first_name, last_name, email, phone_number hire_date, job_id, salary. Set a constraint NOT NULL.

4. Write an SQL statement to create a table named gadgets which should include columns gadget_id, gadget_name, and gadget_type Set gadget_id as primary key.

ANSWER



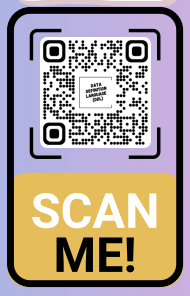


HELPFUL TIPS

CREATE DATABASE db_name;

CREATE TABLE with columns + types

NOTES



SCAN ME!

# DATA DEFINITION LANGUAGE (DDL)

## CREATE

5 Write an SQL statement to create a table named patient that contains five columns; PatientID, PatientName, Address, Disease, AppointmentID. Set PatientID as primary key and AppointmentID as foreign key.

6 Write an SQL statement to create a simple table Songs which should include columns song_id, song_name, song_type and song_rating, artist_id. Set song_id as primary key and artist_id as foreign key.


CREATE TABLE with proper column names & data types

data types:
INT, VARCHAR, DATE, etc.

# DATA DEFINITION LANGUAGE (DDL)

## ALTER & DROP

Given a table named Book in a database named Book_Store as below.

Book

| Column Name | Data Type |
|---|---|
| BookID | Integer |
| Title | Varchar (100) |
| ISBN | Varchar (20) |
| PublicationYear | Date |

HELPFUL TIPS

✅The ALTER TABLE statement is used to add, delete, or modify columns in an existing table

✅The ALTER TABLE statement is also used to add and drop various constraints on an existing table.

1. Write an SQL statement to add new column named Price.

2. Write an SQL statement to add new column named PublisherAdd after column Title.

3. Write an SQL statement to change data type of the PublicationYear column to YEAR.

4. Write an SQL statement to change column name of the ISBN column to ISBN_no.

NOTES

SCAN ME!

# DATA DEFINITION LANGUAGE (DDL)

## ALTER & DROP

5. Write an SQL statement to change data type size of the Title column to the size of 150.

6. Write an SQL statement to delete a column named PublisherAdd.

7. Write an SQL statement to rename the table Book to Books.

8. Write an SQL statement to add BookID as primary key.

9. Write an SQL statement to delete the table Books.

10. Write a SQL statement to delete the database Book_Store.

HELPFUL TIPS

✅ **Add new column**:
ALTER TABLE pelajar ADD email VARCHAR(50);

✅ **Change column:**
ALTER TABLE pelajar MODIFY umur TINYINT;
ALTER TABLE pelajar RENAME COLUMN nama TO nama_penuh;

✅ **Delete column:**
ALTER TABLE pelajar DROP COLUMN email;

✅ **Rename table:**
RENAME TABLE pelajar TO data_pelajar;

NOTES

SCAN ME!

# DATA DEFINITION LANGUAGE (DDL)

## MySQL Lab Exercise 3.1


Use an appropriate DBMS software

1. Create a database named NilaiLibrary.
2. Create a table named Books with the following columns:
    a. BookID (integer, primary key)
    b. Title (varchar (50), NOT NULL)
    c. Author (varchar (20), NOT NULL)
    d. PublishedYear (year, NOT NULL)
3. Create a table named Member with the following columns:
    a. MemberID (integer, primary key)
    b. Name (varchar(100))
    c. Email (varchar(100), unique)
4. Create a table named Loans to track which members borrow which books. Include the following columns:
    a. LoanID (integer, primary key)
    b. BookID (integer, foreign key referencing Books(BookID))
    c. MemberID (integer, foreign key referencing Members(MemberID))
    d. LoanDate (date)
    e. ReturnDate (date)
5. Create a table named Categories to categorize books, with the following columns:
    a. CategoryID (integer, primary key)
    b. CategoryName (varchar(10))
6. Change table name Member to Members.
7. Add **TWO (2)** new columns named Publisher and Price in Books table.
8. Change the size of the data type of the CategoryName column in the Categories table to 50.
9. Change the column name Name to MemberName in the Member table.
10. Delete the column named ReturnDate in the Loans table.

# DATA MANIPULATION LANGUAGE (DML)

1. Explain Data Manipulation Language (DML) functions in SQL.

.....

.....

.....

2. State the usage for each of the command below.

INSERT :.....

DELETE :.....

UPDATE :.....

3. Discuss the implications of using the DELETE statement in a database. What precautions should be taken before executing this command?

.....

.....

.....

.....

4. Discuss the implications of using the DELETE statement in a database. What precautions should be taken before executing this command?

.....

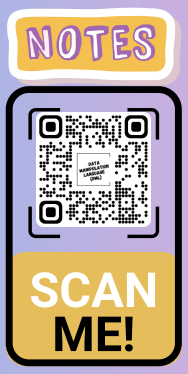
.....

.....

.....


Remember the keywords of DML!

- SQL command
- Make changes to the records in a table
- Add, delete, update records



# DATA MANIPULATION LANGUAGE (DML)

## INSERT

1. Write an SQL statement to a new record into a table named Students with the following columns:
   - StudentID = 1
   - Name = 'Huzaifah'
   - Age = 20

2. Data are given in the table below.

Cats:

| CatID | Name | Breed | Age |
|---|---|---|---|
| C1 | Jellybean | Persian | 4 |
| C2 | Smokey | Scottish Fold | 2 |
| C3 | Cleopatra | Siamese | 6 |
| C4 | Catniss | Burmese | 8 |

Write an SQL statement to insert FOUR (4) rows by a single insert statement.

ANSWER

HELPFUL TIPS

Remember the keywords of DML!
- SQL command
- Make changes to the records in a table
- Add, delete, update records

NOTES

SCAN ME!

# DATA MANIPULATION LANGUAGE (DML)

## INSERT

**3** Data are given in the table below.

Countries

| Column Name | Data Type |
|---|---|
| CountryID | Integer |
| CountryName | Varchar (50) |
| Region | Varchar (100) |

Write an SQL statement to insert **ONE (1)** row into the table Countries for the columns CountryID and CountryName.

**4** Write a query to insert multiple records into a table named Products with the columns ProductID, ProductName, and Price.


✅ **Insert a single row**

INSERT INTO student (name, age) VALUES ('Ali', 20);

✅ **Insert multiple rows**

INSERT INTO student (name, age)
VALUES ('Sara', 21), ('Zul', 22);

# DATA MANIPULATION LANGUAGE (DML)

## UPDATE & DELETE

Data are given in the table below.

Employee

| ID | Name | Age | Address | Position | Salary |
|---|---|---|---|---|---|
| 1 | Khalish | 32 | Kangar | Sales Executive | 2000.00 |
| 2 | Helmi | 25 | Pendang | Clerk | 2000.00 |
| 3 | Farzana | 23 | Batu Pahat | Executive | 2000.00 |
| 4 | Morgan | 25 | Kuantan | Analyst | 6500.00 |
| 5 | Sew Chin | 27 | Ipoh | Manager | 8500.00 |
| 6 | Yumna | 22 | Dungun | Junior Analyst | 4500.00 |
| 7 | Zira | 24 | Pasir Mas | Senior Manager | 10000.00 |

HELPFUL TIPS

✅ Update specific data

UPDATE student SET age = 21 WHERE id = 1;

✅ Always use WHERE! Without it, all rows may be changed.

NOTES

SCAN ME!

1. Write an SQL statement to update the salary for the position of Junior Analyst to 5000.00.

2. Write an SQL statement to update the position of the salary of 2000.00 to Sales Executive.

3. Write an SQL statement to update the Address column to 'Not Available' for employees age is 25.

# DATA MANIPULATION LANGUAGE (DML)

## UPDATE & DELETE

4. Write an SQL statement to update the address of the employee named Yumna to Batu Gajah.

5. Write an SQL statement to update the Age column to 30 for all records in the Employees table.

6. Write an SQL statement to delete records of the employee named Helmi.

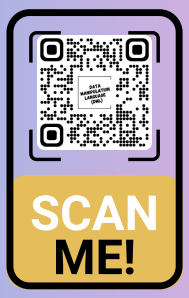
7. Write an SQL statement to delete all records of the employee in the Employee table.


✅ **Delete specific row**

```
DELETE FROM student
WHERE id = 1;
```

✅ **To delete all data**

```
DELETE FROM student;
```

# DATA MANIPULATION LANGUAGE (DML)

## MySQL Lab Exercise 3.2


Use an appropriate DBMS software

DOWNLOAD

https://bit.ly/productsql

Import **product.sql** file into your DBMS.

1. Add a new product, "Air Fryer," under the "Appliances" category, priced at 100.00, with a stock quantity of 35, supplied by "KitchenAid," and added on '2025-05-01'.
2. Add a new product, "Smartwatch," under the "Electronics" category, priced at $150.00, with a stock quantity of 18, supplied by "MobileMart," and added on '2025-05-10'.
3. Update the price of the product "Gaming Console" to 350.00.
4. Increase the StockQuantity of the product "Coffee Maker" by 20.
5. Change the supplier of the product "Tablet" from "TechWorld" to "MobileMart".
6. Update the StockQuantity of the product "LED TV" to 12, and change its price to $480.00.
7. Remove the product "Blender" from the Products table.
8. Delete all products under the "Furniture" category from the Products table.
9. Delete all products that were added before '2025-02-01' from the Products table.
10. Delete all products with a stock quantity less than 10 from the Products table.

# DATA QUERY LANGUAGE (DQL)

1. Explain Data Query Language (DQL) functions in SQL.

.....

.....

.....

2. State the usage for SELECT command.

.....

.....

3. Describe the difference between SELECT * and SELECT column_name in SQL.

.....

.....

.....

.....

.....

4. Discuss the role of the WHERE clause in a SELECT statement.

.....

.....

.....

.....

5. Discuss the role of the ORDER BY clause in SQL.

.....

.....

.....

.....


Remember the keywords of DQL!

- SQL command
- Query and retrieve data
- Focused on reading data

# DATA QUERY LANGUAGE (DQL)

## SELECT

1. Write an SQL statement to retrieve all columns of data from the Employees table.

2. Write an SQL statement to select the StudentID, FirstName, and LastName columns.

3. Write an SQL statement to get distinct department names from the Staff table.

4. Write an SQL statement to select specific columns from a table Books, displaying BookID, Title, and Author.

5. Write an SQL statement to retrieve all distinct values from the City column in a table called Addresses.


✅ **Basic query**
SELECT * FROM student;
✅ **Choose specific columns**
SELECT name, age FROM student;

# DATA QUERY LANGUAGE (DQL)

## SELECT - WHERE

1. Write an SQL statement to retrieve all records from a table called Orders where the OrderDate is after '2025-01-01'.

2. Write an SQL statement to retrieve all records from a table called Products where the Price is greater than 100 and the Category is "Gadget".

3. Write an SQL statement to retrieve all records from the Customers table where the City is "Kota Kinabalu".

4. Write an SQL statement to select the product names and prices for products where the StockQuantity is between 10 and 50.

5. Write an SQL statement to select the EmpName and Address of employees with a Age is between 20 to 25.


✅ **Filter rows with WHERE**

SELECT * FROM student
WHERE age > 20;

# DATA QUERY LANGUAGE (DQL)

## SELECT WITH 'AND'/ 'OR' OPERATOR

1. Write an SQL statement to retrieve all records from the Customers table where the City is either "Gopeng" or "Alor Setar".

2. Write an SQL statement to retrieve all records from the Students table where the Age is greater than 18 and the Gender is "Male".

3. Write an SQL statement to select all columns from the Orders table where the OrderDate is after '2023-01-01' and the Amount is greater than 500.

4. Write an SQL statement to select all employees from the Employees table where the Salary is greater than 5000 or the JobTitle is "Manager".

5. Write an SQL statement to retrieve the OrderID, ProductName, and Quantity from the OrderDetails table where the OrderID is either 101 or 102 and the Quantity is more than 10.


✅ **Use AND to match all conditions**
SELECT * FROM student WHERE course = 'DBMS' AND age > 20;

✅ **Use OR to match any condition**
SELECT * FROM student WHERE course = 'DBMS' OR age > 20;

✅ **Combine AND & OR using brackets**
SELECT * FROM student WHERE course = 'DBMS' AND (age > 20 OR name = 'Ali');

# DATA QUERY LANGUAGE (DQL)

## SELECT WITH ORDER BY CLAUSE

1. Write an SQL statement to retrieve all records from a table called Employees, sorted by LastName in ascending order.

2. Write an SQL statement to retrieve all products from a table called Products, sorted by Price in descending order.

3. Write an SQL statement to select StudentID, FirstName, and LastName from the Students table, ordered by Grade in descending order.

4. Write an SQL statement to retrieve all columns from the Orders table, ordered by OrderDate from the most recent to the oldest.

5. Write an SQL statement to select CustomerID and Name from the Customers table and order them by Name in alphabetical order.


✅ **Sort results in ascending (default) order**
SELECT * FROM student
ORDER BY age;

✅ **Sort in descending order**
SELECT * FROM student
ORDER BY age DESC;

# DATA QUERY LANGUAGE (DQL)

## SELECT WITH GROUP BY CLAUSE

1. Write an SQL statement to retrieve the total number of employees in each department from the Employees table.

2. Write an SQL statement to select the Category and the average Price of products in each category from the Products table.

3. Write an SQL statement to retrieve the City and the number of customers from the Customers table for each city.

4. Write an SQL statement to retrieve the ProductName and the highest quantity sold for each product from the OrderDetails table.

5. Write an SQL statement to retrieve the Department and the number of employees in each department who have a salary above 5000 from the Employees table.


✅ **Group rows with the same value**
SELECT course, COUNT(*)
FROM student
GROUP BY course;

✅ **Use with aggregate functions**
Like COUNT(), SUM(), AVG(), MAX(), MIN()

✅ **The column in GROUP BY must also be in SELECT**

# DATA QUERY LANGUAGE (DQL)

## SELECT WITH 'IN' & 'LIKE' OPERATOR

1. Write an SQL statement to retrieve all records from the Employees table where the Department is either "Sales" or "Marketing".

2. Write an SQL statement to retrieve the StudentID, FirstName, and LastName from the Students table where the Grade is either 'A', 'B', or 'C'.

3. Write an SQL statement to retrieve the CustomerID and Name from the Customers table where the Name starts with the letter 'A'

4. Write an SQL statement to select EmployeeID and FirstName from the Employees table where the FirstName contains the substring 'Johan'.

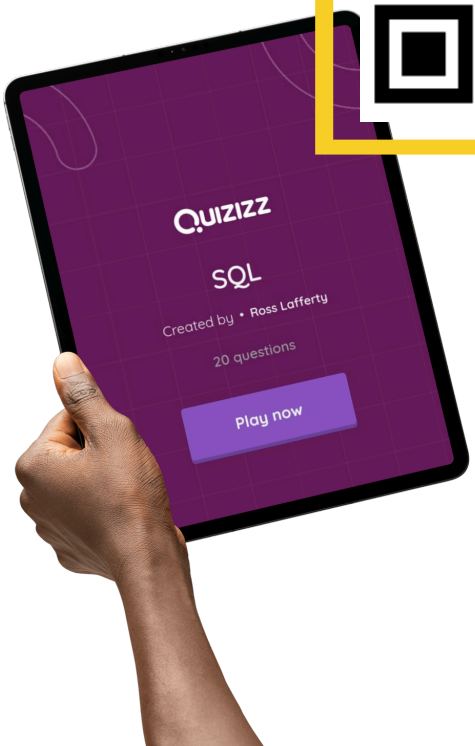
5. Write an SQL statement to retrieve the StudentID, FirstName, and LastName from the Students table where the LastName ends with the letter 'M'.


✅ **Use IN to match multiple values**
SELECT * FROM student WHERE course IN ('DBMS', 'Math', 'IT');

✅ **Use LIKE to search patterns**

# QUIZ time!

**SCAN HERE**



# DATA QUERY LANGUAGE (DQL)

## MySQL Lab Exercise 3.3


Use an appropriate DBMS software

DOWNLOAD

https://bit.ly/studentsql

Import **student.sql** file into your DBMS.

1. Retrieve all columns from the Students table.
2. Select the FirstName, LastName, and GPA of students.
3. Find students whose GPA is greater than 3.5.
4. Retrieve the names of students who are majoring in 'Business'.
5. Select FirstName and GPA for students whose GPA is between 3.0 and 3.5.
6. Find all students who are 22 years old or older.
7. Select FirstName, LastName, and Major for students whose first name starts with the letter 'A'.
8. Retrieve all students whose first name starts with 'N' and is followed by exactly three characters.
9. Find students whose last name contains the letter 'i'.
10. Select students whose first name starts with 'S' and last name starts with 'A'.
11. Retrieve the details of students majoring in either 'Computer Science' or 'Psychology'.
12. Retrieve the details of students, ordered by GPA in descending order.
13. Retrieve the details of students, ordered by Age in ascending order.
14. Find the average GPA of students for each Major and order the result by GPA in descending order.
15. Select the names and ages of students who are 21 years old or older and order by Age in ascending order.
16. Find the highest GPA in the Students table.
17. Select the names of students whose first name starts with 'A' and calculate the average GPA for these students.

# CHAPTER 4

## Functions in Structured Query Language (SQL)

# AGGREGATE FUNCTIONS

1. Explain the purpose of aggregate functions in SQL.

.....

.....

.....

2. State the usage for each of the aggregate functions below.

SUM :.....

AVG :.....

COUNT :.....

MIN :.....

MAX :.....

3. Describe the difference between COUNT(*) and COUNT(column_name) in SQL.

.....

.....

.....

.....

.....

4. Discuss the difference between calculating the total using SUM(column_name) and finding the count of rows with COUNT(column_name).

.....

.....

.....

.....


Remember the keywords of aggregate functions!

- SQL command
- Perform calculations on multiple rows
- Return a single result

# AGGREGATE FUNCTIONS

Data are given in the table below.

Stock

| stock_id | quantity | collection | price | category |
|---|---|---|---|---|
| M110 | 50 | Adi Collection | 100.00 | Baju Melayu |
| K112 | 30 | Amara Fashion | 80.00 | Baju Kurung |
| R132 | 20 | CYD | 50.00 | Kurta |
| D123 | 60 | Meena | 90.00 | Jubah |
| K113 | 40 | Arafah Design | 120.00 | Baju Kurung |

HELPFUL TIPS

✅ **Use SUM to total values**
Adds up all values in the marks column.
✅ **Use AVG to find the average**
Calculates the mean of all marks.

NOTES

SCAN ME!

1. Write an SQL statement to calculate the total quantity of all items across all categories.

2. Write an SQL statement to find the average price of all products in the Stock table.

3. Write an SQL statement to count the number of unique categories in the Stock table.

# AGGREGATE FUNCTIONS

4. Write an SQL statement to calculate the total quantity of products in the "Baju Kurung" category.

5. Write an SQL statement to calculate the total quantity of products in stock for each category.

6. Write an SQL statement to retrieve the highest price of any product in the Stock table.


✅**SUM**
SELECT SUM(marks) AS total_marks
FROM student;
✅ **AVG**
SELECT AVG(marks) AS average_marks
FROM student;
✅ **Can combine with GROUP BY**
SELECT course, AVG(marks)
FROM student
GROUP BY course;

# AGGREGATE FUNCTIONS

Data are given in the table below.

Inventory

| item_id | product_name | quantity | supplier | price | category | warehouse_location | status |
|---|---|---|---|---|---|---|---|
| A101 | Smartphone Pro | 40 | Super Supplier | 150.00 | Electronics | Section A3 | In Stock |
| B202 | T-shirt Deluxe | 60 | MegaMart | 120.00 | Clothing | Section B1 | In Stock |
| C303 | Designer Watch | 30 | Fashion Hub | 80.00 | Accessories | Section C2 | In Stock |
| D404 | Office Chair | 50 | Home Needs | 100.00 | Furniture | Section D1 | In Stock |
| E505 | LED TV | 70 | Super Supplier | 200.00 | Electronics | Section A5 | In Stock |


✅ Use MIN to find the smallest value

✅ Use MAX to find the largest value


1. Write an SQL statement to find the average price of items across all suppliers.

2. Write an SQL statement to retrieve the minimum quantity of items for each supplier.

3. Write an SQL statement to calculate the total quantity of items stored in "Section A3" and "Section A5".

4. Write an SQL statement to find the average price of items supplied by "Super Supplier".

# AGGREGATE FUNCTIONS

5. Write an SQL statement to determine the total quantity of items grouped by their category.

6. Write an SQL statement to find the maximum price of items grouped by their supplier.

7. Write an SQL statement to count the number of unique categories available in the Inventory table.

8. Find the highest quantity of items available in the "Electronics" category

9. Find the total quantity of items for each warehouse location.

10. Count the number of items where the status is "In Stock"


✅ **MIN**

```
SELECT MIN(age) AS
youngest
FROM student;
```

✅ **MAX**

```
SELECT MAX(age) AS
oldest
FROM student;
```

# AGGREGATE FUNCTIONS

## MySQL Lab Exercise 4.1


Use an appropriate DBMS software

DOWNLOAD

https://bit.ly/moviesql

Import **movie.sql** file into your DBMS.

1. Find the average rating of all movies.
2. Find the highest rating among all movies.
3. Find the total duration of all movies combined.
4. Find the movie with the longest duration.
5. Find the shortest duration among all movies.
6. Find the average duration of movies by genre.
7. Find the number of movies released each year.
8. Find the highest rating in each genre.
9. Find the average rating for movies released after the year 2000.
10. Find the average rating and average duration of movies released before the year 2000.
11. Find the average duration for movies released in the 2000s.
12. ind the number of movies that have a rating greater than or equal to 9.0.

# STRING FUNCTIONS

1. Explain the purpose of string functions in SQL.

.....

.....

.....

2. State the usage for each of the string functions below.

LCASE/ LOWER :.....

UCASE/ UPPER :.....

MID/ SUBSTRING :.....

LIMIT :.....

3. Suggest, if you want to extract the first few characters of a string, which SQL function would you use and why?

.....

.....

.....

.....

.....

4. Suggest, if you need to get the first three characters of a string in SQL, which function should you use?

.....

.....

.....

.....


Remember the keywords of string functions!

- SQL command
- Manipulate and retrieve data from string (text) values
- Return a single result

# STRING FUNCTIONS

1. Write an SQL statement to convert the name column in the Customers table to lowercase letters.

2. Construct an SQL statement to return the first 5 characters of the description column from the Products table.

3. Write an SQL statement to convert the email column in the Users table to uppercase letters.

4. Write an SQL query to display the first 10 rows from the Orders table, showing only the order_id and customer_id columns

5. Write an SQL query to retrieve the first 6 characters of the title column from the Books table, but only for the first 3 records.


✅ UPPER() – Converts to uppercase
Result: 'Ali' → 'ALI'
✅ LOWER() – Converts to lowercase
Result: 'ALI' → 'ali'
✅ LENGTH() – Counts number of characters
Result: 'Ali' → 3
✅ CONCAT() – Joins two strings
Result: 'Ali' + 'Zain' → 'Ali Zain'
✅ SUBSTRING() – Extracts part of a string
Result: 'Ameer' → 'Ame'

# STRING FUNCTIONS

Data are given in the table below.

Employees

| EmployeeID | FirstName | LastName |
|---|---|---|
| 1 | Muhammad Hamid | Abdullah |
| 2 | Tan John | Doe |
| 3 | Siti Fatimah | Ismail |

1. Write an SQL statement to find the length of each employees name from the employees table.

2. Write an SQL statement to extract the first three characters of each first name name from the employees table.

3. Write an SQL statement to convert all first and last name in the employees table to uppercase.

4. Write an SQL statement to convert all first and last name in the employees table to lowercase.


✅ **UPPER()**
SELECT UPPER(name)
FROM student;
✅ **LOWER()**
SELECT LOWER(name)
FROM student;
✅ **LENGTH()**
SELECT LENGTH(name)
FROM student;
✅ **CONCAT()**
SELECT
CONCAT(first_name, ' ',
last_name) AS full_name
FROM student;
✅ **SUBSTRING()**
SELECT
SUBSTRING(name, 1, 3)
FROM student;

# STRING FUNCTIONS

5 Write an SQL statement to extract the first 4 characters of each first name.

6 Write an SQL statement to extract the last 3 characters of each last name.

7 Write an SQL statement to extract a substring from the 4th character to the 6th character of each employees first name.


✅ **UPPER()**
SELECT UPPER(name)
FROM student;
✅ **LOWER()**
SELECT LOWER(name)
FROM student;
✅ **LENGTH()**
SELECT LENGTH(name)
FROM student;
✅ **CONCAT()**
SELECT
CONCAT(first_name, ' ',
last_name) AS full_name
FROM student;
✅ **SUBSTRING()**
SELECT
SUBSTRING(name, 1, 3)
FROM student;

# STRING FUNCTIONS

## MySQL Lab Exercise 4.2


Use an appropriate DBMS software

DOWNLOAD

https://bit.ly/restaurantsql

Import **restaurant.sql** file into your DBMS.

1. Retrieve all restaurant names in uppercase from the Restaurants table.
2. Retrieve all cuisine types in lowercase from the Restaurants table.
3. Extract the first 3 characters from the restaurant_name column for all restaurants.
4. Extract the first 4 characters of the location column for each restaurant.
5. Retrieve the first 3 restaurants from the Restaurants table.
6. Extract the first 4 characters of the cuisine_type column for all restaurants.
7. Retrieve the restaurant_name in uppercase and the location in lowercase for all restaurants.
8. Retrieve the restaurant_name in uppercase and extract the first 6 characters of the location column for all restaurants.
9. Retrieve the location of all restaurants in lowercase, but only the first 7 characters of each location.
10. Retrieve the first 2 restaurants in alphabetical order based on restaurant_name.

# JOINS

1. Define the role of joins in SQL.

.....

.....

.....

.....

2. State the usage for each of the joins below.

INNER JOIN :.....

LEFT JOIN :.....

RIGHT JOIN :.....

FULL JOIN :.....

3. Explain the role of primary and foreign keys play in establishing joins between tables.

.....

.....

.....

.....

.....

4. Explain the primary purpose of the ON clause in SQL joins.

.....

.....

.....

.....


Remember the keywords of joins!

- Combines columns from two or more tables
- Based on related columns
- Data retrieved from established relationship between tables

# JOINS

## INNER JOIN

Data are given in the table below.

Employees

| EmployeeID | Name | DepartmentID |
|---|---|---|
| 1 | Alice | 101 |
| 2 | Bob | 102 |
| 3 | Charlie | 103 |

Departments

| DepartmentID | DepartmentName |
|---|---|
| 101 | HR |
| 102 | IT |
| 104 | Marketing |

HELPFUL TIPS

Remember

- Only rows that have matching values in both tables are returned.
- If there's no match between the two tables, those rows are excluded from the result.

1. Write an SQL statement to perform INNER JOIN.

2. Based on the SQL statement in question 1, illustrate the output

3. Explain why the output is like that.

.....

.....

.....

.....

NOTES

SCAN ME!

# JOINS

## RIGHT JOIN

Data are given in the table below.

Orders

| OrderID | CustomerID | OrderDate |
|---|---|---|
| 101 | 1 | 2025-01-10 |
| 102 | 2 | 2025-01-11 |

Customers

| CustomerID | CustomerName |
|---|---|
| 1 | John |
| 3 | Sarah |

**1** Write an SQL statement to perform RIGHT JOIN.

**2** Based on the SQL statement in question 1, illustrate the output

**3** Explain why the output is like that.

.....

.....

.....

.....

HELPFUL TIPS

Remember

- All rows from the right table are included.
- Rows from the left table are included only if there is a match.
- If no match is found, NULL values are returned for the left table's columns.

NOTES

SCAN ME!

# JOINS

## LEFT JOIN

Data are given in the table below.

Books

| BookID | Title | AuthorID |
|---|---|---|
| 1 | SQL Basics | 201 |
| 2 | Advanced Python | 202 |

Authors

| AuthorID | AuthorName |
|---|---|
| 201 | Jane Smith |
| 203 | Robert Brown |

**1** Write an SQL statement to perform LEFT JOIN.

**2** Based on the SQL statement in question 1, illustrate the output

**3** Explain why the output is like that.

.....

.....

.....

.....


Remember

- All rows from the left table are included.
- Rows from the right table are included only if there is a match.
- If no match is found, NULL values are returned for the right table's columns.

# JOINS

## FULL JOIN

Data are given in the table below.

Students

| StudentID | Name | CourseID |
|---|---|---|
| 1 | Emma | 501 |
| 2 | Liam | 502 |
| 3 | Noah | NULL |

Courses

| CourseID | CourseName |
|---|---|
| 501 | Mathematics |
| 503 | Physics |

HELPFUL TIPS

Remember

- Includes all rows from both the left and right tables.
- If there's no match, NULL is shown for missing values.

1. Write an SQL statement to perform FULL JOIN.

2. Based on the SQL statement in question 1, illustrate the output

3. Explain why the output is like that.

.....

.....

.....

.....

NOTES

SCAN ME!

# JOINS

Data are given in the table below.

Tourists

| TouristID | Name | Country | TourID |
|---|---|---|---|
| 1 | John | USA | 101 |
| 2 | Maria | Canada | 102 |
| 3 | Ahmed | Egypt | 103 |
| 4 | Li | China | 104 |
| 5 | Anna | UK | NULL |

Tours

| TourID | TourName | Duration |
|---|---|---|
| 101 | City Tour | 3 days |
| 102 | Adventure Tour | 7 days |
| 103 | Cultural Tour | 5 days |
| 104 | Beach Resort | 10 days |
| 105 | Mountain Trek | 4 days |


1. Write an SQL statement to display the tourists who are participating in a tour, including those who are not currently assigned to any tour.List all tourists and the tours they are associated with.

2. Based on the SQL statement in question 1, illustrate the output.

# JOINS

3. Write an SQL statement to display a list all tours along with the tourists who are part of them. Include tours even if no tourists are assigned to them.

4. Based on the SQL statement in question 3, illustrate the output.

5. Write an SQL statement to display the tourists that are assigned to tours listed in the Tours table. Exclude tourists who are not associated with any tour.

6. Based on the SQL statement in question 5, illustrate the output.

# JOINS

7. Write an SQL statement to display all tours, including the tours which are currently do not have any tourists assigned to them.

8. Based on the SQL statement in question 7, illustrate the output.

# JOINS

Data are given in the table below.

Products

| ProductID | ProductName | Category | Price (MYR) |
|---|---|---|---|
| 1 | Nasi Lemak | Food | 7 |
| 2 | Teh Tarik | Drink | 3 |
| 3 | Roti Canai | Food | 4 |
| 4 | Char Kway Teow | Food | 8 |
| 5 | Milo | Drink | 2 |

Sales

| SaleID | ProductID | Quantity | Date |
|---|---|---|---|
| 1 | 1 | 50 | 2025-01-01 |
| 2 | 2 | 100 | 2025-01-02 |
| 3 | 3 | 75 | 2025-01-02 |
| 4 | 4 | 30 | 2025-01-03 |
| 5 | 5 | 120 | 2025-01-04 |


1. Write an SQL statement to display a list of all products and their associated sales data, including products that have no sales recorded.

2. Based on the SQL statement in question 1, illustrate the output.

# JOINS

3. Write an SQL statement to display all products that are sold, and the corresponding sales data for those products. Exclude products that are not sold.

4. Based on the SQL statement in question 3, illustrate the output.

5. Write an SQL statement to display a list of products and their corresponding sales records, including products that have not been sold and sales that do not correspond to any product.

6. Based on the SQL statement in question 5, illustrate the output.

NOTES

SCAN ME!

# JOINS

7. Write an SQL statement to display a list of all sales and their associated products data, including sales that have no product recorded.

8. Based on the SQL statement in question 7, illustrate the output.

# JOINS

Data are given in the table below.

Authors

| AuthorID | AuthorName | Country |
|---|---|---|
| 1 | Aza Abdullah | Malaysia |
| 2 | John Doe | USA |
| 3 | Fatimah Ismail | Malaysia |
| 4 | Jane Smith | UK |

Books

| BookID | Title | AuthorID | Price |
|---|---|---|---|
| 101 | SQL Basics | 1 | 40 |
| 102 | Advanced SQL | 2 | 50 |
| 103 | Database Design | 3 | 45 |
| 104 | Data Analysis | 1 | 60 |

Sales

| SaleID | BookID | SaleDate | Quantity |
|---|---|---|---|
| 201 | 101 | 2024-01-01 | 10 |
| 202 | 102 | 2024-01-02 | 5 |
| 203 | 105 | 2024-01-03 | 8 |

# JOINS

1

| Title | AuthorName | SaleDate | Quantity |
|---|---|---|---|
| SQL Basics | Aza Abdullah | 2024-01-01 | 10 |
| Advanced SQL | John Doe | 2024-01-02 | 5 |
| NULL | NULL | 2024-01-03 | 8 |

Write an SQL statement to display the output as above.

2

| Title | AuthorName | SaleDate | Quantity |
|---|---|---|---|
| SQL Basics | Aza Abdullah | 2024-01-01 | 10 |
| Advanced SQL | John Doe | 2024-01-02 | 5 |
| Database Design | Fatimah Ismail | NULL | NULL |
| Data Analysis | Aza Abdullah | NULL | NULL |
| NULL | NULL | 2024-01-03 | 8 |

Write an SQL statement to display the output as above.

# JOINS

3

| Title | AuthorName | SaleDate | Quantity |
|---|---|---|---|
| SQL Basics | Aza Abdullah | 2024-01-01 | 10 |
| Advanced SQL | John Doe | 2024-01-02 | 5 |
| NULL | NULL | 2024-01-03 | 8 |

Write an SQL statement to display the output as above.

4

| Title | AuthorName | SaleDate | Quantity |
|---|---|---|---|
| SQL Basics | Aza Abdullah | 2024-01-01 | 10 |
| Advanced SQL | John Doe | 2024-01-02 | 5 |
| Database Design | Fatimah Ismail | NULL | NULL |
| Data Analysis | Aza Abdullah | NULL | NULL |

Write an SQL statement to display the output as above.

# JOINS

## MySQL Lab Exercise 4.3


Use an appropriate DBMS software

Import **medicine.sql, supplier.sql and orders.sql** file into your DBMS.

1. Retrieve a list of medicines along with their suppliers' names.
2. Retrieve all medicines along with their suppliers' names. Include medicines that do not have a supplier.
3. Retrieve all suppliers and the medicines they supply. Include suppliers who do not supply any medicines.
4. Retrieve all medicines and suppliers, including medicines without suppliers and suppliers without medicines.
5. Retrieve a list of orders along with the names of the medicines ordered.
6. Retrieve a list of all orders and their associated medicines. Include orders that do not have a valid medicine ID.
7. Retrieve a list of medicine names, their suppliers' names, and supplier locations.

DOWNLOAD

https://bit.ly/suppliersql
https://bit.ly/ordersql
https://bit.ly/medicinesql

# SQL COMMAND FOR SECURITY PURPOSES

1. State the usage for each of the SQL commands below.

   GRANT : .....

   REVOKE : .....

2. State the types of privileges can be granted to users in SQL. Provide examples.

   .....

   .....

   .....

   .....

   .....

3. Can the REVOKE command remove multiple privileges at once? Explain.

   .....

   .....

   .....

   .....

4. Who can use the GRANT command to assign privileges on a database object?

   .....

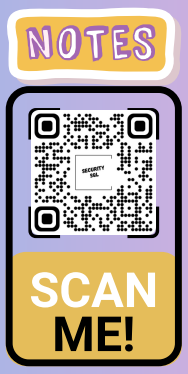
   .....

   .....

   .....


Remember the keywords of GRANT & REVOKE!

- GRANT: Provide privilages
- REVOKE: Remove privilages



# SQL COMMAND FOR SECURITY PURPOSES

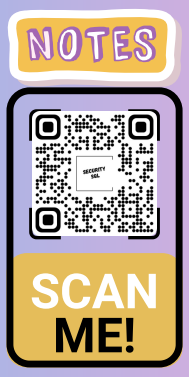
Data are given in the table below.

Employees

| Column | Data Type |
|---|---|
| employee_id | INT |
| name | VARCHAR(50) |
| department | VARCHAR(50) |
| salary | DECIMAL(10,2) |

Departments

| Column | Data Type |
|---|---|
| department_id | INT |
| department_name | VARCHAR(50) |
| location | VARCHAR(50) |




1. User1 should have the ability to SELECT data from the Employees table. Write an SQL statement to grant the required privilege.

2. User2 should have the ability to both SELECT and UPDATE data on the Departments table. Write an SQL statement to grant the required privilege.

3. User3 should be able to SELECT, INSERT, and UPDATE data from the Employees table. Write an SQL statement to grant the required privilege.

# SQL COMMAND FOR SECURITY PURPOSES

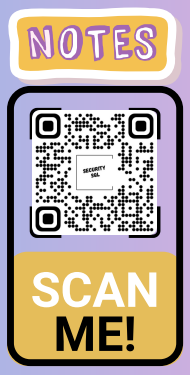
4. User4 should have all privileges on the Departments table. Write an SQL statement to grant the required privilege.

5. Revoke UPDATE privilege from user3 on the Employees table. Write an SQL statement to revoke the privilege.

6. Revoke all privileges from user4 on the Departments table. Write an SQL statement to revoke the privilege.

7. Revoke the INSERT privilege from both user7 and user8. Write an SQL statement to revoke the privilege.

8. Revoke the INSERT and UPDATE on the Employees table from both user2 and user3. Write an SQL statement to revoke the privilege.




SCAN ME

# CHAPTER 5

## Database Security

# DATABASE SECURITY, THREAT & COUNTERMEASURE

1. Describe database security.

.....

.....

.....

2. Explain **THREE (3)** importance of database security.

.....

.....

.....

.....

.....

.....

.....

3. Explain the three primary components of database security.

.....

.....

.....

.....

.....

.....

4. List T**HREE (3)** common types of database security threats.

.....

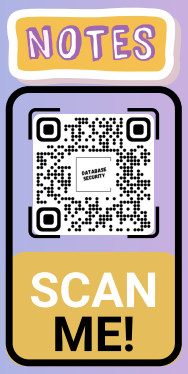
.....

.....

.....


Remember the keywords of database security!

- Protective measures
- Safeguard data stored within a database



# DATABASE SECURITY, THREAT & COUNTERMEASURE

5 Explain database encryption.

.....

.....

.....

.....

6 State the difference between authentication and authorization in database security.

.....

.....

.....

.....

.....

.....

.....

7 Describe the concept of SQL injection.

.....

.....

.....

8 Describe why is regular database backup a critical countermeasure?

.....

.....

.....

.....

9 List **TWO (2)** best practices for securing database passwords.

.....

.....

HELPFUL TIPS

Benefits of Encryption :

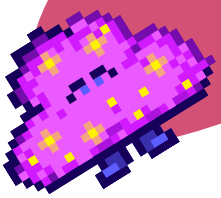
- Protects private data
- Prevents data theft
- Keeps data safe during transfer
- Only trusted people can access data
- Helps follow data protection laws
- Detects data changes (integrity)
- Builds user trust

NOTES

SCAN ME!

QUIZ time!

GAME TIME



# DEMONSTRATION

## Additional Practice Using MySQL

# DEMONSTRATION


Use an appropriate DBMS software

## DEMO 1

### Task 1: ERD design

Draw an Entity-Relationship Diagram (ERD) for a Online Course Management System with cardinality. A Student can enroll in multiple Courses. A Course can have many Students enrolled in it. An instructor can teach multiple Courses. Each Course is taught by one Instructor. Each enrolment involves one Student and one Course.

### Task 2: Data Insertion

Construct an SQL statement to create table and insert data given below.

Menu

| MenuID | Name | Price | Quantity | Spicy |
|---|---|---|---|---|
| 1 | Nasi Lemak Antarabangsa | 15.90 | 5 | Yes |
| 2 | Mee Goreng | 9.90 | 4 | No |
| 3 | Mee Sup | 11.90 | 11 | No |
| 4 | Nasi Ayam Penyet | 17.50 | 7 | Yes |

### Task 3: Basic Query

Construct an SQL statement to display the menus, its price and quantity.

# DEMONSTRATION

**Task 4: Filtering and Advanced Query**

- Construct an SQL statement to display all information about the Menu where the quantity is less than 10.
- Construct an SQL statement to display all information about the menu which price between 9.00 to 15.00.
- Construct an SQL statement to display all information about the Menu in ascending order by quantity.
- Construct an SQL statement to display all information about the Menu in descending order by quantity.
- Construct an SQL statement to display Name, Price and Spicy information on Menu which is spicy.
- Construct an SQL statement to display all information about the menu which it's cuisine name is "Mee Sup".

**Task 5: Aggregation**

- Construct an SQL statement to display total numbers of Menu.
- Construct an SQL statement to display total quantity as "Jumlah Quantity".
- Construct an SQL statement to display name and price for highest price.
- Construct an SQL statement to display name and quantity for lowest quantity.

# DEMONSTRATION


Use an appropriate DBMS software

## DEMO 2

### Task 1: ERD design

Draw an Entity-Relationship Diagram (ERD) for a Library Management System with cardinality. A library has multiple members. Each member can borrow multiple books, and each book can be borrowed by multiple members. Each book belongs to one genre, but a genre can have multiple books. Each member can only be registered at one library, but a library can have many members. Design an ERD that shows these relationships.

### Task 2: Data Insertion

Construct an SQL statement to create table and insert data given below.

Electronics

| ProductID | ProductName | Price (RM) | Quantity |
|---|---|---|---|
| 1 | Smartphone X | 1999.90 | 8 |
| 2 | Laptop Z | 3499.90 | 5 |
| 3 | Wireless Headphones Y | 299.90 | 12 |
| 4 | Smartwatch Q | 599.90 | 6 |

### Task 3: Basic Query

Construct an SQL statement to display the product name, its price and quantity.

# DEMONSTRATION

**Task 4: Filtering and Advanced Query**

- Construct an SQL statement to display all information of the electronics product.
- Construct an SQL statement to display product name and price of the electronics product.
- Construct an SQL statement to display all information of the electronics product where the quantity is less than 10.
- Construct an SQL statement to display product name and quantity of the electronics product in ascending order by quantity.
- Construct an SQL statement to display product name and price of the electronics product where the price between 200.00 to 600.00.

**Task 5: Aggregation**

- Construct an SQL statement to calculate the total quantity of all products in the table.
- Construct an SQL statement to find the average price of all the products.
- Construct an SQL statement to determine the maximum price of the products in the table..
- Construct an SQL statement to count the total number of products in the table.
- Find the total quantity for products priced above RM 1000.
- Find the total number of products where the quantity exceeds 5.

# DEMONSTRATION


Use an appropriate DBMS software

## DEMO 3

### Task 1: ERD design

Draw an Entity-Relationship Diagram (ERD) for a Hotel Management System with cardinality. Each room can be reserved by multiple guests over time, but a guest can only reserve one room at a time. Each reservation is made by one guest, but a guest can have multiple reservations. Each reservation is associated with one payment, but a guest can make multiple payments over time for different reservations. Each room belongs to one category, but a category can have multiple rooms. Build an ERD to represent these relationships

### Task 2: Data Insertion

Construct an SQL statement to create table and insert data given below.

Cloth

| ClothingID | ClothingName | Price (RM) | Quantity |
|---|---|---|---|
| 1 | T-Shirt Black | 49.90 | 15 |
| 2 | Jeans Classic | 99.90 | 10 |
| 3 | Hoodie Gray | 79.90 | 8 |
| 4 | Jacket Leather | 299.90 | 3 |

### Task 3: Basic Query

Construct an SQL statement to display the cloth name, its price and quantity.

# DEMONSTRATION

**Task 4: Filtering and Advanced Query**

- Construct an SQL statement to display all information of the cloth.
- Construct an SQL statement to display product name and price of the cloth.
- Construct an SQL statement to display all information of cloth where the cloth name starts with “J”.
- Construct an SQL statement to display product name and quantity of the cloth which has the highest price.
- Construct an SQL statement to display all information of the cloth which has the lowest quantity.

**Task 5: Aggregation**

- Construct an SQL statement to calculate the total quantity of all products in the table.
- Construct an SQL statement to find the average price of all the products.
- Construct an SQL statement to determine the maximum quantity of the products in the table..
- Construct an SQL statement to calculate the total number of quantity in the table.
- Find the number of product with priced above RM70.00
- Round the price to 0 decimal place.

# REFERENCES

## BOOKS

McGrath, M. (2020). SQL in easy steps. In Easy Steps Limited. (ISBN - 978-1-84078-902-7)

Forta, B. (2020). Sams Teach Yourself SQL in 10 Minutes, Fifth Edition. Pearson Education, Inc. (ISBN - 978-0-13-518279-6)

Elmasri, R. & Navathe, S. (2022). Fundamentals of Database Systems: 7th ed. Pearson.

Anthony DeBaross. (2022). Practical SQL, A Beginner's Guide to Storytelling with Data: 2nd ed. No Starch Press.

Awang, N. F., & Saad, A. M. (2024). Database Management System Workbook. Politeknik Nilai. (e-ISBN 978-967-2742-40-1)

## WEBSITES

Parewa Labs Pvt. Ltd. (n.d.). Learn SQL. Programiz. Retrieved December 10, 2024, from https://www.programiz.com/

Refsnes Data. (n.d.). SQL Tutorial. W3schools. Retrieved December 12, 2024, from https://www.w3schools.com/sql/default.asp

Chapter 9: Structured Query Language (SQL). (n.d.). KnowledgeBoat. Retrieved January 20, 2025, from https://www.knowledgeboat.com/learn/class-12-ncert-computer-science-cbse/solutions/EegX1/structured-query-language

**JABATAN PERDAGANGAN**
**Politeknik Nilai**
**Kompleks Pendidikan Bandar Enstek**
**71760 Bandar Enstek**
**Negeri Sembilan**

Nurul Fadhlun binti Awang
fadhlun@polinilai.edu.my

Azalinda binti Mat Saad
azalinda_matsaad@polinilai.edu.my

Smart
DBMS
WORKBOOK

SMART DBMS WORKBOOK

e ISBN 978-967-2742-52-4

POLITEKNIK NILAI