**TABLE OF CONTENT**

1. Information Systems

2. Components of an Information System

3. The role of Information Systems in Business

4. The Role of IT in Businesses Today

5. The Interdependence Between Business and Information Technology

6. Achieving strategic business objectives through IT

7. Entity–Relationship Diagram

8. ERD components

9. Relationship between entities

10. Data

11. Information

12. Database

13. Database Management System (DBMS)

14. Relational Database

15. Table

16. Rows

17. Columns

18. Writing SQL Statements

19. File-Based Approach

20. Problems with file system data management

21. Programming in 3GL problems

22. Files Processing System

23. What is a Database? Why do we need it?

24. Database vs File-Based Systems

25. DBMS - Functions

26. DBMS - Tasks

27. Types of DBMS

28. Relational Data Model

29. SQL

30. MySQL

31. XAMPP

32. Constraints - Keys

33. Key - Types

34. SQL - categories

35. CREATE, DROP, ALTER

36. SELECT, INSERT, UPDATE, DELETE

37. SQL – DDL & DML

38. SQL – Clause

39. SQL – Operators

40. Data Dictionary

41. Example of a data dictionary

42. Some more SQL clauses and aggregate functions

43. SQL joins

44. Backup and restore a DB using MySQL dump

45. Creating a dump file

46. Using dump file

47. Join query

DATABASE: 1ST SEMESTER





DATABASE: 1ST SEMESTER

**MODULE NAME: DATBASE**

**MODULE CODE: CC4057**

**INTRIDUCTION TO INFORMATION SYSTEM**

Informatics College Pokhara

***Submitted By:*  *Submitted To:***

**Student Name: Janak Devkota Sudip Dhakal**

**Student ID: N/A Module Leader**

**Group: C4 Database**

**Date: 19-12-2023**

**1. What is an Information System?**

- An information system is a set of interconnected components that collect, process, store, and disseminate data to support decision-making, coordination, and control in an organization.

- An information system can be hardware-based, software-based, or a combination of both.

*- Examples of information systems are enterprise resource planning (ERP) systems, customer relationship management (CRM) systems, and supply chain management systems*.

**2. Components of an Information System**

- An Information System comprises five components: hardware, software, data, people, and procedures.

- Hardware comprises physical components such as servers, network devices, computers, and mobile devices.

- Software refers to applications, operating systems, and databases that are used to manage, analyze, and process data.

- Data encompasses the raw facts, information, and knowledge that are stored and processed by the system.

- People represent the human capital involved in managing, operating, and using the information system.

- Procedures are the set of instructions that govern how the system operates and how the people interact with it.

**3. The role of Information Systems in Business**

- Information systems play a crucial role in business by providing a platform for data management, communication, and decision-making.

- Information systems are essential for increasing productivity, efficiency, and effectiveness in an organization.

- Information systems support various business functions such as finance, marketing, human resources, and operations.

- Information systems enable businesses to adapt to changing market and technological environments.

*- Example: A retail business uses an information system to manage inventory, sales, and customer data in real-time.*

DATABASE: 1ST SEMESTER

**4. The Role of IT in Businesses Today**

- Information technology (IT) is the backbone of modern businesses and facilitates the use of information systems.

- IT provides the infrastructure, tools, and techniques for managing, storing, and processing data.

- IT plays a critical role in delivering products and services to customers and enhancing customer satisfaction.

- IT helps organizations to innovate, optimize processes, and gain a competitive advantage in the market.

*- Example: A bank uses IT to provide online banking services, secure transactions, and automation of customer service.*

**5. The Interdependence Between Business and Information Technology**

- Business and IT are interdependent and rely on each other for success.

- Business defines the goals and objectives, and IT provides the technical infrastructure and resources that support achieving strategic objectives.

- Business and IT collaborate to ensure the alignment of technology with business processes and goals.

- Businesses invest in IT to gain a competitive advantage, improve productivity, and reduce costs.

*- Example: An e-commerce business invests in IT to develop scalable platforms, secure payment systems, and efficient supply chain management.*

**6. Achieving strategic business objectives through IT**

- IT plays a key role in achieving strategic business objectives by enabling innovation, optimization, and efficiency in business processes.

- IT helps businesses to respond to market demands, customer preferences, and technological change.

- IT facilitates collaboration among business functions and stakeholders.

- IT contributes to long-term success by increasing customer satisfaction, reducing costs, and improving productivity.

DATABASE: 1ST SEMESTER

*- Example: An automotive company uses IT to design, manufacture, and market vehicles, ensuring efficient inventory management, supply chain coordination, and customer engagement.*

**7. Entity - Relationship Diagram:**

- An entity-relationship diagram (ERD) is a visual representation of the entities, attributes, and relationships involved in a database.

- ERD is used for designing and denoting a database schema.

- ERD is an efficient way of depicting complex relationships in a database.

**8. ERD components:**

- An entity is an object or concept in a database that can be uniquely identified by its attributes.

- Attributes are characteristics or properties of entities.

- A relationship is an association between entities, and it is defined by its cardinality (one-to-one, one-to-many, many-to-one, or many-to-many).

- An ERD also includes connecting lines, keys, and cardinality indicators.

**9. Relationship between entities:**

- A relationship describes the connection between two entities, and the cardinality indicates the number of instances of one entity that is associated with another entity in a relationship.

- Cardinality is defined as either mandatory or optional, depending on whether the association is required or not.

**10. Application of ERD**

- ERDs are used to design and communicate about the structure of a database.

- ERD is used to identify and clarify relationships between entities.

DATABASE: 1ST SEMESTER

- ERD is also used to ensure data integrity and reduce redundant data.

*- Example: ERD can be used to design a database for an online shopping website where the customer can purchase items from different categories and the order is placed and tracked by the website*.

**11. ERD Design Rules:**

- An entity is represented by a rectangle.

- Attributes are represented by oval shapes attached to entities.

- A line represents a relationship between two entities.

- A primary key is represented by an underlined attribute, and a foreign key is represented by a double line pointing towards the primary key of another entity.

- The cardinality ratio is represented by a number (1, 0, or M) or a symbol (1, 0, or ∞).

**12. Symbols Used in ERD:**

- A rectangle represents an entity

- An oval represents an attribute

- A diamond represents a relationship or conditionality

- A circle represents a weak entity's identifier

- A line represents a relationship between entities.

**13. Benefits of ERD:**

- ERD provides a clear understanding of the problem domain and supports communication among stakeholders.

- ERD enables efficient data storage, retrieval, and manipulation.

- ERD promotes data integrity and consistency.

- ERD facilitates the development and maintenance of databases.

- *Example: ERD can be used to design a database for a university where students can enroll in courses, add or drop courses, and receive grades.*

DATABASE: 1ST SEMESTER

**14. What is a Database?**

- A database is an organized collection of data that is stored, managed, and accessed electronically.

- A database can include multiple tables, with each table defining a set of related data.

**15. Why do we need a Database?**

- Databases help to manage and organize large amounts of data.

- Databases improve data accuracy and consistency.

- Databases help to reduce data redundancy.

- Databases provide security and backup mechanisms.

- Databases allow for efficient querying, searching, and reporting of data.

**16. Database vs File-based Systems:**

- A file-based system stores data in individual files, while a database stores data in organized tables.

- A file-based system is limited in terms of scalability, security, and efficiency, while a database can handle large amounts of data, provide security and backup, and improve data accuracy and consistency.

**17. Database Management System (DBMS):**

- A DBMS is a software system that enables users to manage and access a database.

- A DBMS provides tools and interfaces for data storage, retrieval, and manipulation.

- A DBMS manages concurrent user access and provides security and backup mechanisms.

- Popular DBMS include MySQL, Oracle, Microsoft SQL Server, and PostgreSQL.

**18. Relational Data Model:**

- The relational data model is based on the concept of tables that are related to each other through common columns or keys.

- Tables are composed of rows and columns, where each row represents a unique record, and each column represents a data attribute.

DATABASE: 1ST SEMESTER

- The relationship between tables is defined by common keys, which link tables by identifying their related data.

**19. SQL (Structured Query Language):**

- SQL is a standard language used to manage relational databases.

- SQL provides the ability to create, retrieve, update, and delete data in a database.

- SQL consists of data definition language (DDL) statements for creating and modifying database objects and data manipulation language (DML) statements for accessing and modifying data.

**20. DBMS Functions:**

- A DBMS provides several functions such as data storage, retrieval, and manipulation.

- A DBMS ensures data integrity, concurrency control, and security.

- A DBMS enables backup, recovery, and data migration.

**21. DBMS Tasks:**

- DBMS tasks include database design, implementation, maintenance, optimization, and troubleshooting.

- DBA (database administrator) or a team manages these tasks.

**22. Types of DBMS:**

- There are different types of DBMS, including relational, object-oriented, graph, and NoSQL DBMS.

- Each type has its own advantages and disadvantages, and their selection depends on the application requirements.

**23. MySQL:**

- MySQL is an open-source relational DBMS that is widely used in web applications and other database-centric systems.

- MySQL is known for its scalability, speed, and reliability.

- MySQL supports various platforms and programming languages.

DATABASE: 1ST SEMESTER

- MySQL is used by companies such as Facebook, Google, and Twitter.

**24. Table:**

- A table is a collection of related data organized in rows and columns.

- A table has a defined data type for each column that ensures data consistency.

- A table can have one or more columns that uniquely identify each row, which is known as the primary key.

**25. Rows:**

- A row is a single record in a table that contains data for each column.

- A row is also referred to as a record, tuple, or instance.

- A row can be identified by a unique value in its primary key column.

**26. Columns:**

- A column is a data attribute or field in a table.

- A column has a defined data type and a column name that represents the attribute.

- A column can also be referred to as an attribute, a field, or a column header.

**27. Writing SQL Statements:**

- SQL statements are used to retrieve or modify data in a database.

- SQL statements consist of keywords, functions, and operators.

- SQL statements are executed by a DBMS.

DATABASE: 1ST SEMESTER

**28. Constraints - Keys:**

- Constraints are rules that govern the data in a database.

- Keys are attributes or columns that uniquely identify a row in a table.

- There are different types of keys, including primary, foreign, candidate, and super.

**29. SQL - Categories:**

- SQL has four primary categories: Data Definition Language (DDL), Data Manipulation Language (DML), Data Query Language (DQL), and Data Control Language (DCL).

- Each category serves a specific purpose and has a set of SQL statements.

**30. SQL – Clause:**

- A clause is a part of an SQL statement that specifies a particular data requirement.

- Common clauses include SELECT, FROM, WHERE, GROUP BY, and ORDER BY.

- Clauses are combined to form a complete SQL statement.

**31. SQL – Operators:**

- SQL operators are special symbols and words used to perform calculations and comparisons.

- Common operators include arithmetic operators (+, -, \*, /), logical operators (AND, OR, NOT), and comparison operators (=, >, <).

**32. Data Dictionary:**

- A data dictionary is a repository of metadata in a database.

- A data dictionary defines the structure, meaning, and usage of data elements.

- A data dictionary helps to ensure data consistency and quality.

**33. Backup and Restore:**

- Backup and restore are critical tasks for database management.

- Backup involves copying and storing a database to protect it from data loss.

DATABASE: 1ST SEMESTER

- Restore involves recovering a database from a backup in the event of data loss or corruption­­.

DATABASE: 1ST SEMESTER

Remening:-

Certainly! Let's integrate the additional 10 topics into the crash course:

---

\*\*30. File-Based Approach\*\*

- \*\*Disadvantages\*\*

- Involves issues with data redundancy and inconsistency.

- Challenging to maintain data integrity without a centralized system.

- Difficulty in managing large datasets efficiently.

- Limited security measures compared to modern database systems.

- Lacks the ability to provide complex queries and relationships.

- \*\*Advantages for Small Data Size\*\*

- Simplicity in implementation and maintenance for smaller datasets.

- Requires less computational resources for processing.

- Well-suited for applications with minimal data handling requirements.

- Cost-effective for small-scale projects and limited data storage needs.

- Provides straightforward data access and retrieval for small databases.

\*\*31. What is a Database? Why Do We Need It?\*\*

- \*\*Definition\*\*

- A structured collection of data organized for efficient storage and retrieval.

- \*\*Purpose\*\*

- Enables efficient management and organization of large volumes of data.

- Facilitates concurrent access by multiple users for various applications.

- Supports the storage of diverse data types, enhancing versatility.

- Essential for data-driven decision-making and business processes.

- Forms the backbone for applications ranging from websites to enterprise systems.

\*\*32. Database vs File Based Systems\*\*

- \*\*Advantages of Databases\*\*

- Centralized control and management of data.

- Improved data integrity through relational models and constraints.

- Enhanced security measures, including user access controls.

- Efficient querying and reporting capabilities for large datasets.

- Better organization and structure compared to file-based systems.

\*\*33. DBMS - Functions\*\*

- \*\*Data Definition\*\*

- Involves defining the structure of the database, specifying data types and constraints.

- Ensures uniformity in data storage and organization.

- Examples include defining tables, relationships, and constraints.

- \*\*Data Manipulation\*\*

- Covers operations like inserting, updating, and deleting data.

- Allows users to interact with and modify the content of the database.

- Key for maintaining accurate and up-to-date information.

- \*\*Data Retrieval\*\*

- Focuses on extracting specific information from the database.

- Utilizes queries to filter and retrieve relevant data.

- Enables users to obtain meaningful insights from the stored information.

- \*\*Data Administration\*\*

- Involves managing user access, security, and overall database integrity.

- Ensures that only authorized users can interact with the database.

- Maintains the consistency and reliability of data throughout its lifecycle.

\*\*34. DBMS - Tasks\*\*

- \*\*Create\*\*

- Involves creating database objects like tables, views, and indexes.

- Establishes the foundational structure for data storage and retrieval.

- \*\*Retrieve\*\*

- Entails querying the database to extract specific information.

- Utilizes SQL queries to filter and select relevant data.

- \*\*Update\*\*

- Modifies existing data in the database.

- Essential for reflecting changes in real-world scenarios.

- \*\*Delete\*\*

- Involves removing data from the database.

- Helps maintain data relevance and cleanliness.

- \*\*Ensure Integrity\*\*

- Encompasses tasks that guarantee the consistency and accuracy of data.

- Involves enforcing constraints, such as unique keys and foreign keys.

\*\*35. Types of DBMS\*\*

- \*\*Hierarchical\*\*

- Organizes data in a tree-like structure.

- Suitable for representing parent-child relationships.

- \*\*Network\*\*

- Utilizes a more flexible structure with interconnected data.

- Allows for more complex relationships between entities.

- \*\*Relational (Focus on Relational)\*\*

- Organizes data into tables with rows and columns.

- Ensures data integrity through relationships and constraints.

\*\*36. Relational Data Model\*\*

- \*\*Tables and Relationships\*\*

- Represents data as tables with rows and columns.

- Relationships establish connections between tables.

- \*\*Data Integrity\*\*

- Ensures the accuracy and consistency of data through keys and constraints.

- Guarantees that each piece of data is logically and structurally correct.

- \*\*Flexibility\*\*

- Provides a scalable structure that accommodates changing data requirements.

- Easily adaptable to various business scenarios and application needs.

\*\*37. SQL\*\*

- \*\*Structured Query Language\*\*

- Essential for managing and manipulating relational databases.

- Offers a standardized way to interact with databases.

- \*\*Data Operations\*\*

- Includes commands like SELECT, INSERT, UPDATE, DELETE for manipulating data.

- Enables users to define and retrieve data based on specific criteria.

- \*\*Declarative Syntax\*\*

- Utilizes a declarative syntax, focusing on what data to retrieve rather than how.

- Enhances readability and ease of use for database interactions.

\*\*38. MySQL and XAMPP\*\*

- \*\*MySQL\*\*

- A widely used relational database management system.

- Known for its reliability, performance, and ease of use.

- Commonly used in web development for data storage and retrieval.

- \*\*XAMPP\*\*

- A development environment that includes MySQL, Apache, PHP, and Perl.

- Provides a platform for local development, testing, and deployment.

- Ensures compatibility and ease of setup for web development projects.

\*\*39. Constraints - Keys\*\*

- \*\*Primary Key\*\*

- Uniquely identifies each record in a table.

- Ensures data integrity and uniqueness within the table.

- \*\*Foreign Key\*\*

- Establishes relationships between tables.

- Links data in one table to data in another, ensuring referential integrity.

- \*\*Unique Key\*\*

- Ensures that a column or combination of columns has unique values.

- Prevents duplication of data, maintaining consistency.

\*\*40. SQL - Categories\*\*

- \*\*DDL (Data Definition Language)\*\*

- Used for defining and modifying the structure of the database.

- Includes commands like CREATE, DROP, ALTER for managing database objects.

- \*\*DML (Data Manipulation Language)\*\*

- Focuses on manipulating and querying data within the database.

- Includes commands like SELECT, INSERT, UPDATE, DELETE for data operations.

\*\*41. SQL – Clause\*\*

- \*\*Filtering Data\*\*

- SQL clauses, such as WHERE, filter data based on specified conditions.

- WHERE clause is essential for narrowing down query results.

- \*\*Sorting Data\*\*

- ORDER BY clause arranges query results in ascending or descending order.

- Facilitates organized presentation of data for analysis.

\*\*42. SQL – Operators\*\*

- \*\*Comparison Operators\*\*

- Used for comparing values in SQL statements (e.g., =, <>, <, >).

- Essential for filtering and conditionally retrieving data.

- \*\*Arithmetic Operators\*\*

- Perform mathematical operations on numeric data (e.g., +, -, \*, /).

- Useful for calculations within SQL statements.

- \*\*Logical Operators\*\*

- Combine conditions in SQL statements (e.g., AND, OR, NOT).

- Enable the creation of complex queries by specifying multiple criteria.

\*\*43. Data Dictionary\*\*

©created by: janak devkota

- \*\*Definition\*\*

- A repository of information about data in a database.

- Contains metadata, including data definitions, constraints, and relationships.

- \*\*Organizing Information\*\*

- Lists data elements, their attributes, and relationships in a structured manner.

- Enhances understanding and documentation of the database structure.

\*\*44. Example of a Data Dictionary\*\*

- \*\*Content\*\*

- Includes details like data

element names, descriptions, data types, and constraints.

- Defines the structure of tables, relationships, and key constraints.

- \*\*Reference Guide\*\*

- Serves as a reference guide for developers, administrators, and users.

- Enhances collaboration by providing a standardized understanding of data.

\*\*45. More SQL Clauses and Aggregate Functions\*\*

- \*\*Additional Clauses\*\*

- Includes clauses like LIMIT for restricting result set size.

- DISTINCT eliminates duplicate rows from the query result.

- \*\*Aggregate Functions\*\*

- Perform calculations on data (e.g., AVG, SUM, MIN, MAX).

- Facilitate summarization and analysis of large datasets.

\*\*46. SQL Joins\*\*

©created by: janak devkota

- \*\*Combining Data\*\*

- SQL joins merge rows from two or more tables based on related columns.

- Types include INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN.

- \*\*Relationships\*\*

- Establish connections between tables, enabling comprehensive data retrieval.

- Essential for querying data from related tables in a relational database.

\*\*47. Backup and Restore a DB Using MySQL Dump\*\*

- \*\*Creating a Dump File\*\*

- Involves exporting the structure and data of a database for backup.

- Ensures a snapshot of the database's state for recovery purposes.

- \*\*Using Dump File\*\*

- Restoring a database using a dump file brings it back to a previous state.

- Crucial for recovering data in the event of accidental deletion or system failure.

- \*\*Join Query\*\*

- Involves retrieving data from multiple tables using SQL JOIN statements.

- Enables the creation of comprehensive queries for data analysis.

©created by: janak devkota

| **Topic** | **MySQL Commands/Queries** |
| --- | --- |
| **Database Operations** |  |
| Show Databases | SHOW DATABASES; |
| Use Database | USE database\_name; |
| Show Tables | SHOW TABLES; |
| **CREATE, DROP, ALTER** |  |
| Create Table | CREATE TABLE table\_name (id, name, and so on with required constraints eg: PRIMARY KEY, INT, VARCHAR(), UNIQUE, DEFAULT, NOT NULL, AUTO\_INCREMENT, FOREIGN KEY(column\_name) REFERENCES table\_name(column\_name)); |
| Describe Table | DESCRIBE table\_name; |
| Drop Table and Database | DROP DATABASE db\_name;  DROP TABLE tb\_name; |
| Alter Table | sql ALTER TABLE table\_name ADD COLUMN column\_name; ALTER TABLE tb\_name MODIFY COLUMN column\_dtls; ALTER TABLE tb\_name DROP COLUMN column\_name; ALTER TABLE tablename RENAME to newOne; TRUNCATE TABLE table\_name; |
| **Data Manipulation** |  |
| Insert and Select | INSERT INTO table\_name (column\_name, column\_name, column\_name,) VALUES("varchar", int, 'DATE\_YY/M/DAY'); SELECT \* FROM table\_name; SELECT column\_name, column\_name FROM table\_name; |
| Update | UPDATE tb\_name SET column1\_name = value1, column2\_name = value2..... WHERE some\_column = some\_value; |
| Delete | DELETE FROM tb\_name; |
| **SQL Clauses** |  |
| WHERE-Clause | SELECT \* FROM tb\_name WHERE column\_name <operators> <value>; |
| BETWEEN-Clause | SELECT \* FROM tb\_name WHERE address BETWEEN "banepa" AND "ktm";  SELECT \* FROM tb\_name WHERE price BETWEEN 200 AND 500; |
| IN-Clause | SELECT \* FROM tb\_name WHERE customer\_id IN(1,2,3,4); |
| LIKE-Clause | SELECT \* FROM tb\_name WHERE address LIKE "k%"; SELECT \* FROM tb\_name WHERE address LIKE "%u"; SELECT \* FROM tb\_name WHERE address LIKE "%it%"; |
| **Aggregate Functions** |  |
| SUM(), MAX(),  MIN(), AVG() | SELECT SUM(price) FROM table\_name;  SELECT MAX(price) FROM table\_name;  SELECT MIN(price) FROM table\_name;  SELECT AVG(price) FROM table\_name; |
| **ORDER BY-Clause** | SELECT \* FROM tb\_name ORDER BY name, price;  SELECT \* FROM tb\_name ORDER BY name, price DESC;  SELECT \* FROM tb\_name ORDER BY name, price DESC LIMIT 1;  SELECT \* FROM tb\_name ORDER BY name, price DESC LIMIT 3; |
| **Arithmetic Operations** | SELECT 2 + 2 AS sum;  SELECT 5 - 2 AS subs; SELECT 2 \* 2 AS Multiply; SELECT 2 / 2 AS Div;  SELECT 2 % 2 AS remainder;  SELECT c1, c2, price, price - (price \* 0.10) AS new\_price FROM items; |
| **DISTINCT-Clause** | SELECT DISTINCT(price, name, id, address) FROM tb\_name; |
| **COUNT-Clause** | SELECT COUNT(\*) AS total\_orders FROM tb\_name; SELECT COUNT(id) AS total\_buyers FROM tb\_name; |
| **GROUP BY-Clause** | SELECT column\_name, COUNT(\*) AS total\_orders FROM tb\_name GROUP BY column\_name;  SELECT column\_name, COUNT(\*) AS total\_sold FROM tb\_name GROUP BY column\_name ORDER BY total\_sold DESC; |
| **HAVING-Clause** | SELECT column\_name, COUNT(\*) AS total\_orders FROM tb\_name GROUP BY column\_name HAVING total\_sold > 1; |
| **Join Operations** |  |
| Left Join | SELECT \*  FROM table1  LEFT JOIN table2 ON table1.column = table2.column; |
| Right Join | SELECT \*  FROM table1  RIGHT JOIN table2 ON table1.column = table2.column; |
| Full Outer Join | SELECT \*  FROM table1  LEFT JOIN table2 ON table1.column = table2.column  UNION ALL  SELECT \* FROM table1  RIGHT JOIN table2 ON table1.column = table2.column  WHERE table1.column IS NULL; |
| Joining Multiple Tables | SELECT \*  FROM table1  JOIN table2 ON table1.column = table2.column JOIN table3 ON table2.column = table3.column; |

**THE END**

©created by: janak devkota